FunBoxEasy 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…………………………………………………………………………………………………………………………..5

**System Vulnerable**……………………………………………………………………………………………………….12

2.4-Report…………………………………………………………………………………………………………………………………12

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, vulnerabilities were discovered that allowed for privilege escalation, along with some other potential privilege escalation vectors. The main vector to getting access was through default credentials on the store login page for the admin user. With this, it was possible to upload a php shell that allowed access into the network device with the server user.

From there, a SUID set file was available that allowed for a root shell to be gained on the system. This allowed for complete compromise of the device.

Further issues were discovered after uploading a linpeas program to check for vulnerabilities. The vulnerabilities discovered would require other programs to be downloaded to be vulnerable. Given that the fix action for both vulnerabilities is a simple update, it is recommended to go ahead with the update. This way if one of the programs that allows the compromise gets installed down the line, the privilege escalation vectors will not be enabled.

There was also a list of passwords located in a user directory that gave some admin passwords and the ssh access for the user. This was verified to be a valid credential for connection, and it added another SUID enabled file. With this file, it was possible to expose the root hash that could then be brute forced from the attacker’s machine.

1.1-Recommendations

For all the various websites being hosted on this server, they will need to be checked to ensure that no default credentials are left active on them. It is recommended to disable all admin accounts, after creating different usernames that have admin access to take their place.

The shell upload is difficult to overcome; as while there are many ways to prevent it, there are just as many ways to bypass them. The way this was able to exploit, you should be able to mitigate by having specific books set for display on the store page. By jumping ahead of the existing books on the list, it enabled the shell to be executed by being displayed on the store page.

It could also check for file size, file type, removing second .’s in a file, and various other methods. This should all be validated server-side, as client-side validation can be circumvented. Do not rely on headers or file-types supplied by a user; always make sure it is checked independently by your server.

From there, it is recommended to remove all SUID enabled file settings, that do not absolutely need sudo privilege. The mtr and time files were able to be exploited for root shell with time, and root level reading with mtr under the user tony account. These will need to have their privilege level changed to prevent privilege escalation attacks.

There were two vulnerabilities discovered that are both related to the polkit vulnerability that was recently discovered. Both of these vulnerabilities allow for root level privilege escalation. There looks to be a few missing programs on the device that would allow the exploits to function; but it is better to patch it so there is no concern of someone needing to download them in the future and creating the privilege escalation vector. An update of the system, or changing the permissions of pkexec, should prevent this from becoming an issue.

It was also discovered that there was a file loaded on one of the users home directories, that contained a list of passwords. This further compromised the user through secure shell remote access, as well as providing admin credentials for two of the websites. It is recommended that the users have a different method of storing their passwords, to prevent credentials from being stolen by attackers.

It was also found that the SQL database could be accessed through the store website. This allowed for the admin password to be found through a dump of the database using sqlmap. The websites will need to be validated to not allow for injection attacks that can directly access the SQL database through the website.

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 192.168.191.111 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 ports only returned 3 named ports open, none looked available for an exploit. This left it with the following ports as possible exploit vectors:

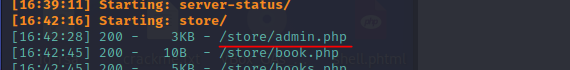
22 Secure Shell Remote Access

80 HTTP Web Services

33060 MYSQLX

2.3-Penetration Testing

Performed the initial intrusive scanning with nmap, nikto, dirbuster, and gobuster. The directories of the web page were found, along with various logon pages. Checked through them with ZAP to see if any obvious vulnerabilities stuck out, none were alerted. Tested through the various login pages, and found the store directory allowed for login with default credentials for the admin account.



Background pattern

Description automatically generated with medium confidence

With this access, it gave the option to add new book to the website. Initially created a book with a reverse shell for the image, that did not allow for the shell to form. It was found that by changing the naming conventions to get to the top of the list would allow it to be displayed on the store front page.

Graphical user interface, text

Description automatically generated

Once the page was refreshed, the reverse shell code was triggered and a shell to the device was formed.Text

Description automatically generated

Graphical user interface, text, chat or text message

Description automatically generated

From there, it was possible to locate the compromised local data and get the contents of the local.txt file.

Text

Description automatically generated

After that, the SUID set files were checked for possible vectors of privilege escalation on the device.

A screenshot of a computer

Description automatically generated with medium confidence

The time file allows for a root shell to be spawned when its SUID bit is set. This was then exploited to gain the root shell on the device. This represented a total compromise of the system, and the root contents of proof.txt were able to be gained.

Graphical user interface, text

Description automatically generated

After verifying root access was gained, started to look around for other possible exploits on the system. Linpeas was uploaded to the device and run, and it found two related CVEs that were possible attack vectors on the system of critical severity.

Text

Description automatically generated with medium confidence

Exploit script taking advantage was attempted, but key programs were missing from the system that would have enabled them to work.

Moving on from there led to enumeration of the user’s home directories.