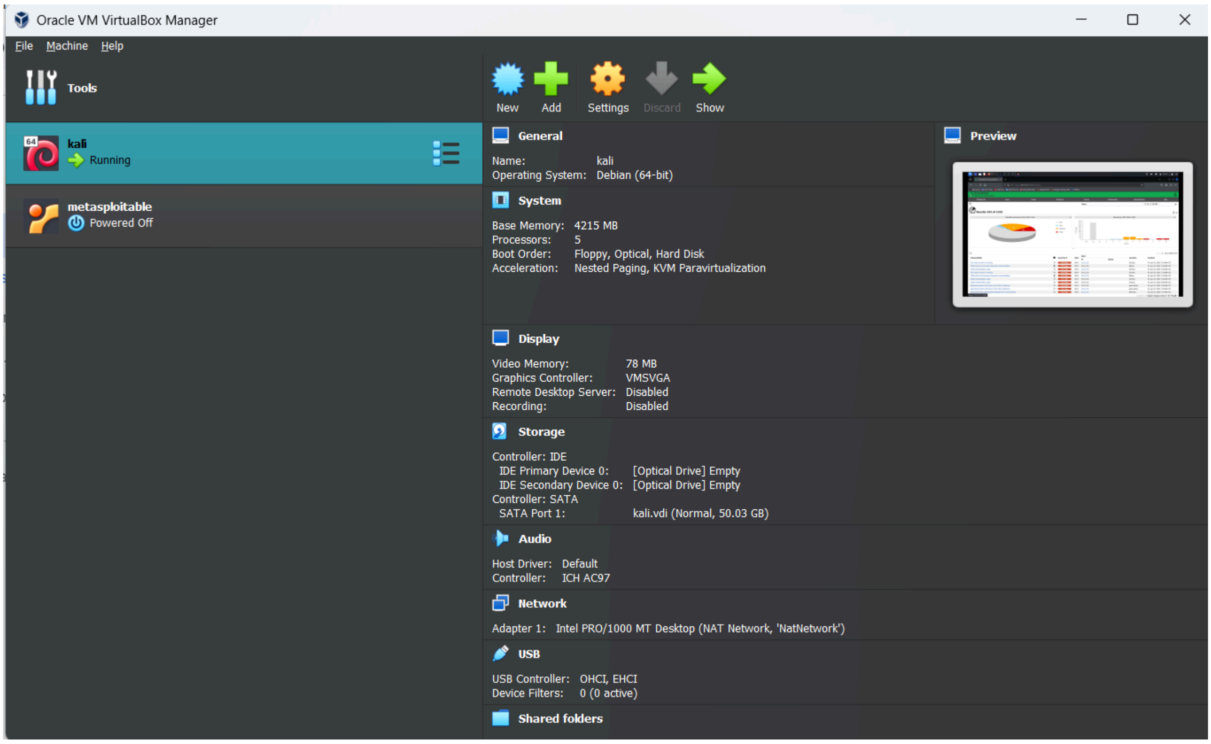
Network Vulnerability Assessment Report

Name: Sahej Hira (cybersecurity intern @Extion)
Date of Scan: January 24, 2025
Target IP Address: 10.0.2.16 (metasploitable2)
Scan Tool Used: OpenVAS
Scan Result: High and Critical Vulnerabilities Identified
Virtualization tool Used: Oracle VM VirtualBox Manager



General

Name: kali
Operating System: Debian (64-bit)

System

Base Memory: 4215 MB
Processors: 5
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Nested Paging, KVM Paravirtualization

Display

Video Memory: 78 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Primary Device 0: [Optical Drive] Empty
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 1: kali.vdi (Normal, 50.03 GB)

Audio

Host Driver: Default
Controller: ICH AC97

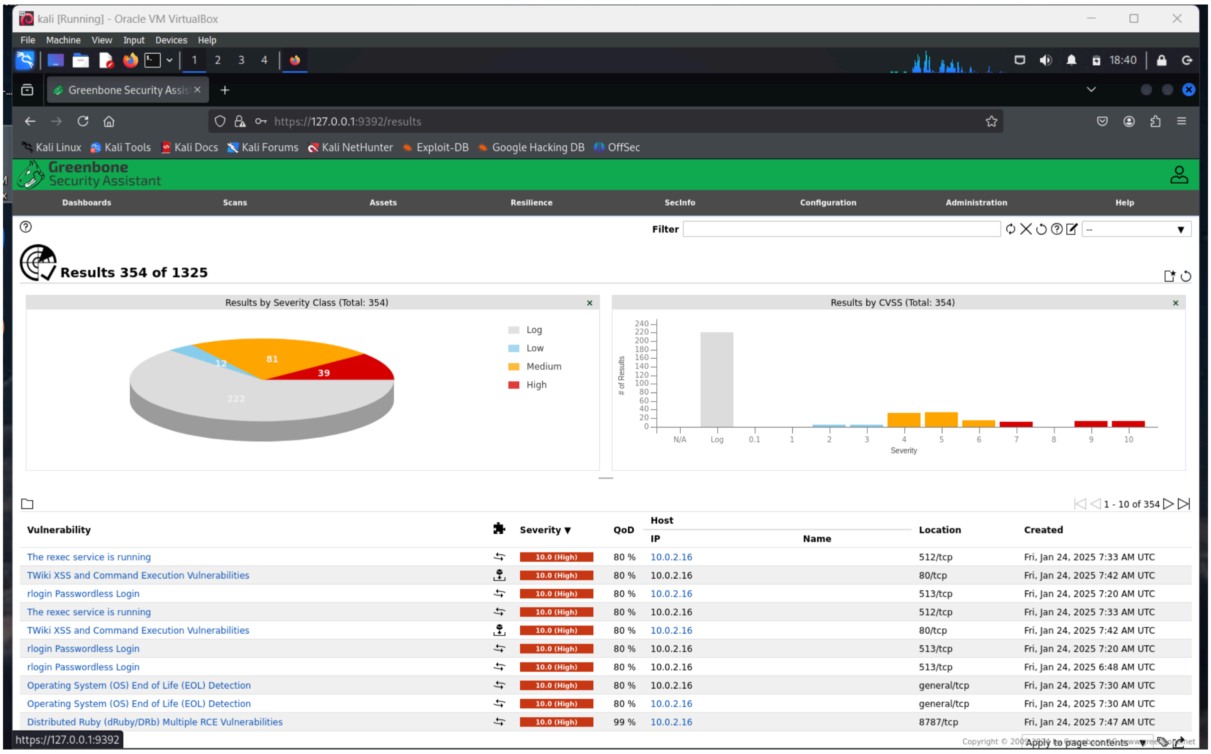
Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT Network, 'NatNetwork')

USB

USB Controller: OHCI, EHCI
Device Filters: 0 (0 active)

Shared folders



Results 354 of 1325

Vulnerability		Severity ▼	QoD	Host IP	Host Name	Location	Created
The rexec service is running	⇆	10.0 (High)	80 %	10.0.2.16		512/tcp	Fri, Jan 24, 2025 7:33 AM UTC
TWiki XSS and Command Execution Vulnerabilities		10.0 (High)	80 %	10.0.2.16		80/tcp	Fri, Jan 24, 2025 7:42 AM UTC
rlogin Passwordless Login	⇆	10.0 (High)	80 %	10.0.2.16		513/tcp	Fri, Jan 24, 2025 7:20 AM UTC
The rexec service is running	⇆	10.0 (High)	80 %	10.0.2.16		512/tcp	Fri, Jan 24, 2025 7:33 AM UTC
TWiki XSS and Command Execution Vulnerabilities		10.0 (High)	80 %	10.0.2.16		80/tcp	Fri, Jan 24, 2025 7:42 AM UTC
rlogin Passwordless Login	⇆	10.0 (High)	80 %	10.0.2.16		513/tcp	Fri, Jan 24, 2025 7:20 AM UTC
rlogin Passwordless Login	⇆	10.0 (High)	80 %	10.0.2.16		513/tcp	Fri, Jan 24, 2025 6:48 AM UTC
Operating System (OS) End of Life (EOL) Detection	⇆	10.0 (High)	80 %	10.0.2.16		general/tcp	Fri, Jan 24, 2025 7:30 AM UTC
Operating System (OS) End of Life (EOL) Detection	⇆	10.0 (High)	80 %	10.0.2.16		general/tcp	Fri, Jan 24, 2025 7:30 AM UTC
Distributed Ruby (dRuby/DRb) Multiple RCE Vulnerabilities	⇆	10.0 (High)	99 %	10.0.2.16		8787/tcp	Fri, Jan 24, 2025 7:47 AM UTC

---

1. Possible Backdoor: Ingreslock

Description:

The **Ingreslock** service, associated with port **1524/tcp**, is known to be a potential backdoor, often used by attackers to gain unauthorized access to the system. Once active, this service allows malicious users to remotely execute arbitrary commands, compromising the security and integrity of the affected system. It is commonly used for **persistent access** by attackers after an initial compromise.

- **Severity Rating:** Critical
- **Exploitability Probability:** 99%
- **Impact:** Allows unauthorized users to execute commands on the system, leading to full system compromise, data theft, or destruction.

Mitigation Plan:

1. **Immediate Action:** Block port **1524/tcp** at the firewall to prevent unauthorized access. You can use firewall rules to block the incoming traffic on this port.
2. **Scan for Malware:** Conduct a thorough scan of the system to identify any malware or backdoors installed by an attacker. Use security tools like **ClamAV** or **Rootkit Hunter** to scan for known malicious programs.
3. **Remove the Ingreslock Service:** If Ingreslock is not required for any legitimate operations, disable and remove it from the system. This can be done by removing it from the configuration files or uninstalling the package.
4. **Audit Logs and Monitor Access:** Review system logs, including **/var/log/auth.log** (authentication logs) and **/var/log/syslog** (general system logs), to detect any signs of exploitation. Set up **log monitoring** to track abnormal activities or unauthorized access attempts.

---

2. Distributed Ruby (dRuby/DRb) Multiple RCE Vulnerabilities

Description:

The **Distributed Ruby (dRuby)** protocol is a **Ruby-based** service that facilitates remote procedure calls (RPC) between different Ruby processes. However, multiple **remote code execution (RCE)** vulnerabilities have been identified in older versions of dRuby. These vulnerabilities can allow remote attackers to execute arbitrary code on the target system, posing a significant risk.

- **Severity Rating:** Critical
- **Exploitability Probability:** 99%
- **Impact:** Attackers can execute arbitrary code, leading to unauthorized access to the system and potential data loss or theft.

Mitigation Plan:

1. **Upgrade dRuby:** Ensure that the **dRuby** version is updated to the latest stable version that includes security patches for these RCE vulnerabilities. Update it using the appropriate package manager for Ruby or using the gem command.
2. **Restrict Access to dRuby:** Limit the access to the dRuby service by implementing firewall rules to allow only trusted IP addresses to connect to the dRuby service. This can help prevent unauthorized access to the vulnerable service.
3. **Remove Service if Unnecessary:** If dRuby is not required for any essential system functionality, disable or remove it completely. You can uninstall the dRuby gem using the gem command or simply remove the relevant service if it's not being used.
4. **Monitor the System:** Continuously monitor for any unusual activity or connections that may indicate exploitation. Use tools like **Fail2ban** or **OSSEC** for real-time intrusion detection.

---

3. rlogin Passwordless Login

Description:

The **rlogin** service is a remote login service that allows users to log in without requiring authentication. This is a critical vulnerability because it exposes systems to unauthorized access, especially if the system is accessible via untrusted networks. Attackers can exploit this to gain remote control of the system without any need for credentials.

- **Severity Rating:** Critical
- **Exploitability Probability:** 80%
- **Impact:** Remote attackers can gain access to the system without credentials, potentially leading to full system compromise.

Mitigation Plan:

1. **Disable rlogin Service:** Disable the **rlogin** service immediately if it is not needed. To disable it, edit the `/etc/inetd.conf` or `/etc/xinetd.d/` configuration files to comment out or remove the rlogin entry.
2. **Replace with SSH:** Use **SSH (Secure Shell)** as a more secure alternative for remote logins. SSH provides strong encryption and authentication, making it far more secure than rlogin. Install and configure SSH using the system's package manager.
3. **Restrict Access:** If **rlogin** must remain active temporarily, restrict access to trusted IPs only by configuring firewall rules to block all other connections.
4. **Audit for Unauthorized Access:** Regularly review the **/var/log/auth.log** to identify any unauthorized login attempts or suspicious activities that may indicate an ongoing attack.

---

4. Apache Tomcat AJP RCE Vulnerability (Ghostcat)

Description:

The **Ghostcat** vulnerability (CVE-2020-1938) affects the **Apache Tomcat** AJP (Apache JServ Protocol) connector. It allows attackers to read sensitive files and execute arbitrary code, potentially compromising the system. This vulnerability is particularly dangerous when exposed to untrusted networks.

- **Severity Rating:** High
- **Exploitability Probability:** 99%
- **Impact:** Attackers can retrieve sensitive data or execute arbitrary commands, leading to full system compromise.

Mitigation Plan:

1. **Update Apache Tomcat:** Ensure the system is running the latest patched version of Apache Tomcat. The latest version addresses the Ghostcat vulnerability.
2. **Disable AJP Connector:** If the AJP connector is not required, **disable it** by commenting out the related connector configuration in the **server.xml** file.
3. **Restrict Access to AJP:** Use firewall rules to restrict AJP access to trusted sources only. Limit the IP addresses that can connect to AJP by applying network filtering rules.
4. **Perform Regular Audits:** Regularly review the Apache Tomcat logs for suspicious activities or attempts to exploit the AJP connector.

---

5. MySQL / MariaDB Default Credentials

Description:

The **MySQL/MariaDB** database is configured with **default credentials** (e.g., 'root' with no password or weak passwords), which makes it vulnerable to unauthorized access. This allows attackers to easily access and modify the database without authentication, leading to potential data loss or corruption.

- **Severity Rating:** High
- **Exploitability Probability:** 95%
- **Impact:** Unauthorized users can access, modify, or delete data, resulting in sensitive data leakage or corruption.

Mitigation Plan:

1. **Change Default Credentials:** Immediately change the default credentials for MySQL/MariaDB. This can be done by logging into the MySQL database and altering the root user’s password.
2. **Remove Unused Accounts:** Disable or remove any unused default accounts that are often created by MySQL/MariaDB installations, such as `root@'%'` or `anonymous` users.

3. **Enforce Strong Password Policies:** Use complex passwords and enforce password policies for all MySQL/MariaDB accounts. Implement **password expiration** and **account lockout policies** to further secure access.
4. **Restrict Database Access:** Configure firewall rules to allow database access only from trusted IPs or local access. This will minimize the risk of unauthorized access from untrusted networks.
5. **Monitor Database Logs:** Set up **real-time monitoring** of MySQL/MariaDB logs to detect any suspicious login attempts or unauthorized access. This will help in identifying potential brute force or exploitation attempts early.

Conclusion

The scan revealed **1325 vulnerabilities** after the vulnerability analysis using Openvas in Kali Linux. The top five most critical and high vulnerabilities in the system, with some having a high probability of exploitation are described in the report above with their mitigation plans. To secure the network from intrusions and attacks of unauthorized parties, it is crucial to implement mitigation plans and monitor the network regularly to ensure security.