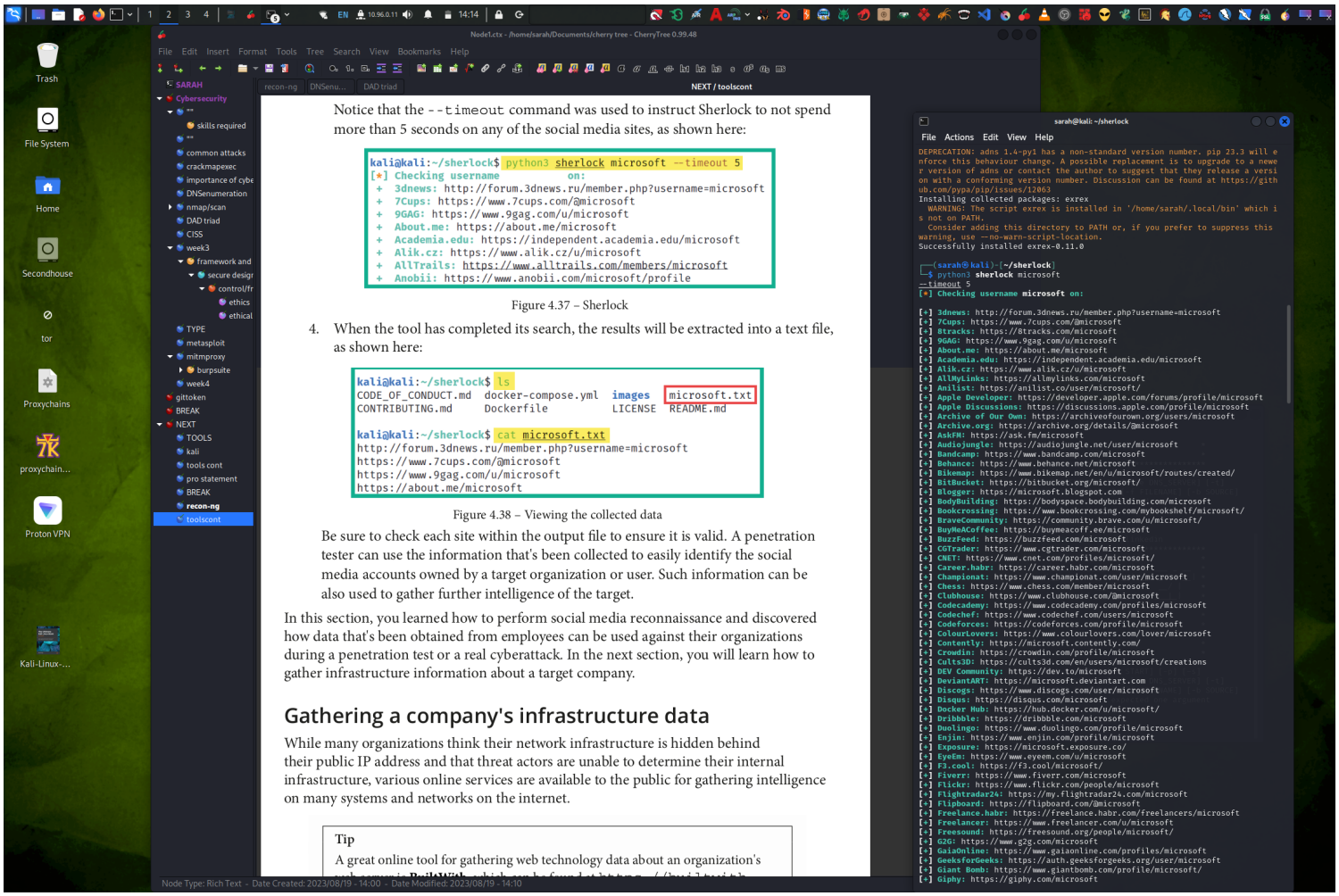


Notice that the `--timeout` command was used to instruct Sherlock to not spend more than 5 seconds on any of the social media sites, as shown here:

```
kali@kali:~/sherlock$ python3 sherlock microsoft --timeout 5
[*] Checking username      on:
 +  3dnews: http://forum.3dnews.ru/member.php?username=microsoft
 +  7Cups: https://www.7cups.com/@microsoft
 +  9GAG: https://www.9gag.com/u/microsoft
 +  About.me: https://about.me/microsoft
 +  Academia.edu: https://independent.academia.edu/microsoft
 +  Alik.cz: https://www.alik.cz/u/microsoft
 +  AllTrails: https://www.alltrails.com/members/microsoft
 +  Anobii: https://www.anobii.com/microsoft/profile
```

Figure 4.37 – Sherlock

4. When the tool has completed its search, the results will be extracted into a text file, as shown here:

```
kali@kali:~/sherlock$ ls
CODE_OF_CONDUCT.md  docker-compose.yml  images   microsoft.txt
CONTRIBUTING.md     Dockerfile          LICENSE  README.md

kali@kali:~/sherlock$ cat microsoft.txt
http://forum.3dnews.ru/member.php?username=microsoft
https://www.7cups.com/@microsoft
https://www.9gag.com/u/microsoft
https://about.me/microsoft
```

Figure 4.38 – Viewing the collected data

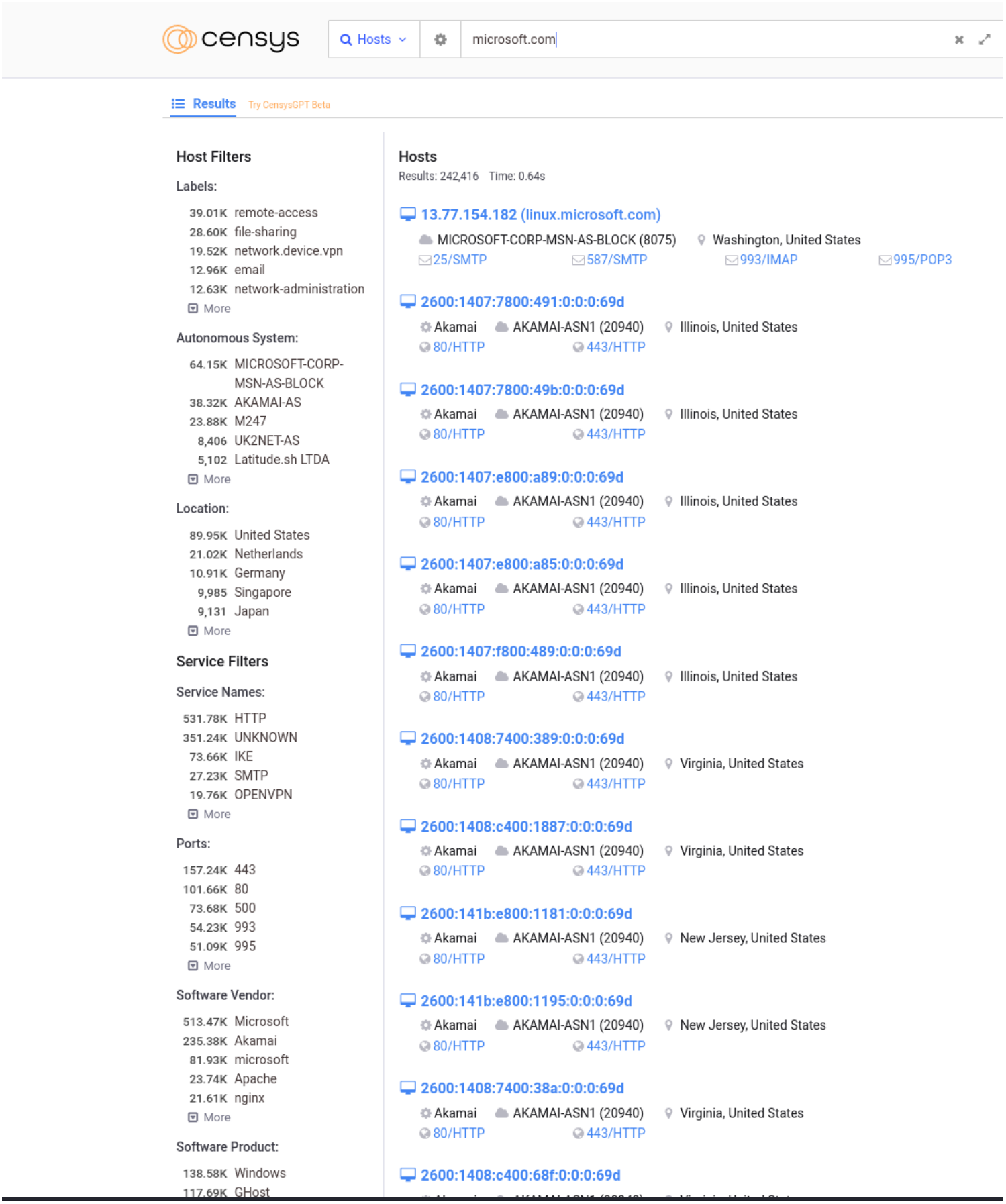
Be sure to check each site within the output file to ensure it is valid. A penetration tester can use the information that's been collected to easily identify the social media accounts owned by a target organization or user. Such information can be also used to gather further intelligence of the target.

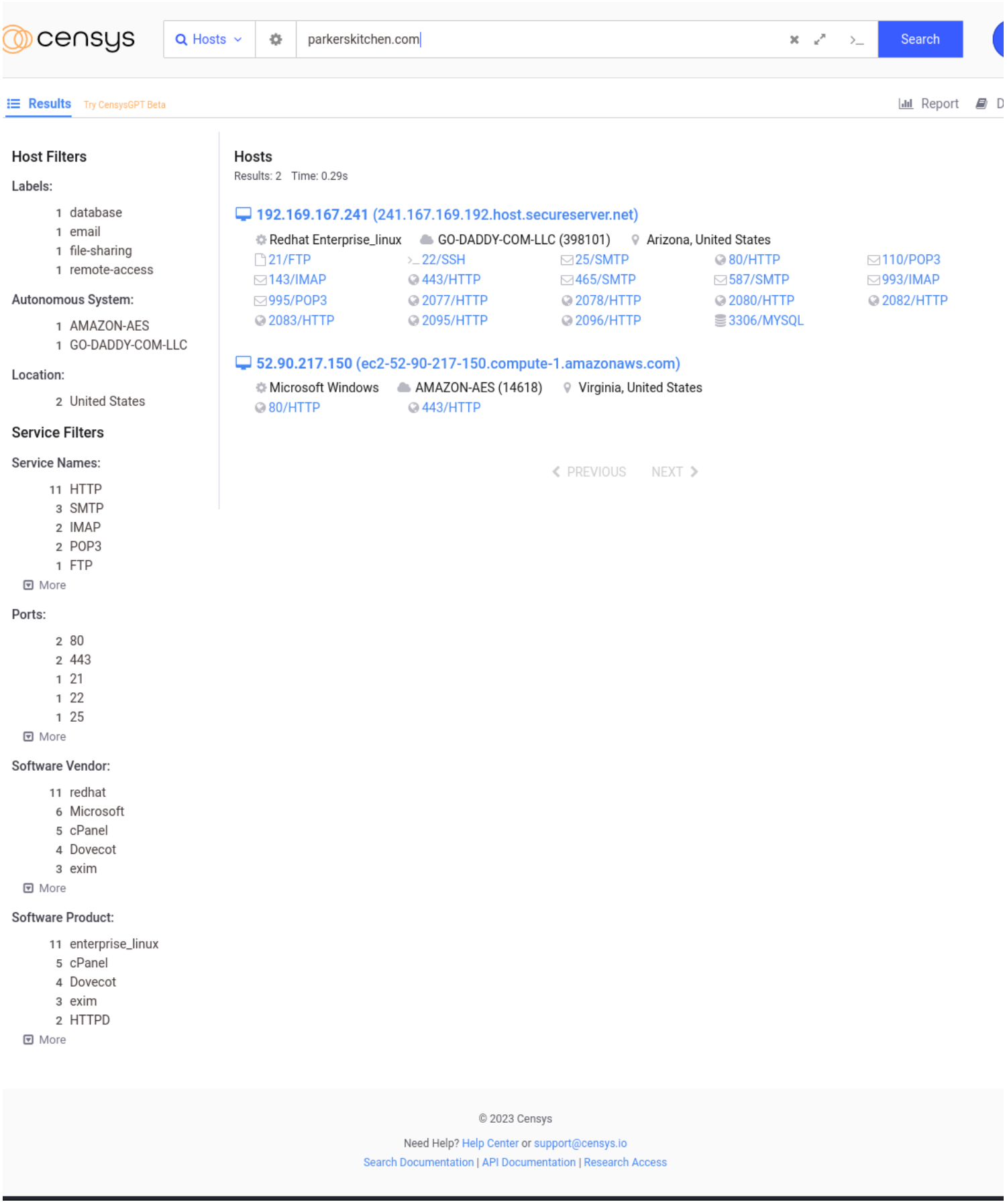
In this section, you learned how to perform social media reconnaissance and discovered how data that's been obtained from employees can be used against their organizations during a penetration test or a real cyberattack. In the next section, you will learn how to gather infrastructure information about a target company.

Gathering a company's infrastructure data

While many organizations think their network infrastructure is hidden behind their public IP address and that threat actors are unable to determine their internal infrastructure, various online services are available to the public for gathering intelligence on many systems and networks on the internet.

> **Tip**
>
> A great online tool for gathering web technology data about an organization's


censys
Hosts
microsoft.com
Results
Try CensysGPT Beta
Host Filters
Labels:
39.01K remote-access
28.60K file-sharing
19.52K network.device.vpn
12.96K email
12.63K network-administration
More
Autonomous System:
64.15K MICROSOFT-CORP-MSN-AS-BLOCK
38.32K AKAMAI-AS
23.88K M247
8,406 UK2NET-AS
5,102 Latitude.sh LTDA
More
Location:
89.95K United States
21.02K Netherlands
10.91K Germany
9,985 Singapore
9,131 Japan
More
Service Filters
Service Names:
531.78K HTTP
351.24K UNKNOWN
73.66K IKE
27.23K SMTP
19.76K OPENVPN
More
Ports:
157.24K 443
101.66K 80
73.68K 500
54.23K 993
51.09K 995
More
Software Vendor:
513.47K Microsoft
235.38K Akamai
81.93K microsoft
23.74K Apache
21.61K nginx
More
Software Product:
138.58K Windows
117.69K GHost
Hosts
Results: 242,416 Time: 0.64s
13.77.154.182 (linux.microsoft.com)
MICROSOFT-CORP-MSN-AS-BLOCK (8075)
Washington, United States
25/SMTP
587/SMTP
993/IMAP
995/POP3
2600:1407:7800:491:0:0:0:69d
Akamai
AKAMAI-ASN1 (20940)
Illinois, United States
80/HTTP
443/HTTP
2600:1407:7800:49b:0:0:0:69d
Akamai
AKAMAI-ASN1 (20940)
Illinois, United States
80/HTTP
443/HTTP
2600:1407:e800:a89:0:0:0:69d
Akamai
AKAMAI-ASN1 (20940)
Illinois, United States
80/HTTP
443/HTTP
2600:1407:e800:a85:0:0:0:69d
Akamai
AKAMAI-ASN1 (20940)
Illinois, United States
80/HTTP
443/HTTP
2600:1407:f800:489:0:0:0:69d
Akamai
AKAMAI-ASN1 (20940)
Illinois, United States
80/HTTP
443/HTTP
2600:1408:7400:389:0:0:0:69d
Akamai
AKAMAI-ASN1 (20940)
Virginia, United States
80/HTTP
443/HTTP
2600:1408:c400:1887:0:0:0:69d
Akamai
AKAMAI-ASN1 (20940)
Virginia, United States
80/HTTP
443/HTTP
2600:141b:e800:1181:0:0:0:69d
Akamai
AKAMAI-ASN1 (20940)
New Jersey, United States
80/HTTP
443/HTTP
2600:141b:e800:1195:0:0:0:69d
Akamai
AKAMAI-ASN1 (20940)
New Jersey, United States
80/HTTP
443/HTTP
2600:1408:7400:38a:0:0:0:69d
Akamai
AKAMAI-ASN1 (20940)
Virginia, United States
80/HTTP
443/HTTP
2600:1408:c400:68f:0:0:0:69d


censys
Hosts
parkerskitchen.com
Search
Results
Try CensysGPT Beta
Report
Host Filters
Labels:
1 database
1 email
1 file-sharing
1 remote-access
Autonomous System:
1 AMAZON-AES
1 GO-DADDY-COM-LLC
Location:
2 United States
Service Filters
Service Names:
11 HTTP
3 SMTP
2 IMAP
2 POP3
1 FTP
More
Ports:
2 80
2 443
1 21
1 22
1 25
More
Software Vendor:
11 redhat
6 Microsoft
5 cPanel
4 Dovecot
3 exim
More
Software Product:
11 enterprise_linux
5 cPanel
4 Dovecot
3 exim
2 HTTPD
More
Hosts
Results: 2 Time: 0.29s
192.169.167.241 (241.167.169.192.host.secureserver.net)
Redhat Enterprise_linux
GO-DADDY-COM-LLC (398101)
Arizona, United States
21/FTP
22/SSH
25/SMTP
80/HTTP
110/POP3
143/IMAP
443/HTTP
465/SMTP
587/SMTP
993/IMAP
995/POP3
2077/HTTP
2078/HTTP
2080/HTTP
2082/HTTP
2083/HTTP
2095/HTTP
2096/HTTP
3306/MYSQL
52.90.217.150 (ec2-52-90-217-150.compute-1.amazonaws.com)
Microsoft Windows
AMAZON-AES (14618)
Virginia, United States
80/HTTP
443/HTTP
PREVIOUS
NEXT
© 2023 Censys
Need Help? Help Center or support@censys.io
Search Documentation | API Documentation | Research Access

```
kali@kali:~/sherlock$ ls
CODE_OF_CONDUCT.md  docker-compose.yml  images   microsoft.txt
CONTRIBUTING.md     Dockerfile          LICENSE  README.md

kali@kali:~/sherlock$ cat microsoft.txt
http://forum.3dnews.ru/member.php?username=microsoft
https://www.7cups.com/@microsoft
https://www.9gag.com/u/microsoft
https://about.me/microsoft
```

Figure 4.38 – Viewing the collected data

Be sure to check each site within the output file to ensure it is valid. A penetration tester can use the information that's been collected to easily identify the social media accounts owned by a target organization or user. Such information can be also used to gather further intelligence of the target.

In this section, you learned how to perform social media reconnaissance and discovered how data that's been obtained from employees can be used against their organizations during a penetration test or a real cyberattack. In the next section, you will learn how to gather infrastructure information about a target company.

Gathering a company's infrastructure data

While many organizations think their network infrastructure is hidden behind their public IP address and that threat actors are unable to determine their internal infrastructure, various online services are available to the public for gathering intelligence on many systems and networks on the internet.

> **Tip**
>
> A great online tool for gathering web technology data about an organization's web server is **BuiltWith**, which can be found at `https://builtwith.com`.

Over the next few sections, you will learn how to gather the infrastructure data about a target organization using various online sources and tools.

Shodan

Shodan is a search engine for **Internet of Things** (**IoT**), systems, and networks that are directly connected to the internet. Ethical hackers, penetration testers, and even threat actors use Shodan to identify their organization's or target's assets, and they check whether they have been publicly exposed on the internet. This online tool helps cybersecurity professionals quickly determine whether their organization's assets have been exposed on the internet.

```
kali@kali:~/sherlock$ ls
CODE_OF_CONDUCT.md  docker-compose.yml  images   microsoft.txt
CONTRIBUTING.md     Dockerfile          LICENSE  README.md

kali@kali:~/sherlock$ cat microsoft.txt
http://forum.3dnews.ru/member.php?username=microsoft
https://www.7cups.com/@microsoft
https://www.9gag.com/u/microsoft
https://about.me/microsoft
```

Figure 4.38 – Viewing the collected data

Be sure to check each site within the output file to ensure it is valid. A penetration tester can use the information that's been collected to easily identify the social media accounts owned by a target organization or user. Such information can be also used to gather further intelligence of the target.

In this section, you learned how to perform social media reconnaissance and discovered how data that's been obtained from employees can be used against their organizations during a penetration test or a real cyberattack. In the next section, you will learn how to gather infrastructure information about a target company.

Gathering a company's infrastructure data

While many organizations think their network infrastructure is hidden behind their public IP address and that threat actors are unable to determine their internal infrastructure, various online services are available to the public for gathering intelligence on many systems and networks on the internet.

> **Tip**
>
> A great online tool for gathering web technology data about an organization's web server is **BuiltWith**, which can be found at `https://builtwith.com`.

Over the next few sections, you will learn how to gather the infrastructure data about a target organization using various online sources and tools.

Shodan

Shodan is a search engine for **Internet of Things** (**IoT**), systems, and networks that are directly connected to the internet. Ethical hackers, penetration testers, and even threat actors use Shodan to identify their organization's or target's assets, and they check whether they have been publicly exposed on the internet. This online tool helps cybersecurity professionals quickly determine whether their organization's assets have been exposed on the internet.

To provide some additional insight, imagine that you want to determine whether your organization has any systems, such as servers that are accessible over the internet. These servers may include open service ports, vulnerable running applications, and services. Imagine that your organization has a legacy system running an older operating system that isn't patched with the latest security updates from the vendor and is directly connected to the internet. A penetration tester or threat actor can use an online tool, such as Shodan, to discover such systems without even sending a probe of any kind directly from the penetration tester's system to the target server, simply because Shodan detects it automatically.

> **Important Note**
>
> Shodan provides limited searches using a free account on their website at `www.shodan.io`. However, it's recommended to have a paid account to unlock all the features, services, and advanced functions of the platform.

To get started with Shodan, please use the following instructions:

1. Using your web browser, go to `https://www.shodan.io/` and create an account.
2. Once you're logged in, use the search bar to perform a search for *windows server 2008*. The following screenshot shows the results:

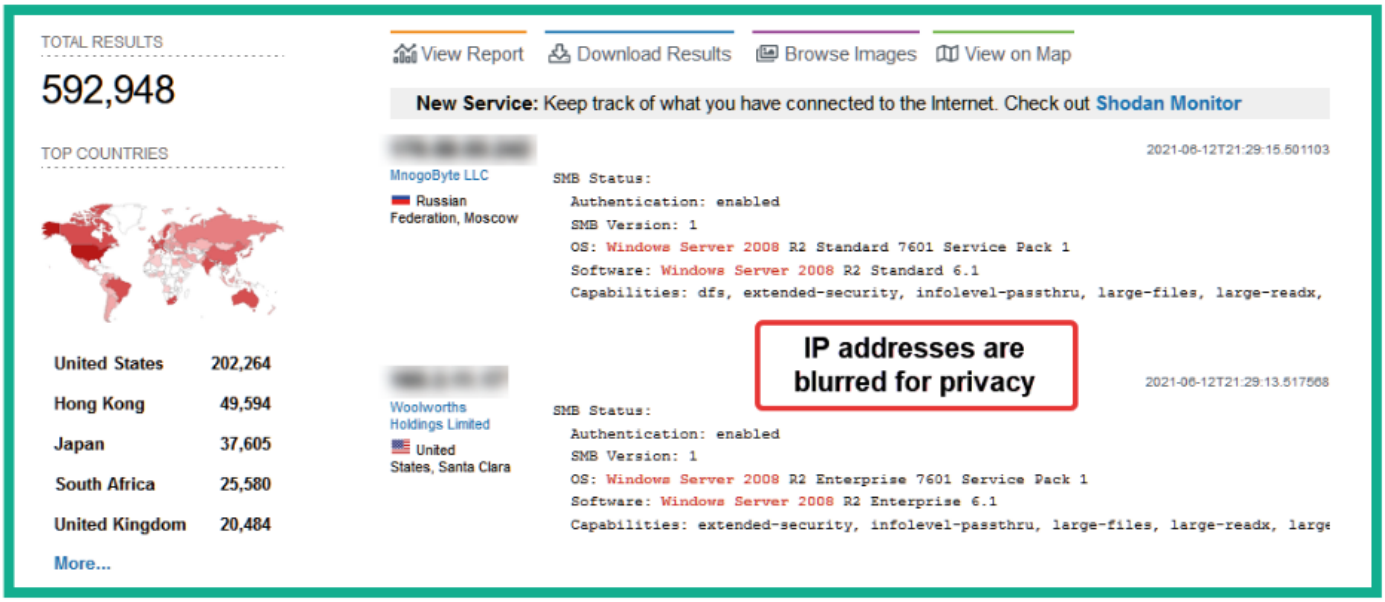


Figure 4.39 – Searching for a specific operating system

As shown in the preceding screenshot, over 500,000 systems have been discovered by Shodan running Windows Server 2008 and they are all directly connected to the internet. A penetration tester or threat actor can simply use Google to search for vulnerabilities on this operating system and find working exploits.

3. Clicking on one of these systems will provide additional information, such as open service port numbers, running services, and the banner of each running service, as shown here:

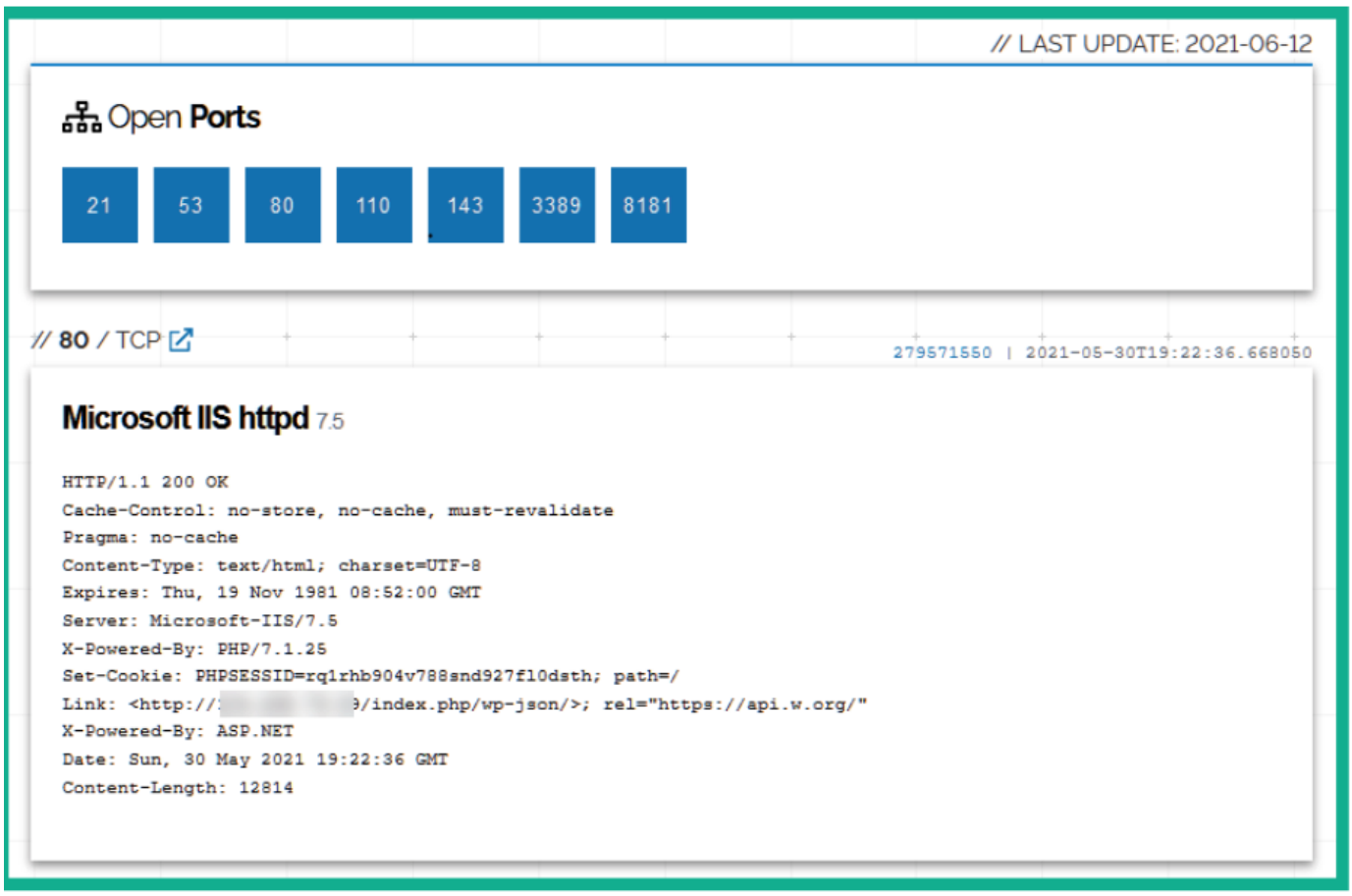


Figure 4.40 – Discovering open ports

Open service ports are doorways to a target system, and they indicate whether services are running on the device. As shown in the preceding screenshot, the following can be determined by a cybersecurity professional and a threat actor:

- Port `21`: There's a **File Transfer Protocol** (**FTP**) server.
- Port `53`: This system is providing **Domain Name System** (**DNS**) services
- Port `80`: There's a web server on this device.
- Port `110`: This device is providing **Post Office Protocol 3** (**POP3**) services for email clients.
- Port `143`: This system is running **Internet Message Access Protocol 4** (**IMAP4**) services for email clients.
- Port `3389`: Microsoft **Remote Desktop Protocol** (**RDP**) operates on this port by default, which means RDP is currently active.
- Port `8181`: Provides email services over this port.

As a penetration tester, these are various points of entry into the system, and each application that is providing these services may contain a known vulnerability that can then be exploited to compromise the system.

4. Additionally, if Shodan detects vulnerabilities on a system, it will provide the following details:

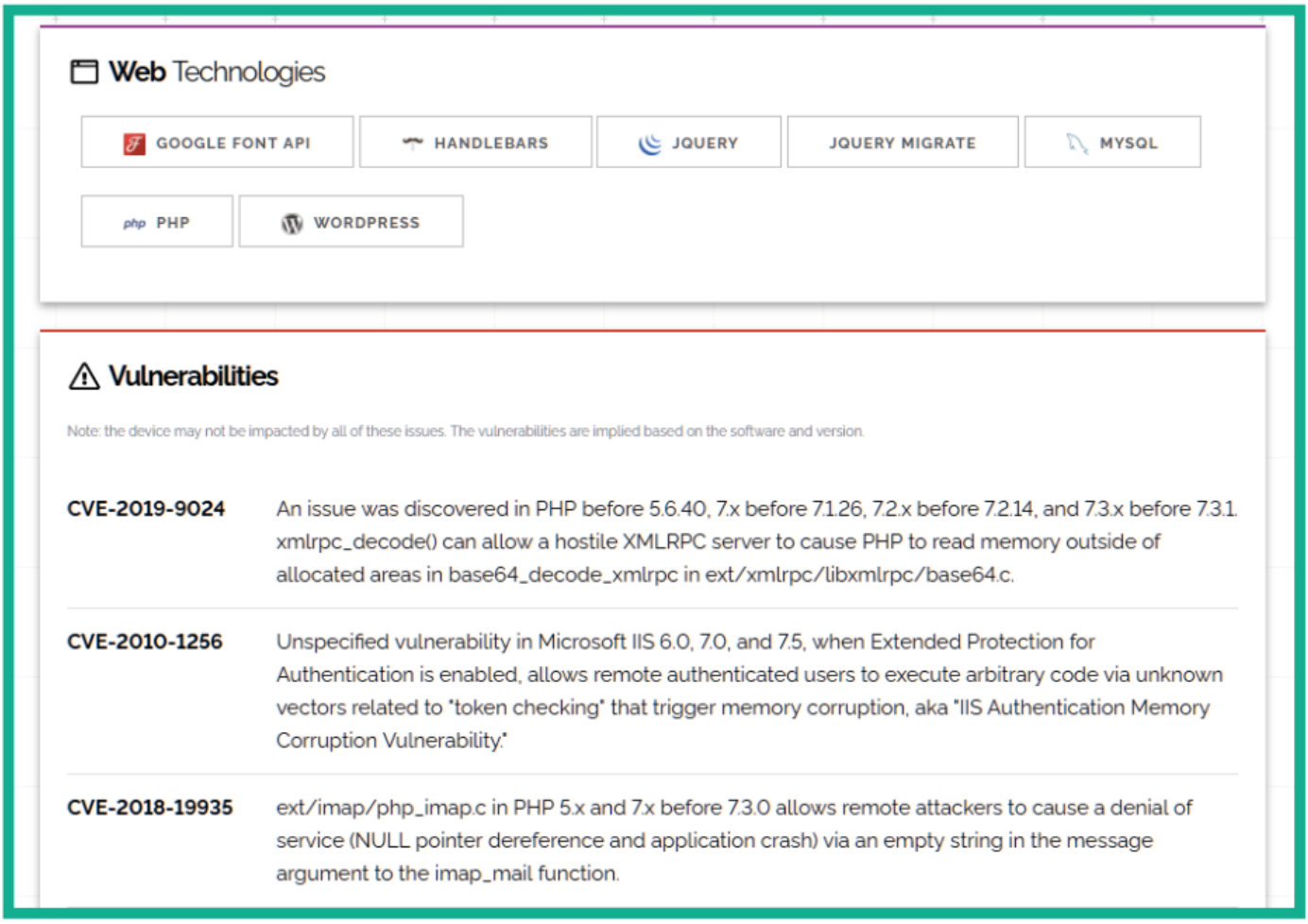


Figure 4.41 – Web technologies and vulnerabilities

As shown in the preceding screenshot, notice how Shodan provides a list of web technologies running on the target server. A penetration tester or threat actor can use this information to research for vulnerabilities and exploits that can be used to compromise the target system. Additionally, Shodan provides a list of known vulnerabilities and their reference **Common Vulnerabilities and Exposure** (**CVE**) IDs, along with brief descriptions of them.

> **Important Note**
>
> Cybersecurity professionals report newly discovered vulnerabilities at `https://cve.mitre.org`, which provides a centralized vulnerability disclosure database for like-minded people and organizations within the cybersecurity industry.

The following screenshot shows details about a running service of a Windows Server 2008 system found on Shodan. Notice that it is running **Server Message Block** (**SMB**) version 1:

Figure 4.42 – Discovering running services

A penetration tester can perform a simple Google search such as *Windows Server 2008 smb vulnerability* or *Windows Server 2008 smb exploit* to quickly get information about known security flaws on this target, as well as possible ways to exploit the target's security weaknesses.

With that, you have seen how simple and easy it is to gather a target organization's infrastructure details without having to place any type of network implants into the company's network. Shodan can help you gather OSINT data about your targets without you having to directly engage a target. In the next section, you will discover how to use another very well-known tool within the industry to gather in-depth intelligence on systems on the internet.

Censys

Censys can gather intelligence on any publicly accessible system or network on the internet. To start gathering data about a target, use the following instructions:

1. First, you will need to register for a free account at `https://censys.io/register`.
2. Next, go to `https://censys.io/login` and log in with your new credentials.

3. You will be presented with a search bar. Simply enter the IP address of a target web server, as shown here:

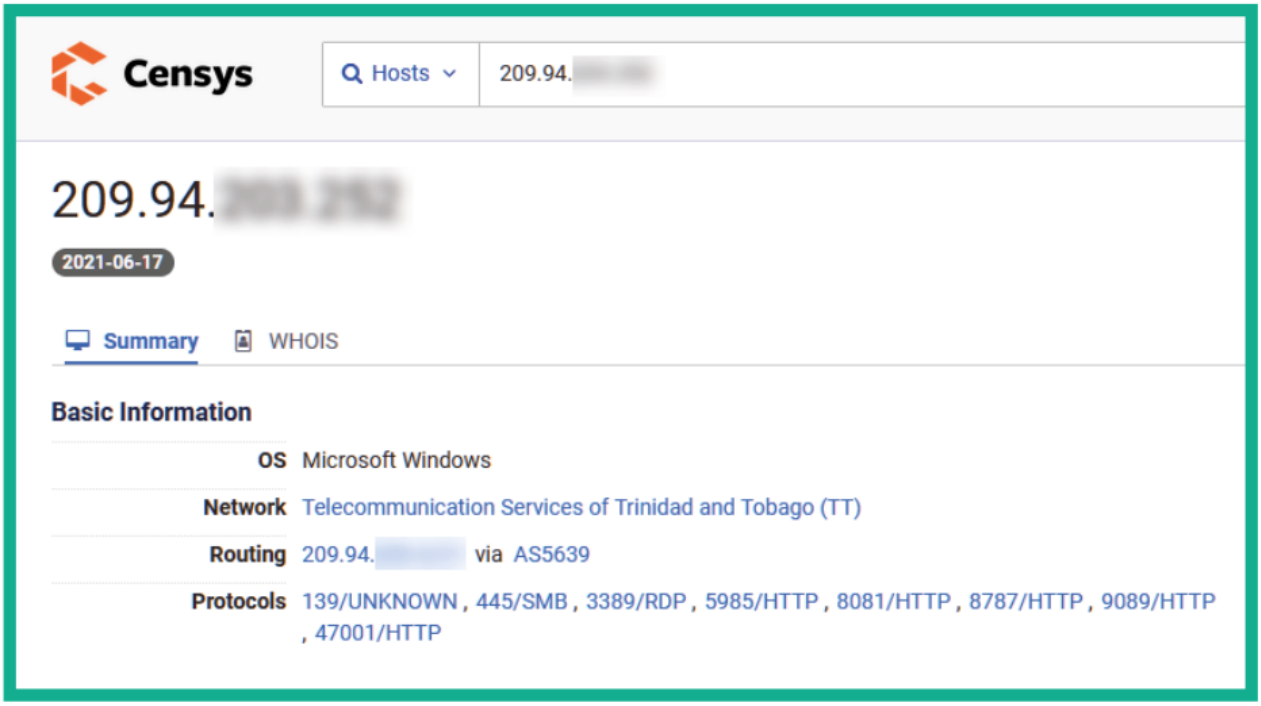


Figure 4.43 – Censys results

As shown in the preceding screenshot, Censys can provide a lot of information about the target, such as the running operating systems, services, open service port numbers, and much more.

Using the information gathered from Censys, a penetration tester can create a profile of which systems are publicly accessible through the internet and the open services ports. Such information can be used to perform research on vulnerabilities and techniques to exploit those security weaknesses. In the next section, you will learn how to use a very awesome Python-based tool to assist in performing passive information-gathering techniques and acquiring OSINT data.

Maltego

Maltego is a graphical open source intelligence tool that was created by Paterva and is now maintained by Maltego Technologies. This tool helps ethical hackers and penetration testers quickly gather an organization's infrastructure data by using a graphical interactive data mining application. This application can query and gather information from various sources on the internet and present data in easy-to-read graphs. These graphs provide visualizations of the relationships between each entity and the target.

more

6. When Maltego loads, you will be required to choose an option from the **Product Selection** window. Select **Maltego CE (Free)** and click on **Run**, as shown here:

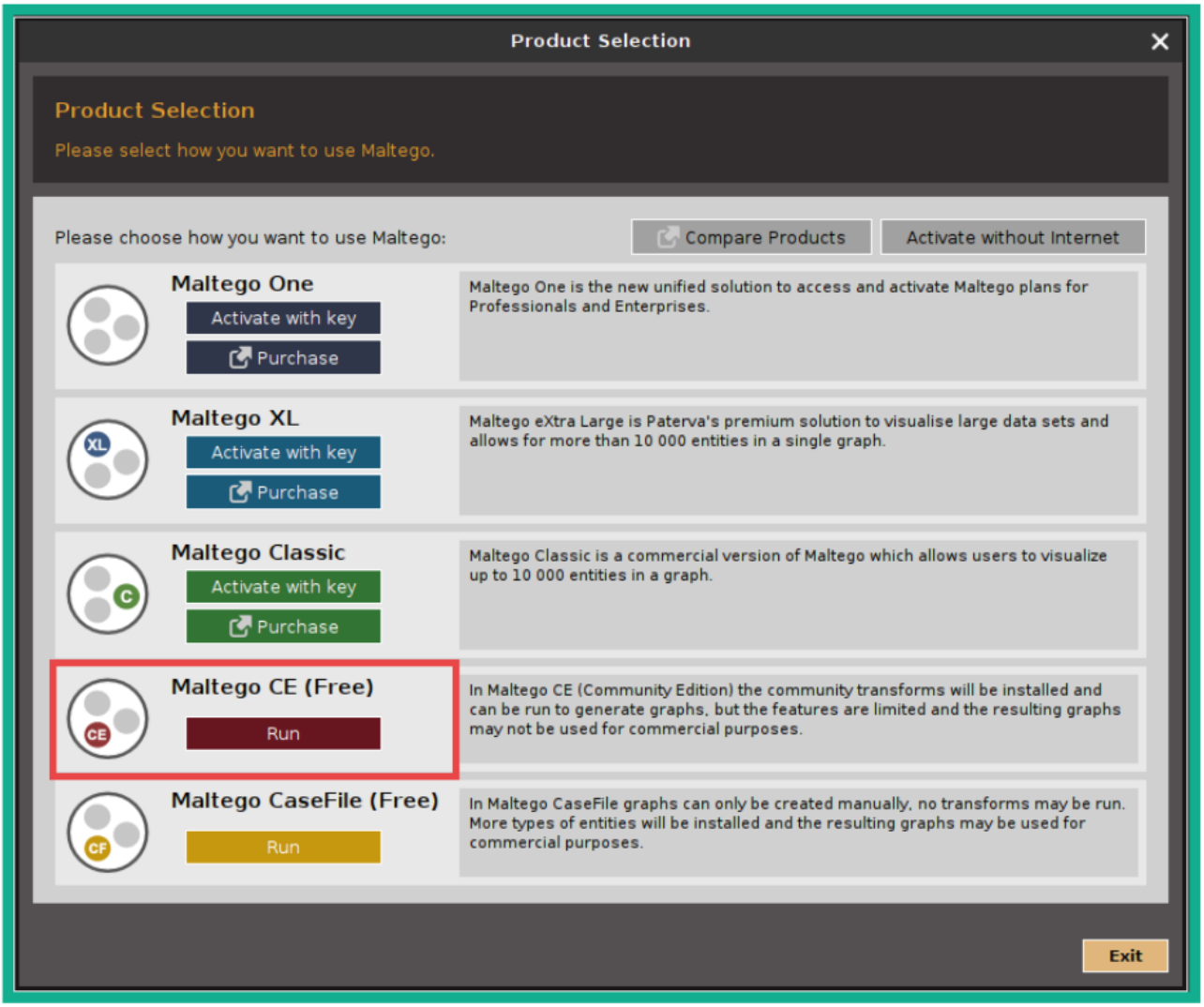


Figure 4.45 – Maltego – Production Selection page

7. You will be required to accept the license agreement using your Maltego account details from *Step 2*. Select a default browser such as Firefox during the initial setup process.

8. To start gathering information on a target organization, you must open a new graph. To do this, click on the **Maltego** icon in the top-left corner, and then click on **New**. Once a new graph has been created, you'll see various types of information (entities) on the left, while on the right-hand side, you'll see **Overview**, **Detail View**, and **Property View**, as shown here:

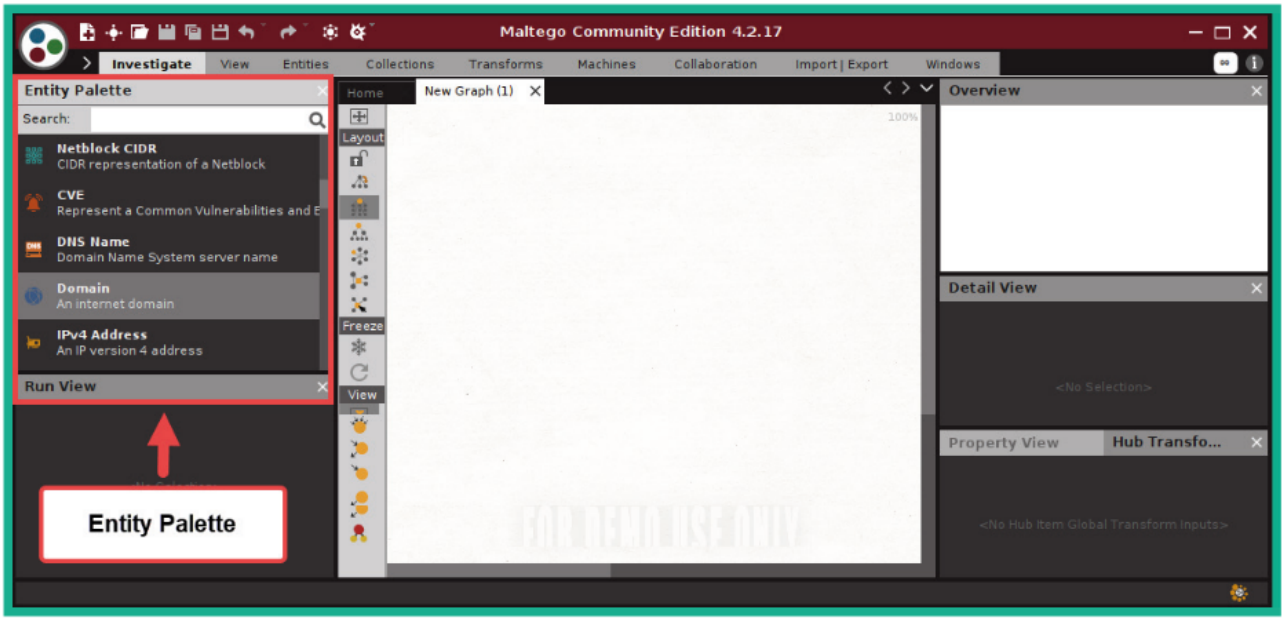


Figure 4.46 – Maltego's user interface

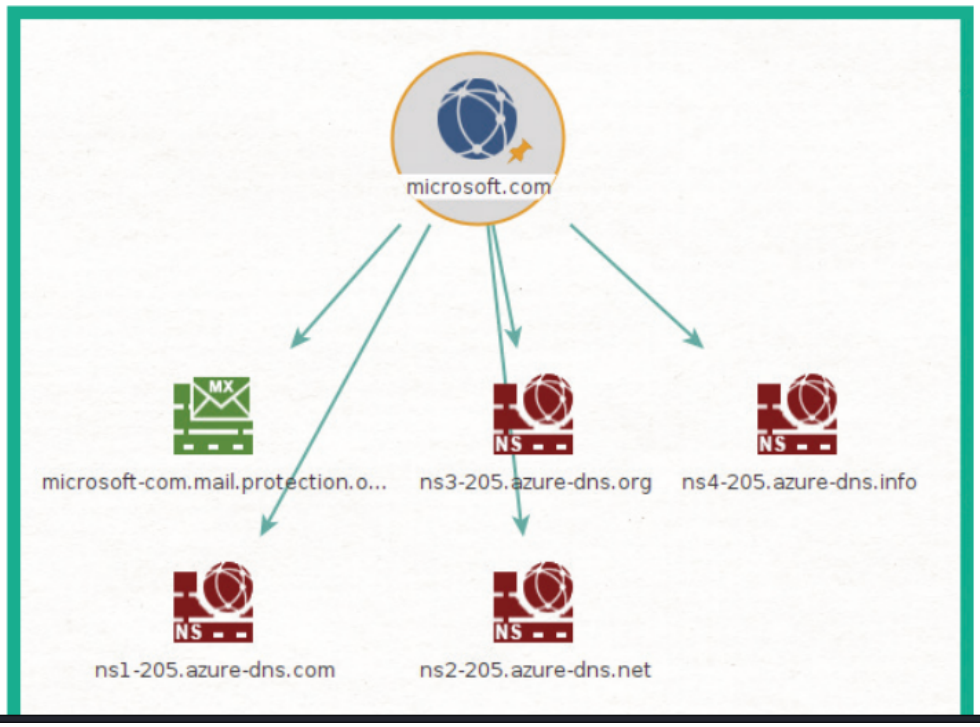
9. To start gathering a target's infrastructure details, from the **Entity Palette** section, drag and drop the **Domain** entity into the middle of the graph.
10. Next, double-click on the **Domain** entity and change the domain name to your target's domain name. For this exercise, I will be gathering OSINT on Microsoft.
11. To gather the **Domain Name System** (**DNS**) information about the domain, right-click on **Domain entity** and select **DNS from Domain** > **To DNS Name – MX (mail server)**. The following screenshot shows that Maltego was able to find Microsoft's email server:

Figure 4.47 – Discovering mail servers

12. To get the IP addresses of an object, such as the email server, right-click on the email server entity and select **Resolve to IP**. The following screenshot shows the IP address associated with the target's email server:

Figure 4.48 – Gathering the IP addresses of assets

13. To discover the **Name Server** (**NS**) of a target domain, right-click on **Domain Entity** > **DNS from domain** > **To DNS Name – NS** (**name server**). The following screenshot shows the NS servers associated with the parent domain:



14. To gather website information about the target domain, right-click on the **Domain** entity and select **DNS from domain** > **To Website** (**Quick lookup**). This will allow you to discover the target's website address.
15. To get a list of all the web links for the target's website, right-click on the **Website** entity and select **Links in and out of site**. The following screenshot shows the links that were found on the website, such as additional subdomains owned by the target:

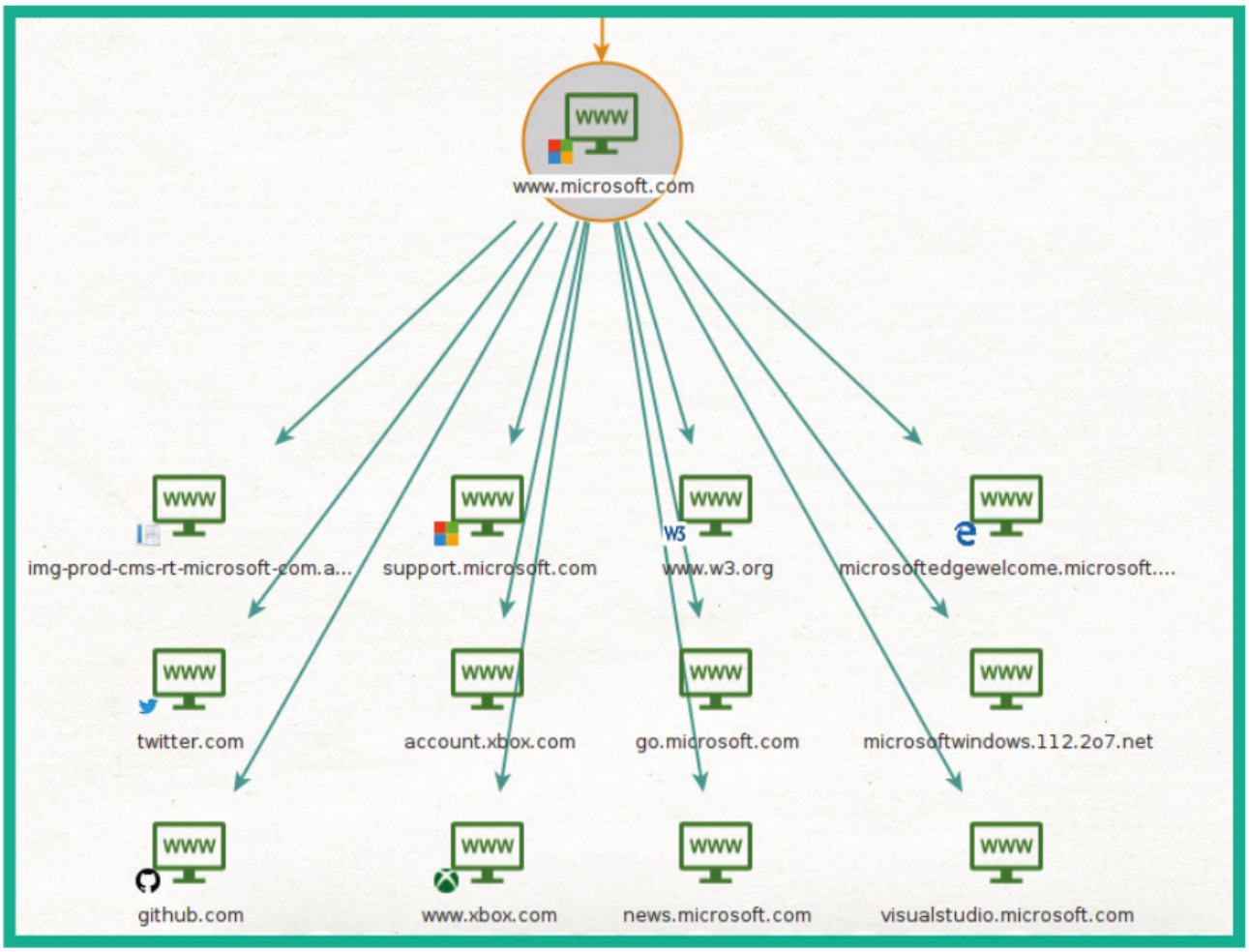


Figure 4.50 – Discovering web links on the target's website

16. To get a list of publicly available email addresses that are associated with the target's domain name, right-click on the **Domain** entity and select **Email addresses from Domain**. The following screenshot shows how easy you can quickly discover employees' email addresses:

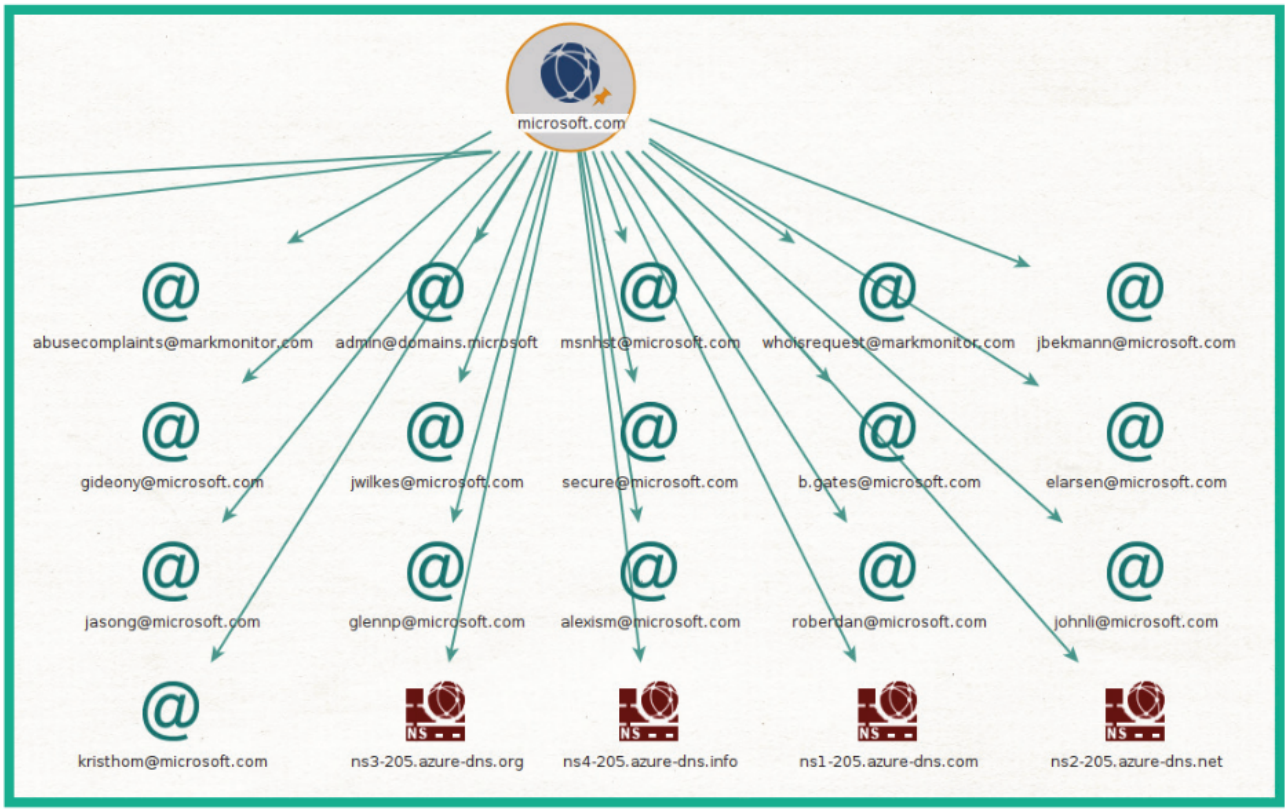


Figure 4.51 – Discovering email addresses

As you can see, using a tool such as Maltego can help automate the process of gathering OSINT data while reducing the time associated with manually performing such a task. A nice feature of Maltego is the relationship mapping on the graph, which helps you analyze information and entities easily. Using the information that's been gathered from Maltego, you can determine publicly accessible servers, IP addresses, employees' email addresses, linked URLs on web pages, and more.

Next, you will learn how to use Netcraft to profile online servers and determine their web technologies and hosting history data.

NetCraft

Netcraft allows you to gather information about a target domain, such as network block information, registrar information, email contacts, the operating system of the hosting server, and the web platform.

To start profiling an organization's online server, please use the following instructions:

1. Go to `https://www.netcraft.com/`. You will see the following search form:

Figure 4.52 – Netcraft

2. Within the search field, enter a website URL such as `www.microsoft.com` and hit *Enter* to begin the search process.
3. Next, Netcraft will provide details about the organization, as shown here:

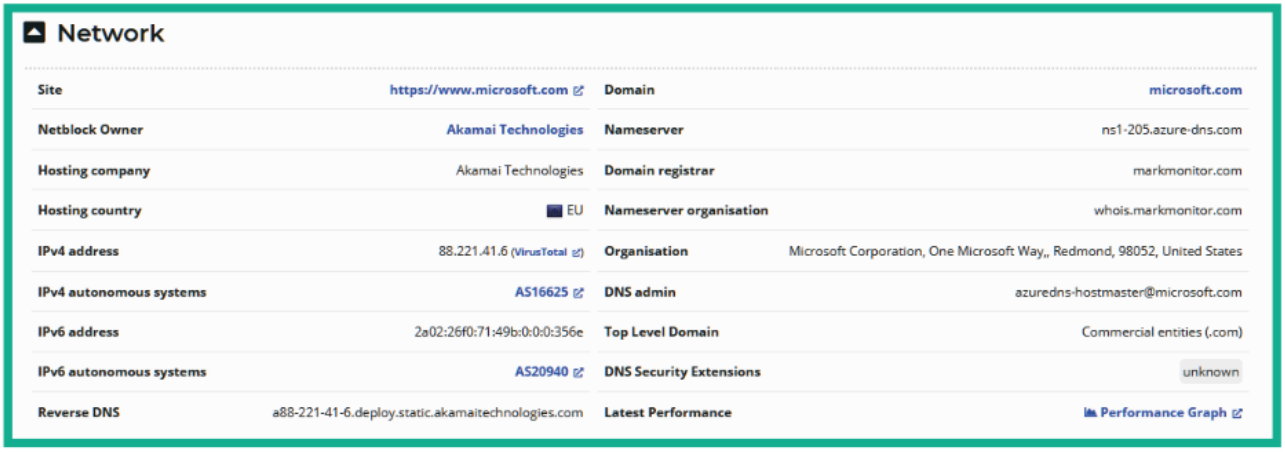


Figure 4.53 – Netcraft results

As shown in the preceding screenshot, Netcraft can identify network information about the target server, such as its name servers, IP addresses, and hosting company.

Background

Site title	Microsoft – Cloud, Computers, Apps & Gaming	**Date first seen**	August 1995
Site rank	86	**Netcraft Risk Rating**	0/10
Description	Explore Microsoft products and services for your home or business. Shop Surface, Microsoft 365, Xbox, Windows, Azure, and more. Find downloads and get support.	**Primary language**	English

Network

Site	http://www.microsoft.com	**Domain**	microsoft.com
Netblock Owner	Akamai Technologies	**Nameserver**	ns1-39.azure-dns.com
Hosting company	Akamai Technologies	**Domain registrar**	markmonitor.com
Hosting country	EU	**Nameserver organisation**	whois.markmonitor.com
IPv4 address	2.19.61.135 (VirusTotal)	**Organisation**	Microsoft Corporation, One Microsoft Way,, Redmond, 98052, United States
IPv4 autonomous systems	AS16625	**DNS admin**	azuredns-hostmaster@microsoft.com
IPv6 address	2a02:26f0:9d00:388:0:0:0:356e	**Top Level Domain**	Commercial entities (.com)
IPv6 autonomous systems	AS20940	**DNS Security Extensions**	unknown
Reverse DNS	a2-19-61-135.deploy.static.akamaitechnologies.com		

IP delegation

IPv4 address (2.19.61.135)

IP range	Country	Name	Description
::ffff:0.0.0.0/96	United States	IANA-IPV4-MAPPED-ADDRESS	Internet Assigned Numbers Authority
↳ 2.0.0.0-2.255.255.255	Netherlands	2-RIPE	RIPE Network Coordination Centre
↳ 2.16.0.0-2.23.255.255	European Union	NL-AKAMAI-20100910	Akamai International B.V.
↳ 2.19.48.0-2.19.63.255	European Union	AKAMAI-PA	Akamai Technologies
↳ 2.19.61.135	European Union	AKAMAI-PA	Akamai Technologies

IPv6 address (2a02:26f0:9d00:388:0:0:0:356e)

IP range	Country	Name	Description
::/0	N/A	ROOT	Root inet6num object
↳ 2a00::/11	European Union	EU-ZZ-2A00	RIPE NCC
↳ 2a00::/12	Netherlands	EU-ZZ-2A00	RIPE Network Coordination Centre
↳ 2a02:26f0::/29	European Union	NL-AKAMAI-20101022	Akamai International B.V.
↳ 2a02:26f0:9d00::/48	European Union	AKAMAI-PA	Akamai Technologies

Recon-ng

Recon-ng is an OSINT reconnaissance framework written in Python. The tool itself contains modules, a database, interactive help, and a menu system, similar to Metasploit. Recon-ng can perform web-based, information-gathering techniques using various open source platforms, and it's one of the must-have tools for any aspiring ethical hacker or penetration tester to have within their arsenal.

The latest version of Kali Linux already has Recon-ng within its pre-installed list of tools. To start using Recon-ng to gather information, use the following instructions:

1. On Kali Linux, open the **Terminal** area, type `recon-ng`, and hit *Enter* to start the framework.
2. Recon-ng uses modules to perform various information-gathering techniques on a target. By default, no modules are installed, so use the following command to install all the modules from the Recon-ng marketplace onto your system:

```
[recon-ng][default] > marketplace install all
```

The following screenshot shows that Recon-ng is downloading and installing all the modules:

```
[recon-ng][default] > marketplace install all
[*] Module installed: discovery/info_disclosure/cache_snoop
[*] Module installed: discovery/info_disclosure/interesting_files
[*] Module installed: exploitation/injection/command_injector
[*] Module installed: exploitation/injection/xpath_bruter
[*] Module installed: import/csv_file
[*] Module installed: import/list
[*] Module installed: import/masscan
[*] Module installed: import/nmap
```

Figure 4.18 – Installing Recon-ng modules

3. To view all the modules that have been installed on Recon-ng, use the `modules search` command, as shown here:

```
[recon-ng][default] > modules search

  Discovery
  _________

    discovery/info_disclosure/cache_snoop
    discovery/info_disclosure/interesting_files

  Exploitation
  ____________

    exploitation/injection/command_injector
    exploitation/injection/xpath_bruter
```

Figure 4.19 – Displaying the modules

As shown in the preceding screenshot, Recon-ng provides a list of all its installed modules and lists them by category (Discovery, Exploitation, Import, Recon, and Reporting).

4. As a penetration tester, you will be working on many projects. Using the `workspaces create <workspace-name>` command will allow you to create a unique workspace within Recon-ng. Here, you can manage the data that's been gathered from each project. Use the following command to create a new workspace:

```
[recon-ng][default] > workspaces create pentest1
```

```
[recon-ng][pentest1] > workspaces list

  +-------------------------------------+
  | Workspaces |       Modified         |
  +-------------------------------------+
  | default    | 2021-06-14 10:46:39    |
  | pentest1   | 2021-06-15 13:00:07    |
  +-------------------------------------+
```

Figure 4.20 – Viewing workspaces

Tip

The `workspaces load <workspace-name>` command allows you to select and work in the specific workspace, while the `workspaces remove <workspace-name>` command removes a workspace from Recon-ng.

. Next, we can search for a module using the `modules search` command. Use the `modules search whois` command to view a list of related modules:

```
[recon-ng][pentest1] > modules search whois
[*] Searching installed modules for 'whois'...

  Recon
  -----
    recon/companies-domains/viewdns_reverse_whois
    recon/companies-multi/whois_miner
    recon/domains-companies/whoxy_whois
    recon/domains-contacts/whois_pocs
    recon/netblocks-companies/whois_orgs

[recon-ng][pentest1] >
```

Figure 4.21 – Searching for modules

. To use a specific module within Recon-ng, use the `modules load` command.

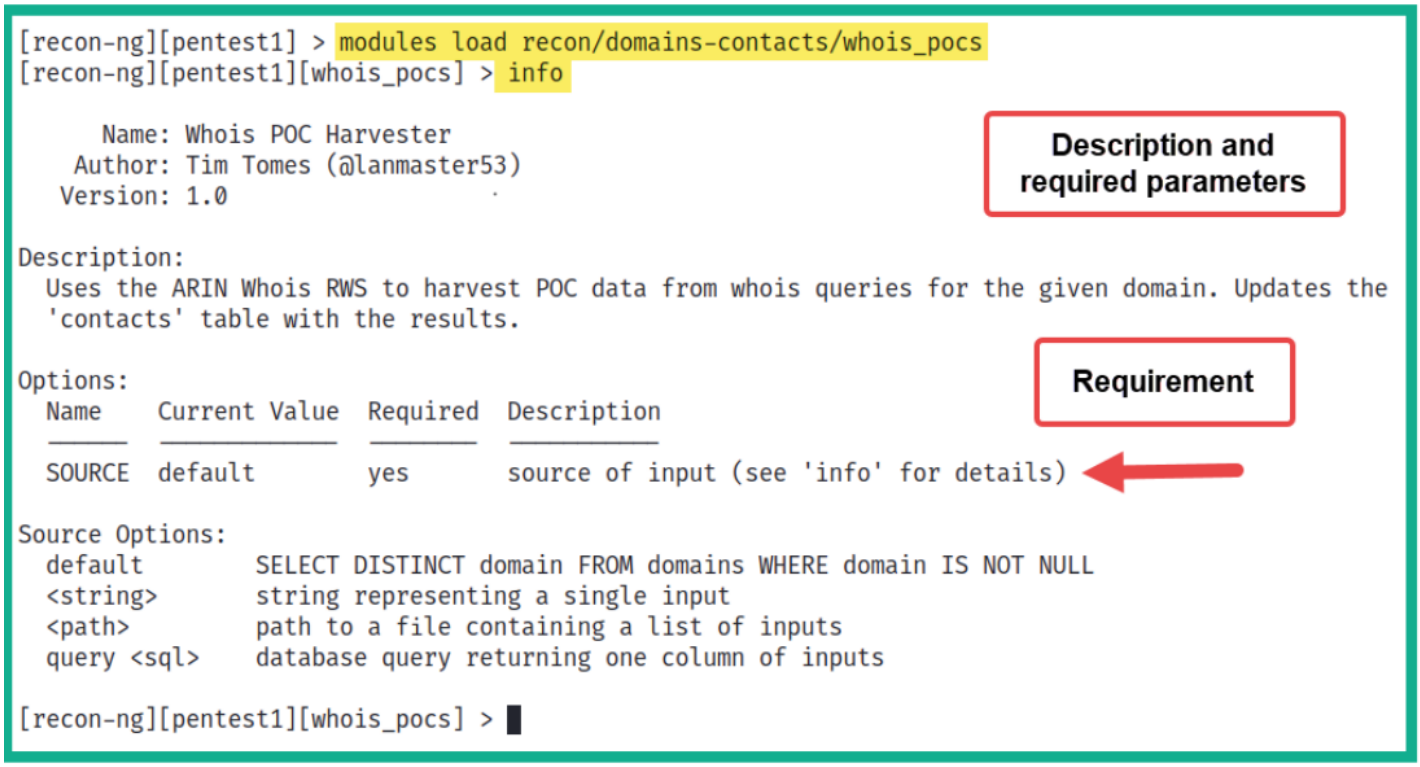


Figure 4.22 – Viewing the required parameters

As shown in the preceding screenshot, the `info` command displays a brief description of the module, how it can be used to gather information about a target, and the required parameters to use the module.

7. To set the requirements for the POCS module, use the following command to set `microsoft.com` as `SOURCE` for our target:

```
[recon-ng][pentest1][whois_pocs] > options set SOURCE microsoft.com
```

> **Tip**
>
> To unset a value within a module, use the `option unset <parameter>` command. Ensure that you use the `info` command to verify whether the parameter value is set or unset within a module.

8. Next, to execute a module, use the `run` command, as shown here:

Figure 4.23 – Executing a module

As shown in the preceding screenshot, the module is gathering OSINT data from various WHOIS databases on the internet.

> **Tip**
>
> To exit a module on Recon-ng, simply use the `back` command.

As you can see, Recon-ng is a very powerful tool and can handle data management quite well. Organizations usually create subdomains for many purposes; some can be used as login portals, or simply as other directories on a website.

9. Next, let's attempt to gather a list of subdomains for our target domain. Use the `modules search bing` command, as shown here:

```
[recon-ng][pentest1] > modules search bing
[*] Searching installed modules for 'bing'...

  Recon
  ─────
    recon/companies-contacts/bing_linkedin_cache
    recon/domains-hosts/bing_domain_api
    recon/domains-hosts/bing_domain_web
    recon/hosts-hosts/bing_ip
    recon/profiles-contacts/bing_linkedin_contacts

[recon-ng][pentest1] >
```

Figure 4.24 – Searching a module

Figure 4.23 – Executing a module

As shown in the preceding screenshot, the module is gathering OSINT data from various WHOIS databases on the internet.

> **Tip**
>
> To exit a module on Recon-ng, simply use the `back` command.

As you can see, Recon-ng is a very powerful tool and can handle data management quite well. Organizations usually create subdomains for many purposes; some can be used as login portals, or simply as other directories on a website.

9. Next, let's attempt to gather a list of subdomains for our target domain. Use the `modules search bing` command, as shown here:

```
[recon-ng][pentest1] > modules search bing
[*] Searching installed modules for 'bing'...

  Recon
  ─────
    recon/companies-contacts/bing_linkedin_cache
    recon/domains-hosts/bing_domain_api
    recon/domains-hosts/bing_domain_web
    recon/hosts-hosts/bing_ip
    recon/profiles-contacts/bing_linkedin_contacts

[recon-ng][pentest1] >
```

Figure 4.24 – Searching a module

10. Use the following commands to select the `bing_domain_web` module and display its required parameters:

```
[recon-ng][pentest1] > modules load recon/domains-hosts/bing_domain_web
[recon-ng][pentest1][bing_domain_web] > info
```

11. Next, use the `options set SOURCE microsoft.com` command to set our target domain and use the `run` command to launch the module.

> **Tip**
>
> To view a list of all supported API modules and their keys on Recon-ng, use the `keys list` command. To add an API key to Recon-ng, use the `keys add <API module name> <API key value>` command.

12. To view all the subdomains, their IP addresses, and geolocations, use the `show hosts` command, as shown here:

```
[recon-ng][pentest1] > show hosts

+---------+-------------------------------+------------+--------+----------------+
| rowid   | host                          | ip_address | region | module         |
+---------+-------------------------------+------------+--------+----------------+
| 1       | windowsupdate.microsoft.com   |            |        | bing_domain_web |
| 2       | education.microsoft.com       |            |        | bing_domain_web |
| 3       | lookbook.microsoft.com        |            |        | bing_domain_web |
| 4       | myinspire.microsoft.com       |            |        | bing_domain_web |
| 5       | supplier.microsoft.com        |            |        | bing_domain_web |
| 6       | myignite.microsoft.com        |            |        | bing_domain_web |
| 7       | myworkaccount.microsoft.com   |            |        | bing_domain_web |
| 8       | rdweb.wvd.microsoft.com       |            |        | bing_domain_web |
| 9       | speech.microsoft.com          |            |        | bing_domain_web |
| 10      | app.whiteboard.microsoft.com  |            |        | bing_domain_web |
```

Figure 4.25 – Viewing hosts

13. To view all the contact information that was found, use the `show contacts` command, as shown here:

```
[recon-ng][pentest1] > show contacts

+---------+--------------+---------------+-------------+-------------------------------------------+---------------+
| rowid   | first_name   | middle_name   | last_name   | email                                     | title         |
+---------+--------------+---------------+-------------+-------------------------------------------+---------------+
| 1       | CHRIS        |               | AADLAND     | [redacted]@microsoft.com                  | Whois contact |
| 2       | CHRISTINA    |               | AADLAND     | [redacted]@microsoft.com                  | Whois contact |
| 3       | Christina    |               | Aadland     | [redacted]@microsoft.com                  | Whois contact |
| 4       |              |               | Abuse       | abuse@microsoft.com                       | Whois contact |
| 5       |              |               | Administrator | ips.global.admin@ipayout.onmicrosoft.com | Whois contact |
| 6       |              |               | AFIADMIN    | NetworkDesign@amfam.onmicrosoft.com       | Whois contact |
| 7       | Melissa      |               | Allison     | [redacted]@ocmcdonald.onmicrosoft.com     | Whois contact |
| 8       | Jeffrey      |               | Amels       | [redacted]@microsoft.com                  | Whois contact |
```

> **Tip**
>
> The `show` command can be used with `show [companies] [credentials] [hosts] [locations] [ports] [pushpins] [vulnerabilities] [contacts] [domains] [leaks] [netblocks] [profiles] [repositories]` to view specific information that was obtained by Recon-ng.

14. To view a summary of your activities, use the `dashboard` command:

```
[recon-ng][pentest1] > dashboard

+-------------------------------------------------+
|                Activity Summary                 |
+-------------------------------------------------+
|                  Module                  | Runs |
+-------------------------------------------------+
| recon/domains-contacts/whois_pocs        | 1    |
| recon/domains-hosts/bing_domain_web      | 1    |
| recon/domains-hosts/builtwith            | 10   |
| recon/domains-hosts/google_site_web      | 1    |
| recon/domains-hosts/netcraft             | 1    |
+-------------------------------------------------+
```

Figure 4.27 – Activity summary

The preceding screenshot shows a summary of activities that were performed by the user and the number of times a module was executed. The following screenshot shows a summary of the amount of data that was collected by Recon-ng:

```
+-------------------------------+
|       Results Summary         |
+-------------------------------+
|     Category     | Quantity   |
+-------------------------------+
| Domains          | 0          |
| Companies        | 0          |
| Netblocks        | 0          |
| Locations        | 0          |
| Vulnerabilities  | 0          |
| Ports            | 0          |
| Hosts            | 100        |
| Contacts         | 30         |
| Credentials      | 0          |
| Leaks            | 0          |
| Pushpins         | 0          |
| Profiles         | 0          |
| Repositories     | 0          |
+-------------------------------+
```

15. Collecting all the data can be very overwhelming to process, but fortunately, Recon-ng has us covered with reporting modules. Use the `modules search report` command to view a list of all the reporting modules:

```
[recon-ng][pentest1] > modules search report
[*] Searching installed modules for 'report'...

  Reporting
  ---------
    reporting/csv
    reporting/html
    reporting/json
    reporting/list
    reporting/proxifier
    reporting/pushpin
    reporting/xlsx
    reporting/xml
```

Figure 4.29 – Reporting modules

16. To generate an HTML report, use the following commands to set the required parameters and output location for the final report:

```
[recon-ng][pentest1] > modules load reporting/html
[recon-ng][pentest1] > info
[recon-ng][pentest1][html] > options set CREATOR Glen
[recon-ng][pentest1][html] > options set CUSTOMER MS-Target
[recon-ng][pentest1][html] > options set FILENAME /home/kali/PenTest1-Report.html
[recon-ng][pentest1][html] > run
```

The following screenshot shows how the command was applied to the module:

```
[recon-ng][pentest1] > modules load reporting/html
[recon-ng][pentest1][html] > info                                                    1

Description:
  Creates an HTML report.

Options:
  Name      Current Value                                                Required  Description
  --------  -----------------------------------------------------------  --------  -----------
  CREATOR                                                                yes       use creator name in the report footer
  CUSTOMER                                                               yes       use customer name in the report header
  FILENAME  /home/kali/.recon-ng/workspaces/pentest1/results.html        yes       path and filename for report output
  SANITIZE  True                                                         yes       mask sensitive data in the report

[recon-ng][pentest1][html] > options set CREATOR Glen                                2
CREATOR ⇒ Glen
[recon-ng][pentest1][html] > options set CUSTOMER MS-Target                          3
CUSTOMER ⇒ MS-Target
[recon-ng][pentest1][html] > options set FILENAME /home/kali/PenTest1-Report.html    4
FILENAME ⇒ /home/kali/PenTest1-Report.html
[recon-ng][pentest1][html] > run
[*] Report generated at '/home/kali/PenTest1-Report.html'.                           5
[recon-ng][pentest1][html] >
```

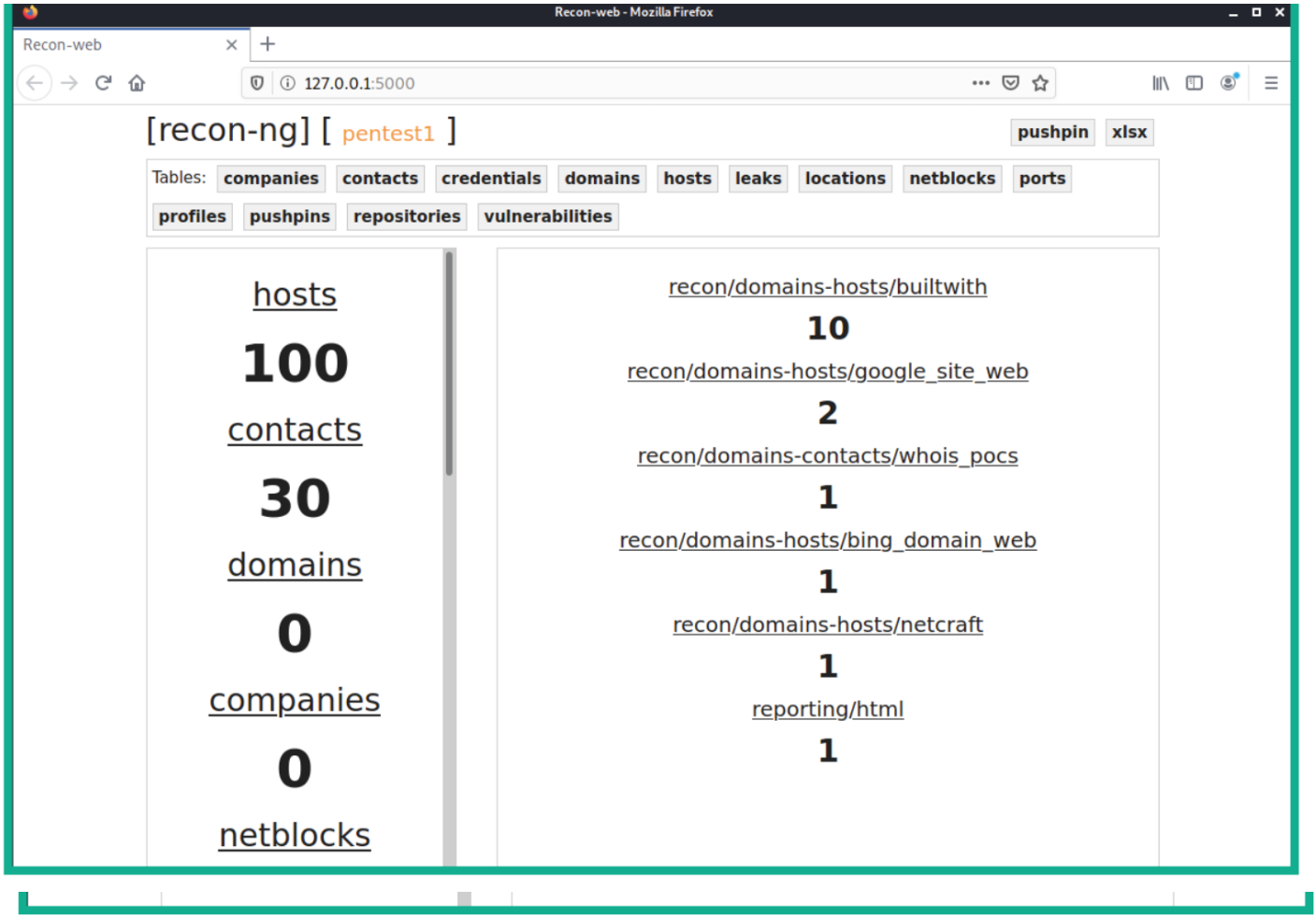


Figure 4.32 – Recon-ng web user interface

As shown in the preceding screenshot, an ethical hacker and penetration tester can perform data analysis and reporting using the web interface of the Recon-ng tool.

> **Important Note**
>
> To learn more about Recon-ng, you can visit the official GitHub repository at `https://github.com/lanmaster53/recon-ng`.

Having completed this section, you have learned how to use Recon-ng, a very popular tool for performing open source intelligence to gather data about a target organization. Next, you will learn how to use another command-line tool to gather employee data from various sources.

theHarvester

Imagine that you can use the internet to gather employees' details for a specific organization, such as their names, email addresses, and even the organization's sub-domains. Such information is valuable to a penetration tester as it can lead to performing social engineering attacks on specific employees within the organization. Pre-installed within Kali Linux is `theHarvester`, a tool that is designed to leverage the power of the internet and the information stored on various public platforms, such as social media platforms.

To get started using `theHarvester`, please use the following instructions:

1. Open the Terminal within Kali Linux and execute the following command to view the `theHarvester` help menu:

```
kali@kali:~$ theHarvester -h
```

   The details provided will help you understand how `theHarvester` can be used with various parameters to retrieve data from specific online sources.

2. Next, let's gather the names of employees who work, or worked, at Microsoft and have a LinkedIn profile by using the following command:

```
kali@kali:~$ theHarvester -d microsoft.com --dns-server
8.8.8.8 -b linkedin
```

   Let's take a look at the syntax that was used in the preceding command:

   - `-d`: Specifies the target organization by using the domain name.
   - `--dns-server`: This allows you to specify a DNS server for all DNS queries.
   - `-b`: Specifies the source to retrieve the information. Use the `theHarvester –h` command to view a list of all the available sources.

3. Additionally, to perform a sub-domain search using the Bing search engine, use the following command:

```
kali@kali:~$ theHarvester -d microsoft.com -b bing
```

Gathering data using WHOIS

What if you can access a database that contains the records of registered domains on the internet? Many domain registrars allow the general public to view publicly accessible information about domains. This information can be found on various **WHOIS** databases on the internet.

The following is a brief list of some information types that are usually stored for public records:

- Registrant contact information
- Administrative contact information
- Technical contact information
- Name servers

- Important dates, such as registration, update, and expiration dates
- Registry domain ID
- Registrar information

Accessing a WHOIS database is quite simple: you can use your favorite online search engine to find various WHOIS databases. The following are some popular WHOIS websites:

- `https://whois.domaintools.com`
- `https://who.is`
- `https://www.whois.com`

However, Kali Linux contains a built-in WHOIS tool. To perform a WHOIS lookup on a domain, open the Terminal on Kali Linux and use the `whois <domain-name>` syntax to begin a search, as shown here:

```
kali@kali:~$ whois microsoft.com
   Domain Name: MICROSOFT.COM
   Registry Domain ID: 2724960_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.markmonitor.com
   Registrar URL: http://www.markmonitor.com
   Updated Date: 2021-03-12T23:25:32Z
   Creation Date: 1991-05-02T04:00:00Z
   Registry Expiry Date: 2022-05-03T04:00:00Z
   Registrar: MarkMonitor Inc.
   Registrar IANA ID: 292
   Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
   Registrar Abuse Contact Phone: +1.2083895740
   Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
   Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
   Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
   Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
   Name Server: NS1-205.AZURE-DNS.COM
   Name Server: NS2-205.AZURE-DNS.NET
   Name Server: NS3-205.AZURE-DNS.ORG
   Name Server: NS4-205.AZURE-DNS.INFO
   DNSSEC: unsigned
```

Figure 4.13 – WHOIS

bash

Figure 2.1 Bash Scripting

Let's divide the command so you can understand what's going on:

- `%s` : Means we're inserting a string (text) in this position
- `%d` : Means we're adding a decimal (number) in this position
- `\n` : Means that we want to go to a new line when the print is finished

Also, take note that we are using double quotes instead of single quotes. Double quotes will allow us to be more flexible with string manipulation than the single quotes. So, most of the time, we can use the double quotes for `printf` (we rarely need to use the single quotes).

To format a string using the `printf` command, you can use the following patterns:

- `%s` : String (texts)
- `%d` : Decimal (numbers)
- `%f` : Floating-point (including signed numbers)
- `%x` : Hexadecimal
- `\n` : New line
- `\r` : Carriage return
- `\t` : Horizontal tab

Variables

What is a variable, and why does every programming language use it anyway?

Consider a variable as a storage area where you can save things like strings and numbers. The goal is to reuse them over and over again in your program, and this concept applies to any programming language (not just Bash scripting).

To declare a variable, you give it a name and a value (the value is a string by default). The name of the variable can only contain an alphabetic character or underscore (other programming languages use a different naming convention). For example, if you want to store the IP address of the router in a variable, first you will create a file `var.sh` (Bash script files will end with `.sh`), and inside the file, you'll enter the following:

```
#!/bin/bash
#Simple program with a variable

ROUTERIP="10.0.0.1"

printf "The router IP address: $ROUTERIP\n"
```

Let's explain your first Bash script file:

- `#!/bin/bash` is called the *Bash shebang*; we need to include it at the top to tell Kali Linux which interpreter to use to parse the script file (we will use the same concept in Chapter 18, "Pentest Automation with Python," with the Python programming language). The `#` is used in the second line to indicate that it's a comment (a comment is a directive that the creator will leave inside the source code/script for later reference).
- The variable name is called `ROUTERIP` , and its value is 10.0.0.1.
- Finally, we're *printing* the value to the output screen using the `printf` function.

To execute it, make sure to give it the right permissions first (look at the following output to see what happens if you don't). Since we're inside the same directory (`/root`), we will use `./var.sh` to execute it:

```
root@kali:~# ./var.sh
bash: ./var.sh: Permission denied
root@kali:~# chmod +x var.sh
root@kali:~# ./var.sh
The router IP address: 10.0.0.1
```

Congratulations, you just built your first Bash script! Let's say we want this script to *run automatically* without specifying its path anywhere in the system. To do that, we must add it to the `$PATH` variable. In our case, we will add `/opt` to the `$PATH` variable so we can save our custom scripts in this directory.

First, open the `.bashrc` file using any text editor. Once the file is loaded, scroll to the bottom and add the line highlighted in Figure 2.2.

```
# Some more alias to avoid making mistakes:
# alias rm='rm -i'
# alias cp='cp -i'
# alias mv='mv -i'
export PATH=$PATH:/opt/
```

Figure 2.2 Export Config

and close all the terminal sessions. Reopen the terminal window and copy the script file to the `/opt` folder. From now on, we don't need to include its path; we just execute it by typing the script name `var.sh` (you don't need to re-execute the `chmod` again; the execution permission has been already set):

```
root@kali:~# cp var.sh /opt/
root@kali:~# cd /opt
root@kali:/opt# ls -la | grep "var.sh"

-rwxr-xr-x 1 root root          110 Sep 28 11:24 var.sh
root@kali:/opt# var.sh
The router IP address: 10.0.0.1
```

Commands Variable

Sometimes, you might want to execute commands and save their output to a variable. Most of the time, the goal behind this is to manipulate the contents of the command output. Here's a simple command that executes the `ls` command and filters out the filenames that contain the word *simple* using the `grep` command. (Don't worry, you will see more complex scenarios in the upcoming sections of this chapter. For the time being, practice and focus on the fundamentals.)

```
#!/bin/bash
LS_CMD=$(ls | grep 'simple')
printf "$LS_CMD\n"
```

Here are the script execution results:

```
root@kali:/opt# simplels.sh
simpleadd.sh
simplels.sh
```

Script Parameters

Sometimes, you will need to supply parameters to your Bash script. You will have to separate each parameter with a space, and then you can manipulate those params inside the Bash script. Let's create a simple calculator (`simpleadd.sh`) that adds two numbers:

```
#!/bin/bash
#Simple calculator that adds 2 numbers
```

```
#Store the first parameter in num1 variable
NUM1=$1
#Store the second parameter in num2 variable
NUM2=$2
#Store the addition results in the total variable
TOTAL=$(($NUM1 + $NUM2))

echo '########################'
printf "%s %d\n" "The total is =" $TOTAL
echo '########################'
```

You can see in the previous script that we accessed the first parameter using the `$1` syntax and the second parameter using `$2` (you can add as many parameters as you want).

Let's add two numbers together using our new script file (take note that I'm storing my scripts in the `opt` folder from now on):

```
root@kali:/opt# simpleadd.sh 5 2
########################
The total is = 7
########################
```

There is a limitation to the previous script; it can add only two numbers. What if you want to have the flexibility to add two to five numbers? In this case, we can use the default parameter functionality. In other words, by default, all the parameter values are set to zero, and we add them up once a real value is supplied from the script:

```
#!/bin/bash
#Simple calculator that adds until 5 numbers

#Store the first parameter in num1 variable
NUM1=${1:-0}
#Store the second parameter in num2 variable
NUM2=${2:-0}
#Store the third parameter in num3 variable
NUM3=${3:-0}
#Store the fourth parameter in num4 variable
NUM4=${4:-0}
#Store the fifth parameter in num5 variable
NUM5=${5:-0}
#Store the addition results in the total variable
TOTAL=$(($NUM1 + $NUM2 + $NUM3 + $NUM4 + $NUM5))
```

```
echo '########################'
printf "%s %d\n" "The total is =" $TOTAL
echo '########################'
```

To understand how it works, let's look at the `NUM1` variable as an example (the same concept applies to the five variables). We will tell it to read the first parameter `{1` from the terminal window, and if it's not supplied by the user, then set it to zero, as in `:-0}` .

Using the default variables, we're not limited to adding five numbers; from now on, we can add as many numbers as we want, but the maximum is five (in the following example, we will add three digits):

```
root@kali:~# simpleadd.sh 2 4 4
########################
The total is = 10
########################
```

TIP

If you want to know the number of parameters supplied in the script, then you can use the `$#` to get the total. Based on the preceding example, the `$#` will be equal to three since we're passing three arguments.

If you add the following line after the `printf` line:

```
printf "%s %d\n" "The total number of params =" $#
```

you should see the following in the terminal window:

```
root@kali:~# simpleadd.sh 2 4 4

########################

The total is = 10

The total number of params = 3

########################
```

User Input

Another way to interact with the supplied input from the shell script is to use the read function. Again, the best way to explain this is through examples. We will ask the user to enter their first name and last name after which we will print the full name on the screen:

```
 #!/bin/bash

read -p "Please enter your first name:" FIRSTNAME
read -p "Please enter your last name:" LASTNAME

printf "Your full name is: $FIRSTNAME $LASTNAME\n"
```

To execute it, we just enter the script name (we don't need to supply any parameters like we did before). Once we enter the script's name, we will be prompted with the messages defined in the previous script:

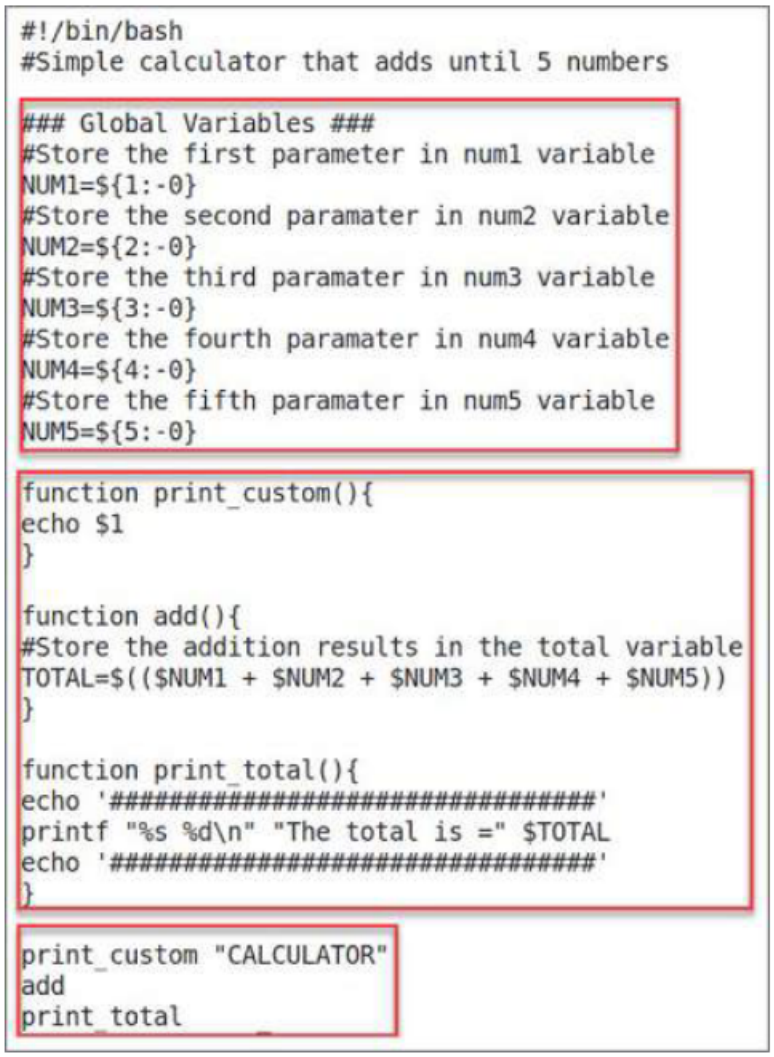
```
root@kali:~# nameprint.sh
Please enter your first name:Gus
Please enter your last name:Khawaja
Your full name is: Gus Khawaja
```

Functions

Functions are a way to organize your Bash script into logical sections instead of having an unorganized structure (programmers call it *spaghetti code*). Let's take the earlier calculator program and reorganize it (refactor it) to make it look better.

This Bash script (in Figure 2.3) is divided into three sections:

- In the first section, we create all the global variables. Global variables are accessible inside any function you create. For example, we are able to use all the `NUM` variables declared in the example inside the `add` function.
- Next, we build the functions by dividing our applications into logical sections. The `print_custom()` function will just print any text that we give it. We're using the `$1` to access the parameter value passed to this function (which is the string `CALCULATOR`).
- Finally, we call each function sequentially (each one by its name). Print the header, add the numbers, and, finally, print the results.



```
#!/bin/bash
#Simple calculator that adds until 5 numbers

### Global Variables ###
#Store the first parameter in num1 variable
NUM1=${1:-0}
#Store the second paramater in num2 variable
NUM2=${2:-0}
#Store the third paramater in num3 variable
NUM3=${3:-0}
#Store the fourth paramater in num4 variable
NUM4=${4:-0}
#Store the fifth paramater in num5 variable
NUM5=${5:-0}

function print_custom(){
echo $1
}

function add(){
#Store the addition results in the total variable
TOTAL=$(($NUM1 + $NUM2 + $NUM3 + $NUM4 + $NUM5))
}

function print_total(){
echo '################################'
printf "%s %d\n" "The total is =" $TOTAL
echo '################################'
}

print_custom "CALCULATOR"
add
print_total
```

Figure 2.3 Script Sections

Conditions and Loops

Now that you know the basics of Bash scripting, we can introduce more advanced techniques. When you develop programs in most programming languages (e.g., PHP, Python, C, C++, C#, etc.), including Bash scripting, you will encounter conditions (`if` statements) and loops, as shown in Figure 2.4.

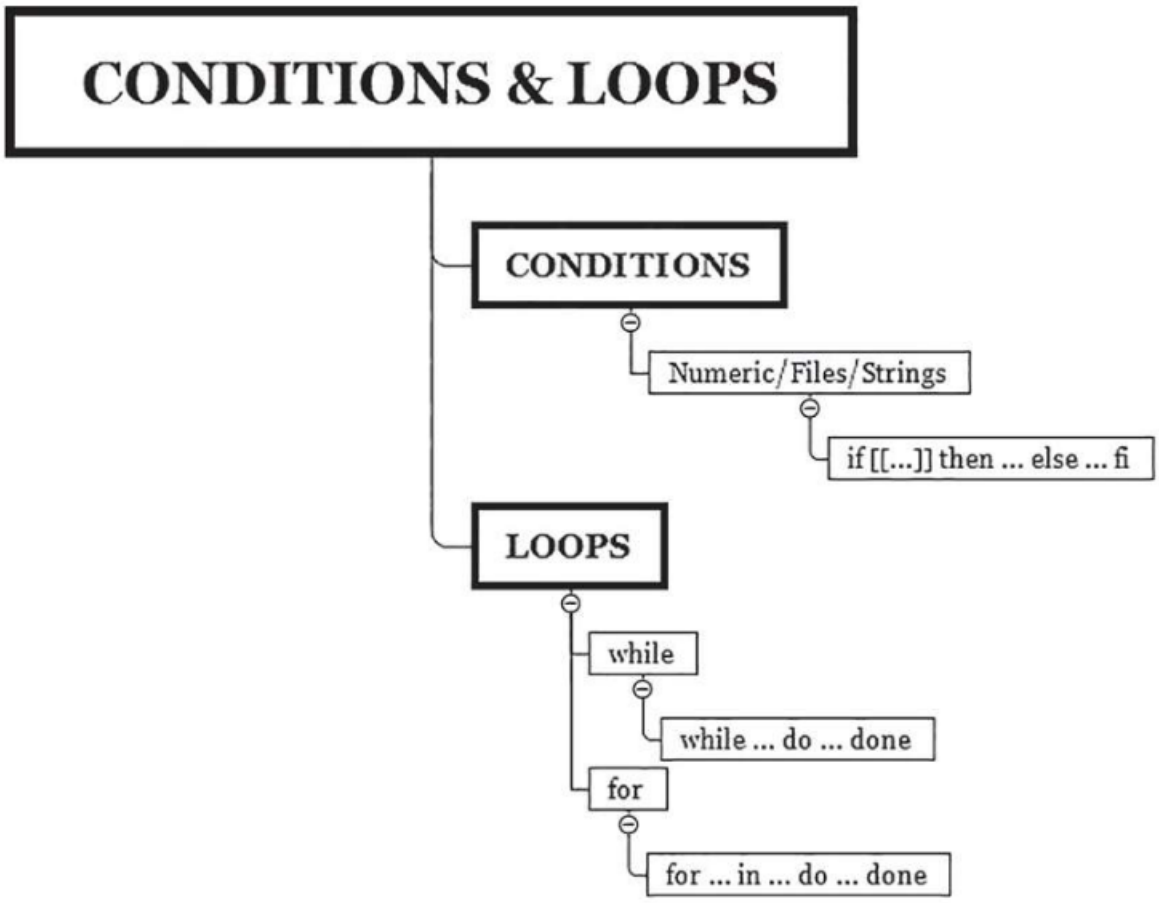


Figure 2.4 Conditions and Loops

Conditions

An `if` statement takes the following pattern:

```
if [[ comparison ]]
then
True, do something
else
False, Do something else
fi
```

If you've been paying attention, you know that the best way to explain this pattern is through examples. Let's develop a program that pings a host using Nmap, and we'll display the state of the machine depending on the condition (the host is up or down):

```
#!/bin/bash
#Ping a host using Nmap
```

```
### Global Variables ###
#Store IP address
IP_ADDRESS=$1

function ping_host(){
ping_cmd=$(nmap -sn $IP_ADDRESS | grep 'Host is up' | cut -d '(' -f
1)
}

function print_status(){
if [[ -z $ping_cmd ]]
then
echo 'Host is down'
else
echo 'Host is up'
fi
}

ping_host
print_status
```

The `nmap` command either returns an empty string text if the host is down or returns the value “Host is up” if it's responding. (Try to execute the full `nmap` command in your terminal window to visualize the difference. If so, replace `$IP_ADDRESS` with a real IP address.) In the `if` condition, the `-z` option will check if the string is empty; if yes, then we print “Host is down” or else we print “Host is up:”

```
root@kali:~# simpleping.sh 10.0.0.11
Host is down
root@kali:~# simpleping.sh 10.0.0.1
Host is up
```

What about other condition statements? In fact, you can compare numbers, strings, or files, as shown in Tables 2.1, 2.2, and 2.3.

Table 2.1 Numerical Conditions

Equal	`[[ x -eq y ]]`
Not equal	`[[ x -ne y ]]`
Less than	`[[ x -lt y ]]`
Greater than	`[[ x -gt y ]]`

Table 2.2 String Conditions

Equal	`[[ str1 == str2 ]]`
Not equal	`[[ str1 != str2 ]]`
Empty string	`[[ -z str ]]`
Not empty string	`[[ -n str ]]`

Table 2.3 File/Directory Conditions

File exists?	`[[ -a filename ]]`
Directory exists?	`[[ -d directoryname ]]`
Readable file?	`[[ -r filename ]]`
Writable file?	`[[ -w filename ]]`
Executable file?	`[[ -x filename ]]`
File not empty?	`[[ -s filename ]]`

Loops

You can write loops in two different ways: using a `while` loop or using a `for` loop. Most of the programming languages use the same pattern for loops. So, if you understand how loops work in Bash, the same concept will apply for Python, for example.

Let's start with a `while` loop that takes the following structure:

```
while [[ condition ]]
do
do something
done
```

The best way to explain a loop is through a counter from 1 to 10. We'll develop a program that displays a progress bar:

```
#!/bin/bash
#Progress bar with a while loop

#Counter
COUNTER=1
#Bar
BAR='##########'
```

```
while [[ $COUNTER -lt 11 ]]
do
#Print the bar progress starting from the zero index
echo -ne "\r${BAR:0:COUNTER}"
#Sleep for 1 second
sleep 1
#Increment counter
COUNTER=$(( $COUNTER +1 ))
done
```

Note that the condition (`[[ $COUNTER -lt 11]]`) in the `while` loop follows the same rules as the `if` condition. Since we want the counter to stop at 10, we will use the following mathematical formula: `counter<11` . Each time the counter is incremented, it will display the progress. To make this program more interesting, let it sleep for one second before going into the next number.

On the other hand, the `for` loop will take the following pattern:

```
for … in [List of items]
do
something
done
```

We will take the same example as before but use it with a `for` loop. You will realize that the `for` loop is more flexible to implement than the `while` loop. (Honestly, I rarely use the `while` loop.) Also, you won't need to increment your index counter; it's done automatically for you:

```
#!/bin/bash
#Progress bar with a For Loop

#Bar
BAR='##########'

for COUNTER in {1..10}
do
#Print the bar progress starting from the zero index
echo -ne "\r${BAR:0:$COUNTER}"
#Sleep for 1 second
sleep 1
done
```

File Iteration

Here's what you should do to simply read a text file in Bash using the `for` loop:

```
for line in $(cat filename)
do
do something
done
```

In the following example, we will save a list of IP addresses in a file called `ips.txt` . Then, we will reuse the Nmap ping program (that we created previously) to check whether every IP address is up or down. On top of that, we will check the DNS name of each IP address:

```
#!/bin/bash
#Ping & get DNS name from a list of IPs saved in a file

#Prompt the user to enter a file name and its path.
read -p "Enter the IP addresses file name / path:" FILE_PATH_NAME


function check_host(){
        #if not the IP address value is empty
        if [[ -n $IP_ADDRESS ]]
        then
                ping_cmd=$(nmap -sn $IP_ADDRESS| grep 'Host is up'
| cut -d '(' -f 1)
                echo '-----------------------------------------
---'
                if [[ -z $ping_cmd ]]
                then
                        printf "$IP_ADDRESS is down\n"
                else
                        printf "$IP_ADDRESS is up\n"
                        dns_name
                fi
        fi
}

function dns_name(){
        dns_name=$(host $IP_ADDRESS)
        printf "$dns_name\n"
}

#Iterate through the IP addresses inside the file
for ip in $(cat $FILE_PATH_NAME)
do
        IP_ADDRESS=$ip
        check_host
done
```

encrypt

1. Open a terminal window and generate a GPG key

The first thing to do is open the terminal window from your desktop menu. Once it's open, you'll want to generate a GPG key with the command:

```
gpg --gen-key
```

You'll be asked to enter your real name and an email address, then type "O "to Okay the information. After that, you type/verify a passphrase for the key.

2. Change into the directory housing the file

With your key created, navigate to the folder housing the file to be encrypted. Let's say the file is in ~/Documents. Change to that directory with the command:

```
cd ~/Documents
```

3. Encrypt the file

We're going to use the *gpg* command to encrypt the file. For example, we'll encrypt the file zdnet_test with the command:

```
gpg -c zdnet_test
```

The -c option tells gpg the zdnet_test file is to be encrypted. You will then be asked to type and verify a password for the encrypted file.

Once you've encrypted the file, you'll notice there are two files: zdnet_test and zdnet_test.gpg. The file with the .gpg extension is the encrypted file. At this point, you can remove the initial test file with the command:

```
rm zdnet_test
```

4. Configure the password cache agent

Oddly enough, the GPG tool caches passwords. Because of this, you (or anyone who has access to your system) could decrypt the file without having to type the password with the command *gpg zdnet_test*. That's not safe. To get around this, we have to disable password caching for the GPG agent. To do this, create a new file with the command:

```
nano ~/.gnupg/gpg-agent.conf
```

In that file, paste the following lines:

```
default-cache-ttl 1
max-cache-ttl 1
```

Next, restart the agent with the command:

```
echo RELOADAGENT | gpg-connect-agent
```

Now, when you (or anyone) types the decrypt command, *gpg zdnet_test*, the password prompt will appear. Until that password is successfully entered, the contents of the file will remain encrypted.

The GUI (Graphical User Interface) method of encrypting files

This method is significantly more efficient.

1. Install the required software

Before you use the GUI method, make sure to take care of Steps 1 and 4 above. You only have to do this once. After that, you'll need to install a piece of software with the command:

```
sudo apt-get install seahorse-nautilus -y
```

If you're using a distribution based on RHEL or Fedora Linux, that command would be:

```
sudo dnf install seahorse-nautilus -y
```

Once installed, restart Nautilus with the command:

```
nautilus -q
```

that I will cover in this section are the `.tar` , `.gz` , `.bz2` , and `.zip` extensions. Here's the list of commands to compress and extract different types of archives:

Tar Archive

To compress using tar extension:

```
$tar cf compressed.tar files
```

To extract a tar compressed file:

```
$tar xf compressed.tar
```

Gz Archive

To create compressed.tar.gz from files:

```
$tar cfz compressed.tar.gz files
```


To extract compressed.tar.gz:

```
$tar xfz compressed.tar.gz
```

To create a compressed.txt.gz file:

```
$gzip file.txt> compressed.txt.gz
```

To extract compressed.txt.gz:

```
$gzip -d compressed.txt.gz
```

Let's extract the `rockyou.txt.gz` file that comes initially compressed in Kali:

```
root@kali:~# gzip -d /usr/share/wordlists/rockyou.txt.gz
```