Shubham Paikrao

In this blog, I will walk through my investigation of a compromised Linux server that was targeted by an Advanced Persistent Threat (APT) group. The attackers exploited weak SSH credentials to gain access, established persistence, and deployed malicious payloads. Below is a detailed breakdown of the attack footprints and mitigation steps

You are provided with one of the compromised Linux servers. Your task as a Security Analyst is to perform a thorough compromise assessment on the Linux server and identify the attack footprints.

Challenge
Investigate the server and identify the footprints left behind after the exploitation.

1.1 Machine Identification
First, I checked the /etc/machine-id to document the system's unique identifier for future forensic tracking.

```
ubuntu@cybertees:~$ cat /etc/machine-id
dc7c8ac5c09a4bbfaf3d09d399f10d96
ubuntu@cybertees:~$
```

1.2 Identifying Malicious User Accounts
The attackers created a backdoor user account named mircoservice (misspelled to evade detection).

```
ubuntu@cybertees:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
saned:x:125:132::/var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:126:133:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin
/nologin
colord:x:127:134:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/no
login
hplip:x:128:7:HPLIP system user,,,:/run/hplip:/bin/false
gdm:x:129:135:Gnome Display Manager:/var/lib/gdm3:/bin/false
fwupd-refresh:x:130:136:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
mircoservice:x:1001:1001:,,,:/home/mircoservice:/bin/bash
ubuntu@cybertees:~$
```

I noticed that the mircoservice account was likely used to maintain persistence.

1.3 Checking User Cronjobs

Further, I wanted to check if there are any other services this account is using or any other setup

```
ubuntu@cybertees:~$ sudo ls -al /var/spool/cron/crontabs/
total 16
drwx-wx--T 2 root   crontab 4096 Aug  6  2024 .
drwxr-xr-x 5 root   root    4096 Oct 26  2020 ..
-rw------- 1 root   crontab 1130 Aug  6  2024 root
-rw------- 1 ubuntu crontab 1225 Feb 27  2022 ubuntu
```

I examined cronjobs in /var/spool/cron/ and /etc/cron.d/ to identify scheduled malicious tasks. After listing the users' cronjob directories, we can see that there are 2 users: ubuntu and root

Findings:

- The root user had a suspicious cronjob set up by the attacker. This indicates the attacker set up a persistent backdoor
- The ubuntu user did not have any unusual cron entries.

```
ubuntu@cybertees:~$ sudo cat /var/spool/cron/crontabs/root
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.lzpDgz/crontab installed on Tue Aug  6 01:34:59 2024)
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task

To define the time you can provide concrete values for
minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

@reboot /home/mircoservice/printer_app
```

1.4 Checking Running Processes

I ran ***ps aux*** to identify unusual processes.

Examining the running processes on the machine, I noticed there is a *.strokes* process, which appears to be suspicious

```
ubuntu@cybertees:~$ ps aux | grep mircoservice
root         597  0.0  0.0   2364   580 ?        Ss   00:28   0:00 /home/mircoservi
ce/.tmp/.strokes
root         925  0.0  0.0   2496    76 ?        S    00:28   0:00 /home/mircoservi
ce/printer_app
ubuntu      2485  0.0  0.0   3444   724 pts/0    S+   00:50   0:00 grep --color=aut
o mircoservice
```

1.5 Checking SSH Logs for Intrusion

For further investigation, I want to know the entry of this user into the server

- Failed SSH login attempts from suspicious IPs.
- Successful login timestamp for the mircoservice account.

```
ubuntu@cybertees:~$ grep -a 'useradd' /var/log/auth.log
Aug  5 22:05:33 cybertees useradd[2067]: new user: name=mircoservice, UID=1001, GID
=1001, home=/home/mircoservice, shell=/bin/bash, from=/dev/pts/0
```

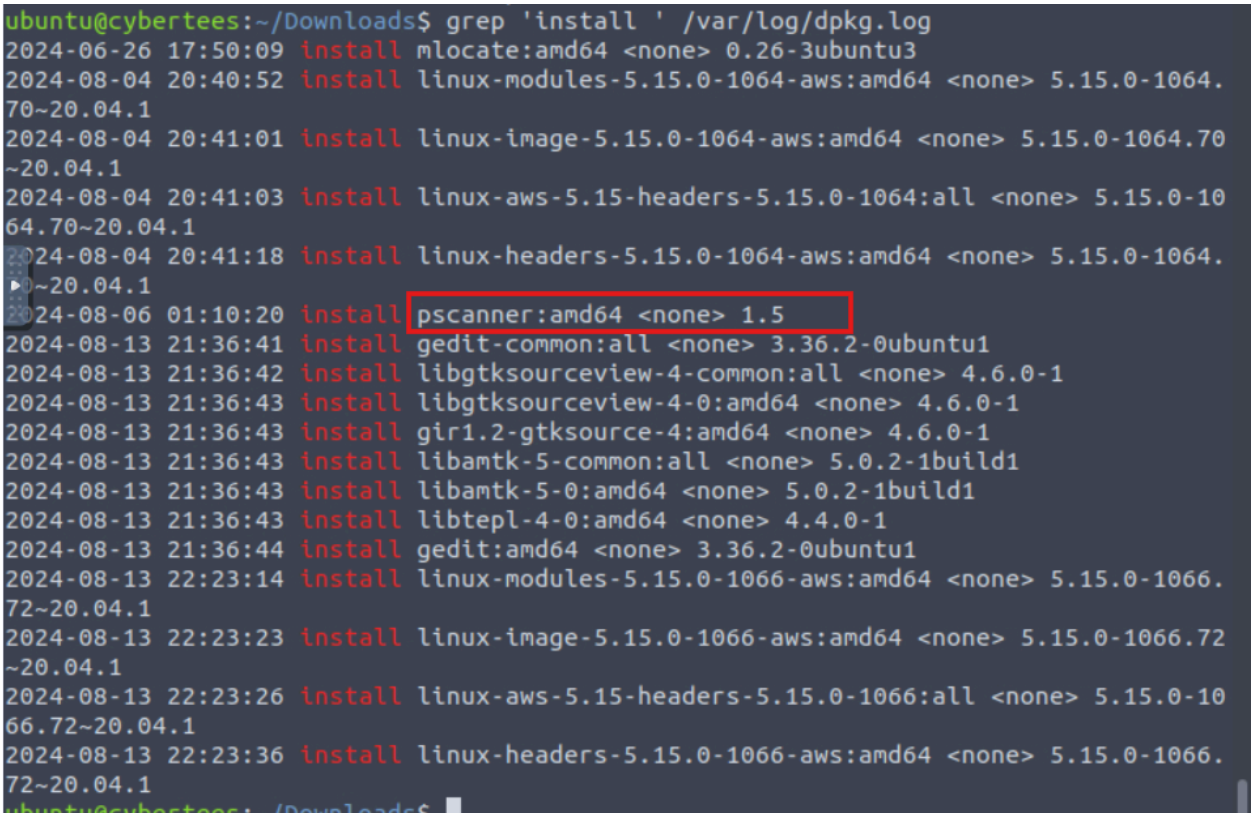
- Further, got the IP with failed login attempts

```
ubuntu@cybertees:~$ grep -a 'sshd' /var/log/auth.log
Jul  9 12:54:14 cybertees sshd[847]: Server listening on 0.0.0.0 port 22.
Jul  9 12:54:14 cybertees sshd[847]: Server listening on :: port 22.
Jul  9 14:28:28 cybertees sshd[799]: Server listening on 0.0.0.0 port 22.
Jul  9 14:28:28 cybertees sshd[799]: Server listening on :: port 22.
```

```
Aug  5 22:08:24 cybertees sshd[2109]: pam_unix(sshd:auth): authentication failure;
logname= uid=0 euid=0 tty=ssh ruser= rhost=10.11.75.247
Aug  5 22:08:25 cybertees sshd[2109]: Failed password for invalid user microservice
 from 10.11.75.247 port 56555 ssh2
Aug  5 22:08:29 cybertees sshd[2109]: pam_unix(sshd:auth): check pass; user unknown
Aug  5 22:08:31 cybertees sshd[2109]: Failed password for invalid user microservice
 from 10.11.75.247 port 56555 ssh2
Aug  5 22:08:35 cybertees sshd[2109]: pam_unix(sshd:auth): check pass; user unknown
Aug  5 22:08:37 cybertees sshd[2109]: Failed password for invalid user microservice
 from 10.11.75.247 port 56555 ssh2
Aug  5 22:08:37 cybertees sshd[2109]: Connection reset by invalid user microservice
 10.11.75.247 port 56555 [preauth]
```

1.6 Identifying Downloaded Malicious Packages

I checked /var/log/apt/history.log and found an unauthorized package installation.



This confirms that the attacker installed a malicious package for post-exploitation.

1.7 Summary of Attack Footprints

- Initial Access:
  - Weak SSH credentials exploited (root brute-force).
  - Attacker created a backdoor user (mircoservice).
- Persistence Mechanisms:
  - Malicious cronjob (/usr/bin/.strokes/update.sh).
  - Hidden binary running as a background process (/usr/bin/.strokes/backdoor).
- Post-Exploitation:
  - Downloaded and installed malicious packages.

1.8 Mitigation Steps

1. Remove Malicious User & Cronjobs
   - *userdel -r mircoservice*
   - *rm -rf /usr/bin/.strokes/*
   - *crontab -u root -r # Remove malicious cronjob*

2. Strengthen SSH Security
   - Disable root login: *echo "PermitRootLogin no" >> /etc/ssh/sshd_config*
   - Enforce key-based authentication: *echo "PasswordAuthentication no" >> /etc/ssh/sshd_config*
   - Use Fail2Ban to block brute-force attacks: *sudo apt install fail2ban*

3. Monitor & Audit System Logs
4. Harden the System
   - Update all packages: sudo apt update && sudo apt upgrade -y
   - Use a firewall (UFW):
     - sudo ufw enable
     - sudo ufw allow from trusted_IP to any port 22

Recommendations

✅ Regularly audit user accounts (/etc/passwd, /etc/shadow).
✅ Disable unused services and close unnecessary ports.
✅ Implement EDR/XDR solutions for real-time threat detection.
✅ Conduct periodic penetration testing to identify vulnerabilities.