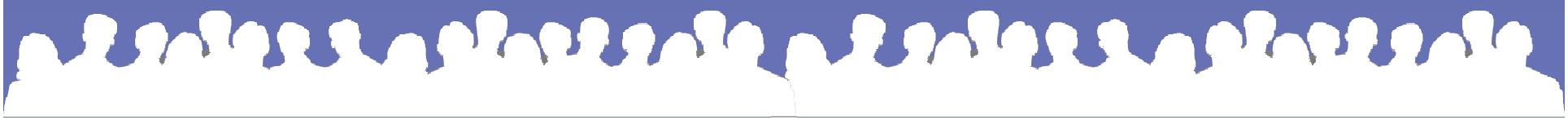
SECURITY 2012

20. ročník konference o bezpečnosti v ICT

Bezpečnost při vývoji aplikací

Jan Svoboda

Rational Presales, IBM

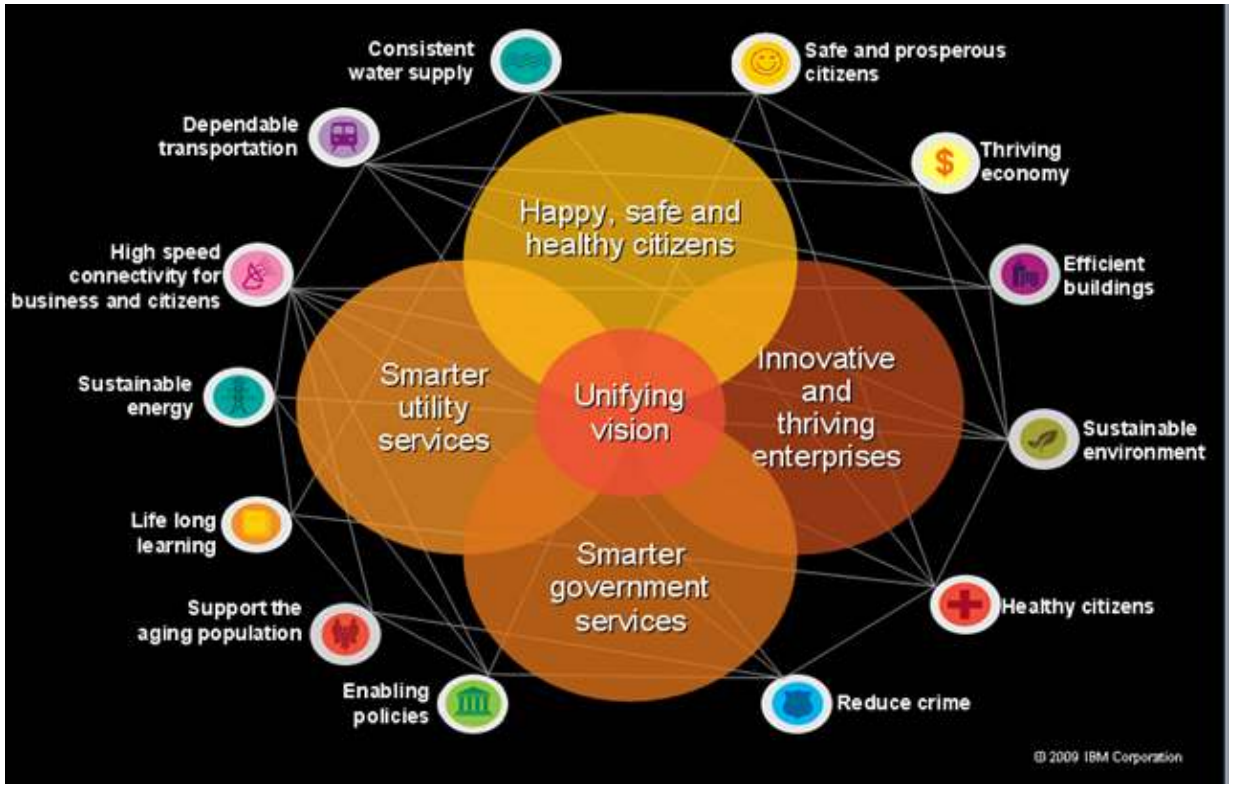


The Smarter Planet

Our world is getting **Instrumented**

Our world is getting **Interconnected**

Our world is getting **Intelligent**



Grow on the Smarter Planet

Key drivers for software security projects

Increasing Complexity

Soon, there will be **1 trillion** connected devices in the world, constituting an "***internet of things***"†

Increasing Exploits and Accidents

900+ Breaches reported
900+M records exposed‡

Increasing Impact

The cost of a US data breach increased to **$204** per compromised customer record and **$6.8M** Million per breach⌜

Sources † ***http://searchcompliance.techtarget.com/news/article/0,289142,sid195_gci1375707,00.html***
‡ *2010 Verizon Business / US Secret Service Data Breach Investigations Report*
⌜ *2010 Ponemon Institute Data*

Complex Picture of Auto / IT System Interconnect

Outsourced / OEM Providers

Internal Development

Remote Support

Mobile Applications

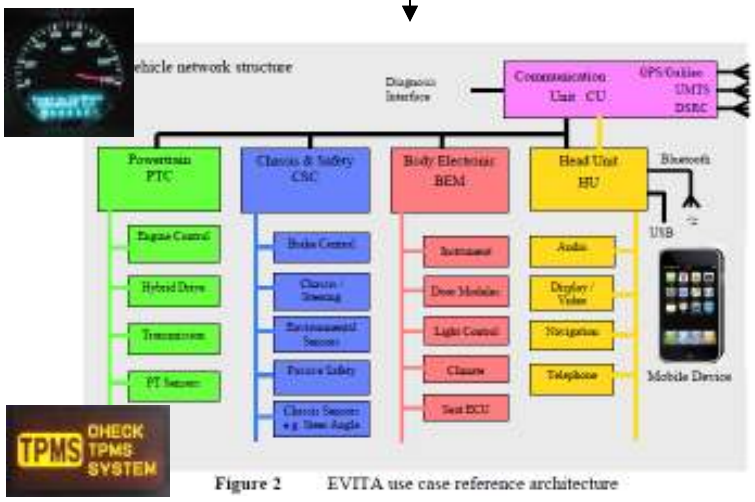


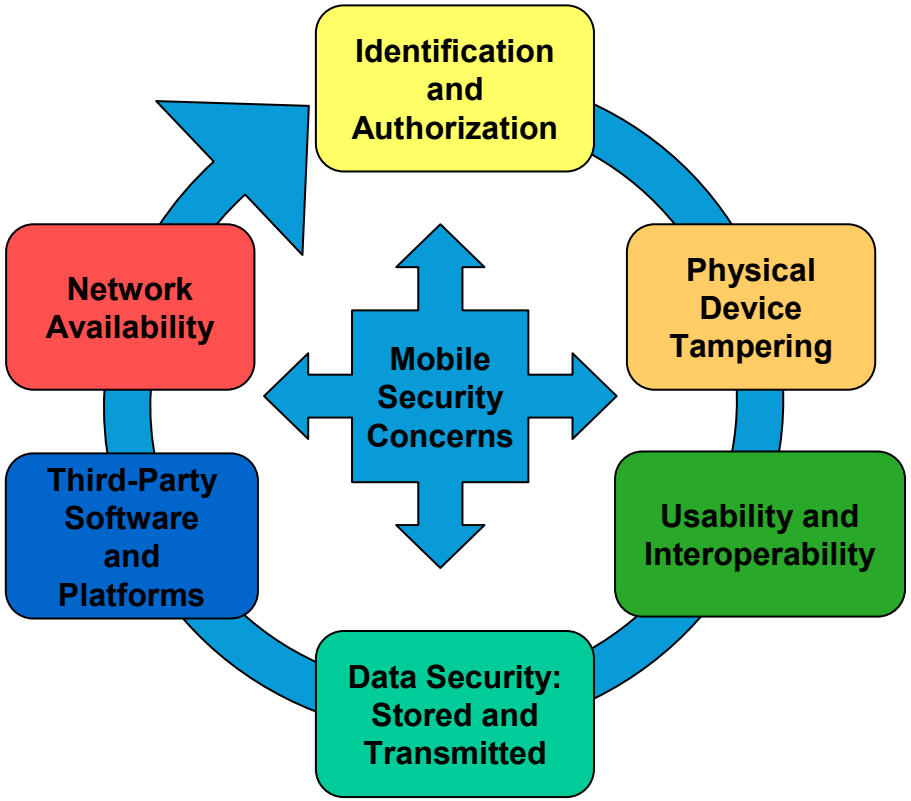
Figure 2 EVITA use case reference architecture

http://www.evita-project.org/Deliverables/EVITAD2.3.pdf

Internal Auto Systems

External Connections

Mobile Platform Popularity Creating New Threat Vectors



"Fake Mobile Banking App Discovered in Android Marketplace"

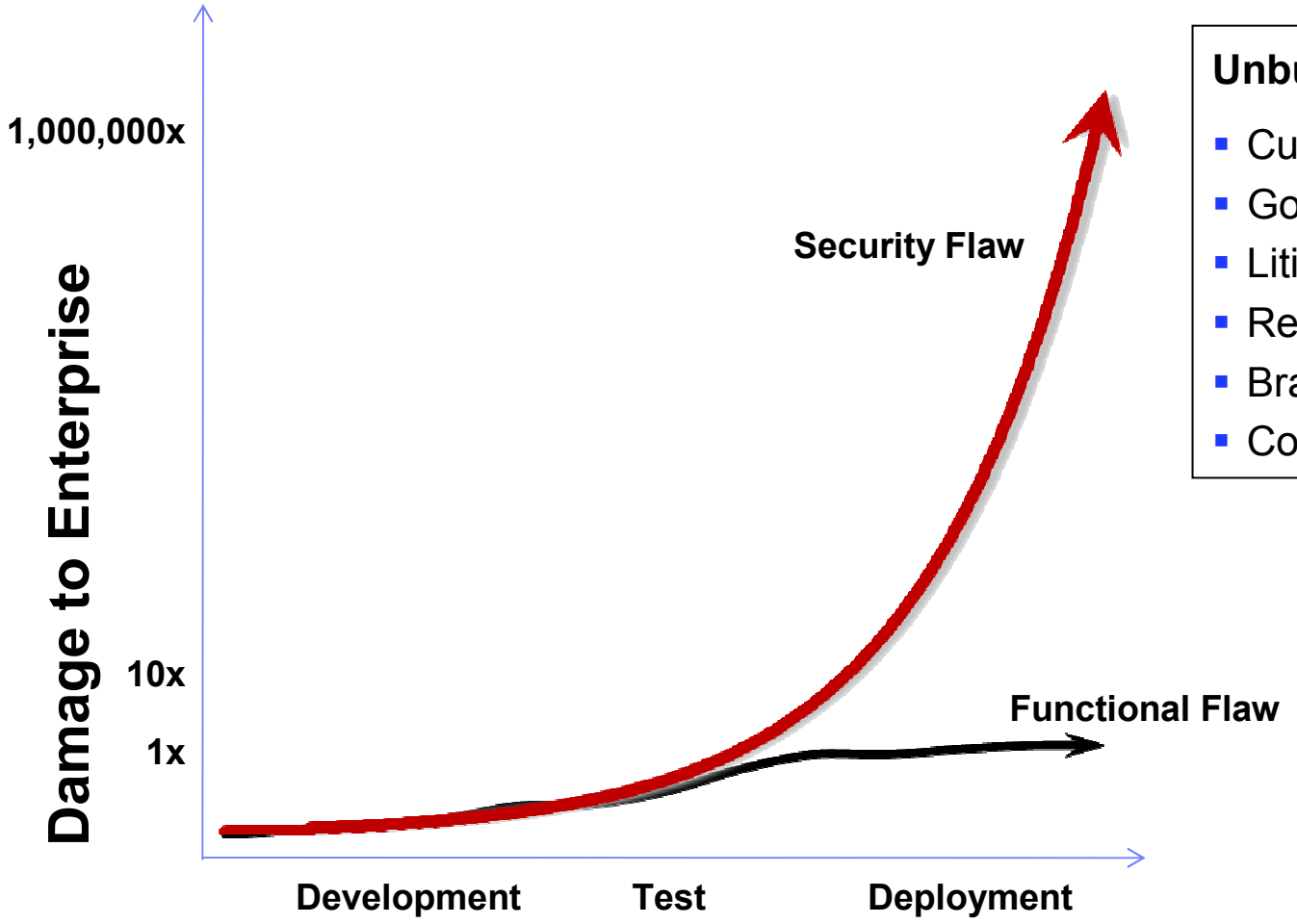
Humberto Saabedra – 01/10/2010

"iPhone worm hijacks ING customers"

John Leyden – 11/23/2009

"Rootkit-based Exploits Could Eavesdrop Smartphones" - 01/25/2010

Sources of Security Breach Costs

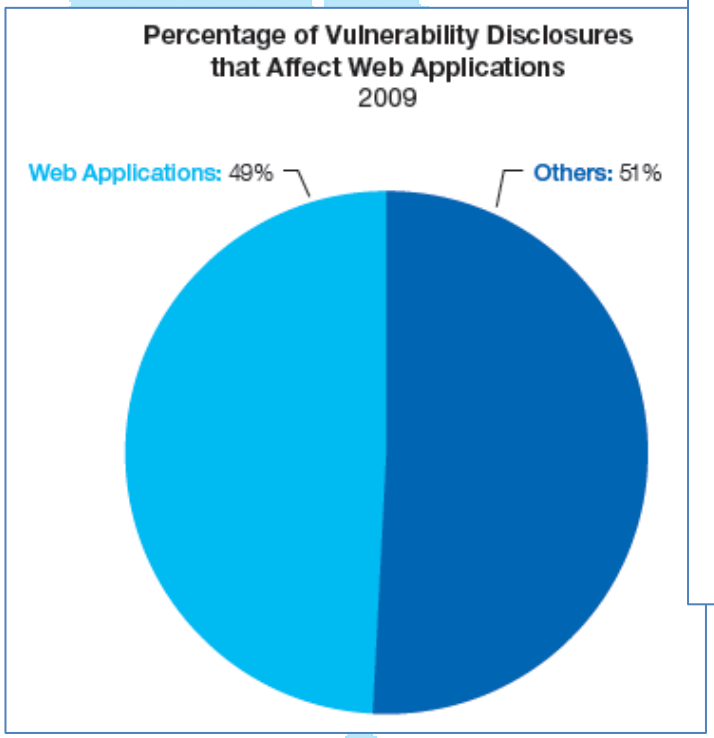


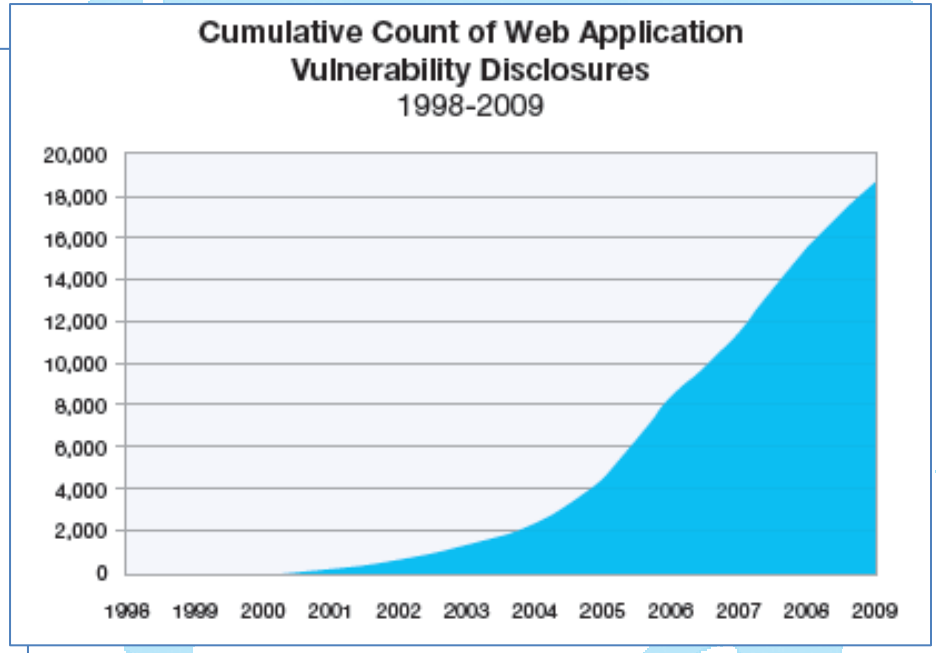
Unbudgeted Costs:

- Customer notification / care
- Government fines
- Litigation
- Reputational damage
- Brand erosion
- Cost to repair

Web Applications are the greatest risk to organizations

- Web application vulnerabilities represented **the largest category** in vulnerability disclosures
- In 2009, **49% of all vulnerabilities** were Web application vulnerabilities
- SQL injection and Cross-Site Scripting are neck and neck in a race for the top spot





IBM Internet Security Systems 2009 X-Force® Year End Trend & Risk Report

Why are Web Applications so Vulnerable?

- Developers are mandated to deliver functionality on-time and on-budget - but not to develop secure applications
- Developers are not educated in secure code practices
- Product innovation is driving development of increasingly complicated software

Volumes of applications continue to be deployed that are riddled with security flaws…

…and are non compliant with industry regulations

Accelerating Awareness and Progress: Secure by Design

Secure by Design is a **cost-effective** approach to constructing **safe and reliable systems** by applying IBM's experience with security technologies and best practices in all phases of system creation, from conception through system design, construction and deployment.

Being **Secure by Design** reduces the **cost**, **risk**, and **unpredictability** of integrating new technologies.

Make Applications Secure, by Design

Cycle of secure application development

Design Phase

- Consideration is given to security requirements of the application
- Issues such as required controls and best practices are documented on par with functional requirements

Development Phase

- Software is checked during coding for:
  - Implementation error vulnerabilities
  - Compliance with security requirements

Build & Test Phase

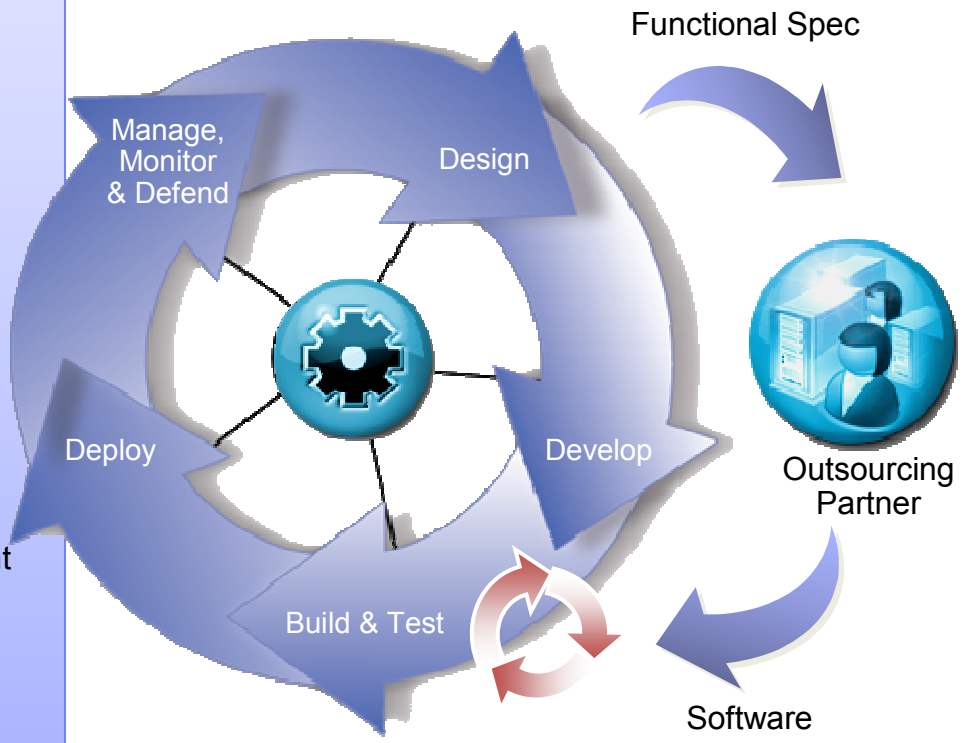
- Testing begins for errors and compliance with security requirements across the entire application
- Applications are also tested for exploitability in deployment scenario

Deployment Phase

- Configure infrastructure for application policies
- Deploy applications into production

Operational Phase

- Continuously monitor applications for appropriate application usage, vulnerabilities and defend against attacks



Cost is a Significant Driver

80% of development costs are spent identifying and correcting defects!*

During the coding phase $80/defect

During the build phase $240/defect

During the QA/Testing phase $960/defect

Once released as a product $7,600/defect + Law suits, loss of customer trust, damage to brand

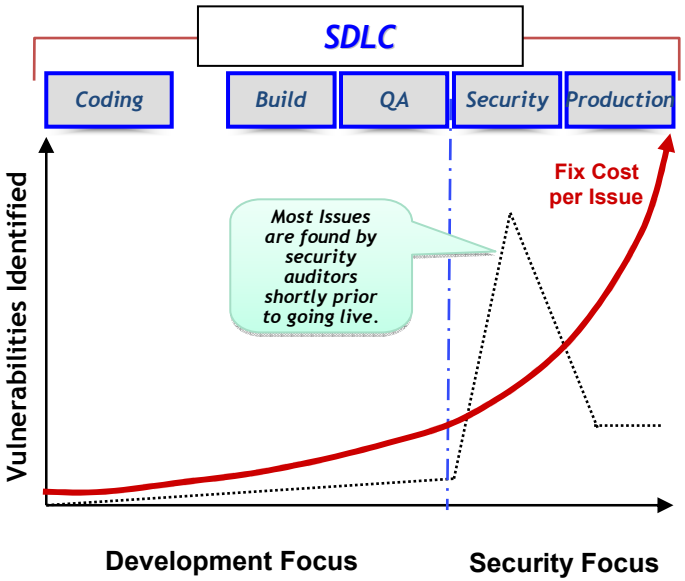
The increasing costs of fixing a defect….

*National Institute of Standards & Technology

Source: GBS Industry standard study

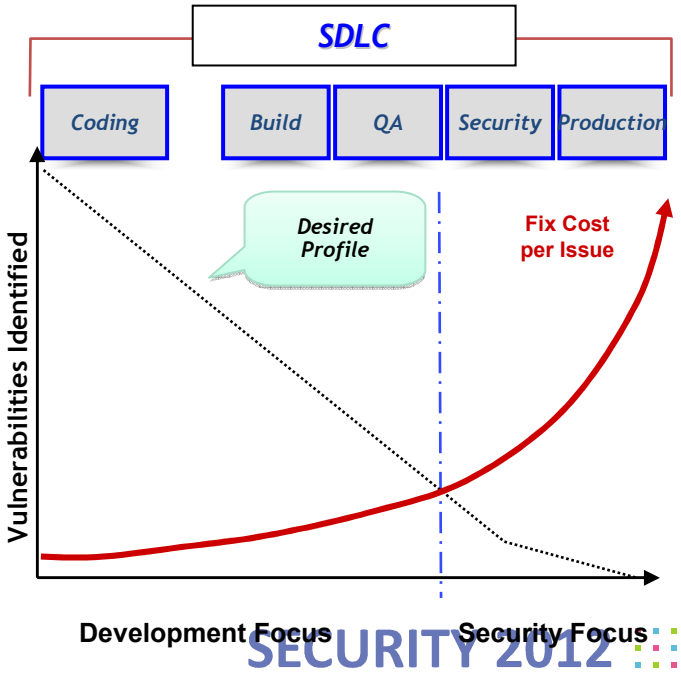
Defect cost derived in assuming it takes 8 hrs to find, fix and repair a defect when found in code and unit test. Defect FFR cost for other phases calculated by using the multiplier on a blended rate of $80/hr.

Moving to a Desirable End State

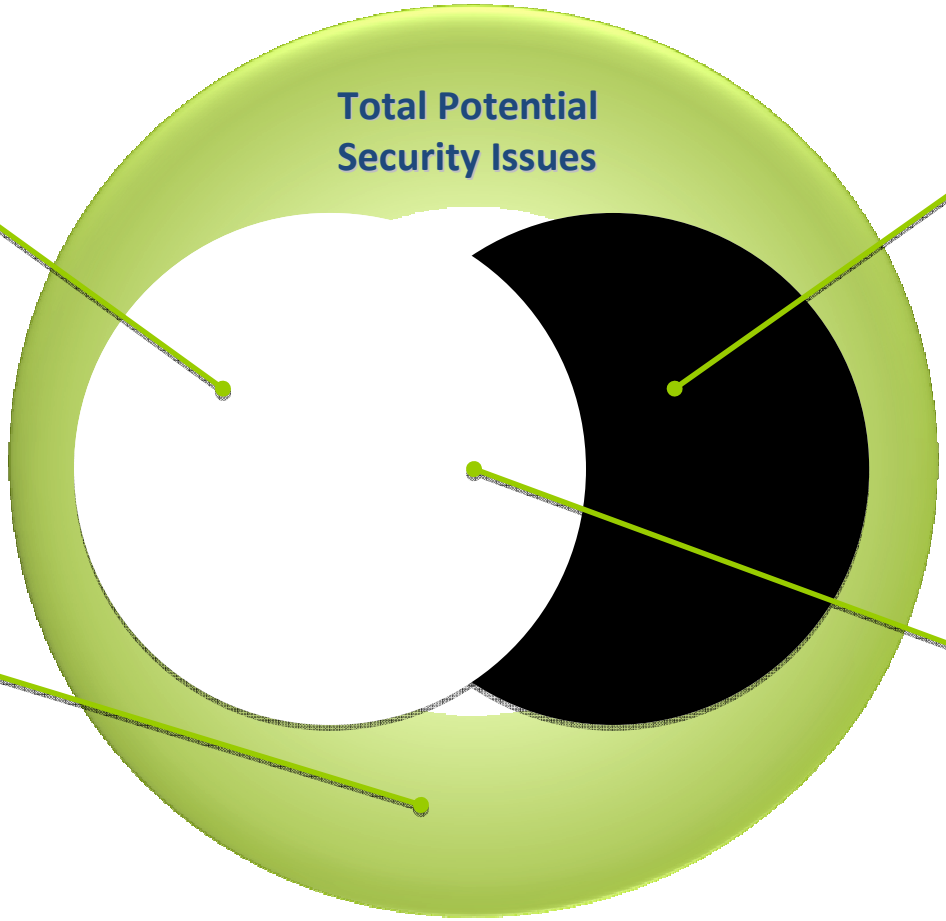


In early stages of adoption, security practitioners will assess applications during pre-deployment testing. **Costs are higher and window is shorter** to mitigate any issues found.

By integrating security into requirements, development and build/test/integration cycles, **identification occurs much earlier**, increasing find rate at a time when **fix costs are lowest**.



Dynamic vs. Static Analysis



Static Analysis

- Null pointer dereference
- Threading issues
- Code quality issues
- Issues in dead code
- Insecure crypto functions
- Issues in back-end application code
- Stored vulneabilities
- Complex SQL injection

Dynamic Analysis

- Environment configuration issues
- Patch level issues
- Runtime privileges issues
- Authentication issues
- Protocol parser/serializer issues
- Session management issues
- Issues in external 3rd party web components
- Cross-site request forgery
- Malware analysis

Manual Assessment

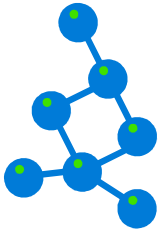
- Business logic issues

Dynamic & Static

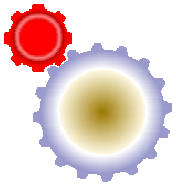
- SQL Injection
- Cross Site Scripting
- HTTP Response Splitting
- OS Commanding
- LDAP Injection
- XPath Injection
- Path Traversal
- Buffer Overflows
- Format String Issues

Static Analysis

Source Code



Model

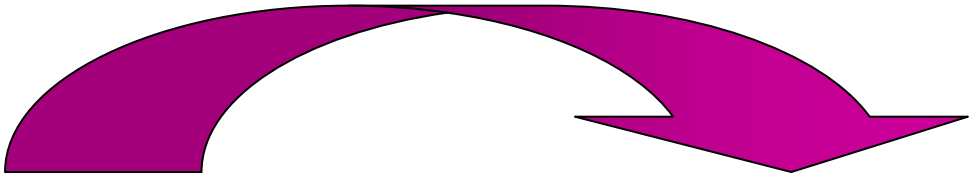


Perform Analysis

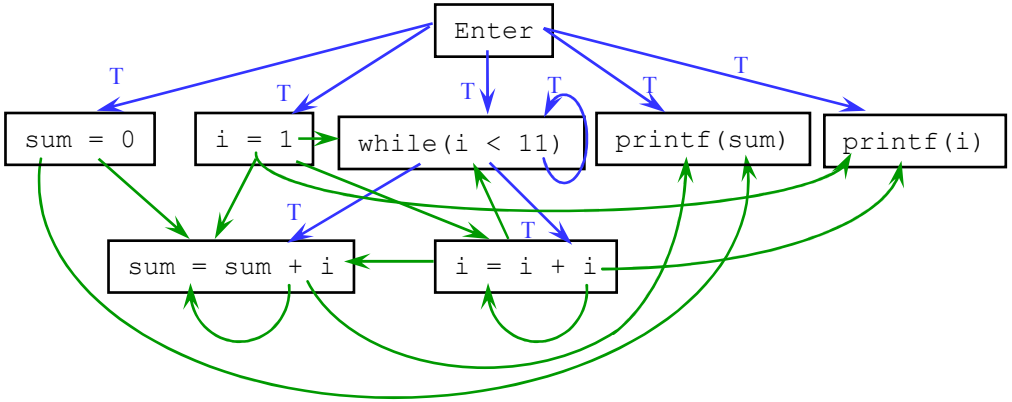
Results

Domain Knowledge (e.g. Security Rules)

Modeling

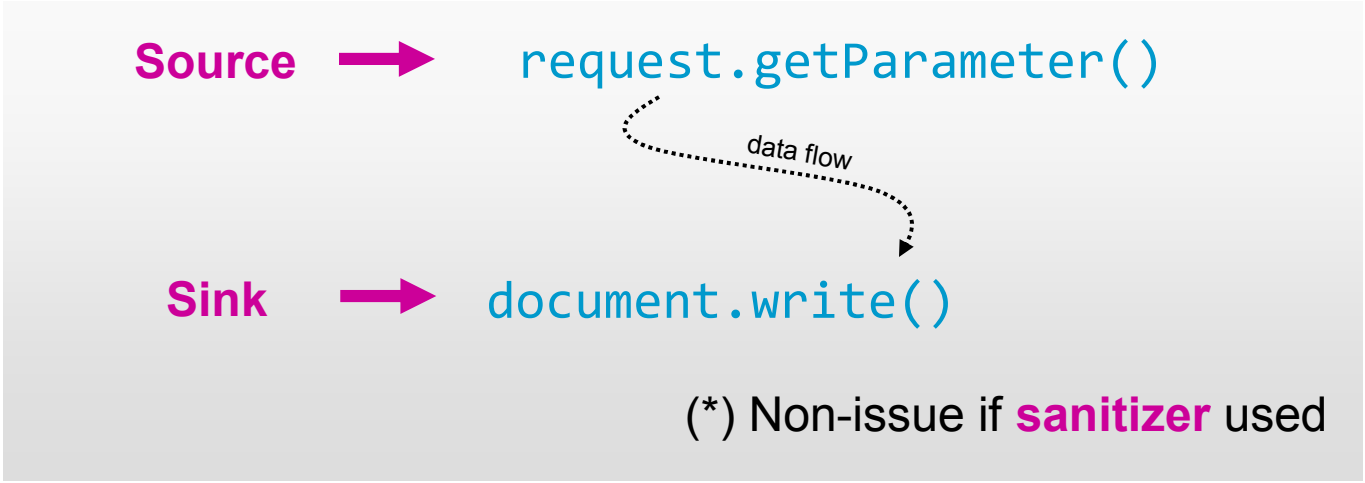


```
int main() {
    int sum = 0;
    int i = 1;
    while (i < 11) {
        sum = sum + i;
        i = i + 1;
    }
    printf("%d\n",sum);
    printf("%d\n",i);
}
```

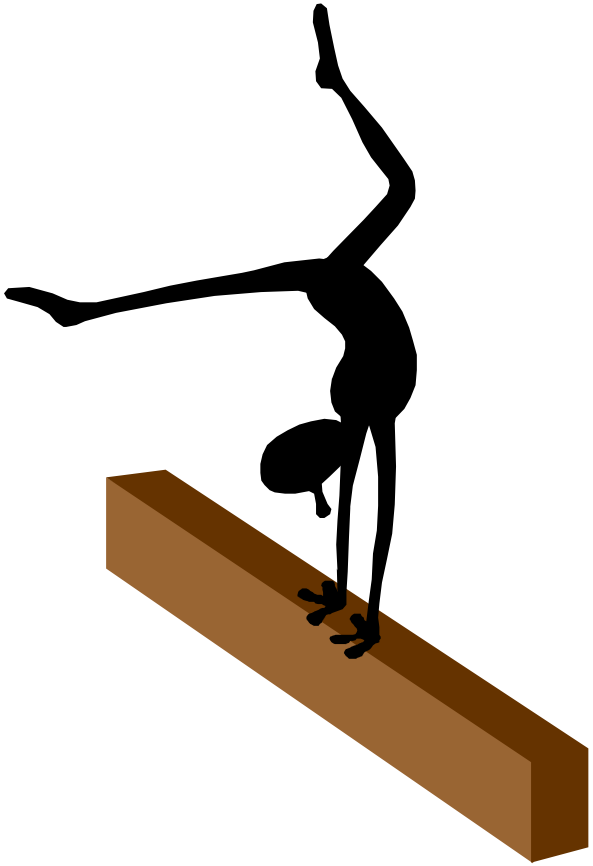


Taint Analysis

- Information-flow violation problems can be solved using **static taint analysis**



Challenges in Static Analysis (1)

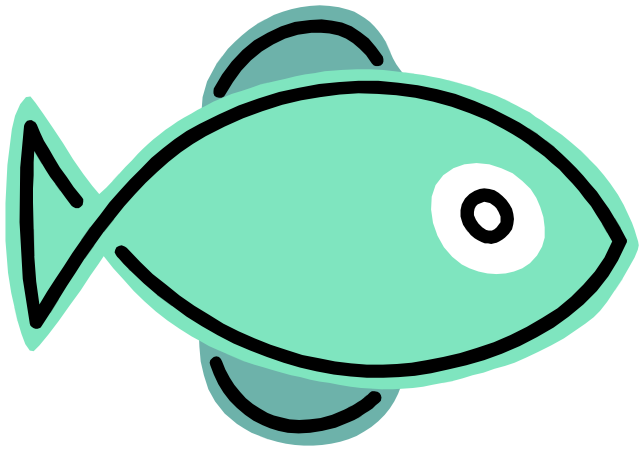


The Balancing Act

Tradeoffs

- Large models or small?
- Faster analysis or more accurate?
- Bias towards false positives or false negatives?

Challenges in Static Analysis (2)



The Babel Fish

Abstraction

Speaking the right language,

picking the right abstraction.

- Taint analysis is a binary analysis: either tainted or not
- But accurate security assessment requires understanding of **string** content and context


DEMO

SECURITY 2012

20. ročník konference o bezpečnosti v ICT

Děkujeme za pozornost.

Jan Svoboda
Rational Presale, IBM
jan_svoboda@cz.ibm.com