Relative Vulnerability: An Empirical Assurance Metric

Crispin Cowan, PhD
CTO, Immunix

The Problem With Current Security Assesments

- On one end: highly formal assurance
  - Common Criteria:
    - *Extremely* expensive: about $1M for initial assessment
    - Meaningless answer:
      - 3 bits: EAL0-7
      - A “high assurance” OS can be rooted the next day by a buffer overflow
      - So how much of this is “enough”?
- On the other end: Bugtraq Whack-a-mole
  - Chronic chain of “gotcha” vulnerability disclosures
  - Each disclosure tells you that you are *not* secure, but when you *are* secure is undecided
  - Not very helpful :)

Assessing the Assurance of Retro-Fit Security

- Commodity systems (UNIX, Linux, Windows) are all highly vulnerable
  - Have to retrofit them to enhance security
- But there are lots of retrofit solutions
  - Are any of them effective?
  - Which one is best?
  - For my situation?
- Instead of “How much security is enough for this purpose?”, we get “Among the systems I can *actually* deploy, which is most secure?”
  - Consumer says “We are only considering solutions on FooOS and BarOS”
  - Relative figure of merit helps customer make informed, realistic choice

Proposed Benchmark: Relative Vulnerability

- Compare a “base” system against a system protected with retrofits
  - E.g. Red Hat enhanced with Immunix, SELinux, etc.
  - Windows enhanced with Entercept, Okena, etc.
- Count the number of known vulnerabilities stopped by the technology
- “Relative Invulnerability”: % of vulnerabilities stopped

Can You Test Security?

- Traditionally: no
  - Trying to test the negative proposition that “this software won’t do anything funny under arbitrary input”, I.e. no surprising “something else’s”
- Relative Vulnerability transforms this into a positive proposition:
  - Candidate security enhancing software stops at least foo% of unanticipated vulnerabilities over time

Vulnerability Categories

Local/remote: whether the attacker can attack from the network, or has to have a login shell first

Impact: using classic integrity/privacy/availability

- **Penetration**: raise privilege, or obtain a shell from the network
- **Disclosure**: reveal information that should not be revealed
- **DoS**: degrade or destroy service

Impact

- Lower barriers to entry
  - Anyone can play -> more systems certified
- Real-valued result
  - Instead of boolean certified/not-certified
- Easy to interpret
  - Can partially or totally order systems
- Empirical measurement
  - Measure *results* instead of *adherence to process*
- Implementation measurement
  - CC **can't** measure most of the Immunix defenses (StackGuard, FormatGuard, RaceGuard)
  - RV can measure their efficacy

IMMUNIX®

Issues

- Does not measure vulnerabilities *introduced* by the enhancing technology
  - Actually happened to Sun/Cobalt when they applied StackGuard *poorly*
- Counting vulnerabilities:
  - When l33t d00d reports “th1s proggie has zilli0ns of bugs” and supplies a patch, is that one vulnerability, or many?
- Dependence on exploits
  - Many vulnerabilities are revealed *without* exploits
    - Should the RV test lab *create* exploits?
    - Should the RV test lab *fix* broken exploits?
  - Probably ***yes***
- Exploit success criteria
  - Depends on the test model
  - Defcon “capture the flag” would *not* regard Slammer as a successful exploit because payload was not very malicious

Work-factor View

- Assume that well-funded attacker can penetrate almost any system eventually
- The question is “How long can these defensive measures resist?”
- RV may probabilistically approximate the work factor to crack a system
  - foo% of native vulnerabilities are not actually exploitable
  - Therefore foo% of the time a well-funded attacker can’t get in that way
  - Attacker takes foo% longer to get in???

Lessons Learned the *Hard* Way

- Security advisories lie
  - often incomplete, or wrong
- Published exploits are mostly broken, deliberately
- Compiled-in intrusion prevention like StackGuard makes it *expensive* to determine whether the defense is really working, or if it is just an incompatibility
  - Also true of diversity defenses

Technology Transfer

- ICSA Labs
  - traditionally certify security products (firewalls, AV, IDS, etc.)
  - no history of certifying secure operating systems
  - interested in RV for evaluating OS security
- ICSA issues
  - ICSA needs a pass/fail criteria
  - ICSA will not create exploits