HTB Cap Written

Summary

Objective: Find user and system flags
Areas of vulnerability: URL information leak, same password, root-owned Python binary
Recommendations:

- Omit URL objects/enumeration
- Enforce password diversity (web accounts must not share the same password as system)
- Implement password hashing
- Put in protections for downloading .pcap files
- Use STFP or SSH instead of FTP
- Do not let root own Python binaries

Tools used:

- `nmap`
- Wireshark
- linPEAS

Findings

1. Information disclosure in website URL (`/data/<id>`)
   Affects: `nathan`
   Found by scanning through the HTML page exposed by `nmap` scan.
   With this, you're able to infer and page through other data items (0-12)
2. Password leak guessing
   Affects: `nathan`
   Found password by going through .pcap file and reading FTP (unencrypted) and tried to use it for SSH as well
   With SSH, can operate as `nathan` and get better view of file system
3. SUID Privilege Escalation
   Affects: system
   Using linPEAS, found a Python binary owned by `root` in usr. `nathan` is able to execute, so use it to change UID to root

Walkthrough

User

First, start off with a scan of the IP.

```
nmap 10.10.10.245 -Pn
```

```
# ------ TRUMCATED OUTPUT ------
# PORT   STATE SERVICE
# 21/tcp open  ftp
# 22/tcp open  ssh
# 80/tcp open  http

# ----------- NOTES ------------
# initially did not do it with -Pn and it showed firewall
# should have used -vv just in case
```

We see that port 80 is up, so there might be a website. Visiting the IP, we look around and find that under "Security Snapshot (5 Second PCAP + Analysis)" that there are PCAP summaries, a download button, and a URL bar that looks like it would be able to enumerate.
Downloading and inspecting the 0th .pcap file, we find that there are HTTP and FTP requests. One of the FTP requests are a username and password pair:

```
USER nathan
PASS Buck3tH4TF0RM3!
```

Trying this to SSH succeeds and we find the user flag at `user.txt`.

System

Now that we have control over the system, we can do a scan for vulnerabilities in the file system with linPEAS (needs to be downloaded). This reveals a vulnerability with the permission configurations for a binary in `/usr/bin/python3.8`. The Python3.8 is owned by root, which means we can escalate privileges:

```
import os               # import the os module
os.system('whoami')     # check the current user
os.setuid(0)            # set the uid to root
os.system('whoami')     # check that the uid has been set correctly
os.system('sh')         # spawn a shell to continue as root
```

In the root folder, finding the flag file and opening it reveals the flag.