HO#2.3
Scanning & Vulnerability Analysis: Part 1

Phase 1- Reconnaissance and Information Gathering

Dear students we have covered the Information gathering phase (reconnaissance) in HO#2.2, that *involves collecting as much public information as possible about the organization, systems, networks, applications, and employees to identify potential vulnerabilities and formulate a strategy for further testing*. Passive information gathering (reconnaissance) involves collecting data without directly interacting with the target system, reducing the risk of detection. Gathering information from publicly available sources like news outlets, blogs and social media platforms (Twitter, Facebook, LinkedIn) is named as Open-Source Intelligence (OSINT). The techniques used for OSINT are Web Scraping, Google Dorking, and social media profiling. The tools that we have used for this in HO#2.2 were `host, nslookup, dig, whois, knockpy, netdiscover, traceroute, whatweb, theHarvester, sherlock, wfw00f, Google Dorking, OSINT framework.`

Phase 2- Scanning and Vulnerability Analysis

Scanning and vulnerability analysis is the second phase of penetration testing in which we *scan the Network, ports, and OS services and determine if any of these are vulnerable.* In cybersecurity, scanning and vulnerability analysis are closely related but differ in terms of purpose, depth, and focus. *Scanning* is broader and focuses on identifying what is present (systems, ports, services), while *vulnerability analysis* digs deeper to determine what is wrong with those systems (misconfigurations, unpatched software, exploitable flaws).

- **Scanning:** The objective of *scanning* is to identify systems, services and potential entry points in a network by performing port scanning, NW scanning and Services detection using tools like `nmap, zenmap, unicorn, nikto` and so on. Its primary focus is to map the target's environment and identify what is running and reachable. Unlike reconnaissance and information gathering, *Scanning is Active information gathering*, because the tools used in this phase directly interact with the target network, hosts, ports, employees, and so on to collect data. So DONOT perform active network scanning unless you have written permission of the system owner to perform that testing. The *tools that we normally use for scanning* are `nmap, zenmap, unicornscan, nikto` and so on. There can be different types of scanning like:
  - Network Scanning: Identifies live hosts, devices, and IP addresses within a network.
  - Port Scanning: Detects which ports are open and which services are running on those ports.
  - Service/OS Detection: Determines which operating systems and services are running on discovered hosts.

  Scanning results provide a list of systems, open ports, and running services. However, it doesn't necessarily reveal whether these are vulnerable or not.

- **Vulnerability Analysis (or Vulnerability Assessment):** The objective of *vulnerability analysis* is to dig deeper and perform an in-depth examination to uncover known vulnerabilities and weaknesses in the systems, applications, and their configurations. Its primary focus is to evaluate the specific risks associated with the discovered systems and services by doing an in-depth examination to uncover known vulnerabilities, misconfigurations, or weaknesses. The *tools that we normally use for vulnerability analysis* are `nessus, searchsploit, OpenVAS, MSF, Burp-Suite, SQLMap` and so on. The steps that we normally perform during vulnerability analysis are:
  - Scanning the target for known vulnerabilities using databases like CVE
  - Assessing the severity of discovered vulnerabilities using metrics like CVSS
  - Submit report highlighting the vulnerability, risk levels, and mitigation steps

CWE vs CVE

Common Weakness Enumeration (CWE) is a general flaw or weakness in s/w design and implementation that could lead to vulnerabilities if not addressed **[CWE-NNN]**. Common Vulnerabilities and Exposures (CVE) is a specific instance of a vulnerability in a version of a real-world application that can be exploited **[CVE-YYYY-NNNNN]**. The primary maintainer of CWE and CVE databases, who develops, updates and publishes their entries is MITRE (MIT Research and Engineering). For details visit: https://cwe.mitre.org, https://cve.mitre.org, https://cve.org, https://cvedetails.com, https://nvd.nist.gov/ , https://nvd.nist.gov/vuln/categories and https://openwasp.org/

CVSS

Common Vulnerability Scoring System is a standardized framework used to assess and communicate the severity of vulnerabilities in software and hardware systems based on factors such as exploitability, impact, and complexity. CVSS provides a numeric score (ranging from 0 to 10) along with a *severity level rating* to help organizations prioritize their responses to vulnerabilities.

- **Critical (9.0 – 10.0)**: Vulnerabilities that are extremely easy to exploit and lead to complete compromise of systems or data.
- **High (7.0 – 8.9)**: Vulnerabilities are easily exploitable and have a significant impact on confidentiality, integrity, or availability.
- **Medium (4.0 – 6.9)**: Vulnerabilities that can be exploited but typically require some conditions, and the impact is moderate.
- **Low (0.1 – 3.9)**: Vulnerabilities that have limited impact or require complex conditions to exploit.
- **None (0.0)**: No impact or exploitation potential. An example is CVE-2015-1235 (Google Chrome Crash) has a CVSS score of 0.0. It is a vulnerability in Google Chrome that allowed attackers to cause a browser crash via a crafted web page. However, no code execution or data compromise was possible, leading to its 0.0 score.

CVE Process

- Discovery: A vulnerability or exposure is discovered by a security researcher, vendor, or another entity.
- Reporting: The issue is reported to a CVE Numbering Authority (CNA) or directly to MITRE. CNAs are organizations authorized to assign CVE IDs and manage CVE entries for their respective domains.
- Review: The reported issue is reviewed for accuracy and completeness. This may involve verifying the details and coordinating with the original reporter or affected vendor.
- Publication: Once validated, the CVE entry is published in the CVE database and made publicly available.

Some Example of Famous CVEs

- **CWE-120 (BoF)**: The program copies an input buffer to an output buffer without verifying that the size of the input buffer is less than the size of the output buffer, leading to a buffer overflow.
  - **CVE-2017-0144 (Eternal Blue):** A vulnerability in Microsoft SMBv1, that allows remote code execution, having a critical CVSS level.
  - **CVE-2015-7547:** A vulnerability pertains to stack based BoF in GNU glibc's `getaddrinfo()` function during DNS resolution. An attacker can exploit this by crafting malicious DNS responses, potentially leading to remote code execution or causing the application to crash.
  - **CVE-2010-3904:** A vulnerability in the reliable datagram sockets (RDS) protocol implementation within the Linux kernel versions prior to 2.6.36. Attacker with local access can gain elevated privileges, potentially obtaining root access.
- **CWE-362 (Race Condition)**: The program contains a code sequence that can run concurrently with other code, and the code sequence requires temporary, exclusive access to a shared resource, but a timing window exists in which the shared resource can be modified by another code sequence that is operating concurrently.
  - **CVE-2016-5195 (Dirty COW):** A vulnerability in Linux Kernel 2.x through 4.x before 4.8.3, that allows local users to gain privileges by leveraging incorrect handling of a copy-on-write feature to write to a read-only memory mapping, having a high CVSS level.
- **CWE-78 (OS Command Injection)**: The program improperly sanitizes the user input before passing it to the operating system to execute as a shell command.
  - **CVE-2014-6271 (Shellshock):** A vulnerability in GNU bash shell version 1.03 to 4.3, that allows remote attackers to execute arbitrary code by exploiting how bash shell processes environment variables.
  - **CVE-2011-2523 (Backdoor):** A vulnerability in vsftpd2.3.4, that opens a reverse shell at port 6200, and allows remote attackers to gain unauthorized root shell access. The attacker logs in It is assigned a critical CVSS level.
- **CWE-281 (Improper preservations of permissions)**: This weakness occurs when an application or system fails to properly enforce or maintain permission settings, allowing unauthorized users to perform actions that should be restricted.
  - **CVE-2019-14287 (sudo):** A vulnerability in the program sudo versions prior to 1.8.28, using which an unprivileged user can gain root privileges by specifying -1 or 4294967295 as the user ID. It is assigned a critical CVSS level.
- **CWE-843 (Type Confusion)**: The program allocates or initializes a resource such as a pointer, object, or variable using one type, but it later accesses that resource using a type that is incompatible with the original type.
  - **CVE-2020-6418:** A vulnerability in Google Chrome V8 JS engine, that exist in version 80.03987.122.
  - **CVE-2021-21571:** A vulnerability in MS Edge's Chakra JS engine, that exist in version 91.0.864.36.
- **H/W Based Vulnerabilities:** CVE-2017-5715 (Spectre) and CVE-2017-5754 (Meltdown) exploit critical vulnerabilities in modern processors related with speculative execution and indirect branch prediction. A malicious program can exploit Meltdown and Spectre to get hold of secrets stored in the memory of other running programs. This might include your passwords stored in a password manager or browser, your personal photos, emails, instant messages and even business-critical documents.

Attack Vector

*An **attack vector** is the path way through which an attacker can exploit a vulnerability in a system, network or application to gain unauthorized access, steal data, deploy malware, or disrupt a service.* There exist different types of attack vectors and I have categorized them into following classes:

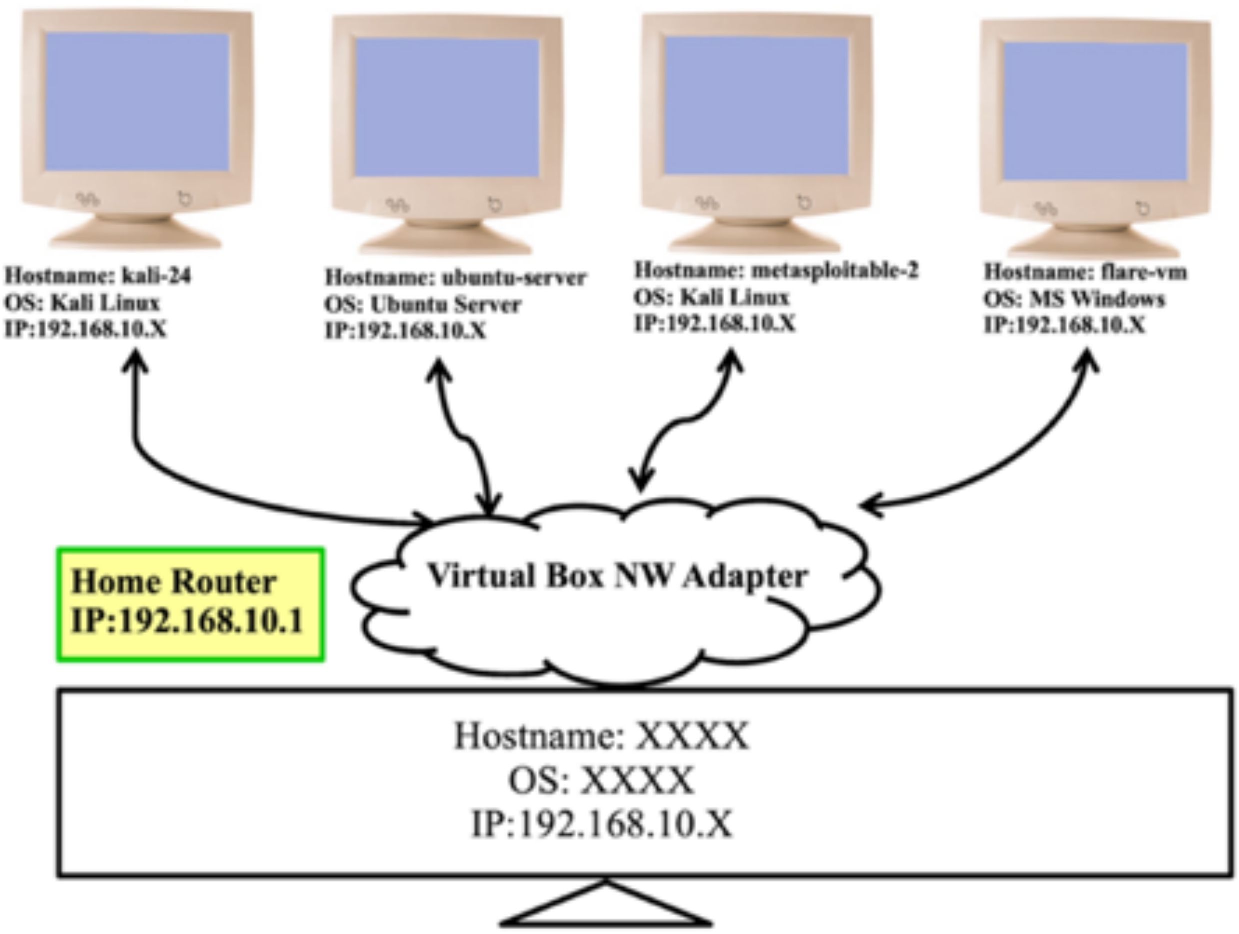
- **Network-based attacks** (e.g., Man-in-the-Middle, DoS/DDoS, ARP Spoofing)
- **Web-based attacks** (e.g., SQL Injection, Cross-Site Scripting, CSRF)
- **System-based attacks** (e.g., Buffer Overflow, Privilege Escalation, Rootkits)
- **Social Engineering attacks** (e.g., Phishing, Pretexting, Baiting)
- **Malware-based attacks** (e.g., Ransomware, Trojans, Worms, Keyloggers)
- **Physical attacks** (e.g., USB-based attacks, Evil Maid Attacks)

Attack Vector	Description	Famous CVE Example	Example Scenario
Remote Code Execution (RCE)	Allows attackers to execute arbitrary code on a target system remotely.	CVE-2017-0144 (EternalBlue)	Used by WannaCry ransomware, exploiting SMBv1 in Windows to spread across networks.
Command Injection	Injecting malicious commands into an application that executes system commands.	CVE-2011-2523 (Backdoor)	The attacker logs in with a specially crafted username ending with **":)"** and password anything to activate the hidden backdoor
Privilege Escalation	Attackers gain higher privileges on a system (e.g., from user to root).	CVE-2016-5195 (Dirty COW)	Linux vulnerability allowing users to gain root privileges by modifying read-only files.
Remote Desktop Exploit	Exploiting vulnerabilities in RDP services to gain control over systems.	CVE-2019-0708 (BlueKeep)	Affects older versions of Windows RDP, allowing attackers to execute code remotely without authentication.
Memory Corruption	Attacker manipulates memory structures to control a system.	CVE-2004-0597 (JPEG GDI+ Overflow)	Windows systems were exploited by maliciously crafted JPEG images causing buffer overflows.
Denial of Service (DoS/DDoS)	Attackers overload a system, making it unavailable.	CVE-2013-5211 (NTP Amplification Attack)	Attackers exploited the monlist command in NTP servers to launch large-scale DDoS attacks.
Man-in-the-Middle (MITM)	Attackers intercept or alter communication between two parties.	CVE-2020-0601 (CurveBall)	Attackers forged cryptographic certificates to impersonate trusted websites.
SQL Injection (SQLi)	Injecting malicious SQL queries to manipulate databases.	CVE-2012-1823 (PHP CGI SQLi)	Attackers gained remote access to web servers by manipulating PHP CGI parameters.
Cross-Site Scripting (XSS)	Injecting malicious scripts into a web page viewed by users.	CVE-2014-0160 (Heartbleed)	Although primarily an OpenSSL memory leak, attackers used it to steal session tokens and conduct XSS attacks.
Brute Force Attack	Repeatedly guessing credentials to gain unauthorized access.	CVE-2018-15473 (OpenSSH User Enumeration)	Allowed attackers to determine valid SSH usernames by analyzing server responses.
Exploit in Cryptographic Libraries	Weaknesses in cryptographic implementations that lead to vulnerabilities.	CVE-2014-0160 (Heartbleed)	OpenSSL flaw allowed attackers to leak sensitive information from memory.
Browser Exploits	Exploiting vulnerabilities in web browsers to execute malicious code.	CVE-2019-5786 (Chrome Zero-Day)	Used in the wild to escape Chrome's sandbox and gain higher privileges.
Malware Delivery via SMB Protocol	Exploiting vulnerabilities in SMB to spread malware.	CVE-2017-0144 (EternalBlue)	Used by NotPetya and WannaCry ransomware to spread laterally across networks.
Cross-Site Request Forgery (CSRF)	Trick users into executing unauthorized actions on a website.	CVE-2010-5298 (Facebook CSRF)	Allowed attackers to force users to "like" pages without their consent.

Environment Setup

You can use the following machines for a hands-on practice of this handout in which I am using kali Linux as attacker machine and scanning Metasploitable 2:

1. Kali Linux (IP: x.x.x.x)
2. Metasploitable 2 (IP: x.x.x.x)



If you want to login to your Metasploitable2 machine from Kali Linux using `ssh`, the simple login command might not work. There is a work around in which you need to mention the `HostKeyAlogirthms` as shown in the following command: ☺

```
   $ ssh   -oHostKeyAlgorithms=+ssh-dss   msfadmin@<ip of M2>
```

Nmap https://nmap.org

The **nmap** (Network Mapper) is a free and open-source utility for network discovery and security auditing. **Nmap** uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. Simply speaking, we can use **nmap** for *network scanning*, *port scanning*, and *vulnerability scanning*. Some of the most useful tasks that **nmap** can perform, are mentioned below:

- **Host Discovery**: Identifying live hosts on a network. This can be done using various techniques such as ICMP echo requests, TCP/UDP packets, and ARP requests.
- **Port Scanning**: Determining which ports on a target are open, closed, or filtered. Different types of scans include SYN scan, TCP connect scan, UDP scan, and FIN scan.
- **Service Version Detection**: Identifying the versions of services running on open ports. This helps in determining specific software vulnerabilities.
- **Operating System Detection**: Fingerprinting the target's operating system and sometimes even determining the OS version and device type.
- **Scriptable Interaction**: Using Nmap Scripting Engine (NSE) to perform advanced network tasks such as vulnerability detection, backdoor detection, and more. NSE scripts can also be customized or created to meet specific needs.
- **Stealth Scanning**: Performing scans in a way that minimizes detection by firewalls and intrusion detection systems (IDS).

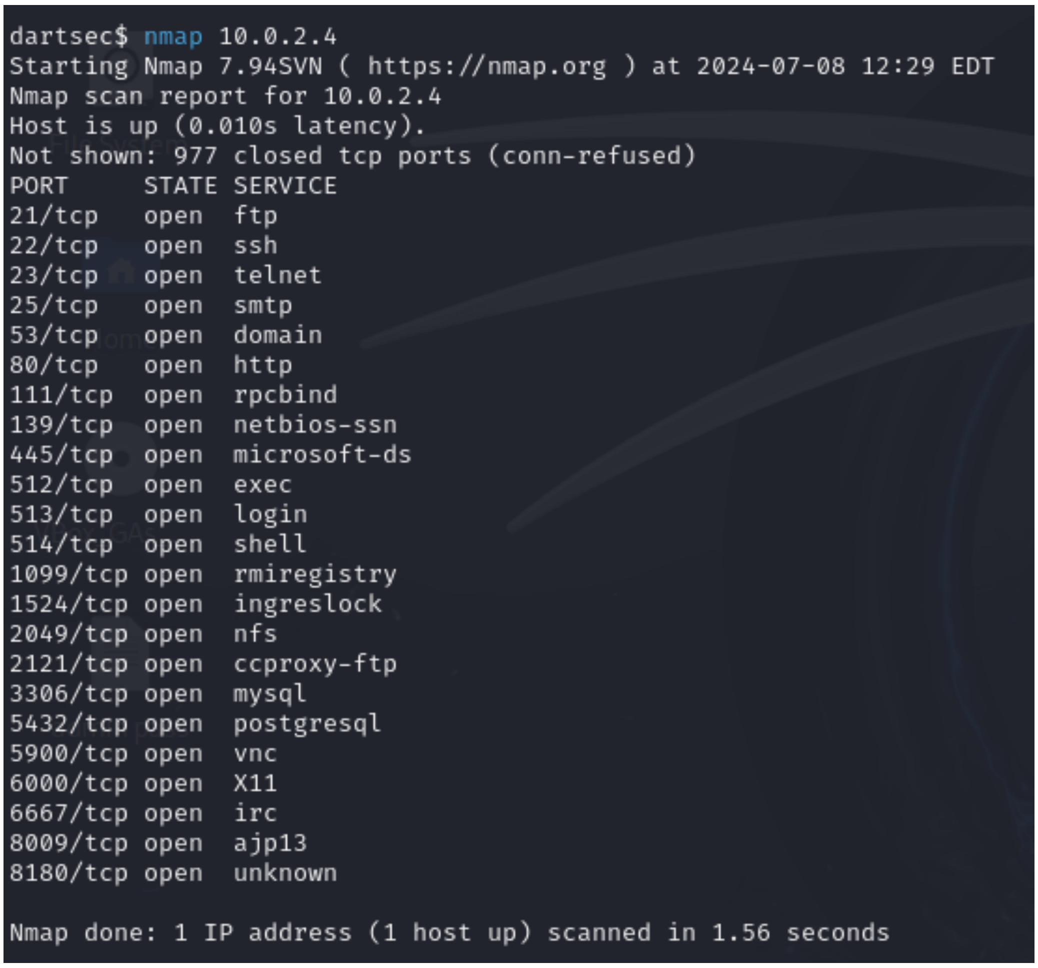
Basic Usage of nmap

```
$ nmap --version
$ nmap -h
$ man nmap
$ nmap <IP/hostname/NW Addr>
```

Command	Description
`$ nmap 192.168.1.1`	Scanning a single IP (By default nmap scans first 1000 commonly used ports)
`$ nmap www.domain.com`	Scanning a hostname
`$ nmap 192.168.1.1-100`	Scanning an IP range
`$ nmap 192.168.1.1/24`	Scanning an entire subnet
`$ nmap -iL list.txt`	Scanning from a predefined list

Command	Description
`$ nmap -p 22 192.168.1.1`	Scanning a single port
`$ nmap -p 20-80 192.168.1.1`	Scanning a range of ports
`$ nmap 20-25,80,443 192.168.1.1`	Scanning a range and individual ports
`$ nmap -p- 192.168.1.1`	Scanning all 65535 ports
`$ nmap -F 192.168.1.1`	Scanning first 100 ports (Fast Scan)
`$ nmap -A 192.168.1.1`	Aggressive Scan enables OS detection, version detection, script scanning and traceroute

`$ nmap <IP of Metasploitable2>`



```
dartsec$ nmap 10.0.2.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-08 12:29 EDT
Nmap scan report for 10.0.2.4
Host is up (0.010s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 1.56 seconds
```

Description of output: Although the STATE of all the ports in this screenshot is open, however, in the output of **`nmap`** scan, a port can be in one of the following six states:

- **Open**: This indicates that an application is accepting connections on this port and is reachable.
- **Closed**: This indicates that the port is reachable but not currently in use, i.e., no application is listening on it.
- **Filtered**: Nmap cannot determine whether the port is open because packet filtering prevents its probes from reaching the port. This is often due to a firewall or other NW security devices.
- **Unfiltered**: The port is accessible, but Nmap cannot determine if it is open or closed.
- **Open|Filtered**: Nmap cannot determine whether the port is *open* or *filtered*. This happens when Nmap does not receive enough information to decide between the two states.
- **Closed|Filtered**: Nmap cannot determine whether the port is *closed* or *filtered*. This is a less common state, indicating ambiguous results.

Scanning the Famous `http://scanme.nmap.org`

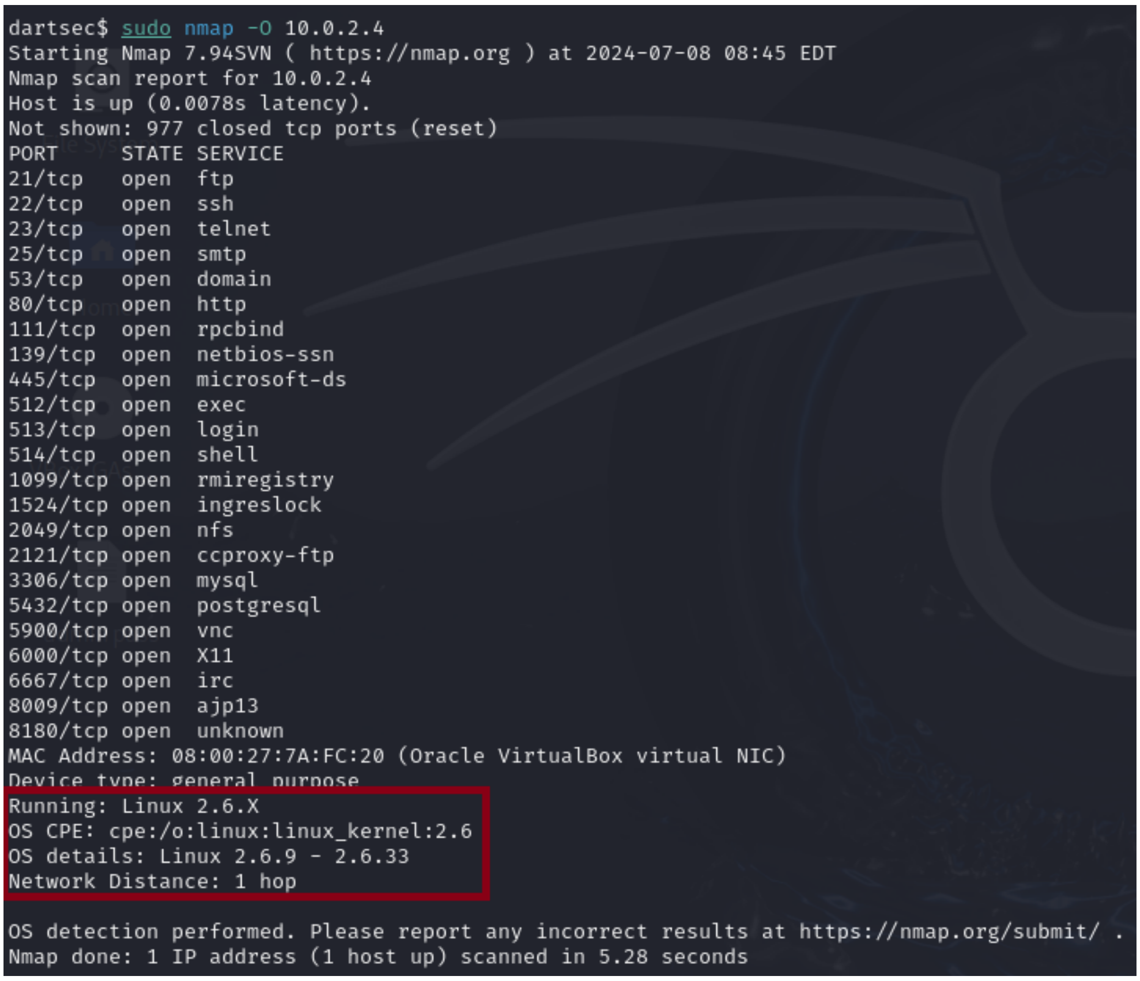
This is a machine that `nmap` has setup for the folks to test and make sure that their Nmap installation (or Internet connection) is working properly. You are authorized to scan this machine with Nmap or other port scanners. Try not to hammer on the server too hard.

```
$ nmap -p 20-100 scanme.nmap.org
$ nmap -p 20-100 45.33.32.156
```

Target OS Discovery

Discovering the operating system (OS) of a target host using `nmap` is a useful feature for network scanning and security assessments. Different operating systems and versions have distinct ways of responding to network probes. By sending probes to both open and closed ports, `nmap` can gather more varied responses. We can perform OS detection using the `-O` option to `nmap`:

```
$ sudo nmap -O <ip of M2>
```

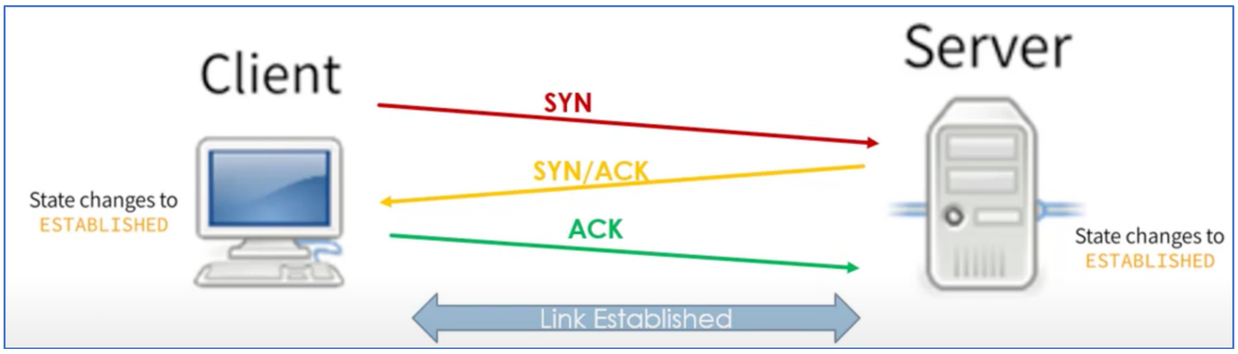


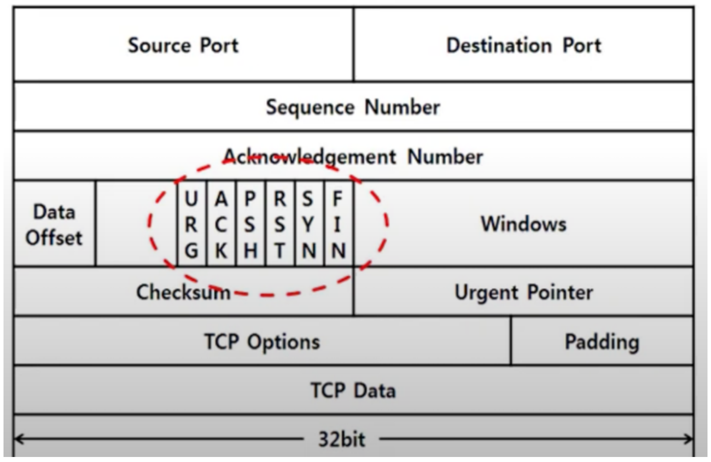
```
dartsec$ sudo nmap -O 10.0.2.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-08 08:45 EDT
Nmap scan report for 10.0.2.4
Host is up (0.0078s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:7A:FC:20 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 5.28 seconds
```

The highlighted output in red tells us about the running Linux kernel. The **`X`** in version numbers like **`2.6.X`** signifies that `nmap` has identified the major and minor version of the OS but does not have enough information to determine the exact patch level or sub-version. This shows that `nmap` scans are not all accurate or give complete information. We need to run more scans with more `nmap` options to find full information.

TCP Three Way Hand Shake (A Recap)

Before we describe **nmap** Ping/NW Scan (`-sn`), SYN Scan (`-sS`), TCP Scan (`-sT`), FIN Scan (`-sF`), ACK Scan (`-sA`), Version Scan (`-sV`), we need to talk about how TCP connections are established. The connections are scanned using a 3-way handshake. The 3-way handshake is a fundamental process used in the TCP/IP protocol suite to establish a reliable connection between a client and a server. It ensures that both parties are ready to communicate and can establish a connection with synchronized parameters. Steps in a handshake are mentioned below:



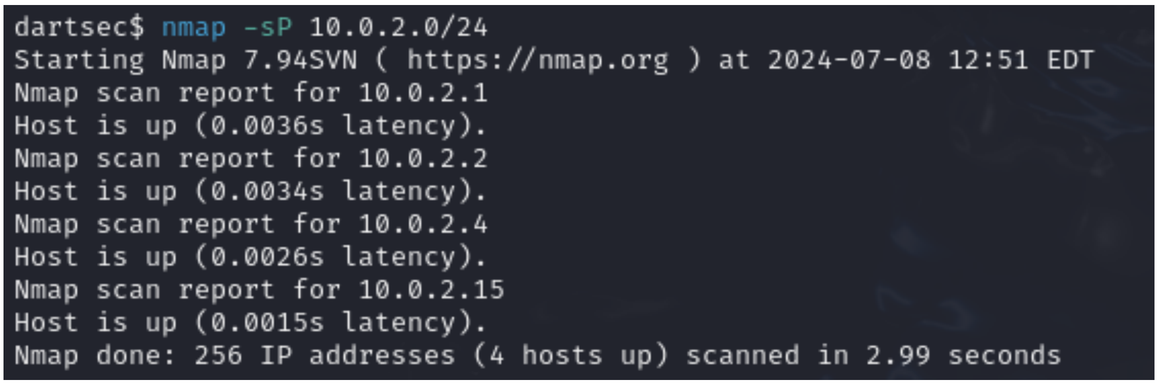


- **SYN (Synchronize):**
    - The client initiates the connection by sending a TCP segment with the SYN (synchronize) flag set to the server. This segment includes an initial sequence number (ISN), which is a random value used to synchronize the sequence numbers between the client and the server.
- **SYN-ACK (Synchronize-Acknowledge):**
    - The server responds to the client's SYN segment with a TCP segment that has both the SYN and ACK (acknowledge) flags set. The server's segment also includes its own initial sequence number (SYN) and acknowledges the client's SYN by setting the ACK number to the client's ISN + 1.
- **ACK (Acknowledge):**
    - The client sends a final acknowledgment (ACK) to the server. This segment acknowledges the server's SYN by setting the acknowledgment number to the server's ISN + 1.
    - At this point, the connection is established, and both parties are ready to exchange data. The ports on both client and server side are now in established state.

Ping/Network Scan

The `nmap` ping scan is used for network discovery, i.e., what all machines are running in the Local Area Network. It sends various probes (like ICMP echo request, TCP SYN to port 443, or ICMP timestamp request) to check if the target is up. By default, it won't scan any ports, only determining whether the host is active. In older versions of Nmap (before version 4.0), **`-sp`** was used to perform a ping scan. However, in recent versions, it has been deprecated in favour of **`-sn`** option.

```
$ nmap -sn 10.0.2.0/24
```



```
dartsec$ nmap -sP 10.0.2.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-08 12:51 EDT
Nmap scan report for 10.0.2.1
Host is up (0.0036s latency).
Nmap scan report for 10.0.2.2
Host is up (0.0034s latency).
Nmap scan report for 10.0.2.4
Host is up (0.0026s latency).
Nmap scan report for 10.0.2.15
Host is up (0.0015s latency).
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.99 seconds
```

TCP SYN Scan

TCP SYN scan, often referred to as a "**half-open**" scan, is one of the most common and efficient scanning techniques used by `nmap`. It works by sending SYN packets to the target ports and analysing the responses. This type of scan is called "half-open" because it does not complete the full TCP handshake. Here's how it works:

```
$ sudo  nmap -sS <ip of M2>
```

- **SYN Packet Sent**: `nmap` sends a SYN packet to the target port.
- **Response Analysis**:
  - **SYN/ACK Received**: If the target port is open, it responds with a SYN/ACK packet. Instead of ACK, Nmap sends an RST (reset) packet to terminate the connection before the handshake completes, which prevents the connection from being logged.
  - **RST Received**: If the target port is closed, it responds with an RST packet.
  - **No Response/Filtered**: If there is no response or if a packet filtering device (like a firewall) is present, the port is marked as filtered.

TCP Connect Scan

A TCP Connect scan perform full 3-way handshake and so do not require root privileges. It is easily detected. By the target machine administrator as it leaves quite a bit of foot print. This is the default `nmap` scan, i.e., if you do not use any option the scan that is performed is TCP connect scan. TCP Scan works in the following way:

```
$ nmap -sT <ip of M2>
```

- **SYN Packet Sent**: Nmap sends a SYN packet to the target port.
- **SYN/ACK Received**: If the target port is open, it responds with a SYN/ACK packet.
- **ACK Sent**: Nmap sends an ACK packet to complete the TCP three-way handshake, establishing a connection.
- **RST Packet Sent**: Immediately after establishing the connection, Nmap sends an RST (reset) packet to close the connection.

UDP Scan

A UDP scan is a type of network scan performed by Nmap to identify open UDP ports on a target system. Unlike TCP, UDP (User Datagram Protocol) is connectionless, which makes it more challenging to scan. However, it is crucial for identifying services that run over UDP, such as DNS (53), DHCP(67), and TFTP (69) and SNMP.

```
$ nmap -sU <ip of M2>
```

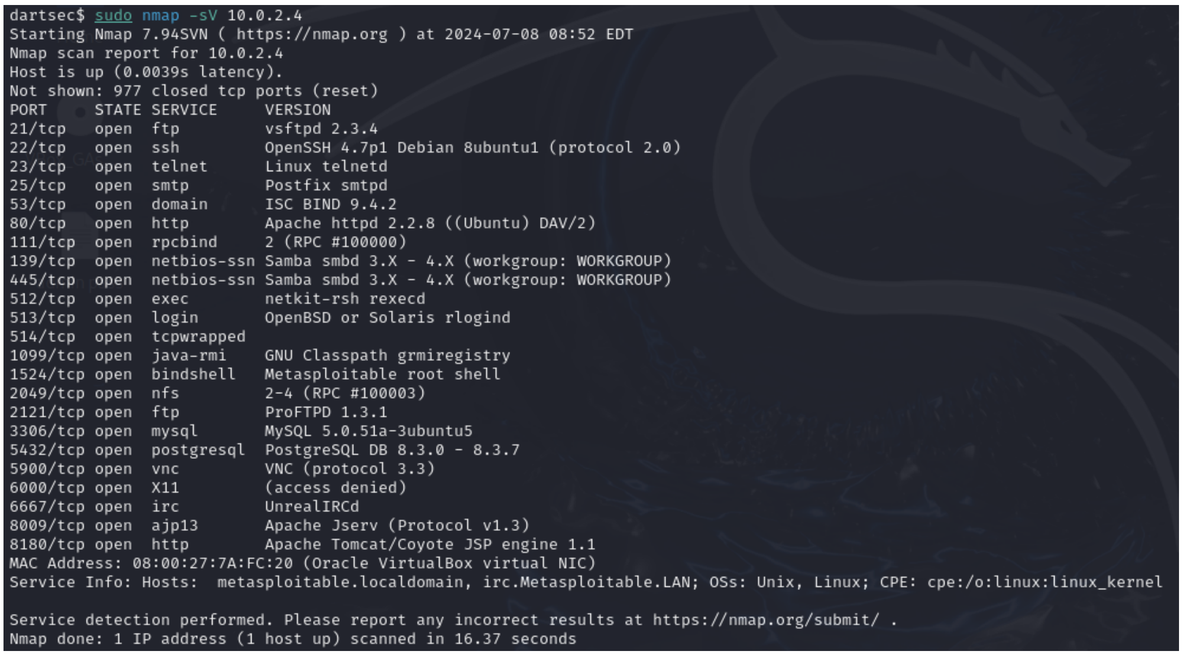
- **UDP Packet Sent**: Nmap sends a UDP packet to the target port. The default payload for the UDP packet is empty, but Nmap can use more specific probes for well-known services.
- **Response Analysis**:
  - **No Response**: If there is no response, the port is assumed to be open or filtered. Many UDP services do not respond unless they receive a valid request, which can make definitive detection challenging.
  - **ICMP Port Unreachable**: If the target sends back an ICMP Port Unreachable message (Type 3, Code 3), the port is closed.
  - **Other ICMP Responses**: Different ICMP responses may indicate that the port is filtered or unreachable for other reasons.

Retries: Nmap may send multiple probes and use different payloads to increase accuracy. This is because UDP scanning is prone to packet loss and other issues that can cause false positives or negatives.

Finding Versions of Services

To tell `nmap` to display the version of different services running on different ports we can use the `-sV` option. Remember for this to work we need to be root or in the sudoer file.

```
$ sudo nmap -sV <ip of M2>
```



```
dartsec$ sudo nmap -sV 10.0.2.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-08 08:52 EDT
Nmap scan report for 10.0.2.4
Host is up (0.0039s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:7A:FC:20 (Oracle VirtualBox virtual NIC)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.37 seconds
```

As we can see along with services, we can now also see the version of each service. We can see that `nmap` has failed to find the versions of some of the services. This can be solved by `--version-intensity` that allows you to control the aggressiveness or intensity of version detection when scanning target services. It determines how much effort `nmap` puts into identifying the version details of the services running on open ports. The higher the intensity the more time it takes, but more detailed and accurate information about the services running on the specified target. We can specify a level of intensity from 0 to 9:

- **0**: Disable version detection.
- **1-4**: Low to moderate intensity, suitable for quick scans with less detailed version information.
- **5 (default)**: Balanced intensity, providing a good compromise between speed and accuracy.
- **6-9**: Higher intensity levels, which increase the accuracy of version detection but might take longer and be more intrusive.

Evading Firewalls/IDS using nmap

A Firewall is a NW security device or software that monitors and controls incoming and outgoing network traffic based on predetermined security rules. The two main types are *Network firewall* that protects an entire network by filtering traffic at the network level and *Host-based firewall* that protects individual devices (like computers or servers) by filtering traffic at the operating system or application level. An Intrusion Detection System (IDS), on the other hand is a security tool that monitors network or system activities for malicious or suspicious activities and alerts administrators or takes automated actions in response. The two main types are *Network-based IDS (NIDS)* that monitors network traffic in real-time, analysing packets and patterns to detect suspicious activities across the network and *Host-based IDS (HIDS)* that runs on individual devices or hosts, monitoring activities. Firewalls are unpredictable as we don't know the rules applied to the firewall. So, we don't know what kind of scan to perform.

The **`-f`** option in **`nmap`** is used to fragment packets. This means `nmap` splits the probes into smaller IP fragments rather than sending them as one complete packet. The goal of using packet fragmentation is to evade detection by firewalls, intrusion detection systems (IDS), and intrusion prevention systems (IPS) that may be set up to detect and block standard `nmap` scanning patterns.

```
$ nmap -f 10.0.2.4          // splits packets into 8-bytes fragments
$ nmap -f -f 10.0.2.4       // splits packets into 16-bytes fragments
$ nmap -mtu 32 10.0.2.4     // specify the exact size of packets
```

Decoy Scanning

The **`-D`** option in **`nmap`** is used to perform decoy scanning, which helps to obscure the true source of the scan by making it appear as if multiple hosts (decoys) are scanning the target. This technique can be useful for evading detection and confusing firewalls, intrusion detection systems (IDS), and intrusion prevention systems (IPS).

```
$ nmap -D RND:5 10.0.2.4
```

The above command will make 5 random decoys and will scan the target machine along with the original source machine.

Using Proxies to Redirect your Nmap Scans

In our previous handout we have already seen and understood different techniques like proxies and Tor browser to anonymize our traffic on the Internet. Following command will use proxy chains to redirect our **`nmap`** scans in order to mask our original IP address:

```
$ proxychains nmap -F scanme.nmap.org
```

Using Nmap Scripts

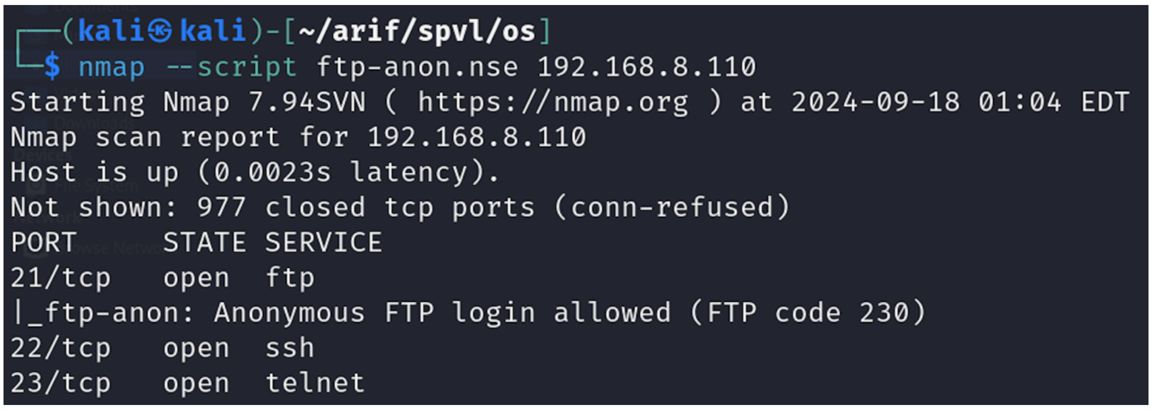
- The **nmap** scripts are part of the Nmap Scripting Engine (NSE), which has extended the capabilities of `nmap` beyond just network scanning and port discovery. These scripts are used to perform more detailed scanning and automate a wide range of tasks, from vulnerability detection to network auditing. These scripts can automate various network tasks, perform complex network reconnaissance, vulnerability detection, exploitation, and more. On Linux machine, these scripts are located at **`/usr/share/nmap/scripts/`** directory containing over 600 different scripts with each script belonging to a specific category. Visit this directory to check out the available scripts on your Kali Linux machine and practice running different scripts on Metasploitable2 machine.

- **Check out what a specific script does:** Some scripts are very noisy, some not at all. So, it is important to read what each script does and if it is easily detectable by the target or not. Let us check what `ftp-anon.nse` script do by using the `--script-help` option of `nmap` as shown below:

```
$ nmap --script-help ftp-anon.nse
```

- **Running single script:** We can run single script of our choice using the following command:

```
$ nmap --script ftp-anon.nse <IP of M2>
```



```
┌──(kali㉿kali)-[~/arif/spvl/os]
└─$ nmap --script ftp-anon.nse 192.168.8.110
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-18 01:04 EDT
Nmap scan report for 192.168.8.110
Host is up (0.0023s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp   open  ssh
23/tcp   open  telnet
```

From the output of above command, we see that Metasploitable2 has anonymous FTP vulnerability, i.e., it allows to login using `anonymous` username and blank password. Let us try it:

```
$ ftp 192.168.8.110
Connected to 192.168.8.110.
220 (vsFTPd 2.3.4)
Name: anonymous
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.

ftp>
```

You get the FTP prompt, now give the help command to get the available ftp commands and enjoy ☺

Categories/Groups of NSE Scripts:
To check out the usage of different script groups visit: https://nmap.org/book/nse-usage.html

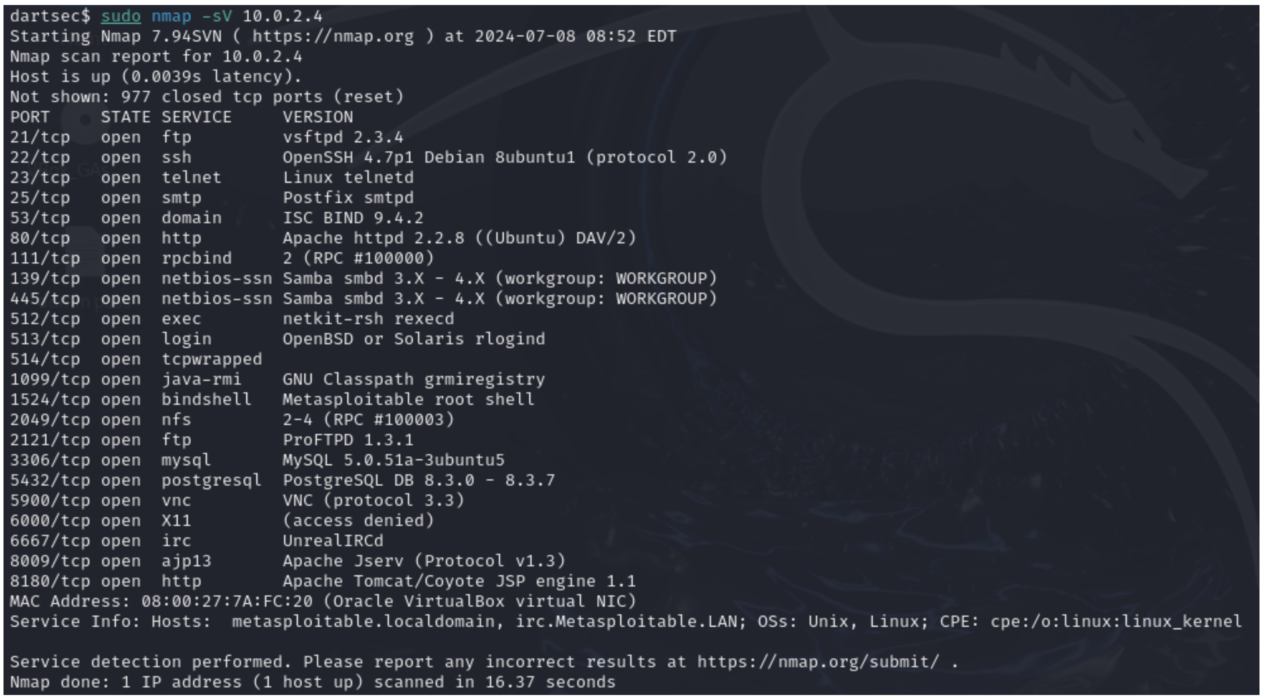
- **auth**: These scripts deal with authentication credentials (or bypassing them) on the target system. For example, `ftp-anon.nse` script attempts to guess FTP login credentials
- **brute**: These scripts use brute force attacks to guess authentication credentials of a remote server. Nmap contains scripts for brute forcing dozens of protocols, including `http-brute-nse`, `oracle-brute.nse`, `snmp-brute.nse` and so on.
- **malware**: These scripts test whether the target platform is infected by malware or backdoors. An example script in this category is `smb-vuln-ms17-010.nse` that checks for the MS17-010 vulnerability exploited by WannaCry.
- **vuln:** These scripts check for specific known vulnerabilities and generally only report results if they are found. Examples include `realvnc-auth-bypass` and `afp-path-vuln`.
- **exploit**: These scripts exploit vulnerabilities to confirm their existence. An example script in this category is `http-sql-injection.nse` that checks for the SQL injection vulnerabilities. Other example include `http-shellshock.nse`
- **default**: These scripts are the default, and are run using using `--script default` or using simply the `-A` option.

- **Running a Script Group:** Try running the following script groups and from their output understand different vulnerabilities in the Metasploitable2 machine that they come up with.

```
$ nmap --script vuln <ip>
```

Searchsploit

We have already seen that we can use the `nmap -sV` option to check for the versions of different services along with their state and port numbers as shown in the following screenshot.



```
dartsec$ sudo nmap -sV 10.0.2.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-08 08:52 EDT
Nmap scan report for 10.0.2.4
Host is up (0.0039s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:7A:FC:20 (Oracle VirtualBox virtual NIC)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.37 seconds
```

Now a $100 question is how can we find out, which out of these services has some vulnerability without using any tool. One way is open a browser and type the `` `service name` `` followed by `` `exploit` `` keyword and press enter (e.g., "`vsftpd 2.3.4 exploit`" or "`Appache httpd 2.2.8 exploit`"). Visit some links and try to find whether these are vulnerable applications, and if yes try finding exploits for these vulnerable applications. Do visit https://www.cve.org/

The **`searchsploit`** is a Linux command-line search tool to search the Exploit Database (EDB), while staying offline. It is used to search for known vulnerabilities and exploits related to hardware, software, operating systems, web applications, and configurations. These exploits are already indexed in the Exploit Database. It allows you to quickly find exploits based on various criteria like software names, CVE identifiers, or platform types. The most common use cases of `searchsploit` are:

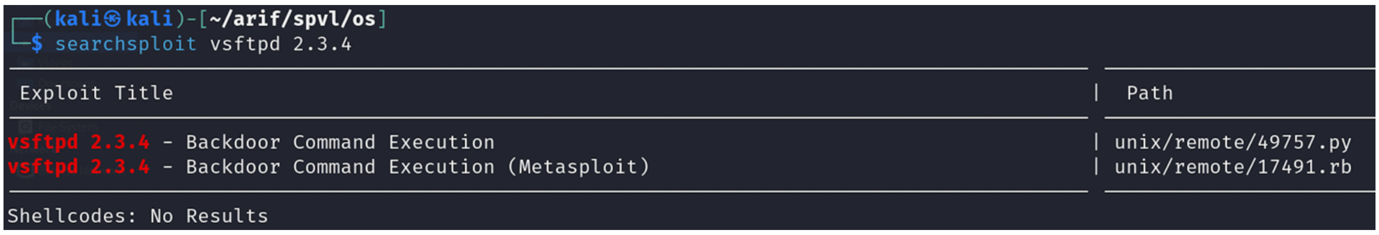
Installing and using searchsploit:

- An easy way to install `exploitdb` and use `searchsploit` on your Kali Linux machine use following command:
  `$ sudo apt-get install exploitdb`

- You can also install from source, available at https://gitlab.com/exploit-database/exploitdb
- Do check out `/usr/share/exploitdb` directory for available exploits and shellcodes for different OSs and different architectures.
- Before you start using the tool, you must update the Exploit Database (EDB), using the following command:
  `$ searchsploit -u`

- Let us now find out if `vsftpd 2.3.4` is vulnerable service, and if there exist an exploit for this vulnerability. To `searchsploit`, we just need to give the name of the service along with its version, and it searches in the local Kali Linux `exploitdb` directory for all the exploits that Kali Linux has and tries to find an exploit that will work for that specific version.

  ```
  $ searchsploit vsftpd 2.3.4
  ```

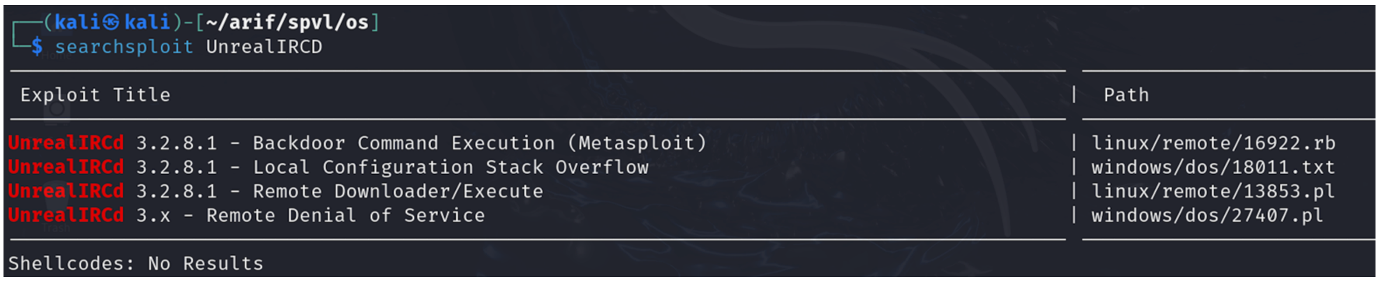
  ```
  ┌──(kali㉿kali)-[~/arif/spvl/os]
  └─$ searchsploit vsftpd 2.3.4
  ---------------------------------------------------------------------- ----------------------
   Exploit Title                                                        |  Path
  ---------------------------------------------------------------------- ----------------------
  vsftpd 2.3.4 - Backdoor Command Execution                             | unix/remote/49757.py
  vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)                | unix/remote/17491.rb
  ---------------------------------------------------------------------- ----------------------
  Shellcodes: No Results
  ```

  From the output of the above command, you can see that it also suffers with the Backdoor Command Execution vulnerability and we have two exploits for it one written in Python and other in Ruby. To check the absolute path of these exploits, we can use the locate command:

  ```
  $ locate 49757.py
  /usr/share/exploitdb/exploits/unix/remote/49757.py
  ```

- Let us now try to find out if `UnrealIRCd` (Internet Relay Chat daemon) is a vulnerable service and if there is an exploit available for it.

  ```
  $ searchsploit unrealircd
  ```

  ```
  ┌──(kali㉿kali)-[~/arif/spvl/os]
  └─$ searchsploit UnrealIRCD
  ---------------------------------------------------------------------- ----------------------
   Exploit Title                                                        |  Path
  ---------------------------------------------------------------------- ----------------------
  UnrealIRCd 3.2.8.1 - Backdoor Command Execution (Metasploit)          | linux/remote/16922.rb
  UnrealIRCd 3.2.8.1 - Local Configuration Stack Overflow               | windows/dos/18011.txt
  UnrealIRCd 3.2.8.1 - Remote Downloader/Execute                        | linux/remote/13853.pl
  UnrealIRCd 3.x - Remote Denial of Service                             | windows/dos/27407.pl
  ---------------------------------------------------------------------- ----------------------
  Shellcodes: No Results
  ```

  From the output of the above command, you can see that there exists an exploit with the name of Backdoor Command Execution, that can be used in MSF.

  ```
  $ locate 16922.rb
  /usr/share/exploitdb/exploits/linux/remote/16922.rb
  ```

- **Tips for Filtering your search Results:**
  - **Filtering by version:**
    ```
    $ searchsploit apache
    $ searchsploit apache 2.4.49
    ```
  - **Removing Unwanted Results using Pipe and grep:**
    ```
    $ searchsploit apache | grep unix
    ```
  - **Searching by CVE:** You can also search using the CVE ID as shown below:
    ```
    $ searchsploit CVE-2017-0144
    ```
  - **Searching multiple terms:** You can add any number of search terms. But remember, `searchsploit` uses an AND operator, not an OR operator. The more terms that are used, the more results will be filtered out:
    ```
    $ searchsploit afd windows local
    ```
  - **Title Searching:** By default, `searchsploit` will check BOTH the title of the exploit as well as the path. Depending on the search criteria, this may bring up false positives (especially when searching for terms that match platforms and version numbers). Searches can be restricted to the titles by using the **`-t`** option:
    ```
    $ searchsploit -t oracle windows
    ```

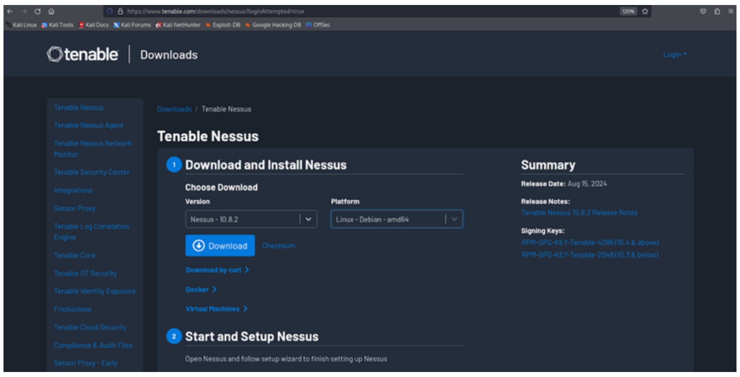
Nessus

Nessus is a platform developed by Tenable that scans for security vulnerabilities in devices, applications, operating systems, cloud services and other network resources. Unlike `nmap`, it checks for specific vulnerabilities tied to known CVEs along with their CVSS scores. The Professional version has a cost to pay, so we will install and use the free *Nessus Essentials* version, which can work only in a Local Area Network and cannot perform a scan on a machine/website on the Internet. Moreover, with Nessus Essentials version, we can scan up to 16 IP addresses within a LAN. Some common features of Nessus are:

- **Vulnerability scanning:** Nessus scans servers for known vulnerabilities. For example, detecting outdated software versions that may be suspectable to exploits.
- **Credential-based scanning:** Authenticated scans with login credentials provide Nessus deeper access, enhancing the accuracy of vulnerability detection.
- **Web Application scanning:** It identifies the vulnerabilities in web applications such as SQL injection or XSS flaws.
- **Malware detection:** Nessus identifies the potential malware indicators by analyzing the system files and configuration.

Downloading, Installing and Configuring Nessus on Kali Machine:

- Open your browser and search for *download nessus*. Use the Tenable link and download the latest version of Nessus for your platform, i.e., Linux-Debian-amd64, which will download the `Nessus-10.8.2-debian10_amd64.deb` file.


https://www.tenable.com/downloads/nessus?loginAttempted=true

tenable | Downloads

Downloads / Tenable Nessus

Tenable Nessus

1 Download and Install Nessus

Choose Download

Version: Nessus - 10.8.2

Platform: Linux - Debian - amd64

Summary

Release Date: Aug 15, 2024

Release Notes: Tenable Nessus 10.8.2 Release Notes

Signing Keys: RPM-GPG-KEY-Tenable-4096 (10.4 & above) RPM-GPG-KEY-Tenable-2048 (10.3 & below)

2 Start and Setup Nessus

Open Nessus and follow setup wizard to finish setting up Nessus

- Now to install it use the following command:

```
$ sudo dpkg -i Nessus-10.8.2-debian10_amd64.deb
```

- After installation is done, you need to start and check status of the service:

```
$ sudo systemctl start nessusd.service
$ systemctl status nessusd.service
```

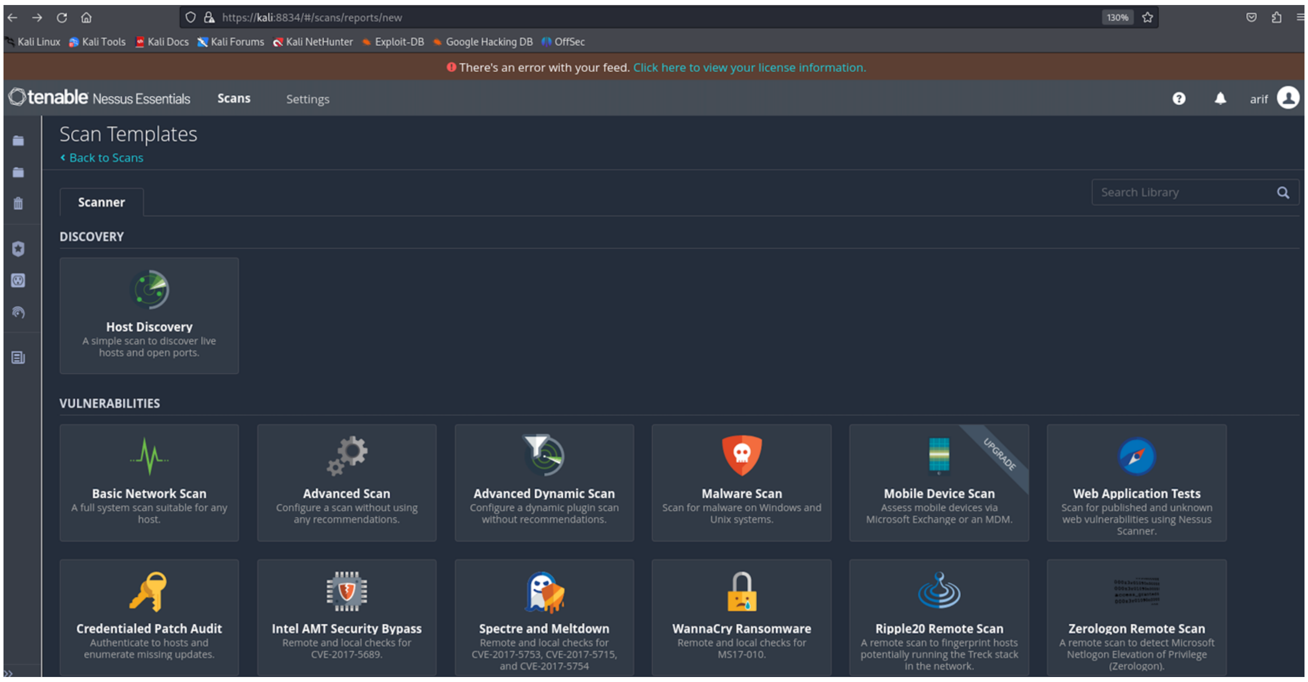
- In order to configure the Nessus Essential (free version), you need to go to https://kali:8834 link inside your browser. Perform the steps and finally it will initialize and download all required plugins. This may take a bit of time depending on your Internet speed. Must remember the credentials (username:password) while creating an account on tenable for Nessus Essentials.

- Inside your browser go to https://kali:8834 login giving your credentials. If you get an error of plugins not installed then use the following command to install plugins:

```
$ sudo systemctl stop nessusd.service
$ sudo /opt/nessus/sbin/nessuscli update
$ sudo systemctl start nessusd.service
```

Basic NW Scan using Nessus:

- Now inside your browser go to https://<ip of kali>:8834 or https://127.0.0.1:8834 and login giving your credentials, and go ahead with your first scan, by clicking the *New Scan* button at the top right.



https://kali:8834/#/scans/reports/new

There's an error with your feed. Click here to view your license information.

tenable Nessus Essentials Scans Settings arif

Scan Templates
‹ Back to Scans

Scanner Search Library

DISCOVERY

Host Discovery
A simple scan to discover live hosts and open ports.

VULNERABILITIES

Basic Network Scan
A full system scan suitable for any host.

Advanced Scan
Configure a scan without using any recommendations.

Advanced Dynamic Scan
Configure a dynamic plugin scan without recommendations.

Malware Scan
Scan for malware on Windows and Unix systems.

Mobile Device Scan
Assess mobile devices via Microsoft Exchange or an MDM.

Web Application Tests
Scan for published and unknown web vulnerabilities using Nessus Scanner.

Credentialed Patch Audit
Authenticate to hosts and enumerate missing updates.

Intel AMT Security Bypass
Remote and local checks for CVE-2017-5689.

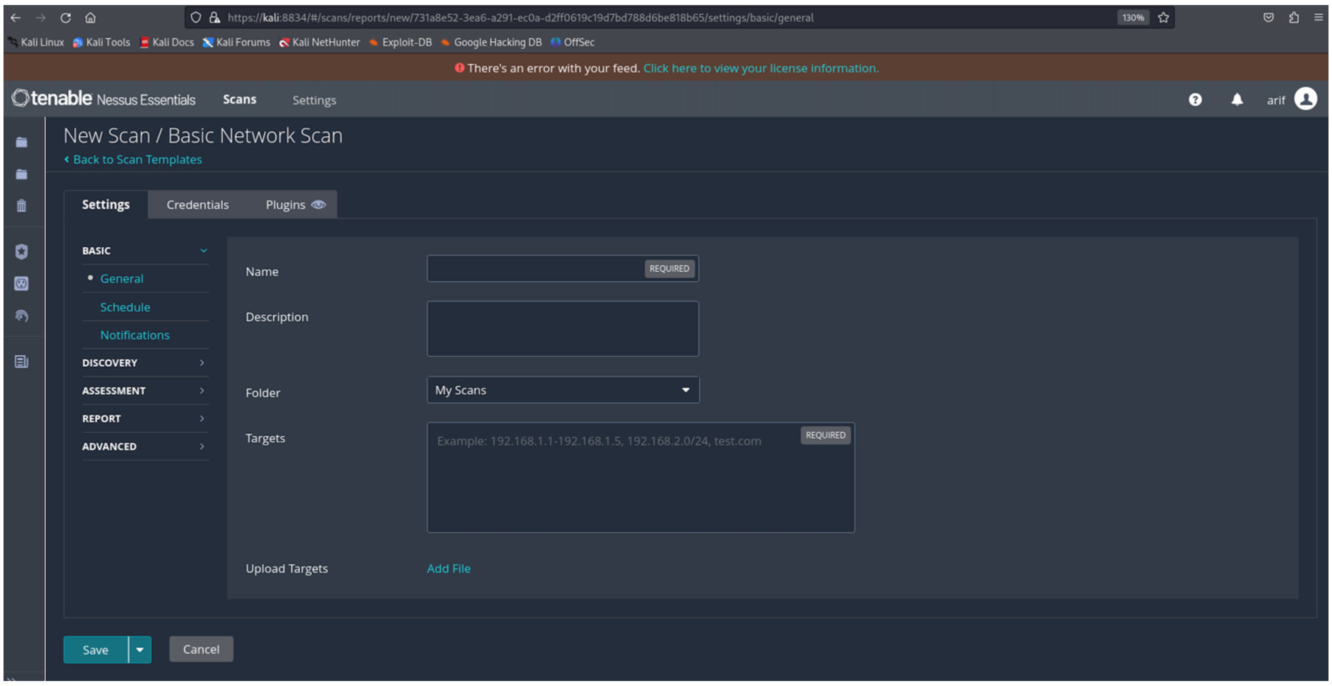
Spectre and Meltdown
Remote and local checks for CVE-2017-5753, CVE-2017-5715, and CVE-2017-5754

WannaCry Ransomware
Remote and local checks for MS17-010.

Ripple20 Remote Scan
A remote scan to fingerprint hosts potentially running the Treck stack in the network.

Zerologon Remote Scan
A remote scan to detect Microsoft Netlogon Elevation of Privilege (Zerologon).

- Let us perform a basic scan by clicking on the Basic Network Scan in the above screenshot and then enter the details about your scan like the IP of your **Metasploitable2** machine. Once done you can click the save button.



https://kali:8834/#/scans/reports/new/731a8e52-3ea6-a291-ec0a-d2ff0619c19d7bd788d6be818b65/settings/basic/general

There's an error with your feed. Click here to view your license information.

tenable Nessus Essentials Scans Settings arif

New Scan / Basic Network Scan
‹ Back to Scan Templates

Settings Credentials Plugins

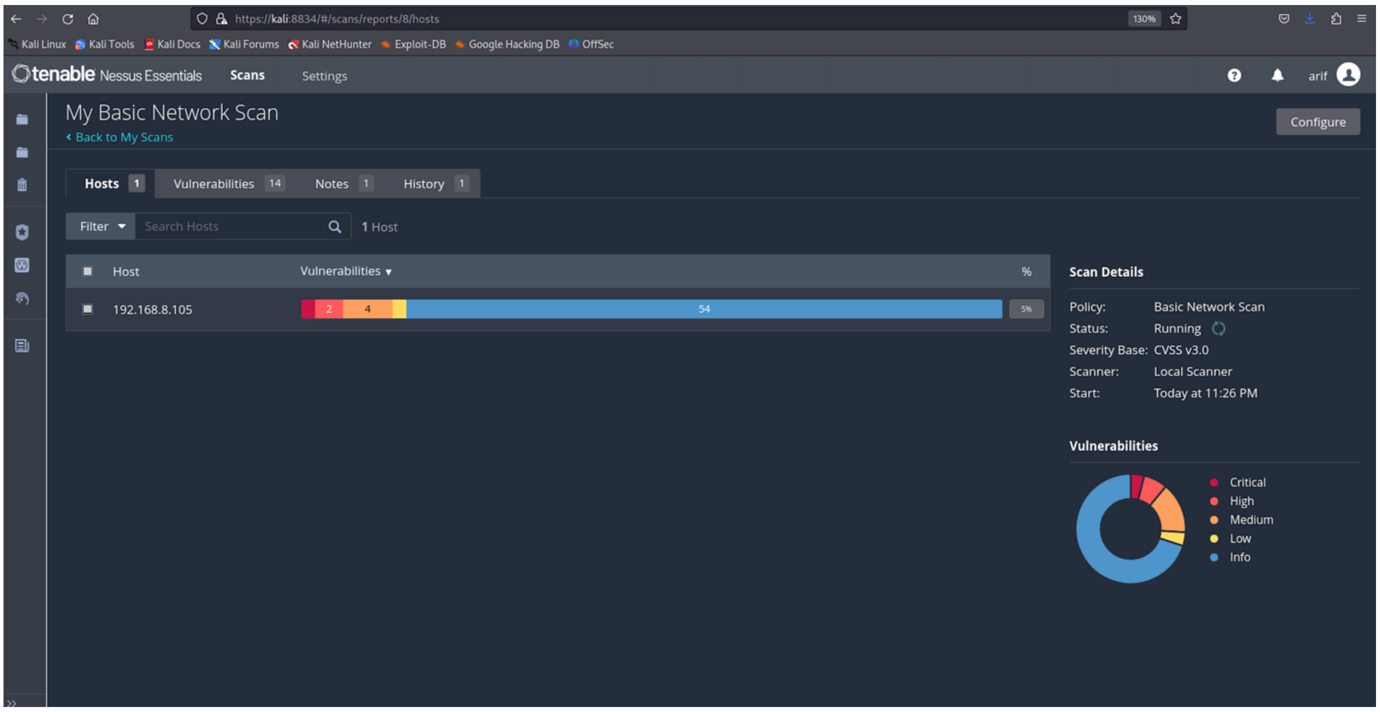
BASIC
- General
- Schedule
- Notifications

DISCOVERY
ASSESSMENT
REPORT
ADVANCED

Name: REQUIRED
Description:
Folder: My Scans
Targets: Example: 192.168.1.1-192.168.1.5, 192.168.2.0/24, test.com REQUIRED
Upload Targets: Add File

Save Cancel

- You can schedule or run your saved scan at time of your convenience. The following screenshot shows the scan in progress. This may take a bit of time.



My Basic Network Scan

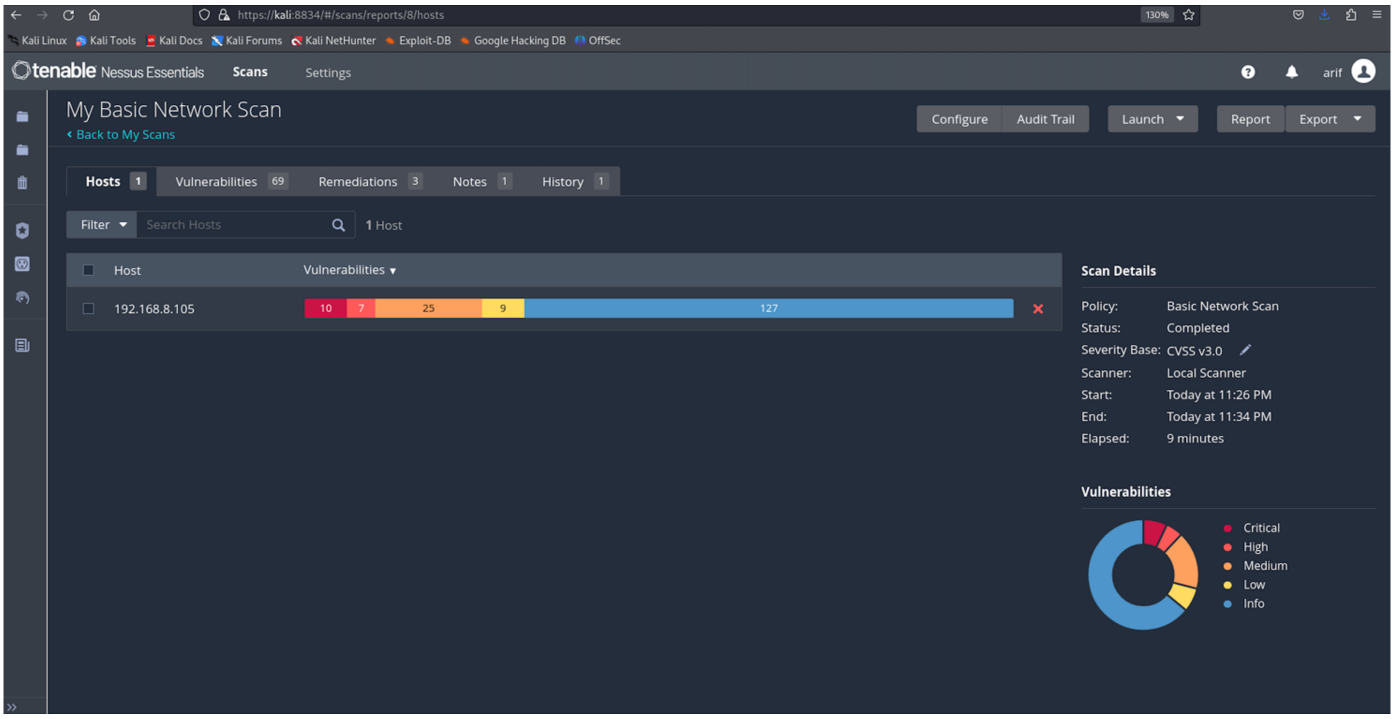
Host	Vulnerabilities	%
192.168.8.105	2 4 54	5%

Scan Details

Policy:	Basic Network Scan
Status:	Running
Severity Base:	CVSS v3.0
Scanner:	Local Scanner
Start:	Today at 11:26 PM

Vulnerabilities

- Following screenshot shows the output when the scan is completed, showing 10 critical, 7 high, 25 medium and 9 low vulnerabilities and 127 information disclosures.



My Basic Network Scan

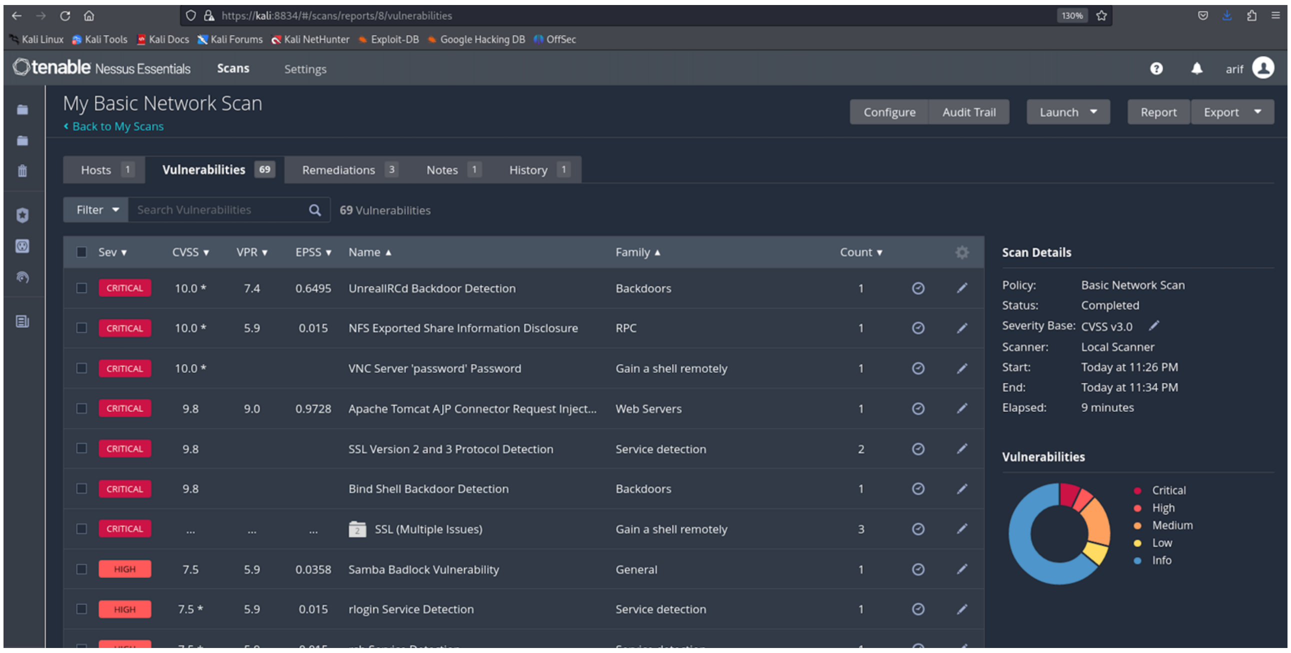
Host	Vulnerabilities
192.168.8.105	10 7 25 9 127

Scan Details

Policy:	Basic Network Scan
Status:	Completed
Severity Base:	CVSS v3.0
Scanner:	Local Scanner
Start:	Today at 11:26 PM
End:	Today at 11:34 PM
Elapsed:	9 minutes

Vulnerabilities

- You can click the **Vulnerabilities tab** to check the total 69 vulnerabilities that were detected by `nessus` on Metasploitable2. , can search for a specific one or use filter.



Sev	CVSS	VPR	EPSS	Name	Family	Count
CRITICAL	10.0 *	7.4	0.6495	UnrealIRCd Backdoor Detection	Backdoors	1
CRITICAL	10.0 *	5.9	0.015	NFS Exported Share Information Disclosure	RPC	1
CRITICAL	10.0 *			VNC Server 'password' Password	Gain a shell remotely	1
CRITICAL	9.8	9.0	0.9728	Apache Tomcat AJP Connector Request Inject...	Web Servers	1
CRITICAL	9.8			SSL Version 2 and 3 Protocol Detection	Service detection	2
CRITICAL	9.8			Bind Shell Backdoor Detection	Backdoors	1
CRITICAL	...	...	...	SSL (Multiple Issues)	Gain a shell remotely	3
HIGH	7.5	5.9	0.0358	Samba Badlock Vulnerability	General	1
HIGH	7.5 *	5.9	0.015	rlogin Service Detection	Service detection	1

Scan Details

Policy:	Basic Network Scan
Status:	Completed
Severity Base:	CVSS v3.0
Scanner:	Local Scanner
Start:	Today at 11:26 PM
End:	Today at 11:34 PM
Elapsed:	9 minutes

- Click critical vulnerability, ***UnrealIRCd Backdoor Detection*** (CVE-2010-2075) having a CVSS of 10.0, to check out the details about this vulnerability, which exist in Internet Relay Chat server. Moreover, you can Google this vulnerability and check out if there exist an exploit for this specific vulnerability and how to use it. (More on this later)
- Click another critical vulnerability, *NFS Exported Share Information Disclosure* (CVE-1999-0629) having a CVSS of 10.0, to check out the details about this vulnerability. Moreover, you can Google this vulnerability and check out if there exist an exploit for this specific vulnerability and how to use it. (More on this later)
- Click *VNC Server 'password' Password*, (CVE-1999-0503) having a CVSS of 10.0. It says that the VNC server running on the remote host is secured with a weak password. Nessus was able to login using VNC authentication and a password of 'password'. A remote unauthenticated attacker could exploit this to take control of the system. (More on this later)
- Click *Bind Shell Backdoor Detection*, (CVE-2001-0500) having a CVSS of 9.8. This is a buffer overflow vulnerability, which says that 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. (More on this later)

To Do:

- Students should visit https://www.cve.org/ and check out the details of the detected CVEs
- Students are also advised to run a basic `nessus` scan on some Windows machine as well (which has not been updated for a bit of time).

OpenVAS https://greenbone.github.io/docs/latest/

OpenVAS (Open Vulnerability Assessment System) is an open-source framework used for network vulnerability scanning and management. It is designed to identify security vulnerabilities in networked systems and services, providing comprehensive reporting and remediation guidance. OpenVAS is widely used by security professionals for vulnerability assessment, compliance auditing, and network security monitoring.

Installing and Running OpenVAS

Installing OpenVAS on Kali Linux is the fastest and recommended way. On other distributions OpenVAS needs to be compiled from source, that is error prone and cumbersome. On Kali Linux you can install OpenVAS with the following commands:

```
$ sudo apt install gvm -y
$ sudo apt install openvas -y
$ sudo gvm-setup // This will do automatic OpenVAS configuration
```

Note: A password will be printed on the screen. Make sure to save that.

```
$ sudo gvm-check-setup // to verify installation
```

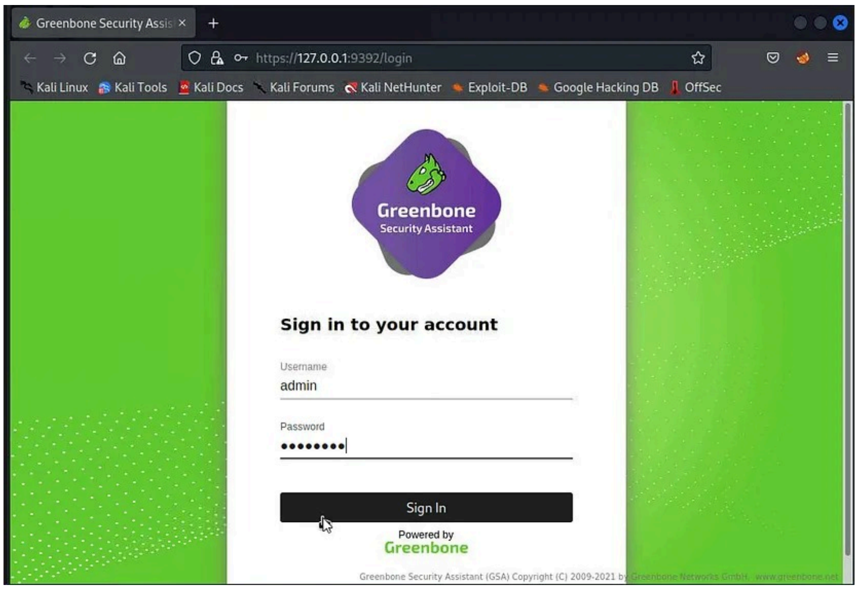
Run OpenVAS using command:

```
$ sudo systemctl start gvmd.service
$ sudo gvm-start
```

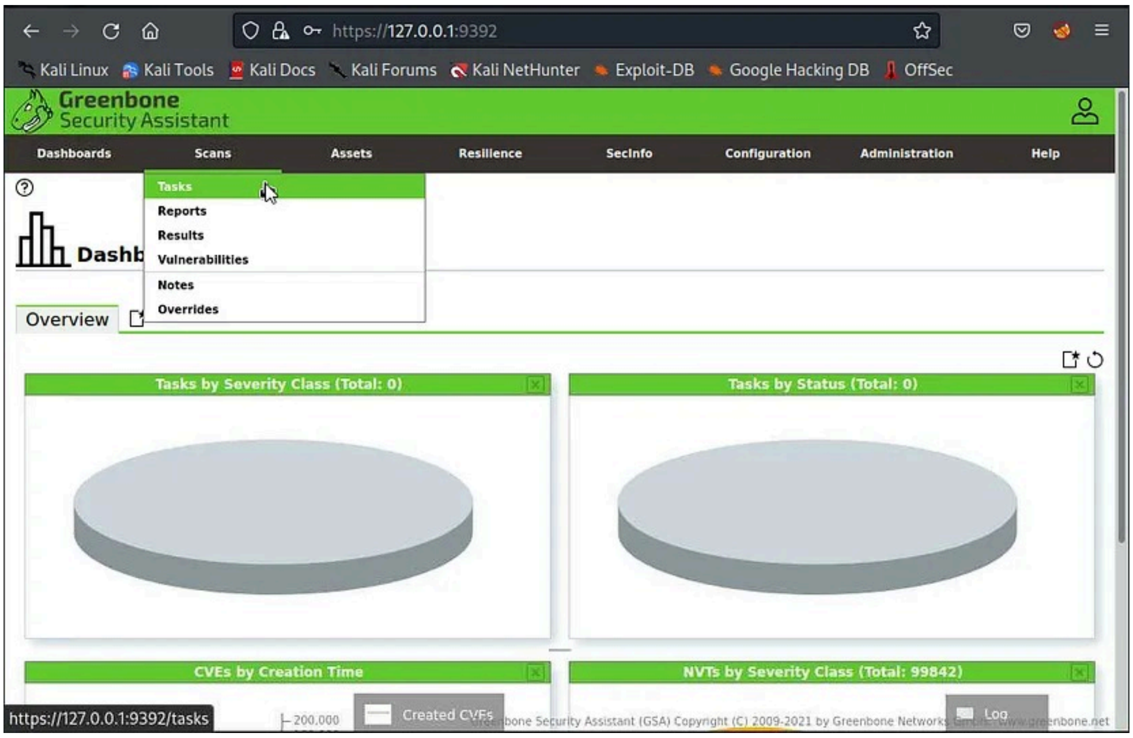
This will start OpenVAS and can be accessed from **https://127.0.0.1:9392**

Login by entering username: `admin` and password: printed on the screen while gvm-setup is running. In case if you want to change the password to admin, use following command:

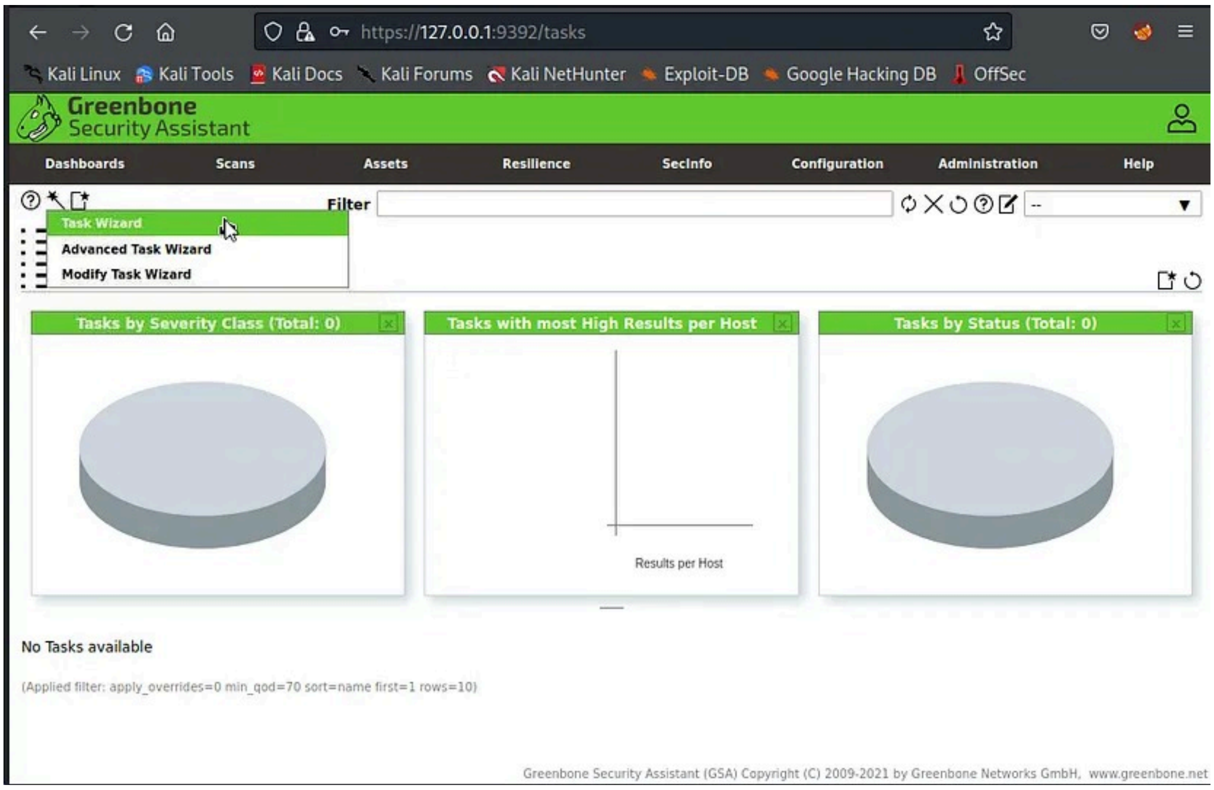
```
$ sudo -E -u _gvm -g _gvm gvmd –user=admin –new-password=admin
```



- Click **Scans** menu and select **Tasks**.



- The Tasks page is displayed. Click Task Wizard on the upper left side.

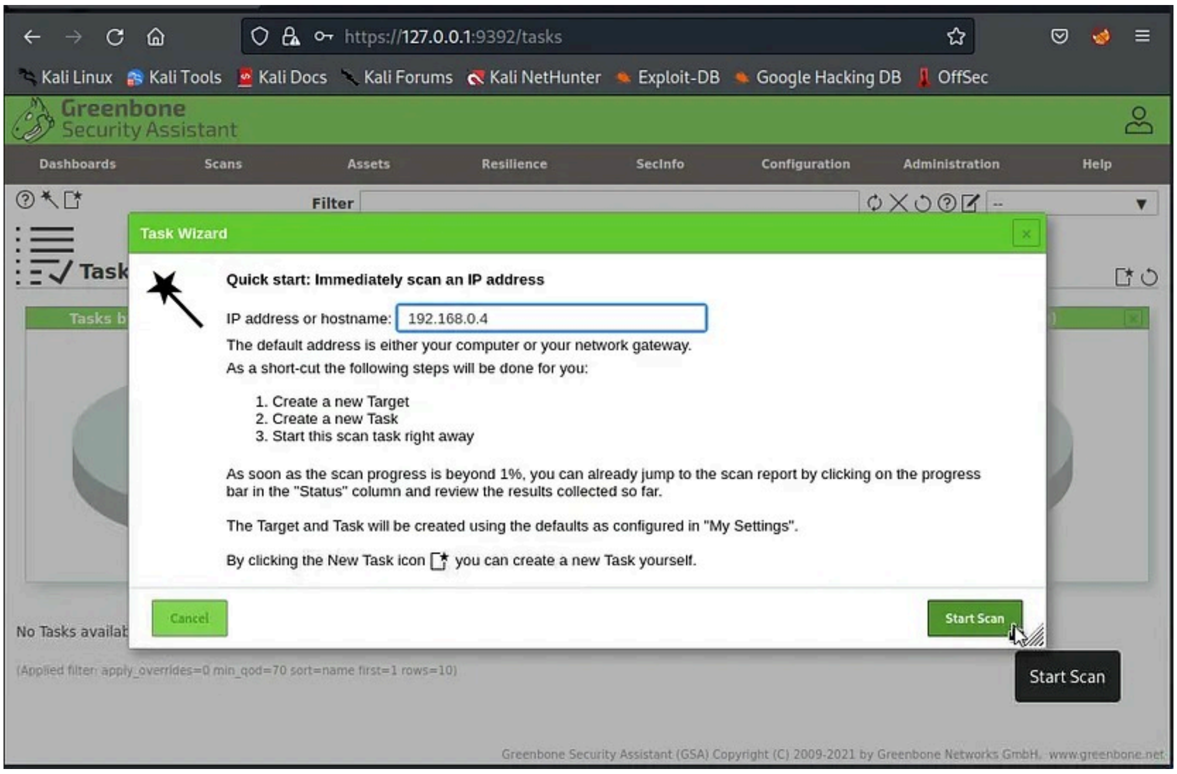

https://127.0.0.1:9392/tasks
Greenbone Security Assistant
Dashboards
Scans
Assets
Resilience
SecInfo
Configuration
Administration
Help
Filter
Task Wizard
Advanced Task Wizard
Modify Task Wizard
Tasks by Severity Class (Total: 0)
Tasks with most High Results per Host
Tasks by Status (Total: 0)
Results per Host

No Tasks available

(Applied filter: apply_overrides=0 min_qod=70 sort=name first=1 rows=10)

Greenbone Security Assistant (GSA) Copyright (C) 2009-2021 by Greenbone Networks GmbH, www.greenbone.net

- In the **Task Wizard** pop-up window, enter the following in the IP address or hostname field: `<target-IP>` and click **Start Scan**



Task Wizard

Quick start: Immediately scan an IP address

IP address or hostname: 192.168.0.4

The default address is either your computer or your network gateway.
As a short-cut the following steps will be done for you:

1. Create a new Target
2. Create a new Task
3. Start this scan task right away

As soon as the scan progress is beyond 1%, you can already jump to the scan report by clicking on the progress bar in the "Status" column and review the results collected so far.

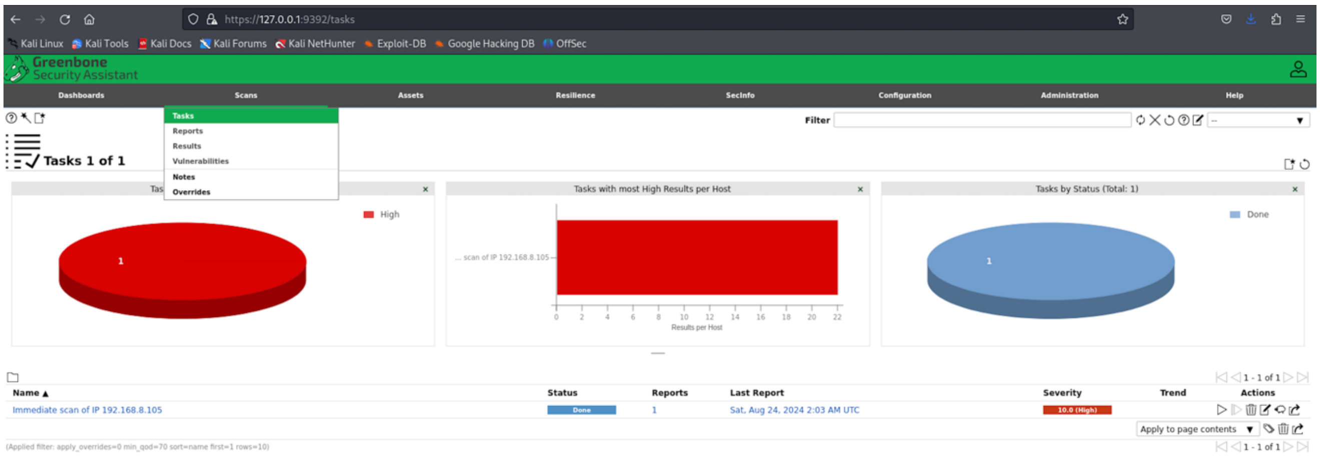
The Target and Task will be created using the defaults as configured in "My Settings".

By clicking the New Task icon you can create a new Task yourself.

(Applied filter: apply_overrides=0 min_qod=70 sort=name first=1 rows=10)

Greenbone Security Assistant (GSA) Copyright (C) 2009-2021 by Greenbone Networks GmbH, www.greenbone.net

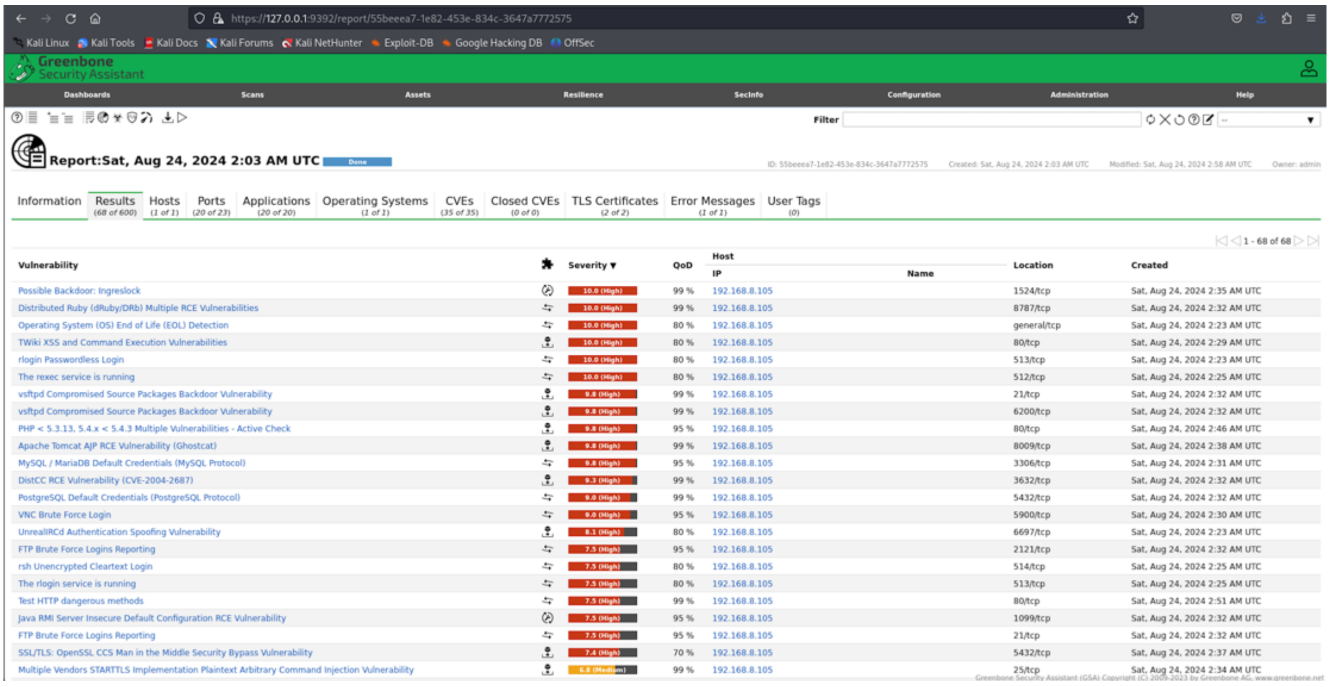
- The task will be created and run. It may take some time to run all the tests. Finally, on Task 1 of 1, under the Scans menu click the **Tasks** field.



Name ▲	Status	Reports	Last Report	Severity	Trend	Actions
Immediate scan of IP 192.168.8.105	Done	1	Sat, Aug 24, 2024 2:03 AM UTC	10.0 (High)		

(Applied filter: apply_overrides=0 min_qod=70 sort=name first=1 rows=10)

- You click the Done button and it will give you a detailed Report, having multiple tabs. Play with it to learn more about vulnerabilities. This is shown in the following screenshot



Report:Sat, Aug 24, 2024 2:03 AM UTC Done

ID: 55beeea7-1e82-453e-834c-3647a7772575 Created: Sat, Aug 24, 2024 2:03 AM UTC Modified: Sat, Aug 24, 2024 2:58 AM UTC Owner: admin

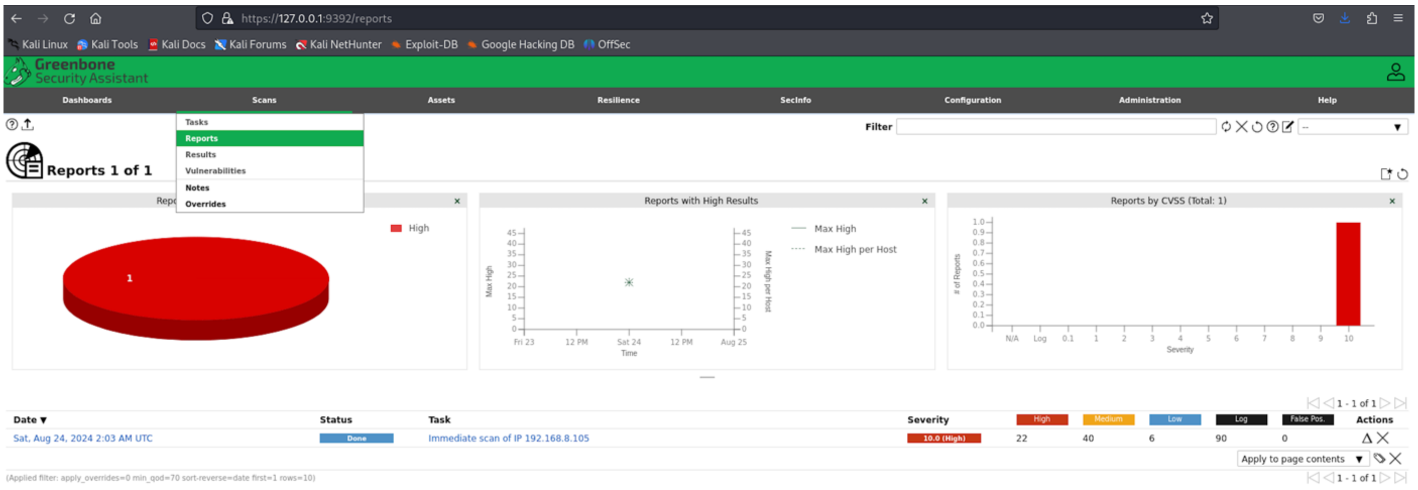
Information | Results (68 of 600) | Hosts (1 of 1) | Ports (20 of 23) | Applications (20 of 20) | Operating Systems (1 of 1) | CVEs (35 of 35) | Closed CVEs (0 of 0) | TLS Certificates (2 of 2) | Error Messages (1 of 1) | User Tags (0)

1 - 68 of 68

Vulnerability	Severity ▼	QoD	Host IP	Host Name	Location	Created
Possible Backdoor: Ingreslock	10.0 (High)	99 %	192.168.8.105		1524/tcp	Sat, Aug 24, 2024 2:35 AM UTC
Distributed Ruby (dRuby/DRb) Multiple RCE Vulnerabilities	10.0 (High)	99 %	192.168.8.105		8787/tcp	Sat, Aug 24, 2024 2:32 AM UTC
Operating System (OS) End of Life (EOL) Detection	10.0 (High)	80 %	192.168.8.105		general/tcp	Sat, Aug 24, 2024 2:23 AM UTC
TWiki XSS and Command Execution Vulnerabilities	10.0 (High)	80 %	192.168.8.105		80/tcp	Sat, Aug 24, 2024 2:29 AM UTC
rlogin Passwordless Login	10.0 (High)	80 %	192.168.8.105		513/tcp	Sat, Aug 24, 2024 2:23 AM UTC
The rexec service is running	10.0 (High)	80 %	192.168.8.105		512/tcp	Sat, Aug 24, 2024 2:25 AM UTC
vsftpd Compromised Source Packages Backdoor Vulnerability	9.8 (High)	99 %	192.168.8.105		21/tcp	Sat, Aug 24, 2024 2:32 AM UTC
vsftpd Compromised Source Packages Backdoor Vulnerability	9.8 (High)	99 %	192.168.8.105		6200/tcp	Sat, Aug 24, 2024 2:32 AM UTC
PHP < 5.3.13, 5.4.x < 5.4.3 Multiple Vulnerabilities - Active Check	9.8 (High)	95 %	192.168.8.105		80/tcp	Sat, Aug 24, 2024 2:46 AM UTC
Apache Tomcat AJP RCE Vulnerability (Ghostcat)	9.8 (High)	99 %	192.168.8.105		8009/tcp	Sat, Aug 24, 2024 2:38 AM UTC
MySQL / MariaDB Default Credentials (MySQL Protocol)	9.8 (High)	95 %	192.168.8.105		3306/tcp	Sat, Aug 24, 2024 2:31 AM UTC
DistCC RCE Vulnerability (CVE-2004-2687)	9.3 (High)	99 %	192.168.8.105		3632/tcp	Sat, Aug 24, 2024 2:32 AM UTC
PostgreSQL Default Credentials (PostgreSQL Protocol)	9.0 (High)	99 %	192.168.8.105		5432/tcp	Sat, Aug 24, 2024 2:32 AM UTC
VNC Brute Force Login	9.0 (High)	95 %	192.168.8.105		5900/tcp	Sat, Aug 24, 2024 2:30 AM UTC
UnrealIRCd Authentication Spoofing Vulnerability	8.1 (High)	80 %	192.168.8.105		6697/tcp	Sat, Aug 24, 2024 2:23 AM UTC
FTP Brute Force Logins Reporting	7.5 (High)	95 %	192.168.8.105		2121/tcp	Sat, Aug 24, 2024 2:32 AM UTC
rsh Unencrypted Cleartext Login	7.5 (High)	80 %	192.168.8.105		514/tcp	Sat, Aug 24, 2024 2:25 AM UTC
The rlogin service is running	7.5 (High)	80 %	192.168.8.105		513/tcp	Sat, Aug 24, 2024 2:25 AM UTC
Test HTTP dangerous methods	7.5 (High)	99 %	192.168.8.105		80/tcp	Sat, Aug 24, 2024 2:51 AM UTC
Java RMI Server Insecure Default Configuration RCE Vulnerability	7.5 (High)	95 %	192.168.8.105		1099/tcp	Sat, Aug 24, 2024 2:32 AM UTC
FTP Brute Force Logins Reporting	7.5 (High)	95 %	192.168.8.105		21/tcp	Sat, Aug 24, 2024 2:32 AM UTC
SSL/TLS: OpenSSL CCS Man in the Middle Security Bypass Vulnerability	7.4 (High)	70 %	192.168.8.105		5432/tcp	Sat, Aug 24, 2024 2:37 AM UTC
Multiple Vendors STARTTLS Implementation Plaintext Arbitrary Command Injection Vulnerability	6.8 (Medium)	99 %	192.168.8.105		25/tcp	Sat, Aug 24, 2024 2:34 AM UTC

Greenbone Security Assistant (GSA) Copyright (C) 2009-2023 by Greenbone AG, www.greenbone.net

- **Reports** field.



1 - 1 of 1

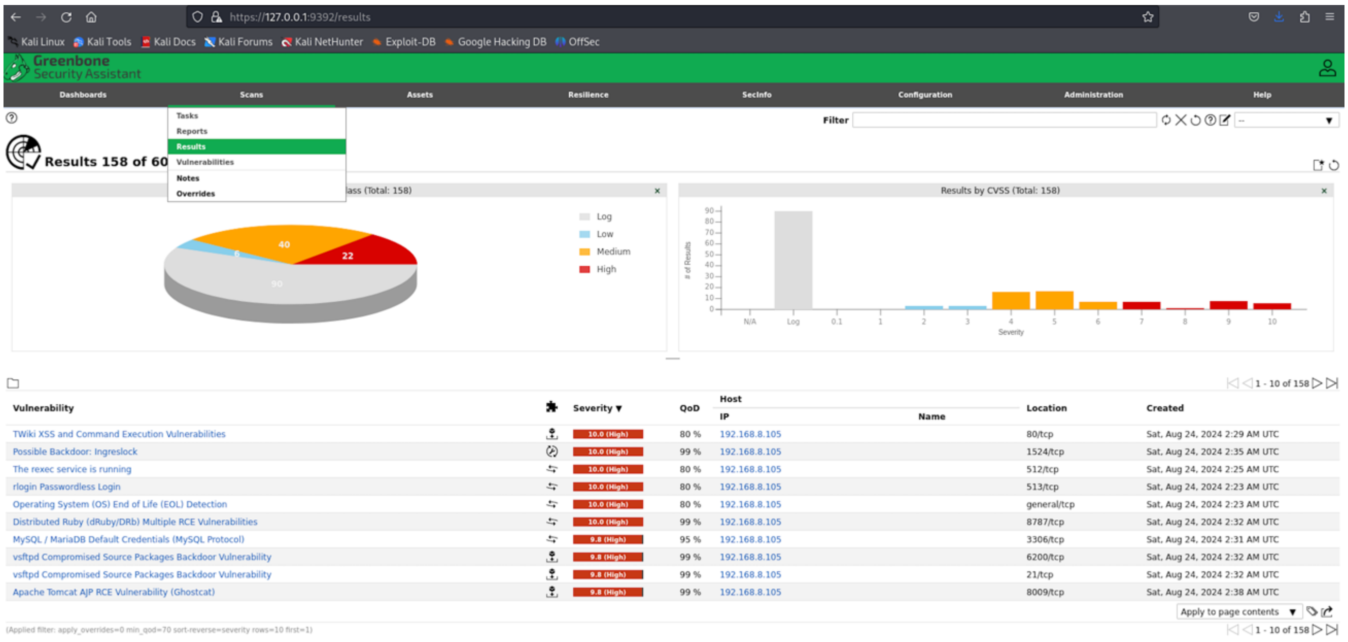
Date ▼	Status	Task	Severity	High	Medium	Low	Log	False Pos.	Actions
Sat, Aug 24, 2024 2:03 AM UTC	Done	Immediate scan of IP 192.168.8.105	10.0 (High)	22	40	6	90	0	

Apply to page contents

(Applied filter: apply_overrides=0 min_qod=70 sort-reverse=date first=1 rows=10)

1 - 1 of 1

- **Results** field.

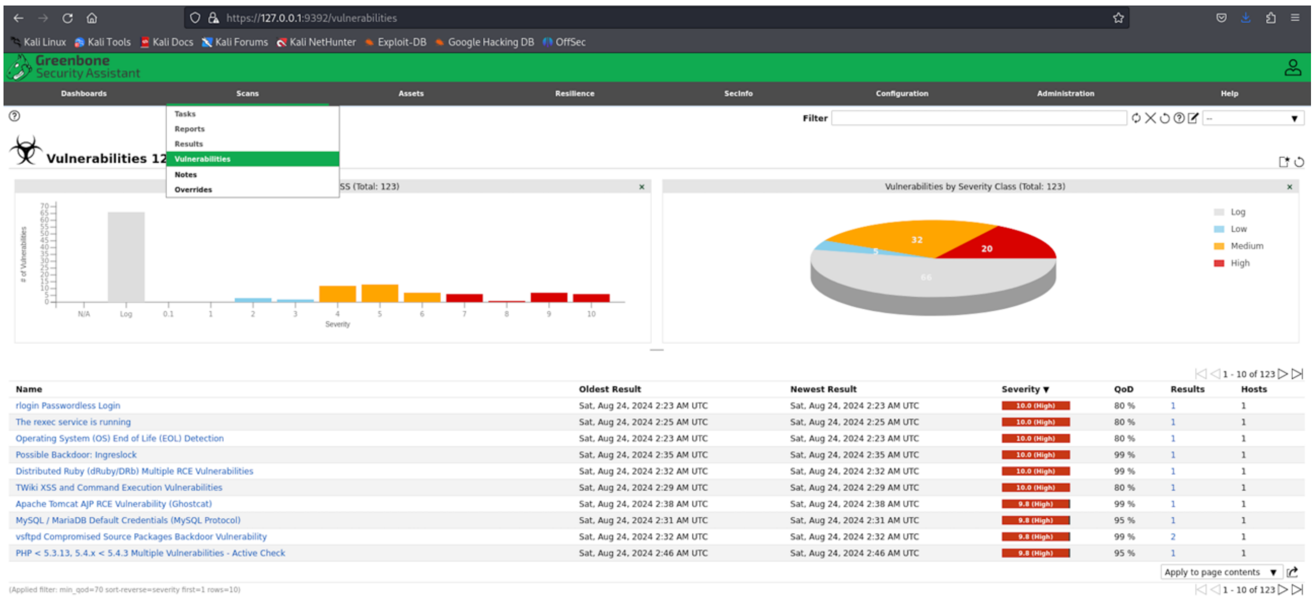


Vulnerability	Severity ▼	QoD	Host IP	Host Name	Location	Created
TWiki XSS and Command Execution Vulnerabilities	10.0 (High)	80 %	192.168.8.105		80/tcp	Sat, Aug 24, 2024 2:29 AM UTC
Possible Backdoor: Ingreslock	10.0 (High)	99 %	192.168.8.105		1524/tcp	Sat, Aug 24, 2024 2:35 AM UTC
The rexec service is running	10.0 (High)	80 %	192.168.8.105		512/tcp	Sat, Aug 24, 2024 2:25 AM UTC
rlogin Passwordless Login	10.0 (High)	80 %	192.168.8.105		513/tcp	Sat, Aug 24, 2024 2:23 AM UTC
Operating System (OS) End of Life (EOL) Detection	10.0 (High)	80 %	192.168.8.105		general/tcp	Sat, Aug 24, 2024 2:23 AM UTC
Distributed Ruby (dRuby/DRb) Multiple RCE Vulnerabilities	10.0 (High)	99 %	192.168.8.105		8787/tcp	Sat, Aug 24, 2024 2:32 AM UTC
MySQL / MariaDB Default Credentials (MySQL Protocol)	9.8 (High)	95 %	192.168.8.105		3306/tcp	Sat, Aug 24, 2024 2:31 AM UTC
vsftpd Compromised Source Packages Backdoor Vulnerability	9.8 (High)	99 %	192.168.8.105		6200/tcp	Sat, Aug 24, 2024 2:32 AM UTC
vsftpd Compromised Source Packages Backdoor Vulnerability	9.8 (High)	99 %	192.168.8.105		21/tcp	Sat, Aug 24, 2024 2:32 AM UTC
Apache Tomcat AJP RCE Vulnerability (Ghostcat)	9.8 (High)	99 %	192.168.8.105		8009/tcp	Sat, Aug 24, 2024 2:38 AM UTC

1 - 10 of 158

(Applied filter: apply_overrides=0 min_qod=70 sort-reverse=severity rows=10 first=1)

- **Vulnerabilities** field.



Name	Oldest Result	Newest Result	Severity ▼	QoD	Results	Hosts
rlogin Passwordless Login	Sat, Aug 24, 2024 2:23 AM UTC	Sat, Aug 24, 2024 2:23 AM UTC	10.0 (High)	80 %	1	1
The rexec service is running	Sat, Aug 24, 2024 2:25 AM UTC	Sat, Aug 24, 2024 2:25 AM UTC	10.0 (High)	80 %	1	1
Operating System (OS) End of Life (EOL) Detection	Sat, Aug 24, 2024 2:23 AM UTC	Sat, Aug 24, 2024 2:23 AM UTC	10.0 (High)	80 %	1	1
Possible Backdoor: Ingreslock	Sat, Aug 24, 2024 2:35 AM UTC	Sat, Aug 24, 2024 2:35 AM UTC	10.0 (High)	99 %	1	1
Distributed Ruby (dRuby/DRb) Multiple RCE Vulnerabilities	Sat, Aug 24, 2024 2:32 AM UTC	Sat, Aug 24, 2024 2:32 AM UTC	10.0 (High)	99 %	1	1
TWiki XSS and Command Execution Vulnerabilities	Sat, Aug 24, 2024 2:29 AM UTC	Sat, Aug 24, 2024 2:29 AM UTC	10.0 (High)	80 %	1	1
Apache Tomcat AJP RCE Vulnerability (Ghostcat)	Sat, Aug 24, 2024 2:38 AM UTC	Sat, Aug 24, 2024 2:38 AM UTC	9.8 (High)	99 %	1	1
MySQL / MariaDB Default Credentials (MySQL Protocol)	Sat, Aug 24, 2024 2:31 AM UTC	Sat, Aug 24, 2024 2:31 AM UTC	9.8 (High)	95 %	1	1
vsftpd Compromised Source Packages Backdoor Vulnerability	Sat, Aug 24, 2024 2:32 AM UTC	Sat, Aug 24, 2024 2:32 AM UTC	9.8 (High)	99 %	2	1
PHP < 5.3.13, 5.4.x < 5.4.3 Multiple Vulnerabilities - Active Check	Sat, Aug 24, 2024 2:46 AM UTC	Sat, Aug 24, 2024 2:46 AM UTC	9.8 (High)	95 %	1	1

1 - 10 of 123

(Applied filter: min_qod=70 sort-reverse=severity first=1 rows=10)