



16 de julio de 2021
Ficha N° 12 SHCHECK de Comandos
CSIRT DE GOBIERNO

Comando de la semana “SHCHECK”

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 Scripting¹, 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. Una de las problemáticas que afecta a sitios y sistemas web es la posibilidad que actores maliciosos secuestren la información de los usuarios que los acceden, mediante la utilización de la técnica Clickjacking.

Clickjacking, una palabra compuesta por click y hijack (secuestrar) referenciando así al “secuestro de los clicks”, también conocido como "ataque de compensación de UI²", es cuando un atacante usa varias capas transparentes u opacas para engañar a un usuario. Pero en este caso no todo el problema

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

² UI se refiere a “User Interface” o en español Interfaz del Usuario, es la vista que permite a un usuario interactuar de manera efectiva con un sistema. Es la suma de una arquitectura de información más patrones de interacción y elementos visuales.



recae en la programación del sitio o sistema web, pues como se expondrá más adelante en la parte servidora se ´pueden levantar controles que ayudan a mitigar esta vulnerabilidad.

En este contexto ya se puede visualizar lo relevante que es identificar esta vulnerabilidad, y por tanto es muy importante poder entenderla, identificarla, detectarla y mitigarla.

¿Qué es Clickjacking?

Es una técnica que puede usarse para acciones maliciosas cuando un atacante usa varias capas transparentes u opacas para engañar a un usuario para que haga click en un botón o enlace en otra página cuando intenta hacer clic en la página del nivel superior. Por lo tanto, el atacante está "secuestrando" los clics destinados a su página y enrutando a otra página, muy probablemente propiedad de otra aplicación, dominio o ambos.

Usando una técnica similar, las pulsaciones de teclas también pueden ser secuestradas. Con una combinación cuidadosamente elaborada de hojas de estilo, iframes y cuadros de texto, se puede hacer creer a un usuario que está escribiendo la contraseña en su correo electrónico o cuenta bancaria, pero en cambio está escribiendo en un marco invisible controlado por el atacante.

Pero antes de profundizar en la técnica misma, estableceremos algunos conceptos necesarios para entender el problema y la solución del mismo.

Los sitios y sistema web en general se construyen utilizando un lenguaje llamado HTML³ (siglas en inglés de HyperText Markup Language) y que actualmente se encuentra en su versión 5. HTML es el lenguaje de marcado central de la World Wide Web. Originalmente, HTML fue diseñado principalmente como un lenguaje para describir semánticamente documentos científicos. Sin embargo, su diseño general ha permitido adaptarlo, durante los años siguientes, para describir otros tipos de documentos e incluso aplicaciones.

La estructura básica de una página web es la siguiente:

```
<html>

<head></head>

<body>

</body>

</html>
```

³ <https://html.spec.whatwg.org/multipage/>



Una estructura HTML se empieza con la etiqueta `<html>` y acaba con `</html>`. Todo lo que esté en medio será la página web. Dentro de `<html></html>` se encuentran dos partes diferenciadas. La primera `<head></head>` es la cabecera de la página. Aquí irá cierta información que no es directamente el contenido de la página. Aquí se pone el título de la página, los metadatos, estilos, código javascript (todo esto se estudiará en capítulos venideros). La primera que se suele estudiar es `<title></title>`, que indica el título de la página (lo que el navegador pone en la parte superior izquierda).

La segunda parte es `<body></body>`. Aquí va propiamente el contenido de la página: fotos, párrafos, formularios, etc. Por ejemplo, siguiendo con el ejemplo de la página anterior, el siguiente código, podemos cambiar el título de la página.

```
<html>

<head>

<title>Esto es el título de la página.</title>

</head>

<body>

Hola mundo!<br>

<b>Esto es negrita.</b><br>

<i>Y esto itálica.</i><br>

</body>

</html>
```

Es importante mencionar que las etiquetas se pueden escribir indistintamente en mayúsculas o minúsculas, es decir `Esto es negrita.` y `Esto es negrita.` produce el mismo resultado. Por otro lado, toda etiqueta que se abre (es decir, se pone la etiqueta sin la barra /) debe cerrarse (es decir, poner su equivalente con el símbolo /), si no, el navegador podría dar resultados inesperados. Excepciones a esto son algunas etiquetas que no lo necesitan, como `
` o `<hr>`.



Dentro de este lenguaje existe un elemento denominado IFRAME⁴. Este es un elemento HTML que nos permite incrustar en un marco, dentro de nuestra página web, contenido de otra página web, y poder verlo sin problema. IFRAME es la abreviatura de Inline Frame (marco incorporado).

Los Atributos globales de iframe son:

- src - Dirección del recurso
- srcdoc - Un documento para renderizar en el iframe
- name- Nombre del contexto de navegación anidado
- sandbox - Reglas de seguridad para contenido anidado
- allow- Política de permisos que se aplicará a los iframedocumentos de
- allowfullscreen- Ya sea para permitir el iframedocumento del contenido de.requestFullscreen()
- width - Dimensión horizontal
- height - Dimensión vertical
- referrerpolicy- Política de referencia para las recuperaciones iniciadas por el elemento.
- loading - Se utiliza para determinar el aplazamiento de carga.

Algunos ejemplos para ilustrar su utilización son los siguientes:

Una página que utiliza iframe para incluir publicidad de un corredor de publicidad:

```
<iframe src="https://ads.example.com/?customerid=923513721&format=banner"
width="468" height="60"></iframe>
```

En este ejemplo, iframedocumento se utiliza para incrustar un mapa de un servicio de navegación en línea. El allowatributo se utiliza para habilitar la API de geolocalización dentro del contexto anidado.

```
<iframe src="https://maps.example.com/" allow="geolocation"></iframe>
```

Aquí, iframedocumento se usa para insertar un reproductor de un sitio de videos. El allowfullscreenatributo es necesario para permitir que el reproductor muestre su video en pantalla completa.

```
<article>
<header>
<p> <b>Fred Flintstone</b></p>
<p><a href="/posts/3095182851" rel=bookmark>12:44</a> — <a href="#acl-
3095182851">Private Post</a></p>
</header>
<p>Check out my new ride!</p>
```

⁴ <https://html.spec.whatwg.org/multipage/iframe-embed-object.html#the-iframe-element>



```
<iframe src="https://video.example.com/embed?id=92469812" allowfullscreen></iframe>
</article>
```

El problema es que estas técnicas pueden usarse de manera maliciosa, y de manera engañosa utilizarlos para engañar a los usuarios y hacerlos pensar, pues visualmente verán que están visitando el sitio web A (<https://www.A.cl>), pero en realidad los click's los están enviando a un sitio web B (<https://www.B.cl>).

Por ejemplo, una implementación que puede ayudar a verificar y demostrar que algunos sitios web es maliciosas es la siguiente:

```
<html>
<meta charset="UTF-8">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<!--<script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>-->

<script type="text/javascript">
    function cargasisito() {
        var sitio = prompt("Ingrese sitio a Validar");
        //swal("", "Ingrese sitio a Validar" , "info" , {content: "input" ,});

        ejemplo = sitio;
        document.getElementById('iframe').src = sitio;
        document.getElementById('idP').innerHTML = "Sitio a Validar : " + sitio;

        //document.getElementById("targetdiv").contentWindow.document.body.onclick
        = alert("hacked");
    }

    $(document).ready(function() {
    $('#targetdiv2').click(function(e) {
        alert("Sitio CLICKJACKING !");
        //swal ("CLICKJACKING" , "Sitio CLICKJACKING !" , "warning");

    });
    });

</script>

<style>
    #targetdiv1 {
        align-content: center;
```



```
height: 75%;  
width: 55%;  
display: block;  
margin: 0;  
padding: 0px 0px 0px 0px;  
position: absolute;  
z-index: 0;  
background-color: #dbdbdb;  
text-align: center;  
font-family: Arial, Helvetica, Sans-serif;  
font-size: 15pt;  
}
```

```
#targetdiv2 {  
height: 73%;  
width: 53%;  
display: block;  
margin: 0;  
padding: 0px 0 0px 0;  
position: relative;  
top: 11%;  
left: 0px;  
z-index: 9999;  
background-color: red;  
opacity: 0.3;  
}
```

```
#targetdiv3 {  
height: 5%;  
width: 53%;  
display: block;  
margin: 0;  
padding: 0px 0 0px 0;  
position: relative;  
top: 6%;  
left: 0px;  
z-index: 999999;  
background-color: ;  
text-align: center;  
text-decoration-color: black;  
font-family: Arial, Helvetica, Sans-serif;  
font-size: 15pt;  
}
```



```
iframe{
  height: 80%;
  width: 97%;
  display: block;
  margin: 1rem;
  padding: 0rem;
  position: absolute;
  z-index: 0;
  /*background-color: ; */
  border: 1px;
  background-color;;
}

</style>

<head >
<!-- INICIO SECCIÓN CARGA PÁGINA NORMAL -->

<body onload="cargasitio()">

<!-- FIN SECCIÓN CARGA PÁGINA NORMAL -->

<!-- INICIO SECCIÓN POTENCIALMENTE MALICIOSA -->
  <div id="targetdiv1" >
    <label> PRUEBA CLICKJACKING </label><br>
    <label id="idP"></label>

    <iframe id="iframe" ></iframe>
  </div>

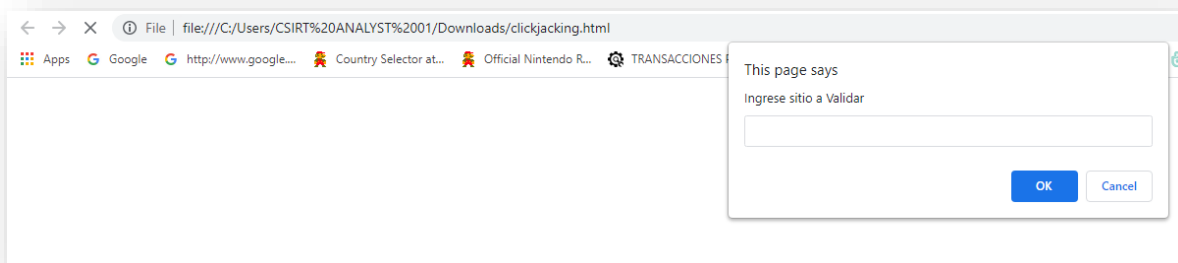
<div id="targetdiv2"> </div>

<div id="targetdiv3">
  <label> Capa para efectuar el CLICKJACKING. </label>
  <br>
  <br>
  <br>
  <input type="button" value="Refrescar" onclick="javascript:window.location.reload();"/>
</div>
<!-- FIN SECCIÓN POTENCIALMENTE MALICIOSA -->

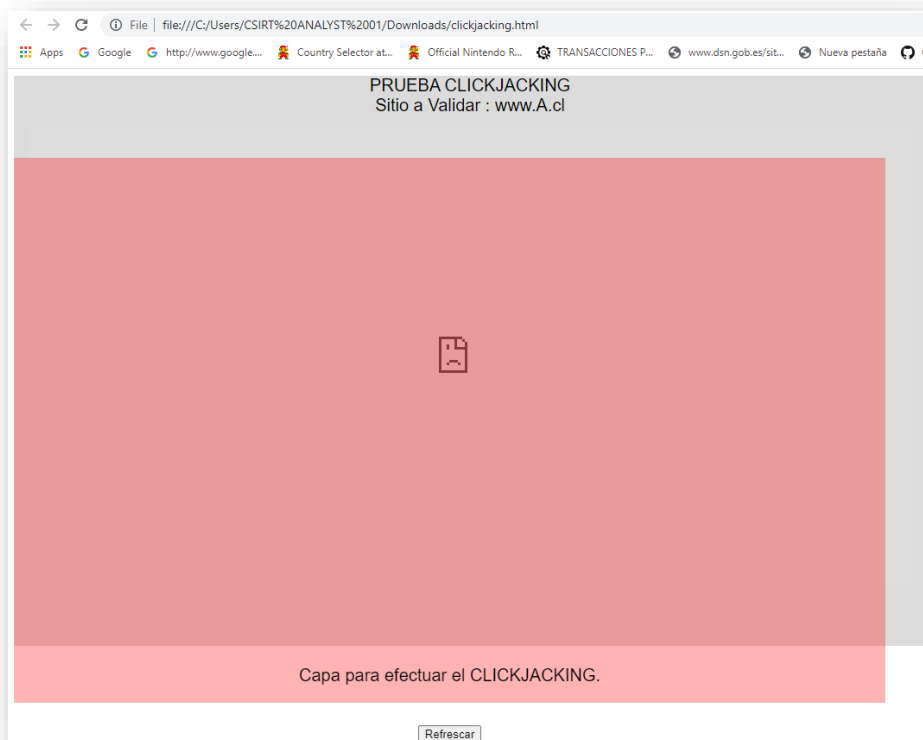
</body>
</html>
```



Lo que visualmente debiera desplegar algo como lo siguiente:



Luego de ingresar una URL (un sitio web inexistente www.A.cl) para probar, se observa algo como lo siguiente:



En esta codificación se puede observar una función JavaScript que, para efectos de demostración, pide ingresar una URL mediante un cuadro de dialogo emergente. Una vez ingresado por el usuario



la URL del sitio o sistema a probar, se despliega el sitio web y por sobre este despliegue se activa el segmento de codificación potencialmente maliciosa, presentando un cuadro semi transparente por sobre la imagen del sitio web cargado previamente. El problema es que esa zona semitransparente está en control del actor malicioso y NO de la parte dueña del sitio web original. Por lo tanto, con una codificación especialmente adaptada al sitio web que se busca atacar se puede engañar al usuario haciéndolo pensar que esta frente al sitio web original, pero todos los datos y clicks que están enviando pueden ir a parar a las manos (o servidores) del actor malicioso.

Entonces una vez que hemos logrado entender los peligros del IFRAME y de si nuestro sitio web, que, aunque no ocupe IFRAME, puede ser atacado con esta técnica, resulta relevante poder detectar si somos vulnerables o no y mitigar la vulnerabilidad lo antes posible.

Para esto estudiaremos un comando que no es nativo de KALI pero que es muy simple y directo para la verificación, y también algunos ejemplos alternativos utilizando herramientas nativas de KALI.

¿Qué es SHCHECK?

Es una herramienta simple, escrita en lenguaje PYTHON, para escanear los encabezados de seguridad de cualquier sitio web, independiente del webserver o sistema operativo sobre el que esté construido.

Los encabezados de seguridad HTTP siempre proporcionan una capa adicional de seguridad al ayudar a mitigar los ataques y las vulnerabilidades de seguridad.

Durante los últimos años, se han introducido una serie de nuevos encabezados HTTP cuyo propósito es ayudar a mejorar la seguridad de un sitio web.

HTTP headers

Las cabeceras (en inglés headers) HTTP permiten al cliente y al servidor enviar información adicional junto a una petición o respuesta. Una cabecera de petición esta compuesta por su nombre (no sensible a las mayusculas) seguido de dos puntos ':', y a continuación su valor (sin saltos de línea). Los espacios en blanco a la izquierda del valor son ignorados

Se pueden agregar cabeceras propietarias personalizadas usando el prefijo 'X-', pero esta convención se encuentra desfasada desde Julio de 2012, debido a los inconvenientes causados cuando se estandarizaron campos no estandar en el RFC 6648⁵; otras están listadas en un registro IANA⁶, cuyo

⁵ <https://tools.ietf.org/html/rfc6648>

⁶ <https://www.iana.org/assignments/message-headers/perm-headers.html>



contenido original fue definido en el RFC 4229⁷, IANA también mantiene un registro de propuestas para nuevas cabeceras HTTP⁸.

Las Cabeceras pueden ser agrupadas de acuerdo a sus contextos:

- Cabecera general: Cabeceras que se aplican tanto a las peticiones como a las respuestas, pero sin relación con los datos que finalmente se transmiten en el cuerpo.
- Cabecera de consulta: Cabeceras que contienen más información sobre el contenido que va a obtenerse o sobre el cliente.
- Cabecera de respuesta: Cabeceras que contienen más información sobre el contenido, como su origen o el servidor (nombre, versión, etc.).
- Cabecera de entidad: Cabeceras que contienen más información sobre el cuerpo de la entidad, como el tamaño del contenido o su tipo MIME.

Las cabeceras también pueden clasificarse de acuerdo a cómo se comportan frente a ellas los proxies:

- Cabeceras de extremo a extremo
Estas cabeceras deben ser enviadas al recipiente final del mensaje; esto es, el servidor (para una petición) o el cliente (para una respuesta). Los proxies intermediarios deben transmitir las cabeceras de extremo-a-extremo sin modificar, y las cachés deben guardarlas tal y como son recibidas.
- Cabeceras de paso
Estas cabeceras sólo son significativas para conexiones de un paso, y no deben ser transmitidas por proxies o almacenarse en caché. Éstas cabeceras son: Connection (en-US), Keep-Alive, Proxy-Authenticate (en-US), Proxy-Authorization (en-US), TE (en-US), Trailer (en-US), Transfer-Encoding and Upgrade (en-US). La cabecera general Connection (en-US) sólo puede usarse para este tipo de cabeceras.

La siguiente lista agrupa las cabeceras HTTP en categorías según su uso:

- Autenticación
- Almacenamiento en caché
- Indicaciones sobre el cliente
- Condicionales
- Gestión de conexiones
- Negociación de contenido
- Controles
- Cookies

⁷ <https://tools.ietf.org/html/rfc4229>

⁸ <https://www.iana.org/assignments/message-headers/prov-headers.html>



- CORS
- Cabeceras sin seguimiento
- Descargas
- Mensajes sobre la información del cuerpo (body)
- Proxies
- Redirecciones
- Contexto de petición
- Contexto de respuesta
- Peticiones de rango
- Seguridad
- Codificación de transferencia
- WebSockets
- Otros

Para efecto de este informe y vulnerabilidad específica, nos concentraremos en los HEADERS de seguridad:

Seguridad

Content-Security-Policy (CSP (en-US))

Controla qué recursos puede cargar el usuario para una página concreta.

Content-Security-Policy-Report-Only (en-US)

Permite a los desarrolladores web experimentar con políticas de acceso, monitorizando (pero sin implementar) sus efectos. Éstos informes de violación de protocolo contienen documentos del tipo JSON enviados mediante una petición HTTP POST hacia el URI especificado.

Expect-CT (en-US)

Permite a los sitios optar por informar y/o hacer cumplir los requerimientos de Transparencia de Certificados, lo que impide que el uso de certificados publicados incorrectamente por ese sitio, pase desapercibido. Cuando un sitio habilita el encabezado Expect-CT, se solicita a Chrome que verifique que cualquier certificado para ese sitio, aparezca en los registros públicos de CT.

Public-Key-Pins (en-US) (HPKP (en-US))

Asocia una clave criptográfica pública y específica con un determinado servidor web para reducir el riesgo de MITM ataques con certificados falsificados.

Public-Key-Pins-Report-Only (en-US)



Envía reportes al report-uri especificado en la cabecera, sin bloquear la conexión entre cliente y servidor aún cuando el pinning ha sido violado.

Strict-Transport-Security (HSTS (en-US))

Fuerza la comunicación utilizando HTTPS en lugar de HTTP.

Upgrade-Insecure-Requests (en-US)

Envía una señal al servidor expresando la preferencia del cliente por una respuesta encriptada y autenticada, y esta puede manejar de forma satisfactoria la directiva upgrade-insecure-requests (en-US).

X-Content-Type-Options

Deshabilita el MIME sniffing y fuerza al navegador a utilizar el tipo establecido en Content-Type.

X-Frame-Options (XFO)

Le indica al navegador que debe renderizar una página utilizando <frame>, <iframe> o <object>.

X-XSS-Protection

Habilita los filtros de cross-site scripting.

SHCHECK en la versión actual nos ayuda a verificar rápidamente los siguientes HEADERS:

- X-XSS-Protection
- X-Frame-Options
- X-Content-Type-Options
- Strict-Transport-Security
- Content-Security-Policy
- X-Permitted-Cross-Domain-Policies
- Referrer-Policy
- Expect-CT
- Permissions-Policy
- Cross-Origin-Embedder-Policy
- Cross-Origin-Resource-Policy
- Cross-Origin-Opener-Policy



Si bien todos los header son importantes para la seguridad y ayudan a defender su sitio o sistema web contra diferentes tipos de ataques, para efectos de la vulnerabilidad clickjacking nos concentraremos en detectar la presencia y configuración de “X-Frame-Options”.

X-Frame-Options

El encabezado de respuesta HTTP X-Frame-Options puede ser usado para indicar si debería permitírsele a un navegador renderizar una página en un <frame>, <iframe> o <object> . Las páginas webs pueden usarlo para evitar ataques de clickjacking , asegurándose que su contenido no es embebido en otros sitios.

La seguridad añadida sólo es proporcionada si el usuario que está accediendo al documento está utilizando un navegador que soporte X-Frame-Options.

Existen tres posibles directivas para X-Frame-Options:

- X-Frame-Options: DENY
- X-Frame-Options: SAMEORIGIN
- X-Frame-Options: ALLOW-FROM <https://example.com/>

Directivas

Si especifica DENY, fallarán no sólo los intentos de cargar la página en un marco desde otros sitios, sino que fallarán cuando sea cargada desde el mismo sitio. Por otro lado, si especifica SAMEORIGIN, puede usar la página en un marco mientras el sitio que la incluya sea el mismo que la sirve.

- DENY
La página no puede ser mostrada en un marco, independiente del sitio que esté intentándolo.
- SAMEORIGIN
La página sólo puede ser mostrada en un marco del mismo origen que dicha página.
- ALLOW-FROM uri
La página sólo puede ser mostrada en un marco del origen especificado. Tenga en cuenta que en Firefox esto todavía sufre del mismo problema que SAMEORIGIN — no verifica los antecesores del marco para ver si están en el mismo origen.

Nota: ¡Configurar el tag meta es inútil! Por ejemplo, <meta http-equiv="X-Frame-Options" content="deny"> no tiene efecto. ¡No lo use!

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.

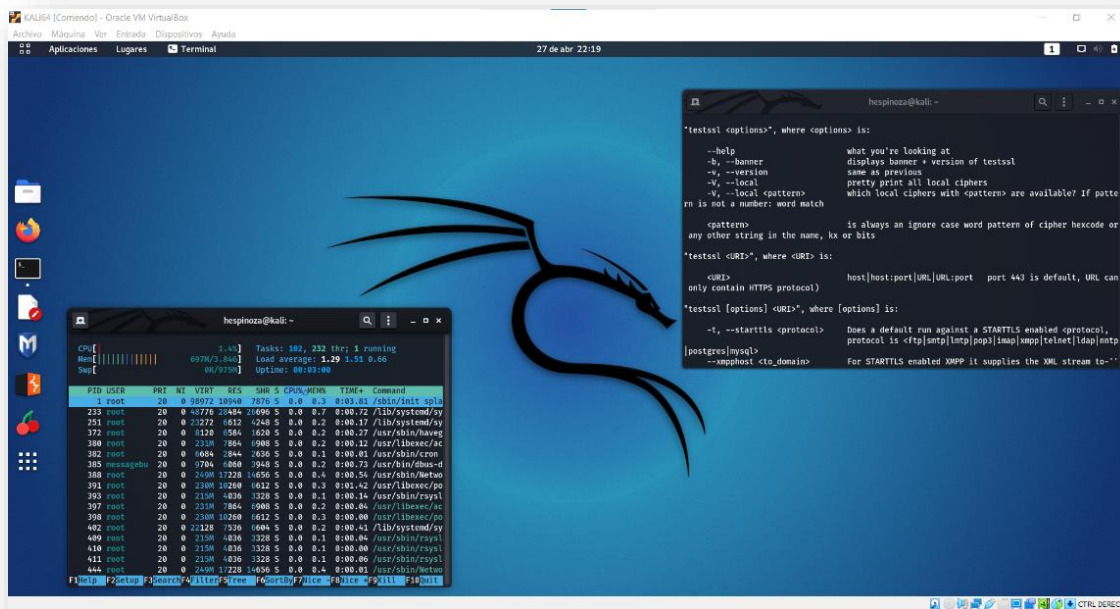
Vea anexo I: Comandos básicos de Linux



III. PASO A PASO

PASO 1: Un entorno adecuado para trabajar.

Primero debe contar con una distribución de Kali⁹ Linux funcionando ya sea en una máquina física o en una máquina virtual¹⁰¹¹.



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

⁹ <https://www.kali.org/downloads/>
10

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

¹¹ <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 instalación¹²

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 Linux¹³ (el fabricante recomienda¹⁴ la imagen marcada como Instalador).

¹² 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.

¹³ <https://www.kali.org/docs/introduction/download-official-kali-linux-images/>

¹⁴ <https://www.kali.org/docs/introduction/what-image-to-download/#which-image-to-choose>



-  Grabe¹⁵ 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 red¹⁶ 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.

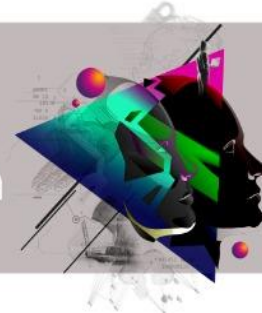
Una vez que tiene preparado todos los materiales y el entorno para comenzar la instalación siga los pasos indicados en la sección “Kali Linux Installation Procedure” del siguiente enlace:

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



¹⁵ <https://www.kali.org/docs/usb/live-usb-install-with-windows/>

¹⁶ <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 KALI¹⁷, pero si no fuere así puede instalarlos con los siguientes comandos, **previamente tomando privilegios de usuario "root"**:

```
# apt update && apt full-upgrade

# git clone https://github.com/m3liot/shcheck.git
```

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# git clone https://github.com/m3liot/shcheck.git
Cloning into 'shcheck'...
remote: Counting objects: 57, done.
remote: Total 57 (delta 0), reused 0 (delta 0), pack-reused 57
Unpacking objects: 100% (57/57), done.
root@kali:~#
```

Ingreda al directorio "shcheck" (cd shcheck), y otorgue el permiso de ejecución al script shcheck.py con la ayuda de "chmod +x shcheck.py".

```
root@kali: ~/shcheck
File Edit View Search Terminal Help
root@kali:~# git clone https://github.com/m3liot/shcheck.git
Cloning into 'shcheck'...
remote: Counting objects: 57, done.
remote: Total 57 (delta 0), reused 0 (delta 0), pack-reused 57
Unpacking objects: 100% (57/57), done.
root@kali:~# cd shcheck/
root@kali:~/shcheck# ls
LICENSE.txt  README.md  screenshot.png  shcheck.py
root@kali:~/shcheck# chmod +x shcheck.py
root@kali:~/shcheck# ./shcheck.py
Usage: ./shcheck.py [options] <target>
```

¹⁷ <https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/>



PASO3: Verificar su instalación.

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

En una consola de su KALI, dentro del directorio donde quedó instalada la aplicación, ejecute el comando para que muestre la ayuda: “./shcheck.py -h”¹⁸.

```
root@kali: ~/shcheck
# ./shcheck.py -h
Usage: ./shcheck.py [options] <target>

Options:
  -h, --help                show this help message and exit
  -p PORT, --port=PORT      Set a custom port to connect to
  -c COOKIE_STRING, --cookie=COOKIE_STRING
                           Set cookies for the request
  -a HEADER_STRING, --add-header=HEADER_STRING
                           Add headers for the request e.g. 'Header: value'
  -d, --disable-ssl-check   Disable SSL/TLS certificate validation
  -g, --use-get-method      Use GET method instead HEAD method
  -j, --json-output         Print the output in JSON format
  -i, --information         Display information headers
  -x, --caching             Display caching headers
  -k, --deprecated          Display deprecated headers
  --proxy=PROXY_URL         Set a proxy (Ex: http://127.0.0.1:8080)
  --hfile=PATH_TO_FILE     Load a list of hosts from a flat file
  --colours=COLOURS        Set up a colour profile
  --colors=COLOURS         Alias for colours for US English

root@kali| ~ | ~/shcheck|
```

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

```
# ./shcheck.py -h
Usage: ./shcheck.py [options] <target>

Options:
  -h, --help                show this help message and exit
  -p PORT, --port=PORT      Set a custom port to connect to
  -c COOKIE_STRING, --cookie=COOKIE_STRING
                           Set cookies for the request
  -a HEADER_STRING, --add-header=HEADER_STRING
                           Add headers for the request e.g. 'Header: value'
  -d, --disable-ssl-check
```

¹⁸ La opción “-h” es relativamente estándar y cada comando debiera desplegar la ayuda de uso, en algunos casos deberá utilizar “--help”.



-g, --use-get-method	Disable SSL/TLS certificate validation
-j, --json-output	Use GET method instead HEAD method
-i, --information	Print the output in JSON format
-x, --caching	Display information headers
-k, --deprecated	Display caching headers
--proxy=PROXY_URL	Display deprecated headers
--hfile=PATH_TO_FILE	Set a proxy (Ex: http://127.0.0.1:8080)
--colours=COLOURS	Load a list of hosts from a flat file
--colors=COLOURS	Set up a colour profile
	Alias for colours for US English



Paso 4: Ponerlo en marcha para verificar nuestra infraestructura.

Un ejemplo de ejecución básica para nuestros primeros pasos:

Probaremos el comando SHCHECK con nuestro KALI en un ataque un sitio web determinado:

EJEMPLO SHCHECK

```
# ./shcheck.py -d -k https://www.A.cl  
-d, --disable-ssl-check  
-k, --deprecated          Display deprecated headers
```

```
root@kali: ~/shcheck  
root@kali) ~/shcheck  
# ./shcheck.py -d -k https://www.████████.cl  
  
===== > shcheck.py - santoru ..... =====  
===== Simple tool to check security headers on a webserver =====  
  
[*] Analyzing headers of https://www.████████.cl  
[*] Effective URL: https://www.████████.cl  
[!] Missing security header: X-XSS-Protection  
[*] Header X-Frame-Options is present! (Value: DENY)  
[!] Missing security header: X-Content-Type-Options  
[!] Missing security header: Strict-Transport-Security  
[!] Missing security header: Content-Security-Policy  
[!] Missing security header: X-Permitted-Cross-Domain-Policies  
[!] Missing security header: Referrer-Policy  
[!] Missing security header: Expect-CT  
[!] Missing security header: Permissions-Policy  
[!] Missing security header: Cross-Origin-Embedder-Policy  
[!] Missing security header: Cross-Origin-Resource-Policy  
[!] Missing security header: Cross-Origin-Opener-Policy  
  
[!] Headers analyzed for https://www.████████.cl  
[+] There are 1 security headers  
[-] There are not 11 security headers  
  
root@kali) ~/shcheck  
#
```

Para nuestro caso de estudio nos interesa que esté presente y configurado el HEADER que nos ayudará a proteger nuestro sitio o sistema web de Clickjacking, cuyos valores a configurar, **DENY** o



SAME ORIGIN o ALLOW-FROM, dependerán de si su sitio web o sistema necesita integrar o conectarse a otros sistemas mediante IFRAME:

[*] Header X-Frame-Options is present! (Value: DENY)

Otras alternativas para buscar el despliegue de los HEADERS de nuestro sitio o Sistema web son los siguientes que son nativos, en general, de KALI:

Comando wget

```
wget -S --spider -U "Mozilla/5.0 (Windows NT 6.1; rv:59.0) Gecko/20100101 Firefox/59.0" URL_SITIO
```

Comando curl

```
curl -I -A "Mozilla/5.0 (Windows NT 6.1; rv:59.0) Gecko/20100101 Firefox/59.0" URL_SITIO
```

Comando HEAD

HEAD URL_SITIO (al usar este comando se recomienda quitar el HTTPS y/o HTTP y los /)

Otros comandos que puede explorar son los siguientes:

- Hsecscan: <https://github.com/riramar/hsecscan>
- Headers: <https://github.com/oshp/headers/>
- http_hardening: https://github.com/amenezes/http_hardening

Estrategias de Mitigación:

Configurar “X-Frame-Options”, con alguno de los posibles valores DENY o SAMEORIGIN o ALLOW-FROM, lo que dependerá de si su sitio o sistema web necesita integrar o conectarse a otros sistemas mediante IFRAME.

Configurando Apache

Agregue lo siguiente a la configuración de su sitio para que Apache envíe el encabezado X-Frame-Options para todas las páginas, en el archivo de configuración “httpd.conf” o equivalente¹⁹:

Header always append X-Frame-Options SAMEORIGIN

¹⁹ También se puede realizar a través de .htaccess, security.conf o apache2.conf, dependiendo del servidor en el que se encuentre montado el sitio web.



Para que Apache envíe X-Frame-Options deny, agregue lo siguiente a la configuración de su sitio:

```
Header set X-Frame-Options DENY
```

Para que Apache envíe el encabezado X-Frame-Options para permitir (ALLOW-FROM) un host en específico, agregue esto a la configuración de su sitio:

```
Header set X-Frame-Options "ALLOW-FROM https://example.com/"
```

Configurando nginx

Para configurar nginx a que envíe el encabezado X-Frame-Options , agregue esto a la configuración, ya sea http, server o location:

```
add_header X-Frame-Options DENY;
```

Configurando IIS

Para hacer que IIS envíe el encabezado X-Frame-Options, agregue esto al archivo Web.config de su sitio:

```
<system.webServer>
...

<httpProtocol>
  <customHeaders>
    <add name="X-Frame-Options" value="SAMEORIGIN" />
  </customHeaders>
</httpProtocol>

...
</system.webServer>
```

Configurando HAProxy

Para hacer que HAProxy envíe el encabezado X-Frame-Options, agregue lo siguiente a su configuración front-end, listen, o backend:

```
rspadd X-Frame-Options:\ SAMEORIGIN
```




Nota: verifique la compatibilidad de estas opciones con los navegadores principales que usted cree utilizarán sus clientes.

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 PowerShell²⁰ (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.

²⁰ <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.