Security domains cybersecurity analysts need to know

As an analyst, you can explore various areas of cybersecurity that interest you. One way to explore those areas is by understanding different security domains and how they're used to organize the work of security professionals. In this reading you will learn more about CISSP's eight security domains and how they relate to the work you'll do as a security analyst.

Domain one: Security and risk management

All organizations must develop their security posture. Security posture is an organization's ability to manage its defense of critical assets and data and react to change. Elements of the security and risk management domain that impact an organization's security posture include:

- Security goals and objectives
- Risk mitigation processes
- Compliance
- Business continuity plans
- Legal regulations
- Professional and organizational ethics

Information security, or InfoSec, is also related to this domain and refers to a set of processes established to secure information. An organization may use playbooks and implement training as a part of their security and risk management program, based on their needs and perceived risk. There are many InfoSec design processes, such as:

- Incident response
- Vulnerability management
- Application security
- Cloud security
- Infrastructure security

As an example, a security team may need to alter how personally identifiable information (PII) is treated in order to adhere to the European Union's General Data Protection Regulation (GDPR).

Domain two: Asset security

Asset security involves managing the cybersecurity processes of organizational assets, including the storage, maintenance, retention, and destruction of physical and virtual data. Because the loss or theft of assets can expose an organization and increase the level of risk, keeping track of assets and the data they hold is essential. Conducting a security impact analysis, establishing a recovery plan, and managing data exposure will depend on the level of risk associated with each asset. Security analysts may need to store, maintain, and retain data by creating backups to ensure they are able to restore the environment if a security incident places the organization's data at risk.

Domain three: Security architecture and engineering

This domain focuses on managing data security. Ensuring effective tools, systems, and processes are in place helps protect an organization's assets and data. Security architects and engineers create these processes.

One important aspect of this domain is the concept of shared responsibility. Shared responsibility means all individuals involved take an active role in lowering risk during the design of a security system. Additional design principles related to this domain, which are discussed later in the program, include:

One important aspect of this domain is the concept of shared responsibility. Shared responsibility means all individuals involved take an active role in lowering risk during the design of a security system. Additional design principles related to this domain, which are discussed later in the program, include:

- Threat modeling
- Least privilege
- Defense in depth
- Fail securely
- Separation of duties
- Keep it simple
- Zero trust
- Trust but verify

An example of managing data is the use of a security information and event management (SIEM) tool to monitor for flags related to unusual login or user activity that could indicate a threat actor is attempting to access private data.

Domain four: Communication and network security

This domain focuses on managing and securing physical networks and wireless communications. This includes on-site, remote, and cloud communications.

Organizations with remote, hybrid, and on-site work environments must ensure data remains secure, but managing external connections to make certain that remote workers are securely accessing an organization's networks is a challenge. Designing network security controls—such as restricted network access—can help protect users and ensure an organization's network remains secure when employees travel or work outside of the main office.

Domain five: Identity and access management

The identity and access management (IAM) domain focuses on keeping data secure. It does this by ensuring user identities are trusted and authenticated and that access to physical and logical assets is authorized. This helps prevent unauthorized users, while allowing authorized users to perform their tasks.

Essentially, IAM uses what is referred to as the principle of least privilege, which is the concept of granting only the minimal access and authorization required to complete a task. As an example, a cybersecurity analyst might be asked to ensure that customer service representatives can only view the private data of a customer, such as their phone number, while working to resolve the customer's issue; then remove access when the customer's issue is resolved.

Domain six: Security assessment and testing

The security assessment and testing domain focuses on identifying and mitigating risks, threats, and vulnerabilities. Security assessments help organizations determine whether their internal systems are secure or at risk. Organizations might employ penetration testers, often referred to as “pen testers,” to find vulnerabilities that could be exploited by a threat actor.

This domain suggests that organizations conduct security control testing, as well as collect and analyze data. Additionally, it emphasizes the importance of conducting security audits to monitor for and reduce the probability of a data breach. To contribute to these types of tasks, cybersecurity professionals may be tasked with auditing user permissions to validate that users have the correct levels of access to internal systems.

Domain seven: Security operations

The security operations domain focuses on the investigation of a potential data breach and the implementation of preventative measures after a security incident has occurred. This includes using strategies, processes, and tools such as:

- Training and awareness
- Reporting and documentation
- Intrusion detection and prevention
- SIEM tools
- Log management
- Incident management
- Playbooks
- Post-breach forensics
- Reflecting on lessons learned

The cybersecurity professionals involved in this domain work as a team to manage, prevent, and investigate threats, risks, and vulnerabilities. These individuals are trained to handle active attacks, such as large amounts of data being accessed from an organization's internal network, outside of normal working hours. Once a threat is identified, the team works diligently to keep private data and information safe from threat actors.

Domain eight: Software development security

The software development security domain is focused on using secure programming practices and guidelines to create secure applications. Having secure applications helps deliver secure and reliable services, which helps protect organizations and their users.

Security must be incorporated into each element of the software development life cycle, from design and development to testing and release. To achieve security, the software development process must have security in mind at each step. Security cannot be an afterthought.

Performing application security tests can help ensure vulnerabilities are identified and mitigated accordingly. Having a system in place to test the programming conventions, software executables, and security measures embedded in the software is necessary. Having quality assurance and pen tester professionals ensure the software has met security and performance standards is also an essential part of the software development process. For example, an entry-level analyst working for a pharmaceutical company might be asked to make sure encryption is properly configured for a new medical device that will store private patient data.

Key takeaways

In this reading, you learned more about the focus areas of the eight CISSP security domains. In addition, you learned about InfoSec and the principle of least privilege. Being familiar with these security domains and related concepts will help you gain insight into the field of cybersecurity.

Mark as completed

Like Dislike Report an issue

Profiling websites using EyeWitness

After discovering the subdomains of a target domain, it's important to check each one to determine which subdomain leads to a login portal or a sensitive directory of the organization. However, there may be a lot of subdomains to check manually, and this process can be very time-consuming. As an aspiring penetration tester, you can be strategic and use a tool such as EyeWitness, which allows you to automate the process of checking each subdomain within a file and taking a screenshot of them.

To get started using EyeWitness, please use the following instructions:

1. On Kali Linux, open the **Terminal** area and use the following command to create an offline copy of Witness:

```
kali@kali:~$ git clone https://github.com/
FortyNorthSecurity/EyeWitness
```

2. Next, use the following commands to install EyeWitness on your Kali Linux system:

```
kali@kali:~$ cd EyeWitness/Python/setup
kali@kali:~/EyeWitness/Python/setup$ sudo ./setup.sh
```

3. Next use the `cd ..` command to go up one directory, as shown here:

```
kali@kali:~/EyeWitness/Python/setup$ cd ..
```

4. Next, use the following commands to allow EyeWitness to capture a screenshot of each subdomain that was found within the `subdomains.txt` file:

```
kali@kali:~/EyeWitness/Python$ ./EyeWitness.py --web
-f /home/kali/subdomains.txt -d /home/kali/screenshots
--prepend-https
```

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

- `--web`: Takes an HTTP screenshot
- `-f`: Specifies the source file, along with the list of domains to check
- `-d`: Specifies the output directory for the screenshots
- `--prepend-https`: Prepends `http://` and `https://` to the domains without either protocol

different and that's OK once it's on the `172.30.1.0/24` network. Knowing the IP address of our machine is important to ensure we exclude it from our other scans, as well as to determine the possible network ID of our connected network.

> **Tip**
> You will need to calculate the network ID of the network that you're connected to. Go to `https://www.calculator.net/ip-subnet-calculator.html`, which allows you to insert the IP addresses and network prefix (`/x`) values on the site and get the network address/ID.

2. Next, let's use **Netdiscover** to perform an active scan of the entire network:

```
kali@kali:~$ sudo netdiscover -r 172.30.1.0/24
```

Netdiscover is a scanning tool that uses **Address Resolution Protocol** (**ARP**) messages to identify live systems on a network. Using the `-r` syntax allows you to specify a range when scanning.

> **Tip**
> You can perform a passive scan of the network using the `-p` syntax, which allows Netdiscover to listen passively for any messages that can be exchanged between hosts on the network.

The following screenshot shows the results of Netdiscover when used on live host machines on the `172.30.1.0/24` network:

```
Currently scanning: Finished!   |   Screen View: Unique Hosts

5 Captured ARP Req/Rep packets, from 2 hosts.   Total size: 300
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
_____________________________________________________________________________
172.30.1.1      08:00:27:bd:1d:71      3     180  PCS Systemtechnik GmbH
172.30.1.26     08:00:27:7f:af:0a      2     120  PCS Systemtechnik GmbH
```

Figure 5.36 – Netdiscover host discovery

As shown in the preceding screenshot, Netdiscover provided the IP addresses, MAC addresses, vendors, and hostnames of the systems. Using this information about the vendor of the NIC, a penetration tester can start researching for known vulnerabilities about the host systems.

To learn how to change your MAC address using MAC Changer, please use the following instructions:

1. On Kali Linux, open the **Terminal** area and use the `ifconfig` command to determine the number of network interfaces, as shown here:

```
kali@kali:~$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:cd:[redacted]  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.30.1.29  netmask 255.255.255.0  broadcast 172.30.1.255
        ether 08:00:27:[redacted]  txqueuelen 1000  (Ethernet)
        RX packets 7321  bytes 488009 (476.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7331  bytes 519400 (507.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 5.32 – Checking network interfaces

As shown in the preceding screenshot, there is an Ethernet connection indicated as `eth0` that is connected to the wired virtual network within our lab environment.

2. Next, use the following commands to logically turn down the `eth0` interface:

```
kali@kali:~$ sudo ifconfig eth0 down
```

Discovering live systems on a network

Discovering live hosts on the network is an essential stage when performing a penetration test. Let's imagine you're an ethical hacker or a penetration tester; your target organization permits you to directly connect your attacker machine with Kali Linux on their network to perform security testing from their internal network. You're eager to start discovering security vulnerabilities and hacking systems, but you're not sure which systems are online, nor their host operating systems.

In this section, you will learn about the skills you will need to perform various types of active reconnaissance on an organization's networks using various tools and techniques. However, to ensure you can perform these exercises in a safe space, please use the following guidelines:

- Ensure you do not scan systems that you do not own or have been granted legal permission.
- Ensure the network adapter of Kali Linux is assigned to the **PentestNet** lab network within VirtualBox Manager.
- Power on both the Metasploitable 2 and OWASP BWA virtual machines and ensure these systems are receiving an IP address on the `172.30.1.0/24` network.
- The PentestNet network will be our simulated organization network.

To get started with this exercise, please use the following instructions:

1. On Kali Linux, open the **Terminal** area and execute the `ip addr` command, as shown here:

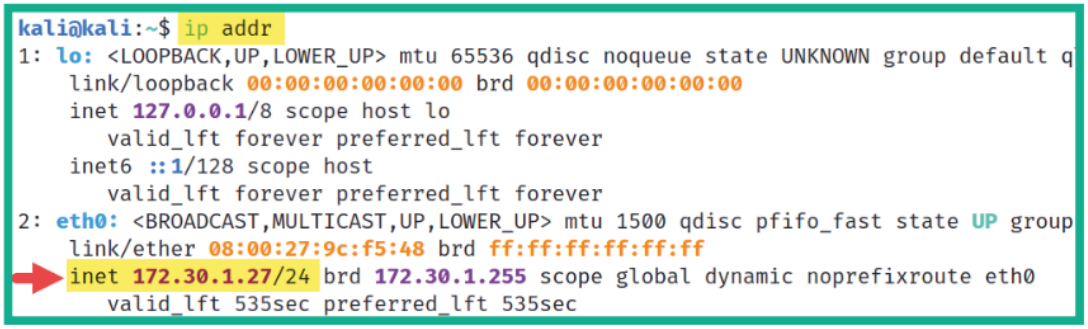


Figure 5.35 – Checking your network

As shown in the preceding screenshot, you can see multiple interfaces on your Kali Linux system and that there's an Ethernet interface called `eth0`. Ethernet interfaces are usually physical NICs on your system, but since we are using VirtualBox to virtualize Kali Linux, they are virtual NICs. Keep in mind that the wireless NICs are identified as `wlan` interfaces.

Profiling websites using EyeWitness

After discovering the subdomains of a target domain, it's important to check each one to determine which subdomain leads to a login portal or a sensitive directory of the organization. However, there may be a lot of subdomains to check manually, and this process can be very time-consuming. As an aspiring penetration tester, you can be strategic and use a tool such as EyeWitness, which allows you to automate the process of checking each subdomain within a file and taking a screenshot of them.

To get started using EyeWitness, please use the following instructions:

1. On Kali Linux, open the **Terminal** area and use the following command to create an offline copy of Witness:

```
kali@kali:~$ git clone https://github.com/
FortyNorthSecurity/EyeWitness
```

2. Next, use the following commands to install EyeWitness on your Kali Linux system:

```
kali@kali:~$ cd EyeWitness/Python/setup
kali@kali:~/EyeWitness/Python/setup$ sudo ./setup.sh
```

3. Next use the `cd ..` command to go up one directory, as shown here:

```
kali@kali:~/EyeWitness/Python/setup$ cd ..
```

4. Next, use the following commands to allow EyeWitness to capture a screenshot of each subdomain that was found within the `subdomains.txt` file:

```
kali@kali:~/EyeWitness/Python$ ./EyeWitness.py --web
-f /home/kali/subdomains.txt -d /home/kali/screenshots
--prepend-https
```

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

- `--web`: Takes an HTTP screenshot
- `-f`: Specifies the source file, along with the list of domains to check
- `-d`: Specifies the output directory for the screenshots
- `--prepend-https`: Prepends `http://` and `https://` to the domains without either protocol

```
kali@kali:~$ nmap -sn 172.30.1.0/24 --exclude 172.30.1.27
```

Using the `-sn` syntax ensures Nmap performs a ping sweep of the network. This means Nmap will send an **Internet Control Message Protocol** (**ICMP**) **Echo Request** message to all devices within the network range. Online devices will typically respond with an **ICMP Echo Reply** message. Then, Nmap will provide the results of the systems that are online, as shown here:

```
kali@kali:~$ nmap -sn 172.30.1.0/24 --exclude 172.30.1.27
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-23 08:41 EDT
Nmap scan report for 172.30.1.26
Host is up (0.0057s latency).
Nmap done: 255 IP addresses (1 host up) scanned in 15.78 seconds
```

Figure 5.37 – Ping sweep using Nmap

As shown in the preceding screenshot, Nmap indicated that the `172.30.1.26` host device (Metasploitable 2) is currently up (online). Furthermore, using the `--exclude` command allows us to specify which IP addresses to exclude from scanning. This command is best used when you are restricted from scanning various IP addresses and subnetworks during a penetration test.

Next, you will learn how to use Nmap to discover open ports and running services on a target system on a network.

Probing open service ports, services, and operating systems

After discovering the hosts on a network, the next phase is to identify any open service ports on the target system and determine which services are mapped to those open ports. There are various techniques that a penetration tester can use to identify the open ports on a target system. Some techniques are manual, while others can simply be automated using the Nmap tool. Let's take a look:

1. Since we have already discovered a live system on our network, let's use the following command to perform a basic Nmap scan of a host (Metasploitable 2):

```
kali@kali:~$ nmap 172.30.1.26
```

```
kali@kali:~$ cat subdomains.txt
064-smtp-in-2a.microsoft.com
1501.microsoft.com
108.61.72.33.microsoft.com
45.76.116.45.microsoft.com
8057.microsoft.com
8075.microsoft.com
abtesting.microsoft.com
ac2.microsoft.com
academymobile.microsoft.com
```

Figure 5.29 – Output file

As shown in the preceding screenshot, the results were stored within the `subdomains.txt` file.

As you have seen, using this tool can allow a penetration tester to save a lot of time when discovering the subdomains of a target domain.

> **Important Note**
>
> To learn more about Sublist3r, please see the official GitHub repository at `https://github.com/aboul3la/Sublist3r`.

Using the information that was found regarding subdomains, penetration testers will need to check these subdomains to determine where they lead, such as to a vulnerable web application or even a login portal for employees or customers.

Having completed this section, you have learned how to efficiently discover the subdomains of a target organization. In the next section, you will learn how to efficiently get a picture of all the subdomains of a target domain.

You can leverage the power of search engines for discovering sub-domains by using the **Sublist3r** tool. Sublist3r is a Python-based tool that is used to enumerate (extract/obtain) the subdomains of a given website using OSINT, such as search engines and other internet indexing platforms.

To get started using Sublist3r, please use the following instructions:

1. On Kali Linux, open the **Terminal** area and use the following commands to install Sublist3r:

```
kali@kali:~$ sudo apt update
kali@kali:~$ sudo apt install sublist3r
```

2. To discover the subdomains of a target domain, use the following command:

```
kali@kali:~$ sublist3r -d microsoft.com
```

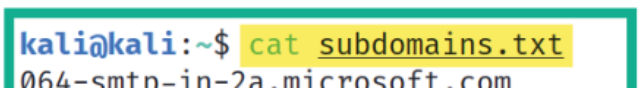
Sublist3r will query a lot of online sources, such as search engines, and return the results in your Terminal window.

3. To perform a search and store the results within an offline file, use the `-o` command, followed by the file's name, as shown here:

```
kali@kali:~$ sublist3r -d microsoft.com -o subdomains.txt
```

The output file will be stored within your present working directory – that is, your filesystem, such as the `/home/kali/` directory on your system. Use the `pwd` command to view your working directory on Kali Linux.

4. To view the contents of the output file, use the `cat subdomains.txt` command, as shown here:



eyewitness presented a problem with driver/gecko.

Simply choose **Footprint** and click on **Run Scan** to begin gathering OSINT about your target.

9. Next, while Spiderfoot is collecting data and converting it into information, you can click on **Scan** > **select your scan** and click on **Graph** to view a visual, as shown here:

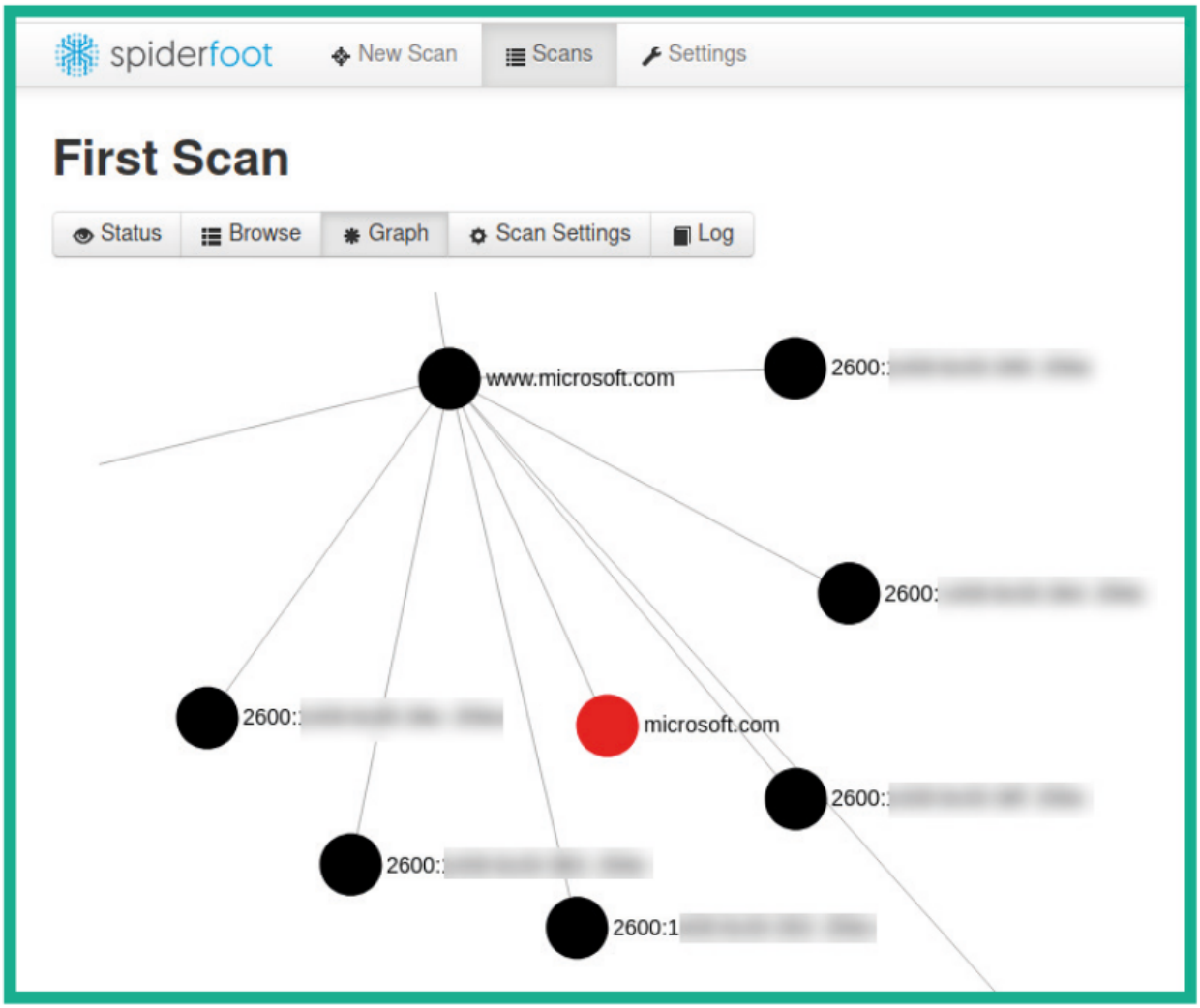


Figure 5.25 – Spiderfoot graph

As shown in the preceding screenshot, notice how Spiderfoot can show how the collected data is related to the target domain.

10. Next, to view the data that was collected based on categories, click on **Browse**, as shown in the following screenshot:

To start working with Spiderfoot, please use the following instructions:

1. Ensure your Kali Linux machine has an active internet connection as Spiderfoot needs to retrieve data from various online sources.
2. Next, open the **Terminal** area and use the `ip addr` command to identify the IP address of your Kali Linux machine:

```
kali@kali:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
    link/ether 08:00:27:9c:f5:48 brd ff:ff:ff:ff:ff:ff
    inet 172.16.17.71/24 brd 172.16.17.255 scope global dynamic noprefixroute eth0
       valid_lft 86352sec preferred_lft 86352sec
```

Figure 5.19 – Checking an IP address

Keep in mind that the network adapter within VirtualBox for your Kali Linux machine should be set to `Bridge` mode. This ensures Kali Linux has an IP address on your network. As shown in the preceding snippet, the IP address of my machine is indicated under the Ethernet (`eth0`) interface. You will need to identify your IP address before proceeding to the next phase.

3. Next, using the IP address of your Kali Linux machine, use the following commands to enable the web user interface of Spiderfoot:

```
kali@kali:~$ sudo spiderfoot -l 172.16.17.71:80
```

Ensure that you substitute the IP address shown in the preceding command with the IP address of your Kali Linux machine.

To start working with Spiderfoot, please use the following instructions:

1. Ensure your Kali Linux machine has an active internet connection as Spiderfoot needs to retrieve data from various online sources.
2. Next, open the **Terminal** area and use the `ip addr` command to identify the IP address of your Kali Linux machine:

```
kali@kali:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
    link/ether 08:00:27:9c:f5:48 brd ff:ff:ff:ff:ff:ff
    inet 172.16.17.71/24 brd 172.16.17.255 scope global dynamic noprefixroute eth0
       valid_lft 86352sec preferred_lft 86352sec
```

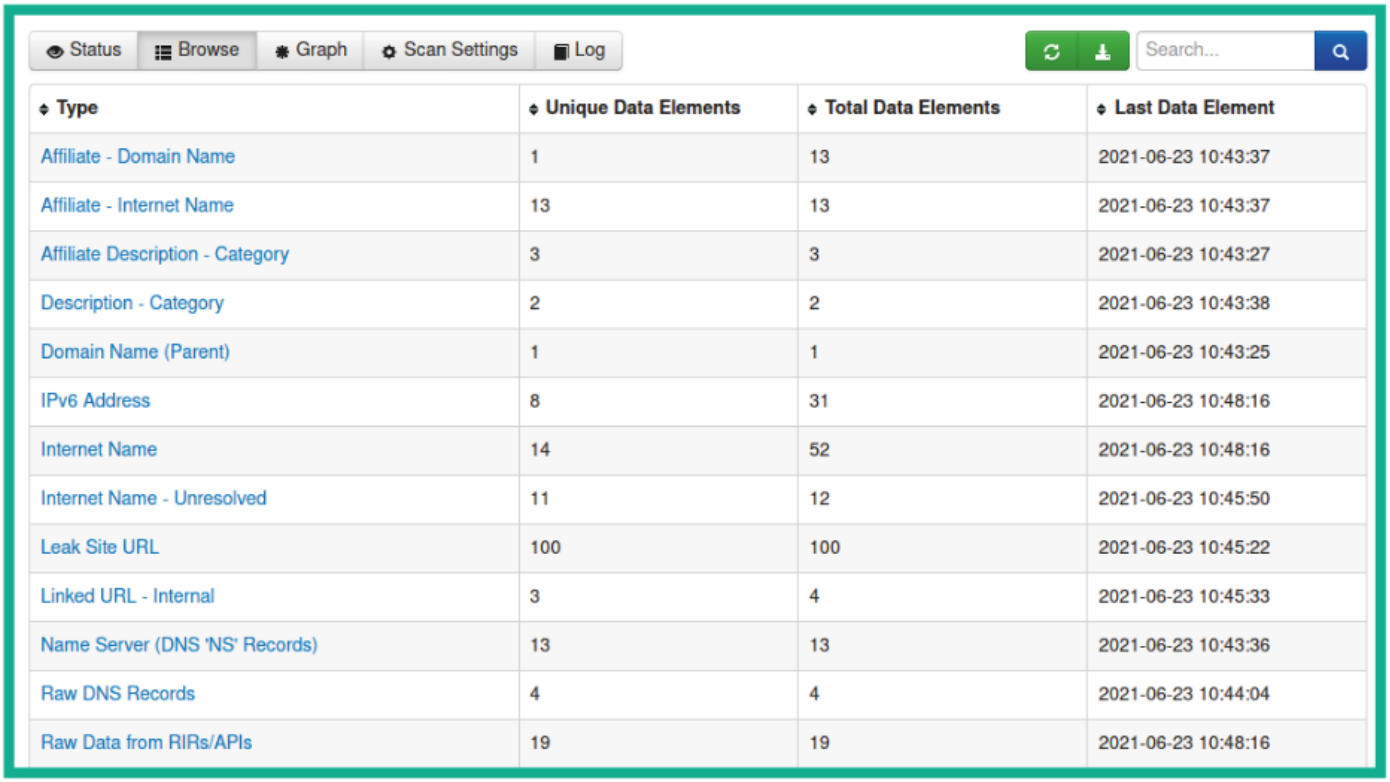
Figure 5.19 – Checking an IP address

Keep in mind that the network adapter within VirtualBox for your Kali Linux machine should be set to `Bridge` mode. This ensures Kali Linux has an IP address on your network. As shown in the preceding snippet, the IP address of my machine is indicated under the Ethernet (`eth0`) interface. You will need to identify your IP address before proceeding to the next phase.

3. Next, using the IP address of your Kali Linux machine, use the following commands to enable the web user interface of Spiderfoot:

```
kali@kali:~$ sudo spiderfoot -l 172.16.17.71:80
```

Ensure that you substitute the IP address shown in the preceding command with the IP address of your Kali Linux machine.



Type	Unique Data Elements	Total Data Elements	Last Data Element
Affiliate - Domain Name	1	13	2021-06-23 10:43:37
Affiliate - Internet Name	13	13	2021-06-23 10:43:37
Affiliate Description - Category	3	3	2021-06-23 10:43:27
Description - Category	2	2	2021-06-23 10:43:38
Domain Name (Parent)	1	1	2021-06-23 10:43:25
IPv6 Address	8	31	2021-06-23 10:48:16
Internet Name	14	52	2021-06-23 10:48:16
Internet Name - Unresolved	11	12	2021-06-23 10:45:50
Leak Site URL	100	100	2021-06-23 10:45:22
Linked URL - Internal	3	4	2021-06-23 10:45:33
Name Server (DNS 'NS' Records)	13	13	2021-06-23 10:43:36
Raw DNS Records	4	4	2021-06-23 10:44:04
Raw Data from RIRs/APIs	19	19	2021-06-23 10:48:16

Figure 5.26 – Viewing data

11. Next, click within the **Internet Name** category to see the data that was collected:

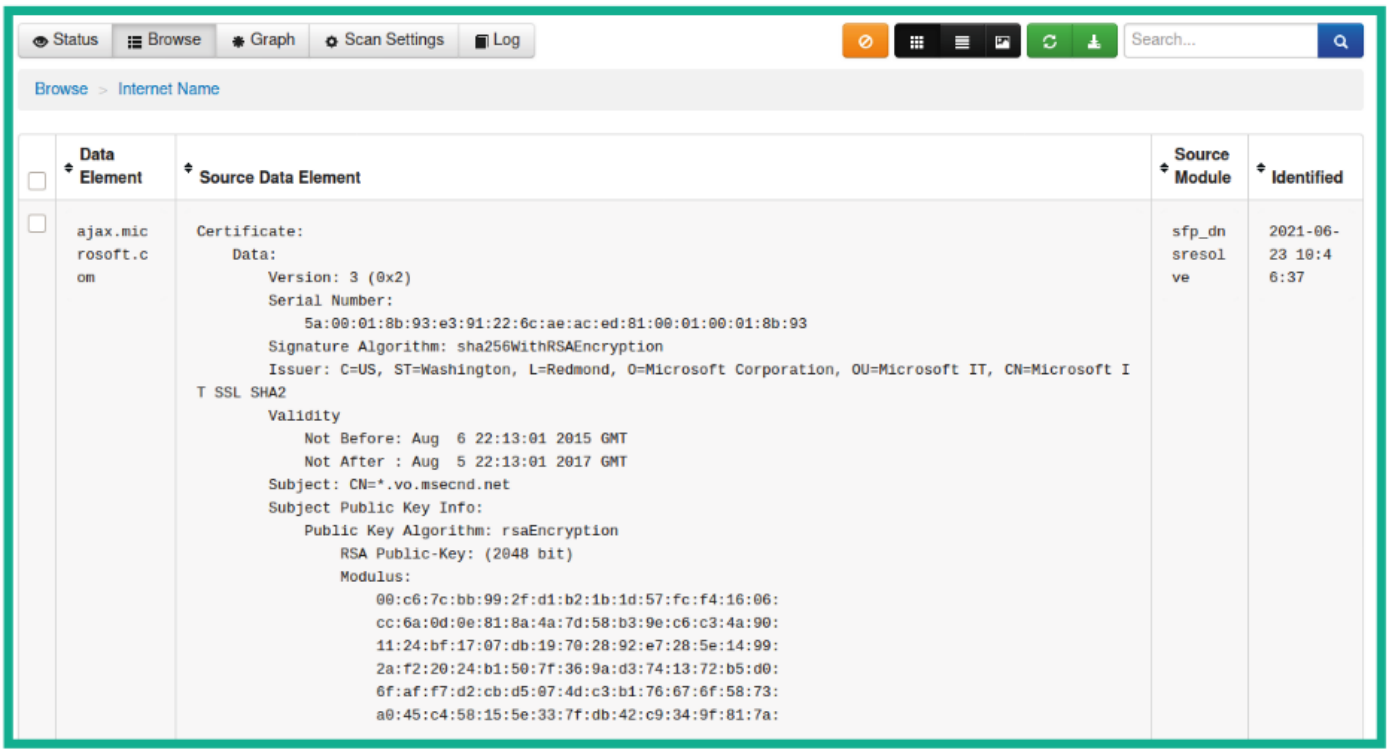


Figure 5.27 – Source data elements

12. Next, click on **Browse** > **RAW DNS Records** to view the DNS information that was collected by Spiderfoot.

4. Next, open the web browser within Kali Linux, enter the IP address of your Kali Linux machine within the address bar, and hit *Enter*.
5. Once the Spiderfoot web user interface loads, click on **Settings**, as shown here:

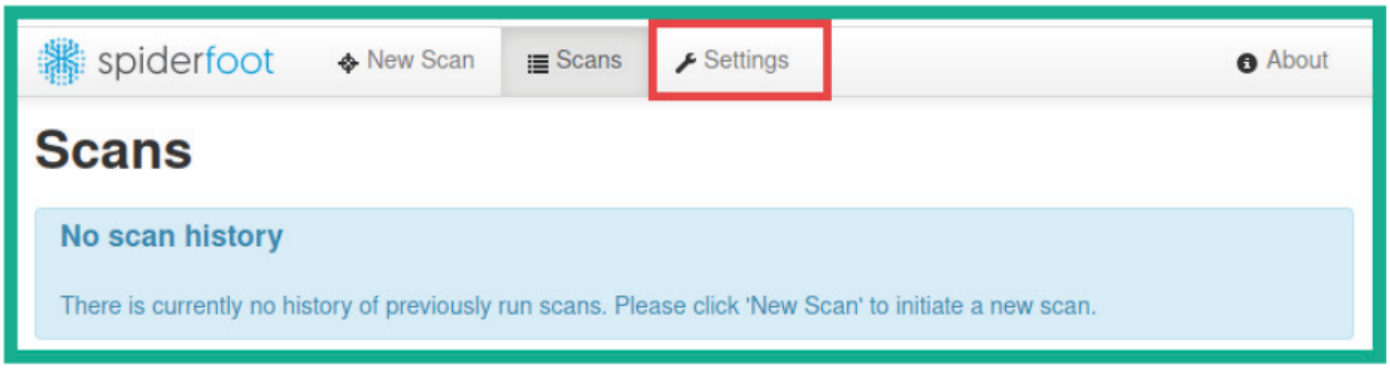


Figure 5.20 – Spiderfoot web interface

Spiderfoot can gather information from a wide range of online sources. However, some of these sources will require an **Application Programming Interface** (**API**) key to allow Spiderfoot to perform queries on some sources. The sources that require an API key are indicated with a lock icon next to their names, as shown here:

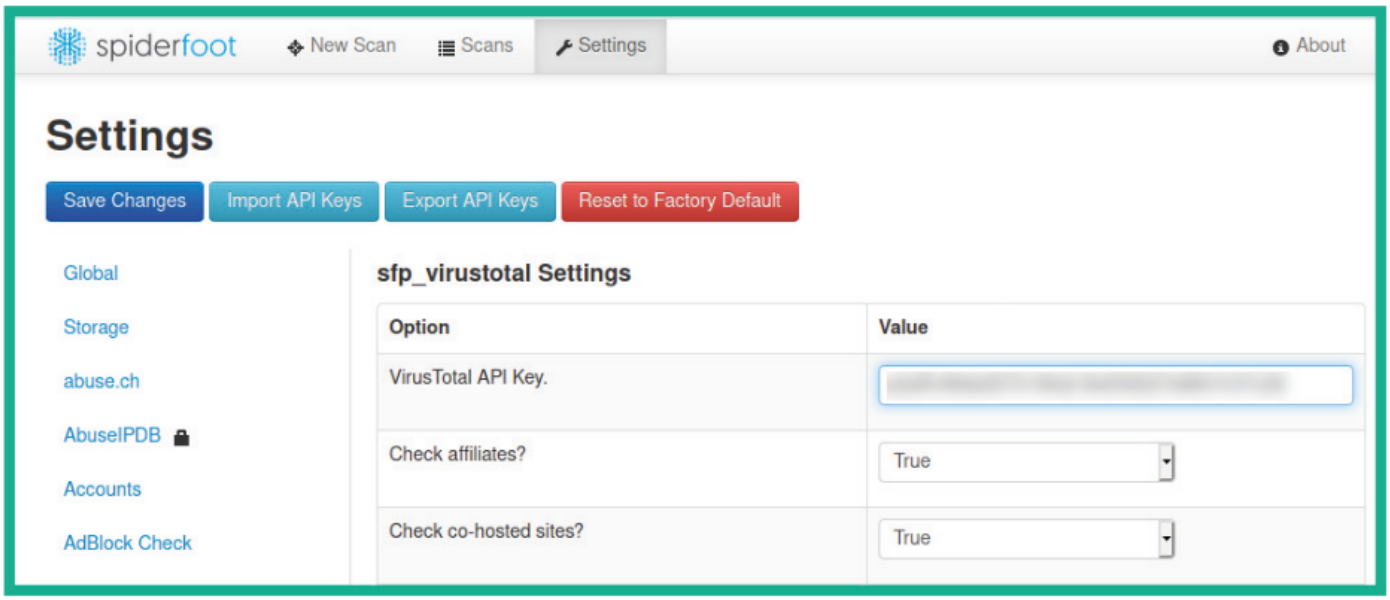


Figure 5.21 – Sources

Keep in mind that Spiderfoot works better when the API keys have been configured within its **Settings** menu. Many of these OSINT sources provide an API key if you register for a free account on their website. Do take the time to register on a couple of these online sources/websites and simply insert your unique API key into the Spiderfoot **Settings** menu.

6. Next, to start automating the OSINT gathering process, simply click on **New Scan**, as shown here:

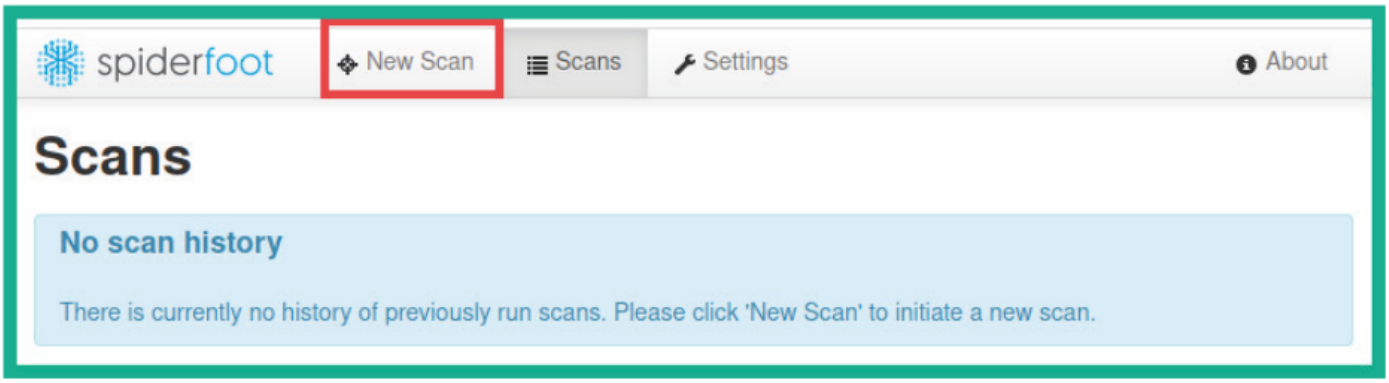


Figure 5.22 – New Scan

7. Next, insert a name for your scan and set the target domain within the **Seed Target** field, as shown here:

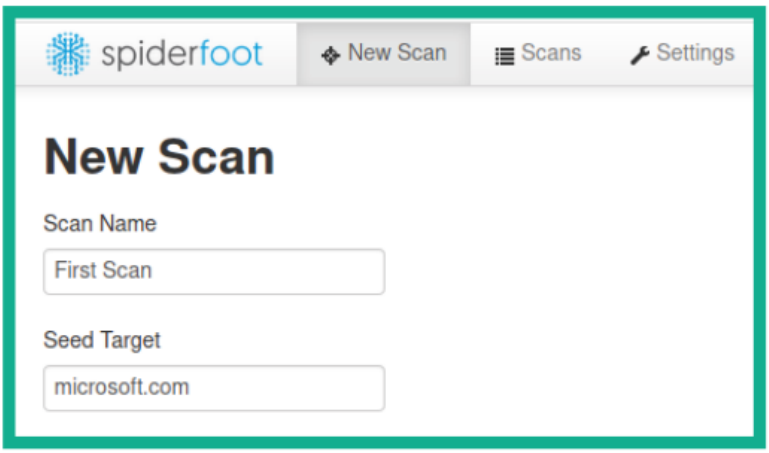
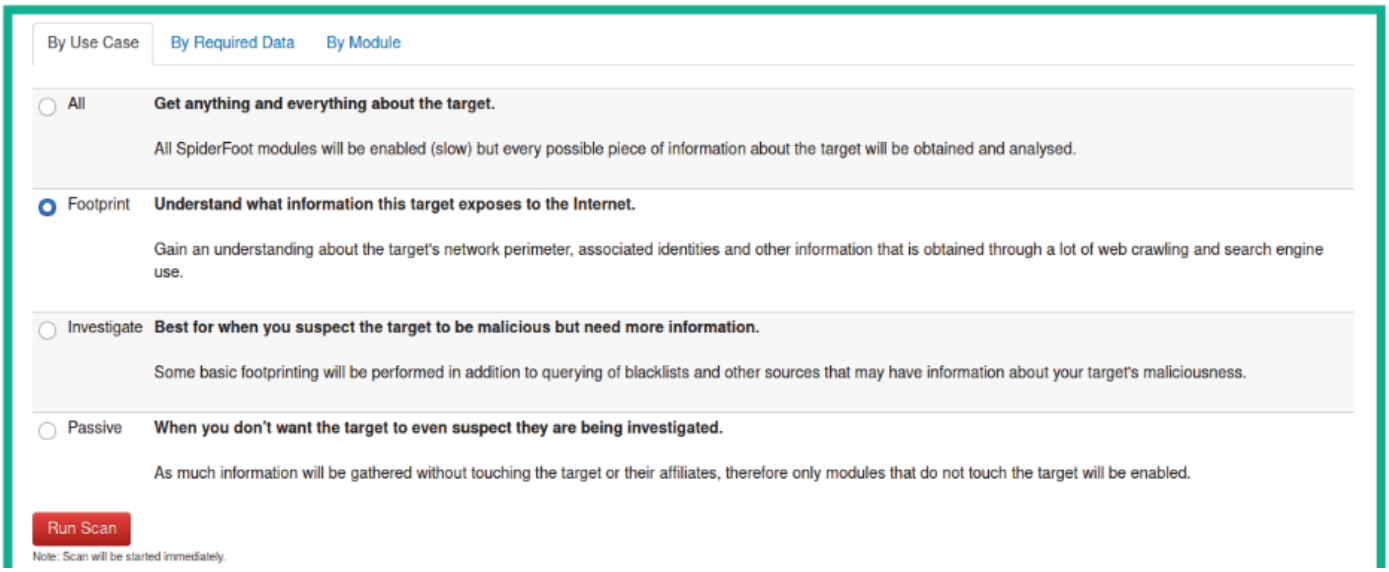


Figure 5.23 – Setting a target

8. Next, on the low section of the **New Scan** menu, you will be provided with various techniques Spiderfoot can use to retrieve information about your target:



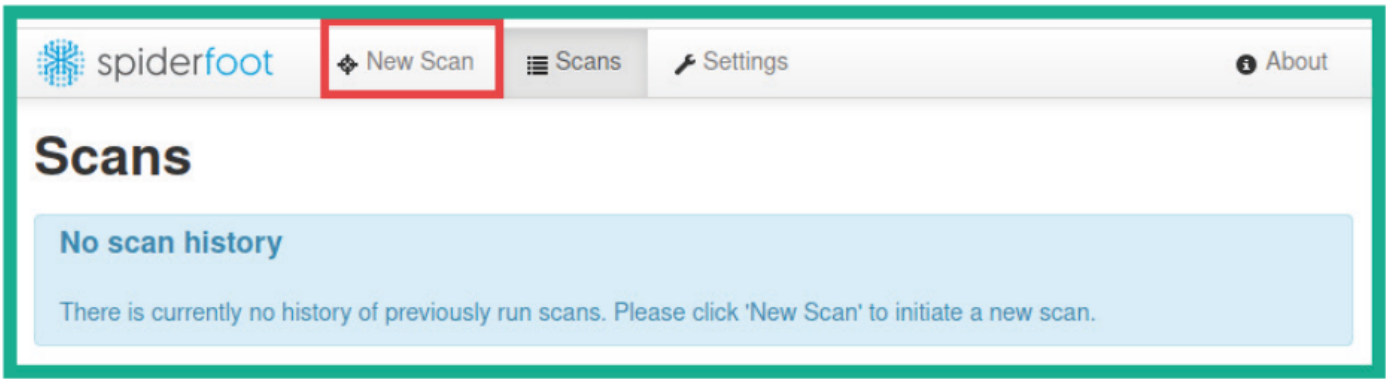


Figure 5.22 – New Scan

7. Next, insert a name for your scan and set the target domain within the **Seed Target** field, as shown here:

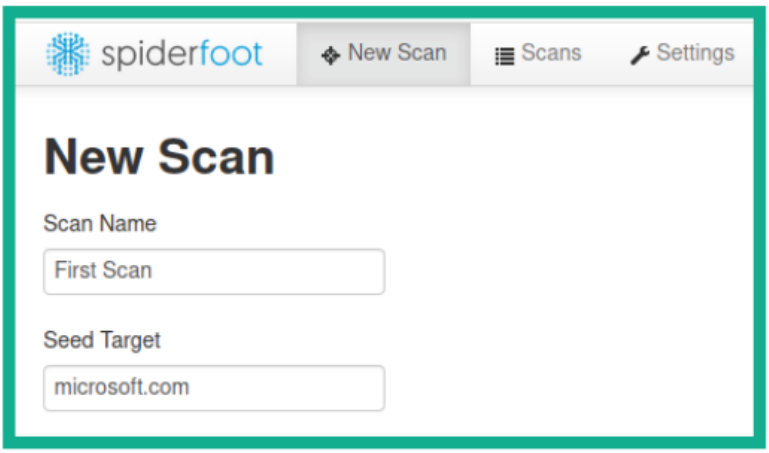
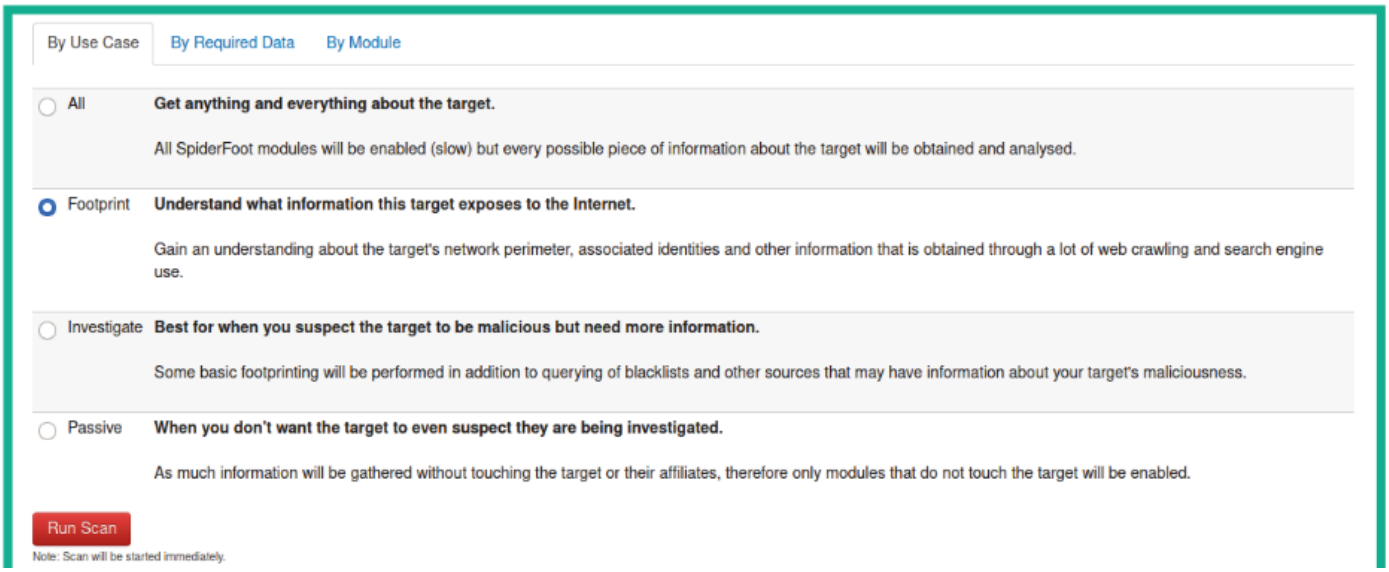


Figure 5.23 – Setting a target

8. Next, on the low section of the **New Scan** menu, you will be provided with various techniques Spiderfoot can use to retrieve information about your target:



While there are so many possibilities when using Google search operators, it can be a bit overwhelming. **Google Hacking Database** (**GHDB**) is maintained by the creators of Kali Linux, *Offensive Security* (`https://www.offensive-security.com`), and can be found at `https://www.exploit-db.com/google-hacking-database`. This website contains a list of various Google dorks (search operators), which are used to find very sensitive information on the internet using Google Search:

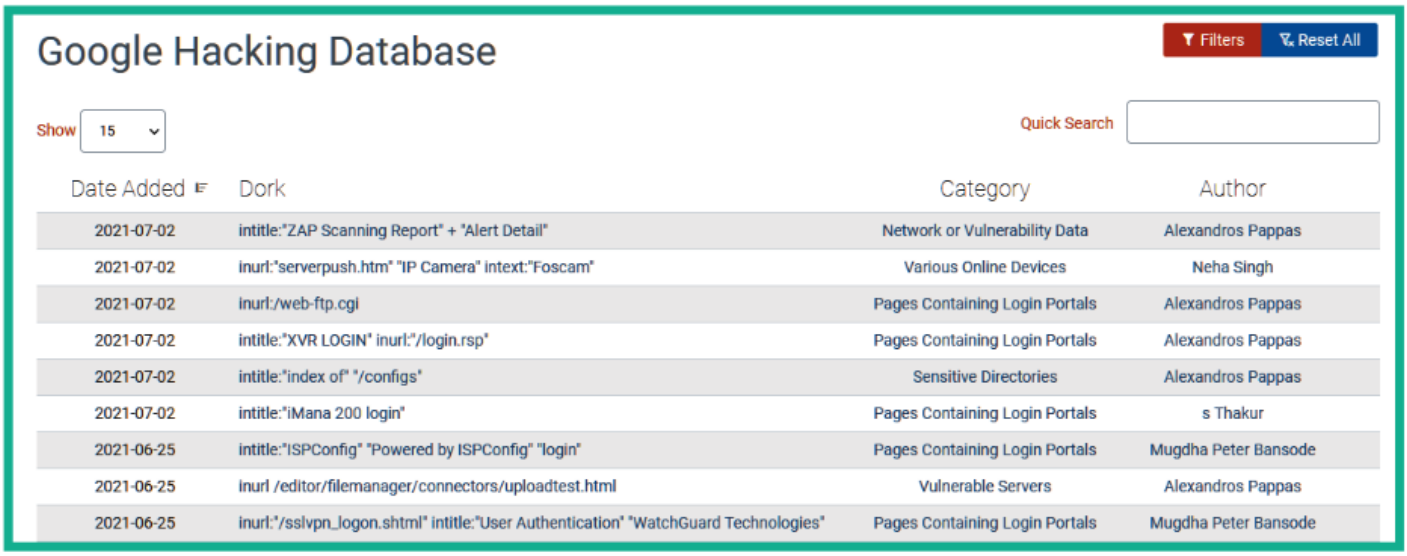


Figure 5.9 – Google Hacking Database

As shown in the preceding screenshot, GHDB is regularly updated with new techniques to help users discover vulnerable services and sensitive directories. A word of caution, though – please be very mindful and careful when lurking around using Google hacking techniques. Do not use the information you find for malicious purposes or to cause harm to a system or network.

Having completed this section, you have learned how ethical hackers and penetration testers can leverage the power of Google Search to discover hidden directories and resources. In the next section, you will learn how to perform DNS reconnaissance.

threats/vulnerablilities

and here it goes

As an entry-level security analyst, one of your many roles will be to handle
an organization's digital and physical assets.
As a reminder,
an asset is an item perceived as having value to an organization.
During their lifespan, organizations acquire all types of assets, including
physical office spaces, computers, customers' PII, intellectual property,

such as patents or copyrighted data, and so much more.
Unfortunately, organizations operate in an environment that presents multiple security threats, risks, and vulnerabilities to their assets.
Let's review what threats, risks, and vulnerabilities are and discuss some common examples of each.

A threat is any circumstance or event that can negatively impact assets.
One example of a threat is a social engineering attack.
Social engineering is a manipulation technique that exploits human error to gain private information, access, or valuables.
Malicious links in email messages that look like they're from legitimate companies or people is one method of social engineering known as phishing.
As a reminder, phishing is a technique that is used to acquire sensitive data, such as user names, passwords, or banking information.

Risks are different from threats.
A risk is anything that can impact the confidentiality, integrity, or availability of an asset.
Think of a risk as the likelihood of a threat occurring.
An example of a risk to an organization might be the lack of backup protocols for making sure its stored information can be recovered in the event of an accident or security incident.
Organizations tend to rate risks at different levels: low, medium, and high, depending on possible threats and the value of an asset.

A low-risk asset is information that would not harm the organization's reputation or ongoing operations, and would not cause financial damage if compromised.
This includes public information such as website content, or published research data.

A medium-risk asset might include information that's not available to the public and may cause some damage to the organization's finances, reputation, or ongoing operations.
For example, the early release of a company's quarterly earnings could impact the value of their stock.

A high-risk asset is any information protected by regulations or laws,
which if compromised, would have a severe negative impact on
an organization's finances, ongoing operations, or reputation.
This could include leaked assets with SPII, PII,
or intellectual property.

Now, let's discuss vulnerabilities.
A vulnerability is a weakness that can be exploited by a threat.
And it's worth noting that both a vulnerability and
threat must be present for there to be a risk.
Examples of vulnerabilities include: an outdated firewall, software, or
application; weak passwords; or unprotected confidential data.
People can also be considered a vulnerability.
People's actions can significantly affect an organization's internal network.
Whether it's a client, external vendor, or employee,
maintaining security must be a united effort.

So entry-level analysts need to educate and
empower people to be more security conscious.
For example, educating people on how to identify a phishing email is a great starting point.
Using access cards to grant employee access to physical spaces while
restricting outside visitors is another good security measure.
Organizations must continually improve their efforts when it comes to
identifying and mitigating vulnerabilities to minimize threats and risks.
Entry-level analysts can support this goal by encouraging employees to
report suspicious activity and actively monitoring and
documenting employees' access to critical assets.

Now that you're familiar with some of the threats, risks, and
vulnerabilities analysts frequently encounter,
coming up, we'll discuss how they impact business operations.

: Added to Selection. Press [CTRL + S] to save as a note

ransomwear

Ransomware is a malicious attack where threat actors encrypt
an organization's data then demand payment to restore access.
Once ransomware is deployed by an attacker,
it can freeze network systems, leave devices unusable, and
encrypt, or lock confidential data, making devices inaccessible.
The threat actor then demands a ransom before providing a decryption key
to allow organizations to return to their normal business operations.
Think of a decryption key as a password provided to regain access to your data.
Note that when ransom negotiations occur or data is leaked by threat actors,
these events can occur through the dark web.

While many people use search engines to navigate to their social media accounts or
to shop online, this is only a small part of what the web really is.
The web is actually an interlinked network of online content that's
made up of three layers: the surface web, the deep web, and the dark web.

The surface web is the layer that most people use.
It contains content that can be accessed using a web browser.

The deep web generally requires authorization to access it.
An organization's intranet is an example of the deep web,
since it can only be accessed by employees or others who have been granted access.

Lastly, the dark web can only be accessed by using special software.
The dark web generally carries a negative connotation since it is the preferred
web layer for criminals because of the secrecy that it provides.

Now, let's discuss three key impacts of threats, risks, and vulnerabilities.

The first impact we'll discuss is financial impact.
When an organization's assets are compromised by an attack, such as the use of malware, the financial consequences can be significant for a variety of reasons.
These can include interrupted production and services, the cost to correct the issue, and fines if assets are compromised because of non-compliance with laws and regulations.

The second impact is identity theft.
Organizations must decide whether to store private customer, employee, and outside vendor data, and for how long.
Storing any type of sensitive data presents a risk to the organization.
Sensitive data can include personally identifiable information, or PII, which can be sold or leaked through the dark web.
That's because the dark web provides a sense of secrecy and threat actors may have the ability to sell data there without facing legal consequences.

The last impact we'll discuss is damage to an organization's reputation.
A solid customer base supports an organization's mission, vision, and financial goals.
An exploited vulnerability can lead customers to seek new business relationships with competitors or create bad press that causes permanent damage to an organization's reputation.
The loss of customer data doesn't only affect an organization's reputation and financials, it may also result in legal penalties and fines.
Organizations are strongly encouraged to take proper security measures and follow certain protocols to prevent the significant impact of threats, risks, and vulnerabilities.
By using all the tools in their toolkit, security teams are better prepared to handle an event such as a ransomware attack.

Coming up, we'll cover the NIST risk management framework's seven steps for managing risk.

••

As you might remember from earlier in the program,
the National Institute of Standards and Technology,
NIST, provides many frameworks that are used by
security professionals to manage
risks, threats, and vulnerabilities.

In this video, we're going to focus on
NIST's Risk Management Framework or RMF.
As an entry-level analyst,
you may not engage in all of these steps,
but it's important to be familiar with this framework.
Having a solid foundational understanding
of how to mitigate and manage risks can
set yourself apart from other candidates as you
begin your job search in the field of security.

There are seven steps in the RMF: prepare, categorize,
select, implement, assess, authorize, and monitor.

Let's start with Step one, prepare.
Prepare refers to activities that are necessary to manage
security and privacy risks before a breach occurs.
As an entry-level analyst,
you'll likely use this step to monitor for risks and
identify controls that can be used to reduce those risks.

Step two is categorize,
which is used to develop
risk management processes and tasks.
Security professionals then use those processes and
develop tasks by thinking about how the confidentiality,
integrity, and availability of
systems and information can be impacted by risk.
As an entry-level analyst,
you'll need to be able to understand how
to follow the processes established
by your organization to reduce risks to
critical assets, such as private customer information.

Step three is select.
Select means to choose, customize,
and capture documentation of
the controls that protect an organization.
An example of the select step would be keeping
a playbook up-to-date or helping to
manage other documentation that allows you and
your team to address issues more efficiently.

Step four is to implement
security and privacy plans for the organization.
Having good plans in place is essential for
minimizing the impact of ongoing security risks.
For example, if you notice a pattern of
employees constantly needing password resets,
implementing a change to
password requirements may help solve this issue.

]
Step five is assess.
Assess means to determine if
established controls are implemented correctly.
An organization always wants
to operate as efficiently as possible.
So it's essential to take the time to
analyze whether the implemented protocols,
procedures, and controls that are in
place are meeting organizational needs.

During this step, analysts identify
potential weaknesses and determine
whether the organization's tools,
procedures, controls,
and protocols should be
changed to better manage potential risks.

Step six is authorize.
Authorize means being accountable for
the security and privacy risks that
may exist in an organization.
As an analyst,
the authorization step could involve generating reports,
developing plans of action,
and establishing project milestones that are
aligned to your organization's security goals.

Step seven is monitor.
Monitor means to be aware of how systems are operating.
Assessing and maintaining technical operations
are tasks that analysts complete daily.
Part of maintaining a low level of
risk for an organization is
knowing how the current systems support
the organization's security goals.
If the systems in place don't meet those goals,
changes may be needed.

Although it may not be
your job to establish these procedures,
you will need to make sure
they're working as intended so that
risks to the organization itself,
and the people it serves, are minimized.

nmap

3. Next, let's use **Network Mapper** (**Nmap**) to scan the entire network while excluding our Kali Linux machine by using the following command:

```
kali@kali:~$ nmap -sn 172.30.1.0/24 --exclude 172.30.1.27
```

Using the `-sn` syntax ensures Nmap performs a ping sweep of the network. This means Nmap will send an **Internet Control Message Protocol** (**ICMP**) **Echo Request** message to all devices within the network range. Online devices will typically respond with an **ICMP Echo Reply** message. Then, Nmap will provide the results of the systems that are online, as shown here:

```
kali@kali:~$ nmap -sn 172.30.1.0/24 --exclude 172.30.1.27
Starting Nmap 7.91 ( https://nmap.org ) at 2021-06-23 08:41 EDT
Nmap scan report for 172.30.1.26
Host is up (0.0057s latency).
Nmap done: 255 IP addresses (1 host up) scanned in 15.78 seconds
```

Figure 5.37 – Ping sweep using Nmap

As shown in the preceding screenshot, Nmap indicated that the `172.30.1.26` host device (Metasploitable 2) is currently up (online). Furthermore, using the `--exclude` command allows us to specify which IP addresses to exclude from scanning. This command is best used when you are restricted from scanning various IP addresses and subnetworks during a penetration test.

Next, you will learn how to use Nmap to discover open ports and running services on a target system on a network.

Probing open service ports, services, and operating systems

After discovering the hosts on a network, the next phase is to identify any open service ports on the target system and determine which services are mapped to those open ports. There are various techniques that a penetration tester can use to identify the open ports on a target system. Some techniques are manual, while others can simply be automated using the Nmap tool. Let's take a look:

1. Since we have already discovered a live system on our network, let's use the following command to perform a basic Nmap scan of a host (Metasploitable 2):

```
kali@kali:~$ nmap 172.30.1.26
```

This information can be used to research for vulnerabilities and exploits of the target. Simply put, we can see that the service version for the **File Transfer Protocol** (**FTP**) service is using `vsftpd2.3.4`. With a bit of *Google Fu*, you will find this link, which provides details about the security vulnerabilities for this specific service version: `https://www.rapid7.com/db/modules/exploit/unix/ftp/vsftpd_234_backdoor/`.

The following screenshot shows another section of the advanced scan results:

```
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE:

Host script results:
|_clock-skew: mean: 59m58s, deviation: 2h00m01s, median: -2s
|_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: metasploitable
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
|_  System time: 2021-06-23T08:55:02-04:00
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_smb2-time: Protocol negotiation failed (SMB2)
```

Figure 5.40 – Operating system profiling

As shown in the preceding screenshot, Nmap was able to identify that the target's operating system is Linux-based, the hostname, and the **Server Message Block** (**SMB**) version of the target machine.

> **Important Note**
>
> SMB is a TCP/IP network protocol that is used to allow file and printer sharing services between host devices on a network. Discovering SMB on a host system is an indication there many a file share located on the target system, and it's something worth checking out.

The following is some additional syntax that can be used with Nmap to gather specific information:

- `-Pn`: This command performs a scan on the target without sending an ICMP Echo Request (ping) message. This command is useful for scanning systems that have ICMP responses disabled.
- `-sU`: This command allows Nmap to perform a UDP port scan on the target. This command is useful for identifying any services that use UDP compared to TCP.

- `-p <port ranges>`: This command allows a penetration tester to scan a single port or range such as `-p80`, `-p 80,443,8080`, or `-p 100-200`.
- `-sV`: This command allows Nmap to send special probes to identify the service versions of any open ports on the target system.
- `-O`: This command allows Nmap to identify and profile the operating system on the target system.
- `-6`: This command enables Nmap to perform scanning on a system or network that has an IPv6 address.

By identifying the operating systems of targets, penetration testers can create an exploit and payload that are designed to work efficiently on those specific operating systems. Simply put, an exploit or payload for a Windows operating system will most likely not work on a Linux-based system and vice versa. Thus far, you have learned how to discover open ports, service versions, operating system, and SMB versions. Next, you will learn how to evade detection while performing active scanning on a network and systems using Nmap.

Working with evasion techniques

Whenever a packet is sent from one device to another, the source IP address is included within the header of the packet. This is the default behavior of the TCP/IP protocol stack; all address details must be included within all packets that need to traverse a network. When performing a scan as an ethical hacker and a penetration tester, we try to remain undetected by our target organization. During a real cyberattack, if an organization is unable to detect suspicious activities on their network and systems, the threat actor can simply achieve their objectives without obstructions. However, if an organization can detect suspicious activities as soon as they occur, the blue team can take action quickly to stop the potential threat and safeguard their systems.

During a penetration test, it's important to simulate a real-world cyberattack to test the threat detection systems within the target organization. While many organizations with security solutions help protect them from cyberattacks and threats, not all network devices and security solutions are configured properly, so they may not detect a penetration tester on their network.

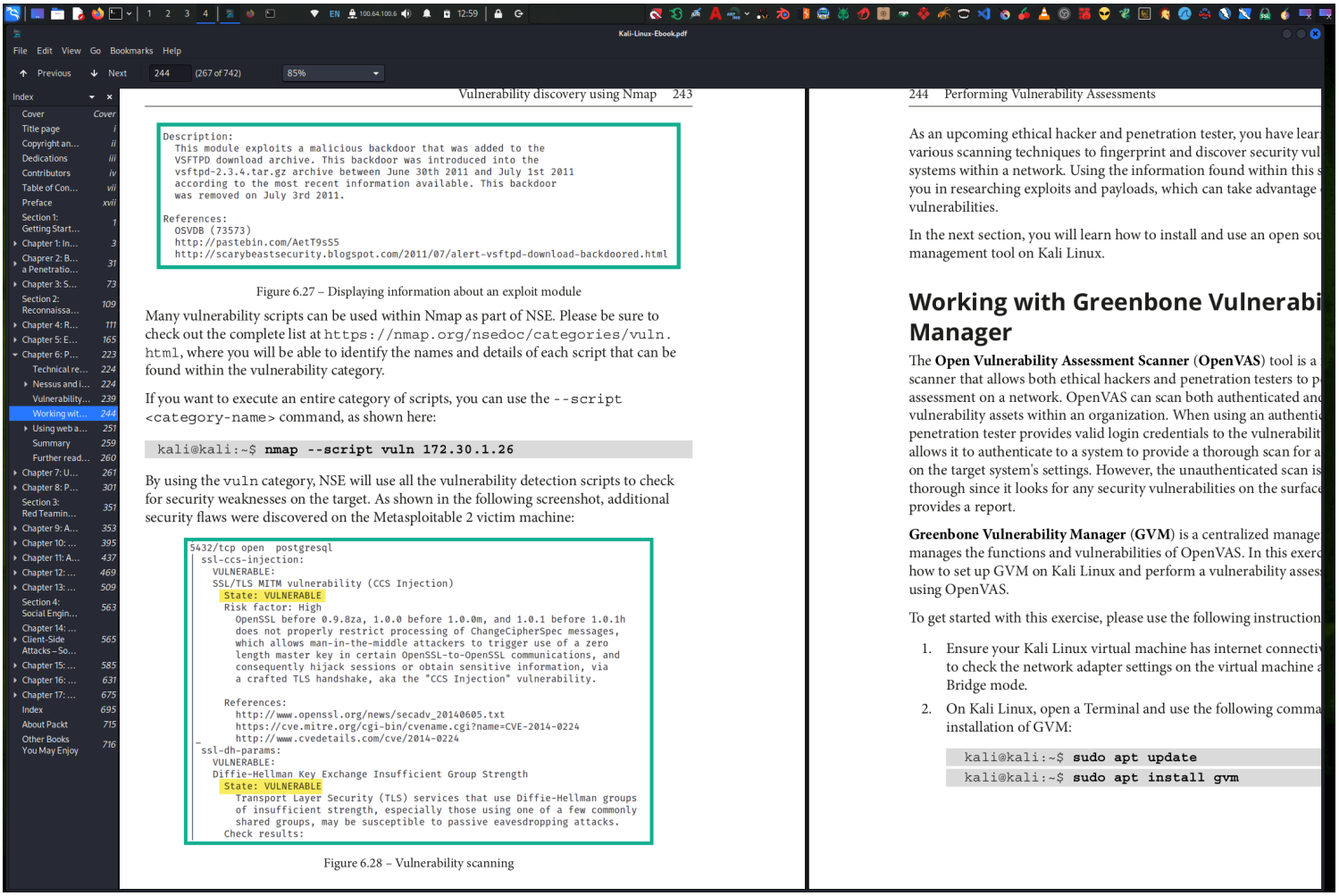
- **Brute**: This category contains scripts that are used to perform some types of brute-force attacks on a remote server to gain unauthorized access.
- **Default**: This category contains a set of default scripts within NSE for scanning.
- **Discovery**: This category contains scripts that are used in active information gathering regarding network services on a target.
- **"DoS"**: This category contains scripts that can simulate a **Denial-of-Service** (**DoS**) attack on a target to check whether the target is susceptible to such types of attacks.
- **Exploit**: This category contains scripts that are used to actively exploit security vulnerabilities on a target.
- **External**: This category contains scripts that usually send data that's been gathered from a target to an external resource for further processing.
- **Fuzzer**: This category contains scripts that are used to send random data into an application to discover any software bugs and vulnerabilities within applications.
- **Intrusive**: This category contains high-risk scripts that can crash systems and cause data loss.
- **Malware**: This category contains scripts that can determine whether a target is infected with malware.
- **Safe**: This category contains scripts that are not intrusive and safe to use on a target system.
- **Version**: This category contains scripts that are used to gather the version information of services on a target system.
- **Vuln**: This category contains scripts that are used to check for specific vulnerabilities on a target system.

> **Important Note**
> To learn more about the functionality of each script that's available within NSE, please see the official website at `https://nmap.org/nsedoc/`.

To perform a scan using a specific script on the Metasploitable 2 machine, use the following commands:

```
kali@kali:~$ nmap --script ftp-vsftpd-backdoor 172.30.1.26
```

The `--script` command allows you to specify either a single script, multiple scripts, or a category of scripts. The following screenshot shows the results of performing a scan on our victim machine:


Kali-Linux-Ebook.pdf
File Edit View Go Bookmarks Help
Previous Next 244 (267 of 742) 85%
Index
Cover
Title page
Copyright an...
Dedications
Contributors
Table of Con...
Preface
Section 1: Getting Start...
Chapter 1: In...
Chapter 2: B... a Penetratio...
Chapter 3: S...
Section 2: Reconnaissa...
Chapter 4: R...
Chapter 5: E...
Chapter 6: P...
Technical re...
Nessus and i...
Vulnerability...
Working wit...
Using web a...
Summary
Further read...
Chapter 7: U...
Chapter 8: P...
Section 3: Red Teamin...
Chapter 9: A...
Chapter 10: ...
Chapter 11: A...
Chapter 12: ...
Chapter 13: ...
Section 4: Social Engin...
Chapter 14: ... Client-Side Attacks – So...
Chapter 15: ...
Chapter 16: ...
Chapter 17: ...
Index
About Packt
Other Books You May Enjoy
Vulnerability discovery using Nmap 243
Description:
This module exploits a malicious backdoor that was added to the
VSFTPD download archive. This backdoor was introduced into the
vsftpd-2.3.4.tar.gz archive between June 30th 2011 and July 1st 2011
according to the most recent information available. This backdoor
was removed on July 3rd 2011.
References:
OSVDB (73573)
http://pastebin.com/AetT9sS5
http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html
Figure 6.27 – Displaying information about an exploit module
Many vulnerability scripts can be used within Nmap as part of NSE. Please be sure to check out the complete list at https://nmap.org/nsedoc/categories/vuln.html, where you will be able to identify the names and details of each script that can be found within the vulnerability category.
If you want to execute an entire category of scripts, you can use the --script <category-name> command, as shown here:
kali@kali:~$ nmap --script vuln 172.30.1.26
By using the vuln category, NSE will use all the vulnerability detection scripts to check for security weaknesses on the target. As shown in the following screenshot, additional security flaws were discovered on the Metasploitable 2 victim machine:
5432/tcp open postgresql
ssl-ccs-injection:
VULNERABLE:
SSL/TLS MITM vulnerability (CCS Injection)
State: VULNERABLE
Risk factor: High
OpenSSL before 0.9.8za, 1.0.0 before 1.0.0m, and 1.0.1 before 1.0.1h
does not properly restrict processing of ChangeCipherSpec messages,
which allows man-in-the-middle attackers to trigger use of a zero
length master key in certain OpenSSL-to-OpenSSL communications, and
consequently hijack sessions or obtain sensitive information, via
a crafted TLS handshake, aka the "CCS Injection" vulnerability.
References:
http://www.openssl.org/news/secadv_20140605.txt
https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0224
http://www.cvedetails.com/cve/2014-0224
ssl-dh-params:
VULNERABLE:
Diffie-Hellman Key Exchange Insufficient Group Strength
State: VULNERABLE
Transport Layer Security (TLS) services that use Diffie-Hellman groups
of insufficient strength, especially those using one of a few commonly
shared groups, may be susceptible to passive eavesdropping attacks.
Check results:
Figure 6.28 – Vulnerability scanning
244 Performing Vulnerability Assessments
As an upcoming ethical hacker and penetration tester, you have lear
various scanning techniques to fingerprint and discover security vul
systems within a network. Using the information found within this s
you in researching exploits and payloads, which can take advantage
vulnerabilities.
In the next section, you will learn how to install and use an open sou
management tool on Kali Linux.
Working with Greenbone Vulnerabi Manager
The Open Vulnerability Assessment Scanner (OpenVAS) tool is a
scanner that allows both ethical hackers and penetration testers to p
assessment on a network. OpenVAS can scan both authenticated an
vulnerability assets within an organization. When using an authenti
penetration tester provides valid login credentials to the vulnerabilit
allows it to authenticate to a system to provide a thorough scan for a
on the target system's settings. However, the unauthenticated scan is
thorough since it looks for any security vulnerabilities on the surface
provides a report.
Greenbone Vulnerability Manager (GVM) is a centralized manage
manages the functions and vulnerabilities of OpenVAS. In this exerc
how to set up GVM on Kali Linux and perform a vulnerability asses
using OpenVAS.
To get started with this exercise, please use the following instruction
1. Ensure your Kali Linux virtual machine has internet connecti
to check the network adapter settings on the virtual machine a
Bridge mode.
2. On Kali Linux, open a Terminal and use the following comma
installation of GVM:
kali@kali:~$ sudo apt update
kali@kali:~$ sudo apt install gvm

gvm

As an upcoming ethical hacker and penetration tester, you have learned how to perform various scanning techniques to fingerprint and discover security vulnerabilities on host systems within a network. Using the information found within this section can help you in researching exploits and payloads, which can take advantage of these security vulnerabilities.

In the next section, you will learn how to install and use an open source vulnerability management tool on Kali Linux.

Working with Greenbone Vulnerability Manager

The **Open Vulnerability Assessment Scanner** (**OpenVAS**) tool is a free vulnerability scanner that allows both ethical hackers and penetration testers to perform a vulnerability assessment on a network. OpenVAS can scan both authenticated and unauthenticated vulnerability assets within an organization. When using an authenticated scan, the penetration tester provides valid login credentials to the vulnerability scanner, which allows it to authenticate to a system to provide a thorough scan for any misconfigurations on the target system's settings. However, the unauthenticated scan is usually not as thorough since it looks for any security vulnerabilities on the surface of the target and provides a report.

Greenbone Vulnerability Manager (**GVM**) is a centralized management tool that manages the functions and vulnerabilities of OpenVAS. In this exercise, you will learn how to set up GVM on Kali Linux and perform a vulnerability assessment on a target using OpenVAS.

To get started with this exercise, please use the following instructions:

1. Ensure your Kali Linux virtual machine has internet connectivity. You may need to check the network adapter settings on the virtual machine and configure it to Bridge mode.
2. On Kali Linux, open a Terminal and use the following commands to start the installation of GVM:

```
kali@kali:~$ sudo apt update
kali@kali:~$ sudo apt install gvm
```

3. Once the installation is complete, use the following command to begin the setup process of GVM:

```
kali@kali:~$ sudo gvm-setup
```

The following screenshot shows the initialization process and creating the user account for GVM:

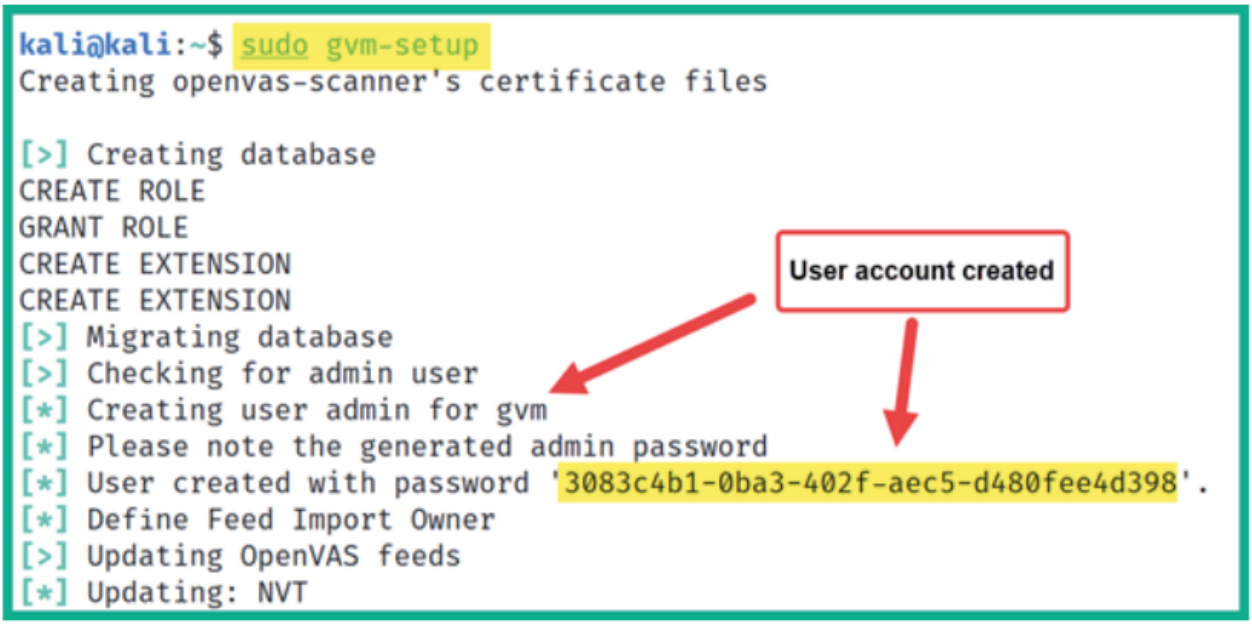


Figure 6.29 – The GVM setup process

This process usually takes some time to complete as GVM downloads updates from its online repository. Once the initialization process is complete, it will provide the username and password once more, as shown here:

```
[*] Checking Default scanner
08b69003-5fc2-4037-a479-93b440211c73  OpenVAS  /var/run/ospd/ospd.sock  0  OpenVAS Default

[+] Done
[*] Please note the password for the admin user
[*] User created with password '3083c4b1-0ba3-402f-aec5-d480fee4d398'.
```

Figure 6.30 – User account

4. Next, use the `sudo gvm-start` command to start the GVM service.

Manage common threats, risks, and vulnerabilities

Previously, you learned that security involves protecting organizations and people from threats, risks, and vulnerabilities. Understanding the current threat landscapes gives organizations the ability to create policies and processes designed to help prevent and mitigate these types of security issues. In this reading, you will further explore how to manage risk and some common threat actor tactics and techniques, so you are better prepared to protect organizations and the people they serve when you enter the cybersecurity field.

Risk management

A primary goal of organizations is to protect assets. An **asset** is an item perceived as having value to an organization. Assets can be digital or physical. Examples of digital assets include the personal information of employees, clients, or vendors, such as:

- Social Security Numbers (SSNs), or unique national identification numbers assigned to individuals
- Dates of birth
- Bank account numbers
- Mailing addresses

Examples of physical assets include:

- Payment kiosks
- Servers
- Desktop computers
- Office spaces

Some common strategies used to manage risks include:

- **Acceptance**: Accepting a risk to avoid disrupting business continuity
- **Avoidance**: Creating a plan to avoid the risk altogether
- **Transference**: Transferring risk to a third party to manage
- **Mitigation**: Lessening the impact of a known risk

Additionally, organizations implement risk management processes based on widely accepted frameworks to help protect digital and physical assets from various threats, risks, and vulnerabilities. Examples of frameworks commonly used in the cybersecurity industry include the National Institute of Standards and Technology Risk Management Framework (NIST RMF) and Health Information Trust Alliance (HITRUST).

Following are some common types of threats, risks, and vulnerabilities you'll help organizations manage as a security professional.

Today's most common threats, risks, and vulnerabilities

Threats

A **threat** is any circumstance or event that can negatively impact assets. As an entry-level security analyst, your job is to help defend the organization's assets from inside and outside threats. Therefore, understanding common types of threats is important to an analyst's daily work. As a reminder, common threats include:

- **Insider threats:** Staff members or vendors abuse their authorized access to obtain data that may harm an organization.
- **Advanced persistent threats (APTs):** A threat actor maintains unauthorized access to a system for an extended period of time.

Risks

A **risk** is anything that can impact the confidentiality, integrity, or availability of an asset. A basic formula for determining the level of risk is that risk equals the likelihood of a threat. One way to think about this is that a risk is being late to work and threats are traffic, an accident, a flat tire, etc.

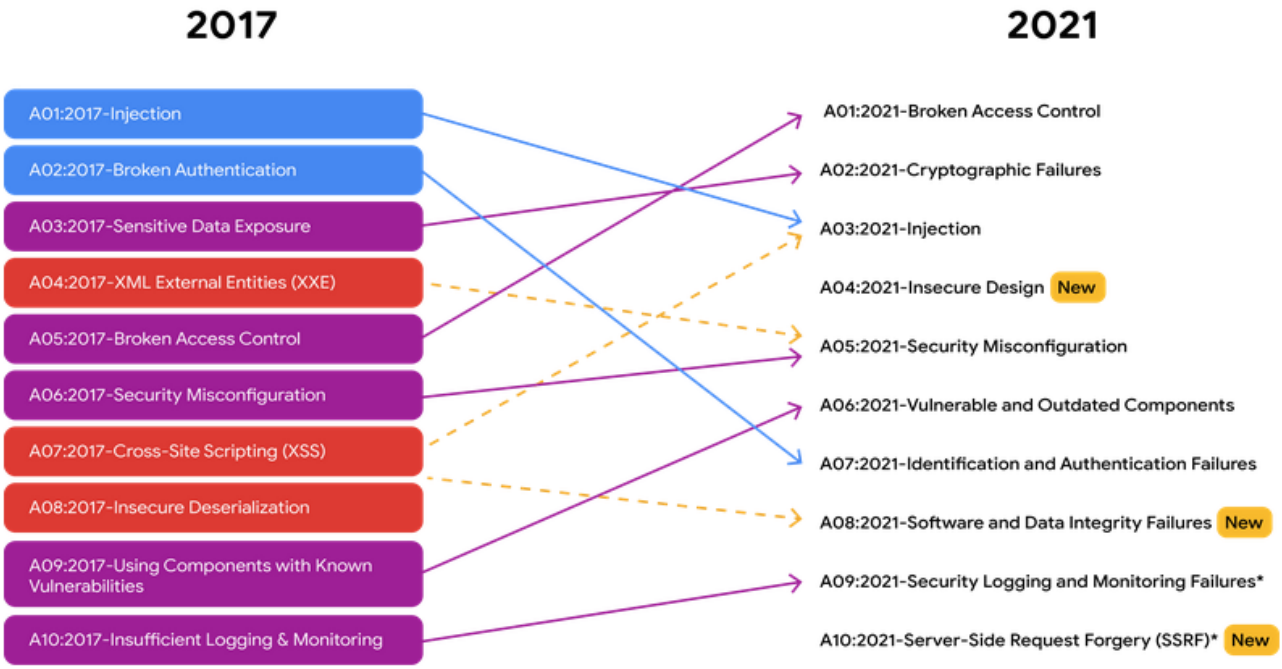
There are different factors that can affect the likelihood of a risk to an organization's assets, including:

- **External risk:** Anything outside the organization that has the potential to harm organizational assets, such as threat actors attempting to gain access to private information
- **Internal risk:** A current or former employee, vendor, or trusted partner who poses a security risk
- **Legacy systems:** Old systems that might not be accounted for or updated, but can still impact assets, such as workstations or old mainframe systems. For example, an organization might have an old vending machine that takes credit card payments or a workstation that is still connected to the legacy accounting system.
- **Multiparty risk:** Outsourcing work to third-party vendors can give them access to intellectual property, such as trade secrets, software designs, and inventions.
- **Software compliance/licensing:** Software that is not updated or in compliance, or patches that are not installed in a timely manner

There are many resources, such as the NIST, that provide lists of cybersecurity risks. Additionally, the Open Web Application Security Project (OWASP) publishes a standard awareness document about the top 10 most critical security risks to web applications, which is updated regularly.

Note: The OWASP's common attack types list contains three new risks for the years 2017 to 2021: insecure design, software and data integrity failures, and server-side request forgery. This update emphasizes the fact that security is a constantly evolving field. It also demonstrates the importance of staying up to date on current threat actor tactics and techniques, so you can be better prepared to manage these types of risks.

techniques, so you can be better prepared to manage these types of risks.



Vulnerabilities

A **vulnerability** is a weakness that can be exploited by a threat. Therefore, organizations need to regularly inspect for vulnerabilities within their systems. Some vulnerabilities include:

- **ProxyLogon:** A pre-authenticated vulnerability that affects the Microsoft Exchange server. This means a threat actor can complete a user authentication process to deploy malicious code from a remote location.
- **ZeroLogon:** A vulnerability in Microsoft's Netlogon authentication protocol. An authentication protocol is a way to verify a person's identity. Netlogon is a service that ensures a user's identity before allowing access to a website's location.
- **Log4Shell:** Allows attackers to run Java code on someone else's computer or leak sensitive information. It does this by enabling a remote attacker to take control of devices connected to the internet and run malicious code.
- **PetitPotam:** Affects Windows New Technology Local Area Network (LAN) Manager (NTLM). It is a theft technique that allows a LAN-based attacker to initiate an authentication request.
- **Security logging and monitoring failures:** Insufficient logging and monitoring capabilities that result in attackers exploiting vulnerabilities without the organization knowing it
- **Server-side request forgery:** Allows attackers to manipulate a server-side application into accessing and updating backend resources. It can also allow threat actors to steal data.

As an entry-level security analyst, you might work in vulnerability management, which is monitoring a system to identify and mitigate vulnerabilities. Although patches and updates may exist, if they are not applied, intrusions can still occur. For this reason, constant monitoring is important. The sooner an organization identifies a vulnerability and addresses it by patching it or updating their systems, the sooner it can be mitigated, reducing the organization's exposure to the vulnerability.

To learn more about the vulnerabilities explained in this section of the reading, as well as other vulnerabilities, explore the NIST National Vulnerability Database and CISA Known Exploited Vulnerabilities Catalog.

Key takeaways

In this reading, you learned about some risk management strategies and frameworks that can be used to develop organization-wide policies and processes to mitigate threats, risks, and vulnerabilities. You also learned about some of today's most common threats, risks, and vulnerabilities to business operations. Understanding these concepts can better prepare you to not only protect against, but also mitigate, the types of security-related issues that can harm organizations and people alike.

Resources for more information

To learn more, click the linked terms in this reading. Also, consider exploring the following sites:

- OWASP Top Ten
- NIST RMF

Mark as completed

Like Dislike Report an issue

nessus

The following screenshot shows the installation process:

```
kali@kali:~/Downloads$ sudo dpkg -i Nessus-8.15.0-debian6_amd64.deb
[sudo] password for kali:
Selecting previously unselected package nessus.
(Reading database ... 283829 files and directories currently installed.)
Preparing to unpack Nessus-8.15.0-debian6_amd64.deb ...
Unpacking nessus (8.15.0) ...
Setting up nessus (8.15.0) ...
Unpacking Nessus Scanner Core Components ...

 - You can start Nessus Scanner by typing /bin/systemctl start nessusd.service
 - Then go to https://kali:8834/ to configure your scanner
```

Figure 6.3 – Nessus installation process

5. Once the installation has been completed, use the following commands to start the Nessus service:

```
kali@kali:~/Downloads$ sudo /bin/systemctl start nessusd.service
```

6. Next, open a web browser, such as Firefox, and go to `https://kali:8834/` to initialize Nessus. You will get a security risk warning because the Nessus web interface is using a self-signed digital certificate, as shown here:

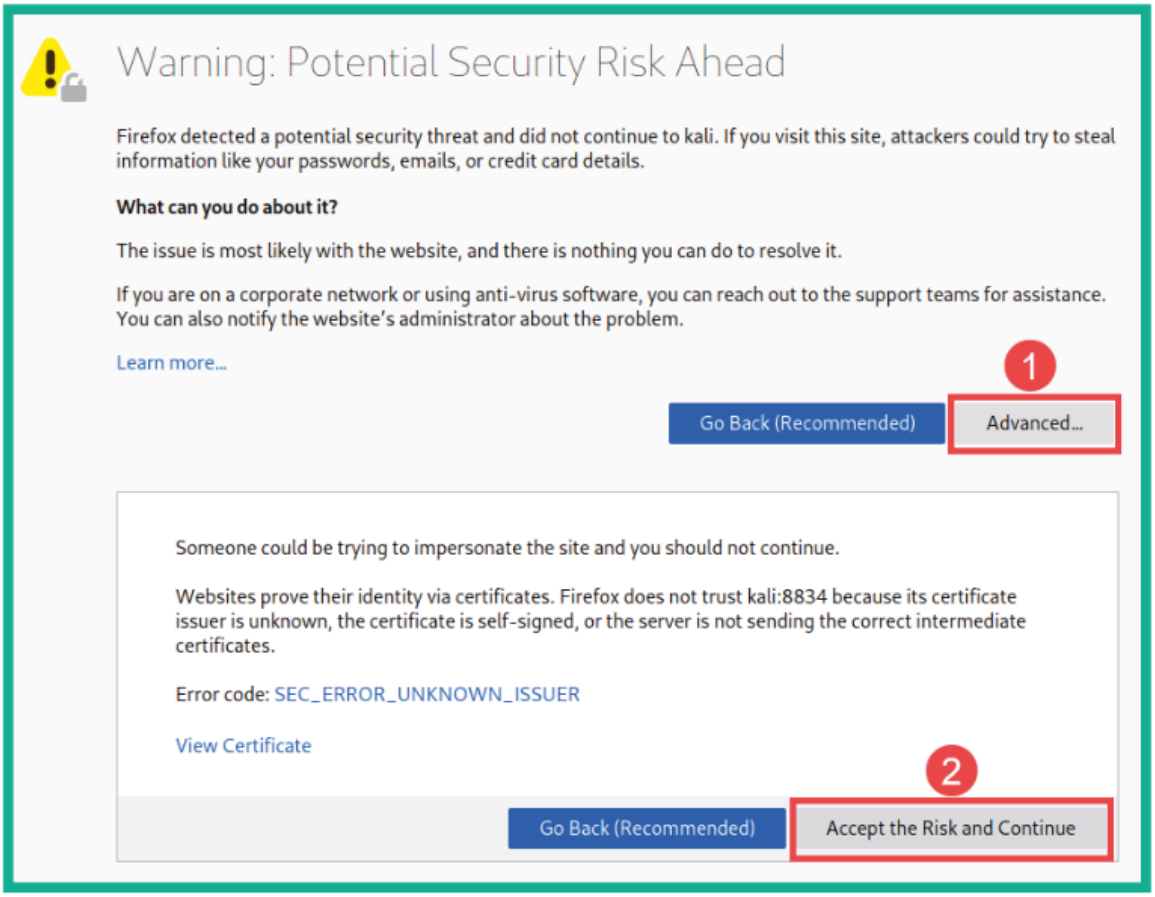


Figure 6.4 – Web browser security warning

week1glossary

Terms and definitions from Course 2, Week 1

Assess: The fifth step of the NIST RMF that means to determine if established controls are implemented correctly

Authorize: The sixth step of the NIST RMF that refers to being accountable for the security and privacy risks that may exist in an organization

Business continuity: An organization's ability to maintain their everyday productivity by establishing risk disaster recovery plans

Categorize: The second step of the NIST RMF that is used to develop risk management processes and tasks

External threat: Anything outside the organization that has the potential to harm organizational assets

Implement: The fourth step of the NIST RMF that means to implement security and privacy plans for an organization

Internal threat: A current or former employee, external vendor, or trusted partner who poses a security risk

Monitor: The seventh step of the NIST RMF that means be aware of how systems are operating

Prepare: The first step of the NIST RMF related to activities that are necessary to manage security and privacy risks before a breach occurs

Ransomware: A malicious attack where threat actors encrypt an organization's data and demand payment to restore access

Risk: Anything that can impact the confidentiality, integrity, or availability of an asset

Risk mitigation: The process of having the right procedures and rules in place to quickly reduce the impact of a risk like a breach

Security posture: An organization's ability to manage its defense of critical assets and data and react to change

Select: The third step of the NIST RMF that means to choose, customize, and capture documentation of the controls that protect an organization

Shared responsibility: The idea that all individuals within an organization take an active role in lowering risk and maintaining both physical and virtual security

Social engineering: A manipulation technique that exploits human error to gain private information, access, or valuables

Vulnerability: A weakness that can be exploited by a threat

week2

more frameworks

In an organization, plans are put in place to protect against a variety of threats, risks, and vulnerabilities. However, the requirements used to protect organizations and people often overlap. Because of this, organizations use security frameworks as a starting point to create their own security policies and processes.

Let's start by quickly reviewing what frameworks are. Security frameworks are guidelines used for building plans to help mitigate risks and threats to data and privacy, such as social engineering attacks and ransomware. Security involves more than just the virtual space. It also includes the physical, which is why many organizations have plans to maintain safety in the work environment. For example, access to a building may require using a key card or badge.

Other security frameworks provide guidance for how to prevent, detect, and respond to security breaches. This is particularly important when trying to protect an organization from social engineering attacks like phishing that target their employees.

Remember, people are the biggest threat to security. So frameworks can be used to create plans that increase employee awareness and educate them about how they can protect the organization, their co-workers, and themselves. Educating employees about existing security challenges is essential for minimizing the possibility of a breach.

Providing employee training about how to recognize red flags, or potential threats, is essential, along with having plans in place to quickly report and address security issues. As an analyst, it will be important for you to understand and implement the plans your organization has in place to keep the organization, its employees, and the people it serves safe from social engineering attacks, breaches, and other harmful security incidents.

Coming up, we'll review and discuss security controls, which are used alongside frameworks to achieve an organization's security goals.

While frameworks are used to create plans to address security risks, threats, and vulnerabilities, controls are used to reduce specific risks. If proper controls are not in place, an organization could face significant financial impacts and damage to their reputation because of exposure to risks including trespassing, creating fake employee accounts, or providing free benefits.

Let's review the definition of controls. Security controls are safeguards designed to reduce specific security risks. In this video, we'll discuss three common types of controls: encryption, authentication, and authorization.

Encryption is the process of converting data from a readable format to an encoded format. Typically, encryption involves converting data from plaintext to ciphertext. Ciphertext is the raw, encoded message that's unreadable to humans and computers. Ciphertext data cannot be read until it's been decrypted into its original plaintext form. Encryption is used to ensure confidentiality of sensitive data, such as customers' account information or social security numbers.

Another control that can be used to protect sensitive data is authentication. Authentication is the process of verifying who someone or something is. A real-world example of authentication is logging into a website with your username and password. This basic form of authentication proves that you know the username and password and should be allowed to access the website. More advanced methods of authentication, such as multi-factor authentication, or MFA, challenge the user to demonstrate that they are who they claim to be by requiring both a password and an additional form of authentication, like a security code or biometrics, such as a fingerprint, voice, or face scan.

Biometrics are unique physical characteristics that can be used to verify a person's identity. Examples of biometrics are a fingerprint, an eye scan, or a palm scan. One example of a social engineering attack that can exploit biometrics is vishing. Vishing is the exploitation of electronic voice communication to obtain sensitive information or to impersonate a known source. For example, vishing could be used to impersonate a person's voice to steal their identity and then commit a crime.

Another very important security control is authorization. Authorization refers to the concept of granting access to specific resources within a system. Essentially, authorization is used to verify that a person has permission to access a resource. As an example, if you're working as an entry-level security analyst for the federal government, you could have permission to access data through the deep web or other internal data that is only accessible if you're a federal employee.

The security controls we discussed today are only one element of a core security model known as the CIA triad. Coming up, we'll talk more about this model and how security teams use it to protect their organizations.

Using web application scanners

As a penetration tester, you will also be required to perform web application security testing based on the scope of your penetration testing engagements. In this section, you will learn how to use various types of web application scanners to identify and fingerprint web applications on a target server.

Before proceeding, make sure you use the following guidelines to ensure you get the same results:

- During the next few sections, the target systems will be Metasploitable 2 and OWASP BWA virtual machines.
- Ensure Kali Linux has end-to-end connectivity with the Metasploitable 2 and OWASP BWA systems.

Let's get started!

WhatWeb

WhatWeb is a tool that is used to help penetration testers easily identify the available technologies and fingerprint web servers and web applications on a target system. WhatWeb is also pre-installed within Kali Linux and should be part of your arsenal of tools to help you on your journey.

To profile a web server and web application, use the following command, with the target set to your Metasploitable 2 or OWASP BWA virtual machine:

```
kali@kali:~$ whatweb 172.30.1.23
```

As shown in the following screenshot, WhatWeb was able to fingerprint the target:

```
kali@kali:~$ whatweb 172.30.1.23
http://172.30.1.23 [200 OK] Apache[2.2.14][mod_mono/2.4.3,mod_perl/2.0.4,mod_python/3.3.1,mod_ssl/2.2.14,proxy_html/3.0.1], Country[RESERVED][ZZ], Email[admin@metacorp.com,admin@owaspbwa.org,bob@ateliergraphique.com,cycloneuser-3@cyclonetransfers.com,jack@metacorp.com,test@thebodgeitstore.com], HTML5, HTTPServer[Ubuntu Linux][Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion_Passenger/4.0.38 mod_perl/2.0.4 Perl/v5.10.1], IP[172.30.1.23], JQuery[1.3.2], OpenSSL[0.9.8k], PHP[5.3.2-1ubuntu4.30][Suhosin-Patch], Passenger[4.0.38], Perl[5.10.1], Python[2.6.5], Script[text/javascript], Title[owaspbwa OWASP Broken Web Applications]
```

As an aspiring ethical hacker and penetration tester, some tools will help you gather information about the web server, while others will discover security vulnerabilities. However, it's important to research all the technologies that are found on a target web server when using WhatWeb; many security researchers share their findings and disclosure vulnerabilities to help others fight the battle against hackers.

To put it simply, WhatWeb provides the following details:

- The web application and its version
- The web technologies and their versions
- The host operating system and its version

By researching the version numbers of each technology, you will be able to find exploits that can take advantage of the vulnerabilities on the target system. In the next section, you will learn how to use Nmap to discover web application vulnerabilities.

Nmap

As you learned in the previous section, Nmap has a lot of very cool features and allows a penetration tester to use various types of scanning techniques and scripts to discover specific details about a target system. Within NSE, many scripts are already pre-loaded onto Kali Linux.

Using the following command, you will be able to see an entire list of all the Nmap scripts that begin with `http`:

```
kali@kali:~$ ls /usr/share/nmap/scripts/http*
```

From the list, you can choose to use a particular script to check for HTTP vulnerabilities on a target system. Let's imagine that you want to identify whether a target web application is vulnerable to **Structured Query Language** (**SQL**) **Injection** attacks. The `http-sql-injection` NSE script will be able to identify such security flaws. The following Nmap command shows how to invoke the SQL Injection script and perform a scan on a target that has port `80` open for web services:

```
kali@kali:~$ nmap --script http-sql-injection -p 80 172.30.1.26
```

The following screenshot shows the results of the Nmap scan:

```
kali@kali:~$ nmap --script http-sql-injection -p 80 172.30.1.26
Starting Nmap 7.91 ( https://nmap.org ) at 2021-07-09 11:45 EDT
Nmap scan report for 172.30.1.26
Host is up (0.00051s latency).

PORT   STATE SERVICE
80/tcp open  http
| http-sql-injection:
|   Possible sqli for queries:
|     http://172.30.1.26:80/mutillidae/index.php?page=set-background-color.php%27%20OR%20sqlspider
|     http://172.30.1.26:80/mutillidae/index.php?page=notes.php%27%20OR%20sqlspider
|     http://172.30.1.26:80/mutillidae/index.php?page=view-someones-blog.php%27%20OR%20sqlspider
|     http://172.30.1.26:80/mutillidae/index.php?page=show-log.php%27%20OR%20sqlspider
|     http://172.30.1.26:80/mutillidae/?page=view-someones-blog.php%27%20OR%20sqlspider
```

Figure 6.39 – Checking for SQL injection

As shown in the preceding screenshot, the Nmap script was able to automate the process of checking whether various URLs and paths are susceptible to a possible SQL Injection attack.

> **Tip**
>
> While many scripts within Nmap can be leveraged to identify vulnerabilities within web applications, it is important to always identify the service version of the web application by simply using the `-A` syntax when performing an initial scan to profile your target. Once you have identified the web application's service version, use the internet to research known vulnerabilities. As a penetration tester, it's always good to perform additional research on vulnerabilities as you may find more information on how to compromise the target.

Be sure to perform additional scanning on the target to discover any hidden security vulnerabilities, and use the information found at `https://nmap.org/nsedoc/` to gain an in-depth understanding of the purpose of various NSE scripts. In the next section, you will learn how to use Metasploit to check for web application vulnerabilities on a target.

The following screenshot shows some of the scan results from our target system:

```
+ Apache/2.2.14 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x b
ranch.
+ OSVDB-39272: /favicon.ico file identifies this app/server as: owasp.org
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. See
http://www.wisec.it/sectou.php?id=4698ebdc59d15. The following alternatives for 'index' were found: index.css, ind
ex.html
+ mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion_Passenger/4.0.38 mod_perl/2.0.4 Perl/v5.10.1 - mod_ssl 2.8.7 and lower are
 vulnerable to a remote buffer overflow which may allow a remote shell. http://cve.mitre.org/cgi-bin/cvename.cgi?n
ame=CVE-2002-0082, OSVDB-756.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ Retrieved x-powered-by header: PHP/5.3.2-1ubuntu4.30
+ Cookie phpbb2owaspbwa_data created without the httponly flag
+ Cookie phpbb2owaspbwa_sid created without the httponly flag
+ OSVDB-3092: /phpmyadmin/changelog.php: phpMyAdmin is for managing MySQL databases, and should be protected or li
mited to authorized hosts.
```

Figure 6.44 – Nikto scan

As shown in the preceding screenshot, Nikto can identify various security vulnerabilities within the target web application. They are listed in bullet format, and the + icon is used to indicate a new result. Take some time to read each line thoroughly as Nikto helps security professionals understand the details of the security vulnerabilities. It also provides references to where the flaws were found and how to resolve those weaknesses.

Next, you will learn how to perform a vulnerability scan on a target WordPress web application using WPScan.

WPScan

While there are many web applications within the e-commerce industry, there are many organizations that deploy the **WordPress** web application as their preferred **Content Management System** (**CMS**). While WordPress provides a very stylish and clean presentation of websites, many organizations do not update their WordPress platforms and plugins, thereby leaving their web server and web application vulnerable to potential cyberattacks from threat actors on the internet.

Within Kali Linux, you will learn about the WPScan tool, which allows penetration testers to perform vulnerability scanning and enumeration on the WordPress web application on a target server. For this exercise, we will be using the OWASP BWA virtual machine as it has a pre-installed WordPress web application located at `http://<OWASP BWA IP address>/wordpress`.

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

1. Open a Terminal within Kali Linux and use the following commands to perform a vulnerability scan on the OWASP BWA virtual machine:

```
kali@kali:~$ wpscan --url http://172.30.1.23/wordpress
--no-update
```

The following is a brief description of the syntax:

- `--url`: Specifies the target URL
- `--no-update`: Performs a scan without checking for updates

The following screenshot shows the vulnerability scan's results:

```
[+] XML-RPC seems to be enabled: http://172.30.1.23/wordpress/xmlrpc.php
 | Found By: Headers (Passive Detection)
 | Confidence: 60%
 | Confirmed By: Link Tag (Passive Detection), 30% confidence
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://172.30.1.23/wordpress/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] WordPress version 2.0 identified (Insecure, released on 2007-09-24).
 | Found By: Rss Generator (Passive Detection)
 |  - http://172.30.1.23/wordpress/?feed=rss2, <!-- generator="wordpress/2.0" -->
 |  - http://172.30.1.23/wordpress/?feed=rss2, <generator>http://wordpress.org/?v=2.0</generator>

[+] WordPress theme in use: default
 | Location: http://172.30.1.23/wordpress/wp-content/themes/default/
 | Last Updated: 2020-02-25T00:00:00.000Z
 | [!] The version is out of date, the latest version is 1.7.2
 | Style URL: http://172.30.1.23/wordpress/wp-content/themes/default/style.css
 | Style Name: WordPress Default
 | Style URI: http://wordpress.org/
 | Description: The default WordPress theme based on the famous <a href="http://binarybonsai.com/kubrick/">Kubrick</ ...
 | Author: Michael Heilemann
 | Author URI: http://binarybonsai.com/
```

Figure 6.45 – WPScan result

As shown in the preceding screenshot, WPScan will check each component of the WordPress installation and configuration on the remote target and provide details of its findings.

2. Next, to enumerate the login username of the target WordPress web application, use the `-e u` syntax, as shown here:

```
kali@kali:~$ wpscan --url http://172.30.1.23/wordpress
--no-update -e u
```

As shown in the following screenshot, WPScan was able to identify the login username of the target web server:

Figure 6.46 – WordPress username enumeration

As you have seen, it's quite simple to perform a vulnerability scan on a WordPress server and gather a list of potentially authorized usernames on the target server.

> **Tip**
>
> To learn more about WPScan and its capabilities, please see `https://tools.kali.org/web-applications/wpscan`.

Having completed this section, you have learned how to perform web scanning using various tools and techniques within Kali Linux. Having gathered a list of web application security vulnerabilities, with some additional research, you will be able to find working exploits to test whether these vulnerabilities are truly exploitable.

Summary

In this chapter, you learned about the importance of discovering security vulnerabilities within an organization and its assets. You also gained hands-on experience and skills with using various tools such as Nessus, Nmap, and Metasploit to perform security assessments on systems. You also discovered how various tools and techniques can be used to easily identify security flaws on web applications.

I hope this chapter has been informative for you and will prove helpful in your journey as an aspiring penetration tester, learning how to simulate real-world cyberattacks to discover security vulnerabilities and perform exploitation using Kali Linux. In the next chapter, *Chapter 7*, *Understanding Network Penetration Testing*, we will focus on how to use various techniques and strategies when performing network penetration testing.