Java Hurdling
Obstacles and Techniques in Java Client Penetration-Testing

Tal Melamed
Application Security Expert
Tal@AppSec-Labs.com
Agenda

- Me
- AppSec Labs
- The problems
  - Fail #1
  - Fail #2
  - Fail #3

© All rights reserved to AppSec Labs Ltd.
Tech Lead @ AppSec Labs
Tal@AppSec-Labs.com
Application Security Expert
Trainer, builder & breaker
Follow me @ appsec.it
https://github.com/nu11p0inter

I don’t always cyber stalk you, but when I do: http://Lnkdin.me/cyber
WHO ARE WE?

WHAT DO WE WANT?

secured apps!

https://appsec-labs.com/
AppSec Labs

Industry vectors:

AppSec Labs provides its high end services to the following industry vectors:

- High Tech software development
- Banking and financing
- National security
- IoT
- Cloud
- Pharmaceuticals
- Commerce
- Travel and transport
- IT Security products
- Biometrics
- Education
- Gaming
- Government
- Telecommunications
We are hiring!

- Experienced PT
- Exp. Code-Review*
- Training skills
- Willing to travel *
- English
- Independent work and self-learning ability

jobs@appsec-labs.com
Disclaimer

- This is a true story. The events depicted in this talk took place in 2016.

- At the request of the survivors, names, characters, places and incidents were changed, and are either products of the author’s imagination or are used fictitiously.

- Any resemblance to actual events or locales or persons, living or dead, is entirely coincidental.

- The rest is told exactly as it occurred.

- **Warning:** this presentation might contain memes...
The Problems

- TCP rather than HTTP
- SSL/TLS
- Certificate Pinning
- Runtime manipulation
- Patching the application
- ProKSy – revealed for the first time...
Day 1: I Got This!

- Let’s use BURP! - set the HTTP Proxy (option in tool)

- Nothing happens...

- Looking at WireShark

- Port 1XXXX TLS - Not HTTP!

- Sure, let use AppSec Labs’ incredible TCP proxy tool (TBC)
Problem #1: No HTTP/S

- We all Burp (nothing to be ashamed about)
- But what if...
  
  the application is not communicating over HTTP(s)?
Echo Mirage – by Wildcroft Security

Link: unknown (good luck with FileHippo)
Advanced Packet Editor

- https://github.com/appsec-labs/Advanced_Packet_Editor
- Based on:
Intercept & tamper with TCP-based comm
What’s new
APE

External Filter – python based

Welcome Job Steves

AppSec Labs

```python
data = data.replace("Gate Bills", "Job Steves")
```
APE – Listen to Requests

HTTP/S? – Why not integrating with Burp?
What Really Happened?

Nothing! Probably SSL...

(Also, doesn’t work on 64-bit)
What Else is There?

Stcppipe

http://aluigi.altervista.org/mytoolz/stcppipe.zip
A Fraction of Hope...

Simple TCP proxy/datapipe 0.4.8b
by Luigi Auriemma
E-mail: aluigi@autistici.org
Web: aluigi.org

- local port: 13000
- remote hosts: 192.168.1.30:13000
- wait connections:
  IN 192.168.1.29:1463
  OUT 192.168.1.30:13000

ac ed 00 05 75 72 00 13 5b 4c 6a 61 76 61 2e 6c 61 6e 67 2e 4f 62 6a 65 63 74 3b 90 ce 58 9f 10
73 29 6c 02 00 00 78 70 00 00 00 03 73 72 00 47
6f 72 67 2e 61 70 70 63 69 63 61 74 74 69 6f 6e 2e 54
72 65 65 53 74 72 75 63 74 75 72 65 4d 61 6e 61
67 65 72 24 54 72 65 65 53 74 72 75 63 74 43 6f
6d 70 6f 6e 65 6e 74 46 59 17 db 9c 4a f6 cf 02

THE HECK?
What is Serialization
- Converting the state of data to a byte stream so that the byte stream can be reverted back into a copy of the object

What is the problem?
- Deserialization of untrusted data

What does that mean?
- De-serializing data coming from the client could abuse the application logic, deny service, or execute arbitrary code.

What to look for?
- ObjectInput.readObject()
- Externalizable.readExternal()
- Serializable.readResolve()
- ObjectOutputStream.replaceObject()
- ObjectInputStream.readUnshared()
- Many more...
All You Need to Know...

- You can find everything here:
  - https://github.com/njfox/Java-Deserialization-Exploit

- PayPal RCE (2016)
  - http://artsploit.blogspot.co.il/2016/01/paypal-rce.html

- Burp Extension
  - https://github.com/NetSPI/JavaSerialKiller
  - https://github.com/federicodotta/BurpJDSer-ng-edited
  - https://appsec-labs.com/belch/

- Scanner
  - https://github.com/federicodotta/Java-Deserialization-Scanner

- Code Analyzer
  - https://github.com/mbechler/serianalyzer
Where Were We?

- I can see the traffic, but how do I tamper with it?
- Tunnel “stripped” traffic onto APE!
- We need to inject APE into stcppipe
And... Fail #1

APE got the encrypted data
How Do We Intercept TCP Over SSL?

- Download **TcpCatcher**

- Download TcpCatcher’s root certificate

- Install it as a RootCA in the KeyStore

- Download **KeyStore Explorer**
How Do We Intercept TCP Over SSL?

- Configure TcpCatcher to communication with both, the client and the server.

- TcpCatcher will now serve as a MitM.
<table>
<thead>
<tr>
<th>Id</th>
<th>Date</th>
<th>Size</th>
<th>Directory</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>9:43:13 AM</td>
<td></td>
<td></td>
<td><code>get-root:admin</code></td>
</tr>
<tr>
<td>3</td>
<td>9:43:13 AM</td>
<td></td>
<td></td>
<td><code>admin:secret:Ads Ministrator:5000</code></td>
</tr>
<tr>
<td>4</td>
<td>9:43:23 AM</td>
<td></td>
<td></td>
<td><code>getFile=about.txt</code></td>
</tr>
<tr>
<td>5</td>
<td>9:43:24 AM</td>
<td>165</td>
<td></td>
<td><code>&lt;br&gt;HacmeJava v1.0&lt;br&gt;Hacking and</code></td>
</tr>
<tr>
<td>6</td>
<td>9:43:43 AM</td>
<td>12</td>
<td></td>
<td><code>getId=products</code></td>
</tr>
<tr>
<td>7</td>
<td>9:43:43 AM</td>
<td>128</td>
<td></td>
<td><code>AppSec Labs Secure Laptop:2500:26;App</code></td>
</tr>
<tr>
<td>8</td>
<td>9:43:43 AM</td>
<td>21</td>
<td></td>
<td><code>getUserBalance:admin</code></td>
</tr>
<tr>
<td>9</td>
<td>9:43:43 AM</td>
<td>5</td>
<td></td>
<td><code>5000</code></td>
</tr>
<tr>
<td>10</td>
<td>9:43:43 AM</td>
<td>17</td>
<td></td>
<td><code>getProfile:admin</code></td>
</tr>
<tr>
<td>11</td>
<td>9:43:46 AM</td>
<td>34</td>
<td></td>
<td><code>admin:secret:Ads Ministrator:5000</code></td>
</tr>
<tr>
<td>12</td>
<td>9:44:13 AM</td>
<td>18</td>
<td></td>
<td><code>getViewUsers:true</code></td>
</tr>
<tr>
<td>13</td>
<td>9:44:13 AM</td>
<td>143</td>
<td></td>
<td><code>attacker:attacker789;Mr. Robot:0;false;admin</code></td>
</tr>
<tr>
<td>14</td>
<td>9:44:14 AM</td>
<td>12</td>
<td></td>
<td><code>getProducts</code></td>
</tr>
<tr>
<td>15</td>
<td>9:44:15 AM</td>
<td>128</td>
<td></td>
<td><code>AppSec Labs Secure Laptop:2500:26;App</code></td>
</tr>
<tr>
<td>16</td>
<td>9:44:15 AM</td>
<td>21</td>
<td></td>
<td><code>getUserBalance:admin</code></td>
</tr>
<tr>
<td>17</td>
<td>9:44:16 AM</td>
<td>5</td>
<td></td>
<td><code>5000</code></td>
</tr>
<tr>
<td>18</td>
<td>9:44:16 AM</td>
<td>18</td>
<td></td>
<td><code>getFile=about.txt</code></td>
</tr>
<tr>
<td>19</td>
<td>9:44:16 AM</td>
<td>165</td>
<td></td>
<td><code>&lt;br&gt;HacmeJava v1.0&lt;br&gt;Hacking and</code></td>
</tr>
<tr>
<td>20</td>
<td>9:44:18 AM</td>
<td>5</td>
<td></td>
<td><code>getProfile:admin</code></td>
</tr>
<tr>
<td>21</td>
<td>9:44:18 AM</td>
<td>34</td>
<td></td>
<td><code>admin:secret:Ads Ministrator:5000</code></td>
</tr>
<tr>
<td>22</td>
<td>9:44:31 AM</td>
<td>47</td>
<td></td>
<td><code>updateProfile=admin:secret:Homer Simpson:20;admin:secret</code></td>
</tr>
<tr>
<td>23</td>
<td>9:44:47 AM</td>
<td>13</td>
<td></td>
<td><code>Homer Simpson: true</code></td>
</tr>
</tbody>
</table>

**Welcome Homer Simpson**
What Really Happened?

- It didn’t work!
- Let’s say I got this ...........

Let’s decode:

```java
System.setProperty("javax.net.ssl.trustStore", "ks");
SSLContext ss1 = (SSLContext)SSLSocketFactory.getDefault();
HostnameVerifier hnf = new HostnameVerifier()
{
    public boolean verify(String arg0, SSLSession arg1)
    {
        try
        {
            Certificate cert = arg1.getPeerCertificates()[0];
            X509Certificate x = X509Certificate.getInstance(cert.getEncoded());
            System.out.println(x.getSubjectDN().getName());
            String owasp = "OWASP II";
            X500Name owner = new X500Name(x.getIssuerDN().getName());
            if ((owner.getOrganization().compareTo("HacmeJava, Inc.") == 0) &&
                (owner.getCommonName().compareTo("hacme.java") == 0) &&
                (owner.getLocality().compareTo("localhost") == 0) &&
                (owner.getOrganizationalUnit().compareTo("Insecurity") == 0) &&
                (owasp.compareTo(owner.getState() + " " + owner.getCountry()) == 0))
            {
                return true;
            }
            return false;
        }
    }
```

Please, do not try to MitM me!
Now, That my friends....

- Is SSL pinning!
- The application validates the info of the received (TcpCatcher’s) certificate, against the wanted info, hardcoded in the class.
- Since it’s a self-signed certificate – we could just replace it with our own.
- You passphrase is: “OpenSSL”
  - Create you own self-signed certificate
  - Fill in the required info (found in the class)
  - Install the new certificate in the KS.
  - Should do the trick!
Keytool

1. keytool -keystore clientkeystore -genkey -alias client

   C:\Users\keizer\Desktop\OWASP\keytool -keystore clientkeystore -genkey -alias client
   Enter keystore password:
   Re-enter new password:
   What is your first and last name? [Unknown]: Who me?
   What is the name of your organizational unit? [Unknown]: SWAT
   What is the name of your organization? [Unknown]: FBI
   What is the name of your City or Locality? [Unknown]: USA
   What is the name of your State or Province? [Unknown]: N/A
   What is the two-letter country code for this unit? [Unknown]: US
   Is CN=Who me?, OU=SWAT, O=FBI, L=USA, ST=N/A, C=US correct? [no]: yes

2. keytool -keystore clientkeystore -certreq -alias client -keyalg rsa -file client.csr
TcpCatcher does not support using your own certificate

only on-the-fly ones with a single value.
Other Possible Scenarios

- Checking that it’s “actually” a Root CA.
  - Create a Root CA, using OpenSSL
  - Sign your certificate with the RootCA
  - Import the new Root CA into the default KeyStore (default password: changeme)

- Pinning the Root CA
  - You might need to actually sign your own certificate

- Pinning the intermediate
  - You’ll probably have to patch the code and replace the int. public key with your own.

- Using self-created KeyStore
  - Replace the KeyStore
  - Might require some patching the bypass possible KS validations (e.g. checksum)
What do we do now?

- Let’s hook in runtime!
- Goodbye stcppipe.
- Hello... JavaSnoop!
Day 2: JavaSnoop

- Attaches into any app running over JVM
- Hook methods
- Tamper with parameters, print stacks, etc.
DEMO TIME
What Really Happened?

I WILL FIND YOU
After 5 Hours (on the 2nd day!)

<table>
<thead>
<tr>
<th>Return</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>java.lang.String</code></td>
<td><code>private java.lang.String MainWindow$9 getUsers(java.lang.Boolean)</code></td>
</tr>
<tr>
<td><code>int</code></td>
<td><code>private int MainWindow$9 getUserBalance(java.lang.String)</code></td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>private void MainWindow$1 createUser(java.lang.String)</code></td>
</tr>
<tr>
<td><code>Ljava.lang....</code></td>
<td><code>private java.lang.String[] MainWindow getUserProfile(java.lang.String)</code></td>
</tr>
</tbody>
</table>
SOUNDS LIKE AN EVIL PLAN...
Fail #2

- Server checked the value... 😞

- What next?
- Let’s patch the JAR!
Day 3: Fail #3

// extract jar
# jar -xf myapp.jar

// pack jar
# jar –cvf <desired.jar> <files>

// update jar
# jar -uf <file.jar> <my.class>
Let’s Modify Classes Directly!

To modify class files, you can open a file explorer and navigate to the directory containing the class files. For example, in the image, the directory `editors` contains various `.class` files. You can then modify these files directly to make changes.

Now, how do you modify class files?
Introducing - JBE


```
C:\workspace\JAVA\HacmeJavaClient\MainWindow.class
```

Specific info:

<table>
<thead>
<tr>
<th>Bytecode</th>
<th>Exception table</th>
<th>Misc</th>
<th>Code Editor</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>aconst_null</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>astore_1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>aload_0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ldc_w #550</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>invokespecial #528</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>astore_1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>aload_1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>areturn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>astore_1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>aload_1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>invokevirtual #535</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>aconst_null</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>ldc_w #538</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>invokestatic #540</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>areturn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th>Java Bytecode</th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>ifeq / ifne</td>
<td>if value is (not) 0, branch to offset</td>
</tr>
<tr>
<td>if_icmpeq / if_icmpne</td>
<td>if ints are equal / not equal</td>
</tr>
<tr>
<td>icontst_0 / icontst_1</td>
<td>load int=0/ int=1</td>
</tr>
<tr>
<td>aload_0</td>
<td>load a reference into a local variable 0</td>
</tr>
<tr>
<td>astore_1</td>
<td>store a reference into local variable 1</td>
</tr>
<tr>
<td>dcmpg</td>
<td>compare two doubles</td>
</tr>
<tr>
<td>areturn</td>
<td>return a reference form a method</td>
</tr>
<tr>
<td>fneg</td>
<td>negate a float</td>
</tr>
<tr>
<td>ireturn</td>
<td>return an integer from a method</td>
</tr>
<tr>
<td>ldc</td>
<td>push a constant from a constant pool to the stack</td>
</tr>
</tbody>
</table>
Demo time...
What REALLY Happened?

0. Load something...
1. If null → jump to 14 (const_0)
4. Load something...
5. Get static “ADMIN”
8. Invoke equals(x,y)
11. If equals → jump to 18 (const_1)
14. (no jump) const_0
15. Go to → 19 (return)
18. const_1
19. return
Before...

Username: attacker

Client says: You are not allowed to view users
I’ll Just....
After 2 days and 6 hours

- Username: attacker
  - Password: attacker789
  - Name: Mr. Robot
  - Balance: -9999
  - Is an Admin: false
- Username: admin
  - Password: secret
  - Name: Gate Bills
  - Balance: 5000
  - Is an Admin: true
- Username: victim
  - Password: victim789
  - Name: John Doe
  - Balance: 2000
  - Is an Admin: false
We needed to create a MitM, to serve as a proxy between the original MitM and the client, replacing its on-the-fly certificate with our own certificate.

So, now we have:
For the first time!

Introducing.... ProKSy

-- What with the “KS”?
-- Stands for KeyStore :P

https://github.com/nu11p0inter/ProKSy
Demo Time!

LIVE DEMO

I ALSO LIKE TO LIVE DANGEROUSLY
The Moral of the Story

- What did not work for me, might work for you

- Java – might not (fun) “writable”, but “readable”

- Never give up - there’s no such thing as “unbreakable”

- We love memes

- Download ProKSy!
APE - TCP (.net) Proxy for Hooking
https://appsec-labs.com/advanced-packet-editor/

ProKSy - TCP/SSL Proxy for SSL Pinning
https://github.com/nu11p0inter/ProKSy/

JavaSnoop - Java Runtime Manipulation
http://www.aspectsecurity.com/tools/javasnoop

JBE/reJ - Java ByteCode Editing
http://set.ee/jbe/
https://sourceforge.net/projects/rejava
Thank you! see you @ OWASP IL 2017