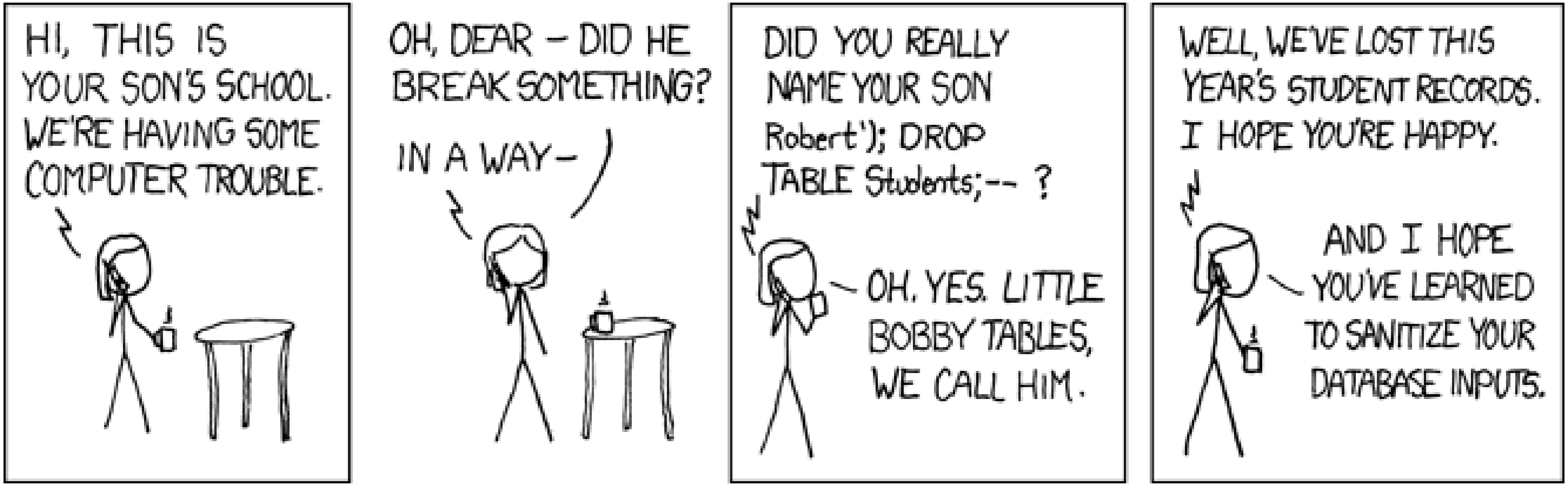
Testing Web Application Scanner Tools

Elizabeth Fong and Romain Gaucher
NIST

Verify Conference

- **Washington, DC, October 30, 2007**

Disclaimer: Any commercial product mentioned is for information only; it does not imply recommendation or endorsement by NIST nor does it imply that the products mentioned are necessarily the best available for the purpose.


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

http://xkcd.com

Outline

- NIST SAMATE Project
- Which tools find what flaws?
- Web Application Scanner tools:
  specification and capabilities
- Testing Web Application Scanner Tools:
  Test methodologies and results

Software Assurance Metrics and Tool Evaluation (SAMATE) Project at NIST

- Project partially funded by DHS and NSA.
- Our focus
    - Examine software development and testing methods and tools to identify deficiencies in finding bugs, flaws, vulnerabilities, etc.
    - Create studies and experiments to measure the effectiveness of tools.

Purpose of Tool Evaluations

- Precisely document what a tool class does and does not do
- Inform users
  - Match the tool to a particular situation
  - Understand significance of tool results
- Provide feedback to tool developers

Details of Tool Evaluations

- Select class of tool
- Develop clear (testable) requirements
  - Tool functional specification aided by focus groups
  - Spec posted for public comment
- Develop a measurement methodology
  - Develop reference datasets (test cases)
  - Document interpretation criteria

Some Tools for specific application*

- Static Analysis Security Tools
- Web Application Vulnerability Tools
- Binary Analysis Tools
- Web Services Tools
- Network Scanner Tools

* Defense Information Systems Agency, "Application Security Assessment Tool Market Survey," Version 3.0 July 29, 2004

Other Types of Software Assurance Security Tools *

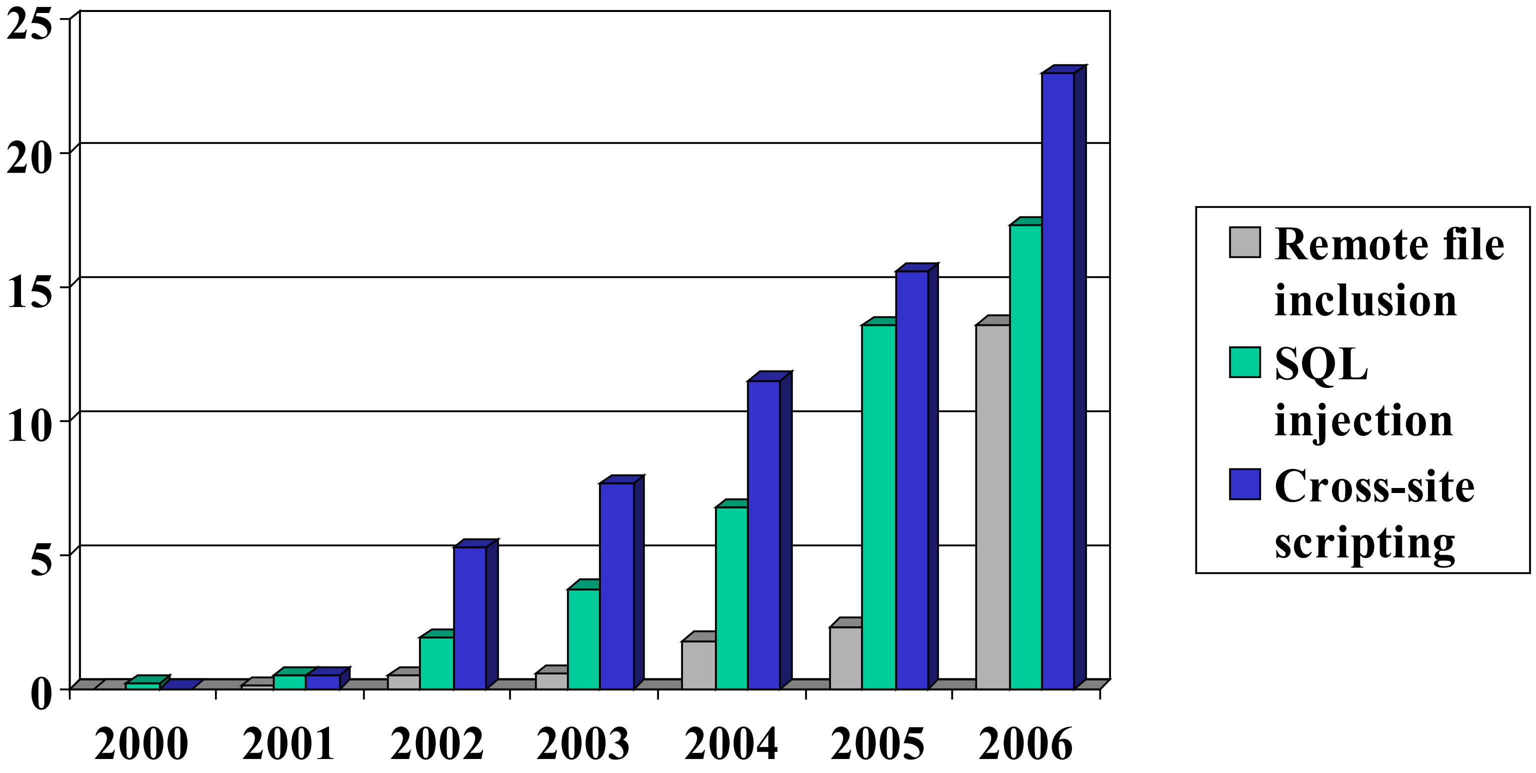
- Firewall
- Intrusion Detection/Prevention System
- Virus Detection
- Fuzzers
- Web Proxy Honeypots
- Blackbox Pen Tester

* OWASP Tools Project

How to Classify Tools and Techniques

- Life Cycle Process (requirements, design, …)
- Automation (manual, semi, automatic)
- Approach (preclude, detect, mitigate, react, appraise)
- Viewpoint (blackbox, whitebox (static, dynamic))
- Other (price, platform, languages, …)

The Rise of Web App Vulnerability



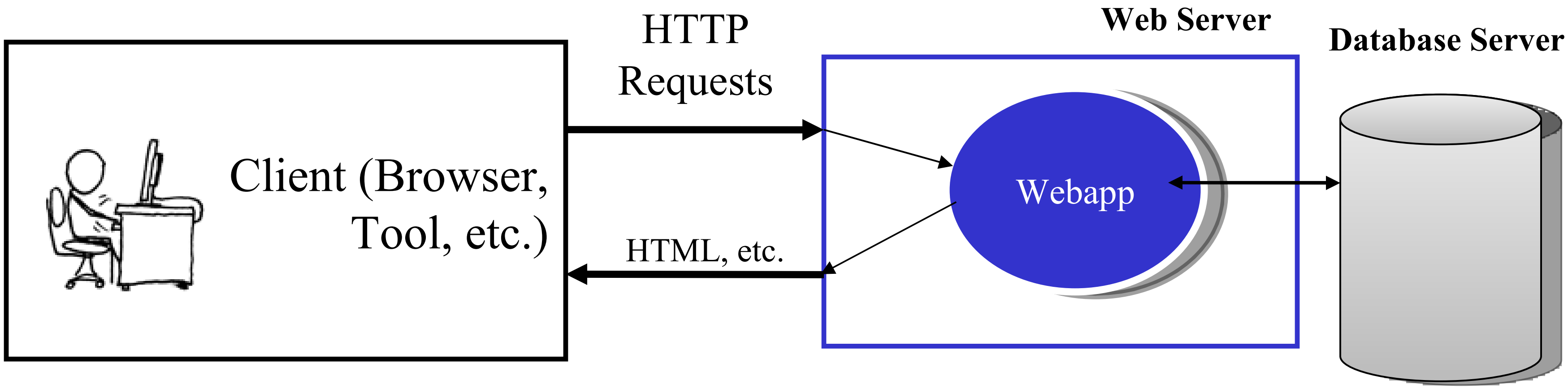
Top web app vulnerabilities as % of total vulnerabilities in NVD

Web Application Security Scanner

*is software which communicates with a web application through the web front-end and identifies potential security weaknesses in the web application.**

* Web Application Security Consortium evaluation criteria technical draft, August 24. 2007.

Web Application Architecture



Characteristics of Web Application

- Client and Server Interaction
- Distributed n-tiered architecture
- Remote access
- Heterogeneity
- Content delivery via HTTP
- Concurrency
- Session management
- Authentication and authorization

Scope – What types of tools does this spec **NOT** address?

- Limited to tools that examine software applications on the web.
- Does not apply to tools that scan other artifacts, like requirements, byte-code, or binary code
- Does not apply to database scanners
- Does not apply to other system security tools, e.g., firewalls, anti-virus, gateways, routers, switches, intrusion detection system

Some Vulnerabilities that Web Application Scanners Check

- Cross-Site Scripting (XSS)
- Injection flaws
- Authentication and access control weaknesses
- Path manipulation
- Improper Error Handling

Some Web Application Security Scanning Tools

- AppScan DE by Watchfire, Inc. (IBM)
- WebInpect by SPI-Dynamics (HP)
- Acunetix WVS by Acunetix
- Hailstorm by Cenzic, Inc.
- W3AF, Grabber, Paros, etc.
- others…

Disclaimer: Any commercial product mentioned is for information only, it does not imply recommendation or endorsement by NIST nor does it imply that the products mentioned are necessarily the best available for the purpose.

Establishing a Framework to Compare

- What is a common set of functions?
- Can they be tested?
- How can one measure the effectiveness?

NIST is "neutral", not consumer reports, and does not endorse products.

Purpose of a Specification

- Precisely document what a tool class does and does not do
- Provide feedback to tool developers
- Inform users
  - Match the tool to a particular situation
  - Understand significance of tool results

How should this spec be viewed?

- Specifies basic (minimum) functionality
- Defines features unambiguously
- Represents a consensus on tool functions and requirements
- Serves as a guide to measure the capability of tools

How should this spec be used?

- Not to prescribe the features and functions that all web application scanner tools must have.
- Use of a tool that complies with this specification does not guarantee the application is free of vulnerabilities.
- Production tools should have capabilities far beyond those indicated.
- Used as the basis for developing test suites to measure how a tool meets these requirements.

Criteria for selection of Web Application Vulnerabilities

- Found in existing applications today
- Recognized by tools today
- Likelihood of exploit or attack is medium to high

Web Application Vulnerabilities

- OWASP Top Ten 2007
- WASC Threat Classification
- CWE – 600+ weaknesses definition dictionary
- CAPEC- 100+ attack patterns for known exploits

Test Suites

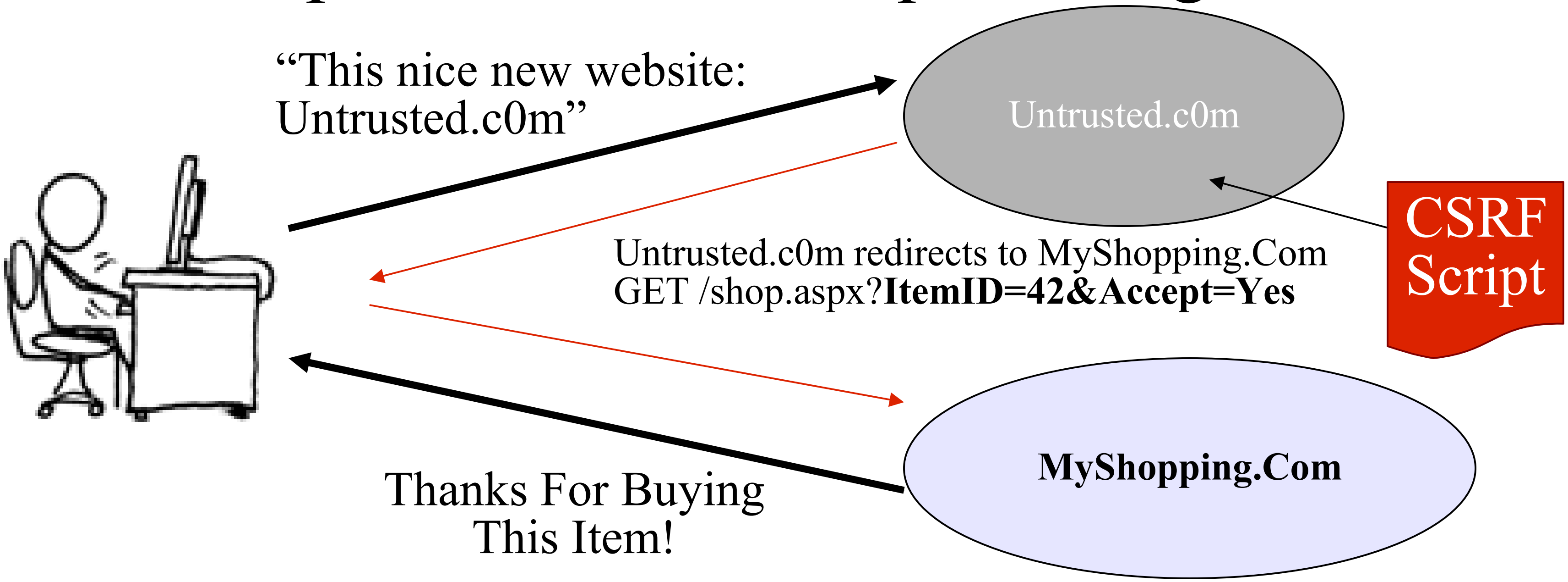
- Test applications that model real security features and vulnerabilities
- Configurable to be vulnerable to one or many types of attack
- Ability to provide increasing level of defense for a vulnerability

Defense Mechanisms

- Different programmers use different defenses
- Defenses/Filters are not all equivalent
- We have different instances of vulnerabilities: levels of defense

Levels of Defense

- Example: Cross-Site Request Forgeries



“This nice new website: Untrusted.c0m”

Untrusted.c0m

CSRF Script

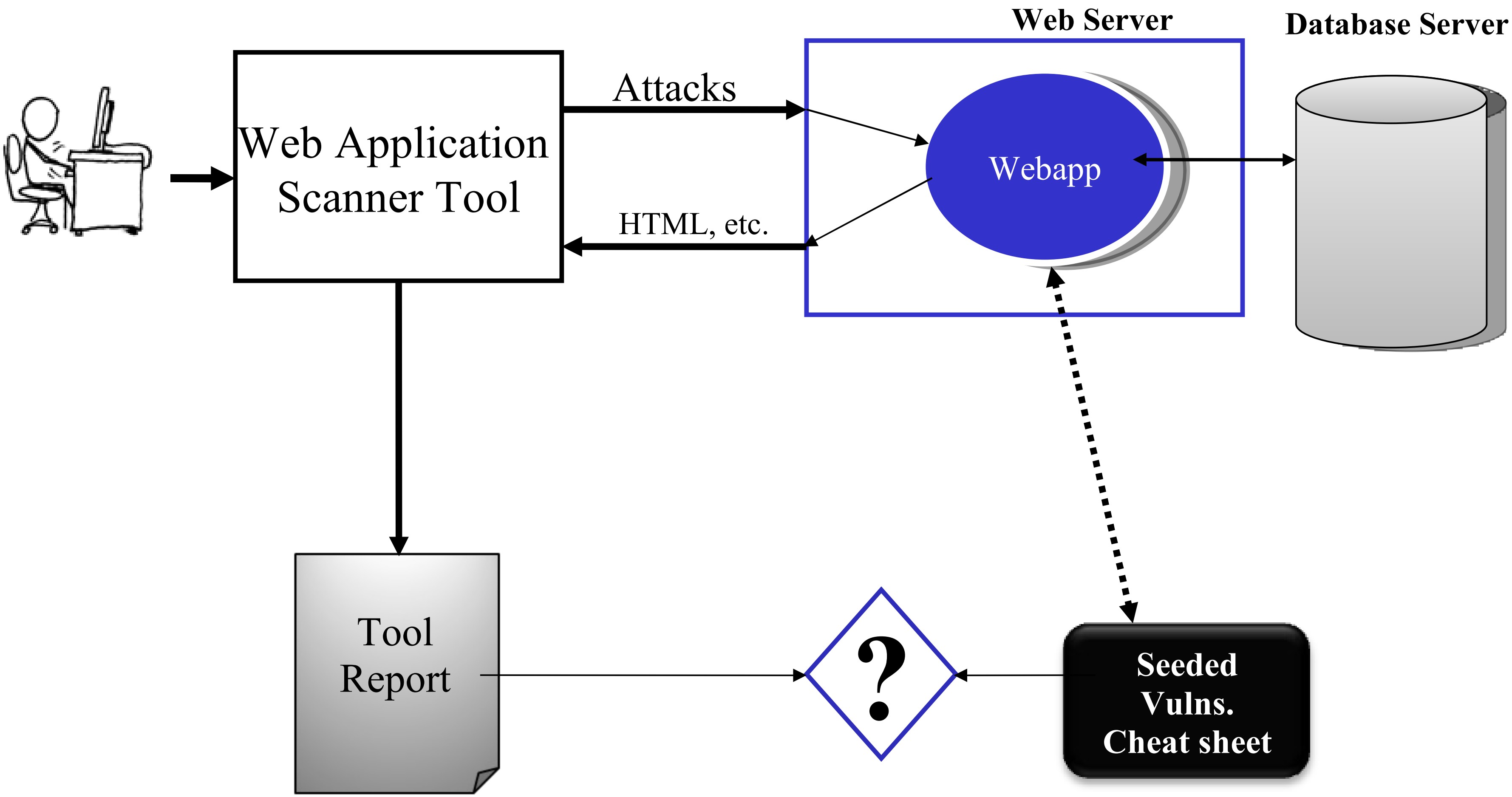
Untrusted.c0m redirects to MyShopping.Com
GET /shop.aspx?**ItemID=42&Accept=Yes**

MyShopping.Com

Thanks For Buying
This Item!

Levels of Defense

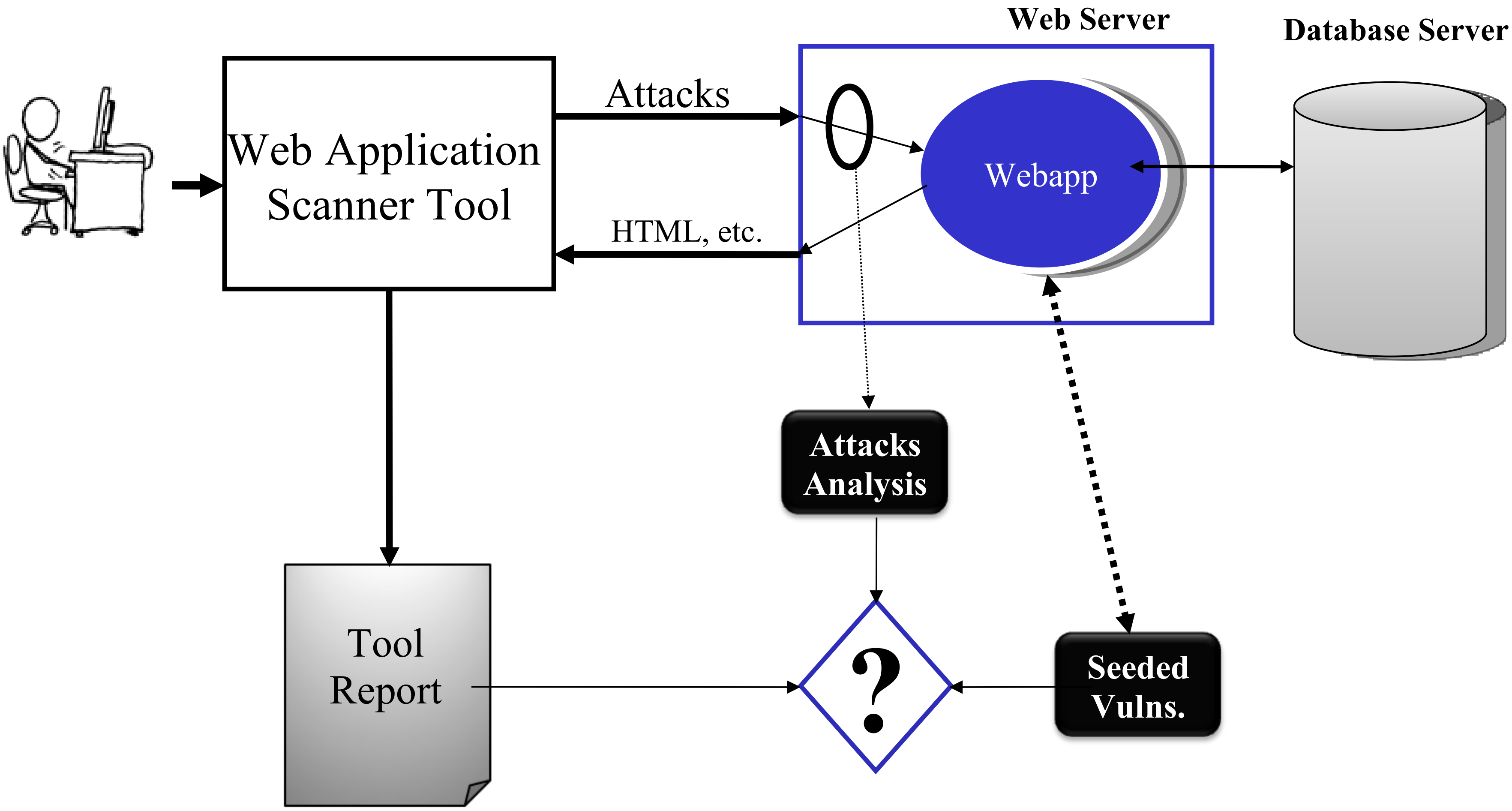
- Example: Cross-Site Request Forgeries
  - Level 0: No Protection (bad)
  - Level 1: Using only POST (well...)
  - Level 2: Checking the referrer (better but referrer may be spoofed)
  - Level 3: Using a nonce (good)
- Higher level means harder to break


Web Server
Database Server
Web Application Scanner Tool
Attacks
HTML, etc.
Webapp
Tool Report
?
Seeded Vulns. Cheat sheet

Attacks Analysis

- An action that exploits a vulnerability
- What exactly is the tool testing?
- What do I need to test in my application?

- Do the results match?

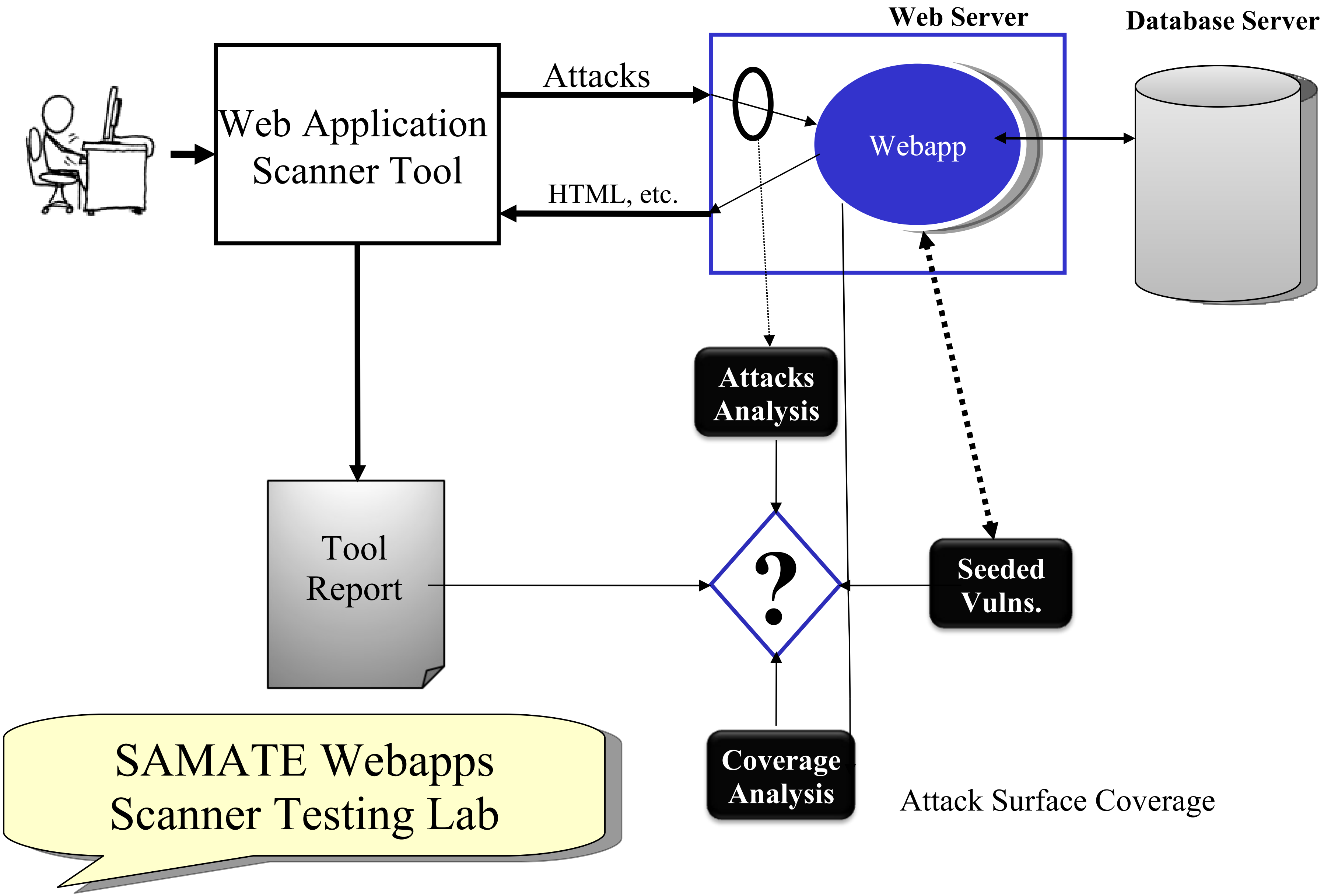

Web Server
Database Server
Web Application Scanner Tool
Attacks
HTML, etc.
Webapp
Attacks Analysis
Tool Report
?
Seeded Vulns.

Attack Surface Coverage

- **Testing the tool accuracy** by inserting check points in most of the attack surface
- Is the tool testing all the application surface?

  Ex: login correctly, with errors, etc.

```
(1) Touch the file [login.php]
if ( all fields are set ) then
    (2) All fields are set [login.php]
    Boolean goodCredentials = checkThisUser(fields)
    if ( goodCredentials ) then
        (3) Credentials are correct; Log in [login.php]
        registerSessionCurrentUser()
    else
        if ( available login test > 0 ) then
            (4) Login information incorrect [login.php]
            displayErrorLogin()
            available login test -= 1
        else
            (5) Too many tries with bad info [login.php]
            displayErrorLogin()
            askUserToSolveCAPTCHA()
        endif
    endif
endif
```



Web Server

Database Server

Web Application Scanner Tool

Attacks

HTML, etc.

Webapp

Attacks Analysis

Tool Report

?

Seeded Vulns.

Coverage Analysis

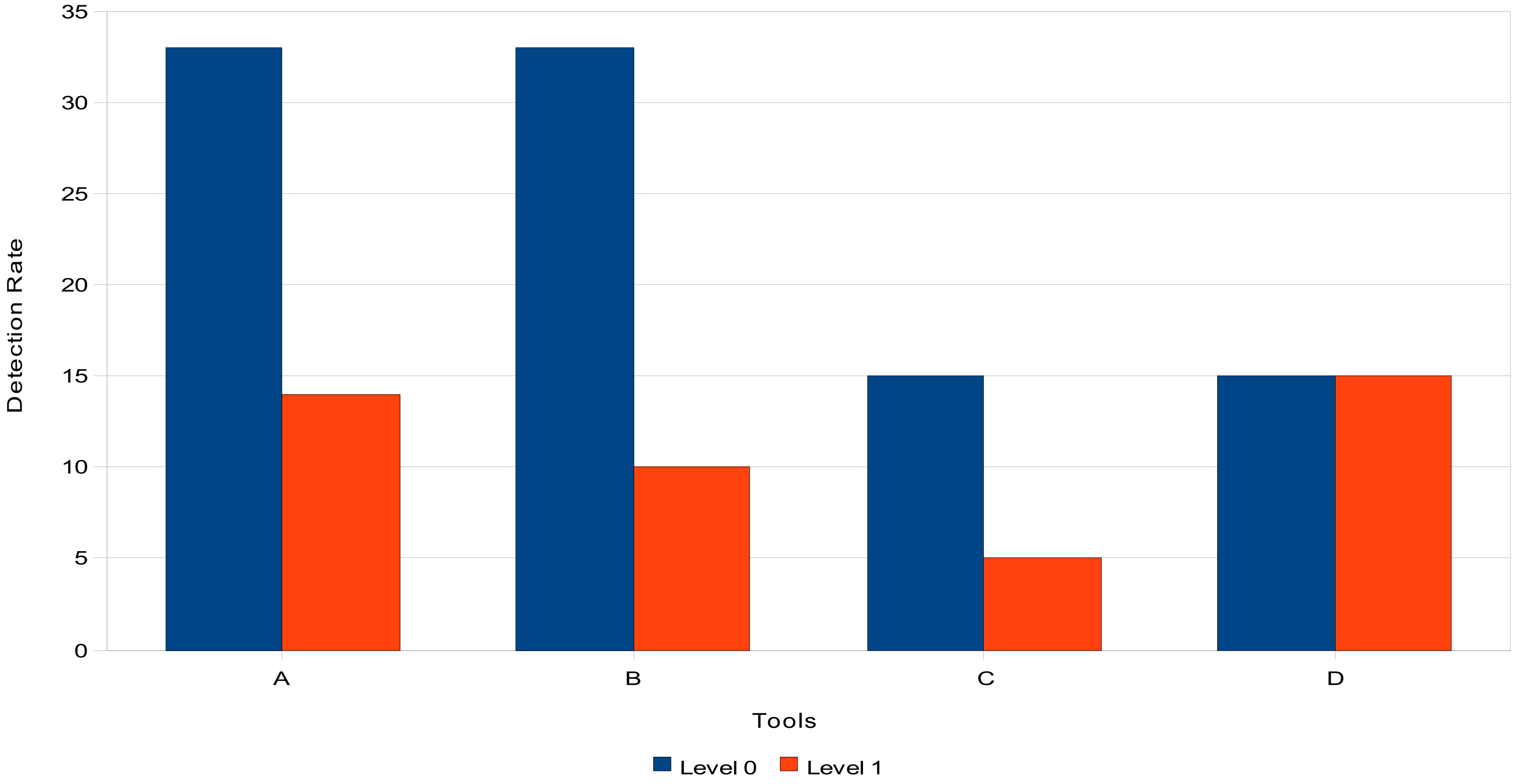
Attack Surface Coverage

SAMATE Webapps Scanner Testing Lab

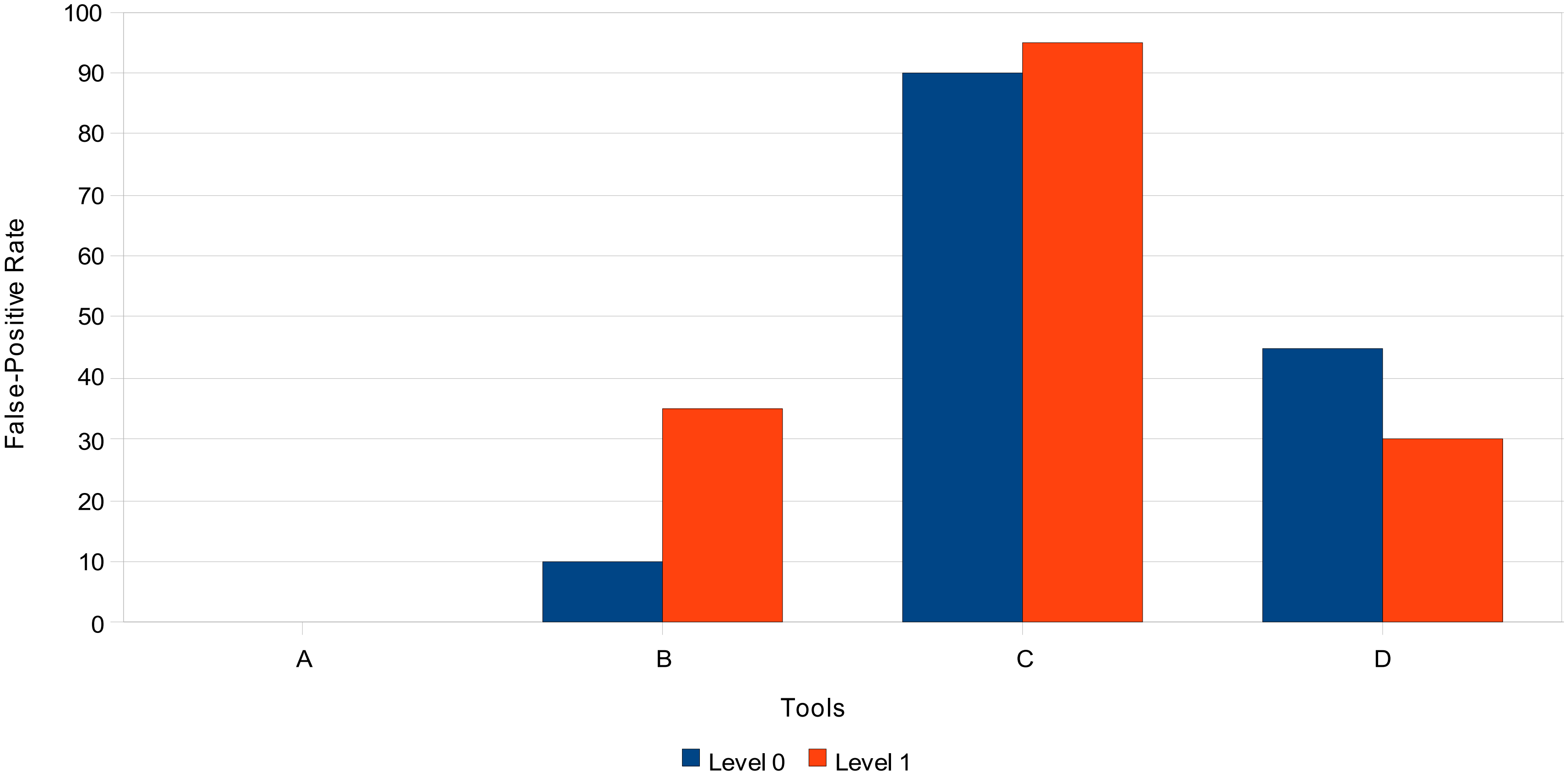
Test Suite Evaluation

- Test Suite with 21 vulnerabilities (XSS, SQL Injection, File Inclusion)
  - PHP, MySQL, Ajax
  - LAMP
- 4 Scanners (Commercial and Open Source)
- One type of vulnerability at the time
- Results (Detection rate, False-Positive rate)

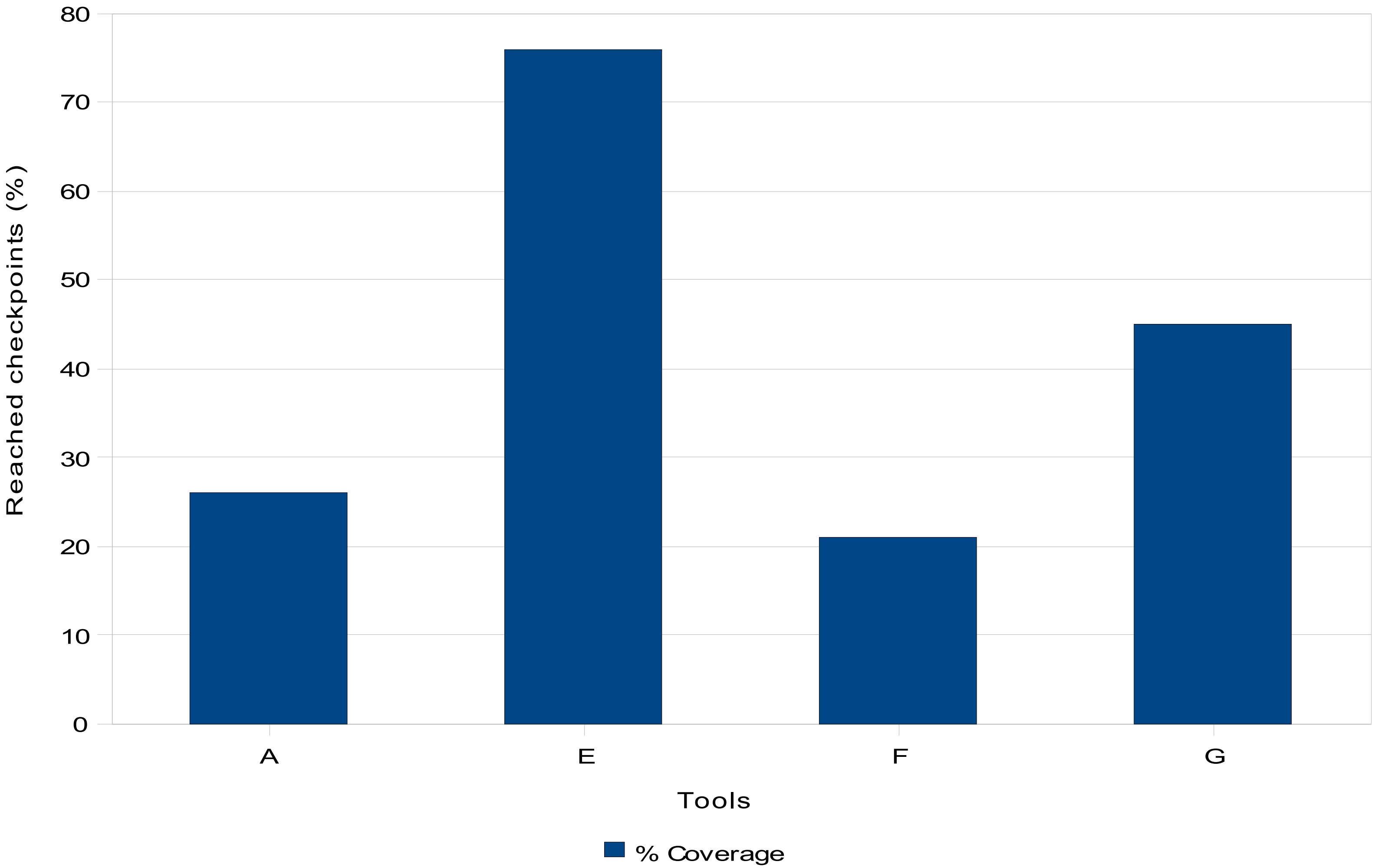
Detection Rates for Different Levels of Defense



False Positive Rates for Different Levels of Defense

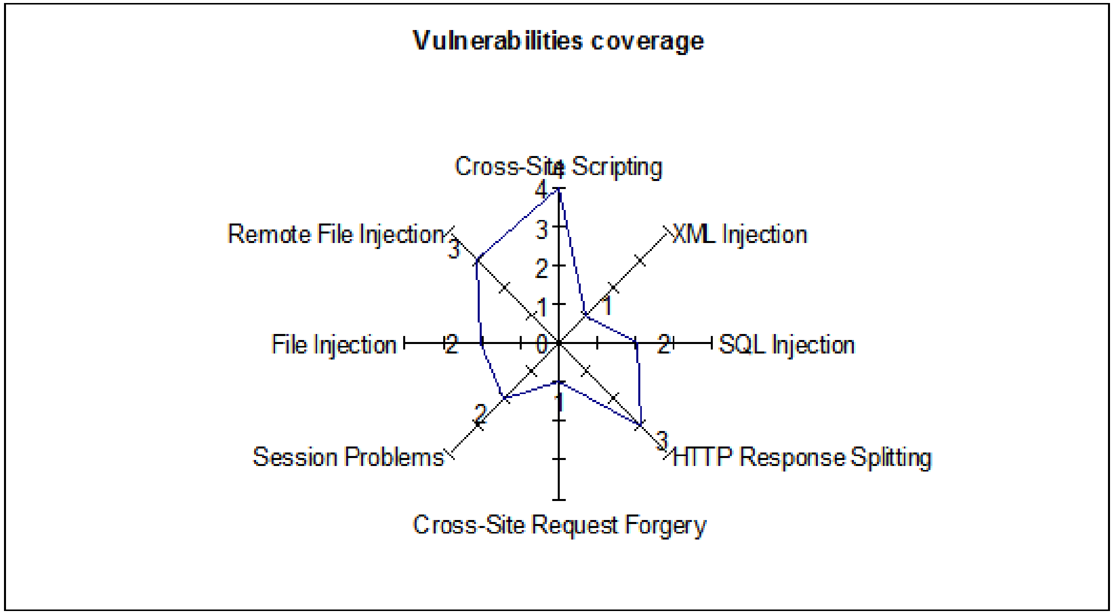


Attack Surface Coverage



Coming next

- Refining level of defense in order to have a better granularity
- Thinking of tool profiles such as:



Coming next (cont.)

- Using different technologies in our test suites (JSP, .NET, etc.)
- More than one vulnerability at a time (combinatorial testing?)
- Metrics? Brian Chess' metric?

t: True Positive
p: False Positive
n: False Negative

$$\frac{100 \cdot t}{t + p + n}$$

Issues with Web Application Scanner Tools

- Tools are limited in scope (companies sell service as opposed to selling tool)
- Speed versus Depth (in-depth testing takes time)
- Difficult to read output reports (typically log files)
- False-Positives
- Tuning versus default mode

We need …

- People to comment on specifications

- People to submit test cases for sharing with the community

- People to help build reference datasets for testing tools?

Contacts

- SAMATE web site **http://samate.nist.gov/**
- Project Leader: Dr. Paul E. Black
- Project Team Members:

  Elizabeth Fong, Romain Gaucher,

  Michael Kass, Michael Koo,

  Vadim Okun, Will Guthrie, John Barkley