



10 de septiembre de 2021
Ficha N° 20 DOTDOTPWN
CSIRT DE GOBIERNO

Comando de la semana “DOTDOTPWN”

I. CONTEXTO

Este documento, denominado, en esta oportunidad, “DOTDOTPWN ”, tiene como objetivo ilustrar sobre una herramienta que puede ser de utilidad para el lector, a objeto de ir potenciando las capacidades locales de autochequeo, detección simple de vulnerabilidades que están expuestas a internet en sus sitios o sistemas web y, a su vez, la obtención de una verificación de la subsanación de aquellas que se les han sido reportadas, facilitando la interacción con el CSIRT de Gobierno. El objetivo no es reemplazar una auditoria de código o evaluación de vulnerabilidades, sino que establecer capacidades básicas de chequeo y obtención de información de manera rápida para temas específicos, como por ejemplo la verificación de la subsanación de alertas o vulnerabilidades reportadas por “CSIRT GOB CL”. Todas estas herramientas al contar con la posibilidad de ser usadas desde una línea de comando, permiten en algún grado la integración dentro de script's o programas escritos en lenguajes que facilitan la automatización tales como PERL, AWK, Shell Scripting¹, Expect, Python, C, C#, C++, Golang, JavaScript, PowerShell, Ruby, Java, PHP, Elixir, Elm, Go, Dart, DLang, Pony, TypeScript, Kotlin, Nim, OCaml, Q#², Reason, Rust (RustyBuer), Swift, entre otros, con miras a automatizar estas actividades y así poder invertir el tiempo de los especialistas o analistas en el estudio e investigación de los datos para encontrar los problemas relevantes y descartar los falsos positivos.

Es importante que conozca al menos lo básico de los lenguajes más nuevos o no convencionales, pues se ha detectado que los desarrolladores de malware van incorporándolos como estrategia de ofuscación, para dificultar la detección y análisis que proveen las soluciones de seguridad.

Solo a modo de curiosidad se comparte un gráfico en el que se muestra el resultado de una encuesta entre muchos desarrolladores, dejando ver que lenguajes son más queridos/temidos (primer gráfico) y luego cuales son los más preferidos³ (segundo gráfico).

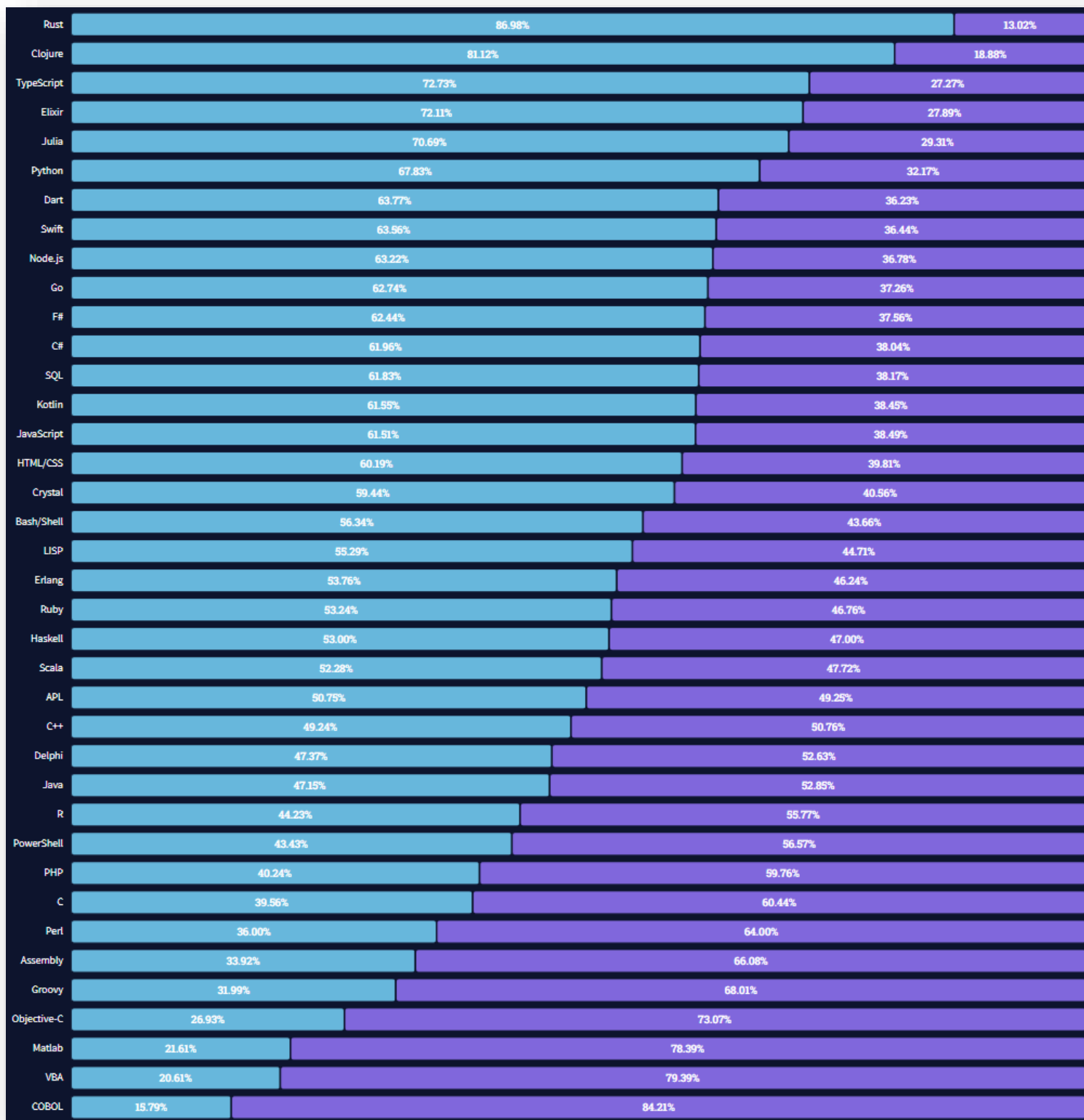
¹ <https://scis.uohyd.ac.in/~apcs/itw/UNIXProgrammingEnvironment.pdf>

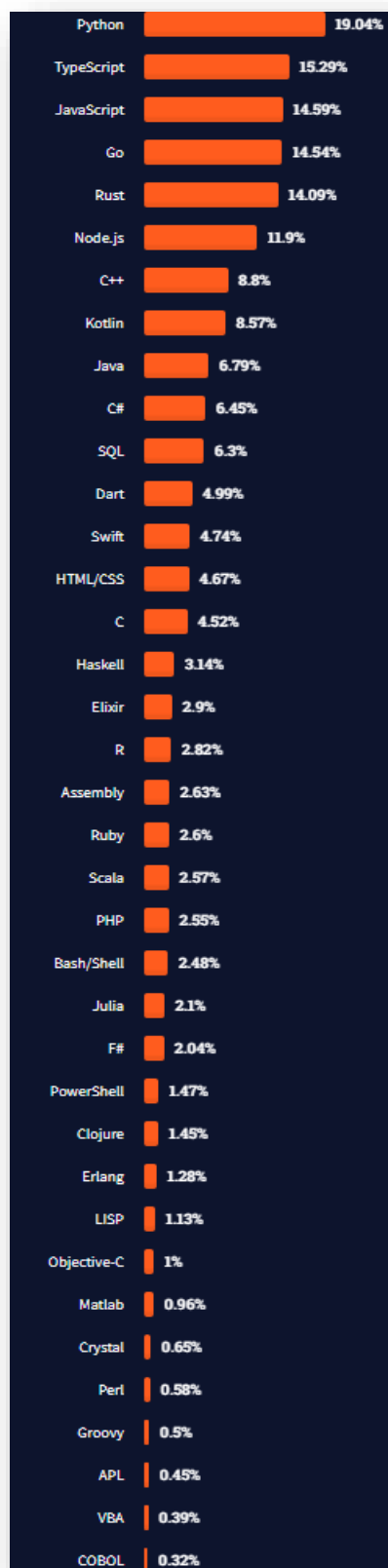
² <https://github.com/Microsoft/QuantumKatas/>

³ <https://insights.stackoverflow.com/survey/2021#most-loved-dreaded-and-wanted-language-love-dread>



Al final de este documento se presenta, a modo de curiosidad, el tradicional “Hola, Mundo” escrito en algunos de estos lenguajes, con el objetivo de motivar al lector a conocerlos, estudiarlos y aplicarlos a sus entornos de trabajo.







II. INTRODUCCIÓN

Una de las tareas regulares que un encargado de ciberseguridad debe realizar es la ENUMERACIÓN o RECONOCIMIENTO. La enumeración es una actividad de reconocimiento en la cual se consigue información de usuarios, grupos o dispositivos, dominios relacionados y demás servicios relacionados con un determinado activo expuesto a Internet.

Conocer esta información es importante, pues es lo que un hacker está haciendo en sus primeros pasos para llevar adelante un ataque en etapas posteriores.

En este sentido es importante tener en perspectiva el concepto de Cyber Kill Chain.

La Cyber Kill Chain, es una secuencia de los pasos que en general siguen los ciberdelincuentes cuando atacan nuestros sitios o sistemas expuestos en Internet:

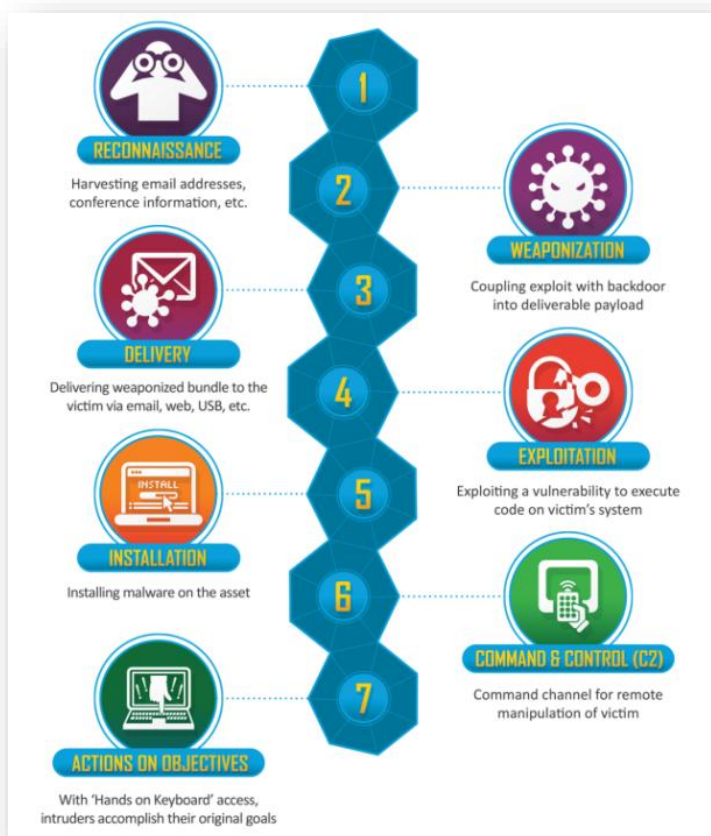


Ilustración 1 Cyber Kill Chain by Lockheed Martin

1) **Reconocimiento**: el intruso selecciona el objetivo, lo investiga e intenta identificar las vulnerabilidades en la red objetivo.

2) **Armamento**: el intruso crea un arma de malware de acceso remoto, como un virus o un gusano, adaptada a una o más vulnerabilidades.

3) **Entrega**: el intruso transmite el arma al objetivo (por ejemplo, a través de archivos adjuntos de correo electrónico, sitios web o unidades USB).

4) **Explotación**: se activa el código del programa del arma de malware, que toma medidas en la red objetivo para aprovechar la vulnerabilidad.



- 5) Instalación: el arma de malware instala un punto de acceso (por ejemplo, "puerta trasera") que puede utilizar un intruso.
- 6) Comando y control: el malware permite al intruso tener acceso persistente "con las manos en el teclado" a la red de destino.
- 7) Acciones sobre el objetivo: el intruso toma medidas para lograr sus objetivos, como la exfiltración de datos, la destrucción de datos o el cifrado para obtener un rescate.



funcionarios, datos de shodan/censys, detalles de nuestros certificados SSL, redes sociales, github dorks, etc. Con esta información se preparan para la siguiente fase de la cyber kill chain.

Al reducir la información innecesaria que se expone a internet, estamos dificultando el accionar de los cracker.

NOTA IMPORTANTE 1: Dado que es relevante un buen manejo de los comandos básicos de Linux, tanto para posteriores manejos de los datos o archivos como para usos de la información resultante de la ejecución de los comandos, es que el comité editorial decidió que se incluya en esta edición y en las subsiguientes un anexo de comandos Linux que son de utilidad para moverse en este sistema operativo. Se sugiere dominarlos todos para facilitar el acceso y manipulación de la información. En futuras ediciones se irán incorporando nociones más avanzadas sobre el uso de estos comandos para procesamiento de archivos, procesos, y de sus usos en scripting.

Vea anexo I: Comandos básicos de Linux

NOTA IMPORTANTE 2: Dado que un altísimo porcentaje de los equipos de usuarios y servidores operando en un entorno Windows, el comité editorial ha decidido ir incorporando “tips” para este entorno computacional.

Vea anexo II: Comandos o aplicativos básicos para Windows: TCPView

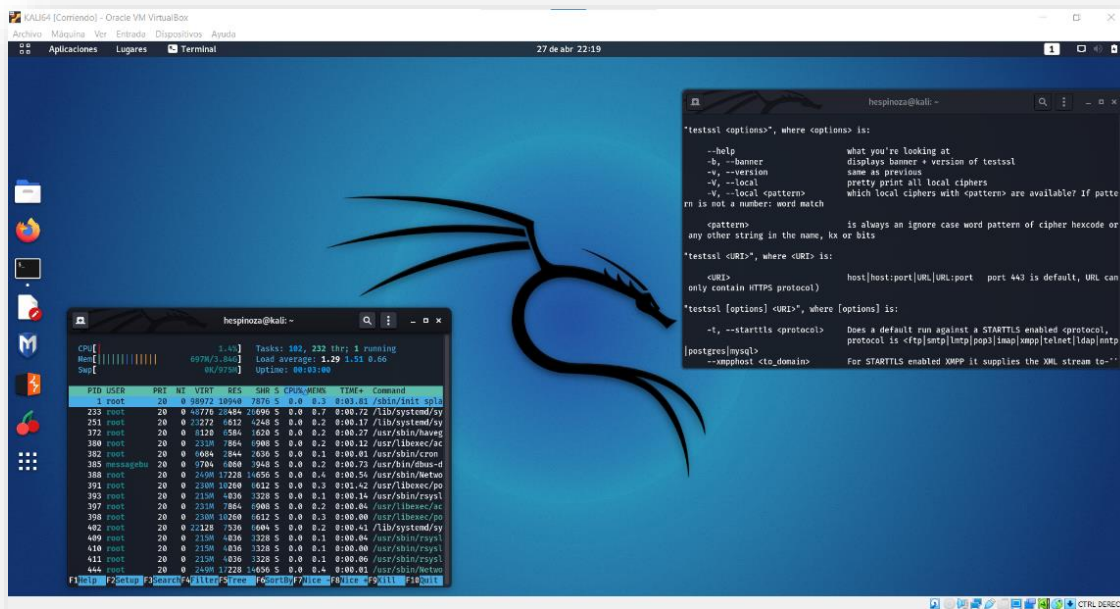
Bonus Track: Ejemplos muy básicos y simples del clásico “Hello, World!” escrito en diferentes lenguajes. El objetivo es despertar su curiosidad por estos lenguajes y ojalá se entusiasme y emprenda la cruzada de aprenderlo en profundidad, para luego aplicarlo en su quehacer cotidiano y dentro de lo posible comparta sus conocimientos con la comunidad.



III. PASO A PASO

PASO 1: Un entorno adecuado para trabajar.

Primero debe contar con una distribución de Kali⁴ Linux funcionando ya sea en una máquina física o en una máquina virtual⁵.



Instalación de Kali Linux

La instalación de Kali Linux (arranque único) en su computadora es un proceso sencillo. Esta guía cubrirá la instalación básica (que se puede realizar en una máquina virtual invitada o sobre un equipo entero), con la opción de cifrar la partición. En ocasiones, es posible que tenga datos confidenciales que preferiría cifrar con Full Disk Encryption (FDE). Durante el proceso de instalación, puede iniciar una instalación cifrada LVM en el disco duro o en las unidades USB.

Primero, necesitará hardware de computadora compatible. Kali Linux es compatible con plataformas amd64 (x86_64 / 64-Bit) e i386 (x86 / 32-Bit). Siempre que sea posible, el fabricante recomienda utilizar las imágenes amd64. Los requisitos de hardware son mínimos como se enumeran en la

⁴ <https://www.kali.org/downloads/>
⁵

https://my.vmware.com/en/web/vmware/downloads/info/slug/desktop_end_user_computing/vmware_workstation_player/16_0

⁶ <https://www.virtualbox.org/wiki/Downloads>



sección siguiente, aunque un mejor hardware naturalmente proporcionará un mejor rendimiento. Debería poder usar Kali Linux en hardware más nuevo con UEFI y sistemas más antiguos con BIOS.

Las imágenes i386, de forma predeterminada, utilizan un kernel PAE, por lo que puede ejecutarlas en sistemas con más de 4 GB de RAM.

En el ejemplo que se menciona más adelante, se instalará Kali Linux en una nueva máquina virtual invitada, sin ningún sistema operativo existente preinstalado.

Requisitos del sistema

Los requisitos de instalación para Kali Linux variarán según lo que le gustaría instalar y su configuración. Para conocer los requisitos del sistema:





En el extremo inferior, puede configurar Kali Linux como un servidor Secure Shell (SSH) básico sin escritorio, utilizando tan solo 128 MB de RAM (se recomiendan 512 MB) y 2 GB de espacio en disco.

En el extremo superior, si opta por instalar el escritorio Xfce4 predeterminado y el kali-linux-default metapaquete, realmente debería apuntar a al menos 2 GB de RAM y 20 GB de espacio en disco.

Cuando se utilizan aplicaciones que consumen muchos recursos, como Burp Suite, recomiendan al menos 8 GB de RAM (¡e incluso más si se trata de una aplicación web grande!) O utilizar programas simultáneos al mismo tiempo.

Requisitos previos de instalación⁷

Esta la guía se harán las siguientes suposiciones al instalar Kali Linux:

-  Usando la imagen del instalador de amd64.
-  Unidad de CD / DVD / soporte de arranque USB.
-  Disco único para instalar.
-  Conectado a una red (con DHCP y DNS habilitados) que tiene acceso a Internet saliente.

Preparación para la instalación




-  Descargue Kali Linux⁸ (el fabricante recomienda⁹ la imagen marcada como Instalador).

⁷ Dependiendo del tipo de instalación que seleccione, se pueden borrar todos los datos existentes en el disco duro, así que haga una copia de seguridad de la información importante del dispositivo en un medio externo.

⁸ <https://www.kali.org/docs/introduction/download-official-kali-linux-images/>

⁹ <https://www.kali.org/docs/introduction/what-image-to-download/#which-image-to-choose>



-  Grabe¹⁰ la ISO de Kali Linux en un DVD o una imagen de Kali Linux Live en una unidad USB. (Si no puede, consulte la instalación en red¹¹ de Kali Linux).
-  Realice una copia de seguridad de la información importante del dispositivo en un medio externo.
-  Asegúrese de que su computadora esté configurada para arrancar desde CD / DVD / USB en su BIOS / UEFI.

Un vez que tiene preparado todos los materiales y el entorno para comenzar la instalación siga los pasos indicados en la sección “Kali Linux Installation Procedure” del siguiente enlace:

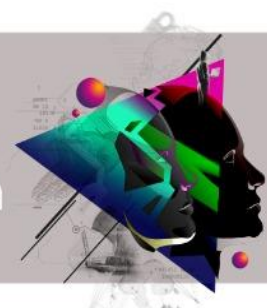
<https://www.kali.org/docs/installation/hard-disk-install/>



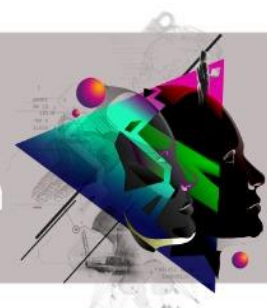
Otras distribuciones que puede considerar son las siguientes:

¹⁰ <https://www.kali.org/docs/usb/live-usb-install-with-windows/>

¹¹ <https://www.kali.org/docs/installation/network-pxe/>



Nombre	Link	Descripción
ARCHSTRIKE	https://archstrike.org/	Distribución linux con foco en ciberseguridad.
BACKBOX	https://www.backbox.org/	Distribución de Linux orientada a pruebas de penetración y evaluación de seguridad que proporciona un conjunto de herramientas de análisis de redes y sistemas.
BLACKARCH	http://blackarch.org/	Herramientas para pruebas de penetración basada en Arch Linux.
BLACKBUNTU	https://archiveos.org/blackbuntu/	Es una distribución GNU / Linux basada en Ubuntu y diseñada con Pentest, Seguridad y Desarrollo en mente para la mejor experiencia.
BUGTRAQ	https://archiveos.org/bugtraq/	Distribución GNU / Linux destinada a análisis forense digital, pruebas de penetración, laboratorios de malware y análisis forense.
CAINE	http://www.caine-live.net/	CAINE (Computer Aided INvestigative Environment) es una distribución GNU / Linux italiana creada como un proyecto de Digital Forensics.
CYBORG HAWK LINUX	https://archiveos.org/cyborg-hawk/	Distribución de Linux basada en la plataforma Ubuntu con el último kernel para profesionales de la seguridad cibernética.
DEFT LINUX	http://www.deftlinux.net/	DEFT es un sistema operativo Linux creado especialmente para profesionales y expertos de seguridad que necesiten un ecosistema para analizar datos, redes y dispositivos y poder recopilar de ellos la mayor cantidad de información posible.
DRACOS LINUX	https://dracos-linux.org/	Dracos Linux es un sistema operativo de código abierto que proporciona pruebas de penetración.
FEDORA SECURITY LAB	https://labs.fedoraproject.org/en/security/	Entorno de prueba seguro para trabajar en auditoría de seguridad, análisis forense, rescate de sistemas y enseñanza de metodologías de prueba de seguridad en universidades y otras organizaciones.
GNACK TRACK LINUX	https://archiveos.org/gnacktrack/	Distribución de Linux basada en Ubuntu que proporciona un conjunto de pruebas de penetración.
JONDO	https://anonymous-proxy-servers.net/en/jondo-live-cd.html	Entorno seguro y preconfigurado para navegación anónima.
KALI	https://www.kali.org/	Distribución de Linux de código abierto basada en Debian orientada a diversas tareas de seguridad de la información, como pruebas de penetración, investigación de seguridad, informática forense e ingeniería inversa.
LIVE HACKING DVD	http://www.livehacking.com/live-hacking-cd/download-live-hacking/	Distribución de Linux basada en Ubuntu que proporciona un conjunto de pruebas de penetración.
MATRIUX	http://matriux.sourceforge.net/	Distribución de seguridad con todas las funciones que consta de un montón de herramientas poderosas, de código abierto y gratuitas que se pueden utilizar para varios propósitos, incluidos, entre otros, pruebas de penetración, piratería ética, administración de sistemas y



		redes, investigaciones forenses cibernéticas, pruebas de seguridad, análisis de vulnerabilidades y mucho más.
MOKI	https://github.com/moki-ics/moki	Modificación de Kali para incorporar varias herramientas ICS / SCADA esparcidas por Internet, para crear un Kali Linux personalizado dirigido a profesionales de pentesting ICS / SCADA.
NETWORK SECURITY TOOLKIT (NST)	https://sourceforge.net/projects/nst/files/	Un kit de herramientas de monitoreo y análisis de seguridad de red para distribución de Linux.
NODEZERO	https://sourceforge.net/projects/nodezero/	Linux basado en Ubuntu diseñado como un sistema completo que también se puede utilizar para pruebas de penetración.
PENTOO	https://pentoo.org/	Live CD y Live USB diseñado para pruebas de penetración y evaluación de seguridad. Basado en Gentoo Linux, Pentoo se proporciona como livecd instalable de 32 y 64 bits.
PARROT SECURITY OS	https://www.parrotsec.org/	Distribución GNU / Linux basada en Debian y diseñada pensando en la seguridad y la privacidad.
SAMURAI WEB TESTING FRAMEWORK	https://www.samuraiwtf.org/	Linux completo para su uso en la formación de seguridad de aplicaciones. Es gratuito y de código abierto, distribuido como VM preconstruidas y como código fuente. La fuente consta de un Vagrantfile, activos estáticos y scripts de compilación. Durante el proceso de construcción, recupera una variedad de herramientas y objetivos de entrenamiento.
SECURITY ONION 2	https://securityonionsolutions.com/	Distribución de Linux de código abierto y gratuito para la búsqueda de amenazas, la supervisión de la seguridad empresarial y la gestión de registros. ¡El asistente de configuración fácil de usar le permite crear un ejército de sensores distribuidos para su empresa en minutos! Security Onion incluye Elasticsearch, Logstash, Kibana, Suricata, Zeek (antes conocido como Bro), Wazuh, Stenographer, TheHive, Cortex, CyberChef, NetworkMiner y muchas otras herramientas de seguridad.
TAILS	https://tails.boum.org/	Sistema operativo portátil que protege la privacidad.
QUBES OS	https://www.qubes-os.org/	Sistema operativo gratuito y de código abierto orientado a la seguridad para la informática de escritorio de un solo usuario. Qubes OS aprovecha la virtualización basada en Xen para permitir la creación y gestión de compartimentos aislados llamados qubes.
WIFISLAX	https://www.wifislax.com/	Linux para auditorías Wireless.
DEMONLINUX	https://demonlinux.com	Distribución de Debian Linux con tema de prueba de penetración.



PASO 2: Instalar el comando.

Una vez que se cuenta con este sistema operativo de manera funcional podemos instalar los comandos; algunos ya vienen preinstalados en la distribución KALI¹², pero si no fuere así puede instalarlos con los siguientes comandos, **previamente tomando privilegios de usuario "root"**:

Si el comando no estuviere pre-instalado en la distribución KALI, proceda con la siguiente instrucción:

```
# apt-get update && apt-get install dotdotpwn
```

Luego verifique que haya quedado instalada:

```
# apt install dotdotpwn
```

Leyendo lista de paquetes... Hecho

Creando árbol de dependencias... Hecho

Leyendo la información de estado... Hecho

dotdotpwn ya está en su versión más reciente (3.0.2-0kali2).

fijado dotdotpwn como instalado manualmente.

Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.

cryptsetup-run gccgo-10 gir1.2-handy-0.0 gnome-tweak-tool libamtk-5-0 libamtk-5-common libavif9

libgeos-3.9.0 libgo-10-dev libgo16 libhandy-0.0-0 libidn11 libkdecorations2private7

libkwineffects12a libkwinglutils12 libkwinxrenderutils12 libntfs-3g883 libplacebo72

libstd-rust-1.48 libstrictures-perl libtepl-5-0 libtracker-control-2.0-0 libtracker-miner-2.0-0

libyara4 python3-gevent python3-gevent-websocket python3-greenlet python3-jupyter-core

python3-m2crypto python3-nbformat python3-parameterized python3-plotly python3-zope.event

Utilice «apt autoremove» para eliminarlos.

0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.

```
# apt search ^dotdotpwn
```

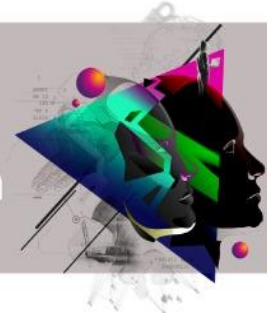
Ordenando... Hecho

Buscar en todo el texto... Hecho

dotdotpwn/kali-rolling,now 3.0.2-0kali2 amd64 **[instalado]**

Directory Traversal Fuzzer.

¹² <https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/>





[illegible]



```

root@kali:~# dotdotpwn v3.0.2 -
The Directory Traversal Fuzzer
http://dotdotpwn.sectester.net
dotdotpwn@sectester.net

by chrix & nitr0us

[*] Report name: Reports/www.csirt.gob.cl_09-10-2021_11-36.txt

[===== TARGET INFORMATION =====]
[*] Hostname: www.csirt.gob.cl
[*] Protocol: http
[*] Port: 80

[===== TRAVERSAL ENGINE =====]
[*] Creating Traversal patterns (mix of dots and slashes)
[*] Multiplying 6 times the traversal patterns (-d switch)
[*] Creating the Special Traversal patterns
[*] Translating (back)slashes in the filenames
[*] Adapting the filenames according to the OS type detected (unix)
[*] Including special suffixes
[*] Traversal Engine DONE ! - Total traversal tests created: 11028

[===== TESTING RESULTS =====]
[*] Ready to launch 3.33 traversals per second
[*] Press Enter to start the testing (You can stop it pressing Ctrl + C)

[*] HTTP Status: 500 | Testing Path: http://www.csirt.gob.cl:80/./etc/passwd
[*] HTTP Status: 404 | Testing Path: http://www.csirt.gob.cl:80/./etc/issue
[*] HTTP Status: 500 | Testing Path: http://www.csirt.gob.cl:80/././etc/passwd
[*] HTTP Status: 404 | Testing Path: http://www.csirt.gob.cl:80/././etc/issue
[*] HTTP Status: 500 | Testing Path: http://www.csirt.gob.cl:80/./././etc/passwd
[*] HTTP Status: 404 | Testing Path: http://www.csirt.gob.cl:80/./././etc/issue
[*] HTTP Status: 500 | Testing Path: http://www.csirt.gob.cl:80/././././etc/passwd
[*] HTTP Status: 404 | Testing Path: http://www.csirt.gob.cl:80/././././etc/issue
[*] HTTP Status: 500 | Testing Path: http://www.csirt.gob.cl:80/./././././etc/passwd
[*] HTTP Status: 404 | Testing Path: http://www.csirt.gob.cl:80/./././././etc/issue
[*] HTTP Status: 500 | Testing Path: http://www.csirt.gob.cl:80/././././././etc/passwd
[*] HTTP Status: 404 | Testing Path: http://www.csirt.gob.cl:80/././././././etc/issue
[*] HTTP Status: 500 | Testing Path: http://www.csirt.gob.cl:80/./././././././etc/passwd
[*] HTTP Status: 404 | Testing Path: http://www.csirt.gob.cl:80/./././././././etc/issue
[*] HTTP Status: 500 | Testing Path: http://www.csirt.gob.cl:80/././././././././etc/passwd
[*] HTTP Status: 404 | Testing Path: http://www.csirt.gob.cl:80/././././././././etc/issue
[*] HTTP Status: 500 | Testing Path: http://www.csirt.gob.cl:80/./././././././././etc/passwd
[*] HTTP Status: 404 | Testing Path: http://www.csirt.gob.cl:80/./././././././././etc/issue
[*] HTTP Status: 500 | Testing Path: http://www.csirt.gob.cl:80/././././././././././etc/passwd
[*] HTTP Status: 404 | Testing Path: http://www.csirt.gob.cl:80/././././././././././etc/issue
[*] HTTP Status: 500 | Testing Path: http://www.csirt.gob.cl:80/./././././././././././etc/passwd
[*] HTTP Status: 404 | Testing Path: http://www.csirt.gob.cl:80/./././././././././././etc/issue
[*] HTTP Status: 500 | Testing Path: http://www.csirt.gob.cl:80/././././././././././././etc/passwd
[*] HTTP Status: 404 | Testing Path: http://www.csirt.gob.cl:80/././././././././././././etc/issue
[*] HTTP Status: 500 | Testing Path: http://www.csirt.gob.cl:80/./././././././././././././etc/passwd
[*] HTTP Status: 404 | Testing Path: http://www.csirt.gob.cl:80/./././././././././././././etc/issue

```

OTROS EJEMPLOS PARA PONER EN PRÁCTICA

```
./dotdotpwn.pl -m http -h 192.168.1.1 -x 8080 -f /etc/hosts -k "localhost" -d 8 -t 200 -s
```

El Motor de Traversal creará cadenas de patrones fuzz con 8 niveles de profundidad (-d 8), entonces DotDotPwn enviará 5 peticiones por segundo (-t) contra el servidor web (-m) que escucha en el puerto 8080 (-x) e instalado en 192.168.1.1 (-h). Además, esto intentará recuperar el archivo /etc/hosts (-f) y para evitar falsos positivos, se hará una comprobación extra contra la respuesta del servidor para encontrar la palabra clave "localhost" dentro, si es así, se considera vulnerable. Además, para obtener información adicional, se recuperará el banner del servidor web (-s). DotDotPwn guardará los resultados del escaneo en un archivo llamado 192.168.1.1_<fecha>_<hora> en la carpeta "carpeta de informes".

```
./dotdotpwn.pl -m http -h 192.168.1.1 -O -X -M POST -e .php -E
```

DotDotPwn intentará detectar el Sistema Operativo que se ejecuta en el objetivo (-O) y una vez detectado el motor de navegación creará patrones de fuzz de acuerdo con el sistema operativo detectado. Después de eso, todas las peticiones HTTP se adaptarán al método especificado, en este caso, el método POST en lugar de GET (-M). Al final de cada solicitud de fuzz, se añadirá la extensión .php (-e). La bandera -E indica a DotDotPwn que busque los archivos @Extra_files definidos en TraversalEngine.pm (por defecto, "config.inc.php" y "web.config"). Por último, se ha habilitado el Algoritmo de Bisección (-X) para detectar la profundidad exacta de una vulnerabilidad, por lo que este algoritmo tratará de detectar la



profundidad exacta con el mínimo número de de peticiones. DotDotPwn guardará los resultados del escaneo en un archivo llamado 192.168.1.1_<fecha>_<hora> en la carpeta "carpeta de informes".

(Caso de estudio de una vulnerabilidad real en Webmin
<http://www.securityfocus.com/bid/18744>)

```
./dotdotpwn.pl -m http-url -u http://192.168.1.1:10000/unauthenticated/TRAVERSAL  
-O -k "root:" -r webmin.txt
```

DotDotPwn intentará detectar el Sistema Operativo que se ejecuta en 192.168.1.1 (-O) y una vez detectado el Motor de Traversal creará los patrones de fuzz de acuerdo con el sistema operativo detectado. Después de eso, el motor reemplazará el token TRAVERSAL dentro de la URL especificada (-u) por los patrones de traversal creados y enviará la petición fuzzed contra el servidor web que escucha en el puerto 10000. Por último, para evitar falsos positivos, se realizará una comprobación adicional contra la respuesta del servidor para encontrar la palabra clave "root:" dentro, si es así, se considera vulnerable. Suponiendo que el archivo a recuperar es /etc/passwd, es casi *seguro* que la palabra clave "root:" está al principio del archivo. DotDotPwn guardará los resultados del escaneo en un archivo llamado webmin.txt en la carpeta "Reports".

Módulo FTP

```
./dotdotpwn.pl -m ftp -h 192.168.1.1 -s -U nitr0us -P nltr0u5pwnzj00 -o windows  
-q -r ftp_server.txt
```

En primer lugar, DotDotPwn intentará obtener el mensaje de banner (-s) del servidor FTP (-m), y luego, tratará de iniciar sesión con el nombre de usuario especificado (-U) y la contraseña (-P) en caso de que el servidor no permita el acceso anónimo. Una vez autenticado, intentará obtener los archivos conocidos en los sistemas operativos Windows (-o) en la carpeta local "retrieved_files". Además, DotDotPwn no imprimirá los detalles de cada intento, sino que trabajará en modo silencioso (-q) y sólo imprimirá los los patrones vulnerables detectados. DotDotPwn guardará los resultados del escaneo en un archivo llamado ftp_server.txt (-r) en la carpeta "Reports".

Módulo TFTP

```
./dotdotpwn.pl -m tftp -h 192.168.1.1 -b -t 1 -f  
windows/system32/drivers/etc/hosts
```

DotDotPwn enviará un patrón de recorrido contra el servidor TFTP (-m) que sirve en 192.168.1.1 (-h) cada 1 milisegundo, es decir, lo más rápido posible. Y entonces, DotDotPwn terminará el fuzz cuando haya encontrado la primera vulnerabilidad (-b). La prueba fuzz se centrará en el archivo específico ubicado en windows/system32/drivers/etc/hosts (-f). DotDotPwn guardará los resultados del análisis en un archivo llamado 192.168.1.1_<fecha>_<hora> en la carpeta "carpeta de informes".

Módulo PAYLOAD

(Caso práctico de una vulnerabilidad real en Webmin
<http://www.securityfocus.com/bid/18744>)



```
./dotdotpwn.pl -m payload -h 192.168.1.1 -x 10000 -p payload_sample_1.txt -k  
"root:" -f /etc/passwd
```

payload_sample_1.txt (también incluido en esta carpeta):
GET /no autenticado/TRAVERSAL HTTP/1.0

El motor de traversal reemplazará el token TRAVERSAL dentro del archivo de carga útil especificado (-p) por los patrones de traversal creados y enviará el payload fuzzed contra el servidor tcp (en este ejemplo suponemos que es Webmin) escuchando en el puerto 10000 (-x) e instalado en 192.168.1.1 (-h). Finalmente, DotDotPwn buscará la palabra clave "root:" (-k) en la respuesta del servidor, y si aparece se considera vulnerable. Suponiendo que el archivo a recuperar es /etc/passwd (-f), es casi *seguro* que la palabra clave "root:" está al principio del archivo. DotDotPwn guardará los resultados del escaneo en un nombre de archivo llamado 192.168.1.1_<fecha>_<hora> en la "carpeta de informes".

Luego de que ha finalizado la ejecución del comando, podemos revisar el reporte y entender que es lo que está viendo de nuestros dominios, un ciberdelincuente, y evaluar posteriormente la seguridad de cada uno de los activos que se encuentran visibles. Es importante tener en consideración que la seguridad debe estar presente en TODOS los activos, pues los ciberdelincuentes buscarán aquellos más débiles para actuar y lograr sus objetivos: exfiltrar datos, destruir los sistemas, encriptar información para cobrar un rescate posteriormente, interceptar información confidencial, robar propiedad intelectual o propiedad industrial, entre otras acciones delictivas posibles.

Tenga presente que es importante que estas pruebas deben ser coordinadas con el equipo de operaciones y en ambientes que estén bajo supervisión.

Antes de proceder a aplicar estos comandos revise sus políticas de seguridad de la información interna, sus códigos de ética, los NDA que haya suscrito y las cláusulas de confidencialidad de su contrato de trabajo.

Defina horarios especiales o ambientes de "test o QA" equivalentes a los de "producción", para mitigar los posibles efectos perjudiciales en los dispositivos de seguridad, el sitio o el sistema web.

Use la información obtenida para visualizar sus activos desde la perspectiva de un externo e identifique vulnerabilidades a mitigar o datos/directorios a proteger.

Estudie las múltiples opciones de los comandos ilustrados en esta ficha, entienda el significado de sus diferentes parámetros con el objetivo de obtener resultados específicos, para diferentes escenarios de ataques o redirigir la salida a un archivo, para su inclusión en informes posteriores.



Tenga presente que para el procesamiento y análisis de los datos es relevante que vaya perfeccionando su manejo de LINUX y comandos PowerShell¹³ (si es un usuario de windows).

En próximas ediciones se irán reforzando estos aspectos para facilitar el manejo de los datos y resultados obtenidos, logrando así una mejor comunicación con sus equipos TIC y con el CSIRT de Gobierno.

En caso de cualquier inquietud no dude en consultarnos a soc-csirt@interior.gob.cl.

Si encuentra algún error en el documento también es importante que nos lo comunique para introducir las correcciones pertinentes en las versiones futuras de esta ficha.

¹³ <https://devblogs.microsoft.com/scripting/table-of-basic-powershell-commands/>



Anexo I: Comandos Básicos de Linux

Comandos básicos

Los comandos son esencialmente los mismos que cualquier sistema UNIX. En las tablas que se presentan a continuación se tiene la lista de comandos más frecuentes.

1. comando “pwd2

Use el comando `pwd` para averiguar la ruta del directorio de trabajo actual (carpeta) en la que se encuentra. El comando devolverá una ruta absoluta (completa), que es básicamente una ruta de todos los directorios que comienza con una barra inclinada (/). Un ejemplo de ruta absoluta es `/home / username`.

2. comando “cd”

Para navegar por los archivos y directorios de Linux, use el comando `cd`. Requiere la ruta completa o el nombre del directorio, según el directorio de trabajo actual en el que se encuentre.

Digamos que estás en `/home / username / Documents` y quieres ir a `Photos`, un subdirectorio de `Documents`. Para hacerlo, simplemente escriba el siguiente comando: `cd Photos`.

Otro escenario es si desea cambiar a un directorio completamente nuevo, por ejemplo, `/home / username / Movies`. En este caso, debe escribir `cd` seguido de la ruta absoluta del directorio: `cd /home / username / Movies`.

Hay algunos atajos que le ayudarán a navegar rápidamente:

- `cd ..` (con dos puntos) para mover un directorio hacia arriba
- `cd` para ir directamente a la carpeta de inicio
- `cd-` (con un guion) para ir a su directorio anterior

En una nota al margen, el shell de Linux distingue entre mayúsculas y minúsculas. Por lo tanto, debe escribir el directorio del nombre exactamente como está.

3. comando “ls”

El comando `ls` se usa para ver el contenido de un directorio. De forma predeterminada, este comando mostrará el contenido de su directorio de trabajo actual.



Si desea ver el contenido de otros directorios, escriba ls y luego la ruta del directorio. Por ejemplo, ingrese ls /home/username/Documents para ver el contenido de Documents.

Hay variaciones que puede usar con el comando ls:

- ls -R también listará todos los archivos en los subdirectorios
- ls -a mostrará los archivos ocultos
- ls -al enumerará los archivos y directorios con información detallada como los permisos, el tamaño, el propietario, etc.

4. comando de “cat”

cat (abreviatura de concatenar) es uno de los comandos más utilizados en Linux. Se utiliza para enumerar el contenido de un archivo en la salida estándar (stdout). Para ejecutar este comando, escriba cat seguido del nombre del archivo y su extensión. Por ejemplo: cat file.txt.

Aquí hay otras formas de usar el comando cat:

- “cat > filename” crea un nuevo archivo
- “cat filename1 filename2> filename3” une dos archivos (1 y 2) y almacena la salida de ellos en un nuevo archivo (3)
- convertir un archivo a mayúsculas o minúsculas, “cat filename | tr az AZ> salida.txt”.

5. comando “cp”

Utilice el comando cp para copiar archivos del directorio actual a un directorio diferente. Por ejemplo, el comando cp scenery.jpg / home / username / Pictures crearía una copia de paisaje.jpg (de su directorio actual) en el directorio de Pictures.

6. comando “mv”

El uso principal del comando mv es mover archivos, aunque también se puede usar para cambiar el nombre de los archivos.

Los argumentos en mv son similares al comando cp. Debe escribir mv, el nombre del archivo y el directorio de destino. Por ejemplo: mv file.txt / home / username / Documents.



Para cambiar el nombre de los archivos, el comando de Linux es “mv oldname.ext newname.ext”.

7. comando mkdir

Utilice el comando mkdir para crear un nuevo directorio; si escribe mkdir Music, se creará un directorio llamado Music.

También hay comandos adicionales de mkdir:

- Para generar un nuevo directorio dentro de otro directorio, use este comando básico de Linux mkdir Music/Newfile
- use la opción p (padres) para crear un directorio entre dos directorios existentes. Por ejemplo, mkdir -p Music/2020/Newfile creará el nuevo archivo “2020”.

8. comando “rmdir”

Si necesita eliminar un directorio, use el comando rmdir. Sin embargo, rmdir solo le permite eliminar directorios vacíos.

9. comando “rm”

El comando rm se usa para eliminar directorios y su contenido. Si solo desea eliminar el directorio, como alternativa a rmdir, use rm -r.

Nota: Tenga mucho cuidado con este comando y verifique dos veces en qué directorio se encuentra. Esto eliminará todo y no se puede deshacer.

10. comando “touch”

El comando touch le permite crear un nuevo archivo en blanco a través de la línea de comandos de Linux. Como ejemplo, ingrese touch /home/username/Documents/Web.html para crear un archivo HTML titulado Web en el directorio Documentos.

11. comando “locate”



Puede usar este comando para ubicar o localizar un archivo, al igual que el comando de búsqueda en Windows. Además, el uso del argumento `-i` junto con este comando hará que no distinga entre mayúsculas y minúsculas, por lo que puede buscar un archivo incluso si no recuerda su nombre exacto.

Para buscar un archivo que contenga dos o más palabras, use un asterisco (*). Por ejemplo, el comando `"locate -i escuela*nota"` buscará cualquier archivo que contenga la palabra "escuela" y "nota", ya sea en mayúsculas o minúsculas.

12. comando "find"

Similar al comando `"locate"`, el uso de `"find"` también busca archivos y directorios. La diferencia es que el comando `"find"` se usa para ubicar archivos dentro de un directorio determinado.

Como ejemplo, el comando `find /home/ -name notes.txt` buscará un archivo llamado `notes.txt` dentro del directorio de inicio y sus subdirectorios.

Otras variaciones al usar el hallazgo son:

- Para buscar archivos en el directorio actual, `"find. -nombre notes.txt"`
- Para buscar directorios desde la raíz, llamados `home`, use `"find / -type d -name home"`

13. comando "grep"

Otro comando básico de Linux que sin duda es útil para el uso diario es `grep`. Te permite buscar en todo el texto de un archivo determinado.

Para ilustrar, `grep blue notepad.txt` buscará la palabra `azul` en el archivo del bloc de notas. Las líneas que contienen la palabra buscada se mostrarán completamente.

14. comando "sudo"

Abreviatura de " SuperUser Do ", este comando le permite realizar tareas que requieren permisos administrativos o de root. Sin embargo, no es recomendable utilizar este comando para el uso diario porque podría ser fácil que ocurra un error si hiciste algo mal.



15. comando “df”

Utilice el comando df para obtener un informe sobre el uso de espacio en disco del sistema, que se muestra en porcentaje y KB. Si desea ver el informe en megabytes, escriba df -m.

16. comando “du”

Si desea comprobar cuánto espacio ocupa un archivo o un directorio, el comando du (Uso del disco) es la respuesta. Sin embargo, el resumen de uso del disco mostrará los números de bloque de disco en lugar del formato de tamaño habitual. Si desea verlo en bytes, kilobytes y megabytes, agregue el argumento -h a la línea de comando.

17. comando “head”

El comando head se usa para ver las primeras líneas de cualquier archivo de texto. De forma predeterminada, mostrará las primeras diez líneas, pero puede cambiar este número a su gusto. Por ejemplo, si solo desea mostrar las primeras cinco líneas, escriba head -n 5 filename.ext.

18. comando “tail”

Este tiene una función similar al comando head, pero en lugar de mostrar las primeras líneas, el comando tail mostrará las últimas diez líneas de un archivo de texto. Por ejemplo, tail -n filename.ext.

19. comando “diff”

Abreviatura de diferencia, el comando diff compara el contenido de dos archivos línea por línea. Después de analizar los archivos, generará las líneas que no coinciden. Los programadores suelen utilizar este comando cuando necesitan realizar modificaciones en el programa en lugar de reescribir todo el código fuente.

La forma más simple de este comando es diff file1.ext file2.ext

20. comando “tar”



El comando tar es el comando más utilizado para archivar varios archivos en un tarball, un formato de archivo común de Linux que es similar al formato zip, con la compresión opcional.

Este comando es bastante complejo con una larga lista de funciones, como agregar nuevos archivos a un archivo existente, enumerar el contenido de un archivo, extraer el contenido de un archivo y muchas más. Consulte algunos ejemplos prácticos para saber más sobre otras funciones.

21. comando “chmod”

chmod es otro comando de Linux, que se utiliza para cambiar los permisos de lectura, escritura y ejecución de archivos y directorios. Como este comando es bastante complicado, puede leer el tutorial completo para ejecutarlo correctamente.

22. comando “chown”

En Linux, todos los archivos pertenecen a un usuario específico. El comando chown le permite cambiar o transferir la propiedad de un archivo al nombre de usuario especificado. Por ejemplo, chown linuxuser2 file.ext hará que linuxuser2 sea el propietario del file.ext .

23. comando “jobs”

El comando jobs mostrará todos los trabajos actuales junto con sus estados. Un trabajo es básicamente un proceso que inicia el shell.

24. comando “kill”

Si tiene un programa que no responde, puede terminarlo manualmente usando el comando kill. Enviará una cierta señal a la aplicación que no funciona correctamente y le indicará a la aplicación que se cierre.

Hay un total de sesenta y cuatro señales que puede usar, pero las personas generalmente solo usan dos señales:



- SIGTERM (15): solicita que un programa deje de ejecutarse y le da algo de tiempo para guardar todo su progreso. Si no especifica la señal al ingresar el comando kill, se usará esta señal.
- SIGKILL (9): obliga a los programas a detenerse inmediatamente. El progreso no guardado se perderá.

Además de conocer las señales, también necesita conocer el número de identificación del proceso (PID) del programa que desea matar. Si no conoce el PID, simplemente ejecute el comando “ps ux”.

Después de saber qué señal desea usar y el PID del programa, ingrese la siguiente sintaxis:

kill [opción de señal] PID .

25. comando “ping”

Utilice el comando ping para verificar el estado de su conectividad a un servidor. Por ejemplo, simplemente ingresando ping google.com, el comando verificará si puede conectarse a Google y también medirá el tiempo de respuesta.

26. comando “wget”

La línea de comandos de Linux es muy útil; incluso puede descargar archivos de Internet con la ayuda del comando wget. Para hacerlo, simplemente escriba wget seguido del enlace de descarga.

27. comando “uname”

El comando uname , abreviatura de Unix Name, imprimirá información detallada sobre su sistema Linux, como el nombre de la máquina, el sistema operativo, el kernel, etc.

28. comando “top”

Como terminal equivalente al Administrador de tareas en Windows, el comando “top” mostrará una lista de procesos en ejecución y cuánta CPU usa cada proceso. Es muy útil monitorear el uso de recursos del sistema, especialmente sabiendo qué proceso debe terminarse porque consume demasiados recursos. Busque referencias sobre “htop”.



29. comando “history”

Cuando haya estado usando Linux durante un cierto período de tiempo, notará rápidamente que puede ejecutar cientos de comandos todos los días. Como tal, ejecutar el comando “history” es particularmente útil si desea revisar los comandos que ha ingresado antes.

30. comando “man”

¿Confundido acerca de la función de ciertos comandos de Linux? No se preocupe, puede aprender fácilmente cómo usarlos directamente desde el shell de Linux usando el comando man. Por ejemplo, ingresar man tail mostrará la instrucción manual del comando tail.

31. comando “echo”

Este comando se usa para mover algunos datos a un archivo. Por ejemplo, si desea agregar el texto "Hola, mi nombre es Juan" en un archivo llamado nombre.txt, debe escribir “echo Hola, mi nombre es Juan >> nombre.txt”.

32. comando “zip,unzip”

Use el comando zip para comprimir sus archivos en un archivo zip y use el comando unzip para extraer los archivos comprimidos de un archivo zip.

33. comando “hostname”

Si desea saber el nombre de su host / red, simplemente escriba hostname. Si agrega un -i al final, se mostrará la dirección IP de su red.

34. comando “useradd, userdel”

Dado que Linux es un sistema multiusuario, esto significa que más de una persona puede interactuar con el mismo sistema al mismo tiempo. useradd se usa para crear un nuevo usuario, mientras que



passwd agrega una contraseña a la cuenta de ese usuario. Para agregar una nueva persona llamada John escriba, useradd John y luego para agregar su tipo de contraseña, passwd 123456789.

Eliminar un usuario es muy similar a agregar un nuevo usuario. Para eliminar el tipo de cuenta de usuario, userdel UserName

Notas:

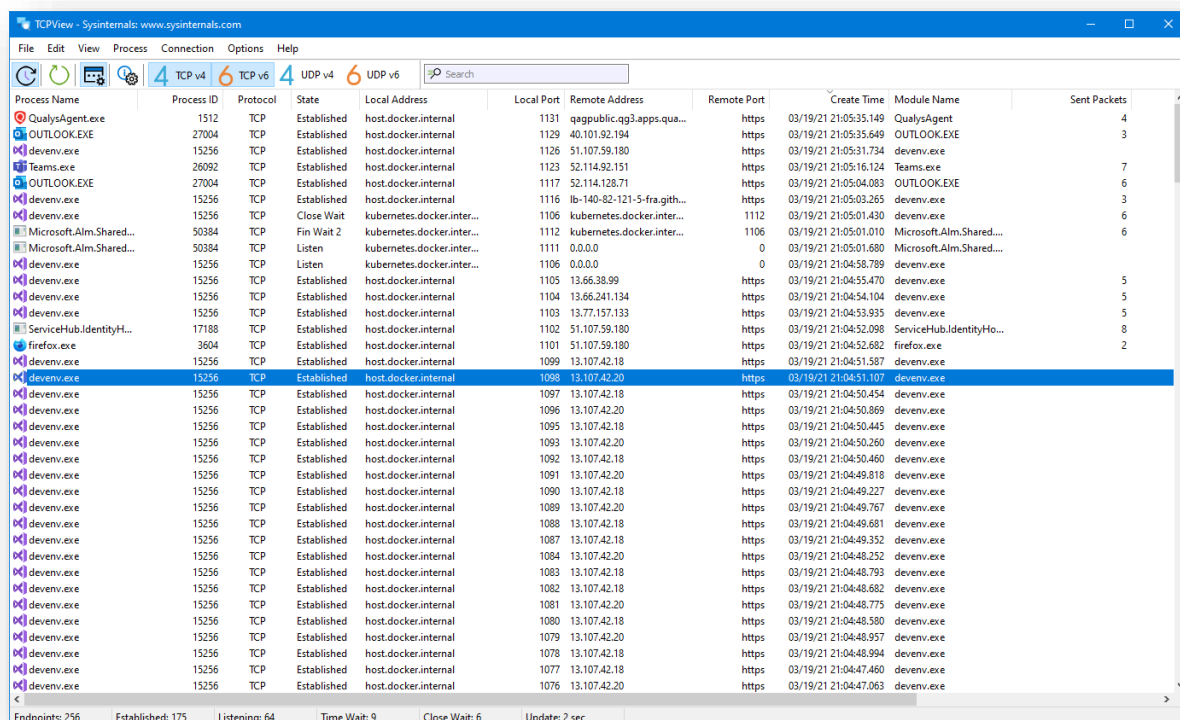
- Utilice el comando “clear” para limpiar la terminal si se llena de demasiados comandos anteriores.
- Pruebe el botón TAB para completar automáticamente lo que está escribiendo. Por ejemplo, si necesita escribir Documentos, comience a escribir un comando (vayamos con cd Docu, luego presione la tecla TAB) y el terminal completará el resto, mostrándole Documentos de cd.
- Ctrl + C y Ctrl + Z se utilizan para detener cualquier comando que esté funcionando actualmente. Ctrl + C detendrá y terminará el comando, mientras que Ctrl + Z simplemente pausará el comando.
- Si accidentalmente congela su terminal utilizando Ctrl + S, basta con descongelar usando Ctrl + Q.
- Ctrl + A lo mueve al principio de la línea, mientras que Ctrl + E lo mueve al final.
- Puede ejecutar varios comandos en un solo comando utilizando el “;” para separarlos. Por ejemplo Command1; Command2; Command3. O use && si solo desea que el siguiente comando se ejecute cuando el primero sea exitoso.



Anexo II: Comandos o aplicativos básicos para Windows: TCPView

En esta segunda versión de comandos o aplicativos para Windows mencionaremos el aplicativo “TCPView de la suite SYSINTERNALS”.

TCPView es un programa de Windows que le mostrará listados detallados de todos los puntos finales TCP y UDP en su sistema, incluidas las direcciones locales y remotas y el estado de las conexiones TCP. En Windows Server 2008, Vista y XP, TCPView también informa el nombre del proceso propietario del endpoint. TCPView proporciona un subconjunto más informativo y convenientemente presentado del programa Netstat que se envía con Windows. La descarga de TCPView incluye Tcpcvcon, una versión de línea de comandos con la misma funcionalidad.



Process Name	Process ID	Protocol	State	Local Address	Local Port	Remote Address	Remote Port	Create Time	Module Name	Sent Packets
QualysAgent.exe	1512	TCP	Established	host.docker.internal	1131	qagpublic.cg3.apps.qua...	https	03/19/21 21:05:35.149	QualysAgent	4
OUTLOOK.EXE	27004	TCP	Established	host.docker.internal	1129	40.101.92.194	https	03/19/21 21:05:35.649	OUTLOOK.EXE	3
devenv.exe	15256	TCP	Established	host.docker.internal	1126	51.107.59.180	https	03/19/21 21:05:31.734	devenv.exe	
Teams.exe	26092	TCP	Established	host.docker.internal	1123	52.114.92.151	https	03/19/21 21:05:16.124	Teams.exe	7
OUTLOOK.EXE	27004	TCP	Established	host.docker.internal	1117	52.114.128.71	https	03/19/21 21:05:04.083	OUTLOOK.EXE	6
devenv.exe	15256	TCP	Established	host.docker.internal	1116	lb-140-82-121-5-fra.git...	https	03/19/21 21:05:03.265	devenv.exe	3
devenv.exe	15256	TCP	Close Wait	kubernetes.docker.inter...	1106	kubernetes.docker.inter...	1112	03/19/21 21:05:01.430	devenv.exe	6
Microsoft.Alm.Shared...	50384	TCP	Fin Wait 2	kubernetes.docker.inter...	1112	kubernetes.docker.inter...	1106	03/19/21 21:05:01.010	Microsoft.Alm.Shared...	6
Microsoft.Alm.Shared...	50384	TCP	Listen	kubernetes.docker.inter...	1111	0.0.0.0	0	03/19/21 21:05:01.680	Microsoft.Alm.Shared...	
devenv.exe	15256	TCP	Listen	kubernetes.docker.inter...	1106	0.0.0.0	0	03/19/21 21:04:58.789	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1105	13.66.38.99	https	03/19/21 21:04:55.470	devenv.exe	5
devenv.exe	15256	TCP	Established	host.docker.internal	1104	13.66.241.134	https	03/19/21 21:04:54.104	devenv.exe	5
devenv.exe	15256	TCP	Established	host.docker.internal	1103	13.77.157.133	https	03/19/21 21:04:53.935	devenv.exe	5
ServiceHub.IdentityH...	17188	TCP	Established	host.docker.internal	1102	51.107.59.180	https	03/19/21 21:04:52.098	ServiceHub.IdentityHo...	8
firefox.exe	3604	TCP	Established	host.docker.internal	1101	51.107.59.180	https	03/19/21 21:04:52.682	firefox.exe	2
devenv.exe	15256	TCP	Established	host.docker.internal	1099	13.107.42.18	https	03/19/21 21:04:51.587	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1098	13.107.42.20	https	03/19/21 21:04:51.107	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1097	13.107.42.18	https	03/19/21 21:04:50.454	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1096	13.107.42.20	https	03/19/21 21:04:50.869	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1095	13.107.42.18	https	03/19/21 21:04:50.445	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1093	13.107.42.20	https	03/19/21 21:04:50.260	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1092	13.107.42.18	https	03/19/21 21:04:50.460	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1091	13.107.42.20	https	03/19/21 21:04:49.818	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1090	13.107.42.18	https	03/19/21 21:04:49.227	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1089	13.107.42.20	https	03/19/21 21:04:49.767	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1088	13.107.42.18	https	03/19/21 21:04:49.681	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1087	13.107.42.18	https	03/19/21 21:04:49.352	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1084	13.107.42.20	https	03/19/21 21:04:48.252	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1083	13.107.42.18	https	03/19/21 21:04:48.793	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1082	13.107.42.18	https	03/19/21 21:04:48.682	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1081	13.107.42.20	https	03/19/21 21:04:48.775	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1080	13.107.42.18	https	03/19/21 21:04:48.580	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1079	13.107.42.20	https	03/19/21 21:04:48.957	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1078	13.107.42.18	https	03/19/21 21:04:48.994	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1077	13.107.42.18	https	03/19/21 21:04:47.460	devenv.exe	
devenv.exe	15256	TCP	Established	host.docker.internal	1076	13.107.42.20	https	03/19/21 21:04:47.063	devenv.exe	

Endpoints: 256 Established: 175 Listening: 64 Time Wait: 9 Close Wait: 6 Update: 2 sec

Este programa puede descargarlo desde:

<https://download.sysinternals.com/files/TCPView.zip>

Cuando inicie TCPView, enumerará todos los puntos finales TCP y UDP activos, resolviendo todas las direcciones IP en sus versiones de nombre de dominio. Puede utilizar un botón de la barra de herramientas o un elemento de menú para alternar la visualización de los nombres resueltos.



TCPView muestra el nombre del proceso que posee cada punto final, incluido el nombre del servicio (si corresponde).

De forma predeterminada, TCPView se actualiza cada segundo, pero puede utilizar el elemento de menú Opciones | Frecuencia de actualización para cambiar la frecuencia. Los puntos finales que cambian de estado de una actualización a la siguiente se resaltan en amarillo; los que se eliminan se muestran en rojo y los nuevos puntos finales se muestran en verde.

Puede cerrar las conexiones TCP / IP establecidas (aquellas etiquetadas con un estado de ESTABLECIDO) seleccionando Archivo | Cerrar conexiones, o haciendo clic con el botón derecho en una conexión y eligiendo Cerrar conexiones en el menú contextual resultante.

Puede guardar la ventana de salida de TCPView en un archivo usando el elemento del menú Guardar.

Nota adicional para “tcpvcon”:

El uso de Tcpcvcon es similar al de la utilidad netstat incorporada de Windows:

Uso:

cmd

 Copiar

```
tcpvcon [-a] [-c] [-n] [process name or PID]
```

Parámetro	Descripción
-a	Mostrar todos los puntos finales (el valor predeterminado es mostrar las conexiones TCP establecidas).
-C	Imprime la salida como CSV.
-norte	No resuelva las direcciones.

Con estos tips básicos buscamos incentivarlo a explorar estas herramientas y sus múltiples usos para ciberseguridad.



“HOLA, MUNDO” EN OTROS LENGUAJES

RUST:

```
fn main() {  
  println!("Hello World!");  
}
```

CLOJURE

```
(ns clojure.examples.hello  
  (:gen-class))  
(defn hello-world []  
  (println "Hello, World!"))  
(hello-world)
```

TYPESCRIPT

```
let message: string = 'Hello, World!';  
console.log(message);
```

ELIXIR

```
IO.puts("Hello, World!")
```

JULIA

```
print("Hello World!")
```

PYTHON:

```
print('Hello, world!')
```

DART

```
void main() {  
  print('Hello, World!');  
}
```



SWIFT

```
import UIKit
```

```
var str = "Hello, World!"
```

NODE JS

```
// server.js  
'use strict';  
const http = require('http');  
const server = http.createServer(function (req, res) {  
    res.writeHead(200, {'content-type': 'text/plain'});  
    res.end('Hello, World!');  
});  
server.listen(8000);
```

GO

```
package main  
import "fmt"  
func main() {  
    fmt.Println("Hello, World!")  
}
```

F#

```
#light
```

```
let main =  
    printfn "Hello, World!"
```

```
do main
```

C#

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```



```
namespace ConsoleApp1  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            Console.WriteLine("Hello, World!");  
  
            Console.ReadLine();  
        }  
    }  
}
```

Kotlin

```
fun main(args: Array<String>) {  
    println("Hello World")  
}
```

JavaScript

```
<!DOCTYPE HTML>  
<html>  
<body>  
    <p>Before the script...</p>  
    <script>  
        alert( 'Hello, world!' );  
    </script>  
    <p>...After the script.</p>  
</body>  
</html>
```

Crystal

```
puts "Hello World"
```
