Vulnerability Detection in Core Internet Systems

ICANN, San Francisco, March 2011

Dr. Andrzej Bartosiewicz, CEO
andrzej@yonita.com
phone: +1 650 2493707

Patrycja Wegrzynowicz, CTO
patrycja@yonita.com
phone: +1 650 2493708

YONITA

Internet Infrastructure

- Routers, Switches, Firewalls etc
- DNS servers
- Registry Infrastructure
  - Registration systems, WHOIS, exports to DNS, monitoring
  - Payments, Accounting, CRM etc.
- Registrars / Resellers Infrastructure
  - Registration systems,
  - Payments…
- Hosting and SaaS services
- etc.

YONITA

DNS (safety) Facts

- DNS is very well protected
  - Resolvers well tested by many parties all over the years
  - Anycast solutions implemented
  - Extensive monitoring solutions implemented
- Due to extensive work on DNS infrastructure, it's (very) difficult to exploit it today.

How to attack the core internet infrastructure?

- Attack DNS itself? No way, but…
- By exploiting Registries' B2B systems
  - Directly on registration systems
  - Through payment or customer care systems
  - Social engineering
- By exploiting Registras' B2C systems
  - Registrars (often) do not invest enough
  - B2B systems are easier to protect than B2C

YONITA

Registry and Registrar Systems

It’s all about software!

YONITA

Application Weaknesses

- Bugs
  - Correctness (internal incorrect execution)
  - Security vulnerabilities (external attacks)
- Bad practices
  - Performance bottlenecks (certain characteristics reveal useful information to attackers and/or allow for certain attacks)
  - Low maintanability (in long term leads to more bugs)
- Backdoors
  - 3rd party (e.g., illegal access/data gathering)

Weaknesses – Sources

- Lack of knowledge
  - Developers not aware about security (and other) issues
  - High rotation of developers
  - Many freshmen developers
  - Changing technologies
- Complexity of software development
  - Changing requirements
  - Size of a codebase
  - Growing technology stack
- Malice or laziness

Traditional Best Practices

- Education
  - Advanced training – dedicated, expert training courses, coaching sessions, workshops.
- Software development process
  - More and more tests: Test-Driven Development (TDD), unit tests, integration tests, performance tests.
  - Continuous integration (and automated execution of tests).
- Independent verification
  - Audits: blackbox testing, code audits

Is This Enough?

Problems

(1) Size does matter! (complexity of software development)

(2) Program testing can be used to show the presence of bugs, but never to show their absence!

-- E. Dijkstra

Solution

Think? Why think! We have computer to do that.

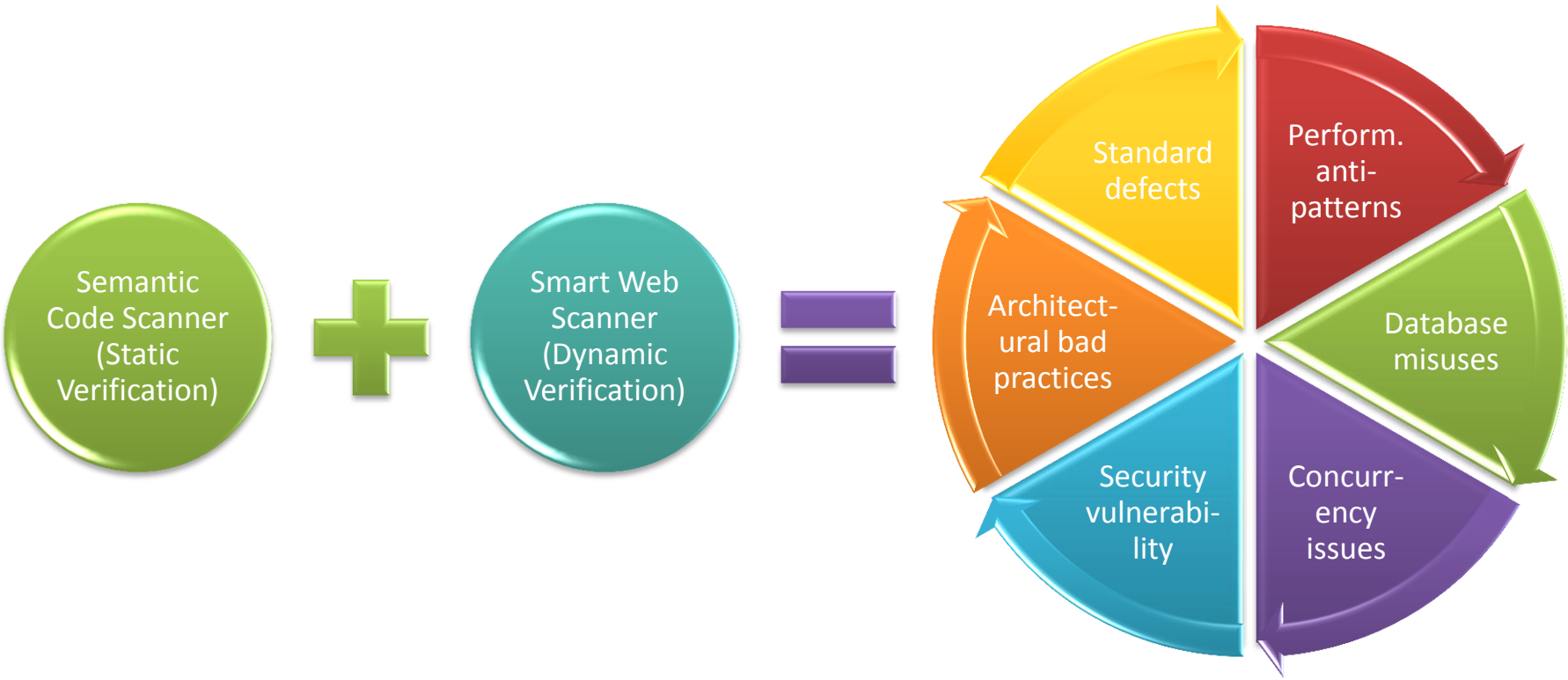
-- J. Rostand

YONITA

Automated Tools

- Automated testers
  - Generate test data and test suites
  - Scan applications e.g., web applications
- Tools to analyze sources, binaries without execution
  - Static analysis
- Tools to analyze execution of a program
  - Dynamic analysis

Yonita Solution



YONITA

Smart Web Scanner

Authentication, authorization, and session management

Input and output validation

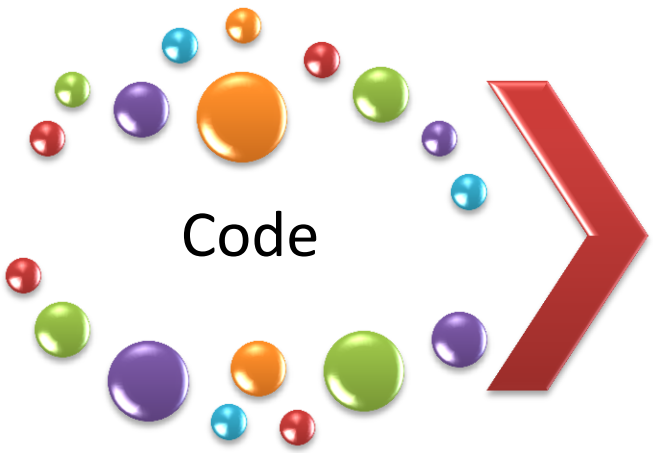
- Injections (e.g., script injection, OS command injection, SQL injection, CRLF injection)
- Cross Site Scripting
- Cross Site Request Forgery
- Forward and Redirect mechanisms
- Content spoofing
- Buffer overflow
- Direct object references

(D)DoS attacks

- Generates test suites
  - Automatically discovers the structure of web applications
  - Analysis of HTML/JS
  - Heuristics based on dictionaries, thesauruses, ontologies
  - Preconfigured forms
- Generates test data
  - To cover various vulnerabilities

YONITA

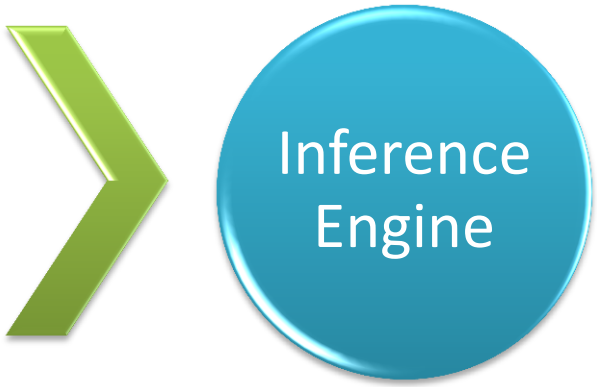
Semantic Code Scanner



- Sourcecode
- Bytecode

Semantic Analyzer

- Structure
- Call flow
- Data flow



- Deductive database
- Stores metamodel
- Infers about defects

YONITA

Summary

While securing Internet infrastructure, don’t forget about software!

YONITA