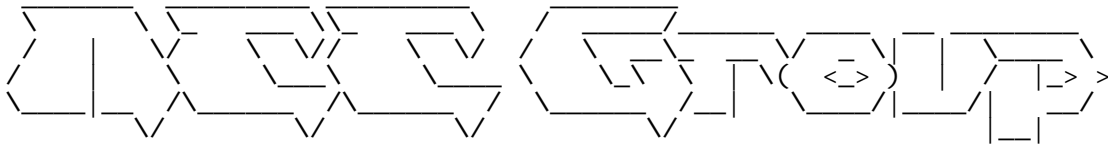


CVE-2017-3241 Java RMI Registry.bind() Unvalidated Deserialization.txt



<https://www.nccgroup.trust/research/>

Vulnerability Summary

Title	Java RMI Registry.bind() Unvalidated Deserialization
Reference	VT-87
Discoverer	Nick Bloor (@NickstaDB)
Vendor	Oracle
Vendor Reference	S0818584
Systems Affected	Java SE <= 6u131, <= 7u121, <= 8u112, Java SE Embedded <= 8u111, JRockit <= R28.3.12
CVE Reference	CVE-2017-3241
Risk	Critical
Status	Fixed

Resolution Timeline

Discovered	01 January 2017
Reported	11 January 2017
Fixed	19 January 2017

Vulnerability Description

Java Remote Method Invocation (RMI) allows objects of classes that implement the `java.rmi.Remote` interface to be exposed over a network allowing one application to call methods on an object that exists on a remote server. Objects are exposed for remote method invocation by binding them to a registry service using the `bind()` method of the `java.rmi.registry.Registry` interface.

The default `java.rmi.registry.Registry` implementation does not validate the class of the object that was passed to the `bind()` method before deserializing and instantiating the object. This means that any object can be passed to the registry for binding, even if the class of that object does not implement `java.rmi.Remote`, and hence can never be bound to the registry. This presents an open entry point for Java deserialization attacks whereby an attacker crafts objects in order to manipulate code that is automatically executed after the object has been deserialized. In many cases Java deserialization can enable arbitrary commands to be executed on the server. Such an attack against the default RMI Registry implementation does not require authentication.

Applications exposing objects through the default RMI Registry won't ever see the exploit payload and hence cannot easily put defensive measures in place in order to prevent the deserialization of arbitrary objects.

Technical Details

This issue can be demonstrated by binding a crafted object to an RMI registry. The following Python script can be used along with a specially crafted object in order to demonstrate this issue:

```
import socket
```


CVE-2017-3241 Java RMI Registry.bind() Unvalidated Deserialization.txt

It was recommended that Oracle implement look-ahead deserialization within the default RMI Registry implementation. By looking ahead at the class of the object passed to the bind() method Java can verify that the class is compatible with java.rmi.Remote and either bind the object or throw an exception accordingly.

Oracle implemented this recommendation in an update to Java which was released with their January 2017 Critical Patch Update. Details of the January 2017 Critical Patch Update and acknowledgement for the vulnerability report can be found at the following URL:

<http://www.oracle.com/technetwork/security-advisory/cpujan2017-2881727.html>

NCC Group

Research	https://www.nccgroup.trust/research
Twitter	https://www.twitter.com/NCCGroupInfoSec / @NCCGroupInfoSec
Open Source	https://github.com/nccgroup
Blog	https://www.nccgroup.trust/en/blog/cyber-security/