Pyexp 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**…………………………………………………………………………………………………………8

2.4-Report…………………………………………………………………………………………………………………………………..8

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 a weak password allowed for access into the network with the root account into the mysql database. Once inside the database, the fernet key with a user’s credentials was able to be decrypted.

Using the credentials allowed for remote connection to the network device through a local user on the system. Using the sudo privileges of the user allowed for a root shell to be spawned. All directories and files were vulnerable for retrieval at that point.

1.1-Recommendations

Changing the base ports of the MySQL database should reduce the number of attackers that will target the service. At this point there are multiple avenues that can be undertaken to prevent brute force attacks.

The IP table can be changed to only allow certain IPs to connect to the service. This will work if there are few users that use the service. Authenticating with certificates is another method that can be used, making it a two-factor authentication method. The final method is to enable MySQL protection on the service. This will block an IP after a certain amount of failed login attempts.

Enforcing a stricter password policy will also prevent easily guessable passwords from being exploited with wordlists such as rockyou.txt. It is recommended to utilize a group policy requiring a passphrase of 15+ characters for current standards.

The Secure Shell remote access service should also be set up with brute force attack protection. There are various mechanisms on the pam.d process that can disable the service for the IP or user when too many failed logins are detected.

The host.allow file can also be updated to only allow access for certain IPs to connect to the service. This method is recommended if there are not too many people who will be connecting to the device from a remote location.

Iptables can be used to limit the rate at which people can connect to the service, which will act as a further barrier against brute force attacks. This setting can easily boost a brute force attack against a known user to take years, if not decades.

Removing credentials from the MySQL database is also a good action to take, so that the compromise of one service does not lead to compromising another service. Keeping a good separation between the various services will prevent a zero-day exploit of one service being discovered from being utilized to gain access to all services on the device.

The final recommendation is to remove SUDO privileges that are not essential for job function. Any SUDO authorized programs should be evaluated, and any that do not meet this criteria should be removed. Many binaries are able to be exploited using those privileges, and allow for root shell access in unintended ways.

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 192.168.191.118 was provided ahead of the pentest commencing.

2.2-Service Enumeration

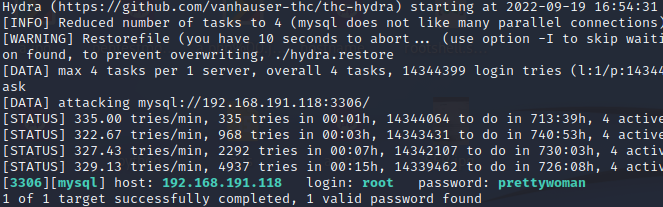
This was mainly accomplished with nmap scan of the base 1000, followed by a scan of all tcp ports on the device. The UDP top port only returned 10 ports open on the top 1000 ports, none looked available for an exploit. This left it with the following ports as possible exploit vectors:

1337 Secure Shell Remote Access

3306 MySQL

2.3-Penetration Testing

During the testing, it was found that two services on the TCP ports were able to be reached externally. Brute force was attempted against both the SSH service and the MySQL service at this time. The results of the testing found that the root password for the MySQL service was vulnerable to brute force, and the password was found.



These credentials were used to enumerate the database, which uncovered a Fernet hash and key.

A screenshot of a computer

Description automatically generated with medium confidence

Using these values exposed the credentials of a local user.

Graphical user interface, text, application, email

Description automatically generated

Testing these credentials against the Secure Shell service provided access to the network device remotely. This compromised all local level files and directories for retrieval from the network.

Text

Description automatically generated

Various privilege exploit paths were checked that the local user was able to utilize, and it was found that there was a SUDO privilege command that ran a program that could have commands injected into them.

Text

Description automatically generated

Using this command, it was possible to read the contents for root files and spawn a root shell for the network. This represented a complete compromise of the network device, with all files and folders available for access.

Text

Description automatically generated

Text

Description automatically generated

**System Vulnerable 192.168.191.118:**

**Vulnerabilities Exploited:**

Weak password for root account on the MySQL database

Sensitive information stored on the MySQL database

SUDO permissions for a program that allowed root shell access

**Severity: Critical**

**Proof of Privilege Escalation:**

Local.txt: 7dc083f68aced88bd85c01cdd737b46f

Proof.txt: a2858fb2f4126dbd80bab7c58f4a990c

2.4-Report: Clean-up

Multiple exploit checking scripts were uploaded to the /tmp directory which listed out possible attack vectors for privilege escalation on the service. Once the output was received, these files were removed from the device. These were the only changes made on the system, and it is currently in the state it was before the penetration test began.