

Step 1: Explore Confidentiality

Objective: Understand how confidentiality is maintained and identify security violations when unauthorized access occurs.

1. Create a File Containing Sensitive Information

- Open a terminal or command prompt.
- Create a text file named `sensitive_file.txt`:
 - Linux: `echo "Confidential Data: Usernames and Passwords" > sensitive_file.txt`
 - Windows: Open Notepad, type "Confidential Data: Usernames and Passwords", and save the file as `sensitive_file.txt`.

2. Restrict File Permissions

- Set file permissions so only the owner can access it:
 - Linux: Run `chmod 600 sensitive_file.txt`.
 - Explanation: The 600 permission allows the owner to read/write the file but denies access to others.
 - Windows: Right-click the file → Properties → Security → Edit permissions → Deny access for all users except the owner.

3. Simulate Unauthorized Access

- Switch to another user or simulate unauthorized access:
 - Linux: Use `su` or create a new user, then try accessing the file: `cat sensitive_file.txt`.
 - Windows: Switch user accounts or create a new user, then try opening the file.
- Observe the error message (e.g., "Permission denied").
- Discussion: Analyze how this protects confidentiality. Discuss real-world examples, such as protecting medical or financial records.

```
student@student-VMware-Virtual-Platform:~$ echo "Confidential Data: Usernames and Passwords" > sensitive_file.txt
student@student-VMware-Virtual-Platform:~$ cat sensitive_file.txt
Confidential Data: Usernames and Passwords
student@student-VMware-Virtual-Platform:~$ chmod 600 sensitive_file.txt.
chmod: cannot access 'sensitive_file.txt.': No such file or directory
student@student-VMware-Virtual-Platform:~$ echo "Confidential Data: Usernames and Passwords" > sensitive_file.txt
student@student-VMware-Virtual-Platform:~$ cat sensitive_file.txt
Confidential Data: Usernames and Passwords
student@student-VMware-Virtual-Platform:~$ chmod 600 sensitive_file.txt
student@student-VMware-Virtual-Platform:~$ ls -l sensitive_file.txt
-rw----- 1 student student 43 Mar 27 16:15 sensitive_file.txt
student@student-VMware-Virtual-Platform:~$ su alice
Password:
alice@student-VMware-Virtual-Platform:/home/student$ cat /home/student/sensitive_file.txt
cat: /home/student/sensitive_file.txt: Permission denied
alice@student-VMware-Virtual-Platform:/home/student$ █
```

Step 2: Analyze Integrity

Objective: Understand how data integrity can be compromised and verify its integrity using

hashing.

1. Create or Access a Log File

- Use an existing log file or create a simulated one:
 - Linux: `sudo nano /var/log/syslog` (requires root access).
 - Windows: Open Event Viewer (eventvwr) or create a text file named logfile.txt.

2. Modify the Log File (Simulate Unauthorized Changes)

- Add or change log entries to simulate a security violation:
 - Linux: Edit the file: `sudo nano /var/log/syslog` → Add a fake entry:
Jan 1 12:00:00 UnauthorizedAccess: Admin login.
 - Windows: Open logfile.txt in Notepad and add UnauthorizedAccess: Admin login.

3. Verify Integrity with Hashing

- Calculate the file's hash before and after modification:
 - Linux: Use `sha256sum logfile.txt` and note the hash.
 - Windows: Use PowerShell: `Get-FileHash .\logfile.txt -Algorithm SHA256`
- Observe the hash difference.
- Discussion: Discuss how unauthorized changes compromise data integrity. Relate this to tampering scenarios in real-world applications, such as altering financial records or audit logs.

```
GNU nano 7.2 /var/log/syslog
2025-03-27T00:00:05.575982+05:30 student-VMware-Virtual-Platform rsyslogd: [origin software="rsyslogd" swVersion="8.2312.0" x-pid="1094" x-info="https://www.>
2025-03-27T00:00:05.584026+05:30 student-VMware-Virtual-Platform systemd[1]: logrotate.service: Deactivated successfully.
2025-03-27T00:00:05.584127+05:30 student-VMware-Virtual-Platform systemd[1]: Finished logrotate.service - Rotate log files.
2025-03-27T00:00:54.377828+05:30 student-VMware-Virtual-Platform systemd[2176]: Started snap.firmware-updater.firmware-notifier.service - Service for snap ap
2025-03-27T00:00:54.471543+05:30 student-VMware-Virtual-Platform kernel: audit: type=1400 audit(1743013854.468:188): apparmor="DENIED" operation="open" class>
2025-03-27T00:00:54.489590+05:30 student-VMware-Virtual-Platform systemd[1]: tmp-snap.rootfs_tWdMid.mount: Deactivated successfully.
2025-03-27T00:00:54.497472+05:30 student-VMware-Virtual-Platform kernel: audit: type=1400 audit(1743013854.495:189): apparmor="DENIED" operation="open" class>
2025-03-27T00:00:54.710204+05:30 student-VMware-Virtual-Platform kernel: audit: type=1400 audit(1743013854.708:190): apparmor="DENIED" operation="open" class>
2025-03-27T00:00:55.792335+05:30 student-VMware-Virtual-Platform kernel: audit: type=1400 audit(1743013855.790:191): apparmor="DENIED" operation="open" class>
2025-03-27T00:00:55.842901+05:30 student-VMware-Virtual-Platform dbus-daemon[840]: [system] Activating via systemd: service name='org.freedesktop.fwupd' unit>
2025-03-27T00:00:55.853198+05:30 student-VMware-Virtual-Platform systemd[1]: Starting fwupd.service - Firmware update daemon...
2025-03-27T00:00:56.059117+05:30 student-VMware-Virtual-Platform dbus-daemon[840]: [system] Activating via systemd: service name='org.bluez' unit='dbus-org.b>
2025-03-27T00:00:56.061406+05:30 student-VMware-Virtual-Platform systemd[1]: bluetooth.service - Bluetooth service was skipped because of an unmet condition >
2025-03-27T00:00:57.674589+05:30 student-VMware-Virtual-Platform fwupd[6327]: 18:30:57.674 FuMain
2025-03-27T00:00:57.678422+05:30 student-VMware-Virtual-Platform dbus-daemon[840]: [system] Successfully activated service 'org.freedesktop.fwupd'
2025-03-27T00:00:57.678727+05:30 student-VMware-Virtual-Platform systemd[1]: Started fwupd.service - Firmware update daemon.
2025-03-27T00:02:29.774624+05:30 student-VMware-Virtual-Platform kernel: perf: Interrupt took too long (10873 > 10753), lowering kernel.perf_event_max_sample>
2025-03-27T00:03:11.254281+05:30 student-VMware-Virtual-Platform rtkit-daemon[1655]: Supervising 10 threads of 7 processes of 1 users.
2025-03-27T00:03:18.116685+05:30 student-VMware-Virtual-Platform rtkit-daemon[1655]: message repeated 5 times: [ Supervising 10 threads of 7 processes of 1 us>
2025-03-27T00:05:01.052185+05:30 student-VMware-Virtual-Platform CRON[6487]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)
2025-03-27T00:06:54.379726+05:30 student-VMware-Virtual-Platform systemd[1]: Starting fwupd-refresh.service - Refresh fwupd metadata and update motd...
2025-03-27T00:06:54.441745+05:30 student-VMware-Virtual-Platform systemd[1]: fwupd-refresh.service: Deactivated successfully.
2025-03-27T00:06:54.441880+05:30 student-VMware-Virtual-Platform systemd[1]: Finished fwupd-refresh.service - Refresh fwupd metadata and update motd.
2025-03-27T00:07:17.233389+05:30 student-VMware-Virtual-Platform systemd[1]: Starting sysstat-summary.service - Generate a daily summary of process accountin>
2025-03-27T00:07:17.278454+05:30 student-VMware-Virtual-Platform systemd[1]: sysstat-summary.service: Deactivated successfully.
2025-03-27T00:07:17.278946+05:30 student-VMware-Virtual-Platform systemd[1]: Finished sysstat-summary.service - Generate a daily summary of process accountin>
2025-03-27T00:10:07.236227+05:30 student-VMware-Virtual-Platform systemd[1]: Starting sysstat-collect.service - system activity accounting tool...
2025-03-27T00:10:07.243974+05:30 student-VMware-Virtual-Platform systemd[1]: sysstat-collect.service: Deactivated successfully.
2025-03-27T00:10:07.244326+05:30 student-VMware-Virtual-Platform systemd[1]: Finished sysstat-collect.service - system activity accounting tool.
2025-03-27T00:12:22.166098+05:30 student-VMware-Virtual-Platform NetworkManager[1122]: <info> [1743014542.1649] dhcp4 (ens33): state changed new lease, addr>

Read 19549 lines
⌘ Help      ⌘ Write Out  ⌘ Where Is   ⌘ Cut        ⌘ Execute    ⌘ Location   ⌘ Undo      ⌘ Set Mark   ⌘ To Bracket
⌘ Exit      ⌘ Read File  ⌘ Replace    ⌘ Paste      ⌘ Justify    ⌘ Go To Line ⌘ Redo      ⌘ Copy       ⌘ Where Was
```

```
alice@student-VMware-Virtual-Platform:/home/student$ sudo nano /var/log/syslog
alice@student-VMware-Virtual-Platform:/home/student$ sha256sum logfile.txt
sha256sum: logfile.txt: Permission denied
alice@student-VMware-Virtual-Platform:/home/student$ sudo nano /var/log/syslog
alice@student-VMware-Virtual-Platform:/home/student$ sha256sum logfile.txt
sha256sum: logfile.txt: Permission denied
alice@student-VMware-Virtual-Platform:/home/student$ ls
ls: cannot open directory '.': Permission denied
alice@student-VMware-Virtual-Platform:/home/student$ █
```

Step 3: Examine Availability

Objective: Explore how system availability is affected during an attack or resource overload.

1. Set Up a Simple Web Server (Optional)

- Linux: Use Python to start a basic web server: `python3 -m http.server 8080`
- Windows: Use IIS or WAMP/XAMPP to set up a local server.

2. Simulate Denial-of-Service (DoS) Attack

- Use a tool to overload the server with requests:
 - Linux: Install and use ab (Apache Benchmark):
`ab -n 1000 -c 100 http://localhost:8080/`
-n: Total number of requests.
-c: Number of concurrent requests.
 - Windows: Use a custom PowerShell script or any load-testing tool like JMeter.

3. Monitor Server Behavior

- Observe the server response time during the attack:
 - Linux: Check server logs or terminal output for delays or errors.
 - Windows: Use Task Manager or Resource Monitor to track CPU and network usage.
- Note any timeouts or connection refusals.

4. Restore Normal Operations

- Stop the attack and ensure normal availability:
 - Linux: Terminate the ab command or server process (Ctrl+C).
 - Windows: Stop the server or restart it via IIS Manager.
- Verify that the server is responsive again.

5. Discussion

- Relate this to real-world examples of denial-of-service attacks on popular websites or applications. Discuss strategies to mitigate such attacks, such as rate-limiting, firewalls, or load balancers.

[illegible]

b. Analyze how each violation impacts the overall security of the system.