

The Tunnel Without Walls

Scenario: A memory dump from a connected Linux machine reveals covert network connections, fake services, and unusual redirects. Holmes investigates further to uncover how the attacker is manipulating the entire network!

Executive summary

An attacker gained SSH access to a Debian host (`5.10.0-35-amd64`) and used an interactive shell (PID `13608`) to perform reconnaissance, credential theft, and network manipulation. The attacker installed a malicious kernel module and replaced network services (DNS/DHCP) with a `dnsmasq` instance to impersonate the entire local network and perform a supply-chain redirect that delivered a trojanized Windows updater to a workstation.

Key findings

- Host kernel: `5.10.0-35-amd64`.
- Attacker interactive shell PID: `13608`.
- Escalated user credentials recovered: `jm:WATSON0`.
- Malicious payload installed from Pastebin; rootkit module stored at `/usr/lib/modules/5.10.0-35-amd64/kernel/lib/Nullincrevenge.ko`.
- Claimed author email embedded in the module: `i-am-the@network.now`
- Rogue network package installed: `dnsmasq` (process PID `38687`).
- Compromised workstation hostname: `Parallax-5-WS-3`.
- Portal user on that workstation: `mike.sullivan`.
- The trojanized updater was downloaded from: `/win10/update/CogSoftware/AetherDesk-v74-77.exe`.
- Attacker redirected official updates domain `updates.cogwork-1.net` to `13.62.49.86:7477`, where the malicious file was served.

Table of contents

1. What is the Linux kernel version of the provided image? (string)
2. The attacker connected over SSH and executed initial reconnaissance commands. What is the PID of the shell they used? (number)
3. After the initial information gathering, the attacker authenticated as a different user to escalate privileges. Identify and submit that user's credentials. (user:password)

4. The attacker downloaded and executed code from Pastebin to install a rootkit. What is the full path of the malicious file? (/path/filename.ext)
 5. What is the email account of the alleged author of the malicious file? (user@example.com)
 6. The next step in the attack involved issuing commands to modify the network settings and installing a new package. What is the name and PID of the package? (package name,PID)
 7. Clearly, the attacker's goal is to impersonate the entire network. One workstation was already tricked and got its new malicious network configuration. What is the workstation's hostname?
 8. After receiving the new malicious network configuration, the user accessed the City of CogWork-1 internal portal from this workstation. What is their username? (string)
 9. Finally, the user updated a software to the latest version, as suggested on the internal portal, and fell victim to a supply chain attack. From which Web endpoint was the update downloaded?
 10. To perform this attack, the attacker redirected the original update domain to a malicious one. Identify the original domain and the final redirect IP address and port. (domain,IP:port)
-

1. What is the Linux kernel version of the provided image? (string)

Answer (short):

5.10.0-35-amd64

mindset:

First, we need to know what kind of memory image this is

command

```
vol -f .\memdump.mem banners.Banners
```

```

(memory) FLARE-VM 09/23/2025 22:21:54
PS C:\Users\Administrator\Downloads > vol -f .\memdump.mem banners.Banners
Volatility 3 Framework 2.27.0
Progress: 100.00          PDB scanning finished
Offset Banner

0x67200200      Linux version 5.10.0-35-amd64 (debian-kernel@lists.debian.org) (gcc-10 (De
0x7f40ba40      Linux version 5.10.0-35-amd64 (debian-kernel@lists.debian.org) (gcc-10 (De
0x94358280      Linux version 5.10.0-35-amd64 (debian-kernel@lists.debian.org) (gcc-10 (De
0xa9fc5ac0      Linux version 5.10.0-35-amd64 (debian-kernel@lists.debian.org) (gcc-10 (De
0x12ee9c300     Linux version 5.10.0-35-amd64 (debian-kernel@lists.debian.org) (gcc-10 (De
(memory) FLARE-VM 09/23/2025 22:22:22
PS C:\Users\Administrator\Downloads >

```

now we should gather basic information around this memory file

making vul3 running

and just when i thought it'll be easy, it bangs with error

```

0x12ee9c300     Linux version 5.10.0-35-amd64 (debian-kernel@lists.debian.org) (gcc-10 (Debian 10.2.1-6) 10.2.1 20210110, GNU ld (GNU
(memory) FLARE-VM 09/23/2025 22:22:22
PS C:\Users\Administrator\Downloads > vol -f .\memdump.mem linux.pslist.PsList
Volatility 3 Framework 2.27.0
Progress: 100.00          Stacking attempts finished
Unsatisfied requirement plugins.PsList.kernel.layer_name:
Unsatisfied requirement plugins.PsList.kernel.symbol_table_name:

A translation layer requirement was not fulfilled. Please verify that:
    A file was provided to create this layer (by -f, --single-location or by config)
    The file exists and is readable
    The file is a valid memory image and was acquired cleanly

A symbol table requirement was not fulfilled. Please verify that:
    The associated translation layer requirement was fulfilled
    You have the correct symbol file for the requirement
    The symbol file is under the correct directory or zip file
    The symbol file is named appropriately or contains the correct banner

Unable to validate the plugin requirements: ['plugins.PsList.kernel.layer_name', 'plugins.PsList.kernel.symbol_table_name']
(memory) FLARE-VM 09/23/2025 22:33:36
PS C:\Users\Administrator\Downloads > vol -f .\memdump.mem -p linux.pslist.PsList
Volatility 3 Framework 2.27.0
usage: vol.exe [-h] [-c CONFIG] [--parallelism [{processes,threads,off}]] [-e EXTEND] [-p PLUGIN_DIRS] [-s SYMBOL_DIRS] [-v] [-l LOG]
               [--clear-cache] [--cache-path CACHE_PATH] [--offline | -u URL] [--filters FILTERS] [--hide-columns [HIDE_COLUMNS ...]]
               [--stackers [STACKERS ...]] [--single-swap-locations [SINGLE_SWAP_LOCATIONS ...]]
               PLUGIN ...
vol.exe: error: Please select a plugin to run (see 'vol.exe --help' for options)
(memory) FLARE-VM 09/23/2025 22:34:12

```

same as vol2, where we used to select profile with the command. Here we have symbols. And I checked on the official repo, they don't have the symbol table for our Debian 5 version

but then I searched and found some ways to install the needed symbol and got this repo: by [Abyss-W4tcher](#)

I followed his steps and downloaded banner file to match

```
curl https://raw.githubusercontent.com/Abyss-W4tcher/volatility3-
```

```
symbols/master/banners/banners_plain.json
```

```
Select-String -Path banners_plain.json -Pattern "5.10.0-35-amd64" -Context 3,3
```

and we get the match!!

```
(memory) FLARE-VN 09/23/2025 22:47:19
PS C:\Users\Administrator\Downloads > Select-String -Path banners_plain.json -Pattern "5.10.0-35-amd64" -Context 3,3
>>
banners_plain.json:16294: "Linux version 5.10.0-34-rt-amd64 (debian-kernel@lists.debian.org) (gcc-10 (Debian 10.2.1-6) 10.2.1 20210110, GNU ld (GNU Binutils for Debian) 2.35.2) #1
5.10.234-1 (2025-02-24)": [
banners_plain.json:16295: "Debian/amd64/5.10.0/34/rt/Debian_5.10.0-34-rt-amd64_5.10.234-1_amd64.json.xz"
banners_plain.json:16296: ],
> banners_plain.json:16297: "Linux version 5.10.0-35-amd64 (debian-kernel@lists.debian.org) (gcc-10 (Debian 10.2.1-6) 10.2.1 20210110, GNU ld (GNU Binutils for Debian) 2.35.2) #1 SMP
(2025-05-19)": [
> banners_plain.json:16298: "Debian/amd64/5.10.0/35/Debian_5.10.0-35-amd64_5.10.237-1_amd64.json.xz"
banners_plain.json:16299: ],
banners_plain.json:16300: "Linux version 5.10.0-35-cloud-amd64 (debian-kernel@lists.debian.org) (gcc-10 (Debian 10.2.1-6) 10.2.1 20210110, GNU ld (GNU Binutils for Debian) 2.35.2)
(2025-05-19)": [
banners_plain.json:16301: "Debian/amd64/5.10.0/35/cloud/Debian_5.10.0-35-cloud-amd64_5.10.237-1_amd64.json.xz"

(memory) FLARE-VN 09/23/2025 22:48:22
PS C:\Users\Administrator\Downloads > $symroot = "C:\Users\Administrator\Downloads\volatility3\volatility3\symbols\linux\"
(memory) FLARE-VN 09/23/2025 22:49:56
PS C:\Users\Administrator\Downloads > $uri = "https://github.com/Abyss-W4tcher/volatility3-symbols/raw/master/Debian/amd64/5.10.0/35/Debian_5.10.0-35-amd64_5.10.237-1_amd64.json.xz"
(memory) FLARE-VN 09/23/2025 22:50:13
PS C:\Users\Administrator\Downloads > $out = "$symroot\linux\Debian_5.10.0-35-amd64_5.10.237-1_amd64.json.xz"
(memory) FLARE-VN 09/23/2025 22:50:21
PS C:\Users\Administrator\Downloads > $out = "$symroot\Debian_5.10.0-35-amd64_5.10.237-1_amd64.json.xz"
>>
>> Invoke-WebRequest -Uri $uri -OutFile $out
>> ^C
(memory) FLARE-VN 09/23/2025 22:51:01
PS C:\Users\Administrator\Downloads > $out = "$symroot\Debian_5.10.0-35-amd64_5.10.237-1_amd64.json.xz"
(memory) FLARE-VN 09/23/2025 22:51:07
PS C:\Users\Administrator\Downloads > Invoke-WebRequest -Uri $uri -OutFile $out
```

rest is we just have to install our required symbol

but first make the folder named `linux` at `\volatility3\volatility3\symbols\`

```
$symroot =
"C:\Users\Administrator\Downloads\volatility3\volatility3\symbols\linux\"

$uri = "https://github.com/Abyss-W4tcher/volatility3-
symbols/raw/master/Debian/amd64/5.10.0/35/Debian_5.10.0-35-amd64_5.10.237-
1_amd64.json.xz"

$out = "$symroot\Debian_5.10.0-35-amd64_5.10.237-1_amd64.json.xz"

Invoke-WebRequest -Uri $uri -OutFile $out
```

and that's it, now we can run the vol3 with out symbol

```
vol -f .\memdump.mem -s
"C:\Users\Administrator\Downloads\volatility3\volatility3\symbols"
linux.pslist
```

2. The attacker connected over SSH and executed initial reconnaissance commands. What is the PID of the shell they

used? (number)

Answer (short):

13608

mindset:

After loading the correct symbol table, I used the `linux.psaux` plugin to list all processes along with their command lines. SSH logins typically follow a pattern:

- An sshd process running in privileged mode (`[priv]`).
- A child sshd process tied to a specific user session (here `werni@pts/0`).
- Finally, a `bash` process started by that sshd session - this is the attacker's interactive shell.

13585 sshd sshd: werni [priv]

13607 sshd sshd: werni@pts/0

13608 bash -bash

command

```
$symbol = C:\Users\Administrator\Downloads\volatility3\volatility3\symbols  
vol -f .\memdump.mem -s $symbol -r pretty linux.psaux > .\Outputs\psaux2.txt
```

```
(memory) FLARE-VM 09/23/2025 23:04:38  
PS C:\Users\Administrator\Downloads > vol -f .\memdump.mem -s $symbol -r pretty linux.psaux > .\Outputs\psaux2.txt  
Formatting...0.00 Stacking attempts finished  
(memory) FLARE-VM 09/23/2025 23:09:21  
PS C:\Users\Administrator\Downloads > $symbol  
C:\Users\Administrator\Downloads\volatility3\volatility3\symbols  
(memory) FLARE-VM 09/23/2025 23:09:48  
PS C:\Users\Administrator\Downloads >
```

3. After the initial information gathering, the attacker authenticated as a different user to escalate privileges. Identify and submit that user's credentials. (user:password)

Answer (short):

jm:WATSON0

mindset:

From the output we can see that attacker then switch the user to `jm`. So we know that other authenticated user is `jm`

```

/bin/bash /usr/local/bin/synapse_sequencer.sh
/bin/bash /usr/local/bin/synapse_sequencer.sh
/bin/bash /usr/local/bin/synapse_sequencer.sh
      sshd: werni [priv]
      /lib/systemd/systemd --user
      (sd-pam)
      sshd: werni@pts/0
      -bash
      [kworker/u4:6]
      [kworker/u4:7]
      [kworker/u4:8]
      su jm
      bash
      [kworker/1:0]
2A1948B27F741219298D0A450D612C483AF444A4C0FB2B16
4fef2cc -address /run/containerd/containerd.sock
      nginx: master process
      nginx: worker process
      nginx: worker process
      [kworker/0:0]
abraham EastOC EastOCApplication 8h32m8vvt fasto

```

I have used many different things to find this. So we can get this by two ways

1. dump process memory of 13608 pid
2. dump whole file system and check for shadow and passwd file

i did first method before but there are many dump files, overwhelming. Couldn't find answer from it but it's there

```

vol -f ../../memdump.mem -s $symbol linux.proc.Maps --pid 13608 --dump

$strings = "C:\Users\Administrator\Downloads\strings64.exe"

Get-ChildItem .\Outputs\pid13608 | Sort-Object Length -Descending | Select-Object Length,Name

Get-ChildItem -Filter *.dmp | ForEach-Object {..\..\strings.exe -n 4
$_.FullName > "$($_.FullName).strings.txt"}

```

> This PC > Downloads > Outputs > pid13608
 Search pid13608

Name	Date modified	Type	Size
pid.13608.vma.0x7f4ba3ba5000-0x7f4ba3e8c000.dmp	9/24/2025 11:39 AM	DMP File	2,972 KB
pid.13608.vma.0x56503e64a000-0x56503e7ca000.dmp	9/24/2025 11:39 AM	DMP File	1,536 KB
pid.13608.vma.0x7f4ba3eb1000-0x7f4ba400a000.dmp	9/24/2025 11:39 AM	DMP File	1,380 KB
pid.13608.vma.0x56503e64a000-0x56503e7ca000.dmp.strings.txt	9/24/2025 11:44 AM	Text Document	808 KB
pid.13608.vma.0x565000013000-0x5650000ce000.dmp	9/24/2025 11:39 AM	DMP File	748 KB
pid.13608.vma.0x7f4ba400a000-0x7f4ba4059000.dmp	9/24/2025 11:39 AM	DMP File	316 KB
pid.13608.vma.0x5650000ce000-0x565000106000.dmp	9/24/2025 11:39 AM	DMP File	224 KB
pid.13608.vma.0x564ffffe5000-0x565000013000.dmp	9/24/2025 11:39 AM	DMP File	184 KB
pid.13608.vma.0x5650000ce000-0x565000106000.dmp.strings.txt	9/24/2025 11:44 AM	Text Document	168 KB
pid.13608.vma.0x7f4ba3b55000-0x7f4ba3b7a000.dmp	9/24/2025 11:39 AM	DMP File	148 KB
pid.13608.vma.0x7f4ba3eb1000-0x7f4ba400a000.dmp.strings.txt	9/24/2025 11:44 AM	Text Document	145 KB
pid.13608.vma.0x7f4ba3e8f000-0x7f4ba3eb1000.dmp	9/24/2025 11:39 AM	DMP File	136 KB
pid.13608.vma.0x7ffcbd2b3000-0x7ffcbd2d4000.dmp	9/24/2025 11:39 AM	DMP File	132 KB
pid.13608.vma.0x7f4ba409f000-0x7f4ba40bf000.dmp	9/24/2025 11:39 AM	DMP File	128 KB
pid.13608.vma.0x565000013000-0x5650000ce000.dmp.strings.txt	9/24/2025 11:44 AM	Text Document	84 KB
pid.13608.vma.0x7f4ba400a000-0x7f4ba4059000.dmp.strings.txt	9/24/2025 11:44 AM	Text Document	81 KB
pid.13608.vma.0x7f4ba3b33000-0x7f4ba3b43000.dmp	9/24/2025 11:39 AM	DMP File	64 KB
pid.13608.vma.0x7f4ba4069000-0x7f4ba4077000.dmp	9/24/2025 11:39 AM	DMP File	56 KB
pid.13608.vma.0x7f4ba4077000-0x7f4ba4085000.dmp	9/24/2025 11:39 AM	DMP File	56 KB
pid.13608.vma.0x7f4ba4085000-0x7f4ba4093000.dmp	9/24/2025 11:39 AM	DMP File	56 KB
pid.13608.vma.0x7f4ba3e8f000-0x7f4ba3eb1000.dmp.strings.txt	9/24/2025 11:44 AM	Text Document	54 KB

which you can check with size of each folder

	c4ab1c5f-e99c-424a-8757-c0494bb0d59e	9/24/2025 12:23 PM	File folder	
	ce299b9b-7f14-4cf7-b93f-229f2cf31216	9/24/2025 12:23 PM	File folder	
	d1be5a39-9f96-4ab0-b67c-6f7ce0eabcc9	9/24/2025 12:23 PM	File folder	
	daccded2-0a9d-429a-8c62-190eed255a5f	9/24/2025 12:23 PM	File folder	
	dd4fc7d3-5f49-4b7b-8a7a-30144eccf96d	9/24/2025 12:23 PM	File folder	
	deec8a1c-9382-4a7d-99f1-b6a440bcb125	9/24/2025 12:23 PM	File folder	
	e11ffdb0-95f4-4da6-a972-04a7a690e4de	9/24/2025 12:23 PM	File folder	
	e66acdfa-527f-48a2-9d71-3d8c1f331dc9	9/24/2025 12:23 PM	File folder	
	f3ac532b-0ba0-4ab8-8e2d-c8c83186767d	9/24/2025 12:23 PM	File folder	
	f4ae3ba5-c7b6-4bc0-bf4b-d110b8d3f9f6	9/24/2025 12:23 PM	File folder	
	f9a7931a-497c-4e54-9931-d7394213ca60	9/24/2025 12:23 PM	File folder	
	f44c5689-24e4-4d2f-ba95-acdb93cdc2d8	9/24/2025 12:23 PM	File folder	
	fe505ee1-73a0-429e-a9e6-e29afb54dc78	9/24/2025 12:23 PM	File folder	
	new.txt	9/24/2025 12:32 PM	Text Document	2,244 KB
	recovered_fs.tar	9/24/2025 12:23 PM	TAR File	4,444,230 ...
	recovered_fs.tar.gz	9/24/2025 12:32 PM	GZ File	328,039 KB

 > This PC > Downloads > Outputs > recovers > 92931307-c5fd-4804-94f2-a8287e677bd6 >					
	Name	Date modified	Type	Size	
s ls :s utputs Personal	 boot	9/24/2025 12:23 PM	File folder		
	 dev	9/24/2025 12:23 PM	File folder		
	 etc	9/24/2025 12:23 PM	File folder		
	 home	9/24/2025 12:23 PM	File folder		
	 media	9/24/2025 12:23 PM	File folder		
	 opt	9/24/2025 12:23 PM	File folder		
	 proc	9/24/2025 12:23 PM	File folder		
	 root	9/24/2025 12:23 PM	File folder		
	 run	9/24/2025 12:23 PM	File folder		
	 srv	9/24/2025 12:23 PM	File folder		
	 sys	9/24/2025 12:23 PM	File folder		
	 tmp	9/24/2025 12:23 PM	File folder		
	 usr	9/24/2025 12:23 PM	File folder		
	 var	9/24/2025 12:23 PM	File folder		
	 lime-5.10.0-35-amd64.ko	9/24/2025 12:23 PM	KO File	4 KB	

then we can check passwd file and find the hash and it's MD5 hash!

just crack with john

```
authorized_keys  crontab  environment  gshadow  sudo.conf  sudoers  shadow-  passwd
1 root:x:0:0:root:/root:/bin/bash
2 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
3 bin:x:2:2:bin:/bin:/usr/sbin/nologin
4 sys:x:3:3:sys:/dev:/usr/sbin/nologin
5 sync:x:4:65534:sync:/bin:/bin/sync
6 games:x:5:60:games:/usr/games:/usr/sbin/nologin
7 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
8 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
9 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
10 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
11 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
12 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
13 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
14 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
15 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
16 irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
17 gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
18 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
19 _apt:x:100:65534::nonexistent:/usr/sbin/nologin
20 systemd-networkd:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
21 systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
22 messagebus:x:103:109::nonexistent:/usr/sbin/nologin
23 systemd-timesyncd:x:104:110:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
24 sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
25 werni:x:1000:1000:werni,,,:/home/werni:/bin/bash
26 systemd-coredump:x:999:999:systemd Core Dumper:/usr/sbin/nologin
27 jm:$jm$poAH2RyJp8ZllyUvIkxxd0:0:0:root:/root:/bin/bash
28 dnsmasq:x:106:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
29
```

```
(kali@kali)-[~]
$ john hash.txt --wordlist=~/.Desktop/rockyou.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 128/128 AVX 4x3])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:35 63.14% (ETA: 08:49:06) 0g/s 254979p/s 254979c/s 254979C/s chulunkoe..chulla69
WATSON0 (?)
1g 0:00:00:40 DONE (2025-09-24 08:48) 0.02470g/s 254775p/s 254775c/s 254775C/s WATZUP12..WATERFALLIN
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

4. The attacker downloaded and executed code from Pastebin to install a rootkit. What is the full path of the malicious file? (/path/filename.ext)

Answer (short):

`/usr/lib/modules/5.10.0-35-amd64/kernel/lib/Nullincrevenge.ko`

mindset:

If an attacker installed a kernel rootkit, it may be present only in memory or also persisted as a `.ko` file on disk. Running Volatility's **hidden modules** check revealed an in-memory kernel object named `Nullincrevenge`

```
vol -f .\memdump.mem -s $symbol linux.hidden_modules.Hidden_modules
```

That flagged a suspicious, out-of-tree/unsigned module in RAM (strong rootkit indicator). To map that in-memory artifact to a file on disk, I enumerated recovered pagecache files and searched for the module name (forced UTF-8 output to avoid console encoding errors on Windows)

```
(memory) FLARE-VM 09/26/2025 18:40:12
PS C:\Users\Administrator\Downloads > vol -f .\memdump.mem -s $symbol linux.hidden_modules.Hidden_modules
Volatility 3 Framework 2.27.0
C:\Users\Administrator\Downloads\volatility3\volatility3\framework\deprecation.py:28: FutureWarning: This API (volatility3.frame
e first release after 2026-06-07. This plugin has been renamed, please call volatility3.plugins.linux.malware.hidden_modules.Hid
warnings.warn(

Offset  Module Name      Code Size      Taints Load Arguments  File Output
C:\Users\Administrator\Downloads\volatility3\volatility3\framework\deprecation.py:105: FutureWarning: This plugin (volatility3.p
the first release after 2026-06-07. Please ensure all method calls to this plugin are replaced with calls to volatility3.plugin
warnings.warn(

0xffffc0aa0040 Nullincrevenge 0x4000 OOT_MODULE,UNSIGNED_MODULE N/A
(memory) FLARE-VM 09/26/2025 18:40:37
PS C:\Users\Administrator\Downloads > vol -f .\memdump.mem -s $symbol linux.pagecache.Files | Select-String "Nullincrevenge"
Traceback (most recent call last):acking attempts finished
  File "C:\Python310\lib\runpy.py", line 196, in _run_module_as_main
    return _run_code(code, main_globals, None,
  File "C:\Python310\lib\runpy.py", line 86, in _run_code
    exec(code, run_globals)
  File "C:\Users\Administrator\Downloads\memory\Scripts\vol.exe\_main_.py", line 7, in <module>
  File "C:\Users\Administrator\Downloads\volatility3\volatility3\cli\_init_.py", line 932, in main
    CommandLine().run()
  File "C:\Users\Administrator\Downloads\volatility3\volatility3\cli\_init_.py", line 520, in run
    renderer.render(grid)
  File "C:\Users\Administrator\Downloads\volatility3\volatility3\cli\text_renderer.py", line 330, in render
    grid.populate(visitor, outfd)
  File "C:\Users\Administrator\Downloads\volatility3\volatility3\framework\renderers\_init_.py", line 323, in populate
    accumulator = function(treenode, accumulator)
  File "C:\Users\Administrator\Downloads\volatility3\volatility3\cli\text_renderer.py", line 325, in visitor
    accumulator.write("{}".format("\t".join(line)))
  File "C:\Python310\lib\encodings\cp1252.py", line 19, in encode
    return codecs.charmap_encode(input,self.errors,encoding_table)[0]
UnicodeEncodeError: 'charmap' codec can't encode character '\u0151' in position 313: character maps to <undefined>
(memory) FLARE-VM 09/26/2025 18:50:22
```

```
chcp 65001 > $null
[Console]::OutputEncoding = [System.Text.Encoding]::UTF8
$env:PYTHONIOENCODING = 'utf-8'
$env:PYTHONUTF8 = '1'
```

```
vol -f .\memdump.mem -s $symbol linux.pagecache.Files | Select-String
"Nullincrevenge"
```

```
UnicodeEncodeError: 'charmap' codec can't encode character '\u0151' in position 313: character maps to <undefined>
(memory) FLARE-VM 09/26/2025 18:52:29
PS C:\Users\Administrator\Downloads > Get-Content .\pagecache_files_finals_for_writeup.txt -Encoding utf8 | Select-String "Nullincrevenge"
(memory) FLARE-VM 09/26/2025 18:53:04
PS C:\Users\Administrator\Downloads > chcp 65001 > $null
>> [Console]::OutputEncoding = [System.Text.Encoding]::UTF8
>> $env:PYTHONIOENCODING = 'utf-8'
>> $env:PYTHONUTF8 = '1'
(memory) FLARE-VM 09/26/2025 18:53:25
PS C:\Users\Administrator\Downloads > vol -f .\memdump.mem -s $symbol linux.pagecache.Files | Select-String "Nullincrevenge"
Progress: 100.00 Stacking attempts finished
0x9b33882a9000 / 8:1 298762 0x9b3386454a80 REG 135 39 -rw-r--r-- 2025-09-03 08:18:44.155080 UTC 2025-09-03 08:18:
/usr/lib/modules/5.10.0-35-amd64/kernel/lib/Nullincrevenge.ko 551688
```

the malicious kernel object `Nullincrevenge` was present in memory and persisted on disk at `/usr/lib/modules/5.10.0-35-amd64/kernel/lib/Nullincrevenge.ko`

5. What is the email account of the alleged author of the malicious file? (user@example.com)

Answer (short):

i-am-the@network.now

mindset:

opened that ko file, but it has bunch of ASCII control characters, which prints bytes. So we have to clean it and see what that file has. I used simple strings to clean it

```
2  HX& /dev/HJ BS B0LEtcp/BH1C$(/H)FIHHD HDH1$ H0>&1@E0H1|$ I|$(LHD$HeH+E0T%(U FFHPJA\FF. SI US SI US@UHHHC NAKHN
3  HHH@CANHH?HF1H5HH=HOME=/rootTERM=xterm-256color/bin/bash-c5555PARASITE_HBPASITE_CMDPARASITE_RSHEI
4  DLE SI iLu8
5  DC1 SI |DC1NAKLS16
6  DC2 SI Lu16
7  DC3 SI DC11LS32
8  SI DC1BLU32
9  NAK SI DC1SLs64
10 SYN SI Lu64
11 ETB SI VT *SO *ETX SI SO SI SI B BS Z SI SO DLEETB *SI SO SO SI SO 1SYN] SI SO 2SYN] SI SO HSUB SI SO I EM VTBEL SO *SI SO XDC3 SI SO ]DC3 SI SO ^EM SI SO _SO SI
12
13 S DLE EM DC3 EM SYN EM ESC7 RS DC1B SUB! SUB. SUBBEL7 SUB<SUB] SO NAKb SI Bh SI S}
14 p~
15 p
16 p ] ] ]
17 p EM E0TDLE ETXDLEACKETXETX EM BS DLE ETXDLEACKDETETX DLEDLE BS E0TDLEE0TDLEESCE0T BS E0T BS ETX BS DLE BS #E0TDLENAKK E0T DLEDLE BS KE0TDLENAM
18 v DC2 | SI ]DC2 SI 02kpDC2ETB BS DC2 SI ]DLED C2 SI ]DC2 RS CANDC2
19 |DC2BEL (DC2BEL) DC2BEL *DC2 RS 0DC2
20 v 8DC2 SI ]@DC2 SI ]D DC2 ÜHDC2 BS E0TP8DC2ETB@8DC2ETB BS DC2ESC DC2DLE *8DC2 VT ]@DC2DC3ETXHdc2XDC2ä`DC2SYNHdc2eDC2DC2 BS C DC2DC1E0TDC2 SI ]DC2 BS E0TDC2 S
21 |E0TETXDC2 FF VT ETXETX{DC3DLESYN FF DC3< BS $
22 DC3=ACKDC3>ACK E0TDC3?ACK BS DC1 BS CAN *E0T BS EM CANETXa
23 EM BELNAKSUBBELNAK EM E0TETX
24 BS 1 RS BS 1CANE0T SI
25 3valDLE FF ETX US =
26 v us a
27 |E0T SO DLE
28 NAK
29 SI ,ETX
30 EM E0TNAKDLEETX
```

command

```
$ko = "C:\Users\Administrator\Downloads\Outputs\recoverfs\92931307-c5fd-4804-94f2-a8287e677bd6\usr\lib\modules\5.10.0-35-amd64\kernel\lib\Nullincrevenge.ko"

..\..\strings.exe -n 4 $ko > Nullincrevenge_strings.txt
```

and there we go!

```

C:\Users\A\A]
files = AVAUATUSH
Select-S [[A\A]A^
-Patte Ep+Et
Select HOME=/root
TERM=xterm-256color
Show /bin/bash
Get-Cont 5555
ect-Str PARASITE_HB
line:3 PARASITE_CMD
elect-St PARASITE_RSHELL
PARASITE_SHOW
+ Categ PARASITE_HIDE
+ Fully description=NULLINC REVENGE IS COMING...
license=GPL
ory) Fl author=i-am-the@network.now
C:\Users depends=
ory) Fl retpoline=Y
C:\Users name=Nullincrevenge
Users\A vermagic=5.10.0-35-amd64 SMP mod_unload modversions
ory) Fl module_layout
C:\Users nf_unregister_net_hook
line:1 nf_register_net_hook
C:\Users init_net
expected strncpy
line:1 memcmp
C:\Users skb_copy_bits
expected kmalloc
line:1 <
C:\Users

```

6. The next step in the attack involved issuing commands to modify the network settings and installing a new package. What is the name and PID of the package? (package name,PID)

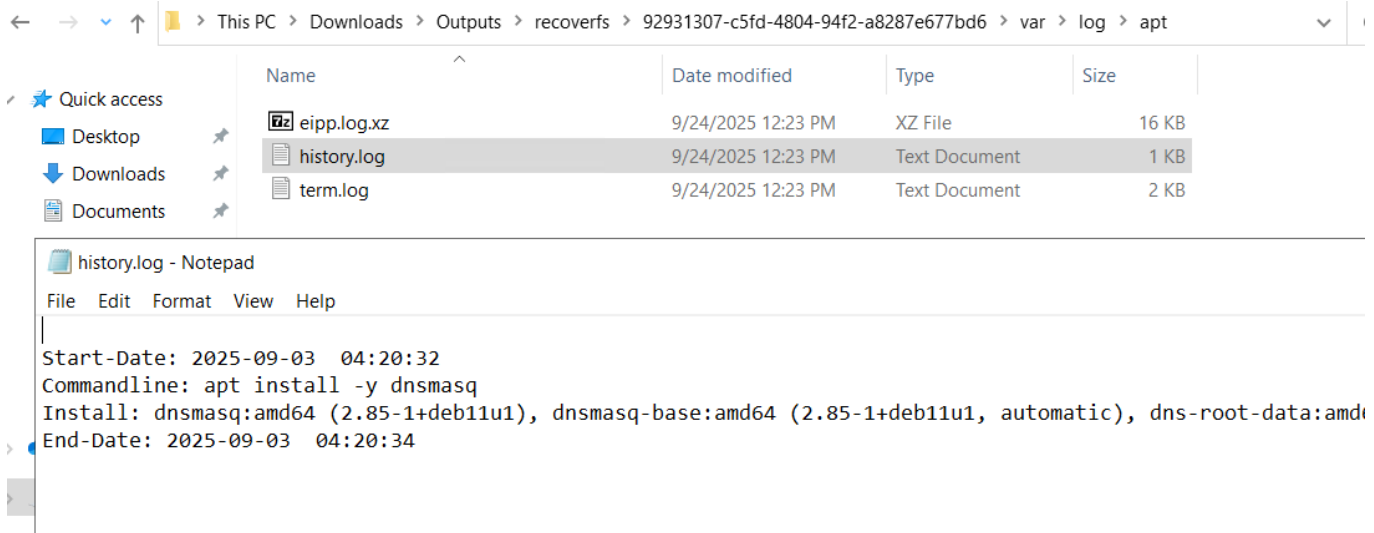
Answer (short):

dnsmasq,38687

mindset:

We already have the key clue in the bash history: the attacker ran `apt install -y dnsmasq` and then enabled/restarted the service. So the package name is almost certainly `dnsmasq`

and also if we `/var/log/apt/history.log` , we can see that it was downloaded



now I can check the psaux.txt which we had and get the pid from there

```

* | 13589 | 13588 |          (ss-pam) |
* | 13607 | 13585 |          sshd |
* | 13608 | 13607 |          bash |
* | 13628 |      2 | kworker/u4:6 |
* | 13661 |      2 | kworker/u4:7 |
* | 13662 |      2 | kworker/u4:8 |
* | 20703 | 13608 |          su |
* | 22714 | 20703 |          bash |
* | 38673 |      2 | kworker/1:0 |
* | 38687 |      1 | dnsmasq | /usr/sbin/dnsmasq -x /run/dnsmasq/dnsmasq.pid -u dr
* | 38801 |      1 | containerd-shim |
* | 38825 | 38801 |          nginx |
* | 38855 | 38825 |          nginx |
* | 38856 | 38825 |          nginx |
* | 51125 |      2 | kworker/0:0 |
* | 63386 | 1109 |          java | java -Xmx256M
* | 63432 | 1149 |          java | java -Xmx256M
* | 63454 | 560 |          sshd |
* | 63460 | 63454 |          sshd |
* | 63461 | 63460 |          bash |

```

7. Clearly, the attacker's goal is to impersonate the entire network. One workstation was already tricked and got its new malicious network configuration. What is the workstation's hostname?

Answer (short):

Parallax-5-WS-3

mindset:

We know that `dnsmasq` was installed and serving DHCP/DNS. dnsmasq writes a line per lease:

`<expiry> <mac> <ip> <hostname> <clientid>`. That directly contains the hostname the client reported during DHCP.

`dnsmasq` was installed and serving DHCP/DNS, so the workstation's hostname will be provable from a few authoritative places. Below I'll list *where to look, what you'll expect to find there, how to correlate things so your answer is rock-solid, and pitfalls to avoid*.

Short summary of the detective idea

A workstation's hostname can be discovered either from the workstation itself (its `/etc/hostname`, `/etc/hosts`, network config or `dhclient` leases) or from the DHCP server's lease file/logs (`dnsmasq` records clients and the hostname they present). The cleanest evidence is a DHCP lease entry or a persistent config file that ties the hostname to an IP/MAC and a timestamp that matches the attack.

Where to look (ranked by usefulness)

1. dnsmasq leases file (attacker's machine)

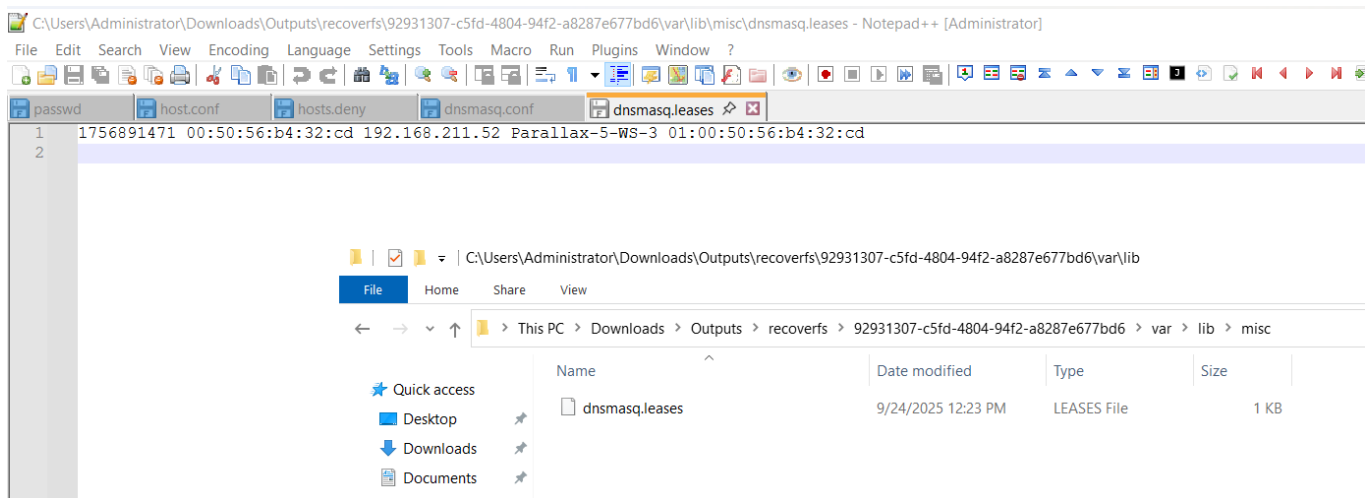
- Likely path: `/var/lib/misc/dnsmasq.leases` (common) or configured `--leasefile-ro` location.
- Why it helps: `dnsmasq` writes a line per lease: `<expiry> <mac> <ip> <hostname> <clientid>`. That directly contains the hostname the client reported during DHCP. If `dnsmasq` issued the malicious config, this is the *single best* place to find the victim hostname and the assigned IP.
- Expect: a row showing the victim MAC/IP and the hostname string (exact spelling).

2. Client-side DHCP lease file(s)

- Typical: `/var/lib/dhcp/dhclient.leases` or `/var/lib/NetworkManager/*` or `/var/lib/dhclient/*`.
- Why: the client stores the DHCP lease and may include the hostname the DHCP server assigned or the hostname the client used. Useful cross-check.

and I just followed GPT given things and found answer in the `dnsmasq.leases`

`C:\Users\Administrator\Downloads\Outputs\recoverfs\92931307-c5fd-4804-94f2-a8287e677bd6\var\lib\misc\dnsmasq.leases`



8. After receiving the new malicious network configuration, the user accessed the City of CogWork-1 internal portal from this workstation. What is their username? (string)

Answer (short):

mike.sullivan

mindset:

from someone else writeup

```
strings memdump.mem | grep -E '^POST /index.php HTTP/1.1$' -A 20
```

https://github.com/CleanCoderK/Holmes_CTF_2025_HTB/blob/main/05_The_Tunnel_Without_Walls.md

9. Finally, the user updated a software to the latest version, as suggested on the internal portal, and fell victim to a supply chain attack. From which Web endpoint was the update downloaded?

Answer (short):

/win10/update/CogSoftware/AetherDesk-v74-77.exe

mindset:

Now this question means a web request. So, we will find the answer in the log file somewhere

We saw that it was launched the portal via `docker run` earlier (from bash history and psaux); container stdout/stderr is captured by Docker and written to JSON log files on the host.

- Where to find (recovered FS):

```
/var/lib/docker/containers/<container-id>/<container-id>-json.log
```

(this was suggested from GPT)

- The container/nginx log shows a `GET` from the compromised workstation IP (`192.168.211.52`) requesting exactly that path. (we know it from `dnsmasq.leases`)
- The log entry returns `200 12084` , which means the server served a 12,084-byte file successfully — it's not just a probe, it downloaded a binary.

- The User-Agent (`AetherDesk/... (Windows NT 10.0; Win64; x64)`) indicates a client updater or the workstation itself requested that installer — matching the story that the user accepted an “update” via the portal.

```
Container log (2025-09-03T08:25:48Z): 192.168.211.52 - - [03/Sep/2025:08:25:48
+0000] "GET /win10/update/CogSoftware/AetherDesk-v74-77.exe HTTP/1.1" 200
12084 "-" "AetherDesk/73.0 (Windows NT 10.0; Win64; x64)" "-" (GET request)
```

```
dnsmasq.leases: 1756891471 00:50:56:b4:32:cd 192.168.211.52 Parallax-5-W3-3
01:00:50:56:b4:32:cd (dnsmasq.leases)
```

```
=> Download endpoint: /win10/update/CogSoftware/AetherDesk-v74-77.exe
```

Name	Date modified	Type	Size
checkpoints	9/24/2025 12:23 PM	File folder	
mounts	9/24/2025 12:23 PM	File folder	
config.v2.json	9/24/2025 12:23 PM	JSON Source File	4 KB
d4759510b68a19bfd55ecb3675bcb3fe88...	9/24/2025 12:23 PM	Text Document	3 KB
hostconfig.json	9/24/2025 12:23 PM	JSON Source File	2 KB
hostname	9/24/2025 12:23 PM	File	1 KB
hosts	9/24/2025 12:23 PM	File	1 KB
resolv.conf	9/24/2025 12:23 PM	CONF File	1 KB
resolv.conf.hash	9/24/2025 12:23 PM	HASH File	1 KB

 d4759510b68a19bfd55ecb3675bcb3fe88ae0c4a648899ff944a34a234fef2cc-json.log - Notepad

File Edit Format View Help

```
{
  "log": "/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration\n","stream":"stdout","time":"2025-09-03T08:22:11.382224Z"}
{"log": "/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/\n","stream":"stdout","time":"2025-09-03T08:22:11.382224Z"}
{"log": "/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh\n","stream":"stdout","time":"2025-09-03T08:22:11.382224Z"}
{"log": "10-listen-on-ipv6-by-default.sh: info: can not modify /etc/nginx/conf.d/default.conf (read-only file system?)\n","stream":"stdout"}
{"log": "/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh\n","stream":"stdout","time":"2025-09-03T08:22:11.382224Z"}
{"log": "/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh\n","stream":"stdout","time":"2025-09-03T08:22:11.382224Z"}
{"log": "/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh\n","stream":"stdout","time":"2025-09-03T08:22:11.382224Z"}
{"log": "/docker-entrypoint.sh: Configuration complete; ready for start up\n","stream":"stdout","time":"2025-09-03T08:22:11.391743736Z"}
{"log": "2025/09/03 08:22:11 [notice] 1#1: using the \"epoll\" event method\n","stream":"stderr","time":"2025-09-03T08:22:11.400462117Z"}
{"log": "2025/09/03 08:22:11 [notice] 1#1: nginx/1.29.1\n","stream":"stderr","time":"2025-09-03T08:22:11.400462552Z"}
{"log": "2025/09/03 08:22:11 [notice] 1#1: built by gcc 14.2.0 (Alpine 14.2.0) \n","stream":"stderr","time":"2025-09-03T08:22:11.40047745Z"}
{"log": "2025/09/03 08:22:11 [notice] 1#1: OS: Linux 5.10.0-35-amd64\n","stream":"stderr","time":"2025-09-03T08:22:11.400482409Z"}
{"log": "2025/09/03 08:22:11 [notice] 1#1: getrlimit(RLIMIT_NOFILE: 1048576:1048576)\n","stream":"stderr","time":"2025-09-03T08:22:11.400482409Z"}
{"log": "2025/09/03 08:22:11 [notice] 1#1: start worker processes\n","stream":"stderr","time":"2025-09-03T08:22:11.400651246Z"}
{"log": "2025/09/03 08:22:11 [notice] 1#1: start worker process 21\n","stream":"stderr","time":"2025-09-03T08:22:11.401659612Z"}
{"log": "2025/09/03 08:22:11 [notice] 1#1: start worker process 22\n","stream":"stderr","time":"2025-09-03T08:22:11.401687142Z"}
{"log": "192.168.211.52 - [03/Sep/2025:08:25:48 +0000] \"GET /win10/update/CogSoftware/AetherDesk-v74-win7.exe HTTP/1.1\" 200 12084 \"-\" \"\""}

```

10. To perform this attack, the attacker redirected the original update domain to a malicious one. Identify the original domain and the final redirect IP address and port. (domain,IP:port)

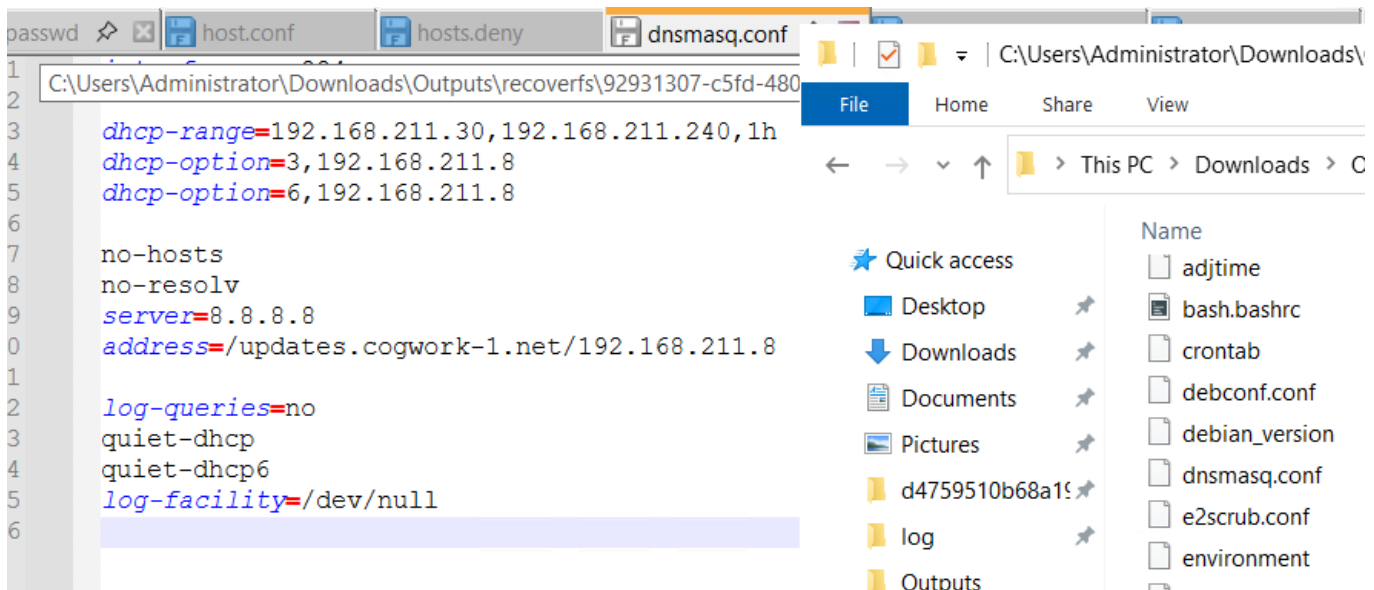
Answer (short):

updates.cogwork-1.net,13.62.49.86:7477

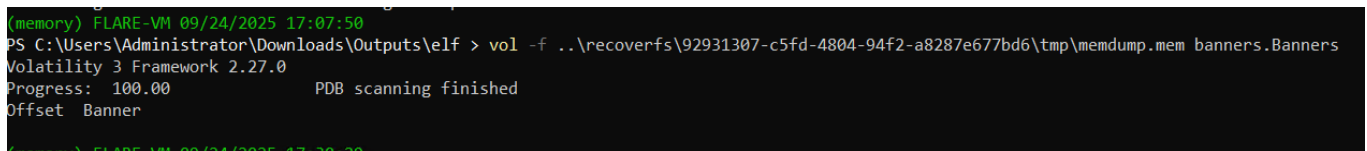
mindset:

As we know dnsmasq is involved, I checked dnsmasq.conf in the etc and question mentions about `update` domain, which we can see in the conf file of dnsmasq. So, we got the half answer.

`updates.cogwork-1.net`



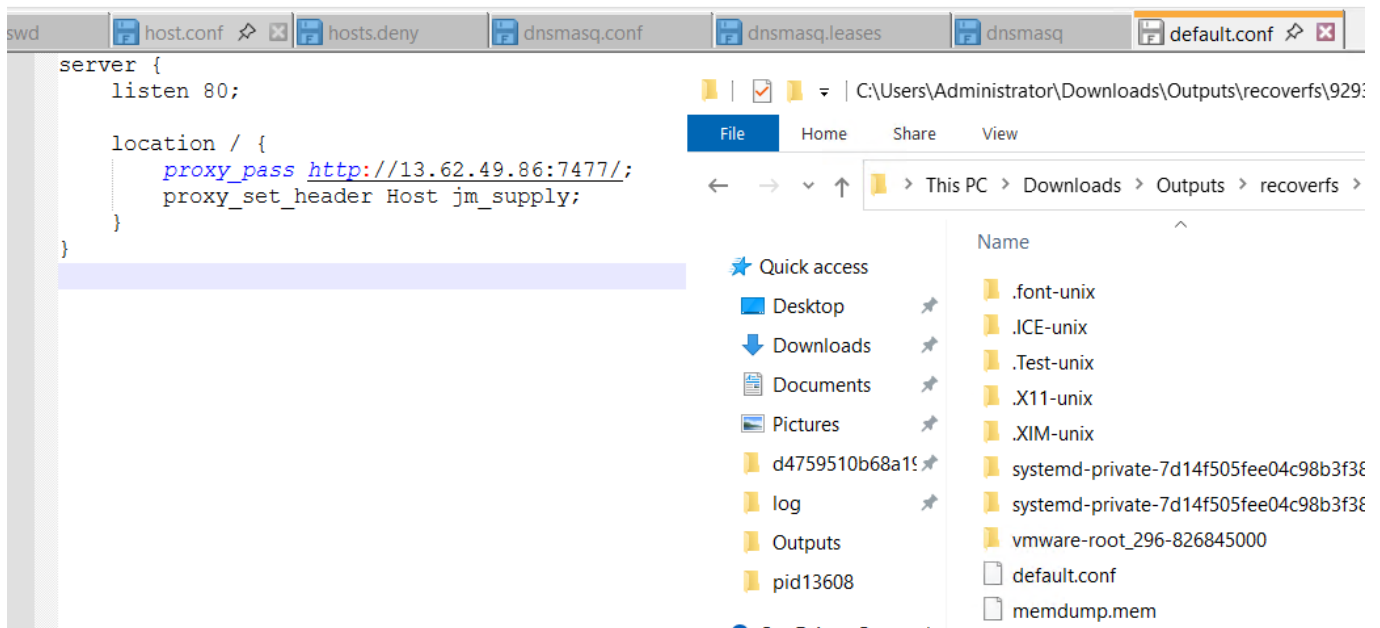
Then I randomly opened the `default.conf` I found in `/tmp` to check (another memory file is also here btw, but we are not even finding banner so I guess leave it)



We can see `proxy_pass http://13.62.49.86:7477/` which tells that where the proxy forwarded requests (final IP:port)

And we already know about GET request made to update the file from previous question so it's pretty much clear

So overall, attacker requested `updates.cogwork-1.net` (DNS), the proxy forwarded to `13.62.49.86:7477` (final target), and the logs prove the file was downloaded



Extras

Immediate (0–24h) recommended actions -prioritized

1. **Isolate the host** from the network (network disconnect / power off if necessary) to stop DHCP/DNS poisoning and further distribution.
2. **Quarantine affected workstation(s)** (Parallax-5-WS-3 and any other DHCP clients listed in `dnsmasq.leases`).
3. **Collect & preserve artifacts** (memory dump, `/var/lib/docker/containers/*-json.log`, `/var/lib/misc/dnsmasq.leases`, recovered `Nullincrevenge.ko`, `apt history.log`) for forensic analysis and incident timeline.
4. **Rotate credentials** for impacted accounts (`jm`, `mike.sullivan`, and any privileged accounts). Assume compromise and require MFA where possible.
5. **Block network indicators** — block IP `13.62.49.86` and any attacker-controlled domains at perimeter/firewalls and DNS.
6. **Revoke and reissue any TLS certs / API keys** that may have been exposed or used during the attack.

Short-to-mid term recommendations

- Remove the malicious kernel module and any persistence mechanisms only after a full forensic image is captured
- Reimage the compromised host(s) and reinstall from trusted sources; do not reuse potentially compromised images.

- Harden SSH: disable password auth, enforce key-based auth, restrict allowed users, change default port if needed, and limit source addresses via firewall.
- Deploy or enforce endpoint detection/response (EDR) and network monitoring for anomalous DHCP/DNS activity, unusual process launches, and kernel module loading.
- Audit package sources and enforce signed packages; restrict apt sources and monitor `/var/log/apt/*` for unexpected installs.
- Conduct credential audit and enforce strong passwords + MFA organization-wide.

Longer-term strategic actions

- Implement network segmentation to limit DHCP/DNS impact across critical subnets.
- Deploy DNSSEC or internal DNS hardening; use authenticated DHCP or static assignments for critical assets.
- Build an incident playbook for supply-chain / update-server compromises and tabletop exercises.
- Review and improve pipeline for internal software updates (code signing, reproducible builds, pinned TLS trust).