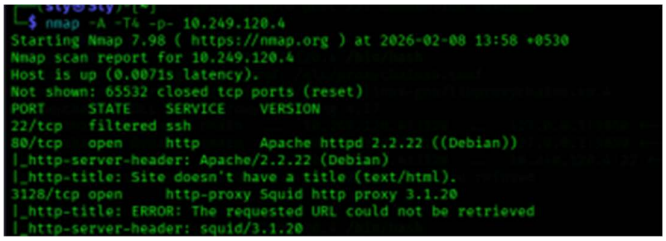
Objective:
Gain access to the flag.txt file inside the /root/directory

Identifying the target system:

- Using ‘netdiscover’ and obtain the IP.
  Command: sudo netdiscover -P -r <IP>
- Ping the device to see if it responds

Step 1 -> Perform a scan for the active ports on the device.

Command:
nmap -A -T4 -p- <Target_IP>

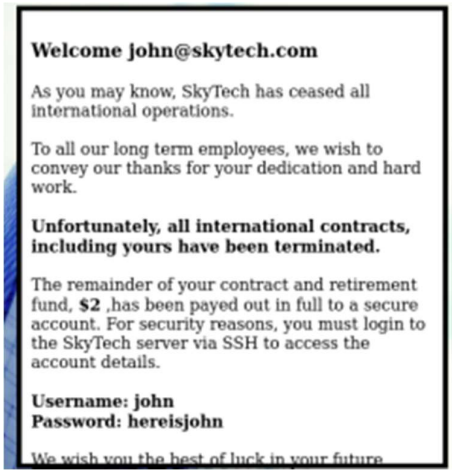


Findings:

- We identify Ports 22, 80, and 3128 are open on the device.
- Port 22 has ssh enabled and is filtered
- Port 80 is running an Apache server
- Port 3128 is running a proxy server

Step 2 -> Access the server on Port 80 through the browser, this will bring up a login page.

- Perform an SQL injection to obtain access to the server.
- Use ‘-’ in the uname and password fields to gain access.



Findings:

- We have now accessed the server through the user ‘John’ and have the account login details for the SkyTech server.
- We can access John's account on the SkyTech server through ssh
- However, the nmap scan revealed Port 22 had ssh enabled, however, the ssh is filtered and hence we will not have direct access to ssh as a particular user.
- Additionally, Port 3128 is running an http proxy, we can now connect through ssh using the http proxy running on the port.
- To do this we use the proxychains4 service.

Step 3 -> Configuring and establishing proxychains4

- We are attempting to ssh into the http proxy hosted on Port 3128 proxychains4.
- Configuring proxychains4 to specify the use of this service prior to establishing the ssh connection.
- Configuration can be done through the /etc/proxychains4.conf file.
- Specify the type of proxychain and the Target machine's IP and port.
- Enable the Tor service through the command ‘service tor start’ and ‘service tor status’
- The config file for the service must be updated with the Target system's IP and Port number that is running the proxy server.

Note:

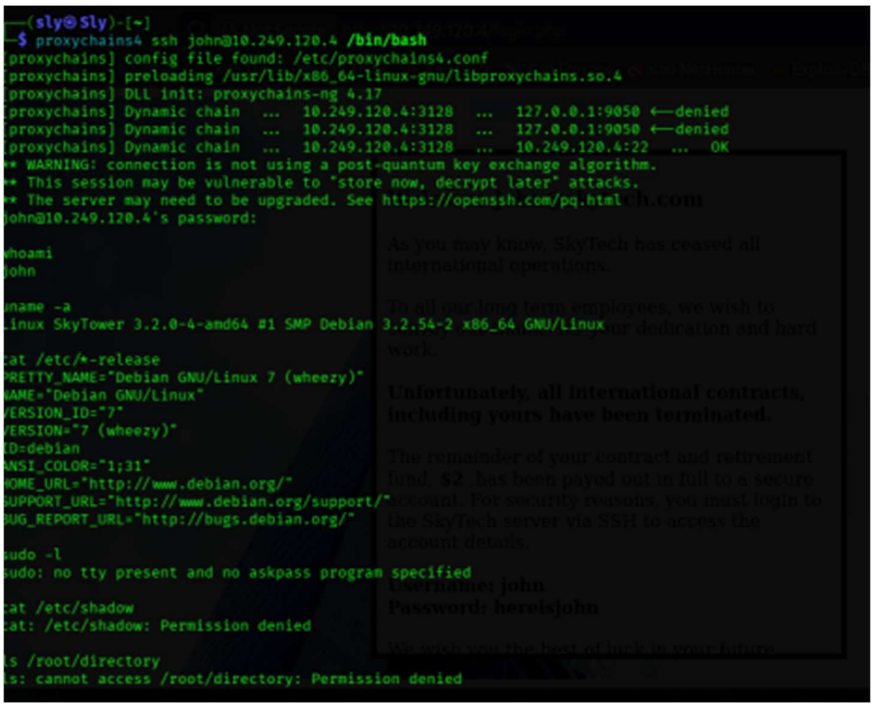
Using the tor service through Proxychains4:

- Proxychains4 is a powerful Linux utility that forces network traffic from any TCP application through a chain of proxy servers (SOCKS4, SOCKS5, HTTP) for enhanced anonymity and to bypass network restrictions, routing traffic through multiple proxies like Tor or others to hide your real IP address and obfuscate activity
- *It works by intercepting TCP calls and redirects them, allowing tools without native proxy support (like nmap, telnet to use proxies, configurable via /etc/proxychains4.conf to define proxy lists and chain modes (strict, dynamic, random) for tasks in penetration testing or general privacy.*

Step 4 -> Connecting to the server via ssh through proxychains4

- John's account details can be used to gain access to the server
- Spawn a bash shell once the connection is established. (/bin/bash)

Command:
proxychains4 ssh john@<Target_IP> /bin/bash



Findings:

- We have now established partial access to the file system as John is not a privileged use.
- John's account does not have any access to using ‘sudo’ commands. (sudo -l)
- The ‘root’ directory is not accessible as it requires admin privileges, hence we must look for another user that could have the required access permissions.

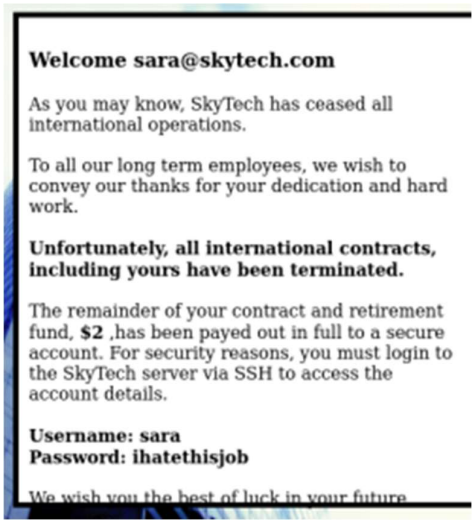
Step 5 -> Navigating the File system to find additional user details

- As John, we are limited to only a handful of files and directories that do not require admin privileges, hence we must find a different user that we could exploit.
- The Home directory displays a list of 3 users in total. (John, Sara, and William)
- We are unable to access the files for Sara and William while logged in as John.

- This brings up the need to find account details for either ‘Sara’ or ‘William’.
- From John's login page earlier, the email format for the users on the SkyTech server was username@skytech
- We can try accessing the user's details by using the email ‘sara@skytech.com’

- Through SQL injection techniques we can gain access to Sara's account and obtain the password.

Method: sara@skytech.com'# (Into the username filed)



Step 6 -> Logging into the http proxy through Sara's credentials.

Command:
proxychains4 ssh sara@<Target_IP> /bin/bash

```
┌──(sly㉿Sly)-[~]
└─$ proxychains4 ssh sara@10.249.120.4 /bin/bash
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Dynamic chain  ...  10.249.120.4:3128  ...  127.0.0.1:9050 ←denied
[proxychains] Dynamic chain  ...  10.249.120.4:3128  ...  127.0.0.1:9050 ←denied
[proxychains] Dynamic chain  ...  10.249.120.4:3128  ...  10.249.120.4:22  ...  OK
** 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
sara@10.249.120.4's password:

whoami
sara

id
uid=1001(sara) gid=1001(sara) groups=1001(sara)

sudo -l
Matching Defaults entries for sara on this host:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User sara may run the following commands on this host:
    (root) NOPASSWD: /bin/cat /accounts/*, (root) /bin/ls /accounts/*

cd /root/directory
/bin/bash: line 8: cd: /root/directory: Permission denied

uname -a
Linux SkyTower 3.2.0-4-amd64 #1 SMP Debian 3.2.54-2 x86_64 GNU/Linux

pwd
/home/sara
```

Findings:

- Sara's account has highest access privileges than of John's.

- While Sara's account does not have complete ‘sudo’ privileges, we identify that her account can access certain files through a specified format.
- This is established through the ‘sudo -l’ command that specifies Sara can access certain files as the root user through passing commands with the syntax:

  Command:
  sudo ls /accounts/../root

- The flag.txt file is located within the root directory; to access the file we use the command:

  Command:
  sudo cat /accounts/../root/directory/flags.txt