



11 de junio de 2021
Ficha N° 7 SLOWHTTPTEST
CSIRT DE GOBIERNO

Comando de la semana “SLOWHTTPTEST”

I. CONTEXTO

Este documento, denominado “comando de la semana”, tiene como objetivo ilustrar sobre herramientas que pueden 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 activos de información 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”.

II. INTRODUCCIÓN

Una de las tareas regulares que en ciberseguridad se realizan es la verificación de los sitios o sistemas que están expuestos a Internet. Una de las vulnerabilidades que buscan algunas entidades maliciosas es la posibilidad de hacer que el servicio del sitio o sistema web colapse por agotamiento de recursos desde el lado servidor o bien un funcionamiento degradado ante requerimientos malformados enviados por el cliente. Este tipo de ataque se incluyen en la familia de ataques de denegación de servicios (DDoS / DoS).

Para este caso existe un comando Linux que nos ayuda a recopilar información manera simple, con una herramienta de código abierto y, en base a sus resultados tomar decisiones de mitigación¹, monitoreo y vigilancia, además de contrastarlo con los reportes internos de impacto en uso de recurso (memoria, cpu, procesos activos del webserver, entre otros): SLOWHTTPTEST.

¹ <https://blog.shekyan.com/2011/11/how-to-protect-against-slow-http-attacks.html>



¿Qué es SLOWHTTPTEST?

SlowHTTPTest es una herramienta altamente configurable que simula algunos ataques de denegación de servicio de la capa de aplicación. Funciona en la mayoría de las plataformas Linux, OSX y Cygwin², un entorno similar a Unix y una interfaz de línea de comandos para Microsoft Windows.

Implementa la mayoría de los ataques DoS³ de la capa de aplicación de bajo ancho de banda, como slowloris, Slow HTTP POST, Slow Read attack (basado en el exploit del temporizador de persistencia TCP) al drenar el pool de conexiones concurrentes, así como el ataque Apache Range Header al causar un uso muy significativo de la memoria y la CPU en el servidor.

Los ataques DoS Slowloris y Slow HTTP POST se basan en el hecho de que el protocolo HTTP, por diseño, requiere que las peticiones sean recibidas completamente por el servidor antes de ser procesadas. Si una petición HTTP no está completa, o si la tasa de transferencia es muy baja, el servidor mantiene sus recursos ocupados esperando el resto de los datos. Si el servidor mantiene demasiados recursos ocupados, se produce una denegación de servicio. Esta herramienta está enviando peticiones HTTP parciales, intentando conseguir la denegación de servicio del servidor HTTP objetivo.

NOTA: Dado que es importante un buen manejo de los comandos básicos de Linux, tanto para posteriores manipulaciones 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

² <https://www.cygwin.com/>

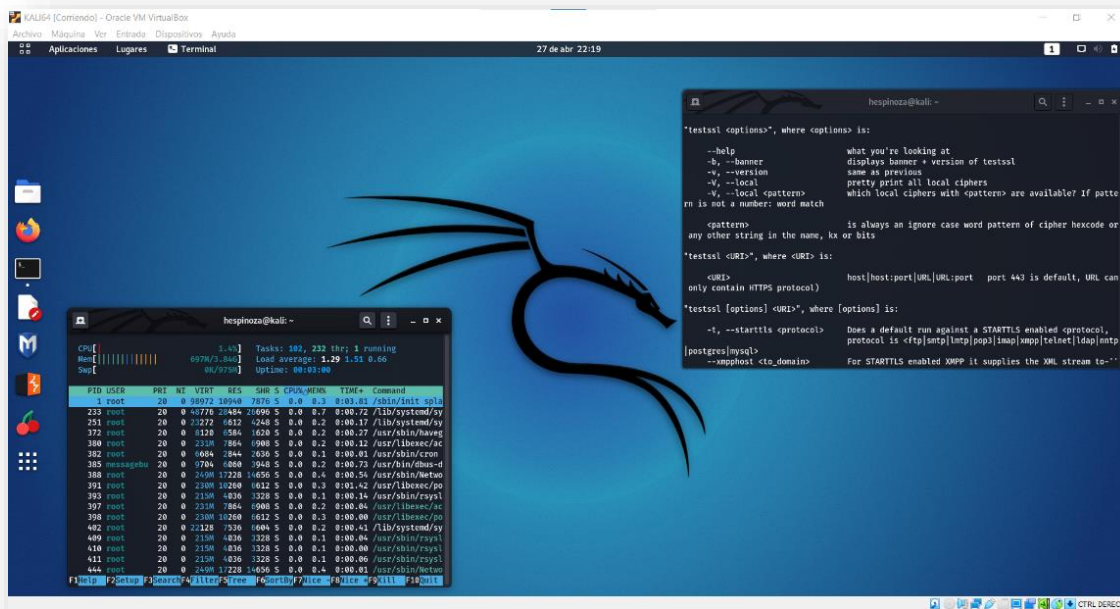
³ <https://blog.shekyan.com/2012/01/are-you-ready-for-slow-reading.html>



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.



PASO 2: Instalar el comando.

Una vez que se cuenta con este sistema operativo de manera funcional podemos instalar el comando “SLOWHTTPTEST”; en general este ya viene preinstalado en la distribución KALI⁷, pero si no fuere así puede instalarlo con los siguientes comandos, **previamente tomando privilegios de usuario “root”**:

```
apt-get install slowhttptest
```

Alternativamente puede descargarse el código fuente y compilar una versión específica para su kernel:

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>

⁷ <https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/>



```
$ tar -xzf slowhttpptest-x.x.tar.gz
```

```
$ cd slowhttpptest-x.x
```

```
$ ./configure --prefix=PREFIX
```

```
$ make
```

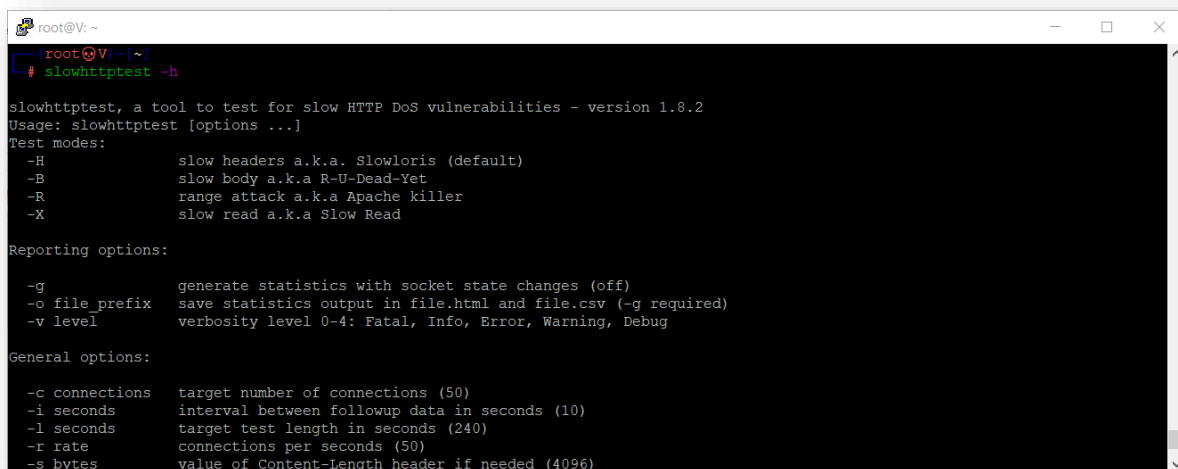
```
$ sudo make install
```



PASO3: Verificar su instalación.

Una vez que se instalado podemos verificar y explorar las múltiples opciones que ofrece para su ejecución:

En una consola de su KALI ejecute el comando para que muestre la ayuda: “slowhttptest -h”.



```
root@V: ~  
root@V: ~  
# slowhttptest -h  
  
slowhttptest, a tool to test for slow HTTP DoS vulnerabilities - version 1.8.2  
Usage: slowhttptest [options ...]  
Test modes:  
-H          slow headers a.k.a. Slowloris (default)  
-B          slow body a.k.a R-U-Dead-Yet  
-R          range attack a.k.a Apache killer  
-X          slow read a.k.a Slow Read  
  
Reporting options:  
  
-g          generate statistics with socket state changes (off)  
-o file_prefix  save statistics output in file.html and file.csv (-g required)  
-v level      verbosity level 0-4: Fatal, Info, Error, Warning, Debug  
  
General options:  
  
-c connections  target number of connections (50)  
-i seconds      interval between followup data in seconds (10)  
-l seconds      target test length in seconds (240)  
-r rate         connections per seconds (50)  
-s bytes        value of Content-Length header if needed (4096)
```

Debiéramos lograr desplegar todas las opciones y parámetros de ejecución, junto a su explicación en la consola.

```
# slowhttptest -h  
  
slowhttptest, a tool to test for slow HTTP DoS vulnerabilities - version  
1.8.2  
Usage: slowhttptest [options ...]  
Test modes:  
-H          slow headers a.k.a. Slowloris (default)  
-B          slow body a.k.a R-U-Dead-Yet  
-R          range attack a.k.a Apache killer  
-X          slow read a.k.a Slow Read  
  
Reporting options:  
  
-g          generate statistics with socket state changes (off)  
-o file_prefix  save statistics output in file.html and file.csv (-g  
required)  
-v level      verbosity level 0-4: Fatal, Info, Error, Warning,  
Debug  
  
General options:
```



```
-c connections      target number of connections (50)
-i seconds          interval between followup data in seconds (10)
-l seconds          target test length in seconds (240)
-r rate             connections per seconds (50)
-s bytes            value of Content-Length header if needed (4096)
-t verb             verb to use in request, default to GET for
                    slow headers and response and to POST for slow body
-u URL              absolute URL of target (http://localhost/)
-x bytes            max length of each randomized name/value pair of
                    followup data per tick, e.g. -x 2 generates
                    X-xx: xx for header or &xx=xx for body, where x
                    is random character (32)
-f content-type     value of Content-type header (application/x-www-form-
urlencoded)
-m accept           value of Accept header
                    (text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5)

Probe/Proxy options:

-d host:port        all traffic directed through HTTP proxy at host:port
(off)
-e host:port        probe traffic directed through HTTP proxy at host:port
(off)
-p seconds          timeout to wait for HTTP response on probe connection,
                    after which server is considered inaccessible (5)
-j cookies           value of Cookie header (ex.: -j "user_id=1001;
timeout=9000")

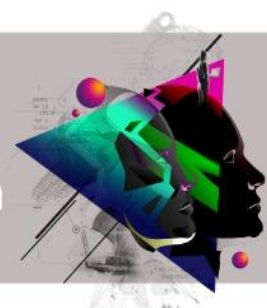
Range attack specific options:

-a start            left boundary of range in range header (5)
-b bytes            limit for range header right boundary values (2000)

Slow read specific options:

-k num              number of times to repeat same request in the connection.
Use to
                    multiply response size if server supports persistent
connections (1)
-n seconds          interval between read operations from recv buffer in
seconds (1)
-w bytes            start of the range advertised window size would be
picked from (1)
-y bytes            end of the range advertised window size would be picked
from (512)
-z bytes            bytes to slow read from receive buffer with single
read() call (5)
```

Donde dichas opciones se describen según la ayuda del mismo comando:



OPTION	DESCRIPCIÓN
-a start	valor inicial del especificador de rangos para la prueba del encabezado de rango
-b bytes	límite del especificador de rangos para la prueba de cabecera de rangos
-c number of connections	limitado a 65539
-d proxy host:port	para dirigir todo el tráfico a través del proxy web
-e proxy host:port	para dirigir sólo el tráfico de la sonda a través del proxy web
-H, B, R or X	especificar la ralentización en la sección de cabeceras o en el cuerpo del mensaje, -R activa la prueba de rango, -X activa la prueba de lectura lenta
-f content-type	valor de la cabecera Content-type
-g	generar estadísticas en formatos CSV y HTML, el patrón es slow_xxx.csv/html, donde xxx es la hora y la fecha
-i seconds	intervalo entre los datos de seguimiento en segundos, por conexión
-j cookies	valor de la cabecera Cookie (ej.: -j "user_id=1001; timeout=9000")
-k pipeline factor	número de veces que se repite la solicitud en la misma conexión para la prueba de lectura lenta, si el servidor admite la canalización HTTP.
-l seconds	duración de la prueba en segundos
-m accept	valor de la cabecera Accept
-n seconds	intervalo entre las operaciones de lectura del búfer de recepción
-o file	ruta y/o nombre del archivo de salida personalizado, efectivo si se especifica -g
-p seconds	tiempo de espera para la respuesta HTTP en la conexión de sondeo, después del cual el servidor se considera inaccesible
-r connections per second	tasa de conexión
-s bytes	valor de la cabecera Content-Length, si se especifica -B
-t verb	verbo personalizado a utilizar
-u URL	<u>URL de destino, el mismo formato que se escribe en el navegador, por ejemplo https://host[:puerto]/</u>
-v level	nivel de verbosidad del registro 0-4
-w bytes	inicio del rango en el que se recogerá el tamaño de la ventana anunciada
-x bytes	longitud máxima de los datos de seguimiento
-y bytes	fin del intervalo del que se extrae el tamaño de la ventana anunciada
-z bytes	bytes a leer del buffer de recepción con una sola operación read()



Paso 4: Ponerlo en marcha para verificar nuestra infraestructura.

Un ejemplo de ejecución básica para nuestros primeros pasos:

Probaremos el sitio web <http://192.168.1.202/index.php> con 1000 conexiones slowloris.

Para lograr esta prueba utilice 1000 conexiones (-c 1000) con el modo Slowloris (-H) y genere estadísticas (-g) con el nombre del archivo de salida (-o slowhttp) . Utilice 10 segundos para esperar datos (-i 10), 200 conexiones (-r 200) con solicitudes GET (-t GET) contra la URL de destino (-u http://192.168.1.202/index.php) con una longitud máxima de 24 bytes (-x 24) y un tiempo de 3 segundos de timeout (-p 3)

EJEMPLO

```
root@kali:~# slowhttptest -c 1000 -H -g -o slowhttp -i 10 -r 200 -t
GET -u http://192.168.1.202/index.php -x 24 -p 3
Sat May 17 10:45:26 2014:
Sat May 17 10:45:26 2014:
    slowhttptest version 1.6
    - https://code.google.com/p/slowhttptest/ -
test type:                        SLOW HEADERS
number of connections:            1000
URL:                             http://192.168.1.202/index.php
verb:                             GET
Content-Length header value:      4096
follow up data max size:          52
interval between follow up data:  10 seconds
connections per seconds:          200
probe connection timeout:         3 seconds
test duration:                    240 seconds
using proxy:                      no proxy

Sat May 17 10:45:26 2014:
slow HTTP test status on 0th second:

initializing:                      0
pending:                           1
connected:                         0
error:                             0
closed:                            0
service available:                 YES
```

Que se ve observa en una consola KALI después de una ejecución:

Vista de un ejemplo: Ejecución del comando:



```
slowhttptest -c 1000 -H -g -o slowhttp -i 10 -r 200 -t GET -u https://www.csirt.gob.cl -x 24 -p 3
```

```
root@V: ~  
- https://github.com/shekya/slowhttptest -  
test type: SLOW HEADERS  
number of connections: 1000  
URL: https://www.csirt.gob.cl/  
Verb: GET  
cookie:  
Content-Length header value: 4096  
follow up data max size: 52  
interval between follow up data: 10 seconds  
connections per seconds: 200  
probe connection timeout: 3 seconds  
test duration: 240 seconds  
using proxy: no proxy  
  
Thu Jun 10 15:46:15 2021:  
slow HTTP test status on 5th second:  
  
initializing: 0  
pending: 45  
connected: 61  
error: 0  
closed: 1  
service available: NO  
[
```

Una vez que ha finalizado el test indica lo siguiente:

```
root@V: ~  
follow up data max size: 52  
interval between follow up data: 10 seconds  
connections per seconds: 200  
probe connection timeout: 3 seconds  
test duration: 240 seconds  
using proxy: no proxy  
  
Thu Jun 10 15:50:10 2021:  
slow HTTP test status on 240th second:  
  
initializing: 0  
pending: 0  
connected: 998  
error: 0  
closed: 2  
service available: YES  
Thu Jun 10 15:50:11 2021:  
Test ended on 241th second  
Exit status: Hit test time limit  
CSV report saved to slowhttp.csv  
HTML report saved to slowhttp.html  
  
(root@V) ~ [~]
```

Como producto de la ejecución se generan dos archivos:

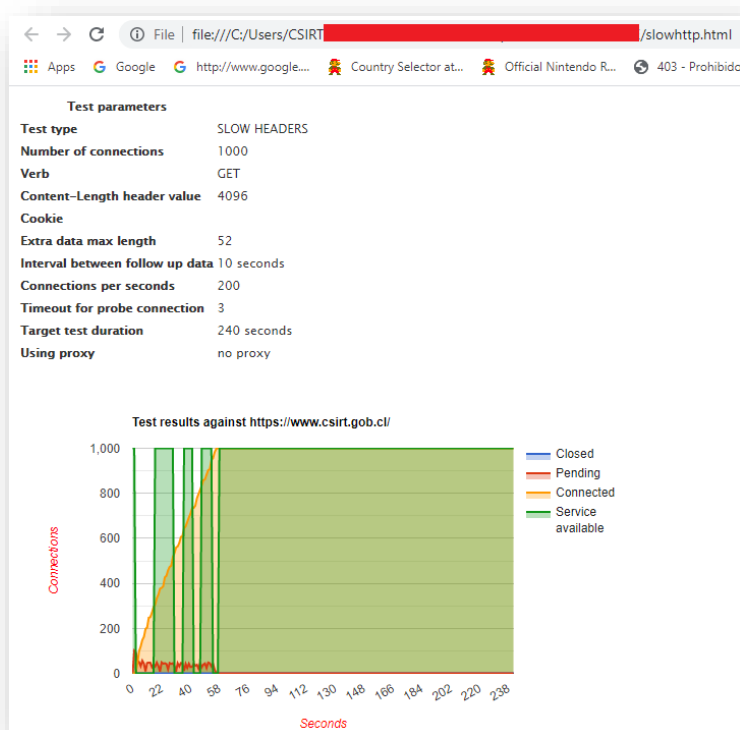
- CSV report saved to slowhttp.csv
- HTML report saved to slowhttp.html



El archivo slowhttp.csv, contiene el resultado de los datos de la prueba en un formato CSV.

```
root@V: ~  
Seconds,Closed,Pending,Connected,Service Available  
0,0,1,0,1000  
3,1,103,2,0  
5,1,45,61,0  
6,1,60,102,0  
8,1,59,105,0  
9,1,23,160,1000  
11,1,49,168,1000  
12,1,28,214,1000  
14,1,57,218,1000  
15,1,19,257,0  
17,1,61,274,0  
18,1,34,302,0  
19,1,61,332,0  
21,1,60,334,0  
22,1,56,391,0  
24,1,62,391,0  
25,1,19,448,0  
27,2,60,451,0  
28,2,31,481,0  
30,2,58,511,0  
31,2,8,562,0  
32,2,60,566,0  
--More--
```

El archivo slowhttp.html se puede abrir con un web browser, pues está escrito en HTML, y aporta una vista gráfica de los resultados obtenidos:





Otros ejemplos de ejecución:

Ejemplo de uso en modo “slow message”:

```
#slowhttpptest -c 1000 -B -g -o my_body_stats -i 110 -r 200 -s 8192 -t FAKEVERB -u  
https://myseceureserver/resources/loginform.html -x 10 -p 3
```

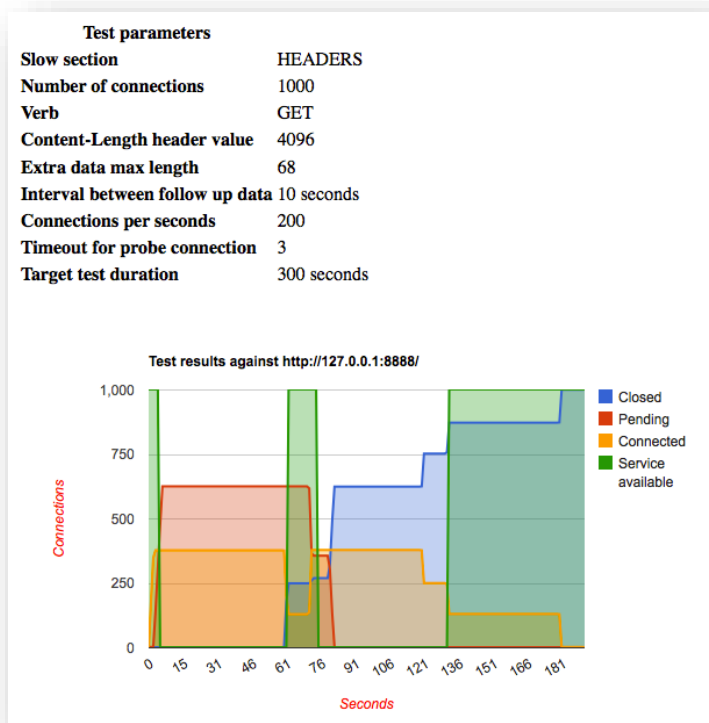
Ejemplo de uso en modo “slowloris”:

```
slowhttpptest -c 1000 -H -g -o my_header_stats -i 10 -r 200 -t GET -u  
https://myseceureserver/resources/index.html -x 24 -p 3
```

Ejemplo de uso en modo “slow read” pasando a través de un proxy x.x.x.x:8080 para obtener la disponibilidad desde una IP distinta a la del cliente generador:

```
slowhttpptest -c 1000 -X -r 1000 -w 10 -y 20 -n 5 -z 32 -u http://someserver/somebigresource -p 5  
-l 350 -e x.x.x.x:8080
```

Otra muestra de la vista HTML del reporte de salida:





Algunos mensajes de error como resultado de la ejecución pueden ser:

ERROR MESSAGE	QUÉ SIGNIFICA
"Hit test time limit"	el programa ha alcanzado el límite de tiempo especificado con el argumento -l
"No open connections left"	el par cerró todas las conexiones
"Cannot establish connection"	no se estableció ninguna conexión durante los primeros N segundos de la prueba, donde N es el valor del argumento -i, o 10, si no se especifica. Esto ocurriría si no hay ruta al host o el peer remoto está caído.
"Connection refused"	el par remoto no acepta conexiones (¿sólo las tuyas? Utiliza un proxy para sondear) en el puerto especificado.
"Cancelled by user"	usted presionó Ctrl-C o envió SIGINT de alguna otra manera.
"Unexpected error"	no debería ocurrir nunca.

Tenga presente que es importante que estas pruebas sean coordinadas con el equipo de operaciones y en ambientes que estén bajo supervisión, pues eventualmente si el sitio o sistema web está mal construido o implementado, puede producirse una denegación de servicio y afectar a sus usuarios.

Defina horarios especiales o ambientes de "test o QA" equivalentes a los de "producción", para mitigar los posibles efectos de una caída del servicio.

Existen otras herramientas que se enfocan en la denegación de servicio por inundación de carga o creación de distractores con muchos falsos positivos para que los IPS/IDS distraigan la atención de los Encargados de Ciberseguridad, y en ese contexto, el atacante tenga espacio de maniobra por otros vectores de ataque.

Estudie las múltiples opciones que tiene el comando para obtener resultados específicos o redirigir la salida a un archivo, para su inclusión en informes posteriores.

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.



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.

Comando/Sintaxis	Descripción	Ejemplos
cat <i>fich1</i> [... <i>fichN</i>]	Concatena y muestra un archivos	cat /etc/passwd
	archivos	cat dict1 dict2 dict
cd [<i>dir</i>]	Cambia de directorio	cd /tmp
chmod <i>permisos fich</i>	Cambia los permisos de un archivo	chmod +x miscript
chown <i>usuario:grupo fich</i>	Cambia el dueño un archivo	chown nobody miscript
cp <i>fich1...fichN dir</i>	Copia archivos	cp foo foo.backup
diff [-e] <i>arch1 arch2</i>	Encuentra diferencia entre archivos	diff foo.c newfoo.c
du [-sabr] <i>fich</i>	Reporta el tamaño del directorio	du -s /home/
file <i>arch</i>	Muestra el tipo de un archivo	file arc_desconocido
find <i>dir test acción</i>	Encuentra archivos.	find . -name ``.bak" – print
grep [-cilmv] <i>expr archivos</i>	Busca patrones en archivos	grep mike /etc/passwd
head -count <i>fich</i>	Muestra el inicio de un archivo	head prog1.c
mkdir <i>dir</i>	Crea un directorio.	mkdir temp
mv <i>fich1 ...fichN dir</i>	Mueve un archivo(s) a un directorio	mv a.out prog1
mv <i>fich1 fich2</i>	Renombra un archivo.	mv .c prog_dir
less / more <i>fich(s)</i>	Visualiza página a página un archivo.	more muy_largo.c
	less acepta comandos vi.	less muy_largo.c
ln [-s] <i>fich acceso</i>	Crea un acceso directo a un archivo	ln -s /users/mike/.profile .



ls	Lista el contenido del directorio	ls -l /usr/bin
pwd	Muestra la ruta del directorio actual	Pwd
rm <i>fich</i>	Borra un fichero.	rm foo.c
rm -r <i>dir</i>	Borra un todo un directorio	rm -rf prog_dir
rmdir <i>dir</i>	Borra un directorio vacío	rmdir prog_dir
tail -count <i>fich</i>	Muestra el final de un archivo	tail prog1.c
vi <i>fich</i>	Edita un archivo.	vi .profile

Comandos Linux/Unix de manipulación de archivos y directorios:

Comando/Sintaxis	Descripción	Ejemplos
at [-lr] <i>hora</i> [<i>fecha</i>]	Ejecuta un comando mas tarde	at 6pm Friday miscript
cal [[<i>mes</i>] <i>año</i>]	Muestra un calendario del mes/año	cal 1 2025
date [<i>mmddhhmm</i>] [+ <i>form</i>]	Muestra la hora y la fecha	Date
echo <i>string</i>	Escribe mensaje en la salida estándar	echo ``Hola mundo"
finger <i>usuario</i>	Muestra información general sobre un usuario en la red	finger nn@maquina.aca.com.co
id	Número id de un usuario	id usuario
kill [-señal] <i>PID</i>	Matar un proceso	kill 1234
man <i>comando</i>	Ayuda del comando especificado	man gcc man -k printer
passwd	Cambia la contraseña.	passwd
ps [<i>axiu</i>]	Muestra información sobre los procesos que se están ejecutando en el sistema	ps -ux
who / rwho	Muestra información de los usuarios conectados al sistema.	who