# <https://app.hackthebox.com/profile/1321956>

# Machine 1: Precious Type: Easy

1)ping to machine IP address

Text

Description automatically generated

2)Conduct a Nmap scan

-Sv determine the version of the service running on port  
-V verbosity level  
-SC Scan with default NSE script

Text

Description automatically generated

Text

Description automatically generated

Extracted from the scan:

Text

Description automatically generated

Redirect to another link, <http://precious.htb>

3) Modify host file.  
nano /etc/hosts



Include the Ip address and domain name and saved.

Text

Description automatically generated

4)Copy address link to http://precious.htb/

Graphical user interface, text, website

Description automatically generated

5) Start a Http server



6) Type ifconfig on another terminal to check on the inet address.

Graphical user interface, text

Description automatically generated

7)Key in Http://10.10.16.17.8080

Graphical user interface, website

Description automatically generated

This will alert the http server and the request to retrieve the document is successful .

Graphical user interface, text, application, chat or text message

Description automatically generated

It will prompt user to save the PDF document .

Graphical user interface, text, application

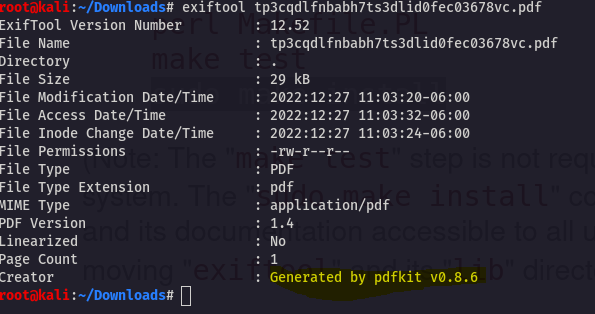
Description automatically generated

I retrieved the directory listing of the webserver

Graphical user interface

Description automatically generated with medium confidence

8) I noted the application that creates the PDF and the version numbner.Use these information to find out for potential exploits.



9) Search pdfkit v0.8.6 for potential exploits in a web browser. It is prone to command injection.

Research

Graphical user interface, text

Description automatically generated

Text

Description automatically generated with medium confidenceGraphical user interface, text, application

Description automatically generated

With command below ,I will check if the webserver is prone to command injection

Graphical user interface, text, application, email, website

Description automatically generated

Using the following code ,I received a response from the Http server. I can confirm that the webserver is prone to command injection

Graphical user interface, text

Description automatically generated

Text

Description automatically generated with low confidence

Yes. It is prone to command injection as its was captured in the python http server.

10) Creating a reverse shell

Graphical user interface, text

Description automatically generated

11)Whoami>Ruby (interacting as Ruby)

Graphical user interface, text

Description automatically generated

There are 2 users ; Henry and Ruby

Graphical user interface, text

Description automatically generated

12) Elevate the privileges as Henry. I check the listings in the Ruby’s directories

A screenshot of a computer

Description automatically generated with medium confidence

Directories

13)Use find command to list every file on this directories

Graphical user interface, text, website

Description automatically generated

There is a file named config located under the bundle directory. “Cat” the content to see the information inside.

Graphical user interface, text, website

Description automatically generated

14)It displays Henry’s password. With Henry’s password I can elevate the privileges from Ruby to Henry.

Graphical user interface, text

Description automatically generated with medium confidence

15) Change privileges to Henry. Currently I’m accessing the system as Henry.

Text

Description automatically generated

Under henry there are 2 files: dependencies.yml and user.txt

Graphical user interface, application

Description automatically generated

16)Read the content in the user.txt.  
e210c900a319bde77759ba8ff7a6d74f



17)Checking on Henry’s pseudo privileges. It seems that we can run a ruby script named “update\_dependencies.rb”.This does not require any password.

Text

Description automatically generated

The script uses a function called YAML.load(File.read(“dependencies.yml”)). This method is vulnerable to deserialization attack.

Text

Description automatically generated

18)Research on YAML.load method

Graphical user interface, text

Description automatically generated

It provides a payload that we can use in Github.

Graphical user interface, text, application

Description automatically generated

19)Copy the codes for the payload execution.

Graphical user interface, text, application

Description automatically generated

20)Checking the read, write access for the different access groups

Text

Description automatically generated

Text

Description automatically generated

21)Changing the mode to ROOT.

Text

Description automatically generated

Text

Description automatically generated

root.txt=0d84d5b477e030dd198a0582ac575ecc

A screenshot of a computer screen

Description automatically generated with medium confidence

# References:

<https://book.hacktricks.xyz/pentesting-web/nosql-injection>

<https://gtfobins.github.io/gtfobins/docker/#shell>

<https://gtfobins.github.io/>

<https://www.techrepublic.com/article/how-to-quickly-give-users-sudo-privileges-in-linux/#:~:text=To%20use%20this%20tool%2C%20you,%2Fsudoers%20file%20for%20editing).&text=Save%20and%20close%20the%20file,full%20range%20of%20sudo%20privileges>.

<https://book.hacktricks.xyz/pentesting-web/nosql-injection>

<https://www.imperva.com/learn/application-security/nosql-injection/#:~:text=The%20main%20differences%20between%20NoSQL,don't%20use%20standardized%20languages>.

<https://github.com/danielmiessler/SecLists>

<https://portswigger.net/web-security/file-path-traversal>

<https://github.com/pentestmonkey/php-reverse-shell>