**Comprehensive Security Assessment of Metasploitable 2**

**1. Overview of the project objectives and methodology:**

The objective of this project was to conduct a comprehensive security assessment of Metasploitable 2, a deliberately vulnerable virtual machine, to identify potential vulnerabilities and recommend remediation actions. The assessment was performed using a combination of manual techniques and automated tools. The methodology involved the following steps:

a. Installation and Configuration: Metasploitable 2 was installed on a virtual machine, and the network was configured to simulate a realistic environment.

b. Nmap Scan: An Nmap scan was conducted to discover open ports, services, and potential vulnerabilities.

c. Nessus Vulnerability Assessment: A Nessus vulnerability assessment was performed to identify specific vulnerabilities and determine their severity levels.

d. Analysis and Remediation: The findings from the Nmap scan and Nessus assessment were analyzed, and appropriate remediation actions were recommended for each identified vulnerability.

**2. Description of the Metasploitable 2 installation process and network configuration:**

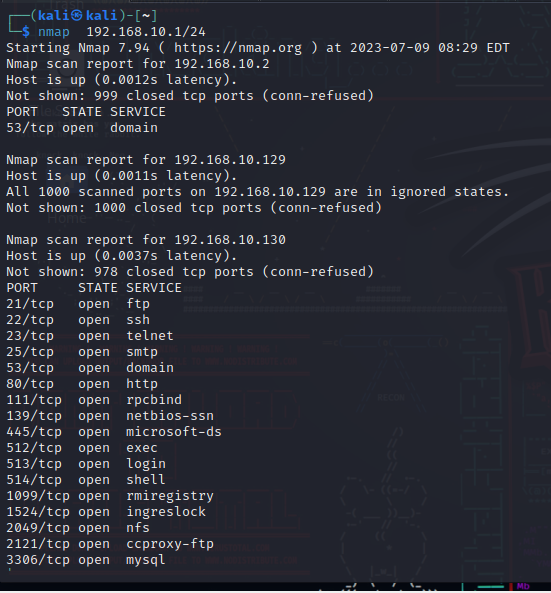
Metasploitable 2 was installed by following the official installation guide provided by the developers. The installation involved setting up a virtual machine with suitable hardware resources and then configuring the necessary network settings.

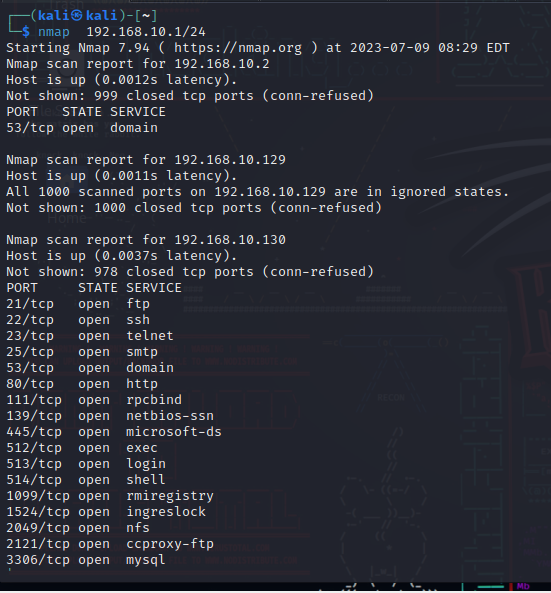
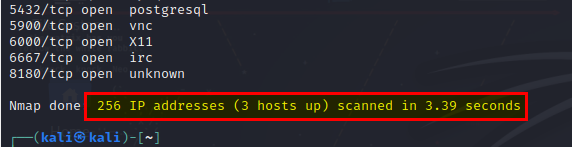
**2.1 Network Configuration:**

* Metasploitable 2 was assigned an IP address within the NAT network.
* The virtual machine was connected to the same NAT network, allowing communication with the host machine and other virtual machines.

**3. Nmap scan:**

First Nmap host discovery is done by executing a normal nmap scan in /24 subnet



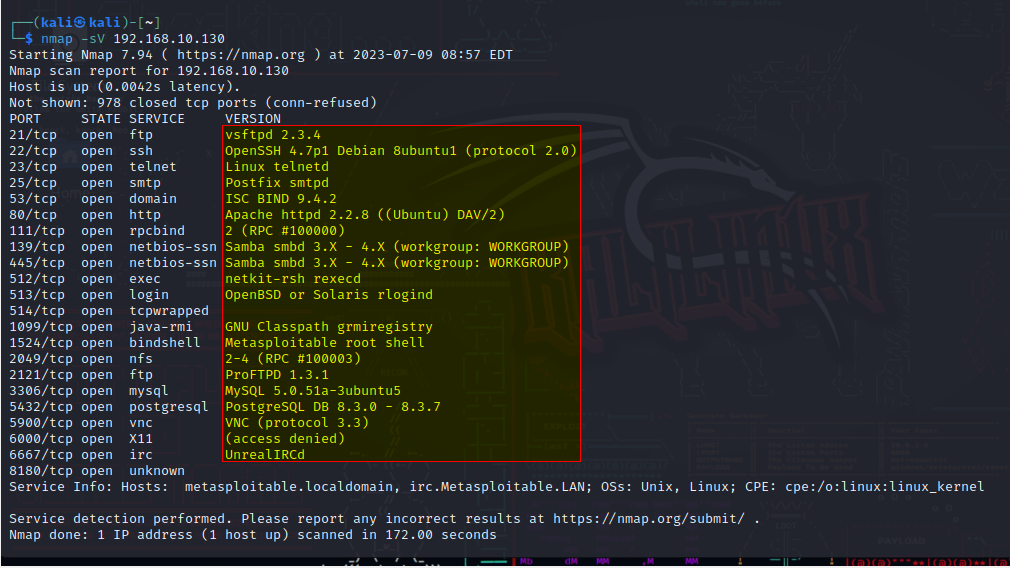
 

The Nmap scan revealed the following information about Metasploitable 2:

* Open Ports: The scan identified several open ports, including:

1. Port 21 (FTP): The FTP service was running.
2. Port 22 (SSH): The SSH service was running.
3. Port 23 (Telnet): The Telnet service was running.
4. Port 25 (SMTP): The SMTP service was running.
5. Port 53 (Domain): The Domain service was running.
6. Port 80 (HTTP): The HTTP service was running.
7. Port 111 (RPCbind): The RPCbind service was running.
8. Port 139 (NetBIOS-SSN): The NetBIOS-SSN service was running.
9. Port 445 (Microsoft-DS): The Microsoft-DS service (SMB/CIFS) was running.
10. Port 512 (Exec): The Exec service was running.
11. Port 513 (Login): The Login service was running.
12. Port 514 (Shell): The Shell service was running.
13. Port 1099 (RMI Registry): The RMI Registry service was running.
14. Port 1524 (Ingreslock): The Ingreslock service was running.
15. Port 2049 (NFS): The NFS service was running.
16. Port 2121 (CCProxy-FTP): The CCProxy-FTP service was running.
17. Port 3306 (MySQL): The MySQL service was running.
18. Port 5432 (PostgreSQL): The PostgreSQL service was running.
19. Port 5900 (VNC): The VNC service was running.
20. Port 6000 (X11): The X11 service was running.
21. Port 6667 (IRC): The IRC service was running.
22. Port 8180 (Unknown): The service running on port 8180 was unidentified.

**3.1 Services and Versions**: The scan also provided information about the running services and their versions.

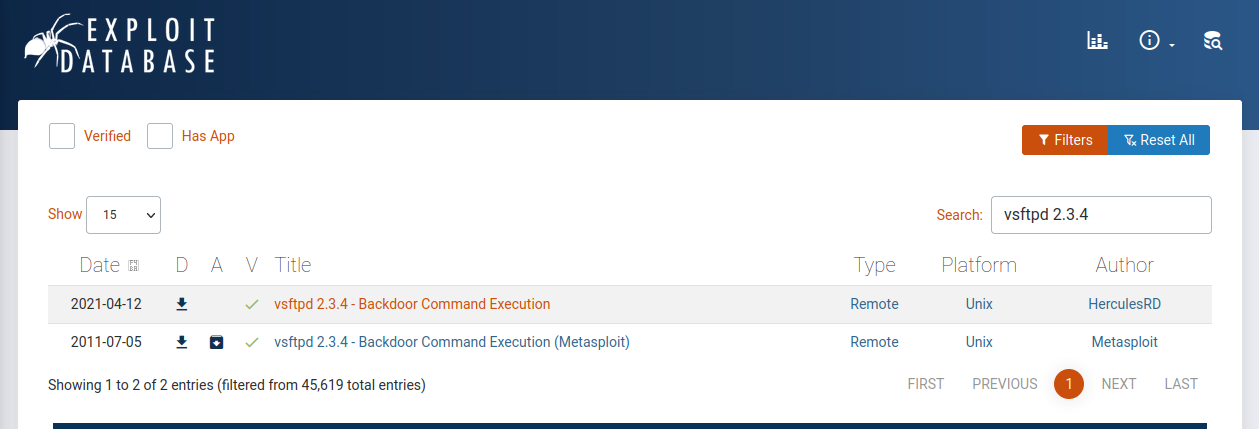


For example:

* FTP service: vsftpd 2.3.4
* SSH service: OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
* Telnet service: Linux telnetd
* HTTP service: Apache httpd 2.2.8 ((Ubuntu) DAV/2)
* NetBIOS service: Samba 3.0.20-Debian

**3.2 Potential Vulnerabilities:**

With the know service and version number we can search any known vulnerabilities out there for example:



**3.2.1 FTP service: vsftpd 2.3.4**

* **Vulnerability: vsftpd 2.3.4 Backdoor Command Execution**
* **Description:** The vsftpd 2.3.4 Backdoor Command Execution vulnerability refers to a security flaw in some earlier versions of vsftpd, including version 2.3.4, where a backdoor was present in the software. This vulnerability allowed remote attackers to execute arbitrary commands on the target system by exploiting the backdoor.
* **Mitigation**: Update to a secure version, Patch the existing version, Restrict access and minimize exposure

**3.2.2 SSH service: OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)**

The OpenSSH version 4.7p1 Debian 8ubuntu1 (protocol 2.0), there have been several vulnerabilities identified. Here is an example of a known vulnerability:

* **Vulnerability: OpenSSH User Enumeration**
* **Description:** This vulnerability allows an attacker to enumerate valid usernames on the OpenSSH server by observing different responses for valid and invalid login attempts. By analyzing the server's response timing or error messages, an attacker can determine whether a specific username exists on the system, which can aid in further attacks or targeted brute-forcing.
* **Mitigation**: Regularly review and update SSH server configurations, Consider stronger authentication methods, Enable auditing and monitoring update to new version and check for patches.

**3.2.3 Telnet service: Linux telnetd**

* **Description:** Linux telnetd, the Telnet daemon, is a service that allows remote users to log in to a Linux system using the Telnet protocol. It is important to note that Telnet is considered insecure because it transmits data, including passwords, in clear text.
* **Mitigation:** Disable Telnet Implement SSH, Configure SSH to meet your security requirements, Regularly update and patch.

**3.2.4 HTTP service: Apache httpd 2.2.8 ((Ubuntu) DAV/2)**

Apache httpd version 2.2.8, there are multiple vulnerabilities and security issues that have been discovered since its release. Here is one example of a known vulnerability

* **Vulnerability: Apache HTTP Server Remote Denial of Service (DoS)**
* **Description:** This vulnerability allows remote attackers to cause a denial of service (DoS) condition on the Apache HTTP Server version 2.2.8. It is caused by a flaw in the mod\_deflate module, which fails to handle certain requests properly. By sending specially crafted requests, an attacker can trigger a crash or excessive CPU usage, leading to a DoS condition.
* **Mitigation:** Update to a newer version Apply patches, Disable or limit modules, Implement a robust monitoring and incident response system

**3.2.5 NetBIOS service: Samba 3.0.20-Debian**

Samba 3.0.20-Debian is an older version of the Samba server software, and there have been several vulnerabilities identified in Samba since its release. Here is an example of a known vulnerability:

* **Vulnerability: Samba 'username map script' Command Execution**
* **Description:** This vulnerability allows remote attackers to execute arbitrary commands with root privileges on the target system. It arises due to a flaw in the Samba server's handling of the 'username map script' parameter in the smb.conf configuration file. Attackers can exploit this vulnerability by manipulating the 'username map script' parameter and executing arbitrary commands on the system.
* **Mitigation:** Update to a secure version, Patch the existing version, Restrict access and minimize exposure.

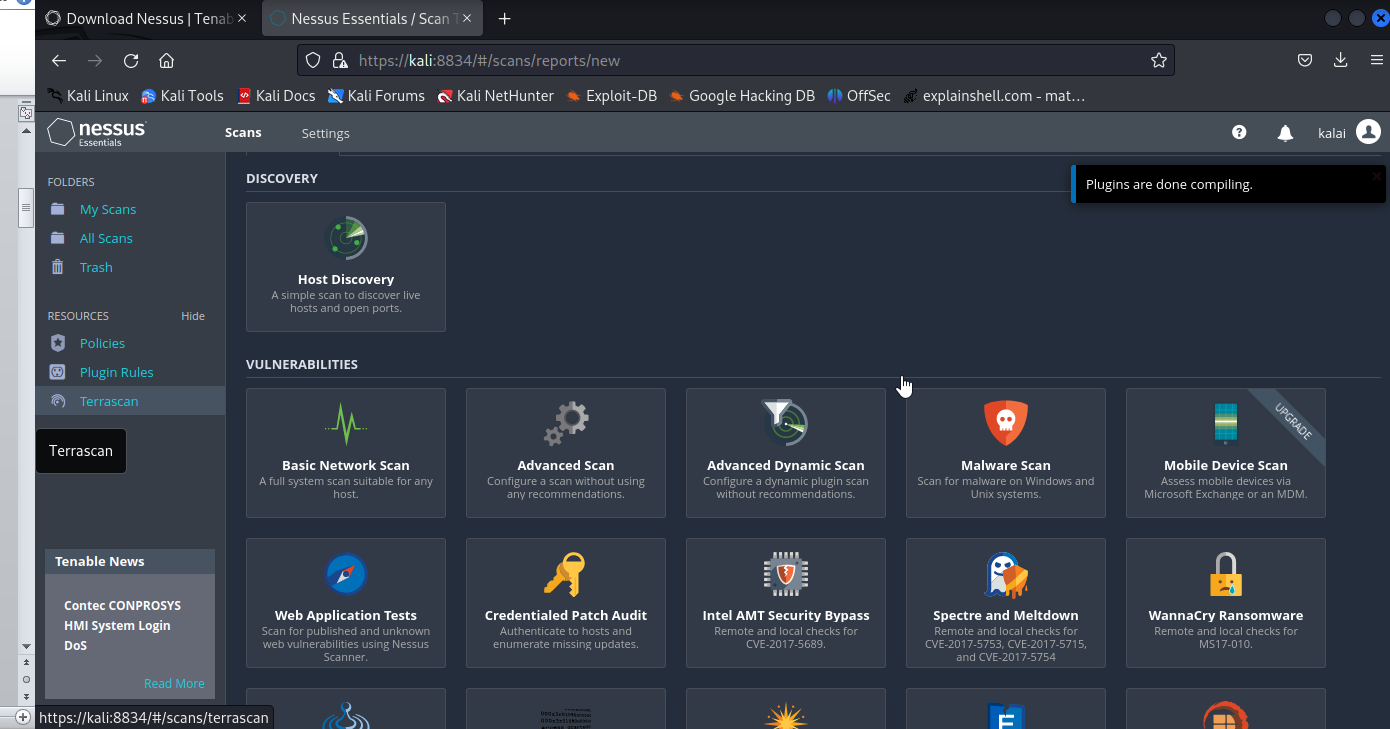
Based on the services and versions identified, the scan suggested potential vulnerabilities associated with outdated software versions and known exploits.

**4. Nessus:**

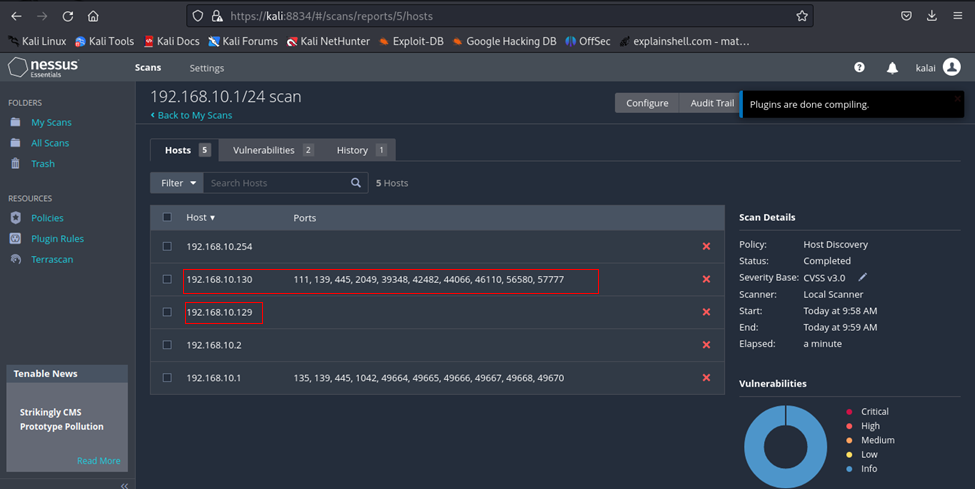
To initiate the Nessus service after installing it, you can use the following command:

**/bin/systemctl start nessusd.service**

* After Plugins are done compiling Nessus is ready for use and Host Discovery is made to find all the available Hosts in the network.

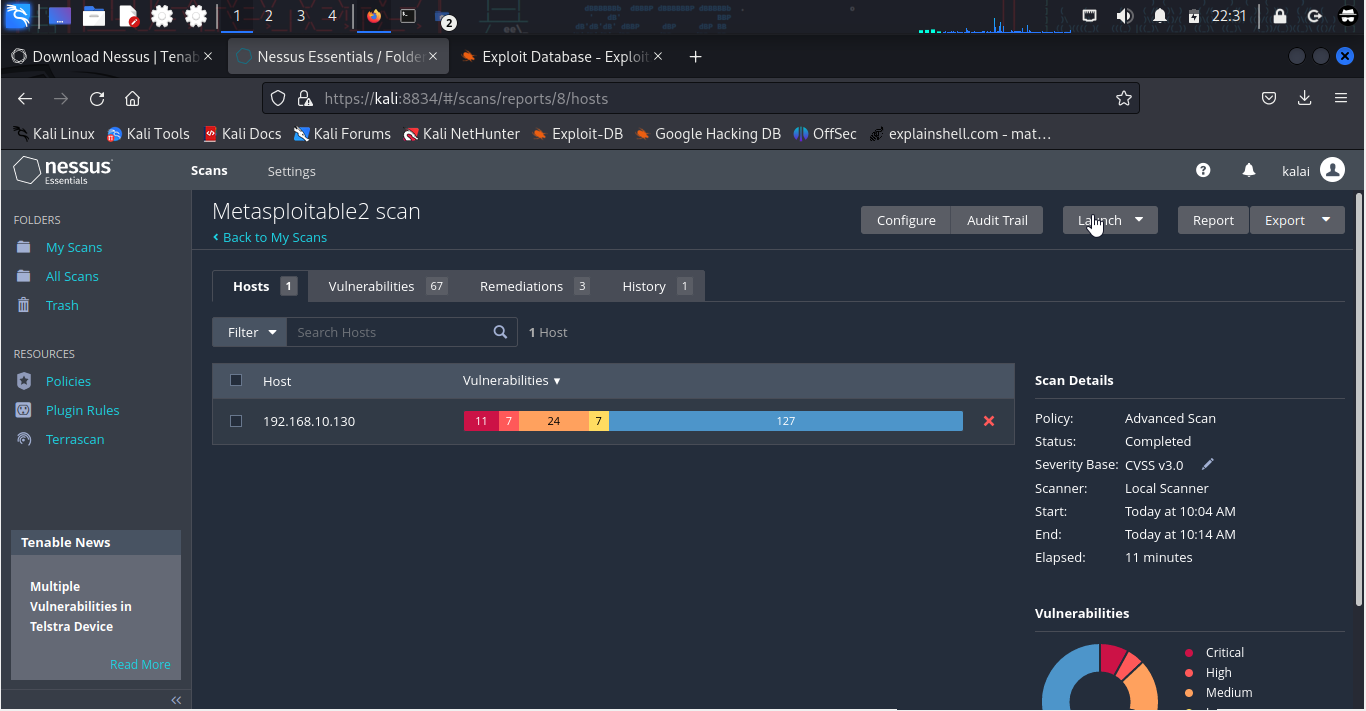


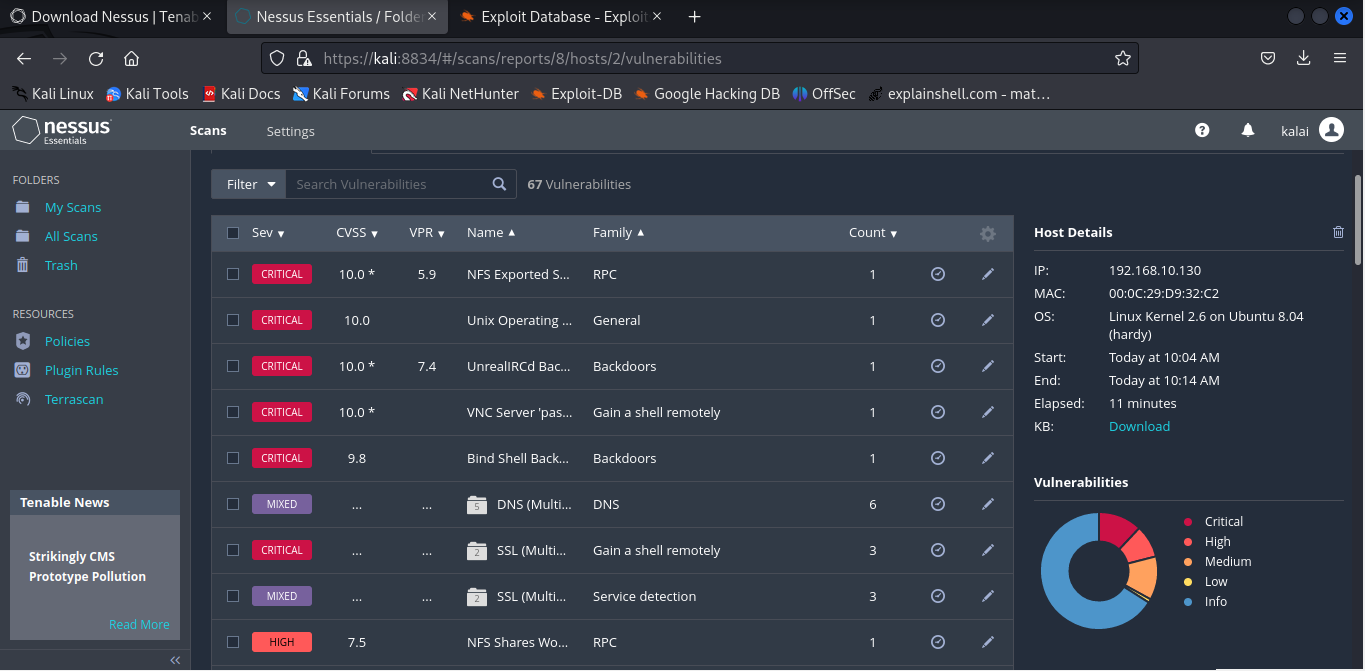
* There are 5 Hosts found in our network but actually 198.168.10.129 and 192.168.10.130 are actual hosts others are Broadcast and gateway address.

****

**4.2 Nessus Vulnerability assessment:**

A vulnerability scan using Nessus is conducted on-demand to identify and assess vulnerabilities within a system or network.





**4.3 Nessus Vulnerability assessment and Mitigation**

The Nessus vulnerability assessment conducted on Metasploitable 2 identified the following vulnerabilities some are:

**4.3.1 Vulnerability 1: NFS Exported Share Information Disclosure**

**Severity:** Critical

**Description:** At least one of the NFS shares exported by the remote server could be mounted by the scanning host. An attacker may be able to leverage this to read (and possibly write) files on remote host.

**Solution:** Configure NFS on the remote host so that only authorized hosts can mount its remote shares.

**4.3.2 Vulnerability 2: Unix Operating System Unsupported Version Detection**

**Severity:** Critical

**Description:** According to its self-reported version number, the Unix operating system running on the remote host is no longer supported. Lack of support implies that no new security patches for the product will be released by the vendor. As a result, it is likely to contain security vulnerabilities.

**Solution:** Upgrade to a version of the Unix operating system that is currently supported.

**4.3.3 Vulnerability 3: Debian OpenSSH/OpenSSL Package Random Number Generator Weakness**

**Severity:** Critical

**Description:** The remote SSH host key has been generated on a Debian or Ubuntu system which contains a bug in the random number generator of its OpenSSL library. The problem is due to a Debian packager removing nearly all sources of entropy in the remote version of OpenSSL. An attacker can easily obtain the private part of the remote key and use this to set up decipher the remote session or set up a man in the middle attack.

**Solution:** Consider all cryptographic material generated on the remote host to be guessable. In particular, all SSH, SSL and OpenVPN key material should be re-generated.

**4.3.4 Vulnerability 4: rlogin Service Detection**

**Severity:** High

**Description:** The rlogin service is running on the remote host. This service is vulnerable since data is passed between the rlogin client and server in clear text. A man-in-the-middle attacker can exploit this to sniff logins and passwords. Also, it may allow poorly authenticated logins without passwords. If the host is vulnerable to TCP sequence number guessing (from any network) or IP spoofing (including ARP hijacking on a local network) then it may be possible to bypass authentication. Finally, rlogin is an easy way to turn file-write access into full logins through the .rhosts or rhosts.equiv files.

**Solution:** Comment out the 'login' line in /etc/inetd.conf and restart the inetd process. Alternatively, disable this service and use SSH instead.

**4.3.5 Vulnerability 5: SSL DROWN Attack Vulnerability (Decrypting RSA with Obsolete and Weakened encryption)**

**Severity:** Medium

**Description:** The remote host supports SSLv2 and therefore may be affected by a vulnerability that allows a cross-protocol Bleichenbacher padding oracle attack known as DROWN (Decrypting RSA with Obsolete and Weakened encryption). This vulnerability exists due to a flaw in the Secure Sockets Layer Version 2 (SSLv2) implementation, and it allows captured TLS traffic to be decrypted. A man-in-the-middle attacker can exploit this to decrypt the TLS connection by utilizing previously captured traffic and weak cryptography along with a series of specially crafted connections to an SSLv2 server that uses the same private key.

**Solution:** Disable SSLv2 and export grade cryptography cipher suites. Ensure that private keys are not used anywhere with server software that supports SSLv2 connections.

**4.3.6 Vulnerability 6: X Server Detection**

**Severity:** Low

**Description:** The remote host is running an X11 server. X11 is a client-server protocol that can be used to display graphical applications running on a given host on a remote client. Since the X11 traffic is not ciphered, it is possible for an attacker to eavesdrop on the connection.

**Solution:** Restrict access to this port. If the X11 client/server facility is not used, disable TCP support in X11 entirely.

**5. Conclusion and recommendations for improving the security posture of Metasploitable 2:**

In conclusion, the security assessment of Metasploitable 2 revealed several vulnerabilities that could pose significant risks to the system if left unaddressed. To improve the security posture of Metasploitable 2, the following recommendations are provided:

* Regularly update the software and services installed on Metasploitable 2 to the latest versions, addressing known vulnerabilities.
* Implement strong access controls and authentication mechanisms to prevent unauthorized access to the system.
* Disable or replace insecure services like Telnet with more secure alternatives like SSH.
* Regularly perform vulnerability assessments and penetration testing to identify and address potential vulnerabilities in a proactive manner.
* Educate users and administrators about best practices for system security, including strong passwords, secure configurations, and regular backups.
* Maintain an ongoing monitoring and patching process to ensure that the system remains secure against emerging threats.
* By implementing these recommendations, the overall security posture of Metasploitable 2 can be significantly improved, reducing the risk of unauthorized access and potential exploitation of vulnerabilities.