

Hack the Box Brutus Writeup by Melisa Nyamukondiwa

Scenario

Sherlock Scenario

In this very easy Sherlock, you will familiarize yourself with Unix auth.log and wtmp logs. We'll explore a scenario where a Confluence server was brute-forced via its SSH service. After gaining access to the server, the attacker performed additional activities, which we can track using auth.log. Although auth.log is primarily used for brute-force analysis, we will delve into the full potential of this artifact in our investigation, including aspects of privilege escalation, persistence, and even some visibility into command execution.

Question 1

Analyze the auth.log. What is the IP address used by the attacker to carry out a brute force attack?

In order to answer this question, the auth.log file was analyzed. Upon scanning, it was revealed that the attacker had attempted to log in as admin multiple times but failed. These multiple failed attempts show that it was a brute force attack.

```
lar 6 06:31:31 ip-172-31-35-28 sshd : Invalid user admin from [REDACTED]
lar 6 06:31:31 ip-172-31-35-28 sshd : Received disconnect from [REDACTED] : Bye Bye [preauth]
lar 6 06:31:31 ip-172-31-35-28 sshd : Disconnected from invalid user admin [REDACTED] [preauth]
lar 6 06:31:31 ip-172-31-35-28 sshd : error: beginning MaxStartups throttling
lar 6 06:31:31 ip-172-31-35-28 sshd : drop connection #10 from [REDACTED] on [172.31.35.28]:22 past
lar 6 06:31:31 ip-172-31-35-28 sshd : Invalid user admin from [REDACTED]
lar 6 06:31:31 ip-172-31-35-28 sshd : pam_unix(sshd:auth): check pass; user unknown
lar 6 06:31:31 ip-172-31-35-28 sshd : pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 t
lar 6 06:31:31 ip-172-31-35-28 sshd : Invalid user admin from [REDACTED]
lar 6 06:31:31 ip-172-31-35-28 sshd : Invalid user admin from [REDACTED]
lar 6 06:31:31 ip-172-31-35-28 sshd : pam_unix(sshd:auth): check pass; user unknown
lar 6 06:31:31 ip-172-31-35-28 sshd : pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 t
lar 6 06:31:31 ip-172-31-35-28 sshd : pam_unix(sshd:auth): check pass; user unknown
lar 6 06:31:31 ip-172-31-35-28 sshd : pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 t
lar 6 06:31:31 ip-172-31-35-28 sshd : Invalid user admin from [REDACTED]
lar 6 06:31:31 ip-172-31-35-28 sshd : Invalid user admin from [REDACTED]
lar 6 06:31:31 ip-172-31-35-28 sshd : Invalid user admin from [REDACTED]
lar 6 06:31:31 ip-172-31-35-28 sshd : Invalid user admin from [REDACTED]
```

Question 2

The bruteforce attempts were successful and attacker gained access to an account on the server. What is the username of the account?

After brute forcing the attacker eventually gained access to the system. Log analysis showed the username of the account that the attacker gained access to was.

```
ip-172-31-35-28 sshd[ ]: Accepted password for [redacted] from [redacted] ssh2
ip-172-31-35-28 sshd[ ]: pam_unix(sshd:session): session opened for user root(uid=0) by (uid=0)
ip-172-31-35-28 systemd-logind[ ]: New session [redacted] of user [redacted].
```

Question 3

Identify the UTC timestamp when the attacker logged in manually to the server and established a terminal session to carry out their objectives. The login time will be different than the authentication time, and can be found in the wtmp artifact.

In order to view the contents of the artifact, the last command was utilized in the linux terminal. Unfortunately it was not possible to view the contents of this file. Therefore the tool provided in the Brutus was utilised. This tool is a python script that allows one to view the contents of a binary file.

Command used: `python3 utmp.py -o wtmp.out wtmp`

Output:

```

RON_LVL 53 ~ ~ runcvcl 6.2.0-1018-aws 0 0 0 2024/03/06 08:17:29 538024 0.0.0.0
"USER" "1583" "pts/0" "ts/0" " " "203.101.190.9" "0" "0" "0" [redacted] "151913" "203.101.190.9"
"USER" "2549" "pts/1" "ts/1" " " [redacted] "0" "0" "0" [redacted] "387923" [redacted]
"DEAD" "2491" "pts/1" "" "" "" "" "0" "0" "0" [redacted] "590579" "0.0.0.0"
"USER" "2667" "pts/1" "ts/1" " " [redacted] "0" "0" "0" [redacted] "475575" [redacted]
```

The wtmp artifact showed that there was a successful manual login on a specific date. A session was established but immediately closed. In order to confirm this: the auth.log file showed that there had been an authentication of a login a second before the session was established. The one second discrepancy can be explained by the fact that before authorisation can occur, authentication must happen; which is the purpose of the auth.log (to log successful authentication).

```

320 Mar 6 [redacted] ip-172-31-35-28 CRON[ ]: pam_unix(cron:session): session closed for user confluence
321 Mar 6 [redacted] ip-172-31-35-28 sshd[ ]: exited MaxStartups throttling after [redacted] connections dropped
322 Mar 6 [redacted] ip-172-31-35-28 sshd[ ]: Accepted password for root from [redacted] ssh2
323 Mar 6 [redacted] ip-172-31-35-28 sshd[ ]: pam_unix(sshd:session): session opened for user [redacted] (uid=0) by (uid=
324 Mar 6 [redacted] ip-172-31-35-28 systemd-logind[ ]: New session [redacted] of user [redacted]
325 Mar 6 [redacted] ip-172-31-35-28 CRON[ ]: pam_unix(cron:session): session opened for user confluence(uid=
```

Question 4

SSH login sessions are tracked and assigned a session number upon login. What is the session number assigned to the attacker's session for the user account from Question 2?

Finding the session id for the successful root login of the attacker was simple because it shows in the auth.log above. When the login was authenticated, a session number was given for the user.

Question 5

The attacker added a new user as part of their persistence strategy on the server and gave this new user account higher privileges. What is the name of this account?

To find the name of the account, the auth logs were analyzed and they revealed the following:

```
Mar 6 06:34:18 ip-172-31-35-28 useradd[ ]: new user: name= , UID= , GID= , home=
Mar 6 06:34:26 ip-172-31-35-28 passwd[ ]: pam_unix(passwd: ): password changed for
Mar 6 06:34:31 ip-172-31-35-28 chfn[ ]: changed user ' ' information
```

Afterward, this user was given higher privileges by being added to the sudo and shadow groups. This means this user can perform higher level activities on this system.

```
Mar 6 06:35:01 ip-172-31-35-28 CRON[ ]: pam_unix(cron:session): session closed for user confluence
Mar 6 06:35:15 ip-172-31-35-28 usermod[ ]: add to group 'sudo'
Mar 6 06:35:15 ip-172-31-35-28 usermod[ ]: add to shadow group 'sudo'
Mar 6 06:36:01 ip-172-31-35-28 CRON[ ]: pam_unix(cron:session): session opened for user confluence(uid= )
```

Question 6

What is the MITRE ATT&CK sub-technique ID used for persistence by creating a new account?

In order to find the subtechnique, the MITRE website was visited. Under enterprise techniques | Persistence and Create Account, the subtechnique was found. The framework's explanation fits how the attacker maintained persistence. They added a local account in Linux using useradd.

TECHNIQUES

Boot or Login

Initialization

Scripts

Cloud Application Integration

Compromise Host

Software Binary

Create Account

Local Account

Domain Account

Cloud Account

Create or Modify

System Process

Event Triggered

Execution

Create Account: Local Account

Other sub-techniques of Create Account (3)

Adversaries may create a local account to maintain access to victim systems. Local accounts are those configured by an organization for use by users, remote support, services, or for administration on a single system or service.

For example, with a sufficient level of access, the Windows `net user /add` command can be used to create a local account. In Linux, the `useradd` command can be used, while on macOS systems, the `dscl -create` command can be used. Local accounts may also be added to network devices, often via common [Network Device CLI](#) commands such as `username`, to ESXi servers via `esxcli system account add`, or to Kubernetes clusters using the `kubect1` utility.^{[1][2]}

Such accounts may be used to establish secondary credentialed access that do not require persistent remote access tools to be deployed on the system.

Procedure Examples

ID:

Sub-technique of: [T1136](#)

① Tactic: [Persistence](#)

① Platforms: Containers, ESXi, Linux, Network Devices, Window: macOS

Contributors: Austin Clark, @c2defense

Version: 1.4

Created: 28 January 2020

Last Modified: 15 April 2025

[Version Permalink](#)

Question 7

What time did the attacker's first SSH session end according to auth.log?

The auth log shows the time the attackers first ssh session when the attacker logged in ended at:

```
Mar 6 06:37:01 ip-172-31-35-28 CRON[ ]: pam_unix(cron:session): session closed for user confluence
Mar 6 06:37:24 ip-172-31-35-28 sshd[ ]: Received disconnect from [REDACTED]: disconnected by
Mar 6 06:37:24 ip-172-31-35-28 sshd[ ]: Disconnected from user [REDACTED]
Mar 6 06:37:24 ip-172-31-35-28 sshd[ ]: pam_unix(sshd:session): session closed for user root
Mar 6 06:37:24 ip-172-31-35-28 systemd-logind[ ]: Session [REDACTED] logged out. Waiting for processes to exit.
Mar 6 06:37:24 ip-172-31-35-28 systemd-logind[ ]: Removed session [REDACTED]
Mar 6 06:37:34 ip-172-31-35-28 sshd[ ]: Accepted password for [REDACTED] from [REDACTED] ssh2
```

The attacker's first ssh session ended at this time and the log shows the process from when the attacker initiated a disconnect from ssh to when the session was terminated.

Question 8

The attacker logged into their backdoor account and utilized their higher privileges to download a script. What is the full command executed using sudo?

Upon looking through the actions performed by the attacker after logging in with their backdoor account, it was noted that this was the command used to download a script. This information was found by looking through the actions the account created by the attacker performed.

COMMAND=

Conclusion

This Sherlock box involved investigating the aftermath of an attack using an auth.log, a wtmp artifact and a [utmp.py](#) script. Using these materials it was possible to determine the ip address of the attacker, the time they successfully logged in, the actions they performed after logging in and how they maintained persistence.

Lessons Learned

- How to use authentication logs to determine successful and unsuccessful log in attempts
- How to utilize a python script to read a binary file and extract valuable information from it such as timestamps
- How to analyze an auth.log to find out what actions an attacker took in the system, any sessions established and closed.
- How to use various sources of information to determine the timeline of an attack and the actions taken.