

Task 6

Log Analysis & Intrusion Detection

Setup

1. Before analyzing logs, we need to ensure that system logging is active.

```
(kali@kali)-[~]  
$ sudo systemctl start systemd-journald  
sudo systemctl enable systemd-journald
```

The unit files have no installation config (WantedBy=, RequiredBy=, UpheldBy=, Also=, or Alias= settings in the [Install] section, and DefaultInstance= for template units). This means they are not meant to be enabled or disabled using systemctl.

Possible reasons for having these kinds of units are:

- A unit may be statically enabled by being symlinked from another unit's .wants/, .requires/, or .upholds/ directory.
- A unit's purpose may be to act as a helper for some other unit which has a requirement dependency on it.
- A unit may be started when needed via activation (socket, path, timer, D-Bus, udev, scripted systemctl call, ...).
- In case of template units, the unit is meant to be enabled with some instance name specified.

2. Check logs:

```

File Actions Edit View Help
instance name specified.

(kali@kali)-[~]
$ journalctl --since "1 hour ago"
cat /var/log/auth.log | tail -50
Mar 25 00:18:22 kali kernel: 02:49:02.991530 timesync vgsvcTimeSyncWorker: Radical guest time change: 33 532 389 731 000ns (
Mar 25 00:18:22 kali kernel: watchdog: BUG: soft lockup - CPU#1 stuck for 1000s! [swapper/1:0]
Mar 25 00:18:22 kali kernel: Modules linked in: ip6t_REJECT nf_reject_ipv6 xt_hl ip6t_rt ipt_REJECT nf_reject_ipv4 xt_LOG nf
Mar 25 00:18:22 kali kernel: usb_common aesni_intel gf128mul crypto_simd cryptd
Mar 25 00:18:22 kali kernel: CPU: 1 UID: 0 PID: 0 Comm: swapper/1 Tainted: G L 6.11.2-amd64 #1 Kali 6.11.2-
Mar 25 00:18:22 kali kernel: Tainted: [L]=SOFTLOCKUP
Mar 25 00:18:22 kali kernel: Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
Mar 25 00:18:22 kali kernel: RIP: 0010:pv_native_safe_halt+0xf/0x20
Mar 25 00:18:22 kali kernel: Code: 22 d7 e9 64 16 01 00 0f 1f 40 00 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 f3 0f 1e
Mar 25 00:18:22 kali kernel: RSP: 0018:ffffb802000bbd8 EFLAGS: 00010212
Mar 25 00:18:22 kali kernel: RAX: 0000000000000001 RBX: ffff963d012f1840 RCX: 000000010042d497
Mar 25 00:18:22 kali kernel: RDX: 0000000000000000 RSI: 0000000000000082 RDI: 0000000054799fc
Mar 25 00:18:22 kali kernel: RBP: 0000000000000001 R08: ffff802000bbe48 R09: 0000000000000000
Mar 25 00:18:22 kali kernel: R10: 00000000e0ccdeeb R11: 000000000000130d R12: 0000000000000000
Mar 25 00:18:22 kali kernel: R13: 0000000000000000 R14: 0000000000000000 R15: 0000000000000000
Mar 25 00:18:22 kali kernel: FS: 0000000000000000(0000) GS:ffff963dbcd00000(0000) knlGS:0000000000000000
Mar 25 00:18:22 kali kernel: CS: 0010 DS: 0000 ES: 0000 CR0: 0000000080050033
Mar 25 00:18:22 kali kernel: CR2: 000056370a317620 CR3: 00000000109ce000 CR4: 0000000000106f0
Mar 25 00:18:22 kali kernel: Call Trace:
Mar 25 00:18:22 kali kernel: <IRQ>
Mar 25 00:18:22 kali kernel: ? watchdog_timer_fn.cold+0x3d/0xa1
Mar 25 00:18:22 kali kernel: ? __pfx_watchdog_timer_fn+0x10/0x10
Mar 25 00:18:22 kali kernel: ? hrtimer_run_queues+0x132/0x2a0
Mar 25 00:18:22 kali kernel: ? hrtimer_interrupt+0xfa/0x210
Mar 25 00:18:22 kali kernel: ? __sysvec_apic_timer_interrupt+0x55/0x100
Mar 25 00:18:22 kali kernel: ? sysvec_apic_timer_interrupt+0x6c/0x90
Mar 25 00:18:22 kali kernel: </IRQ>
Mar 25 00:18:22 kali kernel: <TASK>
Mar 25 00:18:22 kali kernel: ? asm_sysvec_apic_timer_interrupt+0x1a/0x20
Mar 25 00:18:22 kali kernel: ? pv_native_safe_halt+0xf/0x20
Mar 25 00:18:22 kali kernel: default_idle+0x9/0x20
Mar 25 00:18:22 kali kernel: default_idle_call+0x29/0x100
Mar 25 00:18:22 kali kernel: do_idle+0x1fe/0x240
Mar 25 00:18:22 kali kernel: cpu_startup_entry+0x29/0x30
Mar 25 00:18:22 kali kernel: start_secondary+0x11c/0x140
Mar 25 00:18:22 kali kernel: common_startup_64+0x13e/0x141
Mar 25 00:18:22 kali kernel: </TASK>
Mar 25 00:18:22 kali systemd[1]: Starting phpsessionclean.service - Clean php session files...
Mar 25 00:18:22 kali systemd[1]: systemd-journald.service: Main process exited, code=killed, status=6/ABRT
Mar 25 00:18:22 kali systemd[1]: systemd-journald.service: Failed with result 'watchdog'.
Mar 25 00:18:22 kali systemd[1]: systemd-journald.service: Consumed 2.278s CPU time, 3M memory peak.
Mar 25 00:18:22 kali systemd[1]: run-credentials-systemd\x2djournald.service.mount: Deactivated successfully.
Mar 25 00:18:22 kali systemd[1]: systemd-logind.service: Main process exited, code=killed, status=6/ABRT
Mar 25 00:18:22 kali systemd[1]: systemd-logind.service: Failed with result 'watchdog'.
Mar 25 00:18:22 kali systemd[1]: systemd-logind.service: Consumed 1.145s CPU time, 2.2M memory peak.
Mar 25 00:18:22 kali systemd[1]: systemd-journald.service: Scheduled restart job, restart counter is at 2.

```

Exploit

1. **Analyze Logs for Failed SSH Logins:** Find failed attempts:
2. Count occurrences per IP:

Identify Brute-Force Attempts & Unauthorized Access:

- Check repeated failed attempts from the same IP.
- Review timestamps for patterns.
- Validate against legitimate access logs:

```

(kali@kali)-[~]
$ grep "Failed password" /var/log/auth.log
2025-03-25T01:04:13.359237-04:00 kali sudo:    kali : TTY=pts/0 ; PWD=/home/kali ; USER=root ; COMMAND=/usr/bin/grep 'Failed password' /var/log/auth.log
(kali@kali)-[~]
$ grep "Failed password" /var/log/auth.log | awk '{print $(NF-3)}' | sort | uniq -c | sort -nr
1 COMMAND=/usr/bin/grep
(kali@kali)-[~]
$ grep "Accepted password" /var/log/auth.log

```

Mitigation

1. Implement Fail2Ban to Block Repeated Failed Attempts:

Install Fail2Ban

```

(kali@kali)-[~]
$ sudo apt install fail2ban -y

```

Configure SSH protection:

```

(kali@kali)-[~]
$ sudo nano /etc/fail2ban/jail.local

```

```

GNU nano 8.2
[sshd]
enabled = true
port = ssh
filter = sshd
logpath = /var/log/auth.log
maxretry = 3
bantime = 600

```

Restart Fail2Ban

```

(kali@kali)-[~]
$ sudo systemctl restart fail2ban
sudo fail2ban-client status sshd
2025-03-25 01:08:43,696 fail2ban [165701]: ERROR Failed to access socket path: /var/run/fail2ban/fail2ban.sock. Is fail2ban running?

```

2. Set Up Log Monitoring Automation:

```
(kali㉿kali)-[~]  
$ sudo apt install logwatch -y  
  
Installing:  
  logwatch  
File System  
Suggested packages:  
  libsys-cpu-perl  libsys-meminfo-perl  
  
Summary:  
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 1558  
  Download size: 390 kB  
  Space needed: 2,451 kB / 62.7 GB available  
  
Get:1 http://kali.download/kali kali-rolling/main amd64 logwatch all 7.12-3 [390 kB]  
Fetched 390 kB in 23s (16.8 kB/s)  
Selecting previously unselected package logwatch.  
(Reading database ... 403562 files and directories currently installed.)  
Preparing to unpack .../logwatch_7.12-3_all.deb ...  
Unpacking logwatch (7.12-3) ...  
Setting up logwatch (7.12-3) ...  
Processing triggers for man-db (2.13.0-1) ...  
Processing triggers for kali-menu (2024.4.0) ...
```

Generate a security report

```
(kali@kali)-[~]
$ sudo logwatch --detail high --service sshd --range today

##### Logwatch 7.12 (01/22/25) #####
Processing Initiated: Tue Mar 25 01:09:56 2025
Date Range Processed: today
                      ( 2025-Mar-25 )
                      Period is day.
Detail Level of Output: 10
Type of Output/Format: stdout / text
Logfiles for Host: kali
#####

----- SSHD Begin -----

SSHD Killed: 1 Time

SSHD Started: 2 Times

Illegal users from:
  ::1 (localhost): 1 Time
  invalid_user: 1 Time

----- SSHD End -----

##### Logwatch End #####
```

Configure rsyslog for centralized logging

Ensure remote logging is enabled if needed.

```
(kali@kali)-[~]
$ sudo nano /etc/rsyslog.conf
```

Conclusion

In this task, we successfully analyzed SSH login attempts, identified unauthorized access, and implemented security measures. By using system logs (journalctl or /var/log/auth.log), we detected failed login attempts and potential brute-force attacks. Fail2Ban was configured to block repeated failed logins, enhancing system security. Additionally, log monitoring automation was set up using logwatch to ensure continuous threat detection.