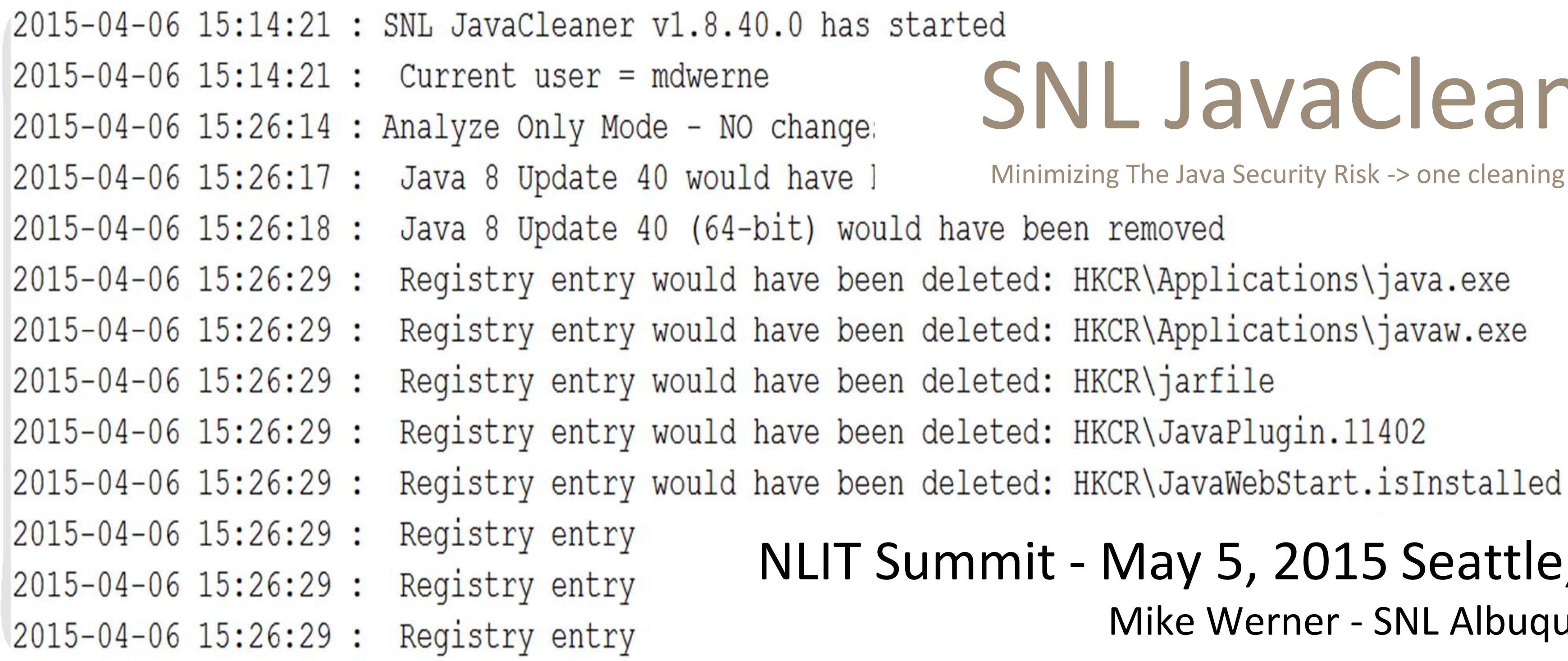
SAND2015-3349C

Exceptional service in the national interest

Sandia National Laboratories

SNL JavaCleaner

Minimizing The Java Security Risk -> one cleaning at a time



```
2015-04-06 15:14:21 : SNL JavaCleaner v1.8.40.0 has started
2015-04-06 15:14:21 :  Current user = mdwerne
2015-04-06 15:26:14 : Analyze Only Mode - NO change
2015-04-06 15:26:17 :  Java 8 Update 40 would have
2015-04-06 15:26:18 :  Java 8 Update 40 (64-bit) would have been removed
2015-04-06 15:26:29 :  Registry entry would have been deleted: HKCR\Applications\java.exe
2015-04-06 15:26:29 :  Registry entry would have been deleted: HKCR\Applications\javaw.exe
2015-04-06 15:26:29 :  Registry entry would have been deleted: HKCR\jarfile
2015-04-06 15:26:29 :  Registry entry would have been deleted: HKCR\JavaPlugin.11402
2015-04-06 15:26:29 :  Registry entry would have been deleted: HKCR\JavaWebStart.isInstalled
2015-04-06 15:26:29 :  Registry entry
2015-04-06 15:26:29 :  Registry entry
2015-04-06 15:26:29 :  Registry entry
```

NLIT Summit - May 5, 2015 Seattle, WA

Mike Werner - SNL Albuquerque

U.S. DEPARTMENT OF ENERGY
NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

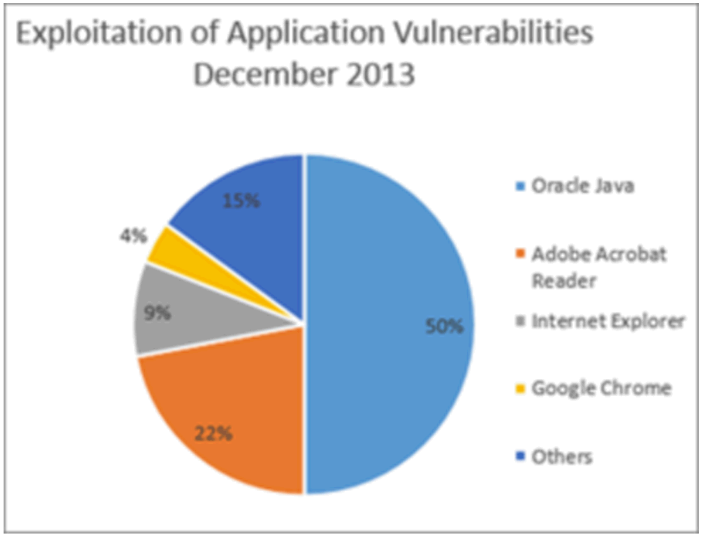
I am NOT a Java™ hater, just a security activist…

- **Significant Advantages of Java Language** *
  - Java is easy to learn.
  - Java is object-oriented.
  - Java is platform-independent.
  - Java is distributed.
  - Java is robust.
  - Java is multithreaded.
  - Java is secure.

***http://www.streetdirectory.com/travel_guide/114362/programming/most_significant_advantages_of_java_language.html**

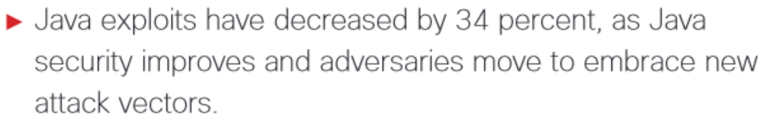
Java Exploitation…an improving situation?

Unpatched Java is easily the most exploited vulnerability, accounting for 50 percent or more of all successful attacks -- and they've held that leadership position since 2012. Prior to that, Java was No. 2 behind Acrobat Reader.



Java made up 50 percent of the exploits discovered in December 2013.

In 2015 Cisco released their Annual Security Report* which states:



▶ Java exploits have decreased by 34 percent, as Java security improves and adversaries move to embrace new attack vectors.

"Of the top 25 vendor- and product-related vulnerability alerts from January 1, 2014, to November 30, 2014, only one was Java-related."

The trend is going in the right direction for Java exploits, but a proactive stance is still necessary.

*http://www.cisco.com/web/offers/pdfs/cisco-asr-2015.pdf

Exploits are down, but patching isn’t up.

According to a new report from Secunia, “48 percent of users aren’t running the latest, patched versions of Java”. Unpatched Java continues to pose the biggest risk to US desktop security.

To minimize the risk Java presents, the problem must be met on multiple fronts…no one method will keep your enterprise safe.

A Corporate Strategy…three possible methods of protection

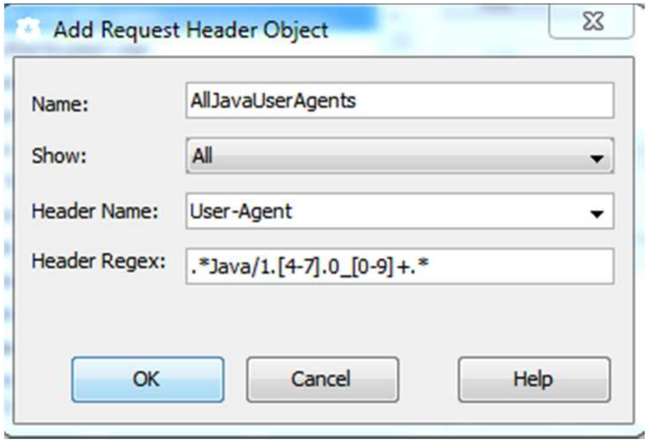
- Block certain Java "User Agent Strings" at the perimeter
  - Could be changed to "ALL Java UAS" during a security event
- Oracle/Java native solution to customers who have no choice but to run older versions of the JRE
  - Oracle/Java Deployment Rule Sets – (preferred)
- Cleanup old (unused or vulnerable) Java remnants from a Windows desktop
  - Primary function of the SNL JavaCleaner

A few other options:

- Patch Java (assuming a patch is available/takes time)
- Disable the browser plugin (for customers who run local Java apps)
- Uninstall Java altogether (might work for some, not for all)

User Agent Strings

- When a browser visits a website through a firewall, a bit of textual data is sent along that identifies both the browser version and certain system details…the version of the Java plugin is one of those details.
- An example firewall rule to block Java User Agents v1.4 thru v1.7 might look something like this:



Add Request Header Object

Name: AllJavaUserAgents

Show: All

Header Name: User-Agent

Header Regex: .*Java/1.[4-7].0_[0-9]+.*

Deployment Rule Sets

- “The Deployment Rule Set feature is for enterprises that manage their Java desktop environment directly, and provides a way for enterprises to continue using legacy business applications in an environment of ever-tightening Java applet and Java Web Start application security policies.“
  - Essentially they are a set of rules defined in an XML file that is then packaged and signed. The rules are processed sequentially until a match is found. For example:

```
<ruleset version="1.0+">
    <rule>
        <id location="http://*.example.org" />
        <action permission="run" />                               /// Allows all Java apps to run on
    </rule>                                                            *.example.org

    <rule>
    <id location="http://*.example.com" />
        <action permission="run" version="SECURE-1.7" />          /// Allows any version of Java 7 to run
    </rule>                                                            on *.example.com

    <rule>
        <id />,
        <action permission="block" />                             /// Blocks Java at all other websites
    </rule>
</ruleset>
```

Another technology that can minimize potential Java risks

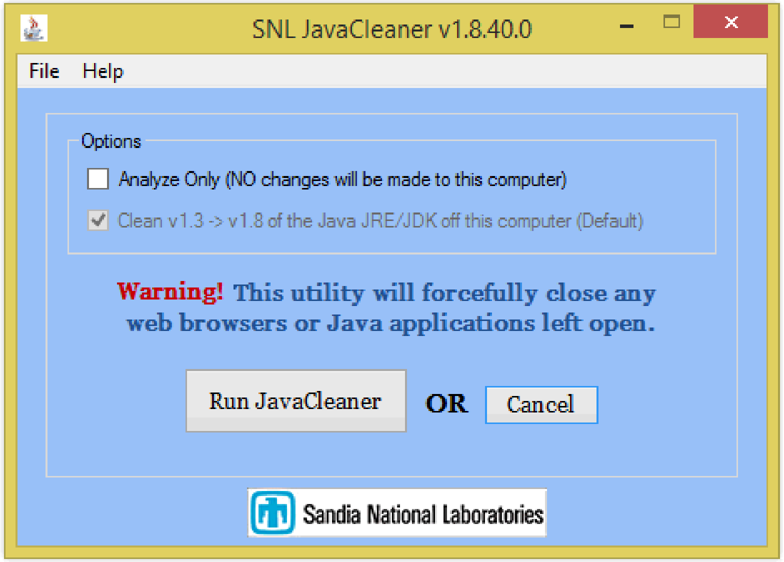
- Microsoft App-V
  - App-V allows an organization to "Sequence" a browser and a specific version of Java into a virtual environment that can be run, essentially sandboxed, from any would be attacker.
  - Due to the relative complexity of using App-V, in some organizations, Deployment Rule Sets are often preferred.

Where it all started…the history behind the JavaCleaner

- Four years ago, it came to our attention that a stand alone network was littered with remnants from past JRE and JDK installations.
- The owner of that network was looking for a way to "Clean-Up" those Java remnants and to reduce the networks attack surface.
- We did some research and found utilities that already existed to clean Java remnants, but none performed a thorough enough cleaning for our needs.
- As a result of that research, the development on the "~~SNL JavaUpdater~~", "~~SNL JavaInstaller~~", "SNL JavaCleaner" began.
- Today the SNL JavaCleaner can be used to ***quickly*** address potentially vulnerable machines in any Windows environment.

The SNL JavaCleaner

Screenshots…

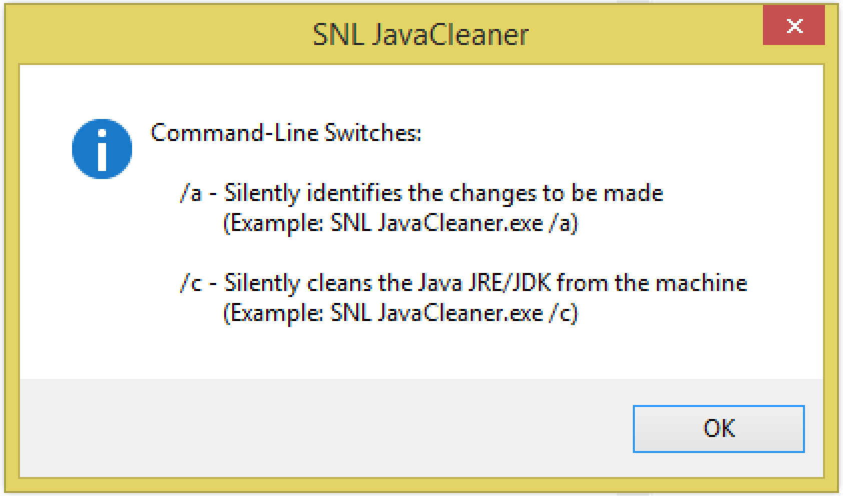


Options

□ Analyze Only (NO changes will be made to this computer)

☑ Clean v1.3 -> v1.8 of the Java JRE/JDK off this computer (Default)

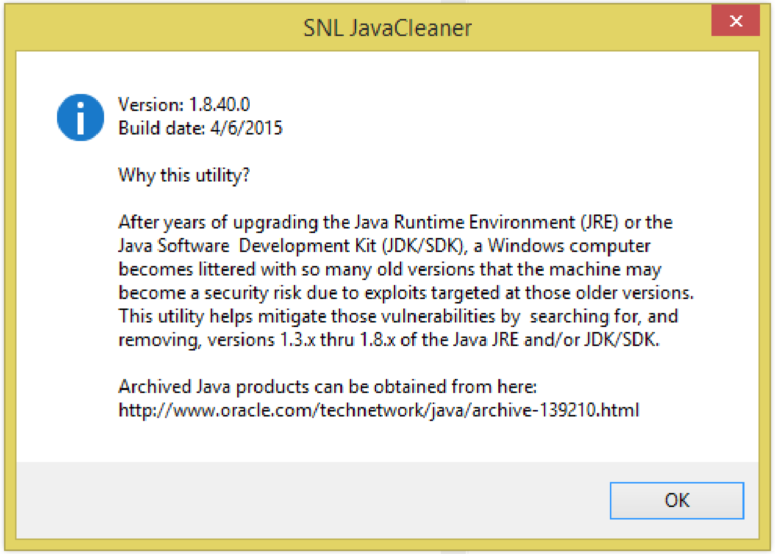
Warning! **This utility will forcefully close any web browsers or Java applications left open.**



Command-Line Switches:

/a - Silently identifies the changes to be made
(Example: SNL JavaCleaner.exe /a)

/c - Silently cleans the Java JRE/JDK from the machine
(Example: SNL JavaCleaner.exe /c)



Version: 1.8.40.0
Build date: 4/6/2015

Why this utility?

After years of upgrading the Java Runtime Environment (JRE) or the Java Software Development Kit (JDK/SDK), a Windows computer becomes littered with so many old versions that the machine may become a security risk due to exploits targeted at those older versions. This utility helps mitigate those vulnerabilities by searching for, and removing, versions 1.3.x thru 1.8.x of the Java JRE and/or JDK/SDK.

Archived Java products can be obtained from here:
http://www.oracle.com/technetwork/java/archive-139210.html

What's happening in the app?

- A broad overview of what's happening behind the scenes…
  - If OS is Win7 or Greater, set a **System Restore Point

    **Only contains a subset of the objects cleaned by the JavaCleaner
  - Close all browsers and Java processes
  - Uninstall identified JRE installations via MSI uninstall codes
  - Uninstall identified JDK installations via MSI uninstall codes
  - Uninstall unidentified Java MSI based Apps
  - Search for, and uninstall, non-MSI based Apps
  - Delete leftover registry keys (identified)
  - Delete leftover files on primary hard drive
  - Delete the Java Cache at the following locations
    - \AppData\LocalLow\Sun
    - \AppData\LocalLow\Oracle\Java
  - Finalize session and the opening of basic log file

Demo of the SNL JavaCleaner

- Modes of Operation:
  - Analyze Only Via GUI
    - Used to identify what files, folders and registry entries will be deleted before running the cleaner. A simple log file is automatically opened. Verbose log file at root of primary HD.
  - Clean VIA GUI
    - Creates a **System Restore Point, and then performs a full cleaning. Log files created and opened as above.
  - Analyze Only Via Command Line "SNL JavaCleaner.exe /a"
  - Clean Via Command Line "SNL JavaCleaner.exe /c"
    - Both modes run silently with no log files being <u>opened</u>, used by IT Administrators

**Only contains a subset of the objects cleaned by the JavaCleaner

QUESTIONS?

- Deployment Rule Sets by Example
  - https://blogs.oracle.com/java-platform-group/entry/deployment_rule_set_by_example
- Administering Java Whitelists
  - http://ephingadmin.com/administering-java
- Tips for using Java securely
  - https://www.java.com/en/download/faq/security-tips.xml
- Blocking the User Agent String at the Perimeter
  - http://zombietango.com/blog/2012/08/blocking-vulnerable-java-requests-at-the-network-layer
- Sequencing Java with App-v
  - http://packageology.com/tag/java

For assistance with the SNL JavaCleaner, contact Mike Werner (mdwerne@sandia.gov)