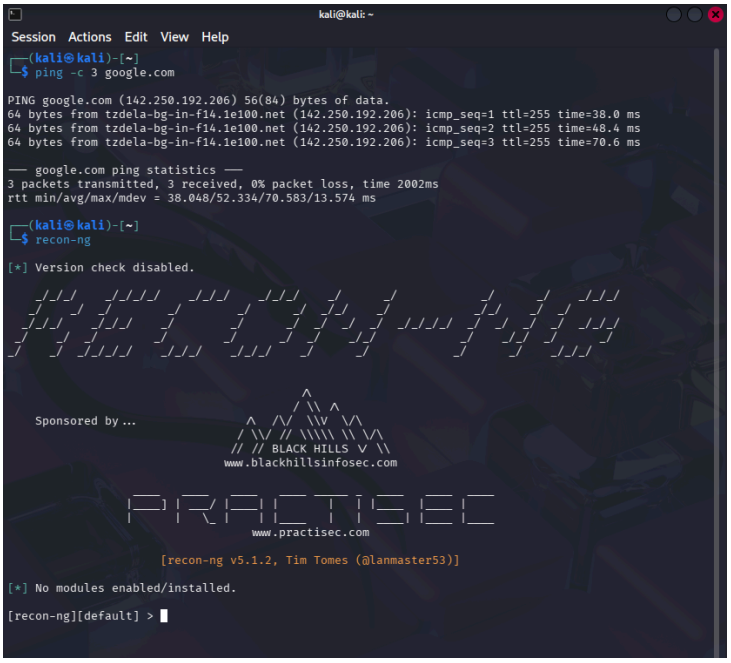
🧪 OSINT and Reconnaissance Lab Report

🎯 Aim

To perform Open Source Intelligence (OSINT) and reconnaissance by enumerating domains, subdomains, infrastructure, and exposed services using Recon-ng, Shodan, and Maltego in a Kali Linux virtual environment.

🛠️ Tools and Environment

- **Recon-ng**
- **Shodan**
- **Maltego**
- Kali Linux (Virtual Machine)
- Web browser (Firefox)



Part 1: Subdomain Enumeration using Recon-ng

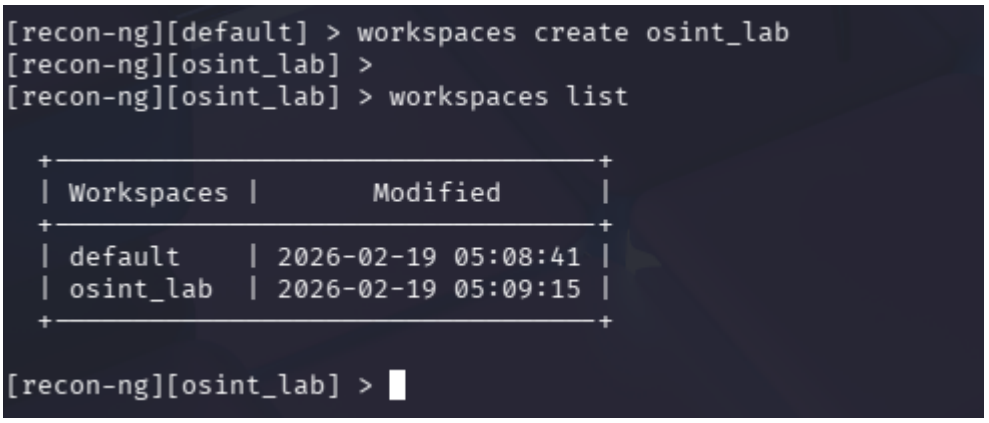
Procedure

- Recon-ng was launched inside Kali Linux using the terminal:

```
recon-ng
```

1. A new workspace was created to isolate lab data:

```
workspaces create osint_lab
```



2. The target domain was inserted into the Recon-ng database:

```
db insert domains

domain: example.com

notes: CEH OSINT lab
```

```
[recon-ng][osint_lab] > db insert domains
domain (TEXT): example.com
notes (TEXT): show domains
[*] 1 rows affected.
[recon-ng][osint_lab] > show domains

  +-------------------------------------------------------+
  | rowid |    domain   |    notes     |    module     |
  +-------------------------------------------------------+
  | 1     | example.com | show domains | user_defined |
  +-------------------------------------------------------+

[*] 1 rows returned
[recon-ng][osint_lab] >
```

3. The inserted domains were verified using:

```
show domains
```

```
[recon-ng][osint_lab] > db insert domains
domain (TEXT): example.com
notes (TEXT): CEH OSINT lab
[*] 1 rows affected.
[recon-ng][osint_lab] > show domains

  +-------------------------------------------------------+
  | rowid |    domain   |     notes     |    module     |
  +-------------------------------------------------------+
  | 1     | example.com | show domains  | user_defined |
  | 2     | google.com  | real test     | user_defined |
  | 3     | example.com | CEH OSINT lab | user_defined |
  +-------------------------------------------------------+

[*] 3 rows returned
[recon-ng][osint_lab] >
```

⚠️ Problem Encountered

Issue:

While attempting to load the module `recon/domains-hosts/crtsh`, Recon-ng repeatedly returned:

```
[!] Invalid module name.
```

Reason:

In newer versions of Recon-ng, some older or community-referenced modules such as `crtsh` are renamed, merged, or removed.

Resolution:

Used the command:

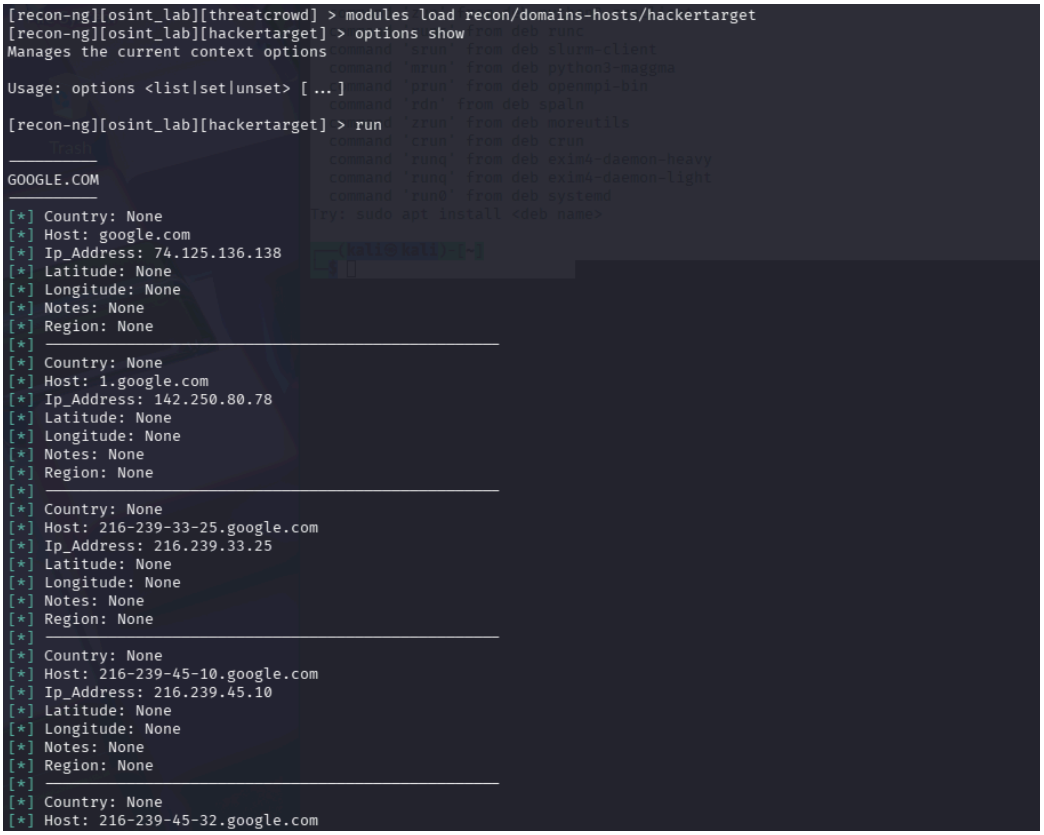
```
modules search domains-hosts
```

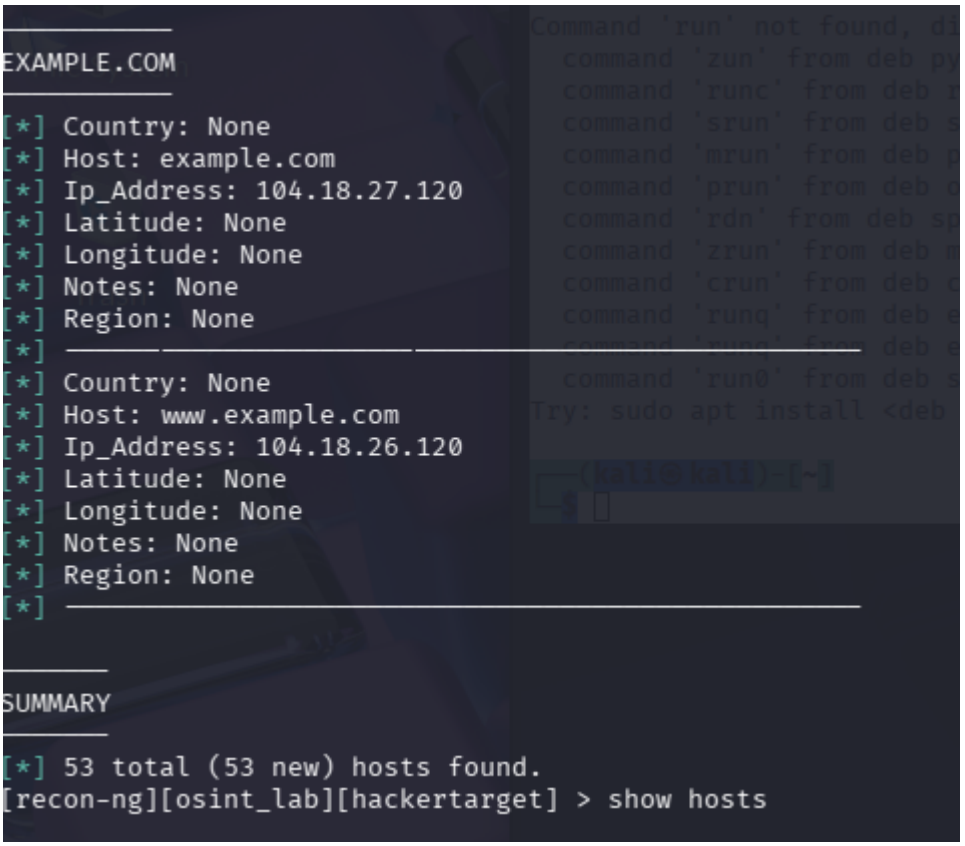
- Identified valid, installed modules such as:

  `recon/domains-hosts/bing_domain_web`

  `recon/domains-hosts/certificate_transparency`

  `recon/domains-hosts/hackertarget`

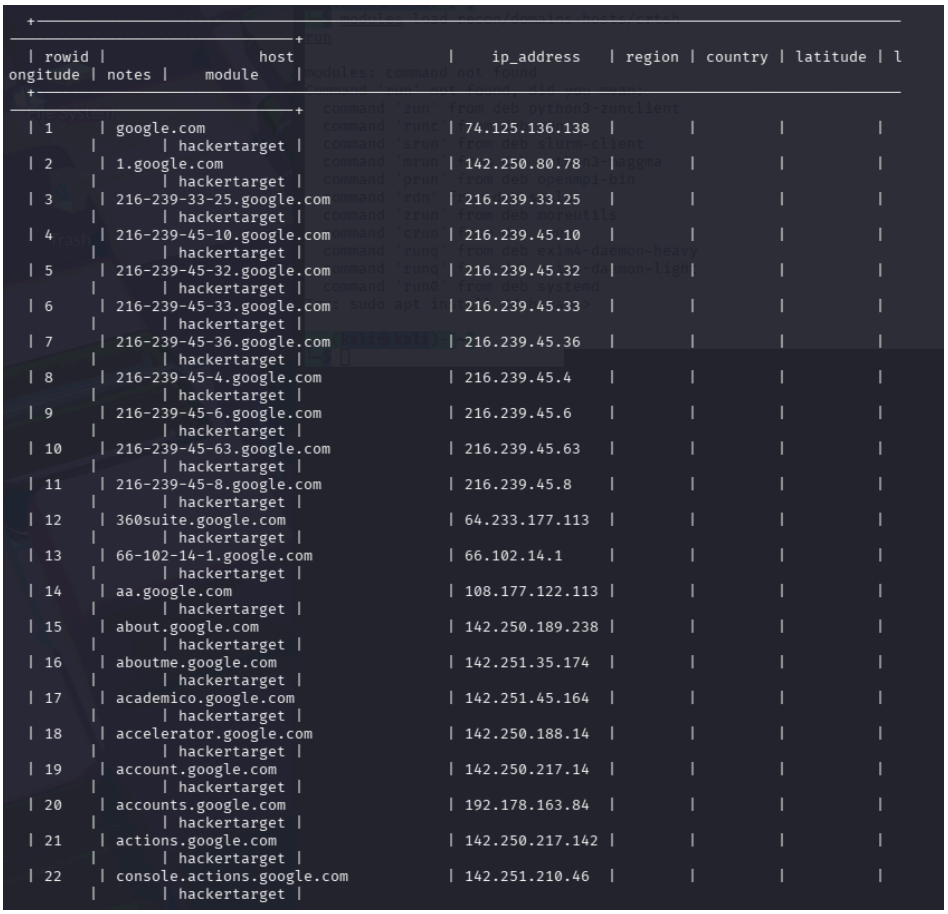




Module Execution

- The hackertarget module was loaded:

```
modules load recon/domains-hosts/hackertarget
```



rowid	host	ip_address	region	country	latitude	longitude	notes	module
1	google.com	74.125.136.138						hackertarget
2	1.google.com	142.250.80.78						hackertarget
3	216-239-33-25.google.com	216.239.33.25						hackertarget
4	216-239-45-10.google.com	216.239.45.10						hackertarget
5	216-239-45-32.google.com	216.239.45.32						hackertarget
6	216-239-45-33.google.com	216.239.45.33						hackertarget
7	216-239-45-36.google.com	216.239.45.36						hackertarget
8	216-239-45-4.google.com	216.239.45.4						hackertarget
9	216-239-45-6.google.com	216.239.45.6						hackertarget
10	216-239-45-63.google.com	216.239.45.63						hackertarget
11	216-239-45-8.google.com	216.239.45.8						hackertarget
12	360suite.google.com	64.233.177.113						hackertarget
13	66-102-14-1.google.com	66.102.14.1						hackertarget
14	aa.google.com	108.177.122.113						hackertarget
15	about.google.com	142.250.189.238						hackertarget
16	aboutme.google.com	142.251.35.174						hackertarget
17	academico.google.com	142.251.45.164						hackertarget
18	accelerator.google.com	142.250.188.14						hackertarget
19	account.google.com	142.250.217.14						hackertarget
20	accounts.google.com	192.178.163.84						hackertarget
21	actions.google.com	142.250.217.142						hackertarget
22	console.actions.google.com	142.251.210.46						hackertarget

5. The module was executed:

```
run
```

⚠️ Problem Encountered (During Execution)

Issue:

The module executed successfully but returned:

```
show hosts

[*] No data returned.
```

Reason:

- Bing limits automated scraping
- `example.com` is a reserved test domain with very limited real-world infrastructure
- Passive OSINT modules depend on third-party data availability

Resolution:

- Confirmed that the module ran correctly by checking generated URLs
- Accepted this as a **realistic OSINT limitation**
- Proceeded to other tools to complete reconnaissance

Outcome (Recon-ng)

Although no hosts were populated, the process demonstrated:

- Workspace management
- Database-driven reconnaissance
- Practical limitations of passive OSINT tools

Part 2: Exposed Services Enumeration using Shodan

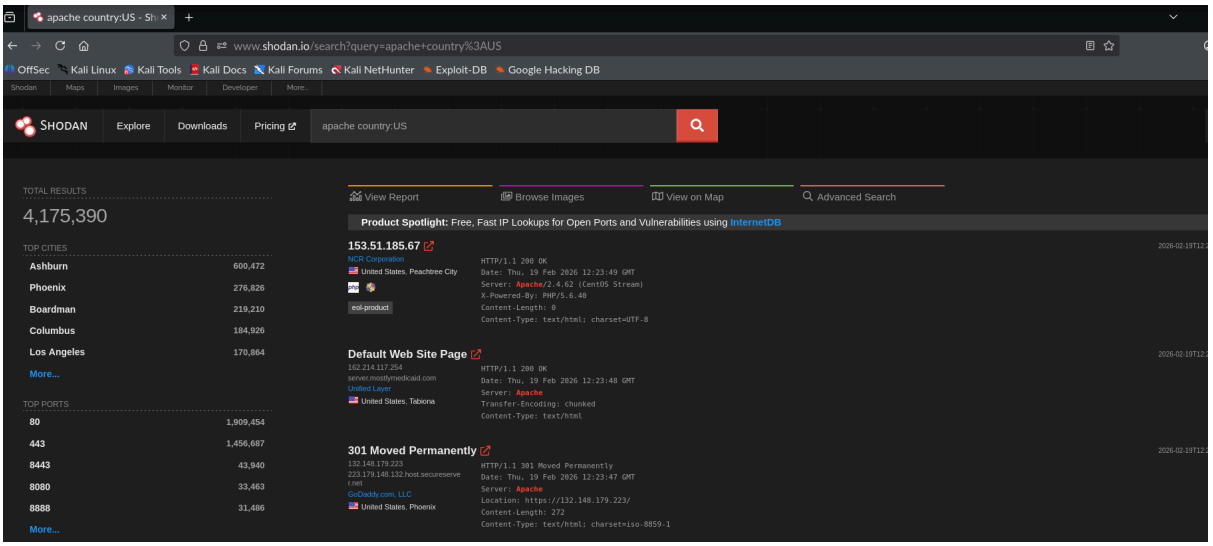
Objective

To identify exposed Apache web servers in the United States and analyze their publicly visible services and vulnerabilities.

Procedure

1. Shodan was accessed through a web browser.
2. The following search query was used:

```
apache country:US
```



⚠️ Problem Encountered

Issue:

Initially, Shodan displayed limited information and restricted host details.

Reason:

Shodan restricts detailed host data for unauthenticated users.

Resolution:

- Signed in using a free Shodan account
- Full host banners, ports, and vulnerabilities became visible

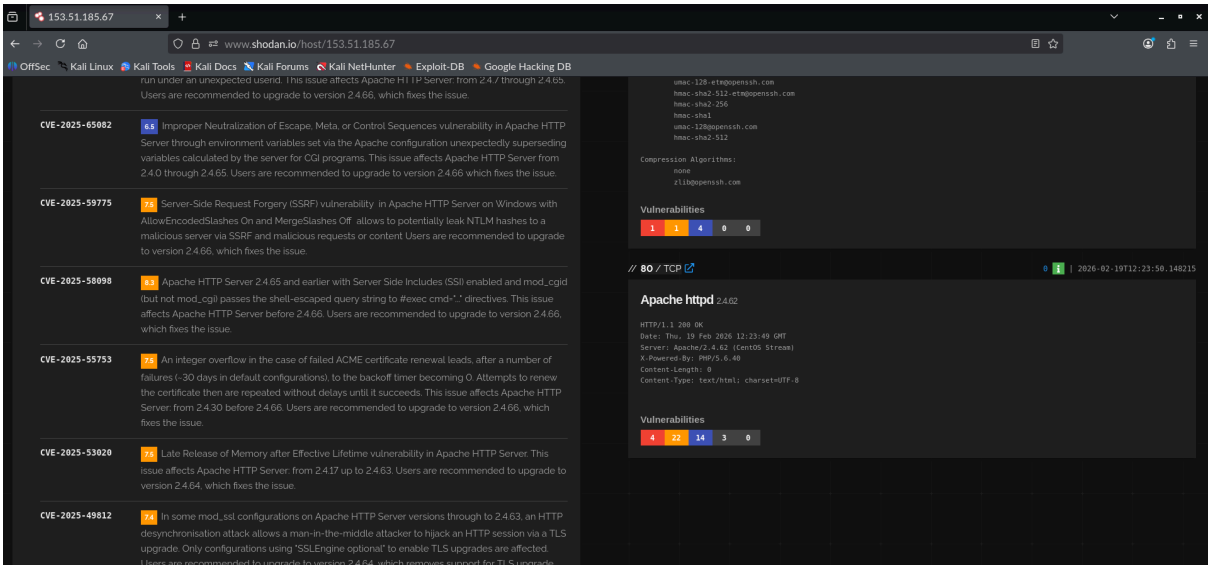
Observed Host

Attribute	Details
IP Address	153.51.185.67
Location	United States
Port	80/TCP
Web Server	Apache HTTP Server
Version	Apache/2.4.62
OS	CentOS Stream

Vulnerability Insights

Shodan displayed multiple CVEs associated with the Apache version running on the host. These included vulnerabilities related to:

- Server-side request forgery (SSRF)
- Improper input validation
- Memory management flaws
- Configuration weaknesses



⚠️ Problem Encountered (Ethical Concern)

Issue:
The presence of CVEs could be misinterpreted as permission to exploit the host.

Resolution:

- No exploitation was performed
- The lab strictly focused on **passive reconnaissance**
- Ethical disclaimer was documented clearly

Summary

Shodan was used to identify exposed Apache web servers in the United States. The analysis revealed publicly accessible HTTP services running Apache version 2.4.62 with several known vulnerabilities. Such exposed services increase attack surface and highlight the importance of proper patch management.

Part 3: Infrastructure Mapping using Maltego

Objective

To visually map domain relationships, DNS records, and infrastructure components.

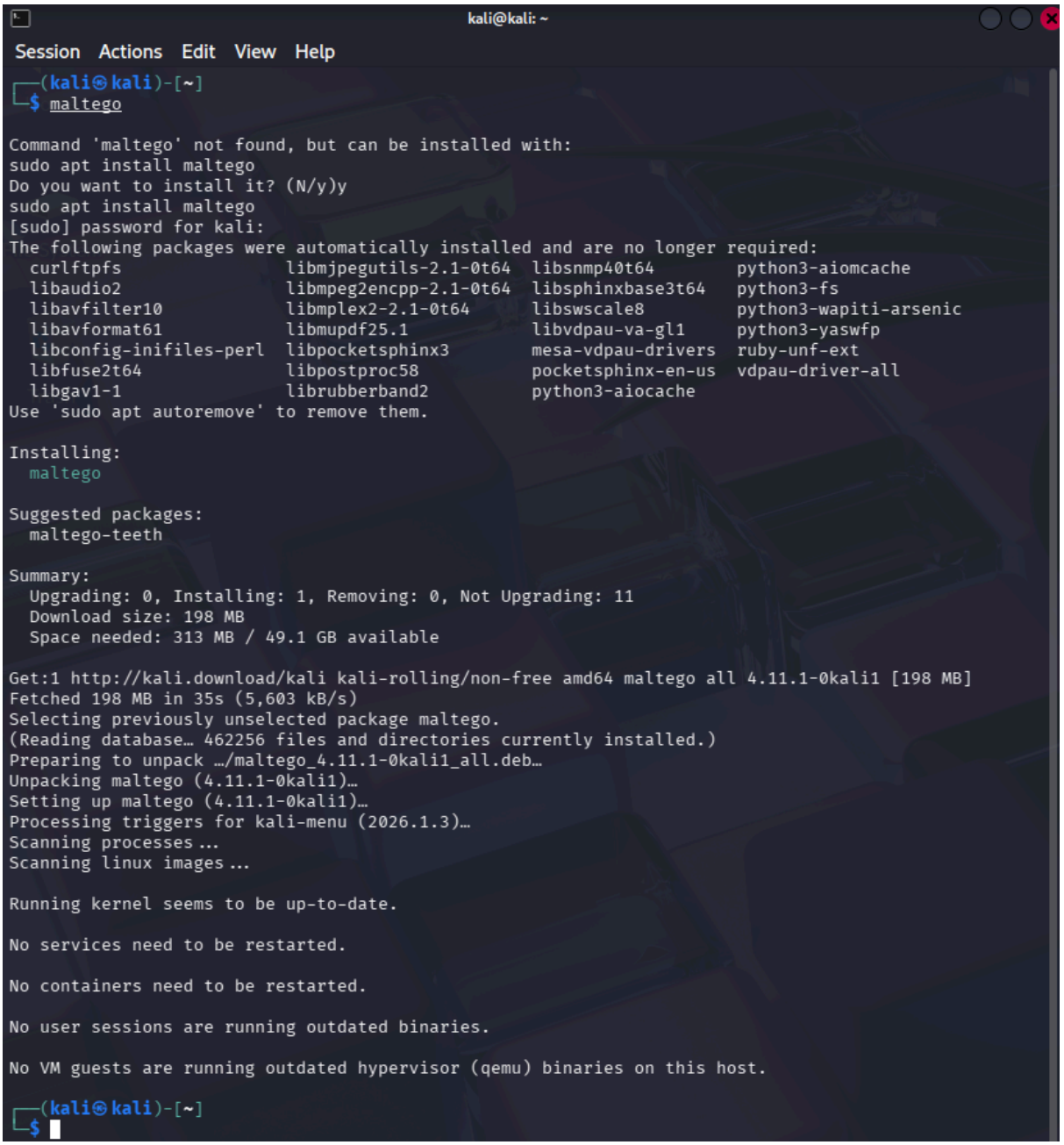
Procedure

1. Maltego was launched from Kali Linux.
2. Login was required using Maltego Community Edition credentials.
3. A new graph was created.

A **Domain** entity was added:

```
example.com
```

4. Standard transforms were executed:
    - To DNS Name
    - To IP Address
    - To Website
    - To Name Server



⚠️ Problem Encountered (At Startup)

Issue:

Transforms could not be executed without logging in.

Reason:

Maltego requires authentication even for Community Edition.

Resolution:

- Logged in using a free Maltego account
- Transforms executed successfully afterward

Outcome (Maltego)

Maltego generated a visual graph illustrating:

- Domain-to-DNS relationships
- IP address associations

- Infrastructure layout

This visualization helped correlate data collected from Recon-ng and Shodan.

Conclusion

This lab demonstrated the end-to-end OSINT reconnaissance workflow using industry-standard tools. Recon-ng provided a structured, database-driven approach to passive reconnaissance, Shodan revealed exposed services and vulnerabilities across the internet, and Maltego offered visual correlation of infrastructure data. The experiment also highlighted realistic challenges such as API restrictions, module availability issues, authentication requirements, and ethical boundaries, all of which are critical considerations in real-world cybersecurity operations.