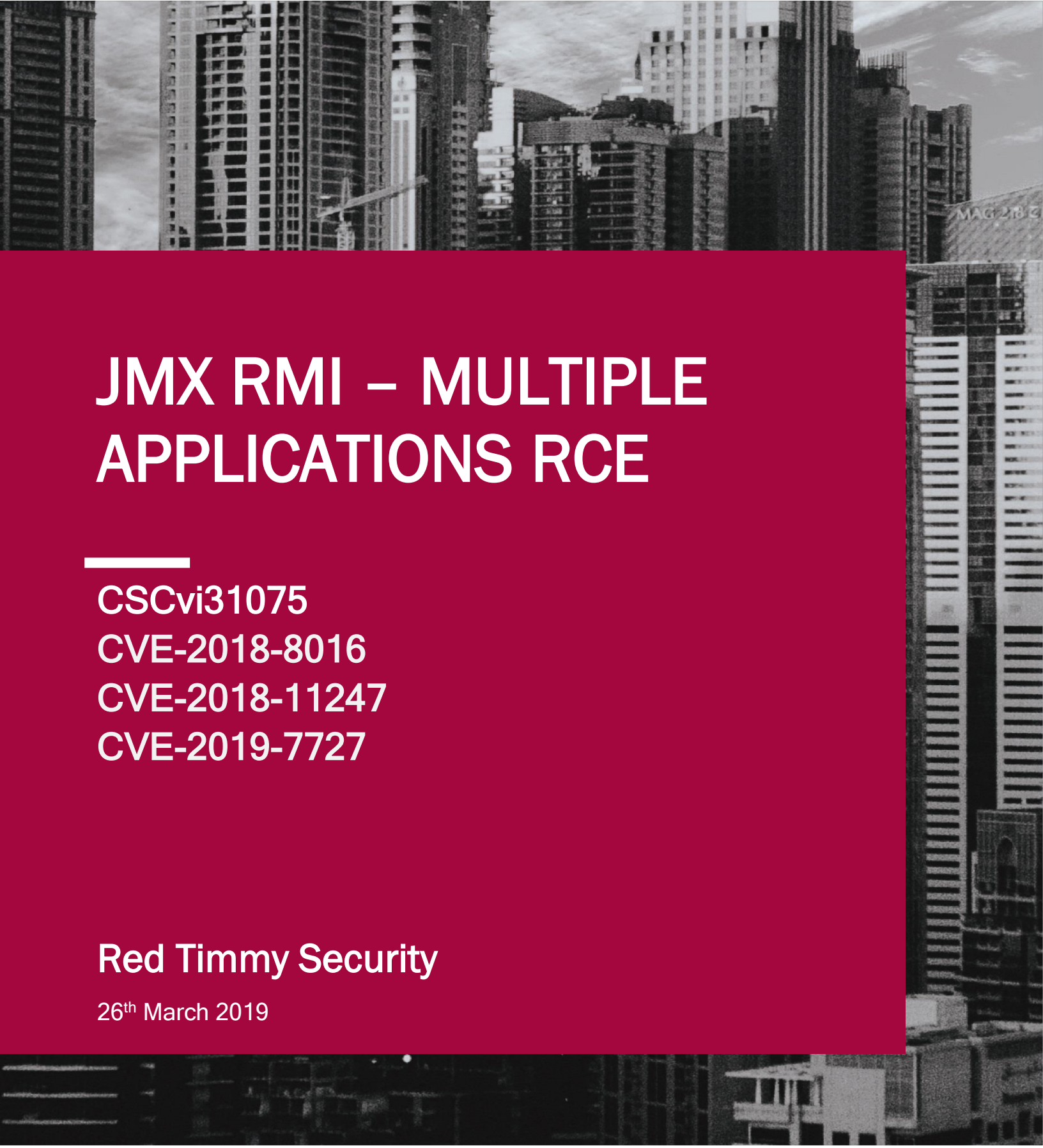

JMX RMI – MULTIPLE APPLICATIONS RCE

CSCvi31075
CVE-2018-8016
CVE-2018-11247
CVE-2019-7727

Red Timmy Security

26th March 2019

Red Timmy Security

https://redtimmysec.wordpress.com/
https://www.twitter.com/redtimmysec

Red Timmy Security

Summary

FOREWORD

After achieving a foothold inside the targeted organization, an attacker surely will search for vulnerabilities giving him/her the ability to compromise other machines and move laterally into the network. Especially for companies adopting Java-based software solutions, one of the most abused services to achieve Remote Command Execution[1] is JMX/RMI. It has revealed to be very pervasive in the business LAN/DMZ contexts. Indeed around one year ago we performed a random analysis of both open source and business tools, checking for the presence of JMX/RMI ports. We ended up discovering (and sometimes directly reporting) some new vulnerabilities. Few CVE(s) were registered as well. This whitepaper includes main highlights of our findings.

[1] https://docs.oracle.com/javase/8/docs/technotes/guides/management/agent.html

JMX/RMI REMOTE COMMAND EXECUTION

WHAT IS JMX

JMX (Java Management Extension) is a documental specification for remote management and monitoring of Java applications. Its main unit is the MBean (management bean), a java object exposing some attributes that can be read/written through the network, and most importantly a series of functions or operations invokable from remote. A so-called "*MBeanServer*" (or more simply *JMX Server*) keeps track of all registered MBeans inside a kind of searchable register/archive. It is the component puts in charge of managing the communication between clients that want access to one or more exposed functions/attributes of an MBean and the MBean itself.

JMX was not built with the security principle in mind. Therefore, whoever is able to reach the network port it is listening to can also invoke the exposed methods anonymously, without going through a formal authentication process.

WHAT IS RMI

The RMI (Remote Method Invocation) protocol is the most common mechanism (as well as the only one that the JMX standard expressly requires to be supported by default) through which the methods and functions the MBeans remotely expose (made available by means of JMX server) are invoked by clients.

WHERE IS THE PROBLEM WITH JMX/RMI?

By default no authentication is enabled for JMX/RMI. Furthermore the authentication, when rarely adopted, is only restricted to a couple of options:

- *File-based*: insecure as passwords are left in clear-text in the filesystem (also transmitted in clear-text over the network);
- *TLS mutual authentication*: difficult to set up and maintain with the growth of the numbers of clients and nodes, as it requires the generation of digital keys and certificates for each of them.

The Oracle Java documentation is self-explanatory when it comes to determine where the problem stems from: (https://docs.oracle.com/javase/8/docs/technotes/guides/management/agent.html)

"***Caution*** *- [...] any remote user who knows (or guesses) your port number and host name will be able to monitor and control your Java applications and platform. Furthermore, possible harm is not limited to the operations you define in your MBeans. 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*".

AFFECTED SOFTWARE

Below follows the list of components and software solutions that we found affected by this specific issue during an analysis conducted in the period February-March 2018.

CISCO UNIFIED CUSTOMER VOICE PORTAL <= 11.X

Cisco Unified CVP is an intelligent IVR (Interactive Voice Response) and call control solution. In its default configuration, until version <= 11.x and potentially above, an unauthenticated JMX/RMI interface is bound to a wildcard address on TCP ports 2098 and 2099. An attacker establishing a network connection to one of these affected ports and sending the malicious payload could easily trigger the RCE.

The vulnerability was first discovered on February 2018. The vendor was made aware almost immediately. After multiple meetings and discussions occurred between April and July 2018, Cisco agreed to document two new security procedures in to its CVP configuration guide[2]:

- Secure JMX Communication between CVP Components
- Secure JMX Communication between OAMP and Call Server using Mutual Authentication.

[2] https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/cust_contact/contact_center/customer_voice_portal/cvp11_6/configuration/guide/cvvp_b_configuration-guide-for-cisco-unified/cvvp_b_configuration-guide-for-cisco-unified_chapter_010000.html

We reopen the dialogue with Cisco on February 2019 when a CVE has been asked but not assigned, as the vendor considers the flaw as a configuration issue.

Anyway Cisco has published a security bulletin for the flaw we reported at the URL https://quickview.cloudapps.cisco.com/quickview/bug/CSCvi31075

NASDAQ BWISE <= 5.X

This is a commercial GRC (Governance, Risk and Compliance Management) solution for risk handling and management. Branch 5.x or Nasdaq BWise is vulnerable because by default the SAP BO Component enables JMX/RMI on TCP port 81 without authentication.

We discovered the vulnerability and contacted the vendor on March 2018. After discussing with them, the release of Service Pack (SP02) has been announced to solve the problem.

A CVE number has not been requested until February 2019, when we realized that in the meantime another security researcher had registered CVE-2018-11247[3] for the same issue. As indicated in their published security bulletin https://packetstormsecurity.com/files/148918/Nasdaq-BWise-5.0-JMX-RMI-Interface-Remote-Code-Execution.html, the researcher had discovered the vulnerability 2 months after us (May 2018).

However as a CVE already existed, we did not requested a new one.

NICE ENGAGE PLATFORM <= 6.5

NICE Engage is an interaction recording platform. Versions <= 6.5[4] (and potentially above) open up the TCP port 6338 where a JMX/RMI service listens to without authentication. Of course this may be abused to launch remote commands by deploying a malicious MBean.

On March 4th 2018 we contacted the vendor and on 7th same month they have recognized the vulnerability. They also declared that no specific fix would have been released because enabling the JMX file-based authentication was considered enough to mitigate the finding. Anyway, this change is not reflected in the default configuration and must be applied manually, leaving at risk the

[3] https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-11247

[4] http://www.smartcustomerservice.com/Articles/News-Features/NICE-Introduces-Engage-6.5-112222.aspx

companies using the product and are not aware of the problem. Only very recently, February 2019, we have registered CVE-2019-7727[5] for this vulnerability.

APACHE CASSANDRA 3.8 THROUGH 3.11.1 AND CLOUDERA ZOOKEEPER/CDH 5.X/6.X

On February 2018 we discovered that the Apache Software Foundation project dubbed Cassandra (release between 3.8 and 3.11) exposed the TCP port 7199 on which JMX/RMI was running. We did not report the finding immediately. Soon after someone else did it and registered CVE-2018-8016[6]. All details are perfectly explain here:

https://lists.apache.org/thread.html/bafb9060bbdf958a1c15ba66c68531116fba4a83858a2796254da066@%3Cuser.cassandra.apache.org%3E

During a contextual security investigation on March 2018 we also managed to spot multiple instances of Cloudera Zookeeper/CDH (versions 5.x and 6.x) affected. In this case the TCP port 9010 was exposing a JMX/RMI service. The vendor is aware of the problem at least since June 2018. In one of their release notes[7] they wrote:

"*A successful attack may leak data, cause denial of service, or even allow arbitrary code execution on the Java process that exposes a JMX port. Beginning in Cloudera Manager 6.1.0, it is possible to configure mutual TLS authentication on ZooKeeper's JMX port*".

The possibility to configure mutual TLS authentication for previous product versions is unknown instead. We did not register a CVE for this vulnerability.

[5] http://cve.mitre.org/cgi-bin/cvename.cgi?name=2019-7727

[6] https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-8016

[7] https://www.cloudera.com/documentation/enterprise/6/release-notes/topics/rg_cm_601_known_issues.html

EXPLOIT CODE

Working tools to exploit JMX/RMI vulnerabilities exist out there. Some good examples are *sjet*[8] and *mjet*[9]. When we have started our first investigations in this field did not manage to find one fitting all requirements (we have had problems to target specific contexts and configurations) and have decided to develop our own. This tool is not going to be publicly shared for now. It probably will in future, so visit our blog (https://redtimmysec.wordpress.com).

Moreover, if you want to know more about JMX/RMI exploitation and mitigation, check out our Blackhat Las Vegas courses on 3-4 and 5-6 August 2019[10], because this will be one of the topics covered there.

Stay tuned!

[8] https://github.com/siberas/sjet

[9] https://github.com/mogwaisec/mjet

[10] https://www.blackhat.com/us-19/training/advanced-java-web-and-client-server-application-hacking-with-a-bit-of-crypto.html