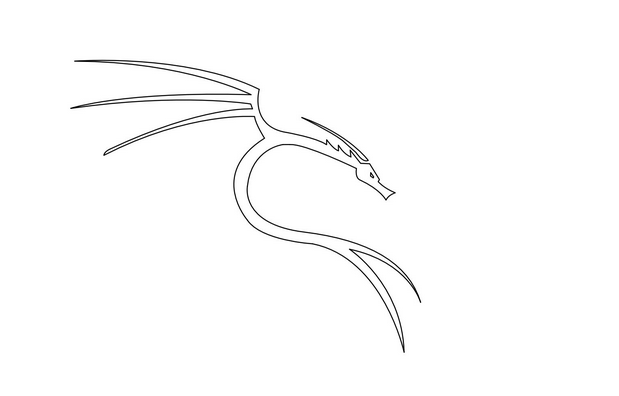
**Penetration Test Report**



* By Anh4ckin3

**Tables of** Contents 2

Executive Summary 3

Summary of result 4

Attack Narrative

Scanning 5

Enumeration 7

Exploit 8

Post exploitation 11

Conclusion

Recommendation 20

Risk ranting 21

Vulnerability Detail and Mitigations

Scanning 22

Smb server access 22

WordPress exploitation 22

Escalate to root thanks to SUID 23

Persistence with ssh 23

**Executive summary**

Anh4ckin3 was contracted by Billy Joel to make a penetration test in is server in order to determine its exposure to a target attack. All activities were conducted in a manner that simulated a malicious actor engaged in a targeted attacks against Billy Joel with the goals of:

* Identified if a remote attacker could penetrate Billy Joel defenses
* Determining the impact of a security breach on :
  + Confidentiality of the Alfred private data
  + Internal infrastructure and availability of Alfred information systems

All actions carried out on this system are controlled and approved by the information system owner. This test has been carried out with the aim of finding vulnerabilities in the information system, which would give access to ill-intentioned people to access private or sensitive data. The penetrator will leave with the target's IP address on the network, connected via an openVPN VPN.

**Summary of result**

The test begins with active reconnaissance using the nmap tool to discover open ports and the services running behind them. There are 3 open ports: port 22 with an SSH server, port 80 with an apache server hosting WordPress and port 445 with a samba server.

An enumeration with the enum4linux tool will give us a user and a share. A brute force attack will be used on the samba server and this will give us the information that the server is fully accessible with an anonymous login, but the server comptenue will be a "rabbit hole".

The enumeration of the WordPress will be mainly with the wpscan tool, thanks to which we'll find the version and themes of the site. A search on known exploit databases will reveal that the WordPress version is vulnerable to an RCE. To do this, we need at least the password of one of the WordPress authors. Using wpscan, we'll list the users and find the password of one of them, giving us access to the machine via the metasploit exploit.

The privilege escalation part will be an explanation of the unusual SUID that will give us root rights.

And finally, we'll establish persistence via the ssh protocol.

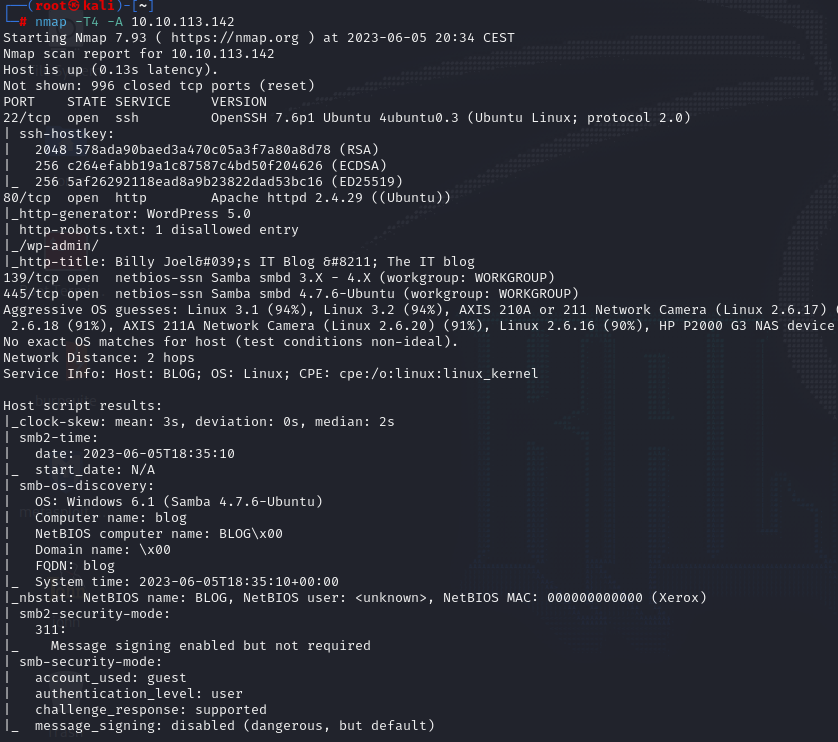
Attack narrative

Scanning

The test starts with active recognition: we'll scan the remote server's ports with the nmap tool to detect open ports and the services running on them.

The nmap command: nmap –T4 –A target\_ip

The result of the scan gives us 3 ports open, first port 22 with the ssh protocol (Openssh version 7.6p1) a port 80 with the http protocol heberger by an apache server (version 2.4.29)

What is interesting is that the scan gives us the information that there is a WordPress hosted on the apache server, we even have the version of the WordPress (5.0). Finally a samba server is active on port 445 (version 4.7.6). The scan also gives us the type of explain system of the server in occurrence this will be Linux and potentially a machine Ubuntu

Attack narrative

Enumeration

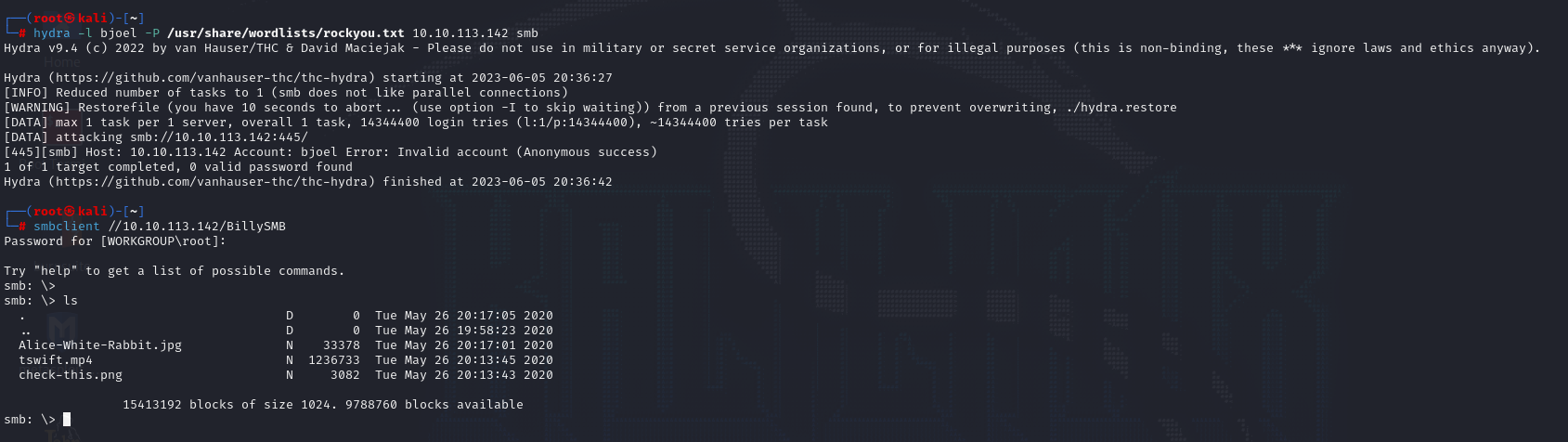
to start the enumeration we will start with the samba server which gives us potentially a quick gateway to the server or important information such as username. To list the samba server we will use the enum4linux tool. Valuable information must be revealed such as shares and even a user (bjoel)

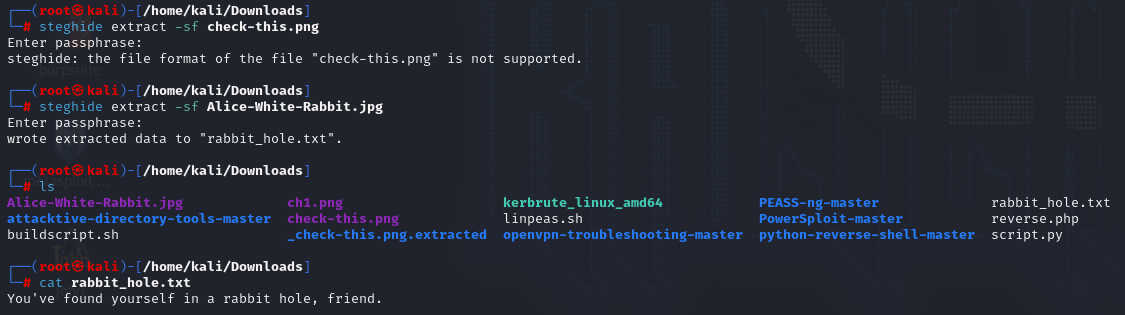
We will therefore perform a brute force attack to find the password of the bjoel user.

Hydra command: hydra –l bjoel –P /usr/share/wordlists/rockyou.txt target\_ip smb

But the attack will quickly end this because the server is in full access via the anonymous login, we will recover the accounts of the server is analyze it.

A small part steganography is announced because there is nothing interesting apart from the images, but it will be only a rabbit hole.





Attack narrative

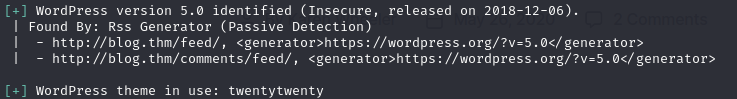
Enumeration

Now it’s time to take a look at the WordPress. To start, you will have to edit the/etc/host file to solve the DNS (blog.thm) and have an optimized access to the website. To find the potential hide page we will launch a tool (Gobuster) that will brute force the website by testing full of common word (http://blog.thm/secret for example).

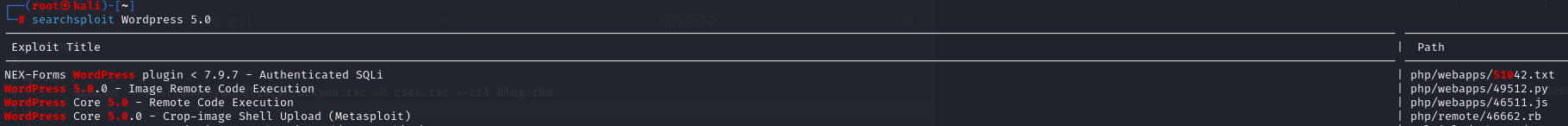
Gobuster command: gobuster dir –u <http://blog.thm/> -w /usr/share/wordlists/dirb/big.txt –x php,txt,js

But the listing with gobuster did not yield anything really interesting. We will now focus on the WordPress and try to find more information. For this the wpscan tool will help us it will give us the version of WordPress (which we already know 5.0) and the themes (twentytwenty).

Wpscan command: wpscan –-url http://blog.thm/



The first thing to do is to see if the WordPress version is vulnerable to a known exploit. The Searchsploit tool can help us find a vulnerability potential already known for this version of WordPress. And yes there is a vulnerability on version 5.0 of WordPress, it is an exploit of type remote code execution that allow us to execute code on the remote server and thus potentially have a command line access to the server. The exploit needs a password and username to allow the RCE vulnerability.

searchsploit command: Searchsploit wordpress 5.0

Attack narrative

Exploitation

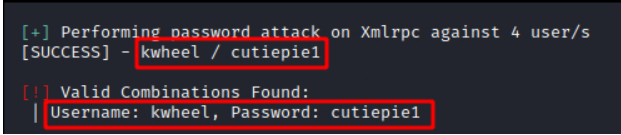
So we need a password and a user for the exploit to be successfully executed, so we know that it is the first phase of exploitation of the wordpress to launch a brute force attack to find the logins. The tool wpscan this load of this it will list the users (kwheel) and find the password (cutiepie1) of one of the authors of the site.

Wpscan command:

Found user: wpscan –-url <http://blog.thm/> –enumerate u

Found password: wpscan –url <http://blog.thm/> -u user.txt –p /usr/share/wordlists/rockyou.txt

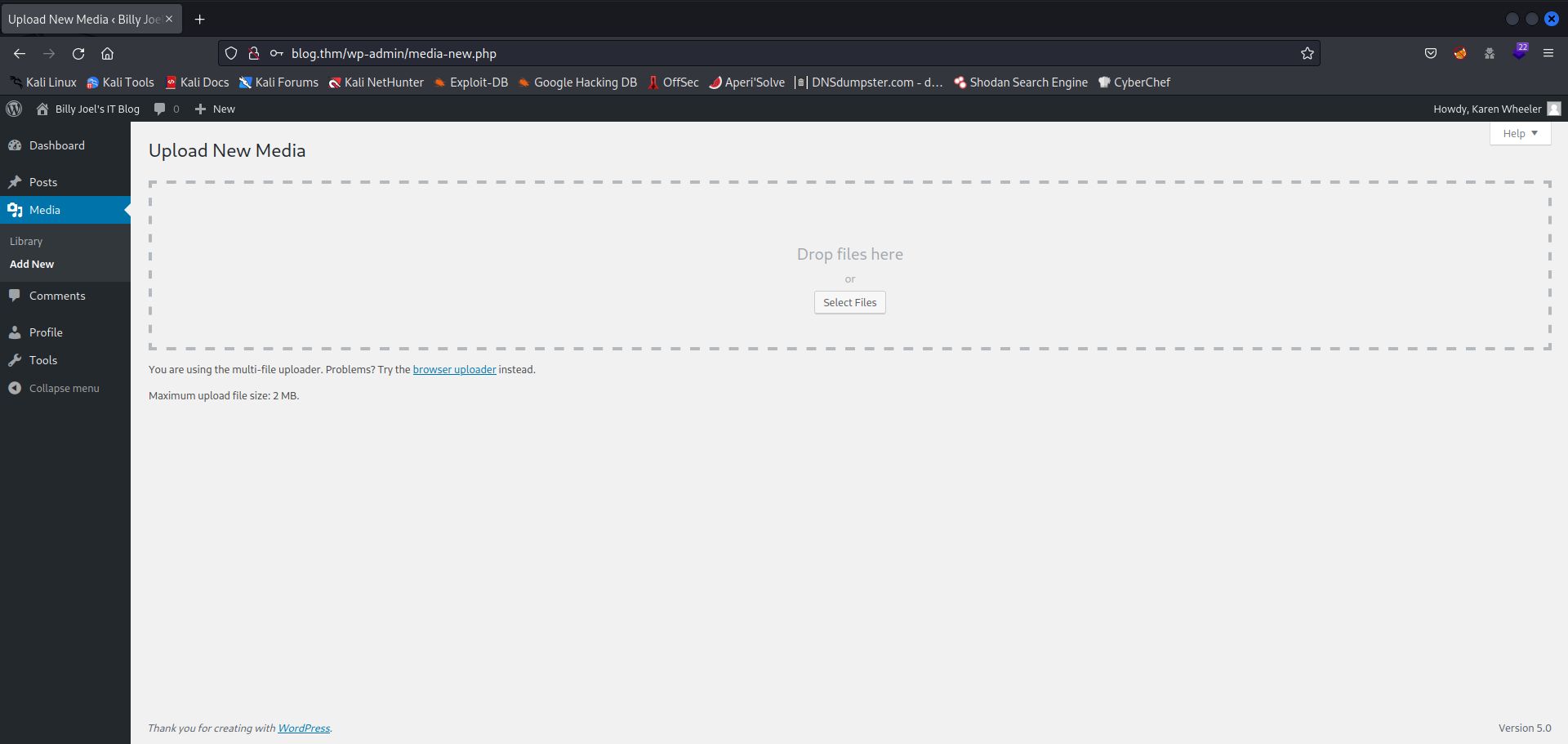




Attack narrative

Exploitation

So we found a user and his password, we now have access to the WordPress via the kwheel user.



It’s time to configure our exploit on metasploit framework and potentially access the remote server.

This module exploits a path traversal and a local file inclusion vulnerability on WordPress versions 5.0.0 and <= 4.9.8. The crop-image function allows a user, with at least author privileges, to resize an image and perform a path traversal by changing the \_wp\_attached\_file reference during the upload. The second part of the exploit will include this image in the current theme by changing the \_wp\_page\_template attribute when creating a post.

Attack narrative

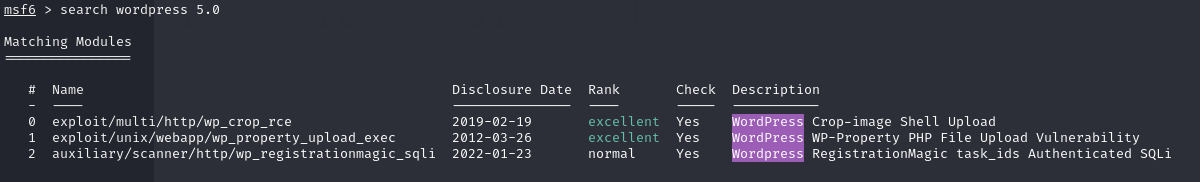
Exploitation

We will configure the exploit via framework metasploit. To get started metasploit and you search for the exploit, then you select it.

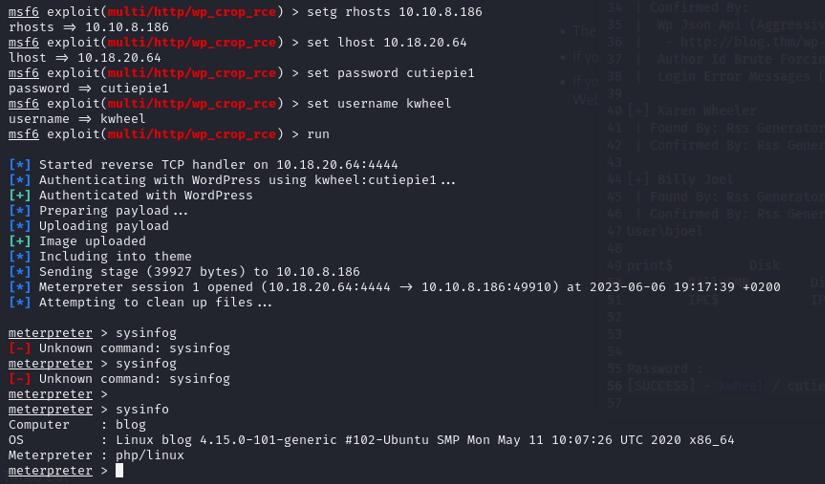
Command:

Run metasploit: msfconsole

Search the exploit: search wordpress 5.0

 Select the exploit: use 0

Once the exploit is selected, enter information such as the RHOST (remote host) password and username you found during the attack on the WordPress. To finish the lhost (the local host) so your ipv4 address and enter the run or exploit command.



Attack narrative

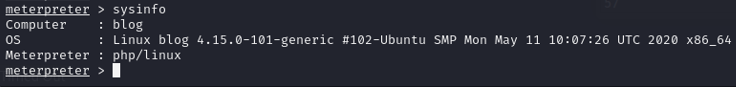
Post exploitation

So we have our feet on the server, it’s time to see the important information listed before we start our privilege elevations. We are on a ubuntu server with the user www.data, a user with some privileges.

meterpreter commands:

our user: getuid

type of operating system: sysinfo



Our quest to get the root user starts, after having mentioned the SUID of the machine (SUID means "Set User ID", and this is a special type of permission that can be given to a file so that the file is always executed with the permissions of the owner instead of the user running it. This is necessary for many programs to work properly in UNIX.) A suid in particular stands out because it is not usual. The name of this binary is checker, this suid when it is running just tells us that we are not by admin. This is where we will go to become root of the server.



Attack narrative

Post exploitation

To start, let’s look at what this binary does. Open a Linux shell via meterpreter and mount it as TTY shell for more interaction with the system. We’ll see that this binary fact calls has a variable called admin. This variable does not exist in our environment. What happens if we create an admin variable with its corresponding value. So we’ll test this.

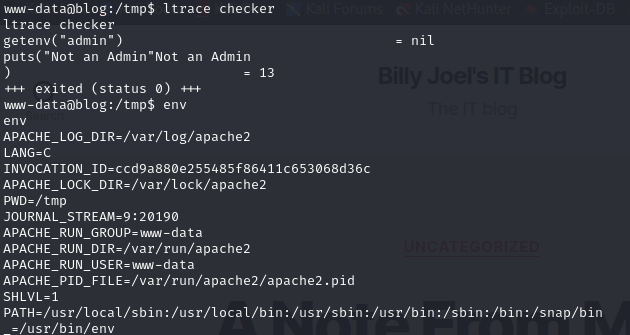
Linux command:

Open a linux shell: shell

Upgrade the shell: python –c ‘import pty; pty.spawn (“/bin/bash”)’

Binary analyse: ltrace checker

Environement listing: env



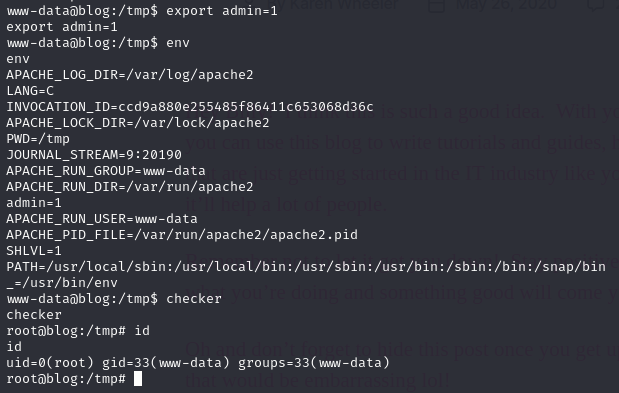
Attack narrative

Post exploitation

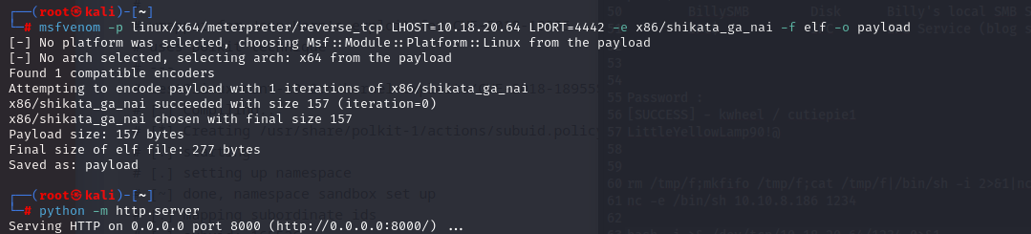
Binary exploitation will be relatively fast. Once we have created an environment variable named admin and we execute the binary the root rights are assigned to us. But our root right is only available when the binary is running so we need to create a reverse shell is the run with the root right.

Linux commands:

Create admin variable: export admin=1



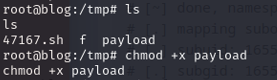
We will create a reverse shell payload with msfvenom and configure a listen pending the connection of our reverse shell payload. After that we start a python web server to download the exploit on the sever.



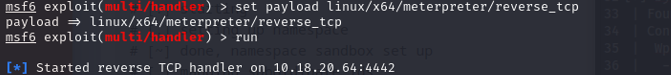
Attack narrative

Post exploitation

Once the exploit creates download it on the remote server preferably on the/tmp directory. Before executing the payload it must be given the right x so that it can be executed.

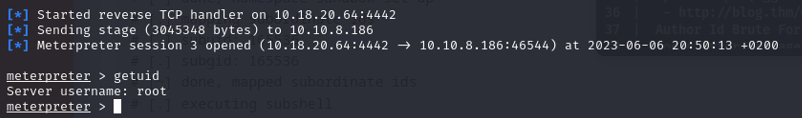


You must also configure the listening with metasploit and its multi/handler module.

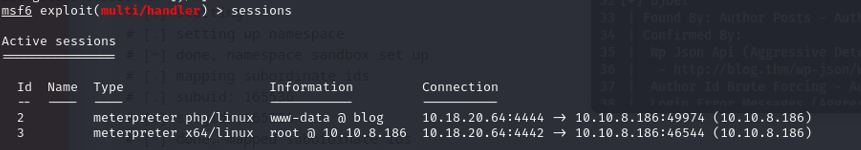


Once this is done you are ready to launch the exploit.





our active session:



Attack narrative

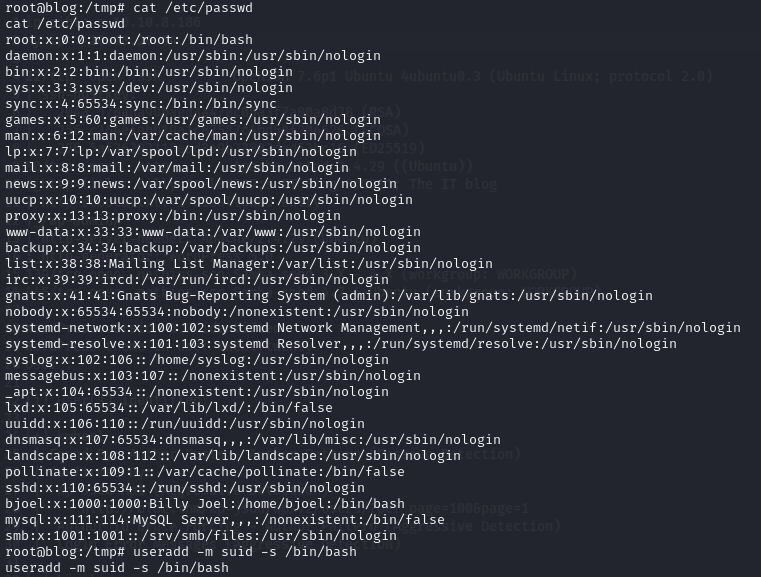
Post exploitation

We have our meterpreter session one with the root user and another with the www.data. We will be able to pass to the last stages of intrusion testing install a persistence to have access with the maximum of privileges when we want.

To do this we will use the ssh protocol to connect to the server.

But for a discreet maximum we will create a user, which we will hide and assign him high privileges.

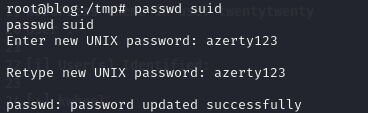
Start creating the user, it will be called suid. Before that we will look at the account of the/etc/passwd file to verify that our user is adding.



Attack narrative

Post exploitation

We will now give him a simple password to remember as part of the test this will be azerty123.



To finish creating the user we will add it to the root groups. You can now look at the account of the/etc/passwd file and see our user.

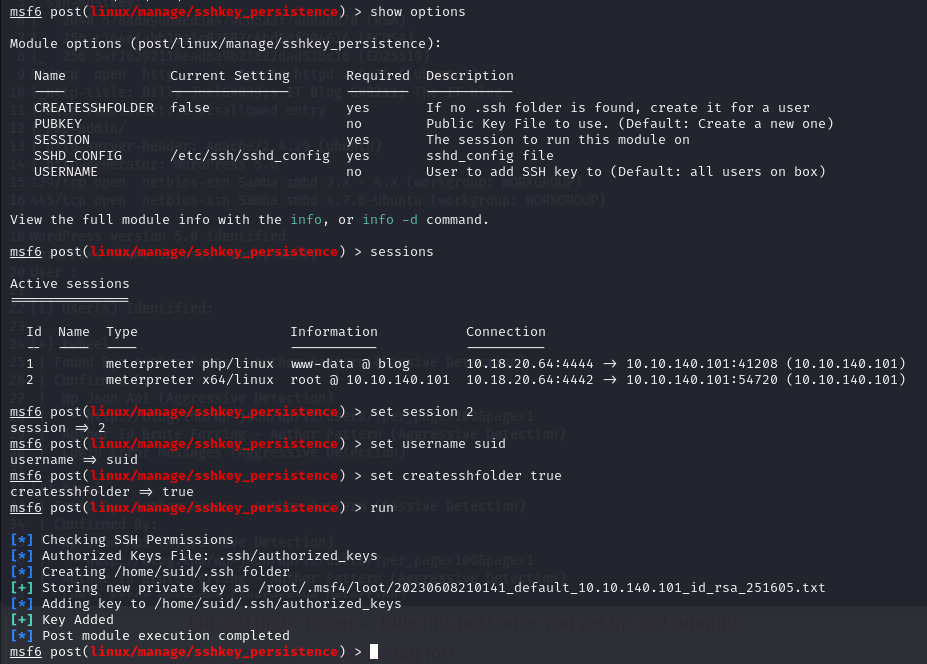


Attack narrative

Post exploitation

To configure persistence via Ssh, we will use a metasploit module called sshkey\_persistence, which will add an SSH key to a specified user (or all), to allow emote connection via SSH at any time.

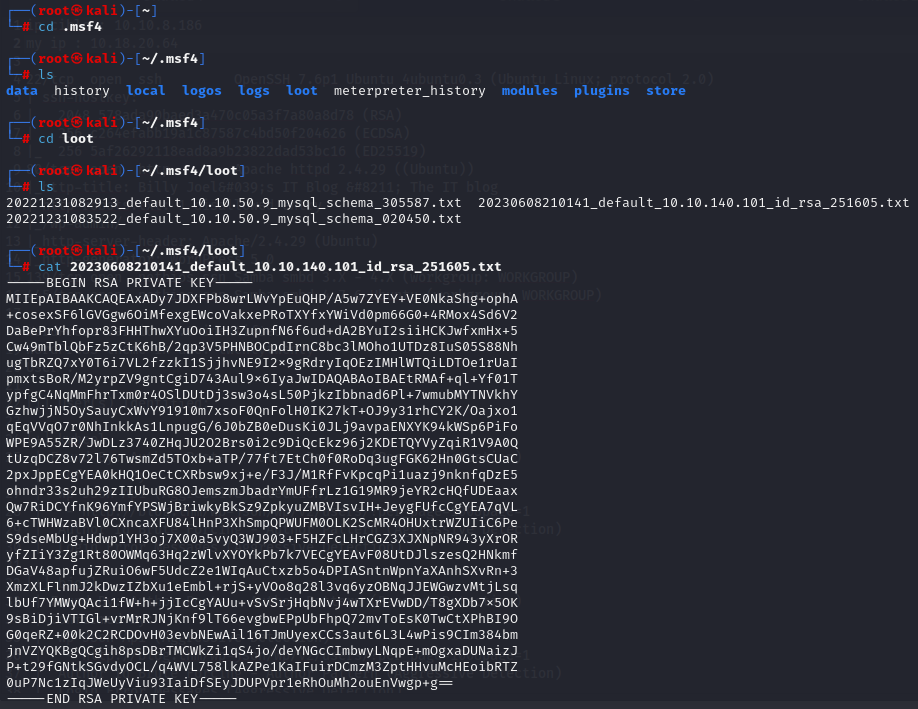
The configuration of the exploit is simple it enter the session meterpreter with high privileges, the name of the user you created, the option createsshfolder is not obligated and run the exploit.



Attack narrative

Post exploitation

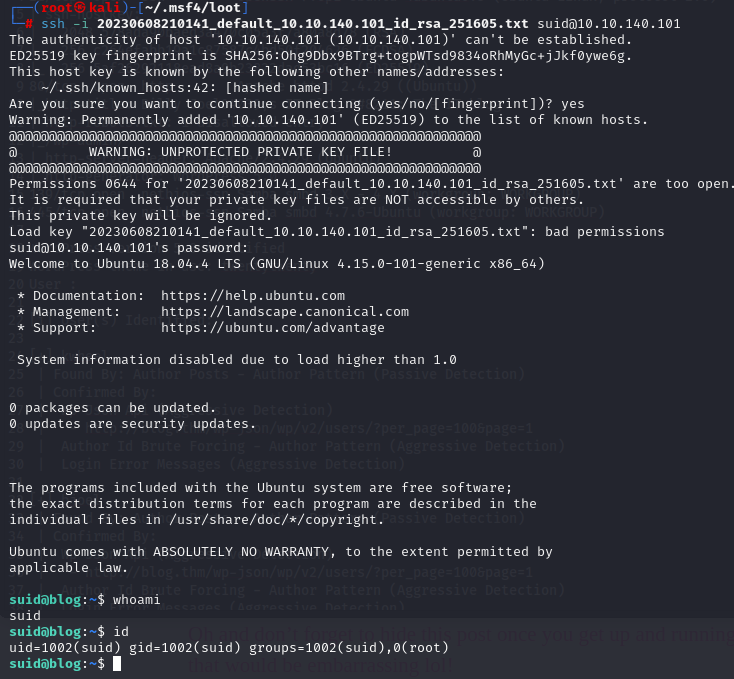
Our private server access key via ssh is available in metasploit databases if you have started it or in the/root/directory. msf4/loot/id\_rsa.txt.



Attack narrative

Post exploitation

Now let’s test our Ssh access to the remote server



We are well connected with our user on the remote server. We are now completely in control of the server.

Conclusion

Recommendation

This test was carried out successfully and led to the total compression of Alfred’s information system. Billy Joel will need to protect himself from potential attacker following some recommendations.

Anh4ckin3 proposes the following recommendations:

* **Implemented a password on your SMB server:** currently, the SMB server is accessible anonymously. This has had no influence on the progress of the operation, but it may become problematic if users start storing important data (such as passwords) on the SMB server. It is therefore necessary to implement an access password, or even encrypt the shared folder.
* **Implement an IPS in your local network:** IPS (Intrusion Prevention System) is a security system that, in addition to detecting intrusions, takes active measures to prevent attacks and protect the network by blocking or filtering suspicious traffic. This will make the active recognition of an attacker much more difficult. There are free ones like Crowdsec which is Open source is at the forefront of user expectations.
* **Upgrade your version of WordPress and implemented a strong password on the users account:** To start with, you need to set up a password policy in your WordPress. This will encourage users to configure strong passwords, which can block brute-force attacks. Finally, you must update your version of WordPress at all costs, as it is vulnerable to an RCE-type exploit.
* **Secure the code of the vulnerable SUID binary:** the binary checker is badly configured and allows an attacker to raise his privileges to the maximum. To avoid this, you'd need to either re-configure the binary, or disable it altogether. What's more, the binary isn't really useful to the system.

Conclusion

Risk ranting

The overall level identified on Billy Joel's server as a result of the penetration test is low, medium and high. A direct path from external attacker to full system compromise has been discovered.

It is reasonable to assume that a malicious entity would be able to successfully execute an attack against Alfred through targeted attacks.

Vulnerability Detail and Mitigations

**Scanning:**

*Rating*: Medium

*Description*: Find information about the target via the local network.

*Impact*: an attack manages to visualize, understand and project itself in its future actions that could lead to the compromise of the system.

*Remediation*: implement an IPS in the local network that will block suspicious traffic and anticipate a potential risk of attack.

**Smb server access:**

*Rating*: Low

*Description*: the attacker gained access to the samba server using an anonymous login.

Impact: the attacker has access to the smb server and can therefore read, modify or delete data.

*Remediation*: implement a password to access the smb server.

**WordPress exploitation:**

*Ranting*: High

*Description*: Thanks to WordPress an attacker can execute remote code on the server.

*Impact*: thanks to this vulnerability, the attacker can access the server via a reverse shell connection.

*Remediation*: To prevent this from happening, you need to update the current version of WordPress and change the passwords used to access the WordPress administration page.

Vulnerability Detail and Mitigations

**Escalate to root thanks to SUID:**

*Risk ranting*: High

*Description*: an attacker can try to play with a poorly configured binary to increase his privileges.

*Impact*: If the attacker succeeds in exploiting the binary, he gains access to the system's most privileged root user.

*Remediation*: Reconfigure binary or even Deactivate.

**Persistence with ssh:**

*Risk ranting*: High

*Description*: With access to the local root account, an attacker can initialize constant access to the machine using the ssh protocol.

*Impact*: Your server is totally compromised and the attacker can go even further by implementing a key logger, for example.

*Remediation*: If the attacker has arrived there, it may be difficult to detect. You'll need to pay close attention to the server's behavior, and if you have any doubts, call in forensic experts and post-incident analysts.