INTEGRIGY

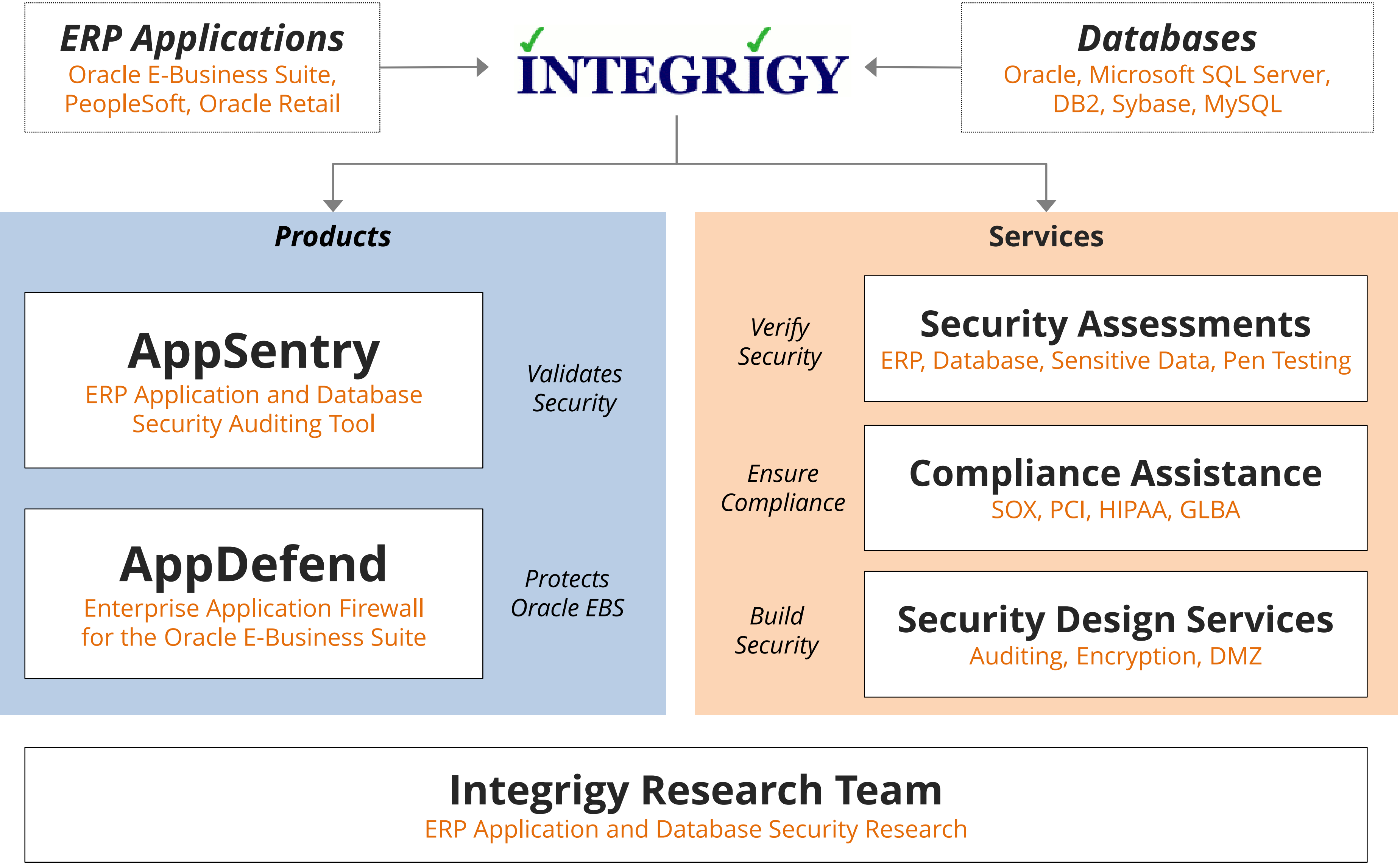
Oracle Java Deserialization Vulnerabilities Explained

December 1, 2016

Stephen Kost
Chief Technology Officer
Integrigy Corporation

Phil Reimann
Director of Business Development
Integrigy Corporation

About Integrigy



ERP Applications
Oracle E-Business Suite, PeopleSoft, Oracle Retail

INTEGRIGY

Databases
Oracle, Microsoft SQL Server, DB2, Sybase, MySQL

Products

AppSentry
ERP Application and Database Security Auditing Tool

Validates Security

AppDefend
Enterprise Application Firewall for the Oracle E-Business Suite

Protects Oracle EBS

Services

Verify Security

Security Assessments
ERP, Database, Sensitive Data, Pen Testing

Ensure Compliance

Compliance Assistance
SOX, PCI, HIPAA, GLBA

Build Security

Security Design Services
Auditing, Encryption, DMZ

Integrigy Research Team
ERP Application and Database Security Research

Java deserialization vulnerabilities in Oracle products are actively being exploited

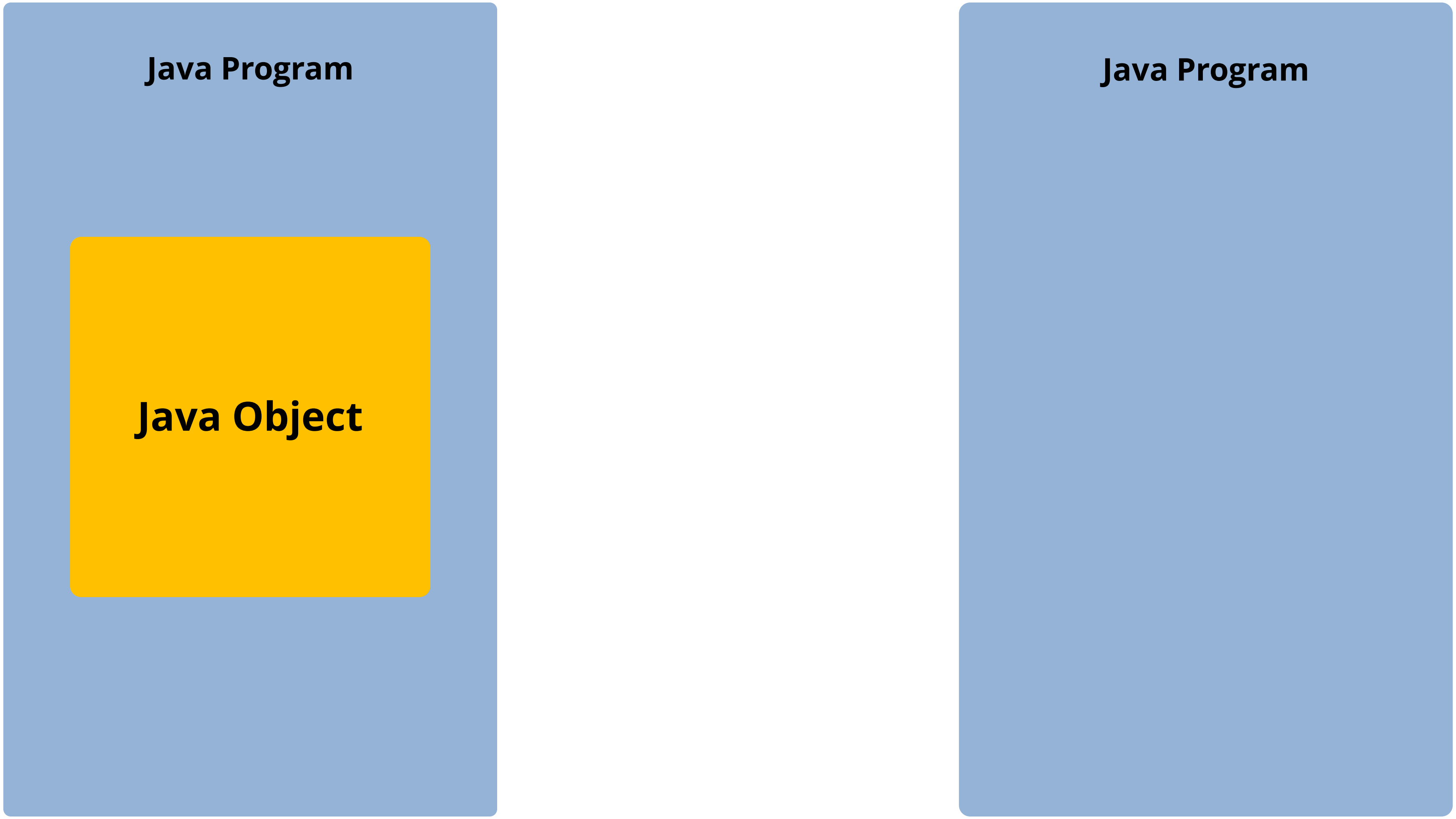
Krebs on Security – November 29, 2016

“It appears our attacker has been using a number of tools which enabled the scanning of large portions of the Internet and several specific targets for vulnerabilities. The most common vulnerability used ‘weblogic unserialize exploit’ and especially targeted **Oracle Corp.** server products, including **Primavera** project portfolio management software.”

“Read this and install patch before you connect your server to internet again,” the attacker wrote, linking to this advisory that Oracle issued for a security hole that it plugged in November 2015.

https://krebsonsecurity.com/2016/11/san-francisco-rail-system-hacker-hacked/

What is Java Object Serialization?

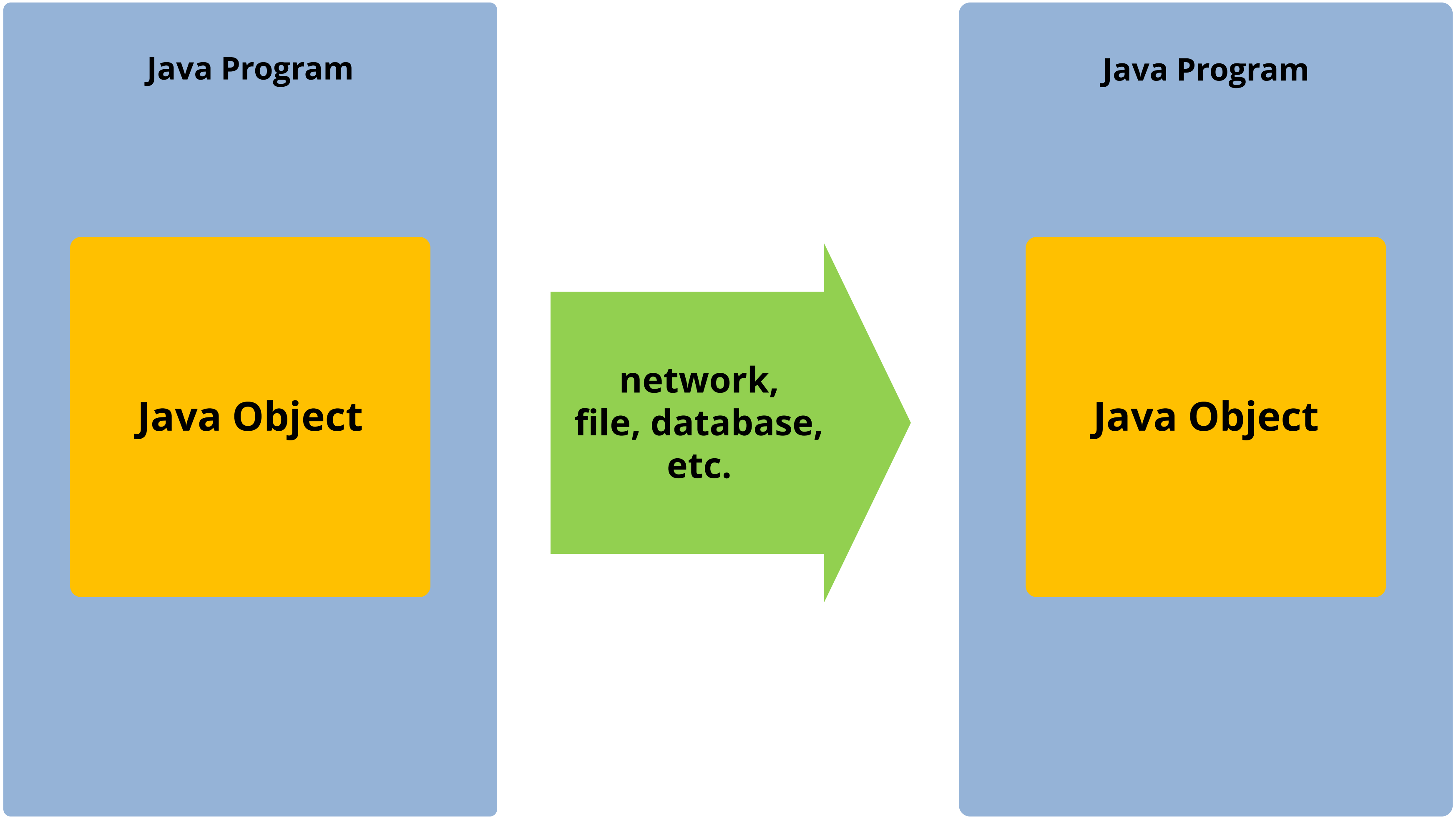


Java Program

Java Object

Java Program

What is Java Object Serialization?



What is Java Object Serialization?

Java object serialization is the conversion of an object to a byte stream (**serialization**), transfer of the byte steam, and conversion of the byte stream back to a Java object (**deserialization**).

Serialization

```
ObjectOutputStream out;

out = new ObjectOutputStream(httpservletresponse.getOutputStream());

out.writeObject(person);
```

Network, file, database, etc.

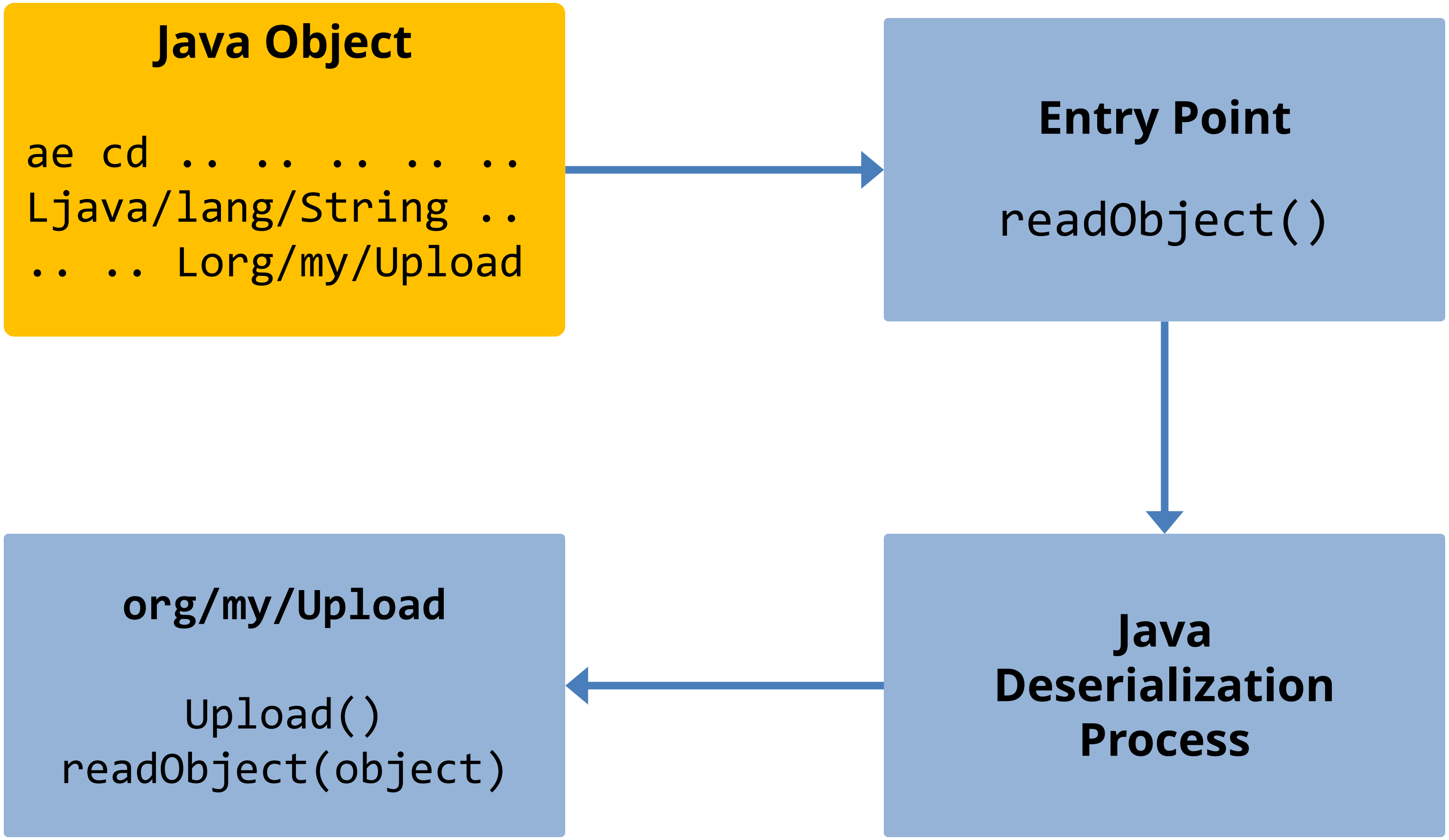
Deserialization

```
ObjectInput Stream in;

in = new ObjectInputStream(httpservletrequest.getInputStream());

Person person = (Person)in.readObject();
```

Java Deserialization Process



Java Object

```
ae cd .. .. .. .. ..
Ljava/lang/String ..
.. .. Lorg/my/Upload
```

Entry Point

`readObject()`

Java Deserialization Process

org/my/Upload

```
Upload()
readObject(object)
```

Java Deserialization Process

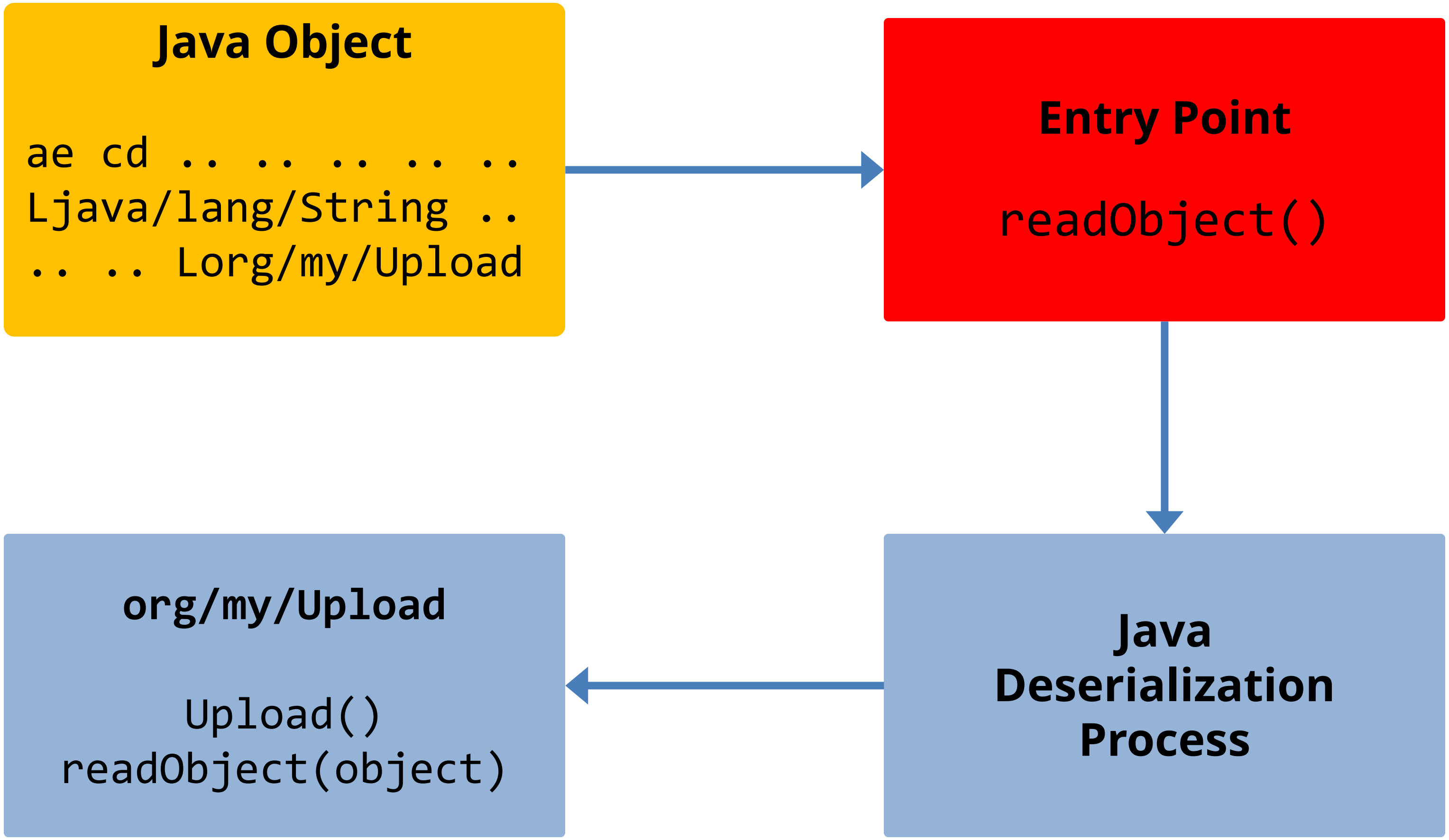
- **Class name is embedded in byte stream**
  - May be one or more classes including hierarchies
  - Can be any serializable class on the classpath
- **ObjectInputStream does no validation**
- **Final class cast is not performed until after all deserialization work is complete**

```
in = new ObjectInputStream(httpservletrequest.getInputStream());

Person person = (Person) in.readObject();
```

All the deserialization work is done here

Java Deserialization Vulnerability – Entry Point



Java Object

```
ae cd .. .. .. .. ..
Ljava/lang/String ..
.. .. Lorg/my/Upload
```

Entry Point

`readObject()`

Java Deserialization Process

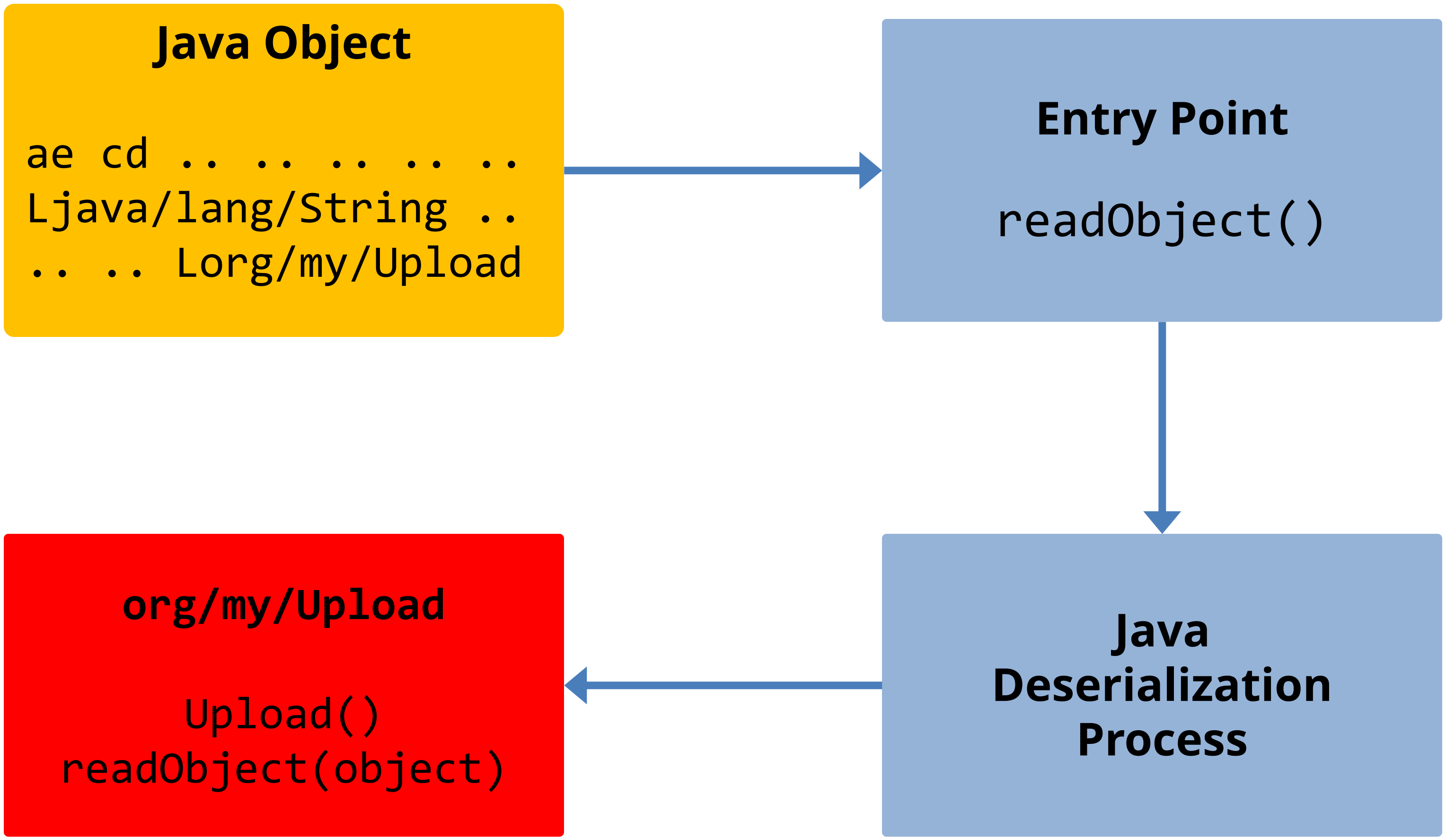
org/my/Upload

```
Upload()
readObject(object)
```

Entry Point Examples

- Remote Method Invocation (RMI)
- Java Management Extension (JMX)
- Java Message Service (JMS)
- Java Server Faces/Oracle ADF ViewState
- Any code using ObjectInputStream with unvalidated or untrusted input
  - Oracle WebLogic (T3)
  - Oracle Primavera
  - Oracle Hyperion
  - Oracle E-Business Suite
  - …

Java Deserialization Vulnerability – “Gadget”



Java Object

```
ae cd .. .. .. .. ..
Ljava/lang/String ..
.. .. Lorg/my/Upload
```

Entry Point

`readObject()`

Java Deserialization Process

org/my/Upload

```
Upload()
readObject(object)
```

Gadgets and “Magic Classes”

- Gadget is where the vulnerability does its work such as writing arbitrary files to the operating system
- Many possible gadgets
- More are being discovered all the time
- Apache Commons-FileUpload Remote Code Execution (RCE)
  - CVE-2013-2186
  - Discovered by Pierre Ernst
- Apache Commons-Collections RCE
  - CVE-2015-7501
  - Discovered by Gabriel Lawrence and Chris Frohoff

Deserialization Exploit Tool

- **ysoserial is an exploit tool used to generate deserialization attack payloads**
  - https://github.com/frohoff/ysoserial/
  - Preloaded with most common gadgets
  - Creates attack payload to send to vulnerable entry point

```
java -jar ./ysoserial-0.0.4-all.jar CommonsCollections1
'ping integrigy.com' > payload
```

Java Deserialization References

- **Java Deserialization Cheat Sheet**
  - https://github.com/GrrrDog/Java-Deserialization-Cheat-Sheet
- **Marshalling Pickles**
  - http://www.slideshare.net/frohoff1/appseccali-2015-marshalling-pickles
- **Java Deserialization Vulnerabilities – The Forgotten Bug Class**
  - http://blog.deepsec.net/deepsec2016-talk-java-deserialization-vulnerabilities-forgotten-bug-class-matthias-kaiser/
- **What Do WebLogic, WebSphere, JBoss, Jenkins, OpenNMS, and Your Application Have in Common? This Vulnerability**
  - https://foxglovesecurity.com/2015/11/06/what-do-weblogic-websphere-jboss-jenkins-opennms-and-your-application-have-in-common-this-vulnerability/

CVE-2015-4852 – Oracle WebLogic

- **Vulnerability in Oracle WebLogic J2EE monitoring and JMX used by WebLogic Scripting Tool (WLST)**
  - Versions 10.3.6.0, 12.1.2.0, 12.1.3.0, 12.2.1.0

- **WebLogic uses the T3 protocol on default port 7001 for management**

- **Able to send serialized objects to port 7001 using T3 protocol**

http://www.oracle.com/technetwork/topics/security/alert-cve-2015-4852-2763333.html

CVE-2015-4852 – Oracle WebLogic

WebLogic used by –

- Oracle E-Business Suite 12.2
- Oracle PeopleSoft
- Oracle Hyperion
- Oracle Agile
- Oracle Banking Platform
- Oracle Enterprise Manager
- Oracle VM Manager
- and others

CVE-2015-4852 – Oracle WebLogic

Detailed exploit code and examples are available

https://github.com/metalnas/loubia

Java Deserialization Vulnerabilities – The Forgotten Bug Class

http://blog.deepsec.net/deepsec2016-talk-java-deserialization-vulnerabilities-forgotten-bug-class-matthias-kaiser/

What Do WebLogic, WebSphere, JBoss, Jenkins, OpenNMS, and Your Application Have in Common? This Vulnerability

https://foxglovesecurity.com/2015/11/06/what-do-weblogic-websphere-jboss-jenkins-opennms-and-your-application-have-in-common-this-vulnerability/

CVE-2015-4852 – Oracle WebLogic

- **Advisory released November 10, 2015**
  - Released due to exploit details published
  - Patches available after release

- **Included in January 2016 Critical Patch Update (CPU)**
  - WebLogic Patch Set Update (PSU)
  - 12.2.1.0.**1**
  - 12.1.3.0.**6**
  - 12.1.2.0.**8**
  - 10.3.6.0.**13**

- **Per Oracle Lifetime Support, 10.3.6 or 12.1.2 is required as a minimum**

CVE-2015-4852 – Oracle WebLogic

- Oracle fixed by implementing a blacklist of forbidden gadgets
- T3 serialized object support still enabled

CVE-2015-4852 – Oracle WebLogic

MOS Note ID 2076338.1 – ***CVE-2015-4852 Mitigation Recommendations for Oracle WebLogic Server Component of Oracle Fusion Middleware***

Option 1 – Filter T3 traffic through a proxy to WLS

- After patching is complete, you may continue filtering T3 traffic at the proxy level. (There is no downside, and it provides security in depth).

Option 2 – Filter T3 access on WLS with connection filters (per IP address)

- After patching is complete, connection filters on WLS are optional as they may have a performance impact.

CVE-2015-4852 – Oracle WebLogic

MOS Note ID 2075927.1 – *CVE-2015-4852 Patch Availability Document for Oracle WebLogic Server Component of Oracle Fusion Middleware*

- Provides detailed information on available patches

CVE-2015-7501 – Oracle Impact

- **CVE-2015-7501 – Apache Commons-Collection Remote Code Execution (RCE)**
  - One of many gadgets
- **April 2016 Critical Patch Update (CPU) includes 1 fix for CVE-2015-7501**
  - Oracle Application Testing Suite

CVE-2015-7501 – Oracle Impact

- **July 2016 Critical Patch Update (CPU) includes 23 fixes for CVE-2015-7501**
  - Oracle Enterprise Manager Ops Center
  - Oracle Transportation Management
  - Oracle Communications ASAP
  - Oracle Banking Platform
  - Oracle Health Sciences and Clinical Development Center
  - Oracle Healthcare Analytics Data Integration
  - Oracle Insurance – 4 modules
  - Oracle Retail – 4 modules
  - Oracle Utilities – 3 modules
  - Oracle Policy Automation – 4 modules
  - Oracle Primavera – 2 modules

CVE-2015-7501 – Oracle Impact

- **October 2016 Critical Patch Update (CPU) includes 19 fixes for CVE-2015-7501**
  - Oracle WebLogic
  - Oracle Agile PLM
  - Oracle Commerce Guided Search/Experience Manager
  - Oracle FLEXCUBE Banking – 7 modules
  - Oracle Financial Services Analytics – 2 modules
  - Oracle Banking Digital Experience
  - Oracle Insurance Istream
  - Oracle Retail – 2 modules
  - Oracle MICROS XBR
  - Oracle Big Data Graph

Summary

- **Java deserialization security issues are not going away any time soon**
  - Oracle products make extensive use of Java serialization
- **Entry points and gadgets are the problem**
  - Many undiscovered and unprotected entry points
  - Fixing gadgets is like “whack-a-mole”
- **Java deserialization vulnerabilities are being actively exploited**

Recommendations

- **Limit access to WebLogic management ports (7001)**
  - Use firewalls or proxies to limit T3 at a minimum
- **Assess all Oracle application technology stacks for impact of WebLogic vulnerabilities**
- **Review all custom development for Java deserialization vulnerabilities**
  - Oracle ADF applications are vulnerable
- **Oracle E-Business Suite – Use Integrigy AppDefend to protect against deserialization vulnerabilities in the application**

Contact Information

Stephen Kost

Chief Technology Officer

Integrigy Corporation

web: **www.integrigy.com**

e-mail: **info@integrigy.com**

blog: **integrigy.com/oracle-security-blog**

youtube: **youtube.com/integrigy**