Seppuku Proving Grounds

Penetration Test Report

Table of Contents

1.0-High-Level Summary………………………………………………………………………………………………………………………….3

1.1-Recommendations………………………………………………………………………………………………………………..4

2.0-Methodologies………………………………………………………………………………………………………………………………….5

2.1-Information Gathering………………………………………………………………………………………………………….5

2.2-Service Enumeration…………………………………………………………………………………………………………….5

2.3-Penetration Testing………………………………………………………………………………………………………………6

**Service Vulnerable**……………………………………………………………………………………………………….13

2.4-Report: Clean-up…………………………………………………………………………………………………………………13

1.0-High-Level Summary

An internal penetration test was performed on the potato network in the Offensive Security Proving Ground Labs. An internal test simulates an attacker that is directly connected into the network, in this case through a VPN tunnel.

The purpose of this test was to simulate an attack where the attacker had access to the network, with attempts made to break into a system and then elevate privileges on the machine.

Over-all, the intent was to enumerate the services on the exposed network, determine an attack vector to get access, and then exploit any flaw found within the system.

During the test, it was found that there were multiple points of access into the various systems from an external access point. The port 7601 web server had a directory that contained a list of passwords. Using these passwords, it was possible to crack the main http web site and log in with admin access.

Checking on the Samba share found that anonymous access was enabled, which allowed for enumeration of the shares, workgroups, and devices. A common name was found on the workgroup and device, so it was added as a username to check for.

With the provided password list, it was possible to brute force the ssh login for the username matching the Samba information. This represented a compromise of the local system, and files on there were available for retrieval.

Checking for vulnerabilities turned up a couple areas to check. One pointed to a location with the admin user and hash, that corresponded to the main website. This was able to be cracked with john the ripper, which enabled breaking into the main website through two different sources.

The system attempted to prevent moving about in the network by spawning a rbash shell, but this was able to be bypassed by spawning a tty shell using local programs to allow free movement within the system that the privilege level allowed access to.

Checking around also revealed private RSA keys were stored on a locally accessible directory. Downloading these keys allowed compromise to another user that was shown in the passwd file on the system.

With the original user, checking around also located a password file was located in the home directory. Checking against the known users on the system discovered it was for the third local user on the device. Logging in to that user showed a sudo escalation path pointing to a directory that did not exist on the second user.

Switching back to the second user allowed for creating the files and directories referenced in the sudo command. Changing back to the third user then allowed for running the files with root level privilege, which spawned a root shell. This represented a complete compromise of the system, and all files and directories.

1.1-Recommendations

It is recommended to update all passwords to a new group policy that prevents common passwords from being utilized for accounts, especially for admin or root level accounts. A passphrase of 15+ characters is recommended to prevent easy brute force methods from being utilized.

All files that are not intended to be publicly viewable should also be removed from their directories and stored in a more secure location that is not publicly facing. This will prevent accidental local file inclusion from occurring and harden the network against attack. Removing the private RSA key and the password list will be required to prevent attackers from easily gaining access to the network.

Write privilege should be removed from guest and anonymous accounts on the SAMBA share, and then the users deactivated from having access. If there should be a need to add a guest user to view the shares, removing the write access will prevent attackers from uploading content to shares if the guest user account is left on by accident.

Using usernames that are different from workgroup, share, and device names is also recommended. This will prevent relatively harmless information from being utilized to narrow down a brute force attack against the network. Avoiding common usernames and non-default accounts is also recommended for this reason as well.

Passwords and username should be removed from password files and notes on the different directories. One was located in users home directory, with another stored on the website directory. A more secure method of tracking passwords for accounts is recommended to prevent hackers from getting access to them for privilege escalations.

All private keys should be removed from being stored on the system. Anyone able to access one of these keys can authenticate as that user without the need for a password to log in with. Anyone with local access can view the /etc/passwd file to determine possible users the key belongs to, which makes breaking in as that user the work of minutes.

Removing unnecessary sudo privileges for users will also prevent abuse of the system. A good rule is to prevent one user from having sudo access to another user’s directories; there is no way for them to track if the other user removes the files, or changes something around. With a file being removed, an attacker can upload their own file in its place and run the sudo command on a file of their choosing.

SUDO should also only be used for files that absolutely must have it. If a file can be run without issue at a lower-level privilege, there is no need for that user to have that higher privilege level.

2.0-Methodologies

Below are the methods that were undertaken to break into the device, and ultimately achieve root access on the device.

2.1-Information Gathering

The information gathering portion was mostly null, as the network address of 192.168.191.90 was provided ahead of the pentest commencing.

2.2-Service Enumeration

This was mainly accomplished with nmap scan of the base 1000, followed by a scan of all tcp ports on the device.

21 FTP file server

22 Secure Shell Remote Access

80 HTTP Web Server

139 SAMBA

445 SAMBA

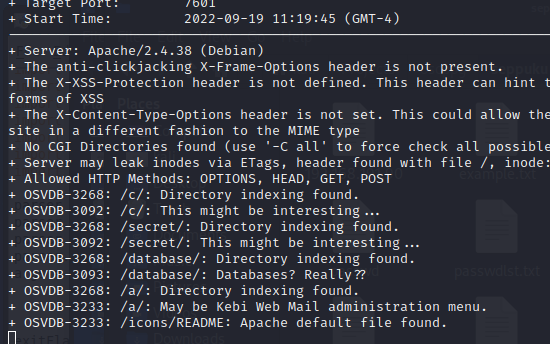
7080 HTTP Web Server

7601 HTTP Web Server

8088 HTTP Web Server

2.3-Penetration Testing

During the testing, it was found that there was a directory labeled secret on the port 7601 server.



Checking this location found a list of passwords that was used to ennumerate different services on the network.

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

This list led to the compromise of two different outward facing services that left them vulnerable to hacking: the main web server, and the Secure Shell remote access service.

Graphical user interface, text, application

Description automatically generated

A picture containing text, mammal

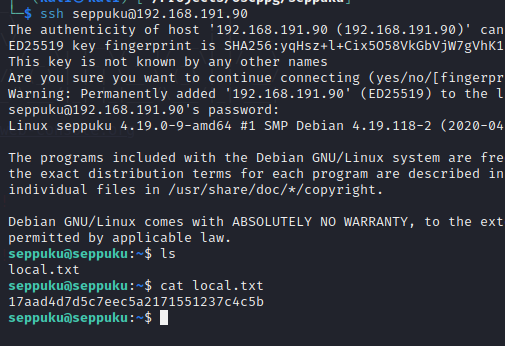
Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

The main website did not currently have any content that could be further exploited, and uploads were not allowed onto any of the directories using the authenticated credentials at this time. Hardening it against future attacks will be necessary to prevent any future upgrades to the site from being vulnerable to attack.

The user that matched the SAMBA information was also able to be hacked with the provided password list, which created a local level compromise of the network device.



It seems Rbash shell was used to prevent movement through the network that could compromise the system. There are many ways to escape it, and the python method allowed for easy bypass of this restriction.

Text

Description automatically generated

With this level access, it was possible to look around for various vulnerabilities on the system. There were multiple vectors found on the device, which also caused the compromise of three accounts on the system.

The first password was obtained in the home directory of the ssh account, that had a password file saved locally.

Text

Description automatically generated

The second method was found through checking for RSA keys that could be used to connect remotely without the need for a password. This was found within the directory of one of the web server locations.

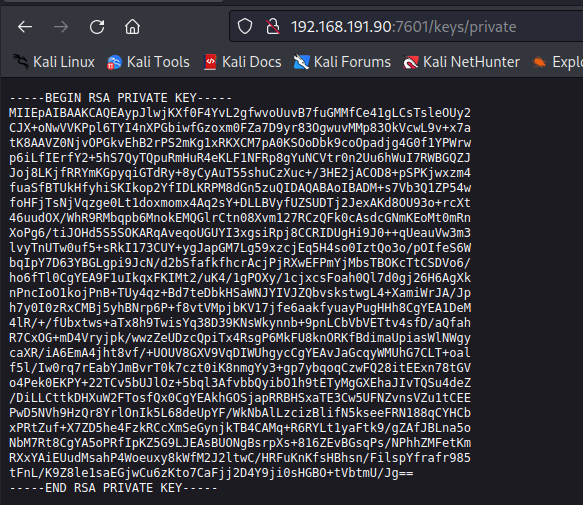
Graphical user interface, website

Description automatically generated

Text

Description automatically generated

Seeing the location this key was stored at, the pen test double-backed to the original external focus and checked the website from an outside source. It was verified that this private key was available for anyone able to bypass the random directory generation that prevents enumeration of the website directories.



With these two new local credentials, the other two users were able to be logged into and checked for possible privilege escalations to the root level.

Text

Description automatically generated

Text

Description automatically generated

It was found that there was a sudo authorization pointing to a different account, that no longer had the file or directory existing on it.

Text

Description automatically generated

Going between the two accounts, it was possible to create the directory and upload the file to two locations referenced in the sudo command. This file was created to spawn a shell with the privilege level of the command calling it.

Text

Description automatically generated

Text

Description automatically generated

With the files in place and permissions set for execution, the sudo command was issued and root shell access was created. This represented a total compromise of the network device, and all files and directories were vulnerable to attack.

Graphical user interface, text

Description automatically generated

**System Vulnerable 192.168.191.90:**

**Vulnerabilities Exploited:**

Local File Inclusion allowed for private key access and passwords to brute force

Weak passwords for the website and ssh allowed for brute forcing the authentications

Credentials stored locally on the system allowed for disclosure to low-privilege users

SUDO commands pointing to a different user pointed to a non-existent folder

**Severity: Critical**

**Proof of Privilege Escalation:**

Local.txt: 17aad4d7d5c7eec5a2171551237c4c5b

Proof.txt: b2edf921e8de41d6de770fa7be9f3ab7

2.4-Report: Clean-up

An exploit enumeration program was uploaded to the device that was removed after the output was received from it. Files were uploaded to the /tmp folder, and a directory was created to upload the same file to. The directory was removed after shell access was achieved, and the file in the /tmp directory was removed as well. This returned the network device to the state it was before the penetration testing started.