Defensive Security

1 Understanding Defensive Security and Challenges

Defensive security, also known as *blue teaming*, involves protecting information systems from unauthorized access, attacks, and misuse. The main objective is to detect, analyze, and mitigate threats before they can compromise the system. Defensive techniques include:

- Monitoring system logs and user activity
- Deploying intrusion detection/prevention systems (IDS/IPS)
- Hardening configurations and access controls
- Incident response and digital forensics

In Capture the Flag (CTF) platforms like Hack The Box, **defensive challenges** simulate real-world breaches. Participants analyze artifacts such as logs, binaries, or memory dumps to:

- Identify malicious activity
- Determine the method of attack
- Attribute actions to specific users or attackers
- Answer questions based on threat hunting or forensic analysis

These challenges help train professionals in roles such as SOC analysts, threat hunters, and incident responders.

2 Analyzing `auth.log`

What is `auth.log`?

`/var/log/auth.log` is a system log file (on Debian-based systems) that records authentication-related events, such as:

- Successful and failed login attempts
- SSH access
- `sudo`, `su`, and `passwd` usage
- PAM (Pluggable Authentication Module) events

On Red Hat-based systems, this log is typically named `/var/log/secure`.

Why is it Valuable in Defensive Security?

1. **Brute-force detection:** Identify repeated failed login attempts.
2. **User behavior monitoring:** Observe login patterns and privilege escalations.
3. **Threat hunting:** Correlate unusual access with other system events.
4. **Forensics:** Reconstruct an attack timeline or detect insider threats.

Identifying Brute Force Attacks

Brute force attacks appear as repeated login failures in `auth.log`:

```
Failed password for invalid user admin from 192.168.1.10
    port 54321 ssh2
```

Useful commands:

- **List all failed login attempts:**

```
grep "Failed␣password" /var/log/auth.log
```

- **Find failed attempts in a specific time frame:**

```
grep "Failed␣password" /var/log/auth.log | grep "Jul
   ␣16"
```

Useful Analysis Commands

- **View the log:**

```
less /var/log/auth.log
```

- **Search SSH logins:**

```
grep "sshd" /var/log/auth.log
```

- **List successful logins:**

```
grep "Accepted␣password" /var/log/auth.log
```

- **List successful logins for a specific IP:**

```
grep "Accepted␣password" /var/log/auth.log | grep <
   ip_address>
```

- **Track sudo usage:**

```
grep "sudo" /var/log/auth.log
```

- **Audit a specific user or IP:**

```
grep "john" /var/log/auth.log
grep "192.168.1.5" /var/log/auth.log
```

3 Understanding and Using `wtmp`

What is `wtmp`?

`/var/log/wtmp` is a binary log file that stores historical records of:

- User logins and logouts
- System boots and shutdowns
- Remote login sessions (e.g., SSH)

Unlike `auth.log`, this file is not human-readable and must be parsed with tools like `last`.

Useful Commands to Read `wtmp`

- **List recent login sessions:**

```
last
```

- **Read the wtmp file explicitly:**

```
last -f /var/log/wtmp
```

- **Show only reboots:**

```
last reboot
```

- **Show SSH logins:**

```
last | grep ssh
```

Why It Matters in Defensive Security

`wtmp` is essential for:

- A historical view of user activity
- Verifying session start/end times
- Detecting system reboots (potential log wiping)
- Correlating logins with other forensic artifacts

Use in a CTF Context

In challenges like Hack The Box's Sherlock:

- Verify login timestamps
- Confirm if an attacker gained SSH access
- See if reboot attempts were used to cover tracks

4 Detecting User Persistence

Attackers often create new user accounts to maintain access over time — a tactic known as persistence. This can often be spotted in `auth.log`.

Look for:

- `useradd` — new user created
- `usermod` — privileges modified
- `groupadd` — new group created

Example commands:

```
grep "useradd" /var/log/auth.log
grep "usermod" /var/log/auth.log
grep "groupadd" /var/log/auth.log
```

These entries often appear when an attacker tries to silently add a new user or escalate its privileges to root/admin.

Conclusion

Logs like `auth.log` and `wtmp` are invaluable in both real-world defensive security and CTF challenges. They provide insight into system access, user behavior, and potentially malicious activity. Understanding how to parse and correlate these files can greatly enhance an analyst's ability to detect and respond to incidents.