Detecting Malicious Java Code Using Virtual Machine Auditing.

S. Soman, C. Krintz, and G. Vigna. 12th USENIX Security Symposium, 2003

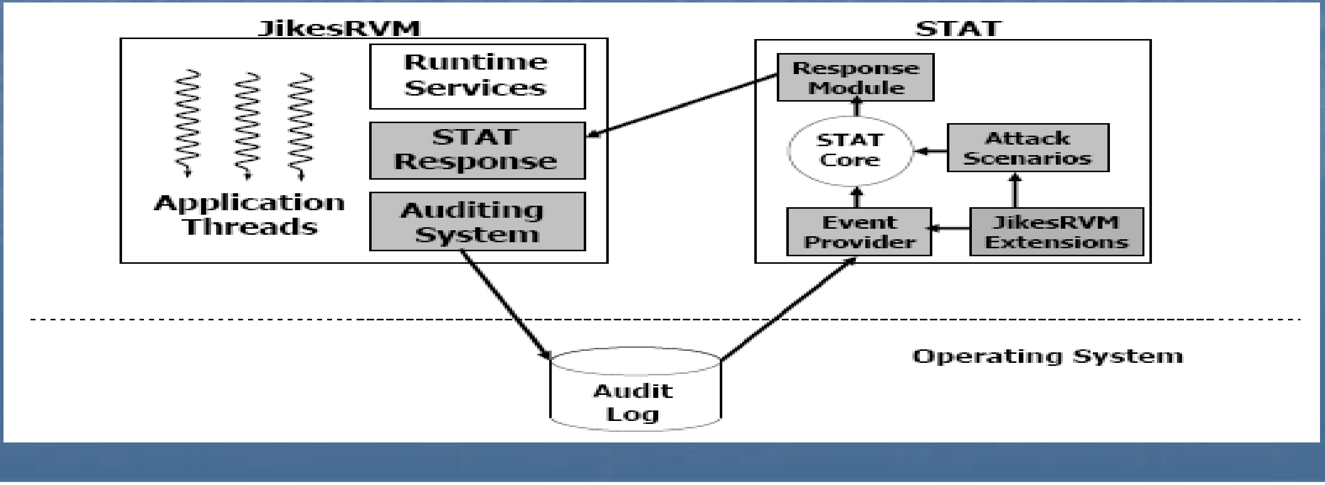
Presented by Jinho Lee

Summary

- Reports on development of a Java Intrusion Detection Tool that operates at JVM layer

- The Problem
  - In Java, applications are run as threads instead of processes within a single JVM process
  - Traditional OS-level ID does not work because of coarse granularity! Ie. Will not capture what individual threads are doing

- The Solution
  - Audit facility for JikesRVM. Extension of existing JikesRVM
  - Leverages the STAT(State Transition Analysis Technique) framework



System Features

- The system records JVM-level events
  - Class Events
  - System Call Events
  - JNI Events
  - Thread Interaction Events

- The system detects using a signature-based approach, assuming a server execution model,
  - Unauthorised Access Detection
  - Harmful Inter-Thread Communication
  - Detecting Network Scans
  - Detecting Transfer of Privileged Information

Appreciative Comment

- It is a novel approach that will complement other ID techniques that operate at different levels. The paper shows that it is important to do ID at JVM-level
  - None of the events except system calls can be recorded at the OS-level
  - Even if it was possible, at OS-level, alerts will simply say the offender is the JVM. Shut down the whole JVM killing the whole server?
  - At application-level it may be possible to detect the 4 events in theory, but in practice it is not feasible to instrument every application. Much easier to instrument the JVM centrally.

Critical Comment 1

- Proposes a taxonomy of extant ID approaches
  - Anomaly Detection – look for deviation from statistical profile of normal behavior (90/10 rule)
  - Formal Specification – look for deviation from formal spec of correct behavior
  - Signature-based ID – look for traces of known attacks
- Proposed taxonomy writes off anomaly detection as something of little merit without appreciating its strengths

Critical Comment 1 contd..

	Pros	Cons
Anomaly Detection	None?!!	▪Difficult to create reliable model ▪High false negative & high false positives
Formal Specification	Little false positives	Generation of specification requires considerable effort and access to source code
Signature-based	Very effective and produce little false positive	Attack signatures must be updated often

Critical Comment 1 contd..

- Anomaly detection has greater chance of detecting previously unknown attacks but the paper does not mention it. In fact all the references on Anomaly Detection that the paper refers to emphasize this but this is ignored in the paper. Moreover examples of promising prototypes can be found in literature.

  One really basic example is

  - Method call profiling method to detect virus and backdoor attacks
    - Strange Brew – First Java virus, modifies infected files' constructor to call itself
    - HTTP backdoor attacks calling arbitrary commands
  - Much more can be done with more sophisticated profiles such as memory behaviour, whole program execution paths.

Critical Comment 1 contd..

- Authors simply claim “signature-based approach is very effective”. However it is also dependent on creating accurate attack signatures to be effective just like good model is necessary for anomaly detection to be successful.

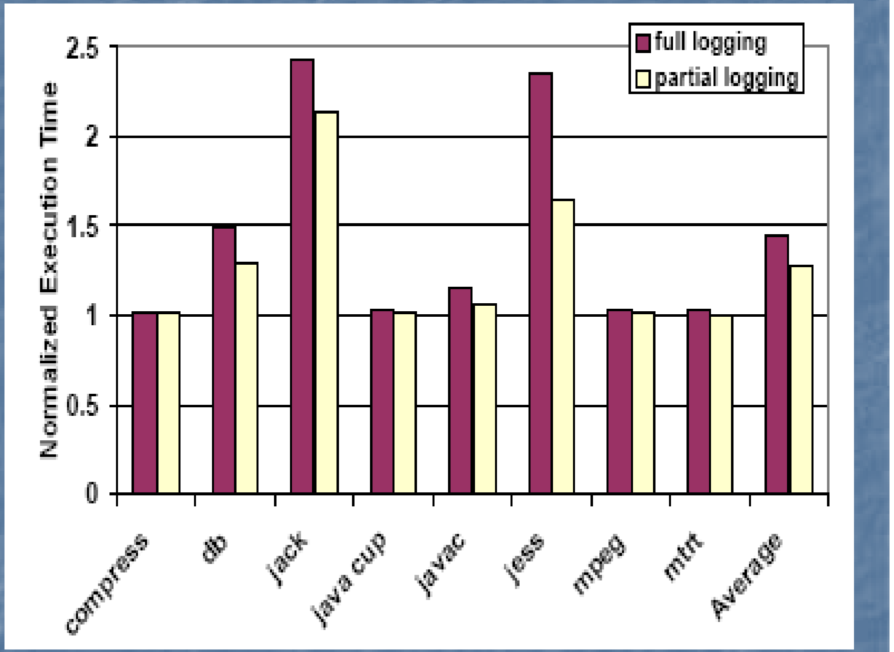
- Signature-based approach has relatively less false-positives but is potentially more susceptible to unknown attacks. Rather than pointing out this fundamental weakness, authors make it sound like updating the signature set is be all end all.

- Is it ethical for an academic to be selective in presenting existing knowledge?

Critical Comment 2

- Used 8 common benchmark applications with and without auditing

- 3 out of 8 benchmark apps showed slowdown factor of about 2 Can we say ‘very little overhead in most cases’??



Question!!

- What are your reasons for using and not using intrusion detection tools? What is your biggest issue with them?

- As another research direction, how does human brain work to detect an intrusion? e.g. break-in to a house for theft. Can you see any similarity/difference between our way of detection and the three approaches ? Do you think there is any promise in applying the human model in future ID research?