HELLO!

I am Hans-Martin Münch

I am here to talk about JMX...
1. Why bother??
Most Penetration Testers know this...

JBoss JMX Console

Tons of tools to pOwn the system

Happy hacker !!!
JEE services with JMX support

- Tomcat
- JBoss
- Jetty
- Sonatype Nexus
- SAP Netweaver
- Active MQ
- Hazelcast
- Weblogic
- Citrix Command Center
What is JMX?

...or the "JMX elevator pitch"...
Java Management Extensions (JMX) is a Java technology that supplies tools for managing and monitoring applications, system objects, devices (e.g. printers) and service-oriented networks.
Simplified:

JMX is SNMP on steroids for Java Applications
3.

JMX Fundamentals

It is all about fundamentals, fundamentals, fundamentals.
Let's start with Beans...
Managed Bean (MBean)

What it is:
- The stuff that you manage via JMX (Ressource)
- The model in MVC concept
- Just a Java class

Class must follow some Rules:
- Implement a interface
- Default Constructor (No parameters)
- Naming Conventions
Mbean Example - Interface

public interface HelloMBean {

    // Attribute "name"
    public String getName();
    public void setName(String newName);

    // Methods
    public String sayHello();
}
MBean Example - Code

```java
public class Hello implements HelloMBEan {

    private String name = "OWASP Munich";

    // Attribute "name"
    public String getName() { return this.name; }
    public void setName(String newName) { this.name = newName; }

    // Methods
    public String sayHello() { return "hello: " + name; }
}
```
MBean-Server

- Service for MBean management
- Registration of MBeans
- Forwards messages to MBeans
- Forwards events from MBeans to external components

Registration/Access:
- Requires a distinct name (like a URL)
- Format: Domain-Name:Key/Property
- Example: De.MogwaiSecurity:Type=OWASPDemo
Mbean Server - Code

// Get local mbean server
MBeanServer mbs = ManagementFactory.getPlatformMBeanServer();

// Create a name and MBean Instance
Hello owaspBean = new Hello();

ObjectName mbeanName = new ObjectName("de.mogwaisecurity:type=OWASPBean");

// Register the name and MBean at the local server
mbs.registerMBean(owaspBean, mbeanName);
**JConsole**

- **Graphical tool (Part of the JDK)**
- **Really useful 😊**

**What it can do:**
- Connect to a Mbean/JMX server
- Graphical Bean overview
- Local – via process ID
- Remote – via Java RMI
JMX Connectors

- Provides remote access to a MBean Server
- Basically a Client-/Server Stub
- No real difference between local/Remote communication
- You can change transfer protocols (HTTP/Morse code/)

Diagram:
- JMX Client
  - JMX Connector Stub
  - JMX Runtime
  - MBean Server Connection
  - JMX Connector
  - Broker
    - MBean Server
    - MBeans
JMX Connectors

- Normally Java RMI (Remote Method Invocation) is used
- Enabled via command line parameters

Example with no Authentication:

-Djava.rmi.server.hostname=192.168.0.32
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=8888
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.authenticate=false
JMX Adaptors

- Similar to JMX Connector
- But - Provides what the client expects (for example HTTP)
- No "Client Stub"
- You can't use everything (like complex Java objects)
- Commonly used by non-Java Software
DEMO

JMX on Tomcat 7
Tomcat 7

JMX Connector via RMI
× enabled at Java start via command line parameters
× Example for Debian: /etc/defaults/tomcat7

JMX http adaptor a.k.a. Proxy Servlet
× Part of the Tomcat admin application
× Requires dedicated user role (manager-JMX)
× /manager/jmxproxy
No Authentication?

Secure your RMI connections:
- TLS/SSL Encryption
- Username/Passwords
- Support for roles/groups, for example readonly access
- Most installations use only one account
Attacking JMX

...give me some shells bro...
JMX Pownage through Mlet Loading

* „Discovered“ by Braden Thomas (Accuvant – now optiv)
* He reads documentation 😊

Braden Thomas

*Senior Research Consultant*

Braden Thomas is a senior research consultant with Accuvant LABS’ research consulting practice. Braden has expertise in vulnerability discovery, fuzzing, exploitation techniques, malware analysis and protocol analysis.
A remote client could create a `javax.management.loading.MLet MBean` and use it to create new MBeans from arbitrary URLs, at least if there is no security manager.

In other words, a rogue remote client could make your Java application execute arbitrary code.
1. Invoke Loading.Mlet
2. Loading.Mlet parses Mlet Configuration file (HTML)
3. Loads and instances Mbean from Mlet File => Attacker Code
4. Attacker invokes malicious MBean
I Wrote a Tool for that...

MJET – Mogwai JMX Exploitation Toolkit

Two parts:

× Metasploit-module (Mlet–Webserver, Payloads as MBeans)
× Java-Part (RMI/JMX communication)

You can download MJET from my GitHub account...
...But you can just use Metasploit

Juan Vazquez added RMI/JMX support to Metasploit including an exploit for insecure JMX Services/Mlet loading

...Awesome wtf work...
DEMO

Exploiting JMX via Metasploit
Is this common?

No

JMX over RMI is not enabled by default

Only works if authentication is disabled

Yes

Monitoring gets more important. Maybe someone forgot to enable Auth.

JMX is part of Java, not a product.

After all you only need to succeed once..
Is this common?

Let's ask searchcode

- 598 results
- Many "test" scripts...
Monitoring with JMX

You can monitor your Hazelcast members via the JMX protocol.

- Add the following system properties to enable JMX agent:
  - `-Dcom.sun.management.jmxremote`
  - `-Dcom.sun.management.jmxremote.port=\_portNo\_ (to specify JMX port) (optional)`
  - `-Dcom.sun.management.jmxremote.authenticate=false (to disable JMX auth) (optional)`

- Enable the Hazelcast property `hazelcast.jmx` (please refer to the System Properties section):
  - using Hazelcast configuration (API, XML, Spring).
  - or by setting the system property `-Dhazelcast.jmx=true`

- Use jconsole, jvisualvm (with mbean plugin) or another JMX compliant monitoring tool.
What authentication is enabled?

- You need to find/brute force credentials
- Loading Mlets is no longer possible if authentication is enabled
- You can still use the available Mbeans
- After all you are talking to a Management Interface
Demo
Exploiting tomcat via JMX methods
What authentication is enabled?

Tomcat example:
× No Mbean to deploy Remote Web Applications (not like JBoss)
× But you can add users and Groups 😊
5. Detection

...Finding JMX services...
Detecting JMX Endpoints

Nmap detects JMX RMI Services as normal Java RMI Services

nmap -sV 192.168.178.236 -p 1099

Starting Nmap 6.47 ( http://nmap.org ) at 2015-09-09 23:19 CEST
Nmap scan report for 192.168.178.236
Host is up (0.00060s latency).
PORT    STATE SERVICE VERSION
1099/tcp open  rmiregistry Java RMI

Nmap done: 1 IP address (1 host up) scanned in 20.46 seconds
Detecting JMX Endpoints

Some Nmap hints

- Default: RMI detection works only on common ports
- Use Option "--version-all"
- Use nmap script "rmi-dumpregistry.nse", search for "JMXRMI"
- To be safe: use jconsole
nmap --script rmi-dumpregistry.nse -sV --version-all -p 1099 192.168.178.236

Starting Nmap 6.47 ( http://nmap.org ) at 2015-09-09 22:21 CEST
Nmap scan report for 192.168.178.236
Host is up (0.0015s latency).
PORT STATE SERVICE VERSION
1099/tcp open java-rmi Java RMI Registry
  rmi-dumpregistry:
    jmxrmi
      javax.management.remote.rmi.RMIServerImplStub
        @192.168.178.236:33701
        extends
          java.rmi.server.RemoteStub
            extends
              java.rmi.server.RemoteObject
# Vulnerability Scanners

**What about Nessus (and others)?**

## Search results:

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Family</th>
</tr>
</thead>
<tbody>
<tr>
<td>23842</td>
<td>JBoss JMX Console Unrestricted Access</td>
<td>CGI abuses</td>
</tr>
<tr>
<td>23843</td>
<td>JBoss Application Server (jbossas) JMX Console DeploymentFileRepository Traversal Arbitrary File Manipulation</td>
<td>CGI abuses</td>
</tr>
<tr>
<td>53337</td>
<td>JBoss Enterprise Application Platform '/jmx console' Authentication Bypass</td>
<td>Web Servers</td>
</tr>
<tr>
<td>70414</td>
<td>Apache Tomcat / JBoss EJBInvokerServlet / JMXInvokerServlet Marshalled Object Remote Code Execution</td>
<td>CGI abuses</td>
</tr>
</tbody>
</table>
Detecting JMX Endpoints

× Nessus detects RMI registry endpoints
× Again: Search for jmxrmi

Output

Here is a list of objects the remote RMI registry is currently aware of:

```
rmi://192.168.178.236:37666/jmxrmi
```

<table>
<thead>
<tr>
<th>Port</th>
<th>Hosts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1099</td>
<td>tcp/rmi_registry 192.168.178.236</td>
</tr>
</tbody>
</table>
5.

Summary

...what you should take away...
Summary

- JMX endpoints are often admin-Interfaces
- Comes in many flavours (RMI/HTTP)
- Often allow remote code execution
- Might be missed by the security team
- We need better tools...
We need better tools...

Metasploit provides native support for RMI, just look at the existing modules for examples....

Some ideas:
- Auxiliary module for RMI/JMX detection
- Loginscanner for RMI Brute force attacks
- Auxiliary modules to add/extract tomcat users via RMI

A nmap or nessus nasl script would also be handy 😊
THAT’S ALL!

Any questions?

You can find me at
@hOng10
muench@mogwaisecurity.de
https://www.mogwaisecurity.de
References

- Wikipedia Article about JMX
- Authentication and Authorization in JMX RMI connectors
  https://blogs.oracle.com/lmalventosa/entry/jmx_authentication_authorization
- Accuvant blog - Exploiting JMX RMI
  https://www.accuvant.com/blog/exploiting-jmx-rmi
- Oracle Documentation: Monitoring and Management using JMX Technology
  https://docs.oracle.com/javase/6/docs/technotes/guides/management/agent.html
- MJET - Mogwai JMX exploitation toolkit
  https://github.com/mogwaisec/mjet
Credits

Special thanks to all the people who made and released these awesome resources for free:

✗ Presentation template by SlidesCarnival
✗ Photographs by Unsplash