Trusting Virtual Trust

Jeremy Powell, Trupti Shiralkar

the information security provider

Agenda

- The Java Virtual Machine
- Unmeasured Trust
- Java's Assurance

The Java Virtual Machine

A Few Quick Disambiguations

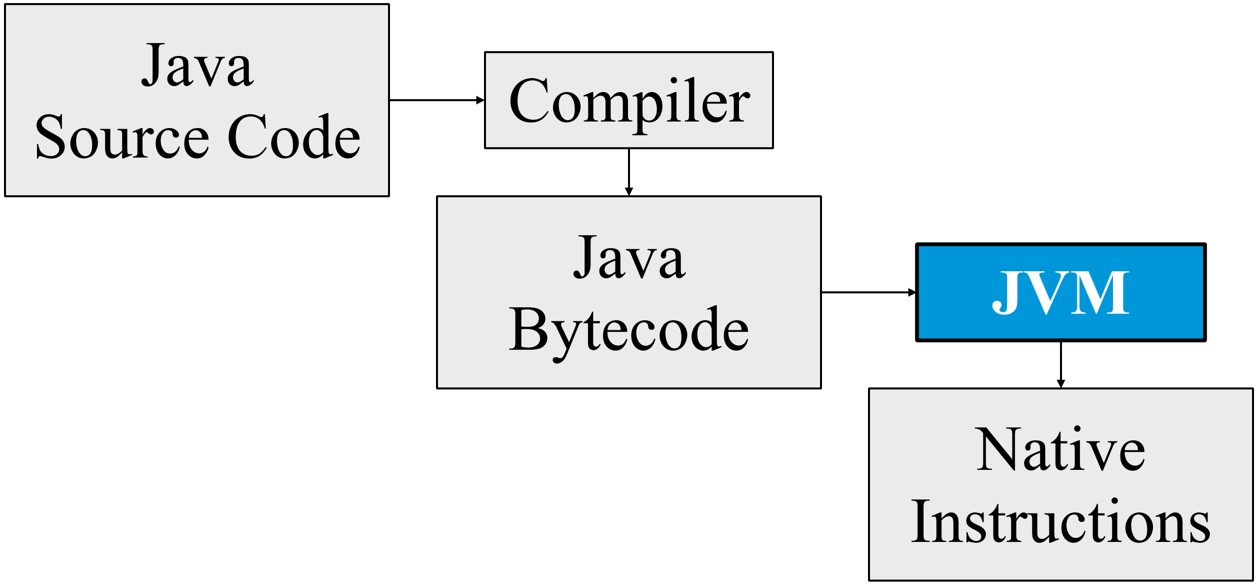
- Overloaded terms
- “Virtual Machine” could mean...
  - A system virtual machine
    - Xen
    - HyperV
    - VMWare
  - A process (application) virtual machine
    - Java
    - Common Language Runtime
- We mean a “process virtual machine”

A Few Quick Disambiguations

- Overloaded terms
- “Java” could mean...
  - ...a spoken language
  - ...an island
  - ...coffee
  - ...a programming language
  - ...an application platform
  - ...a virtual machine
- We mean “the Java Virtual Machine”
- ...usually

Just what is the Java Virtual Machine?

- Developed by Sun Microsystems
- Interprets and runs bytecode
- Virtualizes an abstract processor



Java Source Code

Compiler

Java Bytecode

JVM

Native Instructions

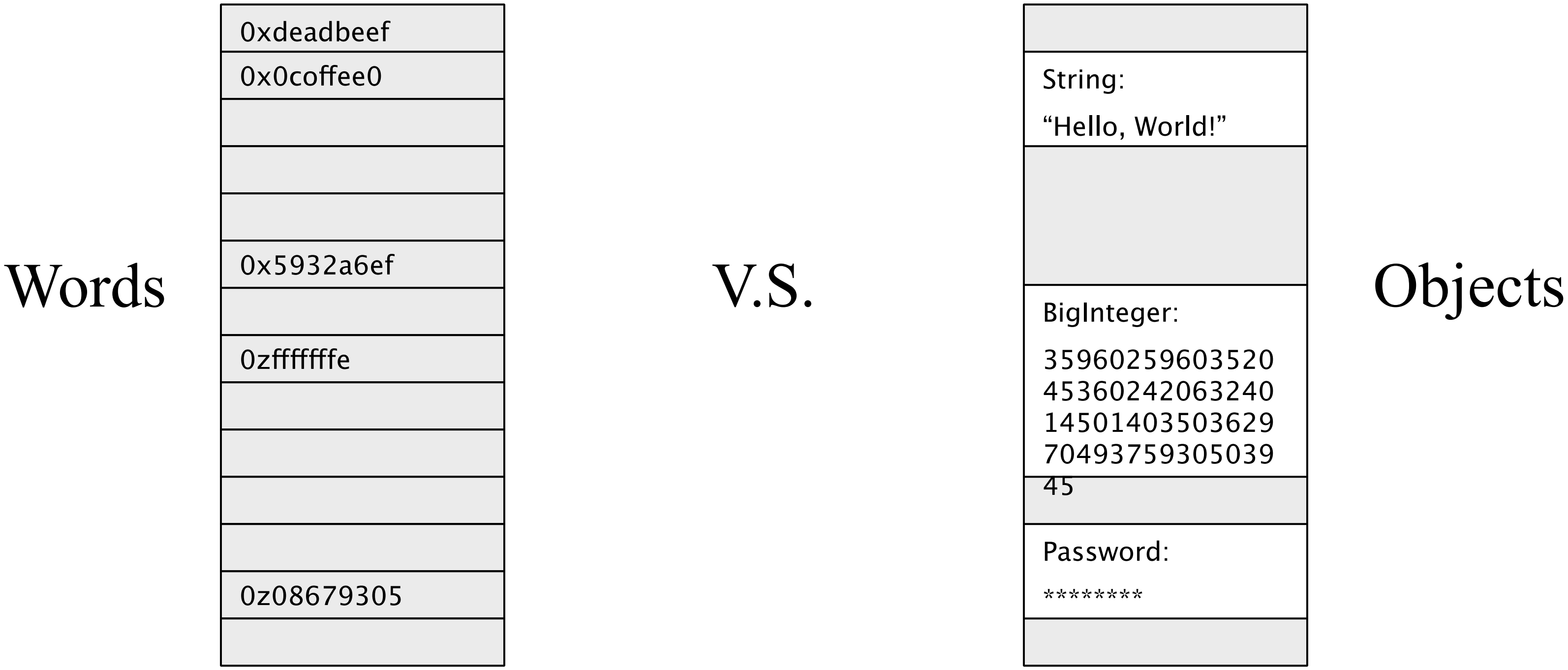
Bytecode

- Looks very much like native assembly code

```
0:      iconst_1
1:      istore_1
2:      iconst_1
3:      istore_2
4:      iload_2
5:      bipush      100
7:      if_icmpge   20
10:     iload_1
11:     iload_2
12:     imul
13:     istore_1
14:     iinc        2, 1
17:     goto        4
20:     getstatic   #2
23:     iload_1
24:     invokevirtual           #3
27:     return
```

Typed Memory

- C / C++ – Memory organized into words
- Java – Memory organized into objects



Words

0xdeadbeef

0x0coffee0

0x5932a6ef

0zfffffffe

0z08679305

V.S.

String:

“Hello, World!”

BigInteger:

35960259603520
45360242063240
14501403503629
70493759305039
45

Password:

****

Objects

Popularity

- Web application servers
- Browser applets
- User Applications
- Smart card platforms
- Cell phones
- Embedded systems
- Game consoles
- Scientific computing

What's the big deal?

- Write Once, Run Anywhere
- Automatic memory management
- Already installed on most computers
- Generous standard libraries
- Heavily specified – reliable behavior
- Free
- Secure

Unmeasured Trust

@sec the information security provider

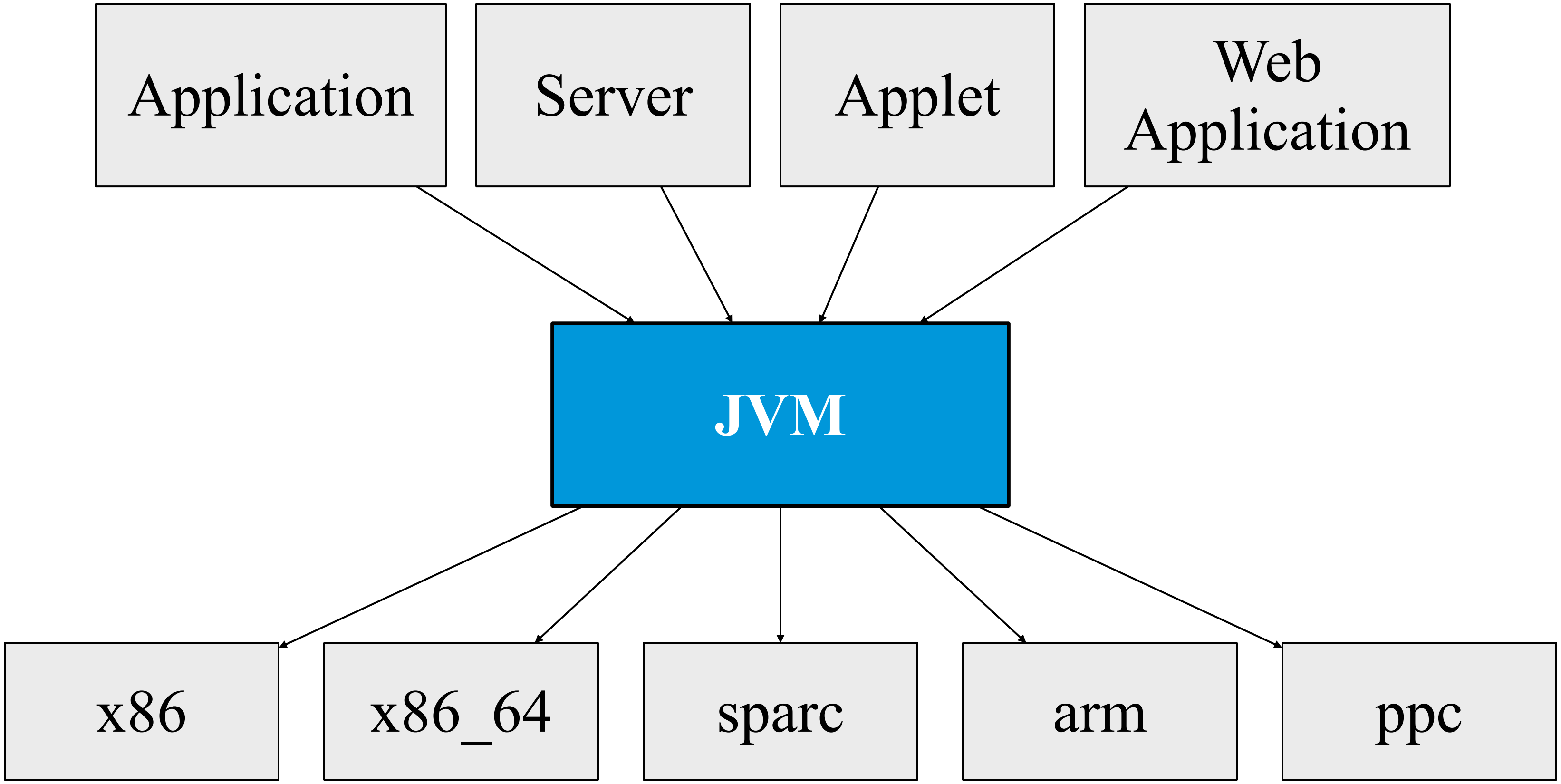
The Need for Measured Assurance

- Measuring assurance answers...
  - What security does Java really provide?
  - How sound is its design?
  - How correct is its implementation?
  - How does one use Java securely?
- Without measured assurance, we take unnecessary risk
- The risk is growing. Fast.

Revisiting Portability

- Write Once, Run Anywhere
- Less code written
  - Eliminates system and hardware nuances
  - Reduces analysis effort
  - Wide deployment of the same code
- Wide deployment means single point of failure

A Single Point of Failure



© 2009 atsec information security

Can't sleep? Count Java implementations!

- CEE-J
- Excelsior
- J9
- JBed
- JamacaVM
- Jblend
- JRockit
- Apple's MRJ
- MicroJvm
- MS JVM
- Blackdown Java
- C virtual machine
- Gemstone
- Golden Code Development
- Intent
- Novell
- NSIcom CrE-ME
- ChaiVM and MicrochaiVM
- Hotspot
- AegisVM
- Apache Harmony
- CACAO
- Dalvik
- IcedTea
- IKVM.NET
- Jamiga
- JamVM
- JC
- Jelatine JVM
- JESSICA
- Jikes RVM
- JNode
- JOP
- Juice
- Jupiter
- JX
- Kaffe
- leJOS
- Maxine
- NanoVM
- SableVM
- and more...

http://en.wikipedia.org/wiki/List_of_Java_virtual_machines

Java's Assurance

The Elephant in the Room

- So, if...
  - Java is very popular
  - Used widely
  - And has many implementations....
- Why hasn't Java been CC evaluated?
  - Seemingly less vulnerabilities than C or C++
  - Not tied to a bottom line
  - Uncertain what security functions are provided

Java Prevents Common C/C++ Problems

- No stack smashing
- No heap corruption
- No format string attacks
- No reference forging
- No type confusion

(...all prevented by the type system)

Developer Incentives Outweigh Costs

- No mandatory evaluation of Java
- Evaluated Java returns same as Unevaluated Java – Nothing
- CC Evaluation is expensive
- Partial Motivation – first implementations evaluated may get edge on market

Java and Security

- The JVM provides memory safety
  - Enforcement of language security (public/ private)
  - References cannot be forged
    - Prevents type confusion
    - Provides capability access control
  - Stack and heap corruption prevented
- Cryptography (JCA)
- Sandboxing access control (JAAS)

A Smattering of SFRs

- User Data Protection
  - Capabilities
  - Access Control Lists
  - Zeroed memory on allocation
- Cryptography
- Identification and Authentication
- Auditing

Next steps

- Understand the necessity
- Write a protection profile
  - Defines the Process Virtual Machine security problem
  - Demonstrates demand
- Evaluate an implementation
  - Java is well studied in academia – higher EALs may be possible
  - Produces ECG – How to use Java securely

Thank You!

Jeremy Powel – jeremy@atsec.com
Trupti Shiralkar – trupti@atsec.com