Dawn Proving Grounds

Penetration Test Report

Table of Contents

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

1.1-Recommendations………………………………………………………………………………………………………………..3

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

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

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

2.3-Penetration Testing………………………………………………………………………………………………………………5

**Service Vulnerable**…………………………………………………………………………………………………………9

2.4-Report…………………………………………………………………………………………………………………………………..9

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 guest access was allowed to the ITDEPT share on the samba service. The guest account was also afforded for READ and WRITE access to the share. Checking through the web service directories found a management log that was able to be downloaded. This listed the processes that were being run, and the directories these files were being activated from.

It was found that two different files were being called that referenced the ITDEPT share, that did not have those files available. From there, two shell files were created with those names and uploaded through the samba share. The calls in the log were occurring frequently, and both shells generated a connection to the network device.

Each shell generated gave access to different accounts, which allowed for greater exploitation due to the different permissions afforded each account.

Through the SUID enabled binaries, it was found the zsh was one of the ones set with the SUID permission. With this, either account was able to achieve root access with the newly spawned shell.

The web server account also had another option through its sudo permissions. The sudo command was able to be run without any password required. Abusing this allowed for spawning a root shell by running a shell command through the sudo root command.

Both of these methods allowed for a total compromise of the network device and rendered all root level content available for retrieval from the penetration tester.

1.1-Recommendations

It is recommended to adjust the permissions of the guest user to read only for the SAMBA service. Once that is accomplished, the guest account should also be deactivated. This two-pronged approach means that even if the guest account is enabled in the future for a reason and left active, there will be no means for someone to exploit its permissions to upload files to the share.

The logs that are kept on the web server had 3 of them with permission denial, and the final one available for download externally. This should be updated so that all logs posted require authentication to access. Ideally, it should be review whether any logs need to be posted to the website. If it is found to not be necessary, then a different storage method should be utilized for keeping the log records.

The management log should be periodically reviewed for any files that are no longer functioning, whether through deletion or moving to a different directory. The system could also be checked to see if the calls are producing results showing no active service for the call. Any of these should be removed from the system, to prevent someone from inserting a file with the name in the location.

If it is only a temporary move of the file, placing dummy files with read only access should also prevent people from uploading malicious shells into the directory.

The SUID setting should be removed from the zsh binary, as that is exploitable through any user account that is able to authenticate to the system. This will always be a liability, and as such should be removed as an exploitation vector.

For the web user account, the no password permission on the sudo command should be removed. Web accounts should have minimal permissions in general, and only be able to run as root specific files that absolutely require it. The web server should not need dynamic root access, as it should be repeating the same functions programmed on it over and over.

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 192.168.191.11 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. This left it with the following ports as possible exploit vectors:

80 HTTP web access

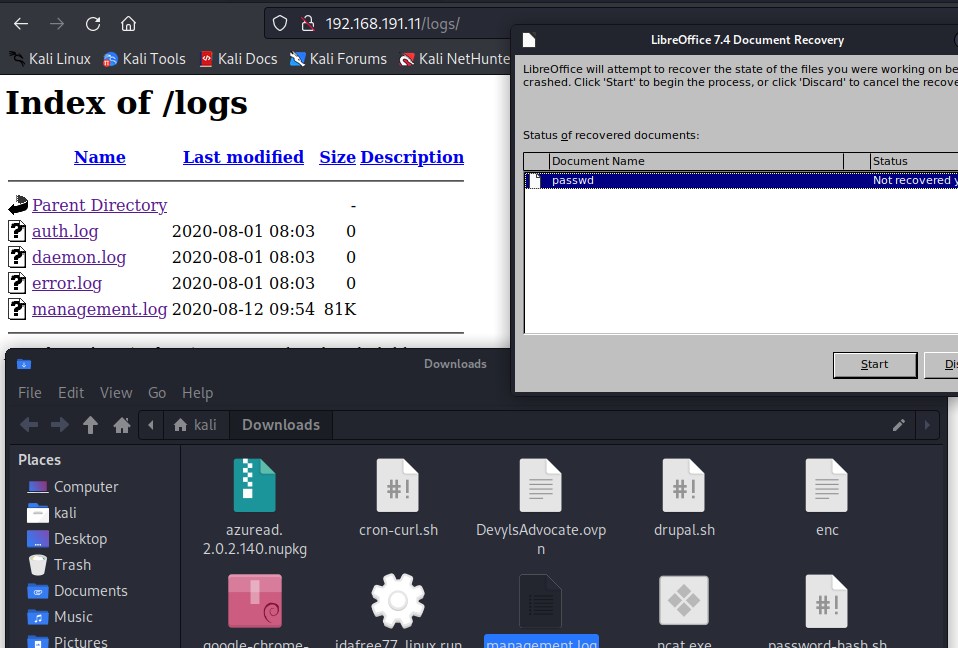
139 Samba

445 Samba

3306 MySQL MariaDB

2.3-Penetration Testing

During the testing, it was found that there were some log files stored on the website while enumerating the directories. Checking the logs showed that one of them was downloadable.



Viewing the content showed that it was a listing of processes running on the system.

A screenshot of a computer

Description automatically generated with medium confidence

It was also found that the smb share enumerated with a script found that guest access was allowed for the ITDEPT share with read/write access. This was verified by uploading a test file onto the service.

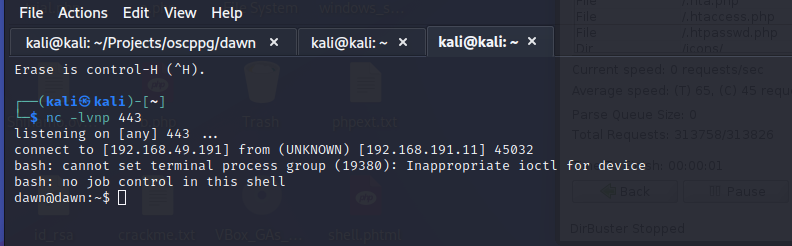
Reviewing the processes showed there were two services that were being called from the ITDEPT folder. There was no listing for them within the share. With this, two shell files were created matching the names from the log file, and uploaded onto the share.

Text

Description automatically generated

This received connections back from the device that authenticated to two different users: the web service and a user on the system.

Both of these users had access to a SUID set binary that was able to be exploited for root access on the system.



Graphical user interface, text, application

Description automatically generated

By running the zsh binary, a root shell was generated that gave root level access to all files and folders on the system. This represented a complete compromise of the network device.

Graphical user interface, text

Description automatically generated

It was also found that the web service user had a sudo privilege assigned to the account that offered another method of achieving root access. With this, a call to the sudo command allowed for spawning a root shell directly.

Text

Description automatically generated

Both methods offered a viable method of privilege escalation to the root level. There is also a possible exploit vector in the user account that offered sudo privileges for the mysql service. This was not able to be exploited during the testing, but should be removed to prevent any possible avenues opened up by updates or changes in the service.

Text

Description automatically generated

**System Vulnerable 192.168.191.11:**

**Vulnerabilities Explioited:**

Guest access enabled for SAMBA share

Write access enabled for Guest account to SAMBA share

Local file inclusion disclosing sensitive information on the web service logs

Incorrect permissions applied on one of the logs on the web service

SUID enabled binary that allows for spawning a root shell

SUDO privilege for the web account that allowed for spawning a root shell

**Severity: Critical**

**Proof of Privilege Escalation:**

Local.txt: ea9a78978593b8372a7261303f1aedfd

Proof.txt: 8599879b745901b6874e0a16d6763766

2.4-Report: Clean-up

Clean up was accomplished by removing the remote shell programs that were uploaded to the SAMBA share. Exploits were uploaded to the /tmp directory to attempt adding a user that were unsuccessful, and subsequently remove from the directory. Linpeas was uploaded to check for any missed exploits on the service, then removed after the output was received. After these reverts, the system is in the same configuration it was from before the penetration test.