Nineveh

Started the enumeration on the machine using **Nmap** tool to scan for open ports and services.

Text

Description automatically generated

There seems to be two websites hosted – one on HTTP and another on HTTPS protocol.

First visited the unsecure webpage to check on possible vulnerabilities on the website.

Graphical user interface, text, application, email

Description automatically generated

The info.php sub-directory contains the backend of the website and the PHP version.

Graphical user interface, application

Description automatically generated

Then visited the secured version of the website.

Graphical user interface, application

Description automatically generated

The certificate details on the website exposes the admin’s email address and possible DNS information of the web server.

Text

Description automatically generated

As enumerated the sub-directories of the website, it exposed a directory - /secure\_notes. It contains a **Nineveh.png.**

A picture containing text

Description automatically generated

Download the same and use binwalk to extract the contents of the PNG file.

Text

Description automatically generated

Once extracted, there are two files which seems to be a RSA file for logging into SSH service but the SSH service was closed when we ran the **Nmap** scan. Hence let us enumerate more onto the web application.

There is also another sub-directory listed in our **Gobuster** tool results.

Text

Description automatically generated

The source code of the sub-directory web page shows that the Login page on **/department** sub-directory needs to be fixed.

Graphical user interface, text, application

Description automatically generated

We used Burpsuite to alter the HTTP response of an invalid username/password and used the same payload on **Hydra** to brute force it and break it.

Finally we get the login credentials of the /**department** sub-directory.

Text

Description automatically generated

Once logged in we can see the site is under developing. The notes section of the website shows that the username/password was hardcoded and the SQL DB is installed.

It also mentions the secret folder which we found in the **Nineveh.png** file. It has the secret keys(private key) to log in.

Graphical user interface

Description automatically generated with medium confidence

As we found another Login page for PHPLiteAdmin v1.9 which only had a field for inserting password. Assuming the username be **admin**, we cracked the password with the same technique used above with **Burpsuite**.

Graphical user interface, application

Description automatically generated

Text

Description automatically generated

Once logged in, we could see there is already a test db being created and we have full functionality of the PHPLiteAdmin website. We will be further exploiting the machine using this website.

A screenshot of a computer

Description automatically generated

There is a specific exploit for the above PHPLiteAdmin software which is vulnerable to Remote Code Execution (RCE) and can be found at - <https://www.exploit-db.com/exploits/24044> .

Use the link above to follow the steps and create a database, a table and insert a row with specific PHP command which then be exploited further.

At first, I created a php file database with name -hack.php but it did not work.

Graphical user interface, website

Description automatically generated

Background pattern

Description automatically generated with medium confidence

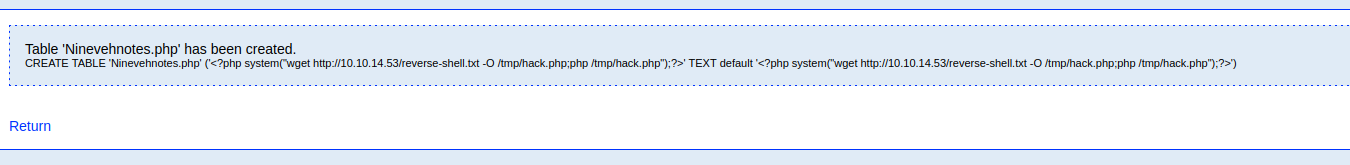
After multiple attempts it occurred to me that the Ninevehnotes.txt is blocking any other file to be visited using LFI vulnerability of that webpage as shown above.

Hence created another database with the same name as above and created a table with the same name – **Ninevehnotes.php**

Graphical user interface, text, application

Description automatically generated

Then inserted a row in a way that the reverse-shell be uploaded to the website from our local machine using Python.



The below python script will host a HTTP channel on our local machine and our code on the webpage has **wget** command included which will connect to our local machine and retrieve the files and insert it into our newly created database.

Graphical user interface, text

Description automatically generated

Graphical user interface, application, website

Description automatically generated

Now let’s access the database fie we created via web browser and simultaneously open up a listener on our local machine for the reverse shell to be created.

Graphical user interface

Description automatically generated

Finally, we get the reverse shell onto our local machine. Stabilize the shell using the python script.

As we enumerate more on the server, there is a mail folder which doesnot have much information.

Text

Description automatically generated

Lets try to find the privilege escalation vector for getting root access, this can be enumerated using **linpeas.sh** script which gives all the information of the machine and possible privilege escalation vectors.

Graphical user interface

Description automatically generated

Download the **Linpeas.sh** script onto the server using **Wget** and **Python** script.

One of the results of the script show that the machine us vulnerable to the below mentioned CVE’s.

Text

Description automatically generated

Researched about the above CVE’s and found CVE 2017-16695 a suitable one to used and exploit the machine.

Followed the steps on the website and executed the same to get the root access on the machine.

Text

Description automatically generated

Finally we get the root.txt file and associated root flag for the machine.

Text

Description automatically generated with medium confidence