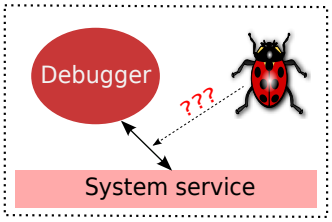
Virt-ICE: invisible debugger for malware analysis

Nguyen Anh Quynh, Kuniyasu Suzaki - RCIS, AIST

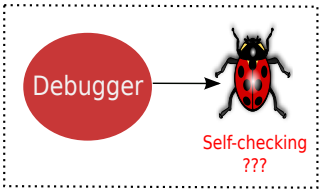
Problems of debugger against malware

Malware can detect debugger



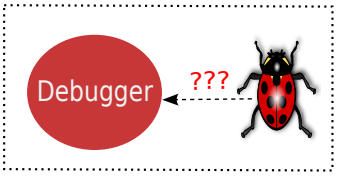
Detect debugger using system service to handle debug events (ex: Swen)

1



Detect debugger modifying malware process (MyDoom)

2



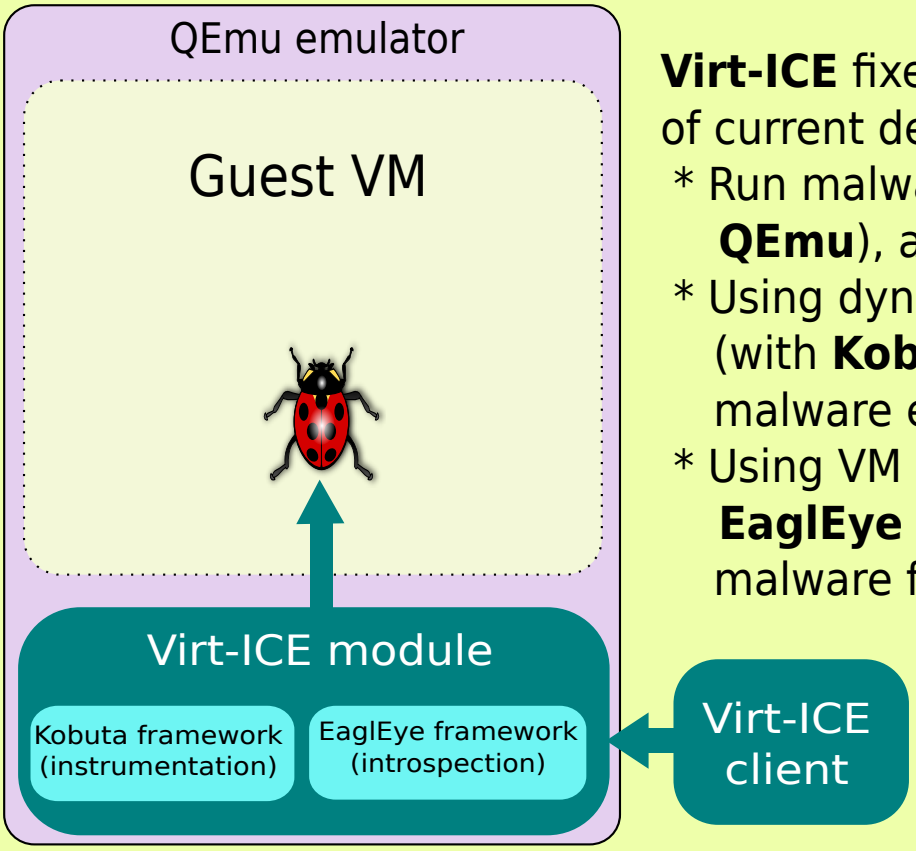
Detect debugger's presence (Conficker.C)

3

Malware can tamper with debugger

Attack or modify system to render debugger useless (Rustock.C)

4



Virt-ICE fixes the outstanding problems of current debuggers against malware

* Run malware inside guest VM (using **QEmu**), and analyze it from outside
* Using dynamic binary instrumentation (with **Kobuta** framework) to intercept malware execution
* Using VM instrospection (thanks to **EaglEye** framework) to inspect malware from emulator layer

Features

* Invisible to malware
  * Not use any system service → 1 fixed
  * Not modify malware process → 2 fixed
  * Stay outside of malware's domain → 3 fixed
* Tamper-resistant against malware
  * Stay out of reach of malware → 4 fixed
* Provide convenient tools to improve efficiency of malware analyst (API monitoring, tainting analysis, ...)