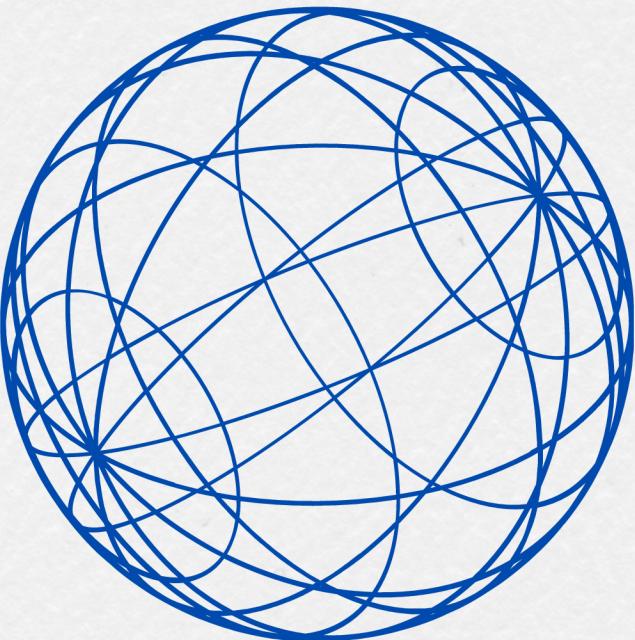


INFORME EJECUTIVO Y TÉCNICO



Host: Fuzzing

Team Challenge S7 - Jose Maria Jimenez

Indice

-Introducción General.....pag. 1

INFORME EJECUTIVO:

-Introducción.....pag. 2
-Alcance.....pag. 3
-Vulnerabilidades encontradas.....pag. 4
-Soluciones y Recomendaciones.....pag. 5

INFORME TÉCNICO:

-Proceso de Explotación.....pag. 6
-Vulnerabilidades encontradas.....pag. 15
-Soluciones y Conclusión.....pag. 16
-Anexo.....pag. 17

Introducción general:

Se va a proceder a la búsqueda y explotación de las vulnerabilidades en la máquina ‘Fuzzing’ aportada por la empresa The Bridge para su estudio y resolución de problemas.

El objetivo de ello será realizar diversas pruebas y métodos de exploración de servicios para descubrir que vulnerabilidades posee este host y poder llevar a cabo una explotación de ellas con una posterior evaluación y resolución de dichas vulnerabilidades.

-INFORME EJECUTIVO

Introducción:

Por petición de la empresa The Bridge se va a proceder a una prueba de pentesting a la máquina virtual ‘Fuzzing’ para poder valorar mediante diversas herramientas y métodos las vulnerabilidades existentes y posteriormente la explotación de ellas. Terminando con una resolución de las vulnerabilidades encontradas.

En primer lugar se procederá a conectar el host de la prueba ‘Fuzzing’ en la misma red que el otro host desde el que vamos a realizar la prueba de pentesting ‘Kali Linux’ en un entorno de laboratorio controlado para evitar cualquier conexión con el exterior de la red y así poder evitar que cualquier cibercriminal puedan acceder a ellas en el periodo de la prueba.

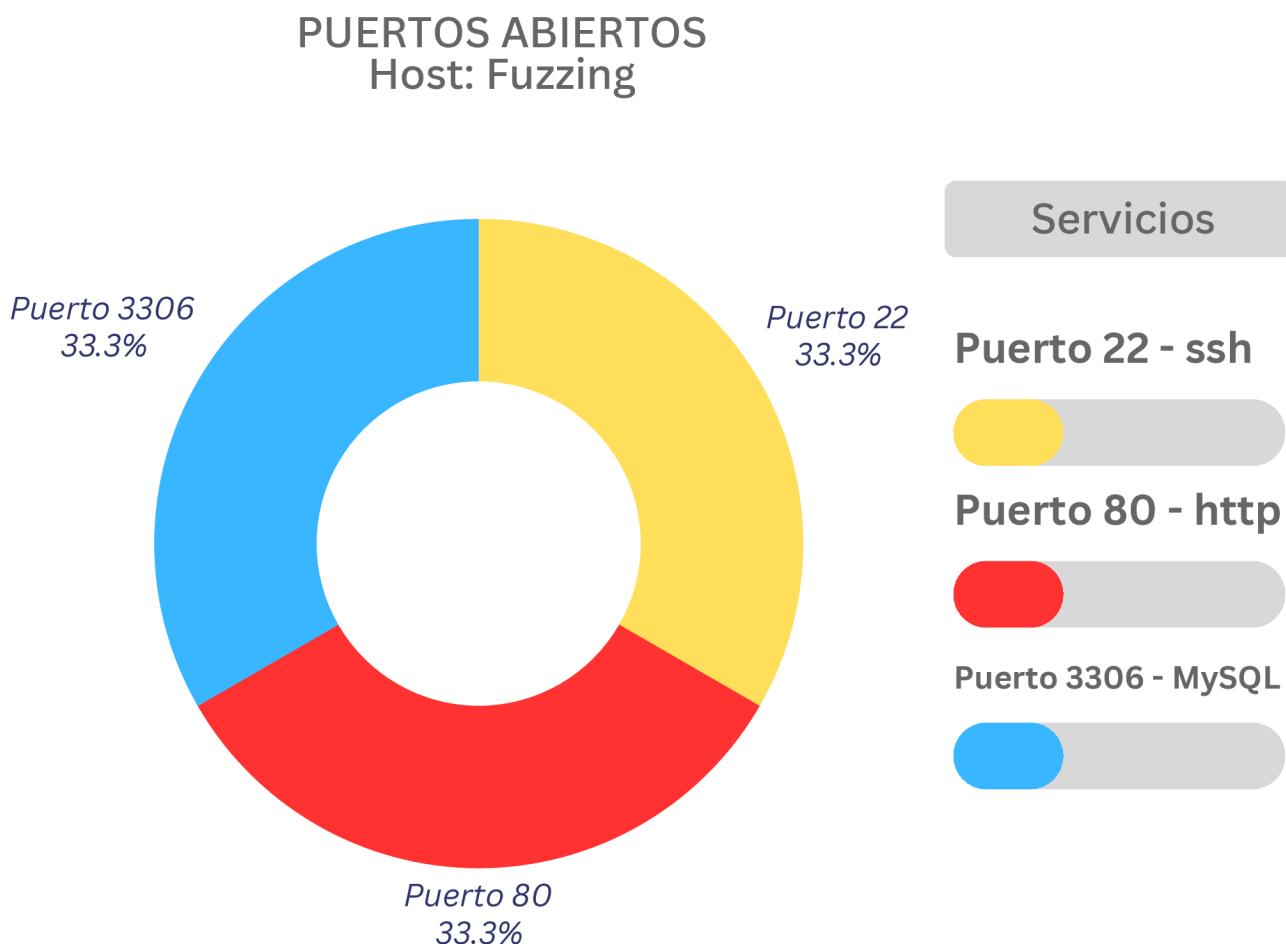
En primer lugar se procederá a realizar un escaneo de host en la misma red mediante ARP y una vez encontrado el host un análisis con la herramienta Nmap, con la cual podremos confirmar los puertos abiertos en el host ‘Fuzzing’ así como los servicios y versiones que se ejecutan en ellos para su posterior valoración de vulnerabilidades.

En segundo lugar se va a proceder a un análisis de Fuzzing con el cual dispondremos de información sobre directorios y archivos que puedan ser visibles o accesibles desde el navegador web y que puedan poner en peligro el host con información sensible donde un cibercriminal podría aprovechar estos datos con información para acceder al sistema y comprometerlo. Este análisis se efectuará con las herramienta Dir desde la terminal de Kali linux.

Por último también se usará BurpSuite para capturar y modificar peticiones al host con el objetivo final de obtener el usuario y contraseña del administrador del host.

Alcance:

Después de una exhaustiva búsqueda de vulnerabilidades sobre el host, se ha llegado a la conclusión. De los tres puertos encontrados abiertos el de mayor peligrosidad para utilizar como vector de ataque para un cibercriminal sería el puerto 80. El cual aloja un servicio http con una página web vulnerable ante un ciberataque.



Vulnerabilidades encontradas:

Se han encontrado varias vulnerabilidades en este estudio, las cuales se muestran a continuación:

- Archivos encontrados al hacer Fuzzing con información sensible:
 - (Credenciales de usuarios y contraseñas).
- Código web con poca seguridad permitiendo a un cibercriminal acceder al administrador del sistema:
 - (Editando solo la url del sitio web podría accederse a la cuenta de administrador).

VULNERABILIDADES ENCONTRADAS

Riesgos de explotación:



Soluciones:

A continuación se recomiendan soluciones para mitigar las vulnerabilidades encontradas.

- Archivos encontrados al hacer Fuzzing con información sensible:
 - (Credenciales de usuarios y contraseñas).

Recomendación: No disponer dentro del contenido de la web archivos con credenciales aunque estas no sean indexadas por los buscadores.

- Código web con poca seguridad permitiendo a un cibercriminal acceder al administrador del sistema:
 - (Editando solo la url del sitio web puedes acceder al usuario administrador).

Recomendación: Usar un código robusto para confeccionar la página web evitando que un cibercriminal pueda mediante la modificación de la url acceder a cambiar permisos de un usuario o administrador del sistema.

-INFORME TÉCNICO

Proceso de explotación:

En primer lugar se ha realizado un escaneo de la red mediante ARP para identificar si el host 'Fuzzing' se encontraba en ella para su posterior análisis.

```
(jose㉿kali)-[~]
$ sudo arp-scan -I eth0 -l
Interface: eth0, type: EN10MB, MAC: 08:00:27:d1:47:5a, IPv4: 10.0.2.14
WARNING: Cannot open MAC/Vendor file ieeeoui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10.0.2.1      52:54:00:12:35:00      (Unknown: locally administered)
10.0.2.3      08:00:27:52:c7:30      (Unknown)
10.0.2.2      52:54:00:12:35:00      (Unknown: locally administered)
10.0.2.15     08:00:27:24:46:64      (Unknown) •
```

En segundo lugar se ha realizado mediante la herramienta Nmap un escaneo de los puertos abiertos, servicios que trabajan en ellos y versiones.

```
(jose㉿kali)-[~]
$ sudo nmap -A 10.0.2.15
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-15 17:24 CEST
Nmap scan report for 10.0.2.15
Host is up (0.0020s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0) •
| ssh-hostkey:
|_ 256 95:cb:12:1d:90:1c:9b:3b:0d:66:f4:4d:1f:63:9b:1a (ECDSA)
|_ 256 66:f4:69:d5:ca:a1:39:75:da:02:74:d4:73:7a:4e:56 (ED25519)
80/tcp    open  http     Apache httpd 2.4.59 ((Debian)) •
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.59 (Debian)
| http-robots.txt: 2 disallowed entries
|_/wap/ /corp/
3306/tcp   open  mysql   MySQL (unauthorized) •
MAC Address: 08:00:27:24:46:64 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1  2.04 ms  10.0.2.15

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 53.06 seconds
```

Al descubrir que tiene el puerto 80 abierto con un servicio web http activo en él se ha procedido a usar la herramienta Dirb para examinar mediante fuzzing y utilizando el diccionario 'common.txt'. Con esta herramienta procederemos a la investigación de directorios ocultos al indexador para así poder acceder a ellos mediante el navegador web y descubrir información sensible.

En el cual se han hallado los tres marcados en la imagen con un punto verde como de mayor interés.

```
(jose@kali)-[~]
$ dirb http://10.0.2.15/
```

Papetera

```
DIRB v2.22
By The Dark Raver
```

START_TIME: Mon Jul 15 22:38:27 2024
URL_BASE: http://10.0.2.15/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

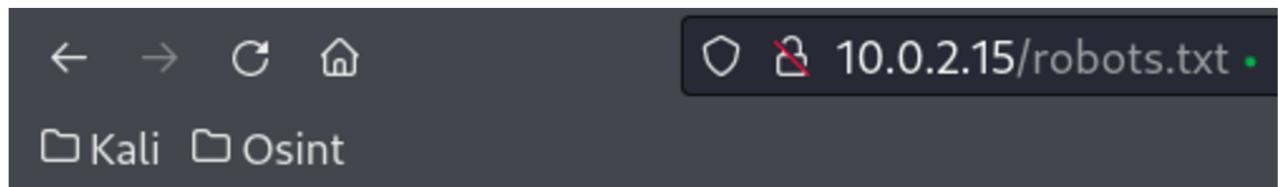
```
--- Scanning URL: http://10.0.2.15/ ---
==> DIRECTORY: http://10.0.2.15/corp/
+ http://10.0.2.15/index.html (CODE:200|SIZE:2511)
+ http://10.0.2.15/robots.txt (CODE:200|SIZE:47)
+ http://10.0.2.15/server-status (CODE:403|SIZE:274)
==> DIRECTORY: http://10.0.2.15/static/
==> DIRECTORY: http://10.0.2.15/wap/
```

```
--- Entering directory: http://10.0.2.15/corp/ ---
==> DIRECTORY: http://10.0.2.15/corp/config/
==> DIRECTORY: http://10.0.2.15/corp/conn/
==> DIRECTORY: http://10.0.2.15/corp/css/
==> DIRECTORY: http://10.0.2.15/corp/fonts/
==> DIRECTORY: http://10.0.2.15/corp/img/
==> DIRECTORY: http://10.0.2.15/corp/inc/
+ http://10.0.2.15/corp/index.php (CODE:200|SIZE:1601)
==> DIRECTORY: http://10.0.2.15/corp/js/
+ http://10.0.2.15/corp/LICENSE (CODE:200|SIZE:35149)
+ http://10.0.2.15/corp/Readme (CODE:200|SIZE:359)
==> DIRECTORY: http://10.0.2.15/corp/uploads/
```

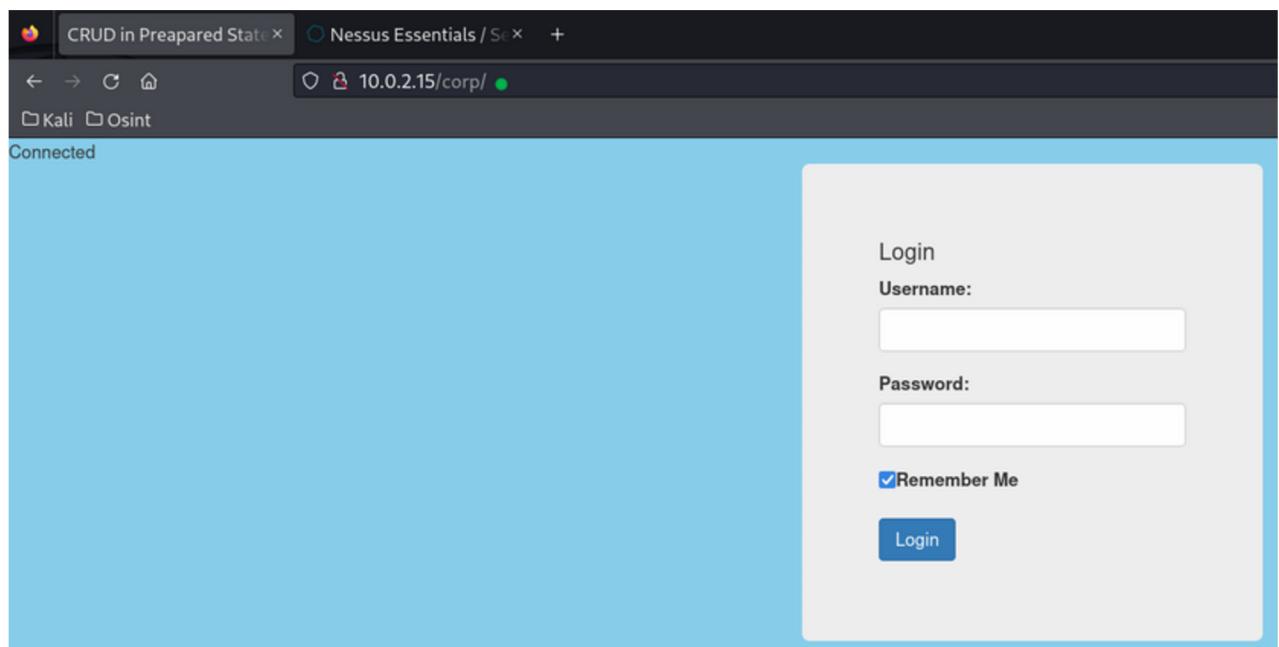
```
--- Entering directory: http://10.0.2.15/static/ ---
--- Entering directory: http://10.0.2.15/wap/ ---
+ http://10.0.2.15/wap/index.html (CODE:200|SIZE:4629)
+ http://10.0.2.15/wap/LICENSE (CODE:200|SIZE:1066)
```

```
--- Entering directory: http://10.0.2.15/corp/config/ ---
+ http://10.0.2.15/corp/config/users (CODE:200|SIZE:164)
```

Primero hemos accedido al 'robots.txt' usado normalmente por administradores de sitios web para indicar a los indexadores o motores de búsqueda que no puedan acceder a ellos. Aquí ya tenemos una vulnerabilidad que nos muestra un directorio que quiere ser ocultado y hemos podido acceder a él.



Como segundo paso hemos accedido al directorio '/corp' hemos dado con un panel de login para usuarios, el cual puede ser peligroso si un cibercriminal obtuviera las credenciales de usuario o administrador y pudiese acceder al panel de control de la web.



Y en tercer lugar de esta exploración de directorios se ha realizado la consulta a '/corp/conf/users' encontrado gracias a la herramienta Dirl. El cual nos muestra más información sobre directorios a los cuales poder tener acceso. Ambos con palabras clave para un cibercriminal para seguir con ese vector de ataque y exploración 'user01' y 'keys'

```
← → C ⌂
  10.0.2.15/corp/config/users •
  Kali Osint
```

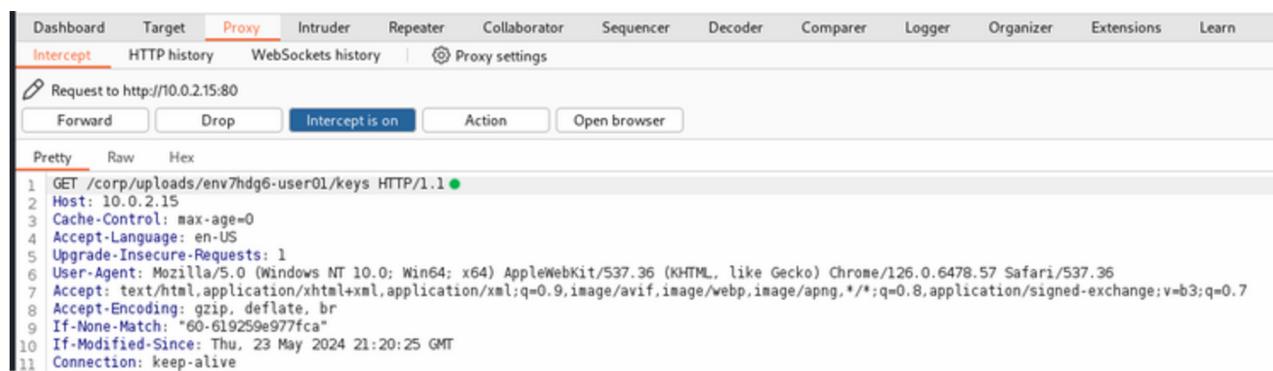
```
User: admin
Uploads: /corp/uploads/env7hdg6-user01 •
Avatar: /corp/uploads/env7hdg6-user01/avatar.png
Keys: /corp/uploads/env7hdg6-user01/keys •
Created: 12 Jun 2020
```

Accediendo al directorio 'corp/uploads/env7hdg6-user01/keys' encontramos el correo electrónico 'admin@gmail.com' con el cual se podría usar como usuario en el panel de login anteriormente encontrado en el directorio '/corp').

```
← → C ⌂
  10.0.2.15/corp/uploads/env7hdg6-user01/keys •
  Kali Osint
```

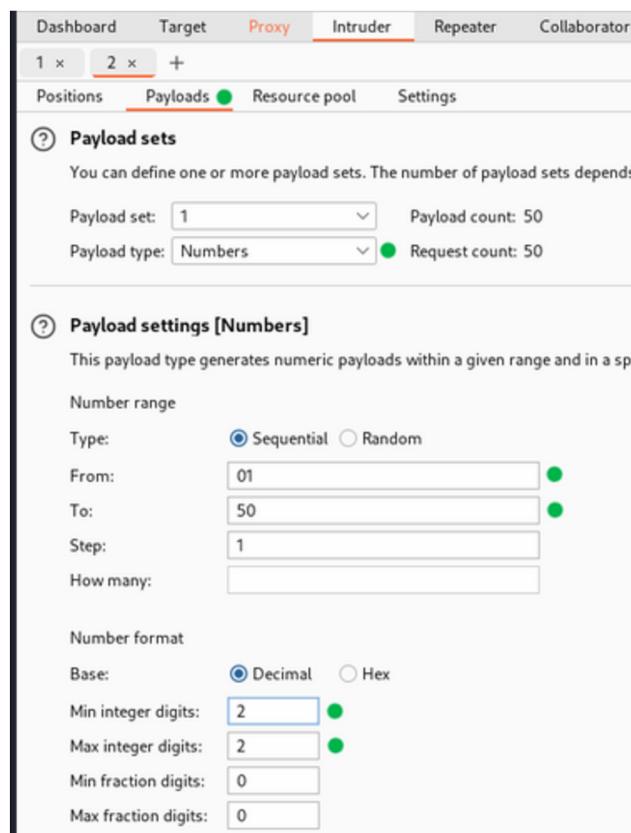
```
Server: PRO-wpaenv7
User: admin@gmail.com •
Pass: ***** // Obfuscated for security reasons.
```

Al observar en la url el ‘user01’ con los datos descubiertos anteriormente se intuye que deberían de haber más usuarios en el sistema los cuales podrían darnos más información. Para la exploración de más usuarios se ha procedido a utilizar la herramienta BurpSuite. Capturando el tráfico web con la petición sobre esa misma url encontrada anteriormente.



```
1 | GET /corp/uploads/env7hdg6-user01/keys HTTP/1.1 ●
2 | Host: 10.0.2.15
3 | Cache-Control: max-age=0
4 | Accept-Language: en-US
5 | Upgrade-Insecure-Requests: 1
6 | User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.57 Safari/537.36
7 | Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
8 | Accept-Encoding: gzip, deflate, br
9 | If-None-Match: "60-619259e977fca"
10 | If-Modified-Since: Thu, 23 May 2024 21:20:25 GMT
11 | Connection: keep-alive
```

Una vez obtenida la consulta original se va a proceder a la modificación de ella mediante la herramienta interna ‘intruder’ de Burpsuite. Con este paso buscaremos si hay más usuarios en el sistema configurando el payload para que busque entre los 01 y 50 usuarios posibles.

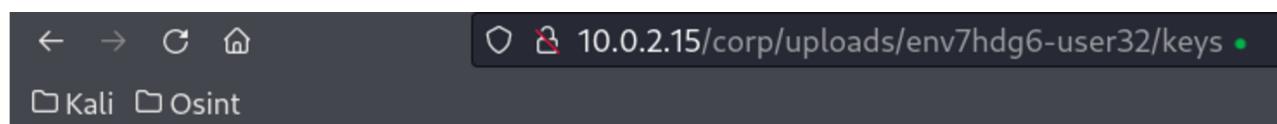


The screenshot shows the BurpSuite Intruder tab interface. It displays two payload sets, with the second one currently selected. The payload type is set to 'Numbers' and the request count is 50. In the 'Payload settings [Numbers]' section, the 'Type' is set to 'Sequential'. The 'From' field is set to '01', 'To' to '50', 'Step' to '1', and 'How many:' is empty. Under 'Number format', the base is set to 'Decimal' with 'Min integer digits' at 2 and 'Max integer digits' at 2. The 'Min fraction digits' and 'Max fraction digits' are both set to 0.

Una vez finalizado el ataque con BurpSuite se puede observar que el 'status code' del usuario número 32 es 200. Eso indica que el servicio con el usuario 32 está activo y existe.

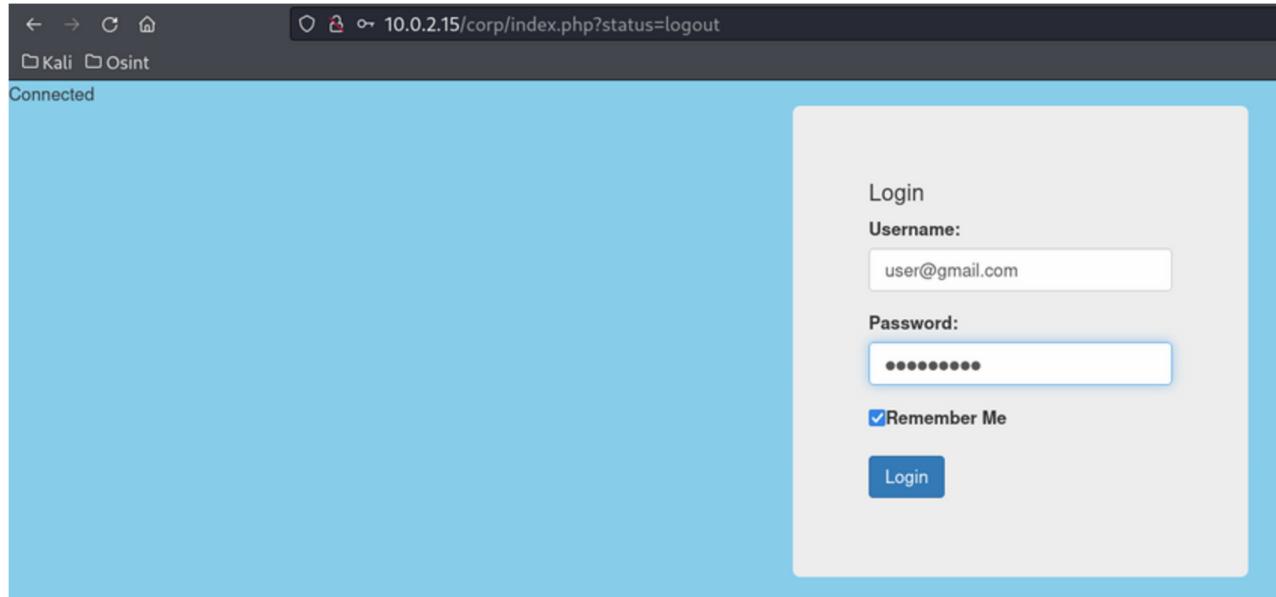
Results	Positions	Payloads	Resource pool	Settings
▼ Intruder attack results filter: Showing all items				
Request	Payload			Status code ^
32 ●	32 ●			200 ●
1	01			304
0				304
50	50			404
49	49			404
48	48			404
47	47			404
46	46			404
45	45			404
44	44			404
43	43			404
42	42			404
41	41			404
40	40			404
39	39			404
38	38			404
37	37			404
36	36			404
35	35			404
34	34			404
33	33			404
31	31			404
30	30			404
29	29			404
28	28			404

Con la información obtenida anteriormente se procede a cambiar en la url el usuario '01' por '32' encontrando información muy valiosa para un cibercriminal. Usuario y contraseña del 'user 32'.

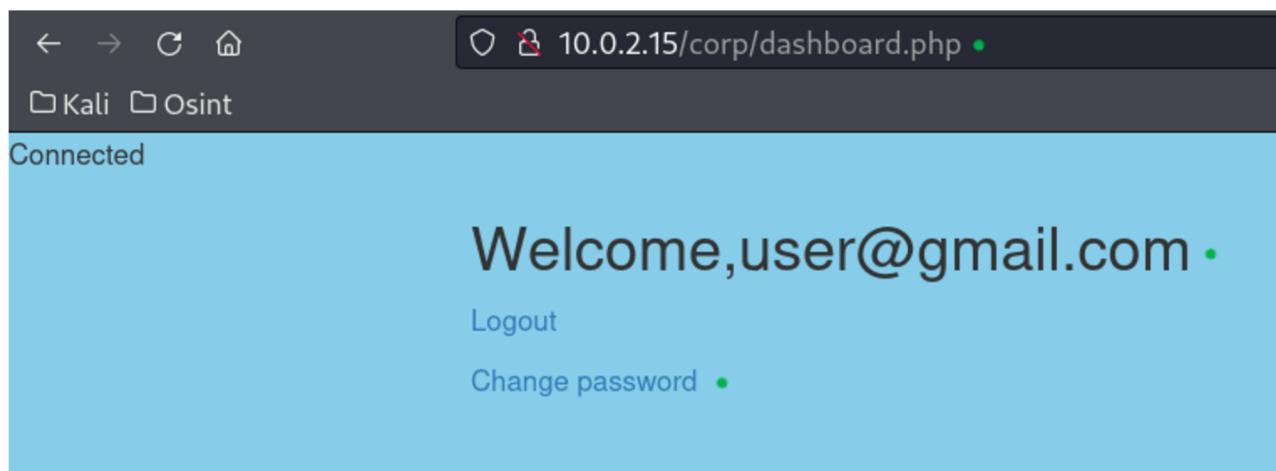


Con las credenciales obtenidas en el paso anterior se ha vuelto al panel de login para usarlas:

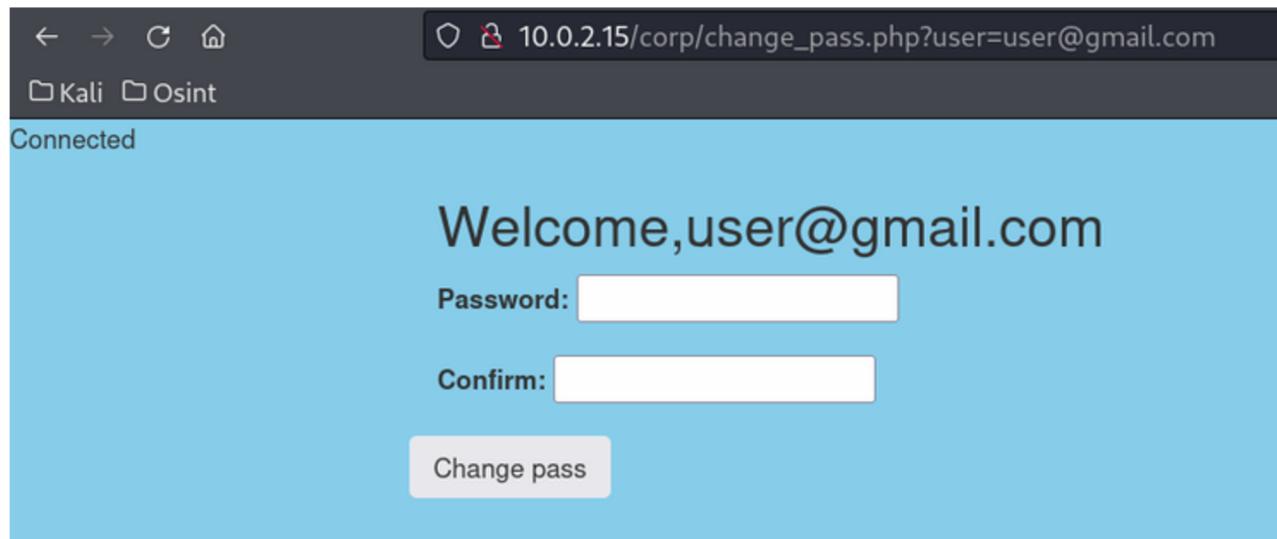
- Username: user@gmail.com
- Password: webcorp56



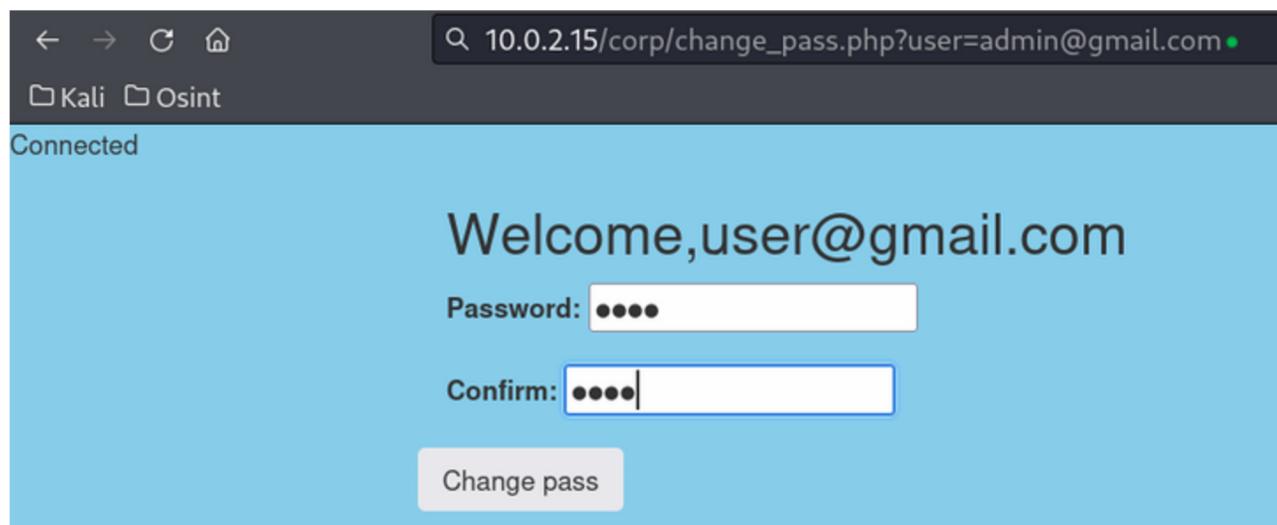
Teniendo éxito con ello y logrando acceder a los recursos del usuario 32.



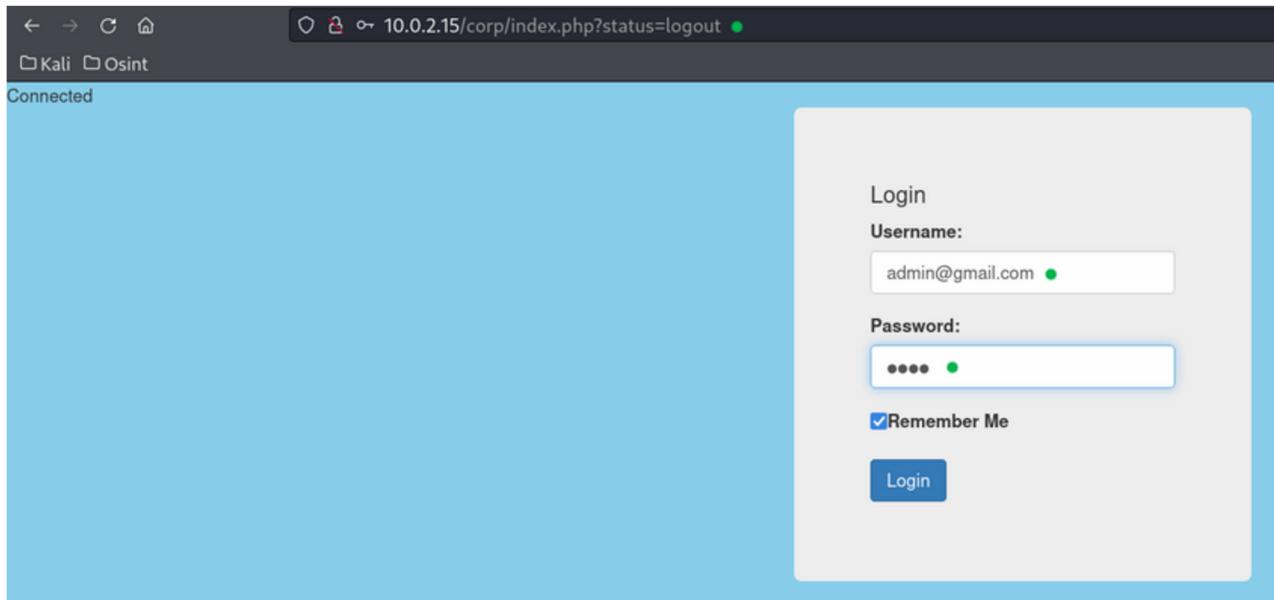
Al pulsar el botón de 'Change password' se ha podido acceder al cambio de contraseña de 'user@gmail.com'. Se puede observar en el link como la consulta que se está realizando en este recurso de cambio de contraseña aparece el nombre de usuario por completo.



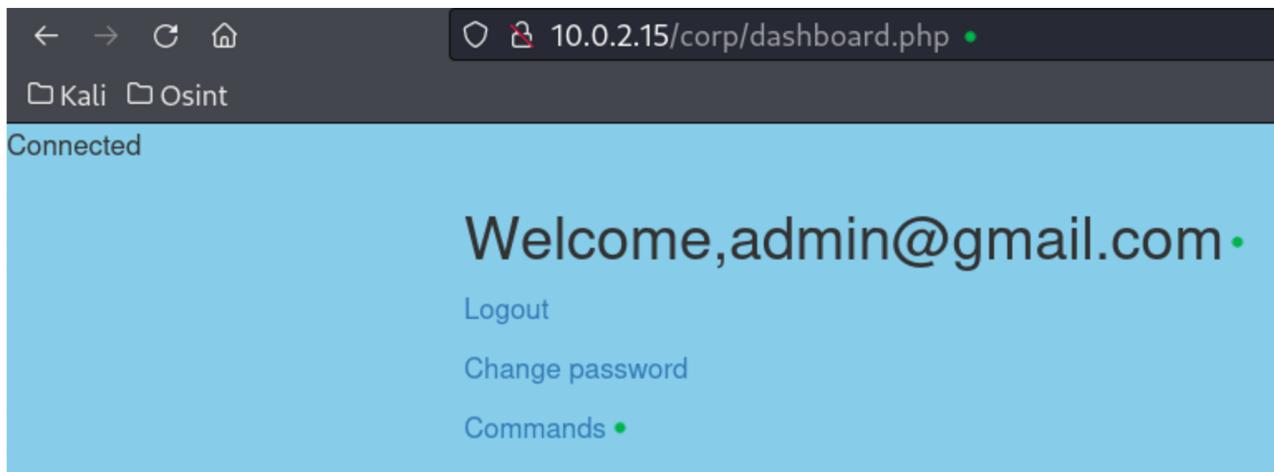
Viendo esto y conociendo el nombre de usuario del administrador 'admin@gmail.com' se va a proceder a la edición de la url para cambiar la contraseña del administrador en lugar de la del 'user@gmail.com' y así poder acceder a su panel de administración.



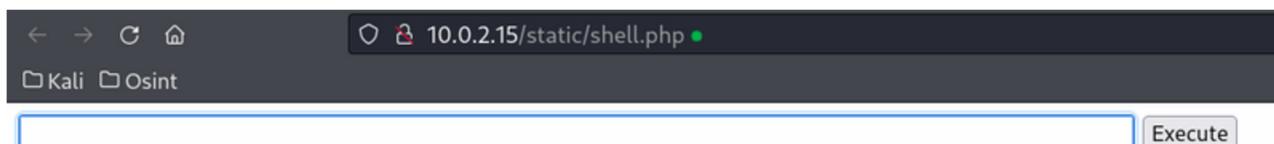
Una vez realizado este paso, se ha procedido al cierre de la sesión y posterior logueo con el usuario 'admin@gmail.com' y la contraseña cambiada.



Obteniendo la sesión del administrador y el control total del host.



Aquí la prueba de una webshell encontrada en el botón 'Commands' que solo tenía acceso el administrador del sistema. Desde aquí y con las credenciales de administrador es posible ver todo el contenido alojado en el host, siendo esto potencialmente peligroso si un cibercriminal llegase a obtener este acceso.



Vulnerabilidades encontradas:

Control de Acceso Roto (Broken Access Control): Esta vulnerabilidad ocurre cuando las restricciones sobre lo que los usuarios autenticados pueden hacer no son correctamente aplicadas. Los atacantes pueden explotar estas fallas para acceder a funcionalidades y datos no autorizados, como cambiar información de usuarios o acceder a funciones administrativas y escalada de privilegios.

Esta vulnerabilidad encontrada es valorada como crítica en OWASP (Open Web Application Security Project) y la número 1 de su top 10.

Soluciones y Conclusión:

Solución de la Vulnerabilidad:

1- Implementación de Controles de Acceso Adecuados:

- Verificar los permisos del usuario en cada solicitud en el servidor.
- Asegurarse de que los controles de acceso no dependan únicamente de la autenticación, sino también de la autorización adecuada.

2- Validación de la Identidad del Usuario:

- Utilizar tokens de sesión o JWT (JSON Web Tokens) para validar la identidad del usuario en cada solicitud.
- Implementar mecanismos de expiración y renovación de sesiones para mitigar el riesgo de secuestro de sesiones.

3- Protección de URLs y Parámetros:

- Validar y sanitizar todos los parámetros de entrada.
- Evitar exponer identificadores sensibles en las URLs, utilizando identificadores indirectos o UUIDs.

4- Pruebas y Revisiones de Seguridad:

- Realizar auditorías de seguridad regulares y pruebas de penetración para identificar y corregir posibles fallos de control de acceso.
- Utilizar herramientas de análisis de código estático y dinámico para detectar vulnerabilidades de acceso.

5- Principio de Menor Privilegio:

- Aplicar el principio de menor privilegio, asegurándose de que cada usuario tenga solo los permisos mínimos necesarios para realizar sus tareas.
- Revisar y actualizar regularmente las políticas de acceso según sea necesario.

Conclusión:

- La vulnerabilidad de Control de Acceso Roto es crítica y común, permitiendo a atacantes escalar privilegios y acceder a funciones no autorizadas. Implementar controles de acceso robustos, validación de identidad adecuada y realizar auditorías de seguridad regulares son esenciales para mitigar esta vulnerabilidad y proteger los sistemas contra accesos no autorizados.

Anexo:

Herramientas usadas para detectar las vulnerabilidades y explotarlas:

