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:

MEMBER 1: VENNAPUSA SARATHENDRA VENKATA SAI REDDY (20FE1A03B1).

MEMBER 2: YARRAMSETTY VENKATA RAVI KIRAN (20FE1A03B3).

MEMBER 3: YERRABOTHULA NAGA VEERA VENKATESWAR REDDY (20FE5A03B6).

PROJRCT REG NO: SBAP0007919.

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.

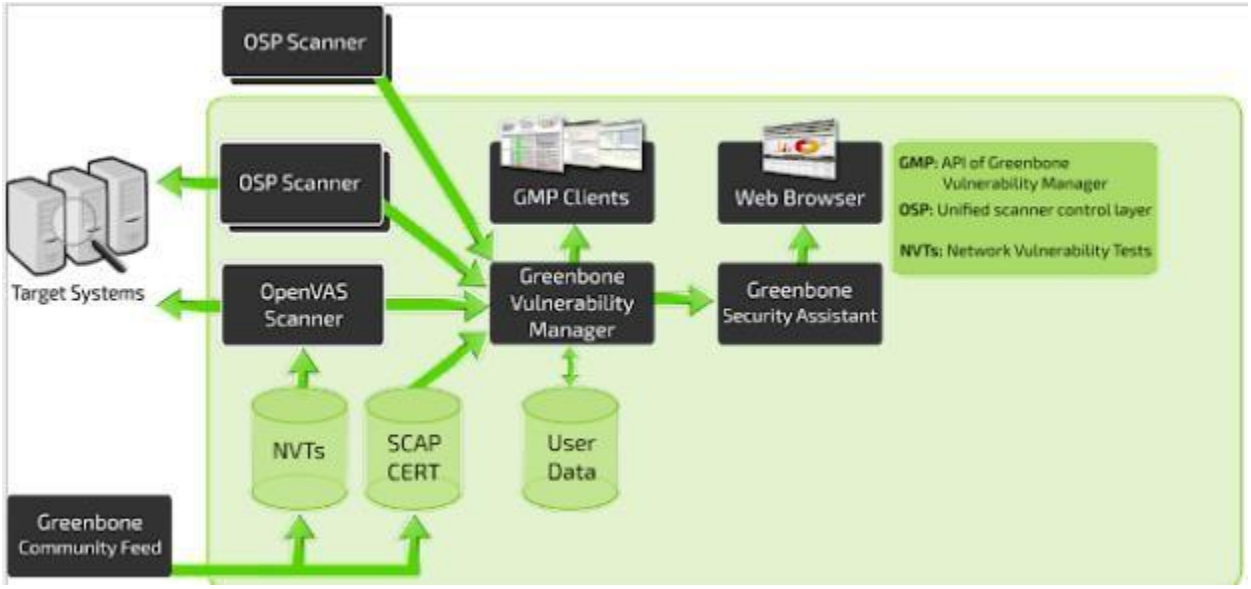
5. Controlled Exercise Execution: Implement safeguards and monitoring mechanisms during red team exercises to ensure activities remain within the predefined boundaries. Continuous monitoring and oversight can help identify and address any unexpected issues promptly.

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.

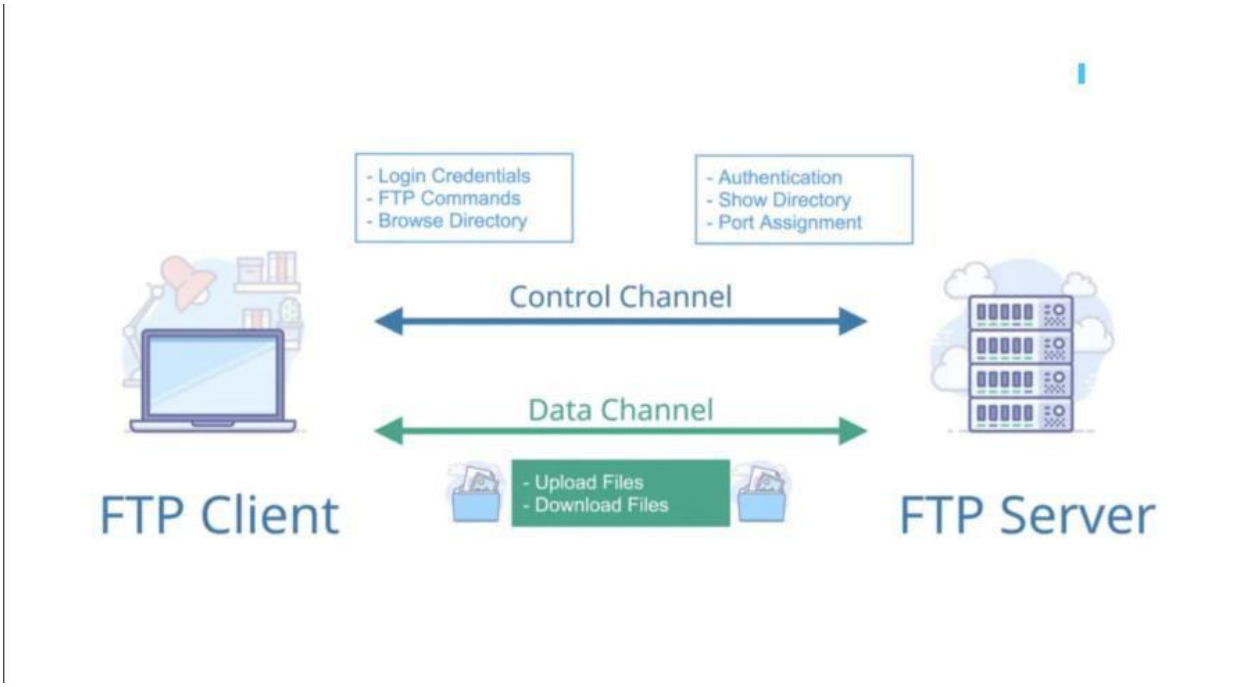
By implementing these proposed solutions, organizations can mitigate the potential for unintended consequences or collateral damage during red team exercises, ensuring a balance between realistic testing and minimizing disruptions to normal operations.

THEORITICAL ANALYSIS

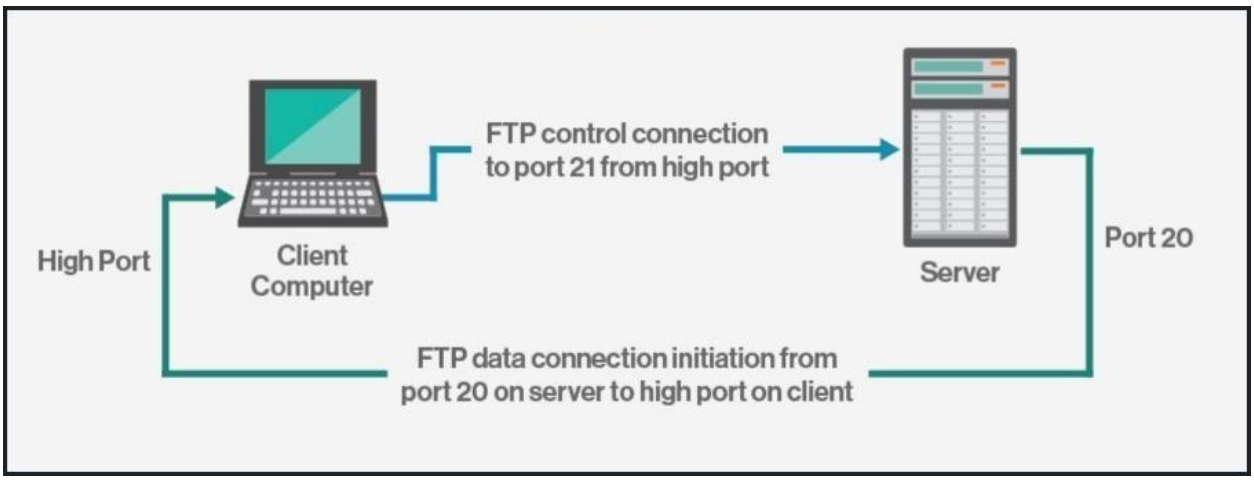
3.1 Block diagram



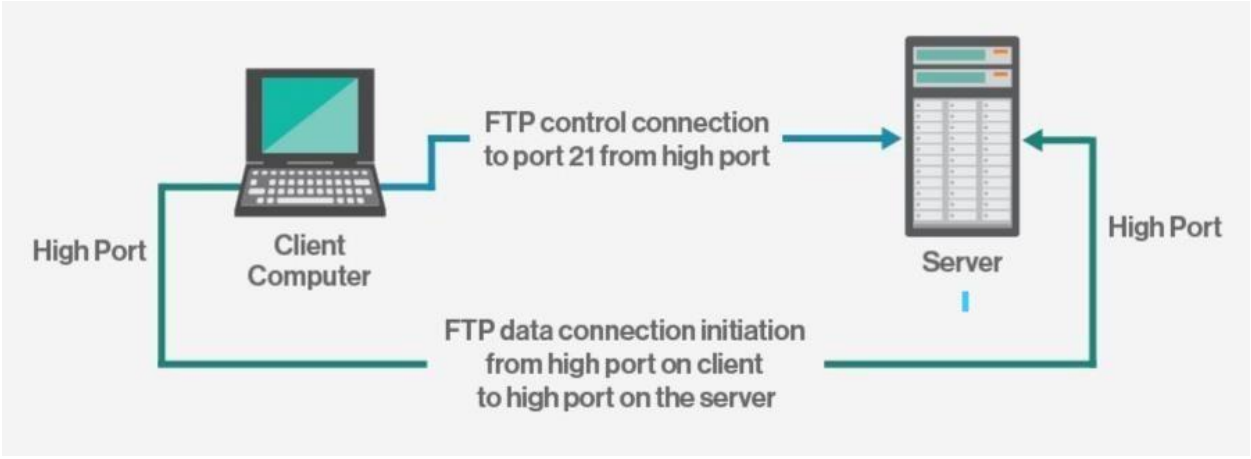
This is the working of an automatic vulnerability scanning tool OpenVAS which can be used in understanding the different vulnerabilities that can be present and thus be exploited to summarize solutions in order to mitigate them.



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:

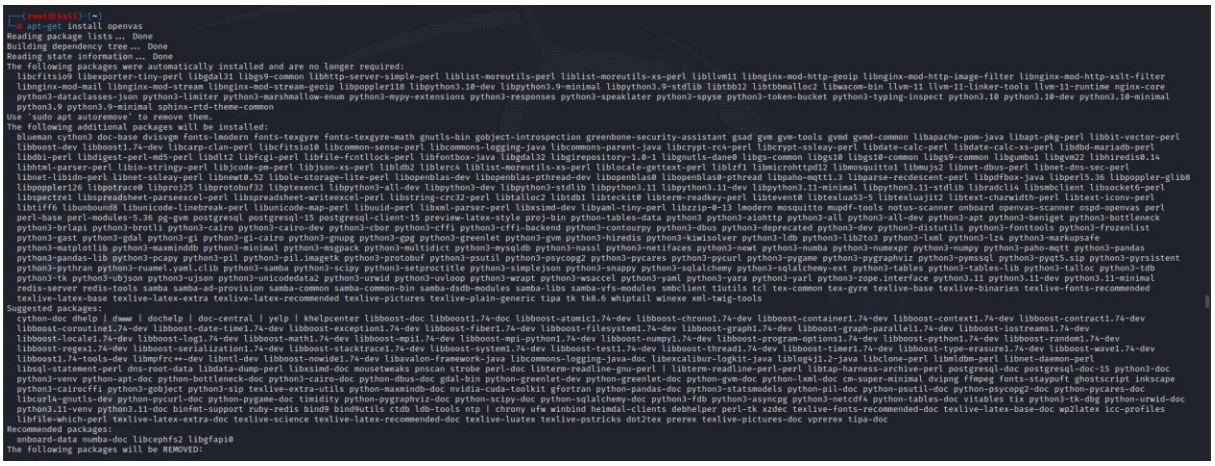
1. It is free and open source.
2. Numerous supported networks and systems
3. Large vulnerability check database
4. Strong scanning engine Simple to use.

can be combined with additional security instrument

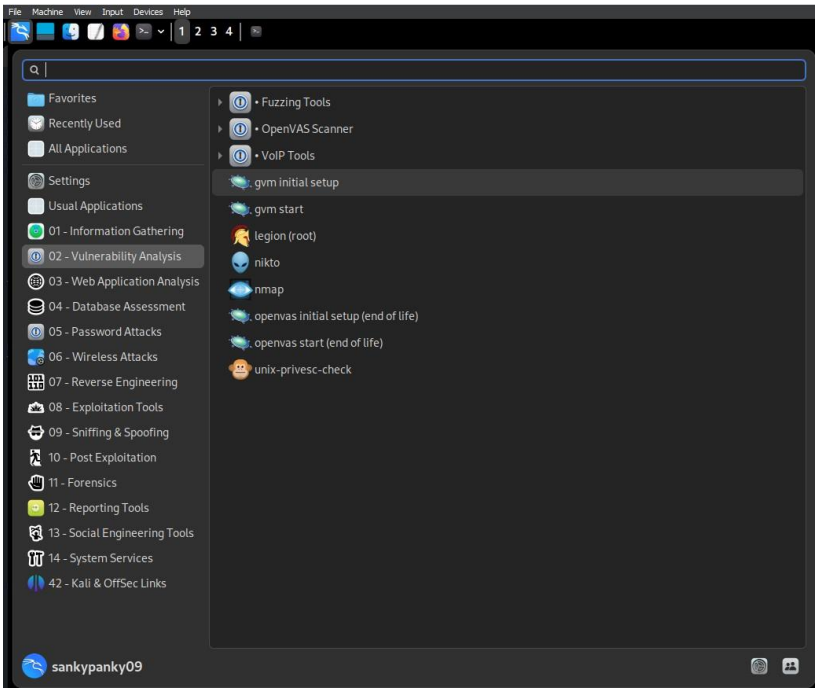
Step 1: Installing OpenVAS

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

We Type in the command to install it.

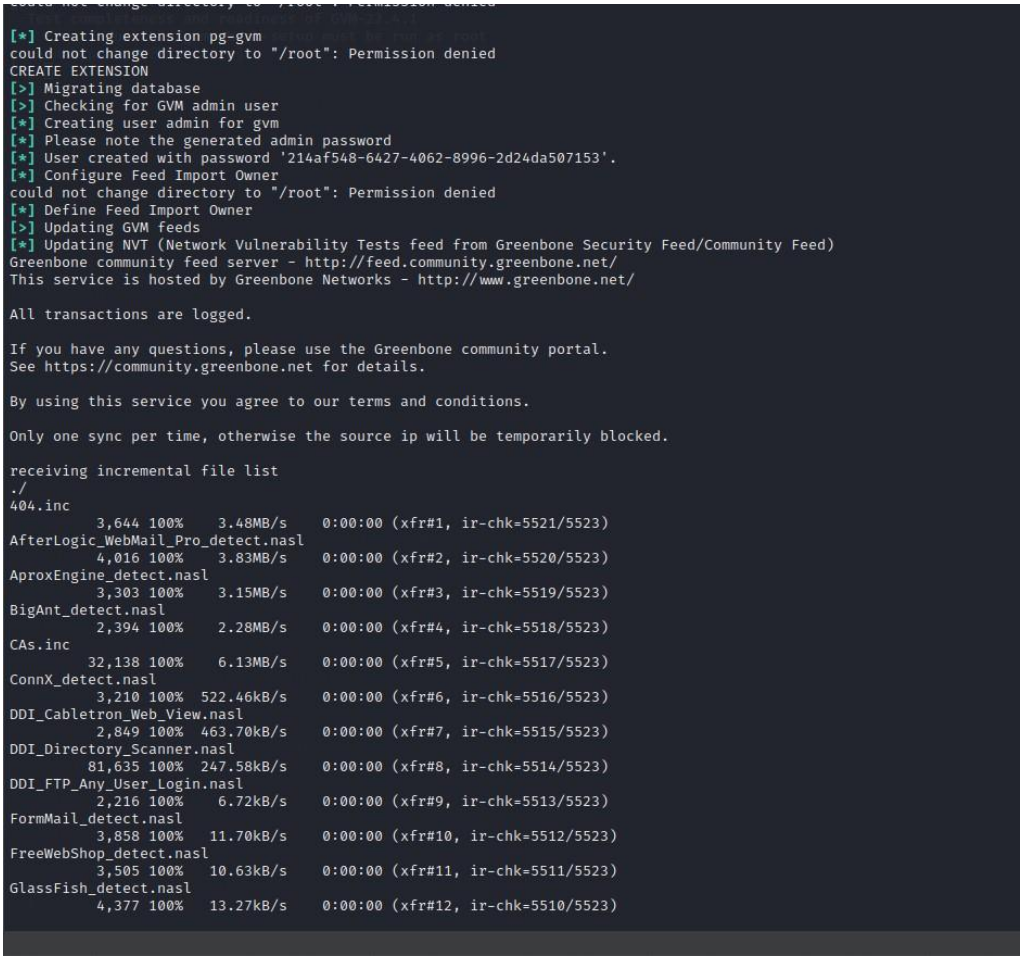


After Installation, we can successfully see that we are getting our required menus related to the OpenVAS, under the vulnerability analysis column.



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-ossp
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
```



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 GVMD 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 |            | libc       |
16451|pg-gvm|10|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.
```

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

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

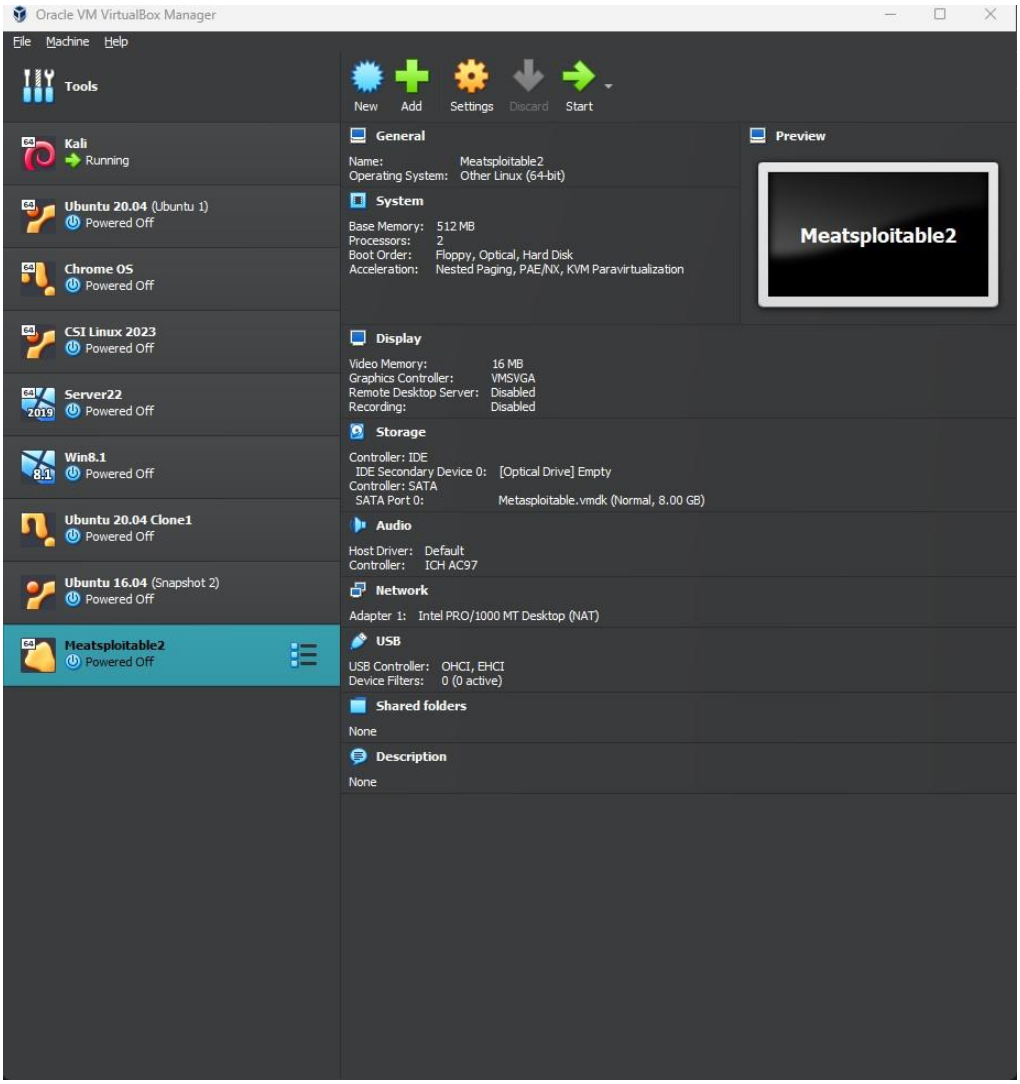
The steps to be followed in order to make the instance are below.

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

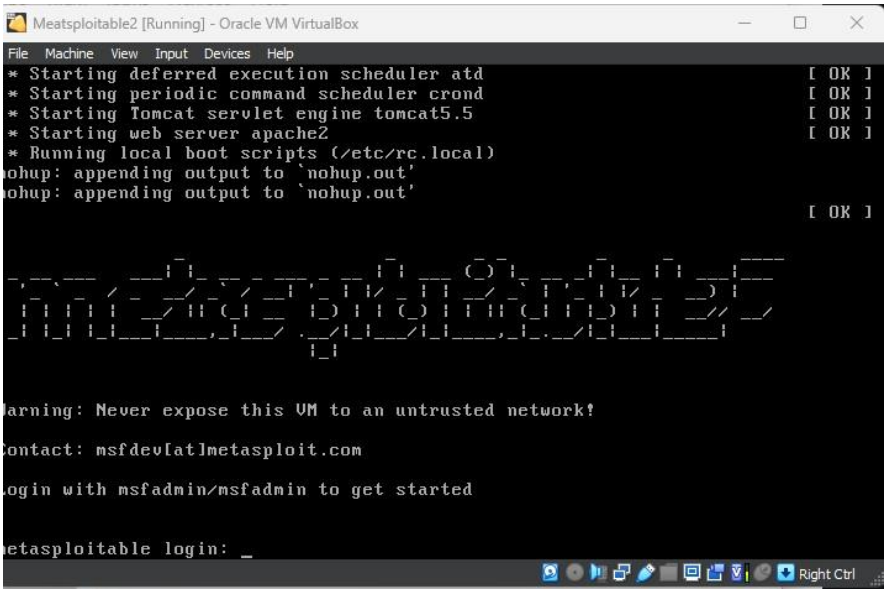
Then, we click the add on left top, and then select the downloaded .vmdk file from its location.

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

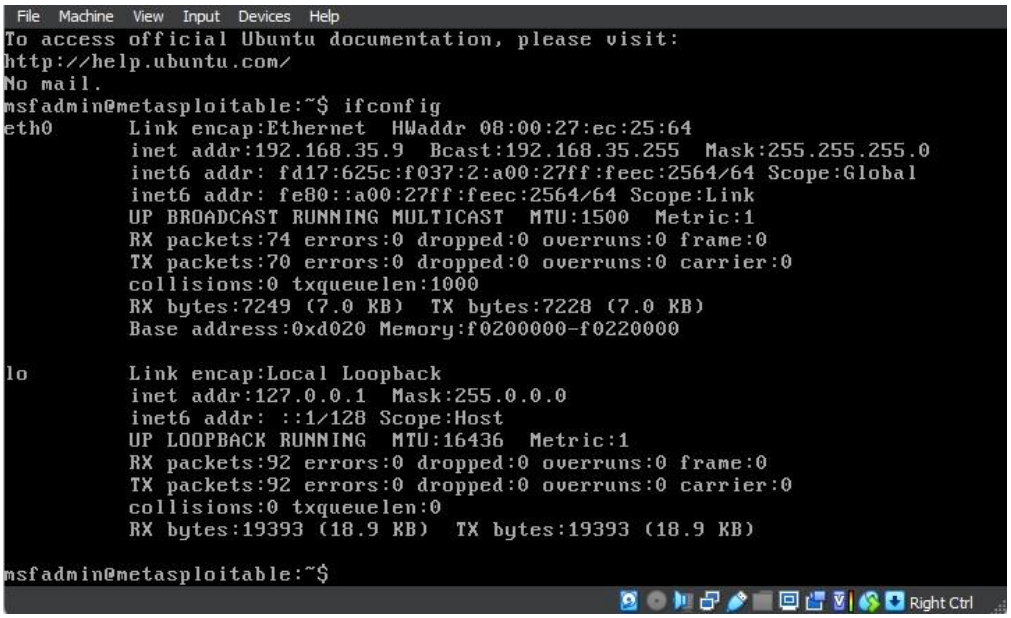
This is the final step, as the size of our virtual disk is also selected by default from the downloaded file configuration. We then click on next in order to finalize and save the entire settings



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



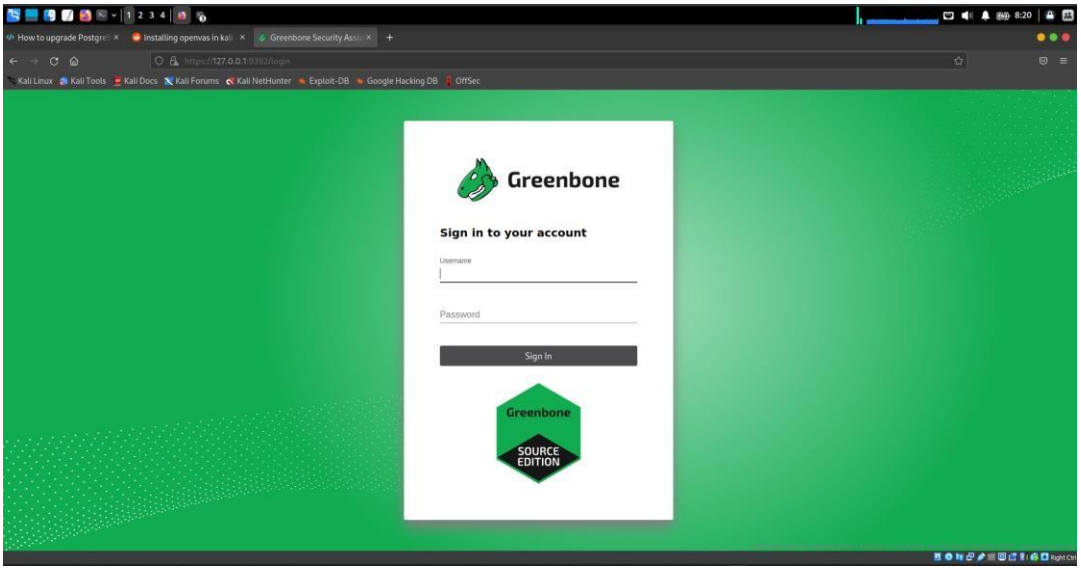
We will then enter the default login credentials, i.e., **login: msfadmin; pass: msfadmin.**



Step 3: Opening the OpenVAS web portal.

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

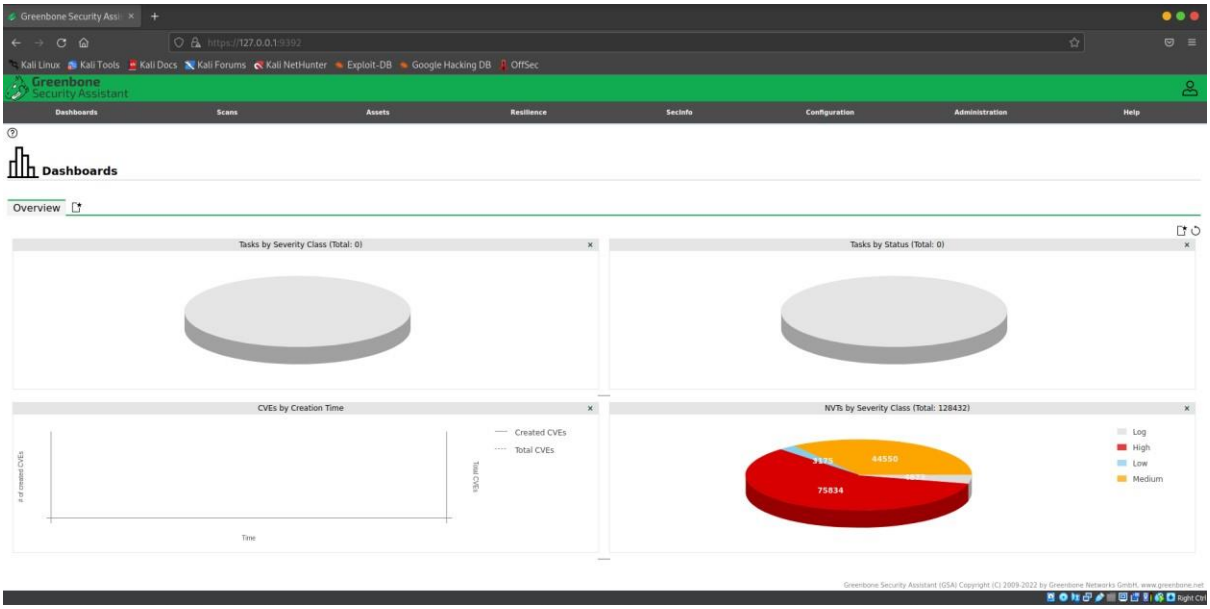
It will then, take us to the homepage of the OpenVAS.



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

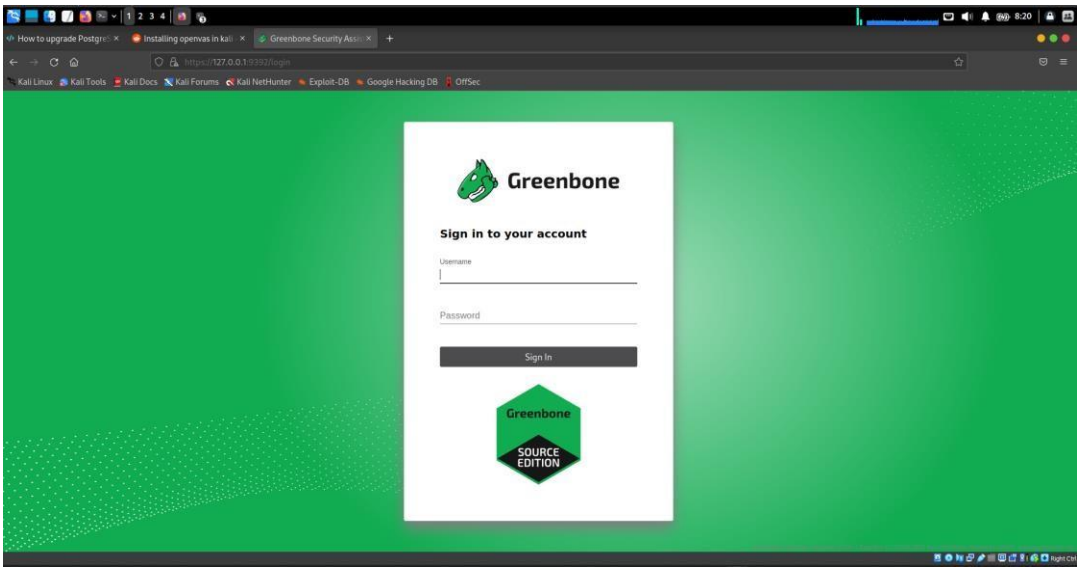
To begin scanning, we will be requiring the IP address of the target server/pc.

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

Step 3: Opening the OpenVAS web portal.

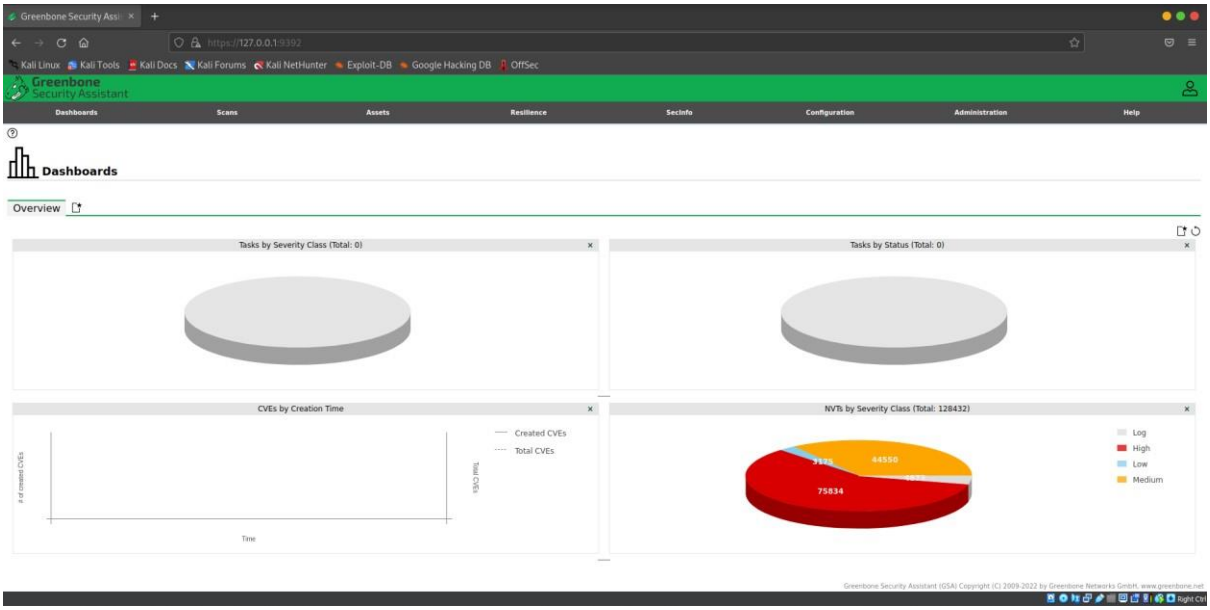
After successfully installing, go to browser and type the following IP Address to access the portal.

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

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

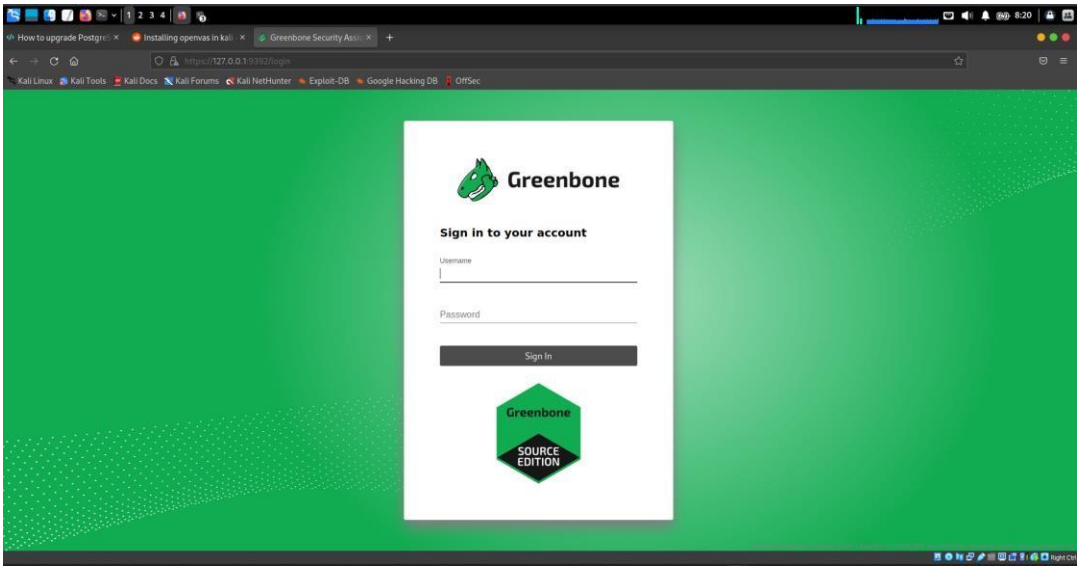
To begin scanning, we will be requiring the IP address of the target server/pc.

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.

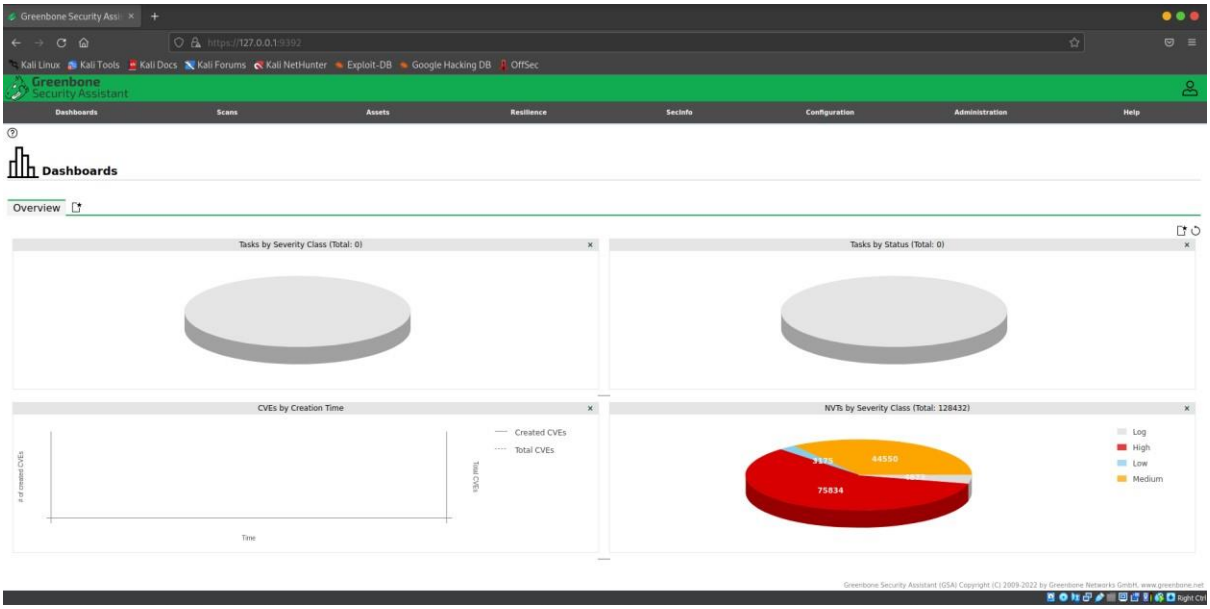
It will then, take us to the homepage of the OpenVAS.



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'.
```

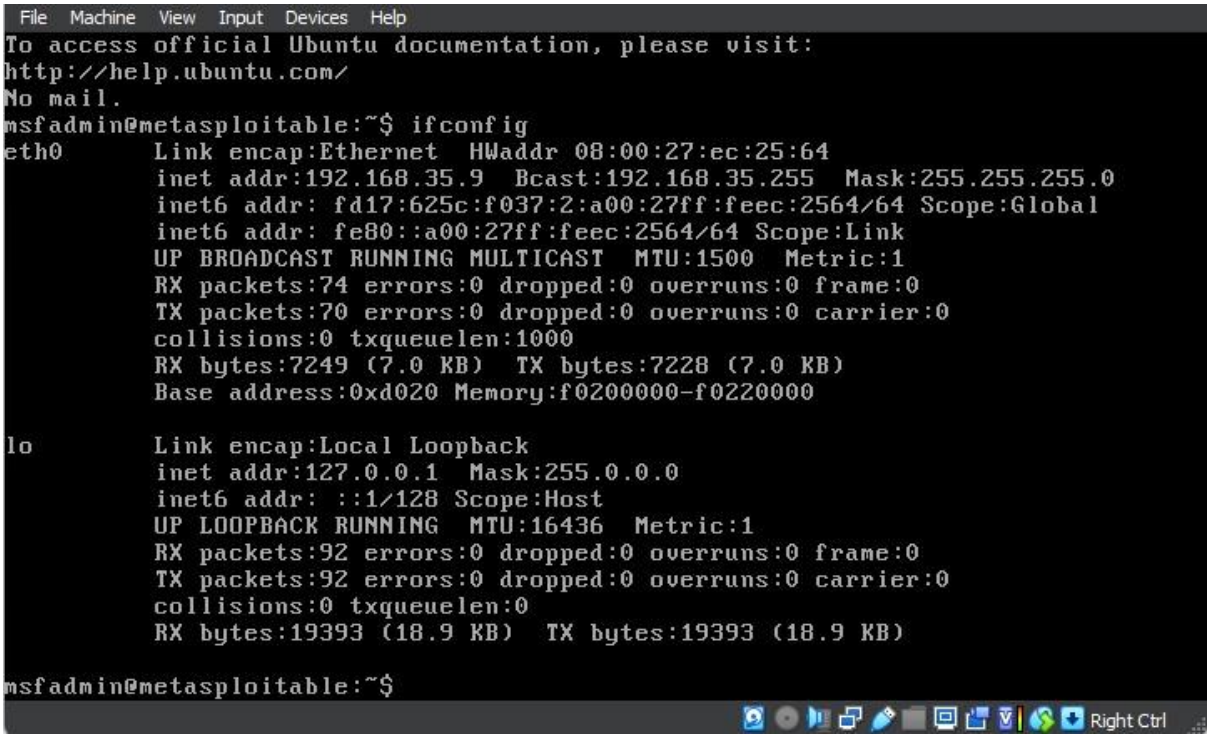
Then, we finally arrive at the homepage, where we can see all the dashboards that are present and what all do they indicate.



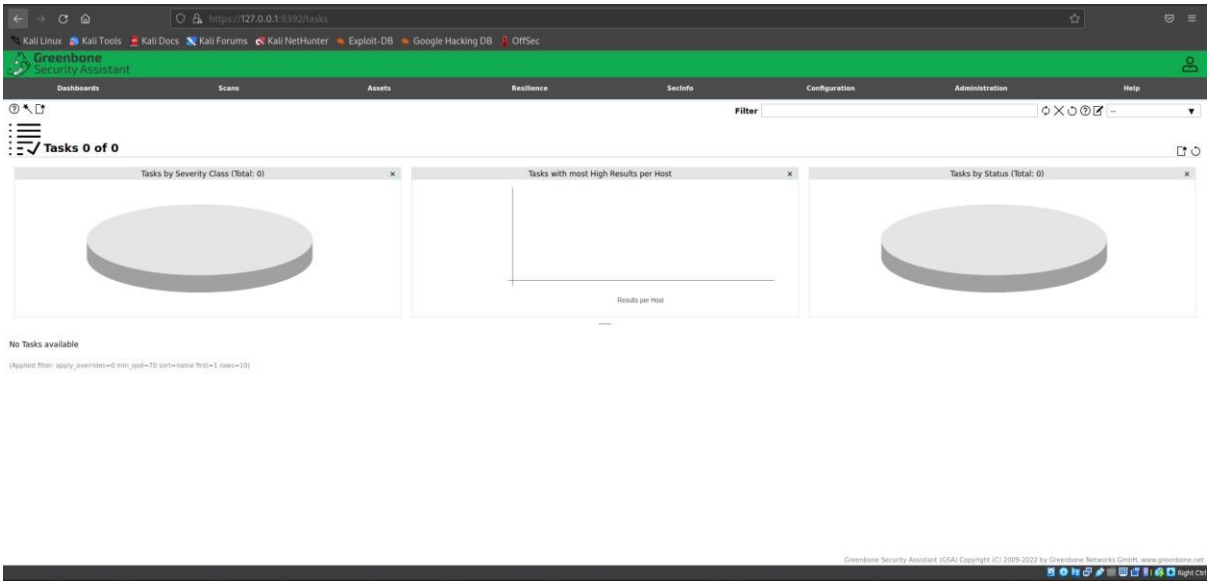
Step 4: Scanning for vulnerabilities

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

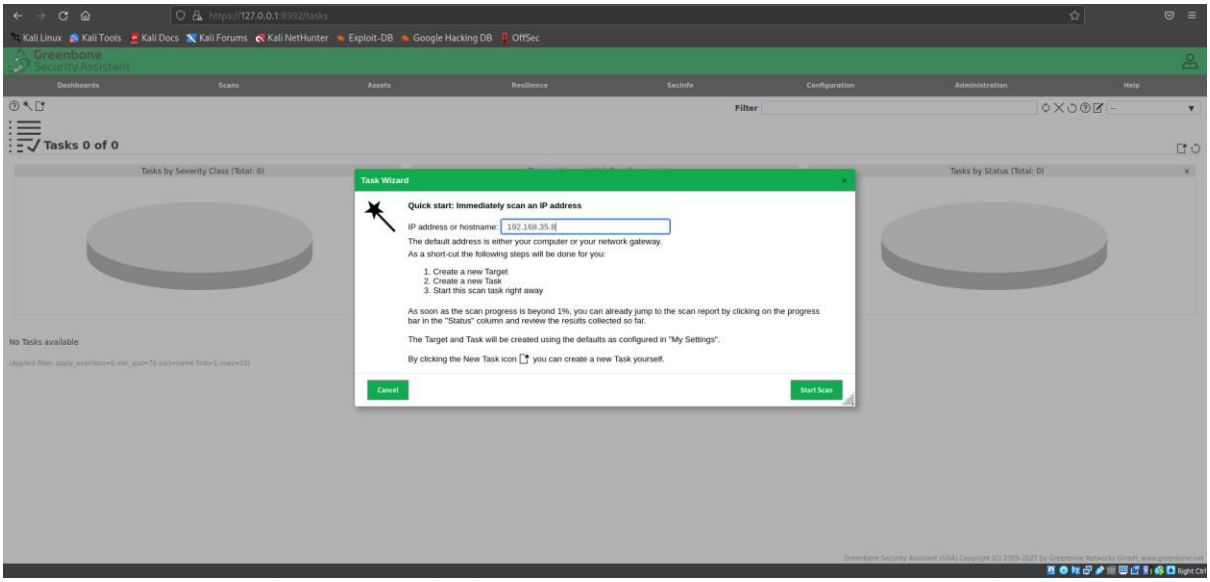
In our case, we will run the **-ifconfig** command in our metasploitable2 to get its IP address.



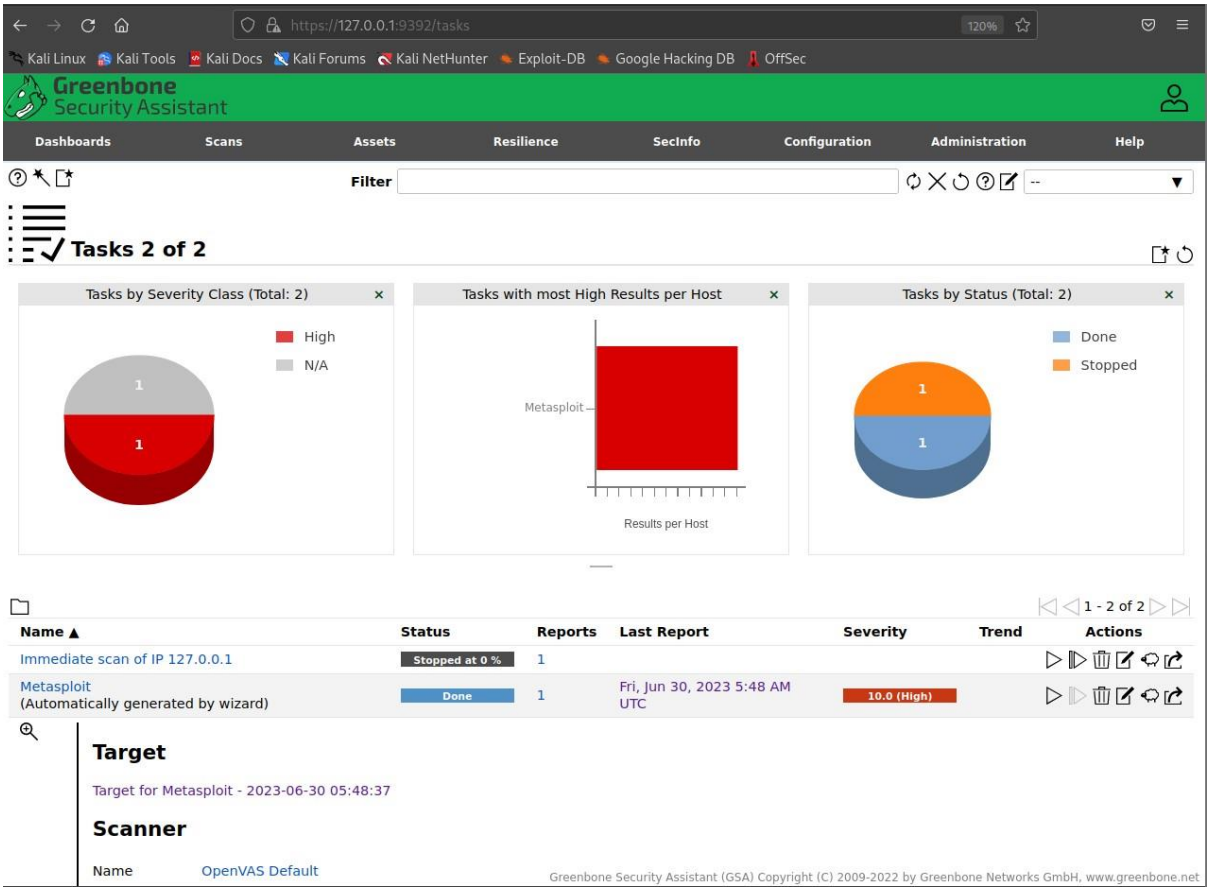
Then, in the web portal, under scans, we will find tasks. Click on that.

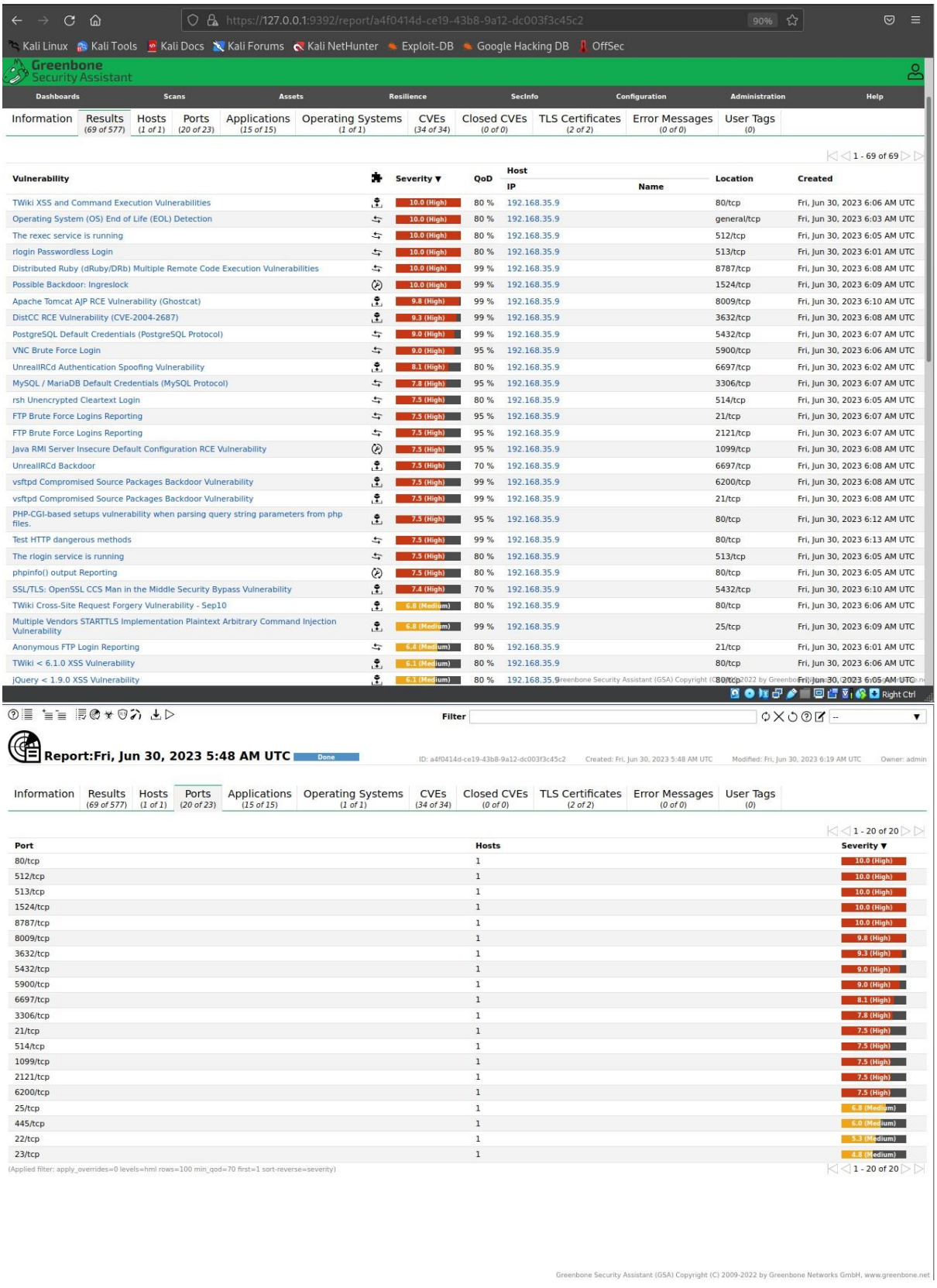


After that, we will select the task wizard. Inside it, we will put the IP address of the metasploitable.

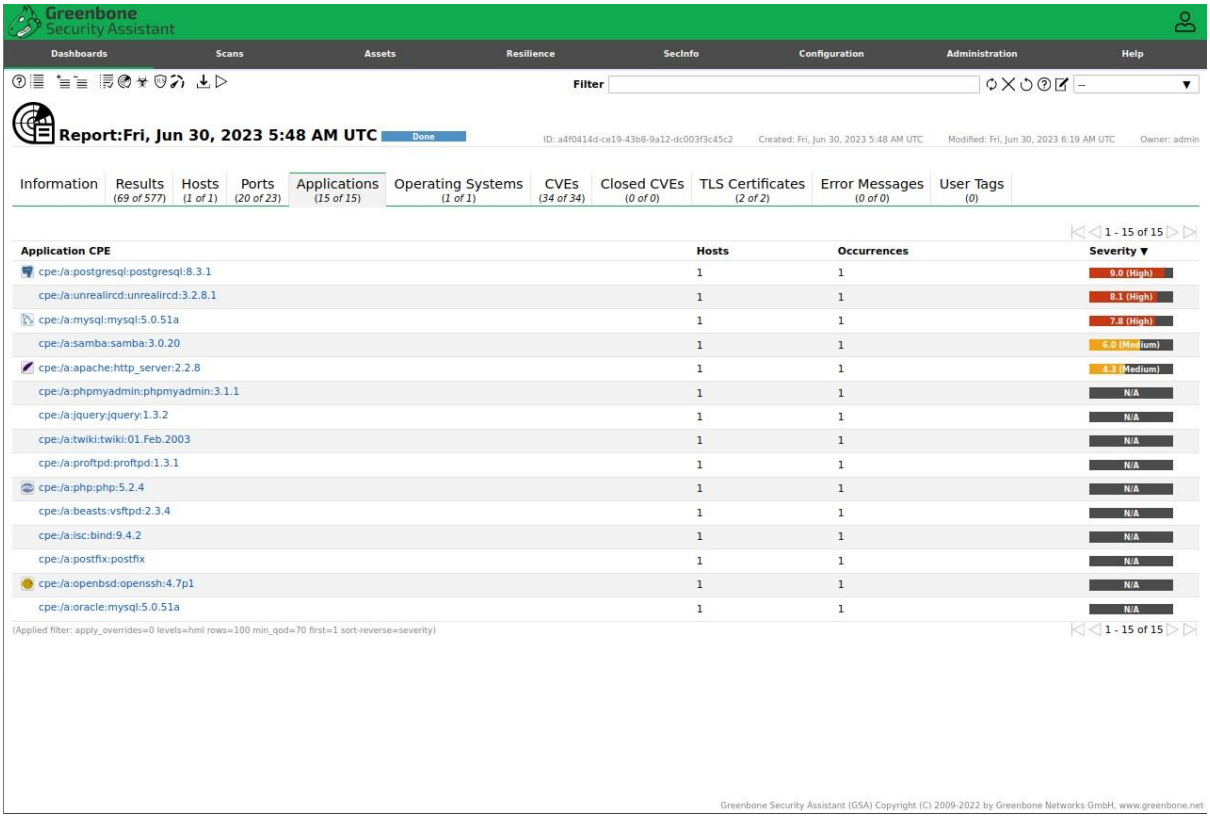


Now, we see that after completing the scans, we get the results on our metasploitable2, which is very highly vulnerable.

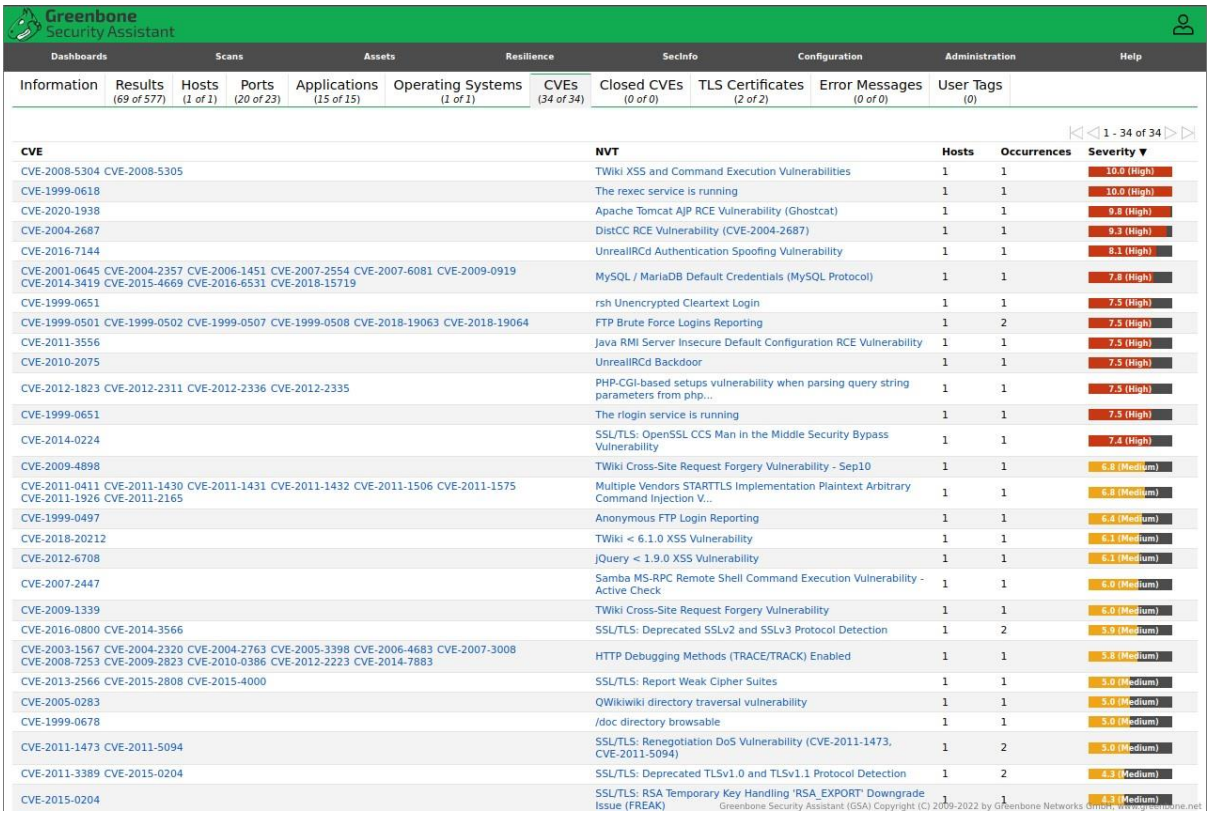



https://127.0.0.1:9392/report/a4f0414d-ce19-43b8-9a12-dc003f3c45c2
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Greenbone Security Assistant
Dashboards Scans Assets Resilience SecInfo Configuration Administration Help
Information Results (69 of 577) Hosts (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 - 69 of 69
Vulnerability Severity ▼ QoD Host IP Name Location Created
TWiki XSS and Command Execution Vulnerabilities 10.0 (High) 80 % 192.168.35.9 80/tcp Fri, Jun 30, 2023 6:06 AM UTC
Operating System (OS) End of Life (EOL) Detection 10.0 (High) 80 % 192.168.35.9 general/tcp Fri, Jun 30, 2023 6:03 AM UTC
The rexec service is running 10.0 (High) 80 % 192.168.35.9 512/tcp Fri, Jun 30, 2023 6:05 AM UTC
rlogin Passwordless Login 10.0 (High) 80 % 192.168.35.9 513/tcp Fri, Jun 30, 2023 6:01 AM UTC
Distributed Ruby (dRuby/DRb) Multiple Remote Code Execution Vulnerabilities 10.0 (High) 99 % 192.168.35.9 8787/tcp Fri, Jun 30, 2023 6:08 AM UTC
Possible Backdoor: Ingreslock 10.0 (High) 99 % 192.168.35.9 1524/tcp Fri, Jun 30, 2023 6:09 AM UTC
Apache Tomcat AJP RCE Vulnerability (Ghostcat) 9.8 (High) 99 % 192.168.35.9 8009/tcp Fri, Jun 30, 2023 6:10 AM UTC
DistCC RCE Vulnerability (CVE-2004-2687) 9.3 (High) 99 % 192.168.35.9 3632/tcp Fri, Jun 30, 2023 6:08 AM UTC
PostgreSQL Default Credentials (PostgreSQL Protocol) 9.0 (High) 99 % 192.168.35.9 5432/tcp Fri, Jun 30, 2023 6:07 AM UTC
VNC Brute Force Login 9.0 (High) 95 % 192.168.35.9 5900/tcp Fri, Jun 30, 2023 6:06 AM UTC
UnrealIRCd Authentication Spoofing Vulnerability 8.1 (High) 80 % 192.168.35.9 6697/tcp Fri, Jun 30, 2023 6:02 AM UTC
MySQL / MariaDB Default Credentials (MySQL Protocol) 7.8 (High) 95 % 192.168.35.9 3306/tcp Fri, Jun 30, 2023 6:07 AM UTC
rsh Unencrypted Cleartext Login 7.5 (High) 80 % 192.168.35.9 514/tcp Fri, Jun 30, 2023 6:05 AM UTC
FTP Brute Force Logins Reporting 7.5 (High) 95 % 192.168.35.9 21/tcp Fri, Jun 30, 2023 6:07 AM UTC
FTP Brute Force Logins Reporting 7.5 (High) 95 % 192.168.35.9 2121/tcp Fri, Jun 30, 2023 6:07 AM UTC
Java RMI Server Insecure Default Configuration RCE Vulnerability 7.5 (High) 95 % 192.168.35.9 1099/tcp Fri, Jun 30, 2023 6:08 AM UTC
UnrealIRCd Backdoor 7.5 (High) 70 % 192.168.35.9 6697/tcp Fri, Jun 30, 2023 6:08 AM UTC
vsftpd Compromised Source Packages Backdoor Vulnerability 7.5 (High) 99 % 192.168.35.9 6200/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
PHP-CGI-based setups vulnerability when parsing query string parameters from php files. 7.5 (High) 95 % 192.168.35.9 80/tcp Fri, Jun 30, 2023 6:12 AM UTC
Test HTTP dangerous methods 7.5 (High) 99 % 192.168.35.9 80/tcp Fri, Jun 30, 2023 6:13 AM UTC
The rlogin service is running 7.5 (High) 80 % 192.168.35.9 513/tcp Fri, Jun 30, 2023 6:05 AM UTC
phpinfo() output Reporting 7.5 (High) 80 % 192.168.35.9 80/tcp Fri, Jun 30, 2023 6:05 AM UTC
SSL/TLS: OpenSSL CCS Man in the Middle Security Bypass Vulnerability 7.4 (High) 70 % 192.168.35.9 5432/tcp Fri, Jun 30, 2023 6:10 AM UTC
TWiki Cross-Site Request Forgery Vulnerability - Sep10 6.8 (Medium) 80 % 192.168.35.9 80/tcp Fri, Jun 30, 2023 6:06 AM UTC
Multiple Vendors STARTTLS Implementation Plaintext Arbitrary Command Injection Vulnerability 6.8 (Medium) 99 % 192.168.35.9 25/tcp Fri, Jun 30, 2023 6:09 AM UTC
Anonymous FTP Login Reporting 6.4 (Medium) 80 % 192.168.35.9 21/tcp Fri, Jun 30, 2023 6:01 AM UTC
TWiki < 6.1.0 XSS Vulnerability 6.1 (Medium) 80 % 192.168.35.9 80/tcp Fri, Jun 30, 2023 6:06 AM UTC
jQuery < 1.9.0 XSS Vulnerability 6.1 (Medium) 80 % 192.168.35.9 80/tcp Fri, Jun 30, 2023 6:05 AM UTC
Greenbone Security Assistant (GSA) Copyright (C) 2009-2022 by Greenbone Networks GmbH, www.greenbone.net
Right Ctrl
Filter
Report:Fri, Jun 30, 2023 5:48 AM UTC Done
ID: a4f0414d-ce19-43b8-9a12-dc003f3c45c2 Created: Fri, Jun 30, 2023 5:48 AM UTC Modified: Fri, Jun 30, 2023 6:19 AM UTC Owner: admin
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 - 20 of 20
Port Hosts Severity ▼
80/tcp 1 10.0 (High)
512/tcp 1 10.0 (High)
513/tcp 1 10.0 (High)
1524/tcp 1 10.0 (High)
8787/tcp 1 10.0 (High)
8009/tcp 1 9.8 (High)
3632/tcp 1 9.3 (High)
5432/tcp 1 9.0 (High)
5900/tcp 1 9.0 (High)
6697/tcp 1 8.1 (High)
3306/tcp 1 7.8 (High)
21/tcp 1 7.5 (High)
514/tcp 1 7.5 (High)
1099/tcp 1 7.5 (High)
2121/tcp 1 7.5 (High)
6200/tcp 1 7.5 (High)
25/tcp 1 6.8 (Medium)
445/tcp 1 6.0 (Medium)
22/tcp 1 5.3 (Medium)
23/tcp 1 4.8 (Medium)
(Applied filter: apply_overrides=0 levels=hml rows=100 min_qod=70 first=1 sort-reverse=severity)
1 - 20 of 20
Greenbone Security Assistant (GSA) Copyright (C) 2009-2022 by Greenbone Networks GmbH, www.greenbone.net

This shows the vulnerabilities of the open ports that are present in the system, in a decreasing manner

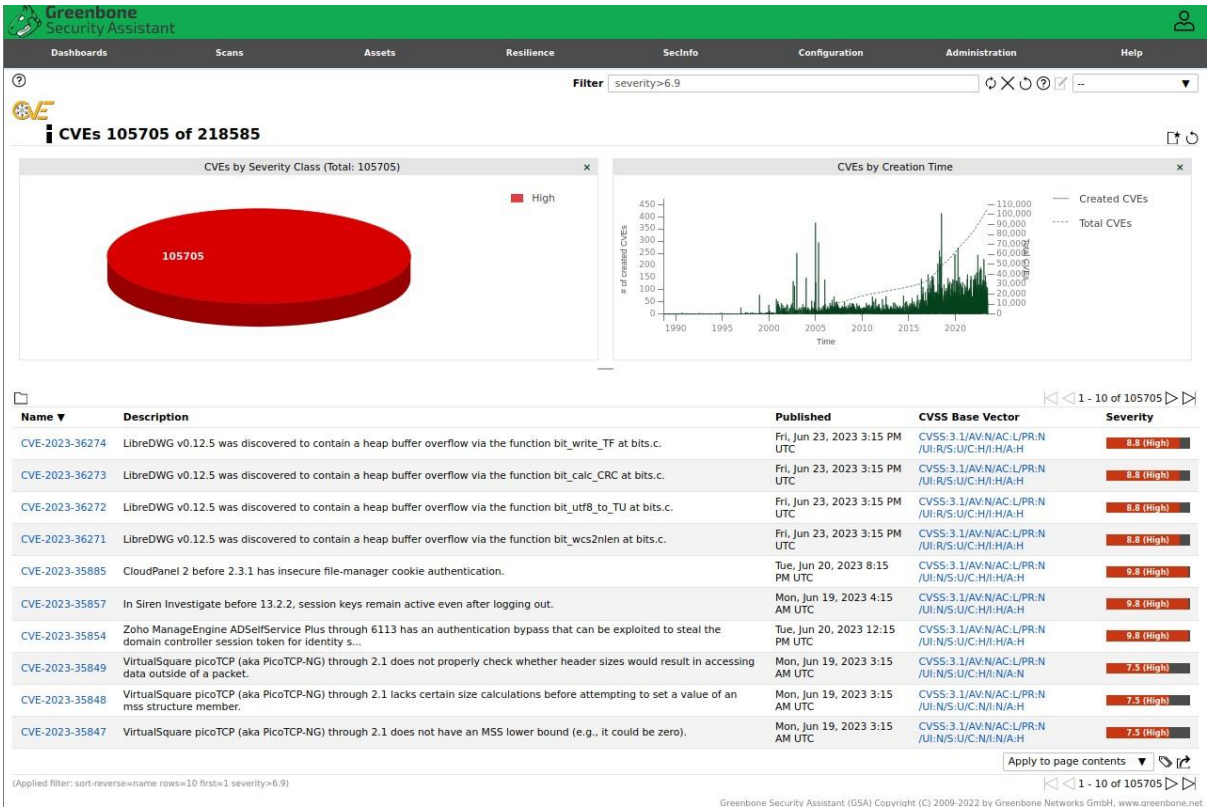


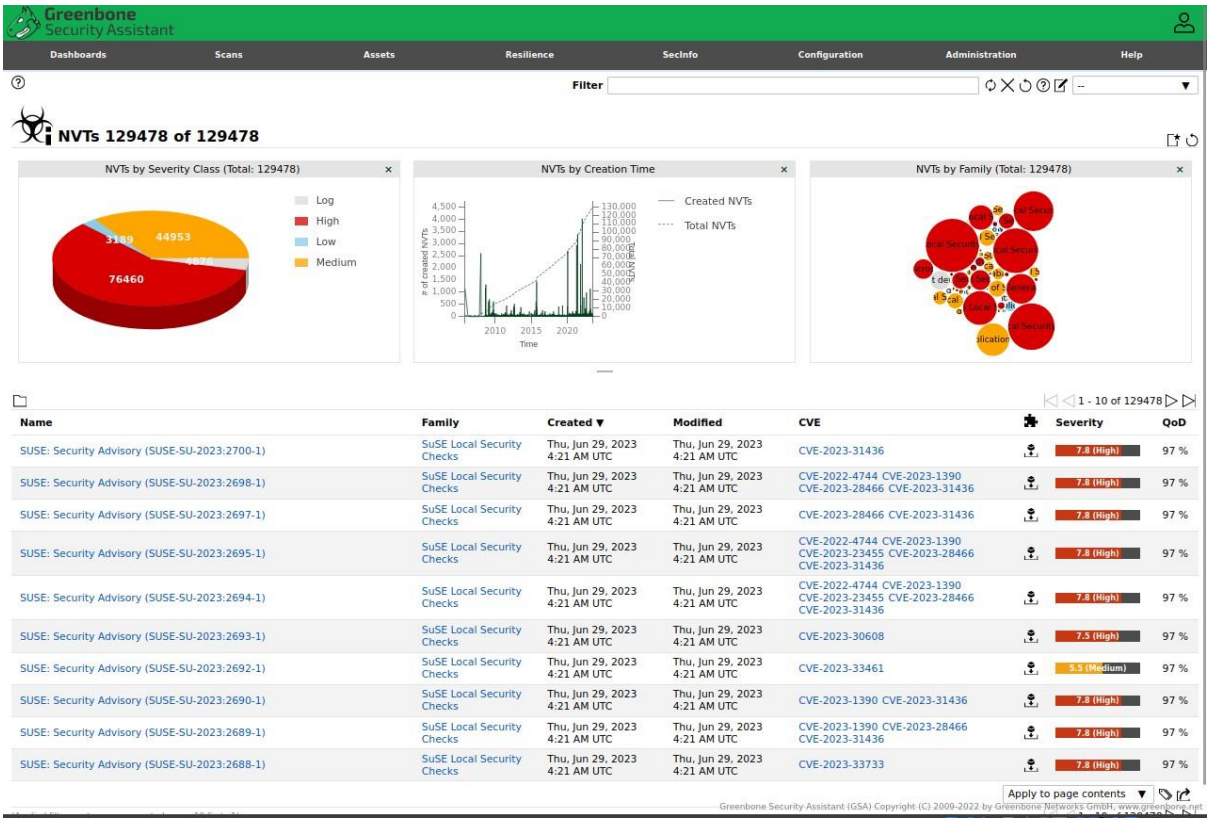
This figure indicates the vulnerabilities that can / are caused by the applications that are running in the system, as improper setup and configuration of applications can actually lead to a potential attack surface.



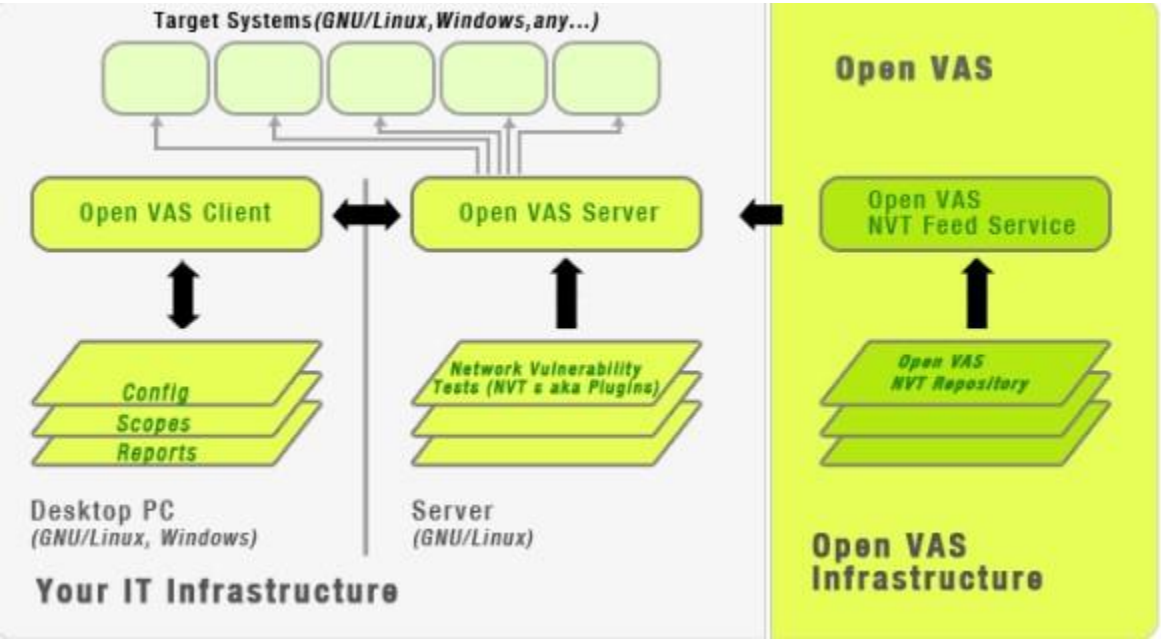
CVE is a glossary that classifies vulnerabilities. The glossary analyses vulnerabilities and then uses the Common Vulnerability Scoring System (CVSS) to evaluate the threat level of a vulnerability.

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.


Greenbone
Security Assistant
Dashboards
Scans
Assets
Resilience
SecInfo
Configuration
Administration
Help
Filter
severity>6.9
CVEs 105705 of 218585
CVEs by Severity Class (Total: 105705)
High
105705
CVEs by Creation Time
Created CVEs
Total CVEs
# of created CVEs
Time
1 - 10 of 105705
Name
Description
Published
CVSS Base Vector
Severity
CVE-2023-36274
LibreDWG v0.12.5 was discovered to contain a heap buffer overflow via the function bit_write_TF at bits.c.
Fri, Jun 23, 2023 3:15 PM UTC
CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H
8.8 (High)
CVE-2023-36273
LibreDWG v0.12.5 was discovered to contain a heap buffer overflow via the function bit_calc_CRC at bits.c.
Fri, Jun 23, 2023 3:15 PM UTC
CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H
8.8 (High)
CVE-2023-36272
LibreDWG v0.12.5 was discovered to contain a heap buffer overflow via the function bit_utf8_to_TU at bits.c.
Fri, Jun 23, 2023 3:15 PM UTC
CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H
8.8 (High)
CVE-2023-36271
LibreDWG v0.12.5 was discovered to contain a heap buffer overflow via the function bit_wcs2nlen at bits.c.
Fri, Jun 23, 2023 3:15 PM UTC
CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H
8.8 (High)
CVE-2023-35885
CloudPanel 2 before 2.3.1 has insecure file-manager cookie authentication.
Tue, Jun 20, 2023 8:15 PM UTC
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
9.8 (High)
CVE-2023-35857
In Siren Investigate before 13.2.2, session keys remain active even after logging out.
Mon, Jun 19, 2023 4:15 AM UTC
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
9.8 (High)
CVE-2023-35854
Zoho ManageEngine ADSelfService Plus through 6113 has an authentication bypass that can be exploited to steal the domain controller session token for identity s...
Tue, Jun 20, 2023 12:15 PM UTC
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
9.8 (High)
CVE-2023-35849
VirtualSquare picoTCP (aka PicoTCP-NG) through 2.1 does not properly check whether header sizes would result in accessing data outside of a packet.
Mon, Jun 19, 2023 3:15 AM UTC
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N
7.5 (High)
CVE-2023-35848
VirtualSquare picoTCP (aka PicoTCP-NG) through 2.1 lacks certain size calculations before attempting to set a value of an mss structure member.
Mon, Jun 19, 2023 3:15 AM UTC
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H
7.5 (High)
CVE-2023-35847
VirtualSquare picoTCP (aka PicoTCP-NG) through 2.1 does not have an MSS lower bound (e.g., it could be zero).
Mon, Jun 19, 2023 3:15 AM UTC
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H
7.5 (High)
Apply to page contents
(Applied filter: sort-reverse=name rows=10 first=1 severity>6.9)
1 - 10 of 105705
Greenbone Security Assistant (GSA) Copyright (C) 2009-2022 by Greenbone Networks GmbH, www.greenbone.net


Greenbone
Security Assistant
Dashboards
Scans
Assets
Resilience
SecInfo
Configuration
Administration
Help
Filter
NVTs 129478 of 129478
NVTs by Severity Class (Total: 129478)
Log
High
Low
Medium
3189
44953
76460
NVTs by Creation Time
Created NVTs
Total NVTs
# of created NVTs
Time
NVTs by Family (Total: 129478)
1 - 10 of 129478
Name
Family
Created
Modified
CVE
Severity
QoD
SUSE: Security Advisory (SUSE-SU-2023:2700-1)
SuSE Local Security Checks
Thu, Jun 29, 2023 4:21 AM UTC
Thu, Jun 29, 2023 4:21 AM UTC
CVE-2023-31436
7.8 (High)
97 %
SUSE: Security Advisory (SUSE-SU-2023:2698-1)
SuSE Local Security Checks
Thu, Jun 29, 2023 4:21 AM UTC
Thu, Jun 29, 2023 4:21 AM UTC
CVE-2022-4744 CVE-2023-1390 CVE-2023-28466 CVE-2023-31436
7.8 (High)
97 %
SUSE: Security Advisory (SUSE-SU-2023:2697-1)
SuSE Local Security Checks
Thu, Jun 29, 2023 4:21 AM UTC
Thu, Jun 29, 2023 4:21 AM UTC
CVE-2023-28466 CVE-2023-31436
7.8 (High)
97 %
SUSE: Security Advisory (SUSE-SU-2023:2695-1)
SuSE Local Security Checks
Thu, Jun 29, 2023 4:21 AM UTC
Thu, Jun 29, 2023 4:21 AM UTC
CVE-2022-4744 CVE-2023-1390 CVE-2023-23455 CVE-2023-28466 CVE-2023-31436
7.8 (High)
97 %
SUSE: Security Advisory (SUSE-SU-2023:2694-1)
SuSE Local Security Checks
Thu, Jun 29, 2023 4:21 AM UTC
Thu, Jun 29, 2023 4:21 AM UTC
CVE-2022-4744 CVE-2023-1390 CVE-2023-23455 CVE-2023-28466 CVE-2023-31436
7.8 (High)
97 %
SUSE: Security Advisory (SUSE-SU-2023:2693-1)
SuSE Local Security Checks
Thu, Jun 29, 2023 4:21 AM UTC
Thu, Jun 29, 2023 4:21 AM UTC
CVE-2023-30608
7.5 (High)
97 %
SUSE: Security Advisory (SUSE-SU-2023:2692-1)
SuSE Local Security Checks
Thu, Jun 29, 2023 4:21 AM UTC
Thu, Jun 29, 2023 4:21 AM UTC
CVE-2023-33461
5.5 (Medium)
97 %
SUSE: Security Advisory (SUSE-SU-2023:2690-1)
SuSE Local Security Checks
Thu, Jun 29, 2023 4:21 AM UTC
Thu, Jun 29, 2023 4:21 AM UTC
CVE-2023-1390 CVE-2023-31436
7.8 (High)
97 %
SUSE: Security Advisory (SUSE-SU-2023:2689-1)
SuSE Local Security Checks
Thu, Jun 29, 2023 4:21 AM UTC
Thu, Jun 29, 2023 4:21 AM UTC
CVE-2023-1390 CVE-2023-28466 CVE-2023-31436
7.8 (High)
97 %
SUSE: Security Advisory (SUSE-SU-2023:2688-1)
SuSE Local Security Checks
Thu, Jun 29, 2023 4:21 AM UTC
Thu, Jun 29, 2023 4:21 AM UTC
CVE-2023-33733
7.8 (High)
97 %
Apply to page contents
Greenbone Security Assistant (GSA) Copyright (C) 2009-2022 by Greenbone Networks GmbH, www.greenbone.net

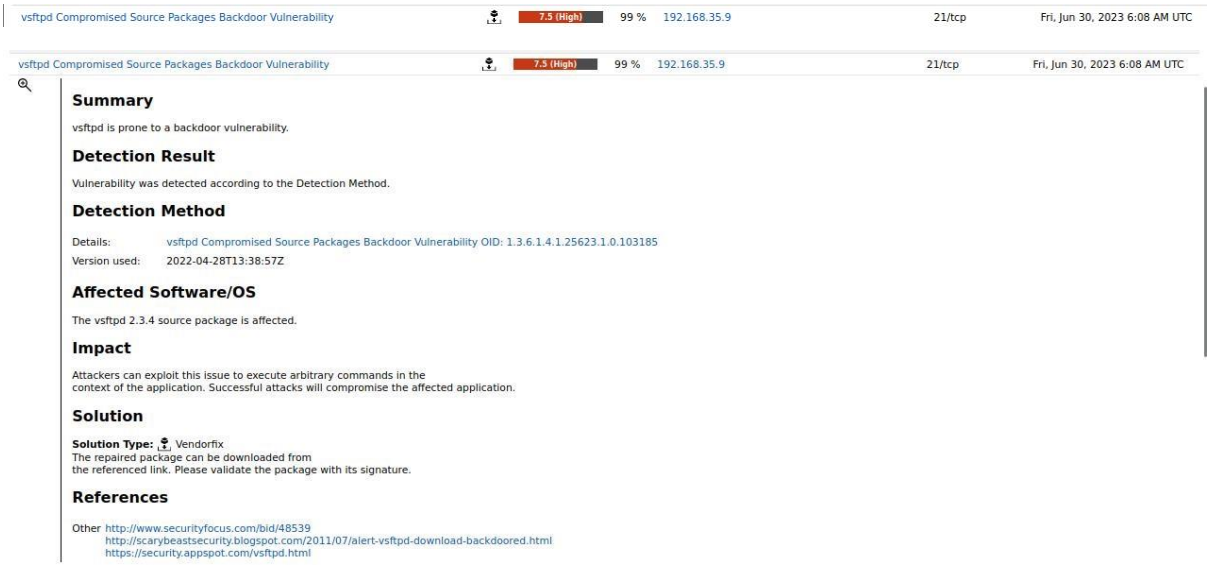
FLOWCHART



RESULT

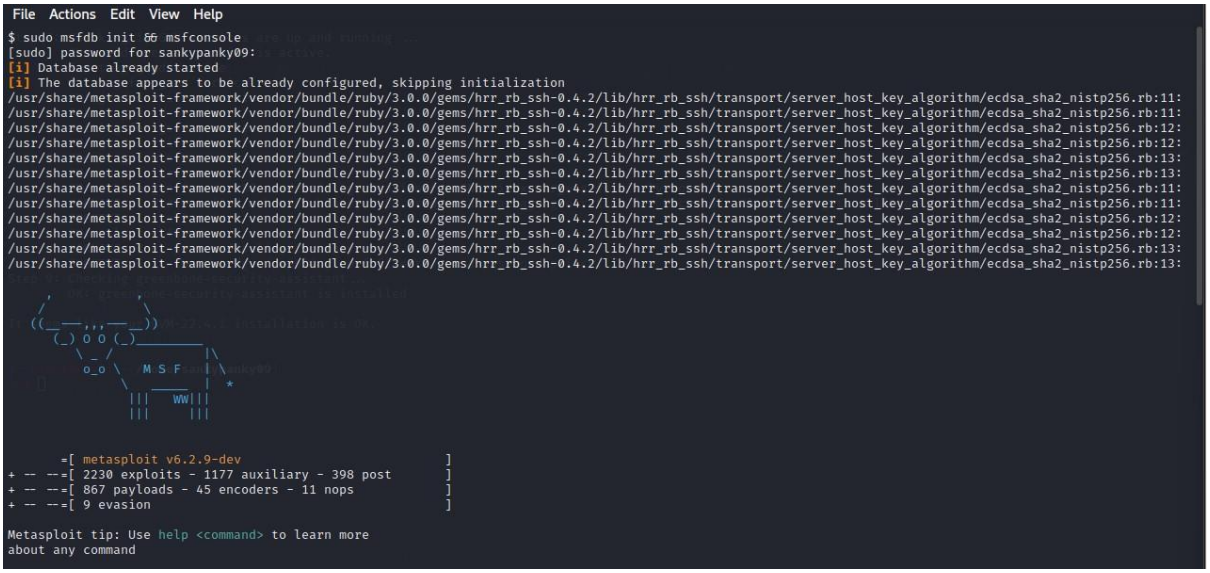
We were able to exploit a known vulnerability called vsftpd vulnerability successfully.

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

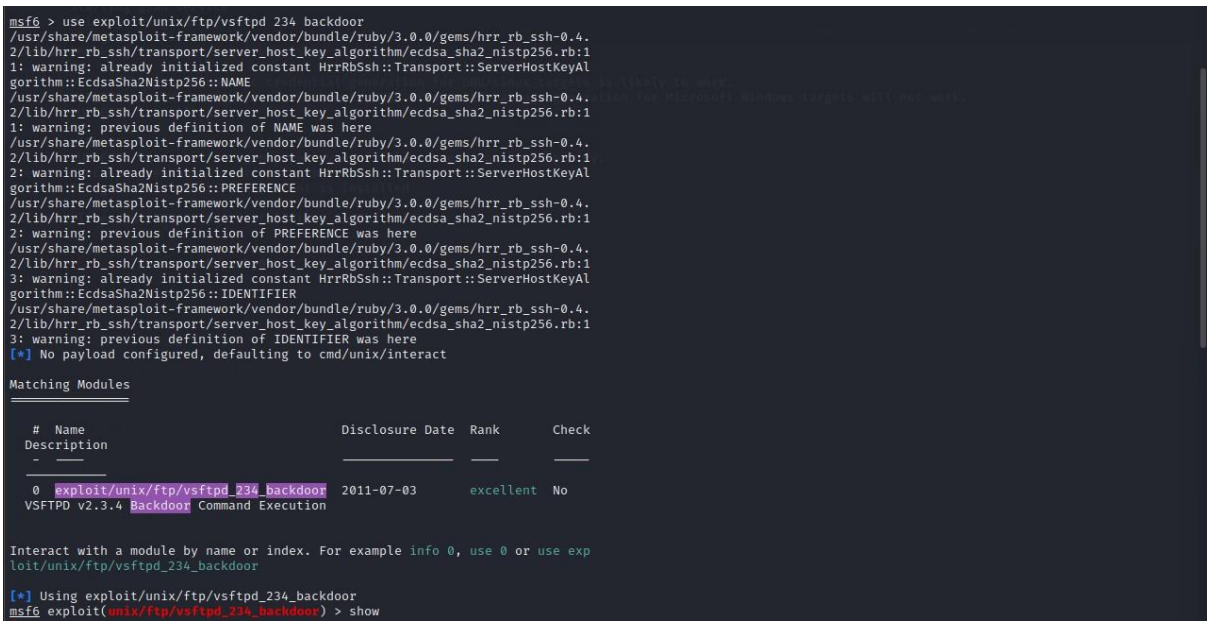


We will use a pen testing tool called Metaploit framework in order to gain a backdoor access to our target system

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



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



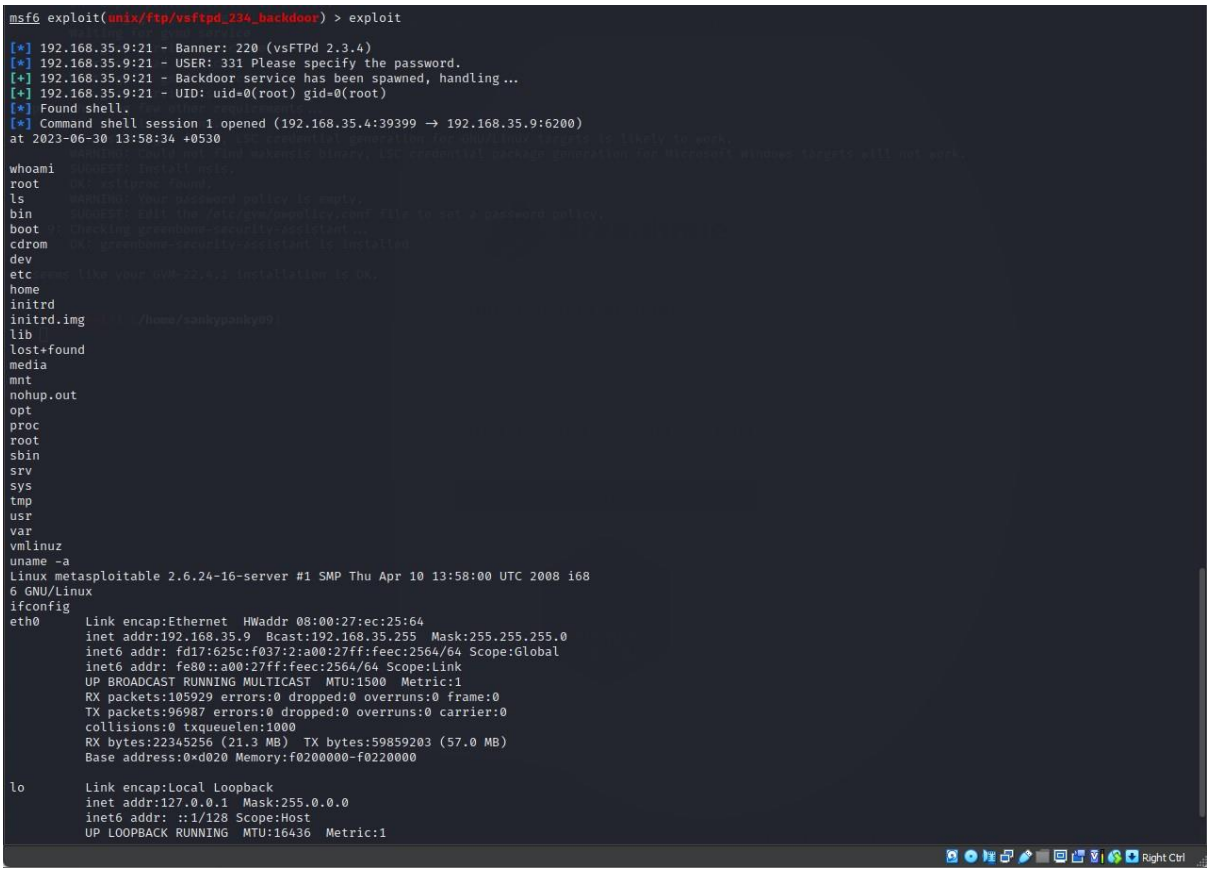
Then, we use the command show option in order to get the settings

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

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://githu
                                      b.com/rapid7/metasploit-framework/wik
                                      i/Using-Metasploit
   RPORT   21               yes       The target port (TCP)


Payload options (cmd/unix/interact):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


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://githu
                                      b.com/rapid7/metasploit-framework/wik
                                      i/Using-Metasploit
   RPORT   21               yes       The target port (TCP)


Payload options (cmd/unix/interact):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


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



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. T**echnical 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.

.