Information Technologies Security

Assignment

Looking for vulnerabilities in large networks

Author: Adam Žilla

ID: 92177

Contents

[Assignment clarification 3](#_heading=h.gjdgxs)

[CVE-2020-11969 3](#_heading=h.30j0zll)

[CWE-287 3](#_heading=h.1fob9te)

[Analysis of problem area 3](#_heading=h.3znysh7)

[Analysis of existing solutions 3](#_heading=h.2et92p0)

[Example Usage 4](#_heading=h.tyjcwt)

[Environment 4](#_heading=h.3dy6vkm)

[Apache 4](#_heading=h.1t3h5sf)

[ActiveMQ broker 4](#_heading=h.4d34og8)

[TomEE 5](#_heading=h.2jxsxqh)

[JMX 5](#_heading=h.2s8eyo1)

[Nmap 5](#_heading=h.3rdcrjn)

[NSE 5](#_heading=h.26in1rg)

[Python 5](#_heading=h.lnxbz9)

[Detection design 5](#_heading=h.35nkun2)

[Code snippets 5](#_heading=h.1ksv4uv)

[Bibliography 7](#_heading=h.44sinio)

# Assignment clarification

The specific task in my assignment is to look for and choose some recent CVE. Next part is to implement a simple Python script with use of NSE scripting language or some more advanced scanner like Shodan or so, in edge case I can use Metasploit framework, for detecting vulnerable devices in specified IP range.

# CVE-2020-11969

I have been looking for vulnerability I can detect via network. I chose CVE-2020-11969. Short description of this vulnerability is the following. If Apache TomEE is configured to use the embedded ActiveMQ broker, and the broker URI includes the useJMX=true parameter, a JMX port is opened on TCP port 1099, which does not include authentication. This affects Apache TomEE 8.0.0-M1 - 8.0.1, Apache TomEE 7.1.0 - 7.1.2, Apache TomEE 7.0.0-M1 - 7.0.7, Apache TomEE 1.0.0 - 1.7.5. (1)

# CWE-287

As stated in mitre Common Weakness Enumeration database, above mentioned CVE is type of Improper Authentication. IT means that an actor claims to have a given identity, the software does not prove or insufficiently proves that the claim is correct. (2)

# Analysis of problem area

I divide this analysis into two parts. First is to setup the apache server with ActiveMQ broker as a working example with useJmx set to true and JMX running on port 1099. The second part is implementation of detection as NSE script.

# Analysis of existing solutions

There are plenty of solutions to different CVE’s on github, stackoverflow or elsewhere on the internet. For my CVE there is currently no solution as it is recent vulnerability or I was not able to find one.

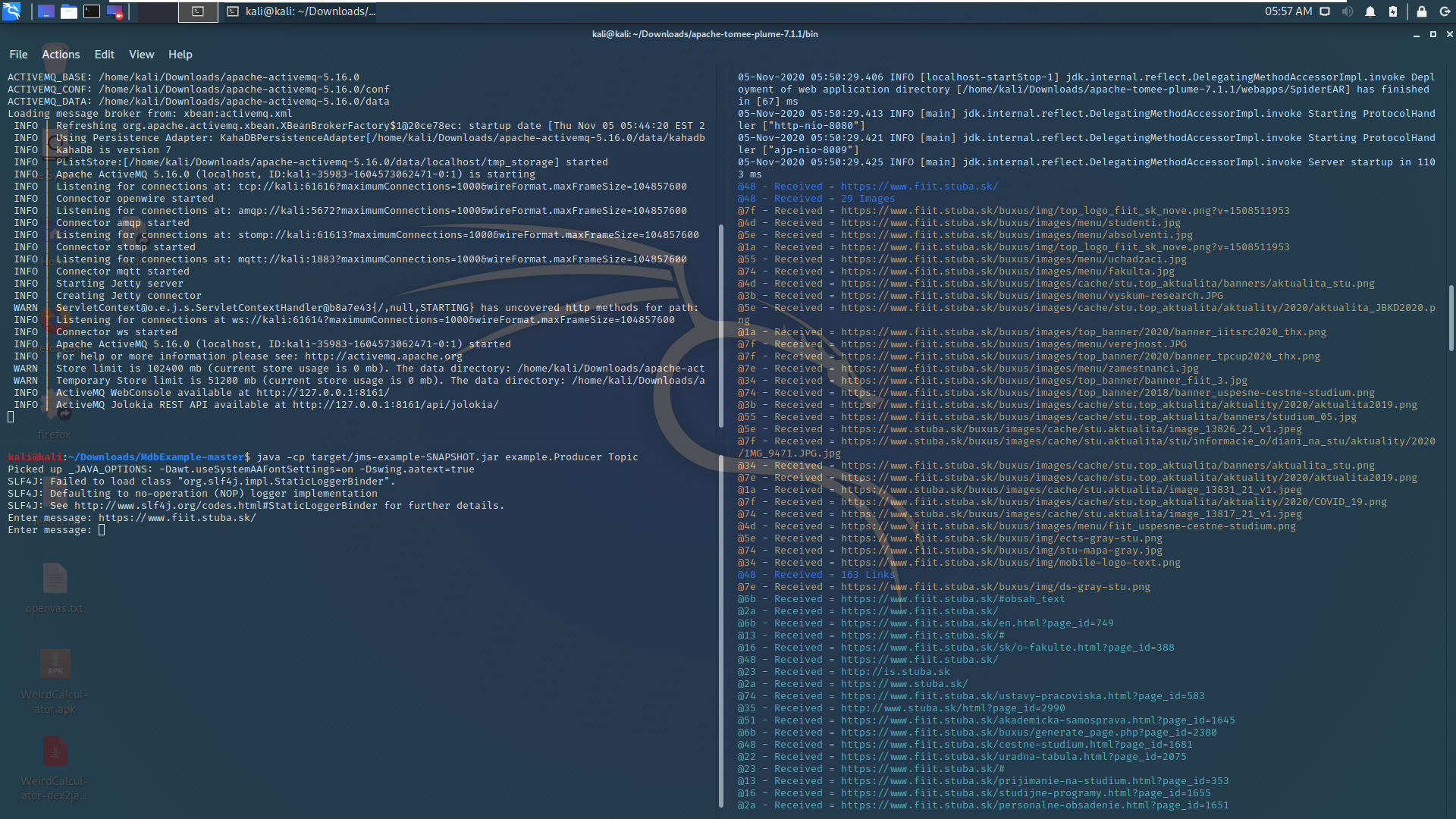
## Example Usage of broker

As example usage of such tool I chose the tutorial from website tomitribe (3). Imagine running Apache on some raspberry pi with spider tool listening. Spider or so called Web crawler is an Internet bot that crawls/browse the world wide web or the specific website for images or links like in this example.

My spider firstly browse the website for images in website. Sample website is <https://www.fiit.stuba.sk>.

In apache terminal we can see orange links, these refers to images found on fiit website. After spider is finished with images spider then begin to browse for all hyperlinks.

This is just example for how the apache can be used.



**ActiveMQ**

**Apache**

**JMS client**

# Environment

I am working in Vmware workstation, virtual machine Kali linux, where my apache server with ActiveMQ broker and MDB example are running. My public git repo can be found on <https://github.com/itchybunion6235/BIT_project>

# Apache

Following chapters contains some introduction to problematic. What are individual modules in our apache and so on.

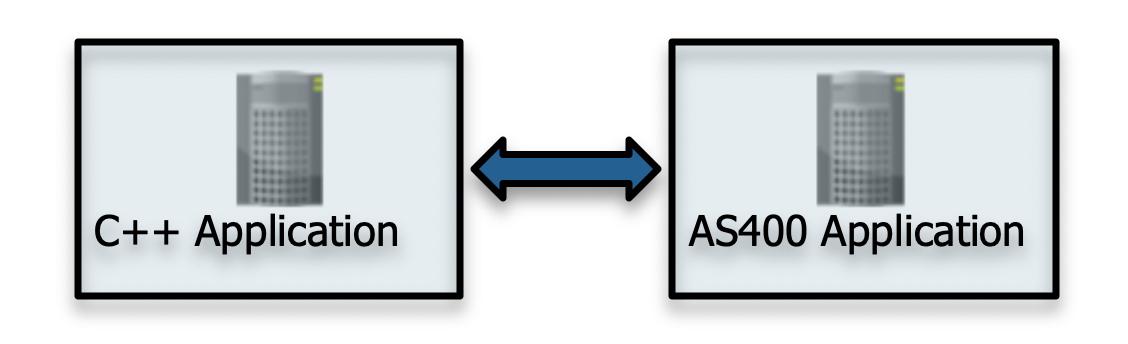
## ActiveMQ broker

Simple answer to what is ActiveMQ is. ActiveMQ is a popular open source Java message-oriented middleware (MoM). For me that was not enough.

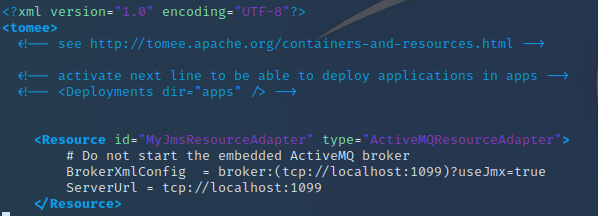
Another notion is JMS which stands for Java Messaging Service. So, JMS is like protocol but its specified by java, it allows applications to communicate in an easy way, send messages back and forth. ActiveMQ brokers are the implementation of JMS. It is very useful to have some standards in messaging services because as written in openlogic website (4):

Before messaging systems like ActiveMQ, it was difficult — if not impossible — to “federate” applications that:

* Were written in disparate languages.
* Resided on heterogeneous platforms.
* Used different data structures and protocols.



To support CVE configuration I need to setup ActiveMQ broker to listen on port 1099 and useJMX parameter in broker URI to be true. I am able to configure these parameter in conf folder in tome.xml file. The screenshot below shows the final configuration.



## TomEE

Brief introduction for those who are interested in web servers especially apache. In apache website the description of tome is following:

Apache TomEE, pronounced "Tommy", is an all-Apache Java EE 6 Web Profile certified stack where Apache Tomcat is top dog. Apache TomEE is assembled from a vanilla Apache Tomcat zip file. We start with Apache Tomcat, add our jars and zip up the rest. The result is Tomcat with added EE features - TomEE. (5) From my experience after I have read some articles and documentation, I can say the TomEE is very useful server and very simple to use. It consists of several open-source components that are shown below:

|  |  |
| --- | --- |
| **Component** | **Description** |
| [Apache Tomcat](https://en.wikipedia.org/wiki/Apache_Tomcat) | HTTP server and Servlet container supporting [Java Servlet](https://en.wikipedia.org/wiki/Java_Servlet) and [JavaServer Pages](https://en.wikipedia.org/wiki/JavaServer_Pages) (JSP). |
| [Apache OpenEJB](https://en.wikipedia.org/wiki/Apache_OpenEJB) | Open-source [Enterprise JavaBeans](https://en.wikipedia.org/wiki/Enterprise_JavaBean) (EJB) container system. |
| [Apache OpenWebBeans](https://en.wikipedia.org/wiki/Apache_OpenWebBeans) | Open-source [Java Contexts and Dependency Injection](https://en.wikipedia.org/w/index.php?title=Java_Contexts_and_Dependency_Injection&action=edit&redlink=1) (CDI) implementation. |
| [Apache OpenJPA](https://en.wikipedia.org/wiki/Apache_OpenJPA) | Open-source [Java Persistence API](https://en.wikipedia.org/wiki/Java_Persistence_API) (JPA) 2.1 implementation. |
| [Apache Geronimo Transaction](https://en.wikipedia.org/w/index.php?title=Apache_Geronimo_Transaction&action=edit&redlink=1) | Open-source [Java Transaction API](https://en.wikipedia.org/wiki/Java_Transaction_API) (JTA) 1.2 implementation. |
| [Apache MyFaces](https://en.wikipedia.org/wiki/Apache_MyFaces) | Open-source [Java Server Faces](https://en.wikipedia.org/wiki/Java_Server_Faces) (JSF) implementation. |
| [Apache ActiveMQ](https://en.wikipedia.org/wiki/Apache_ActiveMQ) | Open-source [Java Message Service (JMS)](https://en.wikipedia.org/wiki/Java_Message_Service) implementation. |
| [Apache CXF](https://en.wikipedia.org/wiki/Apache_CXF) | [Web Services](https://en.wikipedia.org/wiki/Web_Services) frameworks with a variety of protocols - such as SOAP, XML/HTTP, [RESTful](https://en.wikipedia.org/wiki/Representational_State_Transfer) [HTTP](https://en.wikipedia.org/wiki/HTTP). |
| [Apache Derby](https://en.wikipedia.org/wiki/Apache_Derby) | Full-fledged [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS) with native [Java Database Connectivity](https://en.wikipedia.org/wiki/Java_Database_Connectivity) (JDBC) support. |

In my setup I am using apache TomEE plume version 7.1.1 which is vulnerable version as stated in [CVE-2020-11969](#_heading=h.30j0zll) chapter.

I also want to mention that in version 7.1.1 which I am using, there are many releases with different components for example the table below is for version TomEE plus. There are versions like plume, webprofile, microprofile, OpenEJB Standalone and few more. You can see comparision of different releases in [this link](http://tomee.apache.org/comparison.html).

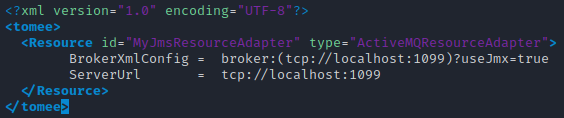
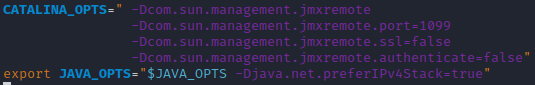
Deploying webapps is very simple as moving .war extension files to webapps folder. Apache then recognizes new application and apache then configure and run the application in server.

### JMX

Java management extension in apache allows you to monitor and control the behavior of ActiveMQ broker. You can control or monitor things like heap memory, threads, CPU usage and configure various settings such as MBeans and so on. Example JMX tool is JConsole. It lets you connect to a remote apache server and monitor it with a simple GUI and use some features to manage apache TomEE.

# CVE setup

Following chapter is step by step guide for setup the Apache with all dependencies to support the CVE requirements. To follow my environment and my steps there are prerequisites to have already Kali Linux as VM installed and connected to internet. Then follow these steps:

1. Download Apache TomEE plume v 7.1.1 from [this](https://tomee.apache.org/download-archive.html) website as TAR.GZ.
2. Extract it with some archive tool or with command – **tar -xvf apache-tomee-7.1.1-plume.tar.gz**
3. There is a new folder in your working directory called **apache-tomee-7.1.1-plume**
4. Open directory from previous step and navigate to conf directory
5. There is **tomee.xml** file. Open it in your favorite editor and make sure it looks like in the picture below. 
6. Next you have to go to bin directory and create the **setenv.sh** file and make it look like in image below. 
7. Make the file executable by running command **chmod +x setenv.sh**
8. Last step is to run TomEE by executing command **./catalina.sh run**
9. Now your Apache TomEE is running wit embedded ActiveMQ broker and JMX configured on port 1099.

# Nmap

We all know Nmap so just a brief introduction for those who do not know Nmap. It is a free, open source tool for security auditing, network scanning and more. The important part in Nmap for this assignment is the NSE, which we will describe in the NSE chapter below.

## NSE

NSE stands for Nmap Scripting Engine, it allows network administrators or users to write their own scripts for Nmap, it uses Lua programming language. Nmap has its prewritten scripts, these are divided into following categories: auth, broadcast, default, discovery, dos, exploit, external, fuzzer, intrusive, malware, safe, version, and vuln. One script can be under many categories, for example my script will be under vuln and safe. Vuln scripts checks for specific known vulnerabilities and safe means it is not intrusive.

# Python

I am using Python for evaluating the result of two NSE scripts.

# Detection design

# Code snippets

# Bibliography

1. **NIST. [Online] https://nvd.nist.gov/vuln/detail/CVE-2020-11969#vulnCurrentDescriptionTitle.**

**2. Mitre. *cve.* [Online] https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-11969.**

**3. Monson-Haefel, Richard. 5 Minutes or Less: Message-Driven Beans with ActiveMQ and TomEE. *tomitribe.* [Online] 5 23, 2019. https://www.tomitribe.com/blog/5-minutes-or-less-message-driven-beans-with-activemq-and-tomee/.**

**4. Reock, Justin. what-apache-activemq. *openlogic.* [Online] 6 4, 2020. [Cited: 11 3, 2020.] https://www.openlogic.com/blog/what-apache-activemq.**

**5. apache-tomee. *apache.* [Online] [Cited: 11 3, 2020.] http://openejb.apache.org/apache-tomee.html.**

**6. [Online] https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-11969.**

**7. r. [Online]**