Lab#08: Web Server Defacement – Incident Response Simulation Lab

🎯 Objective

Students will:

- Investigate a defaced website hosted locally on Apache.
- Identify and remove the malicious file.
- Restore the original website.
- Patch and harden the system post-incident.

🔧 Step-by-Step Instructions with Explanations

🔹 Step 1: Install and Configure Apache Web Server

🔹 Step 2: Deploy the Original Website

🔹 Step 3: Simulate a Defacement Attack

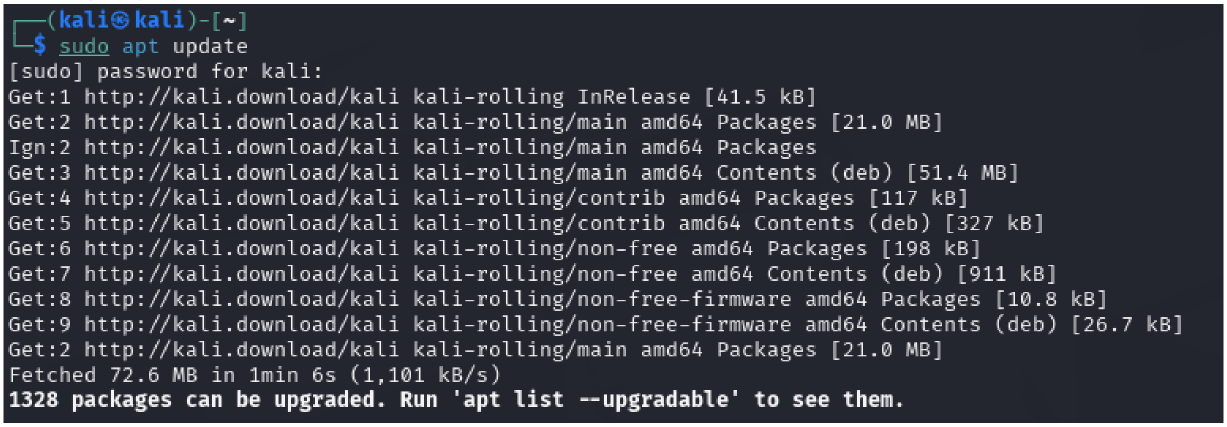
🔍 Step 4: Investigate the Incident

🧹 Step 5: Remove the Malicious File & Restore the Original

🔐 Step 6: Patch & Harden the System

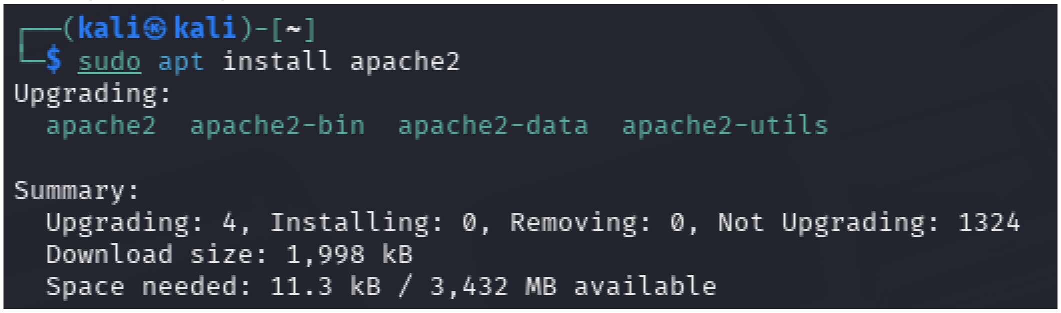
Step 1: Install and Configure Apache Web Server

```
sudo apt update
```



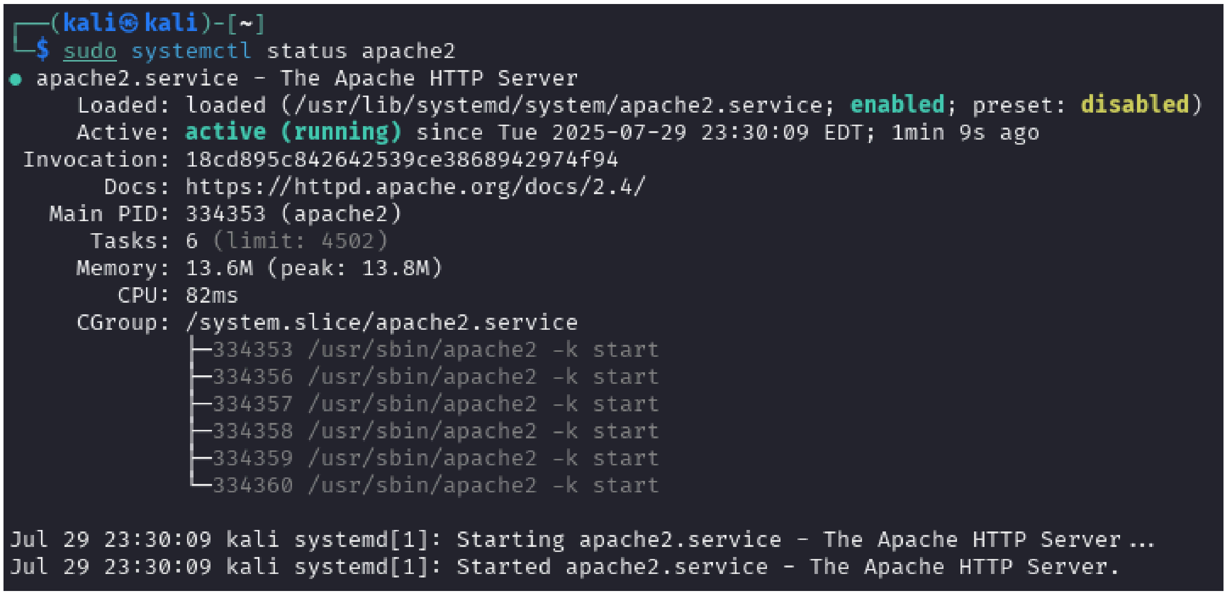
```
┌──(kali㉿kali)-[~]
└─$ sudo apt update
[sudo] password for kali:
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [21.0 MB]
Ign:2 http://kali.download/kali kali-rolling/main amd64 Packages
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [51.4 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [117 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [327 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [198 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [911 kB]
Get:8 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [10.8 kB]
Get:9 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb) [26.7 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [21.0 MB]
Fetched 72.6 MB in 1min 6s (1,101 kB/s)
1328 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

sudo apt install apache2



```
┌──(kali㉿kali)-[~]
└─$ sudo apt install apache2
Upgrading:
  apache2  apache2-bin  apache2-data  apache2-utils

Summary:
  Upgrading: 4, Installing: 0, Removing: 0, Not Upgrading: 1324
  Download size: 1,998 kB
  Space needed: 11.3 kB / 3,432 MB available
```

After installation, check if the service is running:
sudo systemctl status apache2

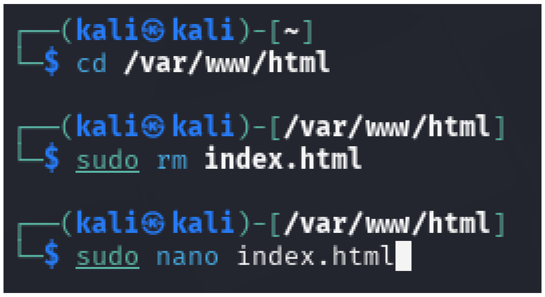


```
┌──(kali㉿kali)-[~]
└─$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: disabled)
     Active: active (running) since Tue 2025-07-29 23:30:09 EDT; 1min 9s ago
 Invocation: 18cd895c842642539ce3868942974f94
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 334353 (apache2)
      Tasks: 6 (limit: 4502)
     Memory: 13.6M (peak: 13.8M)
        CPU: 82ms
     CGroup: /system.slice/apache2.service
             ├─334353 /usr/sbin/apache2 -k start
             ├─334356 /usr/sbin/apache2 -k start
             ├─334357 /usr/sbin/apache2 -k start
             ├─334358 /usr/sbin/apache2 -k start
             ├─334359 /usr/sbin/apache2 -k start
             └─334360 /usr/sbin/apache2 -k start

Jul 29 23:30:09 kali systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jul 29 23:30:09 kali systemd[1]: Started apache2.service - The Apache HTTP Server.
```

Step 2: Deploy the Original Website

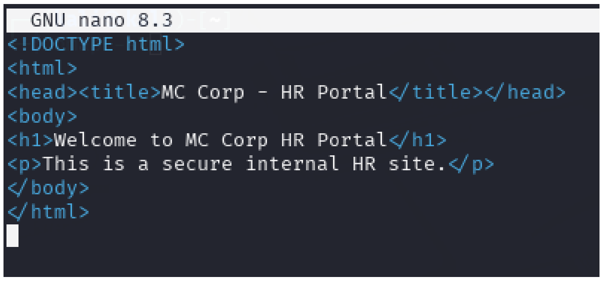
Navigate to web root:
cd /var/www/html
sudo rm index.html

Creating the new web page
Explanation: This simulates a legitimate site hosted on an internal web server.
sudo nano index.html

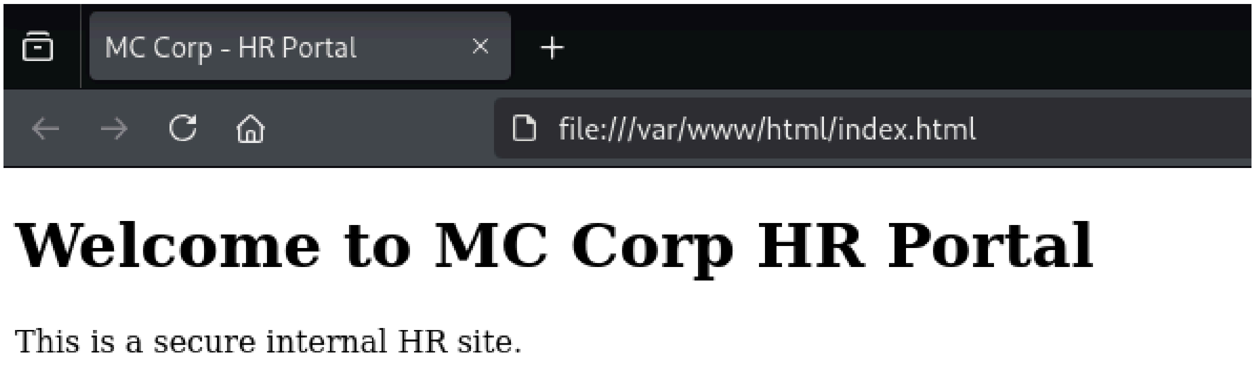


```
┌──(kali㉿kali)-[~]
└─$ cd /var/www/html

┌──(kali㉿kali)-[/var/www/html]
└─$ sudo rm index.html

┌──(kali㉿kali)-[/var/www/html]
└─$ sudo nano index.html
```

Creating the new web page



```
  GNU nano 8.3
<!DOCTYPE html>
<html>
<head><title>MC Corp - HR Portal</title></head>
<body>
<h1>Welcome to MC Corp HR Portal</h1>
<p>This is a secure internal HR site.</p>
</body>
</html>
```

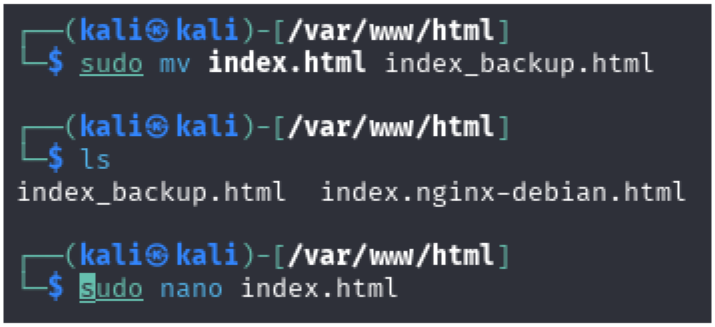


Welcome to MC Corp HR Portal

This is a secure internal HR site.

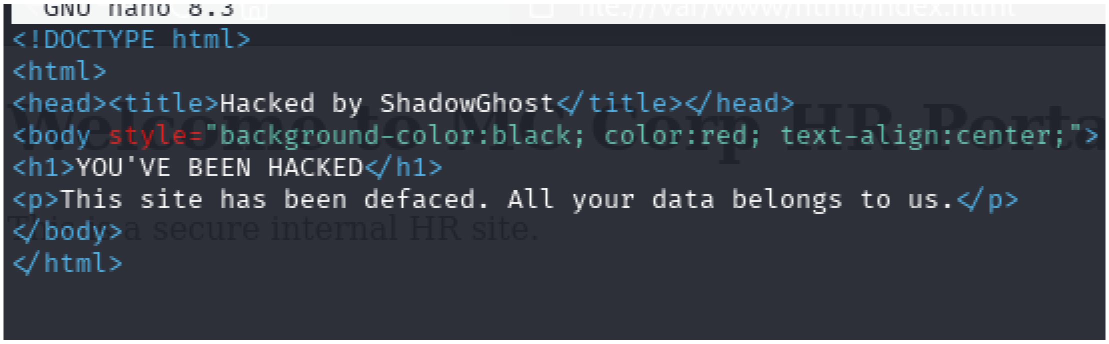
Step 3: Simulate a Defacement Attack

Replace the page with defaced content:
sudo mv index.html index_backup.html
sudo nano index.html



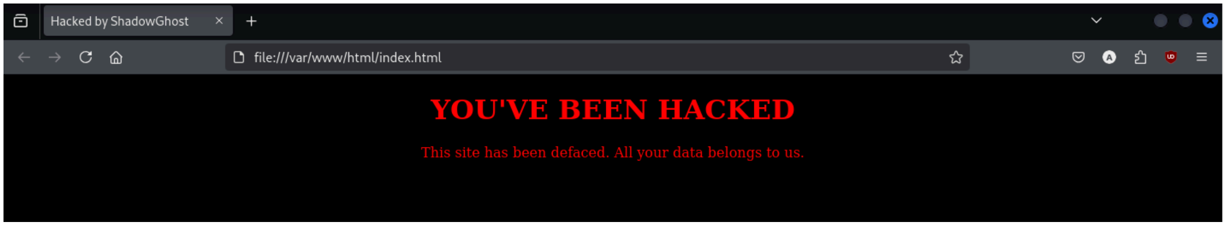
```
┌──(kali㉿kali)-[/var/www/html]
└─$ sudo mv index.html index_backup.html

┌──(kali㉿kali)-[/var/www/html]
└─$ ls
index_backup.html  index.nginx-debian.html

┌──(kali㉿kali)-[/var/www/html]
└─$ sudo nano index.html
```

The content being replaced:



```
  GNU nano 8.3
<!DOCTYPE html>
<html>
<head><title>Hacked by ShadowGhost</title></head>
<body style="background-color:black; color:red; text-align:center;">
<h1>YOU'VE BEEN HACKED</h1>
<p>This site has been defaced. All your data belongs to us.</p>
</body>
</html>
```

The result after the change:

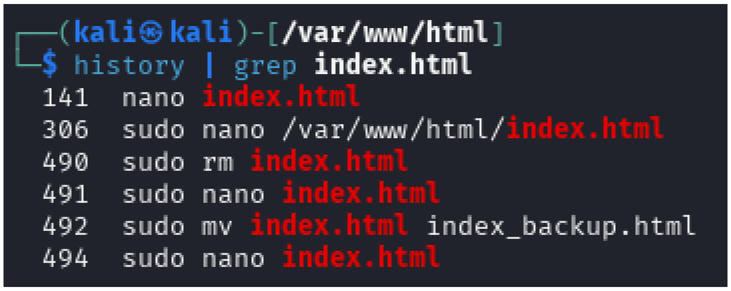


YOU'VE BEEN HACKED

This site has been defaced. All your data belongs to us.

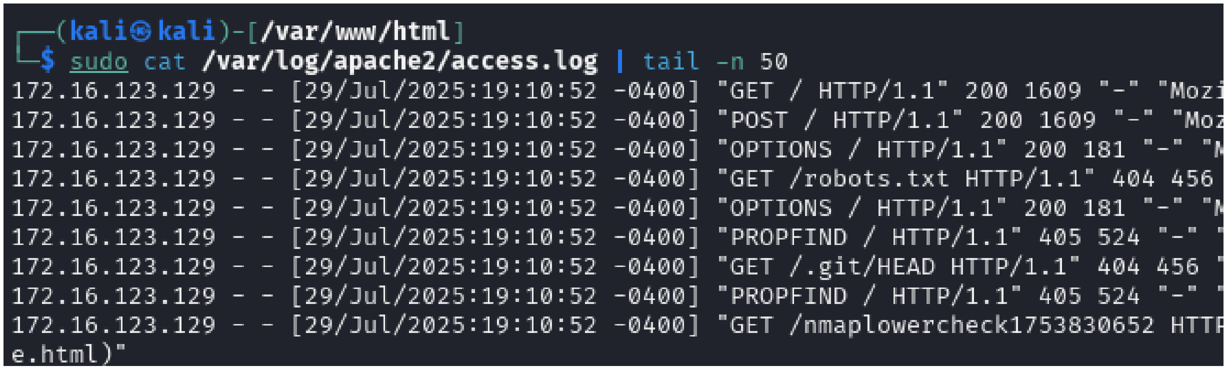
Step 4: Investigate the Incident

Check the bash history
history | grep index.html



```
┌──(kali㉿kali)-[/var/www/html]
└─$ history | grep index.html
  141  nano index.html
  306  sudo nano /var/www/html/index.html
  490  sudo rm index.html
  491  sudo nano index.html
  492  sudo mv index.html index_backup.html
  494  sudo nano index.html
```

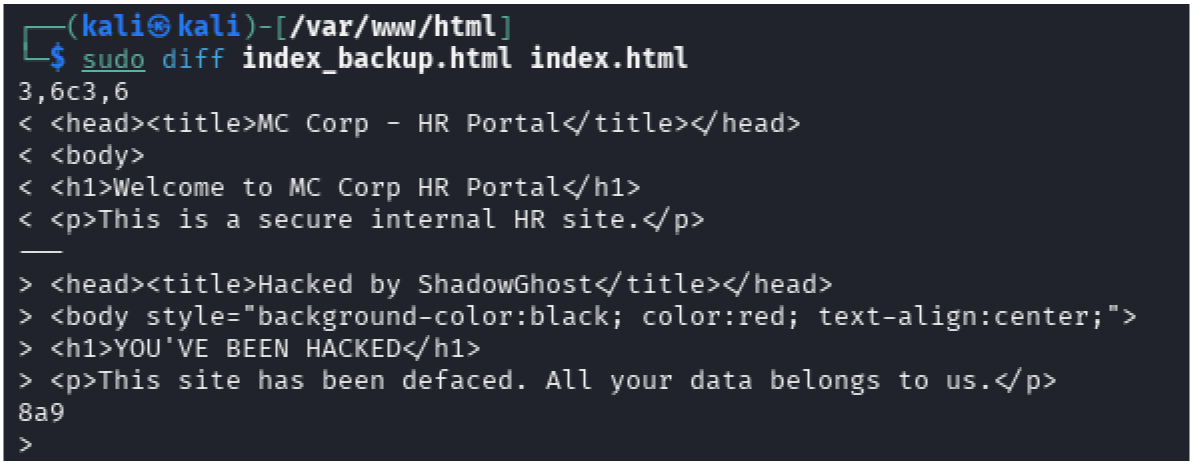
Check the Apache logs
sudo cat /var/log/apache2/access.log | tail -n 50



```
┌──(kali㉿kali)-[/var/www/html]
└─$ sudo cat /var/log/apache2/access.log | tail -n 50
172.16.123.129 - - [29/Jul/2025:19:10:52 -0400] "GET / HTTP/1.1" 200 1609 "-" "Mozi
172.16.123.129 - - [29/Jul/2025:19:10:52 -0400] "POST / HTTP/1.1" 200 1609 "-" "Moz
172.16.123.129 - - [29/Jul/2025:19:10:52 -0400] "OPTIONS / HTTP/1.1" 200 181 "-" "M
172.16.123.129 - - [29/Jul/2025:19:10:52 -0400] "GET /robots.txt HTTP/1.1" 404 456
172.16.123.129 - - [29/Jul/2025:19:10:52 -0400] "OPTIONS / HTTP/1.1" 200 181 "-" "M
172.16.123.129 - - [29/Jul/2025:19:10:52 -0400] "PROPFIND / HTTP/1.1" 405 524 "-" "
172.16.123.129 - - [29/Jul/2025:19:10:52 -0400] "GET /.git/HEAD HTTP/1.1" 404 456 "
172.16.123.129 - - [29/Jul/2025:19:10:52 -0400] "PROPFIND / HTTP/1.1" 405 524 "-" "
172.16.123.129 - - [29/Jul/2025:19:10:52 -0400] "GET /nmaplowercheck1753830652 HTTP
e.html)"
```

Use diff to compare original vs defaced:

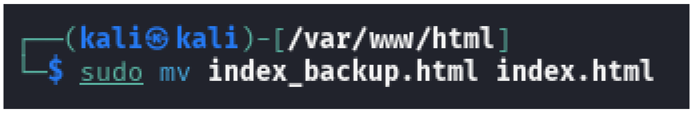
sudo diff index_backup.html index.html



```
┌──(kali㉿kali)-[/var/www/html]
└─$ sudo diff index_backup.html index.html
3,6c3,6
< <head><title>MC Corp - HR Portal</title></head>
< <body>
< <h1>Welcome to MC Corp HR Portal</h1>
< <p>This is a secure internal HR site.</p>
---
> <head><title>Hacked by ShadowGhost</title></head>
> <body style="background-color:black; color:red; text-align:center;">
> <h1>YOU'VE BEEN HACKED</h1>
> <p>This site has been defaced. All your data belongs to us.</p>
8a9
>
```

Step 5: Remove the Malicious File & Restore the Original

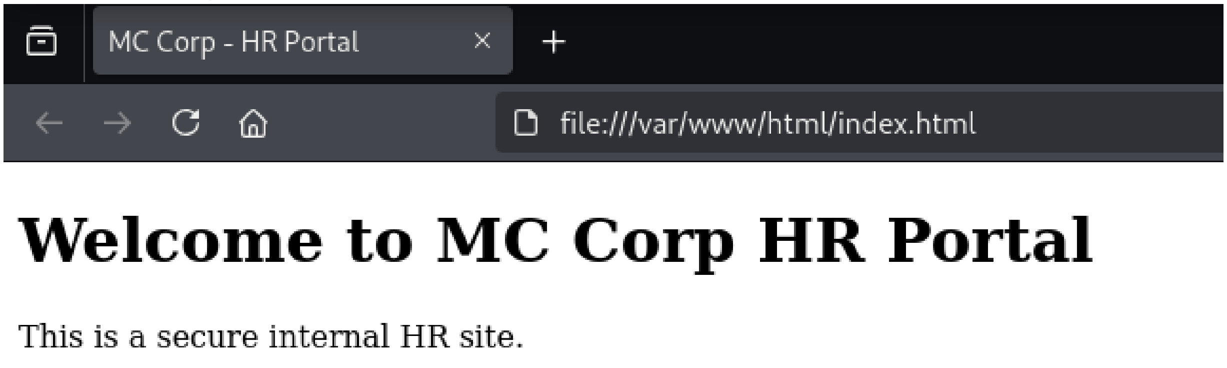
sudo mv index_backup.html index.html



```
┌──(kali㉿kali)-[/var/www/html]
└─$ sudo mv index_backup.html index.html
```

Explanation: This brings the site back to the legitimate version.

After refreshing the browser:

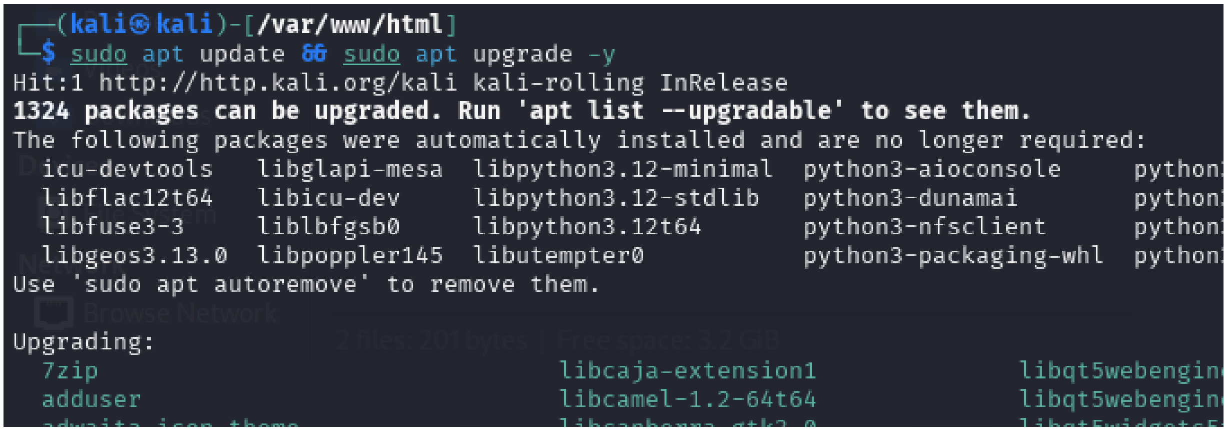


Welcome to MC Corp HR Portal

This is a secure internal HR site.

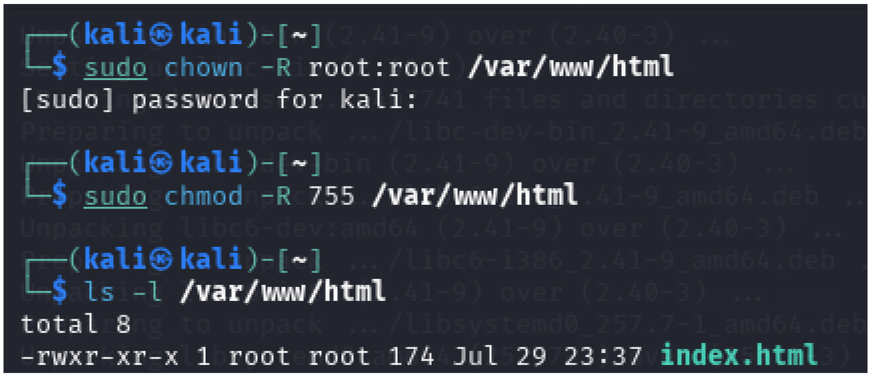
Step 6: Patch & Harden the System

Update all packages
sudo apt update && sudo apt upgrade -y



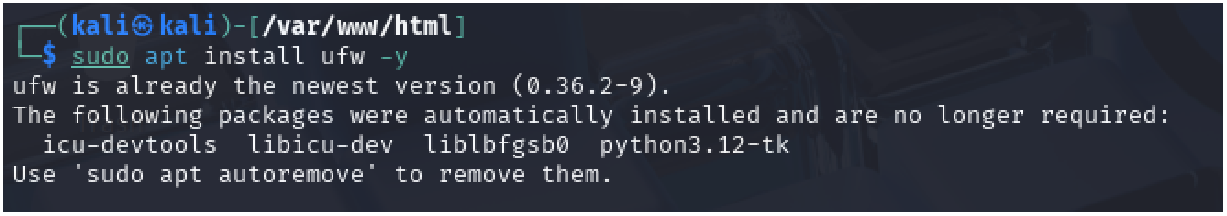
```
┌──(kali㉿kali)-[/var/www/html]
└─$ sudo apt update && sudo apt upgrade -y
Hit:1 http://http.kali.org/kali kali-rolling InRelease
1324 packages can be upgraded. Run 'apt list --upgradable' to see them.
The following packages were automatically installed and are no longer required:
  icu-devtools   libglapi-mesa   libpython3.12-minimal  python3-aioconsole     python
  libflac12t64   libicu-dev      libpython3.12-stdlib   python3-dunamai        python
  libfuse3-3     liblbfgsb0      libpython3.12t64       python3-nfsclient      python
  libgeos3.13.0  libpoppler145   libutempter0           python3-packaging-whl  python
Use 'sudo apt autoremove' to remove them.

Upgrading:
  7zip                                libcaja-extension1              libqt5webengin
  adduser                             libcamel-1.2-64t64              libqt5webengin
```

Restrict Apache file permissions
sudo chown -R root:root /var/www/html
sudo chmod -R 755 /var/www/html
ls -l /var/www/html



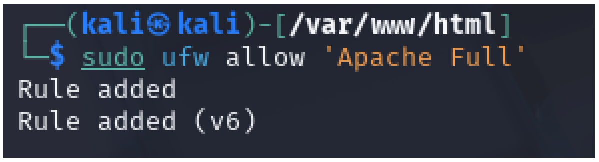
```
┌──(kali㉿kali)-[~]
└─$ sudo chown -R root:root /var/www/html
[sudo] password for kali:

┌──(kali㉿kali)-[~]
└─$ sudo chmod -R 755 /var/www/html

┌──(kali㉿kali)-[~]
└─$ ls -l /var/www/html
total 8
-rwxr-xr-x 1 root root 174 Jul 29 23:37 index.html
```

Enable and configure the UFW firewall
sudo apt install ufw -y



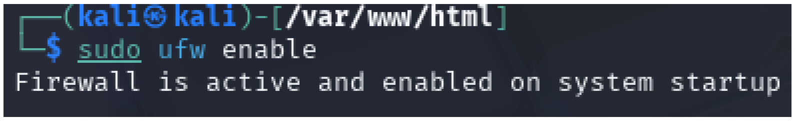
```
┌──(kali㉿kali)-[/var/www/html]
└─$ sudo apt install ufw -y
ufw is already the newest version (0.36.2-9).
The following packages were automatically installed and are no longer required:
  icu-devtools  libicu-dev  liblbfgsb0  python3.12-tk
Use 'sudo apt autoremove' to remove them.
```

sudo ufw allow 'Apache Full'



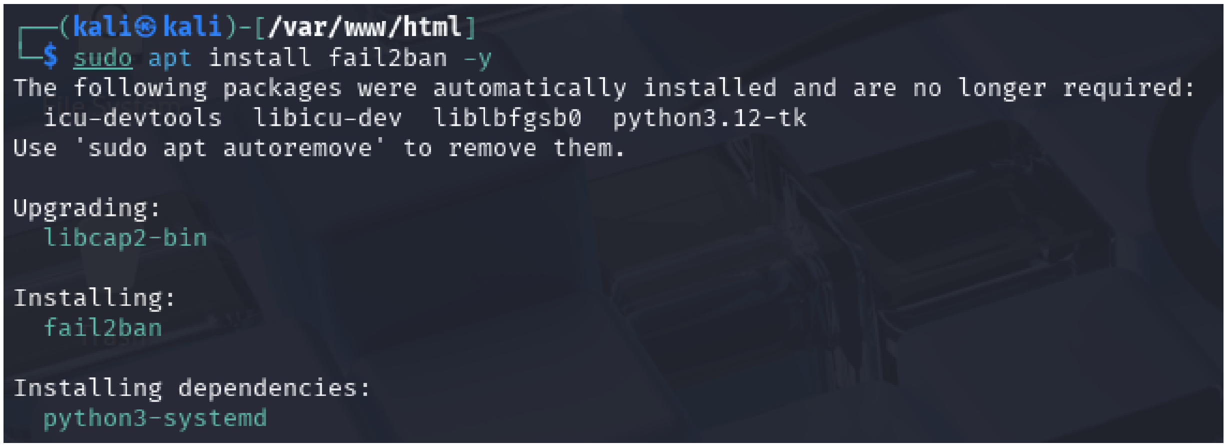
```
┌──(kali㉿kali)-[/var/www/html]
└─$ sudo ufw allow 'Apache Full'
Rule added
Rule added (v6)
```

sudo ufw enable



```
┌──(kali㉿kali)-[/var/www/html]
└─$ sudo ufw enable
Firewall is active and enabled on system startup
```

Install fail2ban to block brute-force attempts
sudo apt install fail2ban -y



```
┌──(kali㉿kali)-[/var/www/html]
└─$ sudo apt install fail2ban -y
The following packages were automatically installed and are no longer required:
  icu-devtools  libicu-dev  liblbfgsb0  python3.12-tk
Use 'sudo apt autoremove' to remove them.

Upgrading:
  libcap2-bin

Installing:
  fail2ban

Installing dependencies:
  python3-systemd
```

🧰 Tools & Skills Used

Tools used

- OS: Kali Linux or Ubuntu (or any Debian-based Linux VM)
- Packages: apache2
- Tools: grep, diff, history, log files

🧠 Reflection & Takeaways

This incident response simulation was a valuable, hands-on exercise that reinforced several critical cybersecurity principles. It demonstrated the full lifecycle of an incident, from initial investigation to final system hardening.

My key takeaways from this lab are:

1. **The Power of Foundational Tools:** A successful investigation doesn't always require complex forensic software. In this scenario, fundamental Linux command-line tools like history, cat, grep, and diff were sufficient to effectively investigate the incident, identify

the malicious changes, and confirm the root cause. This highlights the importance of mastering the basics.

2. **Logging is the Cornerstone of Incident Response:** The investigation would have been nearly impossible without access to the bash_history and Apache access.log files. This lab was a clear reminder that without comprehensive and accessible logs, a security analyst has no visibility into what occurred on a system, making effective incident response incredibly difficult.
3. **Recovery is Only as Good as Your Preparation:** The restoration of the website was simple and fast *only because* a backup of the original index.html file existed. This emphasizes that an effective recovery strategy depends entirely on proactive preparation, such as having a robust and regularly tested data backup plan.
4. **Security Doesn't End at Remediation:** Simply restoring the file is not enough. The final hardening steps—updating all packages, restricting file permissions, and enabling a firewall with fail2ban—are what truly secure the system against future attacks. This demonstrates the critical importance of moving from a reactive to a proactive security posture after an incident.