Evil Box One Proving Grounds

Penetration Test Report

Table of Contents

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

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

2.0-Methodologies………………………………………………………………………………………………………………………………….4

2.1-Information Gathering………………………………………………………………………………………………………….4

2.2-Service Enumeration…………………………………………………………………………………………………………….4

2.3-Penetration Testing………………………………………………………………………………………………………………4

**System Vulnerable**………………………………………………………………………………………………………..7

2.4-Report………………………………………………………………………………………………………………………………….7

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 testing, it was found that the website contained a php file that was able to inject read commands on files within the network. This was abused to get both viable users configured on the device, and then to gain access to their private SSH key. After cracking the hash, it was possible to use this key to gain access into the network.

Incorrect permissions assigned to the passwd file allowed for using a local-privilege account to create a root level access user to inject into the file. This user account was then used to gain access to all files and folders on the network.

1.1-Recommendations

The php file’s access to sensitive files on the network needs to be prevented to avoid information disclosure to public access. This can be done by sanitizing the inputs available for the program to interpret, disabling, or removing the php file, or recoding its function for the directly targeted use it is intended for.

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.

The permissions for write access to the /etc/passwd file should be updated to prevent any non-root user from being able to modify the file. This bypasses the separation of privileges intended with the root and local user system. By making the permissions for write to be root only, it will prevent an easy escalation method from local privilege user to root privilege.

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 192.168.191.73 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:

22 Secure Shell Remote Access

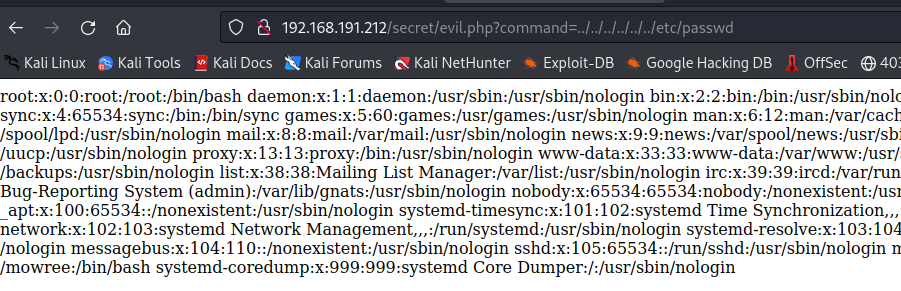
80 Web Service

2.3-Penetration Testing

During the test, enumeration was begun to determine services that were able to be reached externally from the network. This turned up two ports available for exploitation avenues: the SSH service and the web server.

Brute force was set to run against the SSH service while enumeration was done to determine all available files and folders on the website. Common vulnerabilities were also checked for on the php files found in the directories. Testing determined that there was a php file loaded in the secret directory that did not seem to have a function, and the web page loaded blank for it.

The file was checked to find which function call the php required to be activated, which led to the discovery that network files were able to be viewed externally from the web browser.



This information disclosure provided the local level user account that was configured on the device. Checks were made to determine what information could be gained from the website function, and a private ssh key was discovered.

Graphical user interface, text, application

Description automatically generated

Attempting to use the key requested a password to access the contents of the private key for access to the network to be granted.

Text

Description automatically generated

The hash used a weak password to lock it out from exploitation. Using a hash cracking program, the password was quickly found using a common password list.

Text

Description automatically generated

Using the key with the recovered password allowed for access to the network device, and access to the local privilege files and folders on the system. This was proven out with the contents of the local.txt file being retrieved.

Text

Description automatically generated

Vulnerability checks were made on the system and it was found that the /etc/passwd file allowed for write privileges for local privilege accounts.



Using these privileges, a new user was created with root privileges within the file. This new user was exploited to gain root access to the system. This represented a complete compromise of the network device, with access to all files and folders on the system.

Text

Description automatically generated

**System Vulnerable 192.168.191.212:**

**Vulnerabilities Exploited:**

Sensitive information disclosure on the website providing users and private keys

Weak passwords used on the private key

Incorrect permissions on the /etc/passwd file allowing for root level user creation.

**Severity: Critical**

**Proof of Privilege Escalation:**

Local.txt: bfc7037241253279fdf1be8a5aaf611d

Proof.txt: 14b0e757ca490de37d041c437cf4a30c

2.4-Report: Clean-up

Exploit scripts were uploaded to the /tmp folder to take advantage of common vulnerabilities in the current binary versions on the system. These exploits were removed after finding required binaries not installed on the system to successfully accomplish the exploit.

Vulnerability enumeration scripts were uploaded to the /tmp folder to verify there were no common missed vulnerabilities in the manual checks on the system. Once the outputs were received, the scripts were removed from the network device.

All information retrieved from the device and found through testing are isolated in their own directory on the attack system and will be removed pending hand-over of all information requested per the terms of engagement for this test.