

Building Secure Applications



Joseph Feiman

Gartner[®]

It Is Not What We Think

- Firewalls, IDSs, VPNs, SSL and authentication are not sufficient for application protection
- Security is not a synonym of quality
- Security testing is not to assure that the application is doing what it is supposed to do
- Vulnerabilities get introduced at analysis, design and construction phases. Yet, today's security efforts start and end at the operation phase
- Reusable components could be a source of reusable vulnerabilities
- You cannot make chaos secure
- You cannot secure a system that you do not understand. The key to legacy security is legacy understanding

Key Issues

1. How should the application development process change to make applications more secure?
2. Which vendors, tools and concepts enable better security?

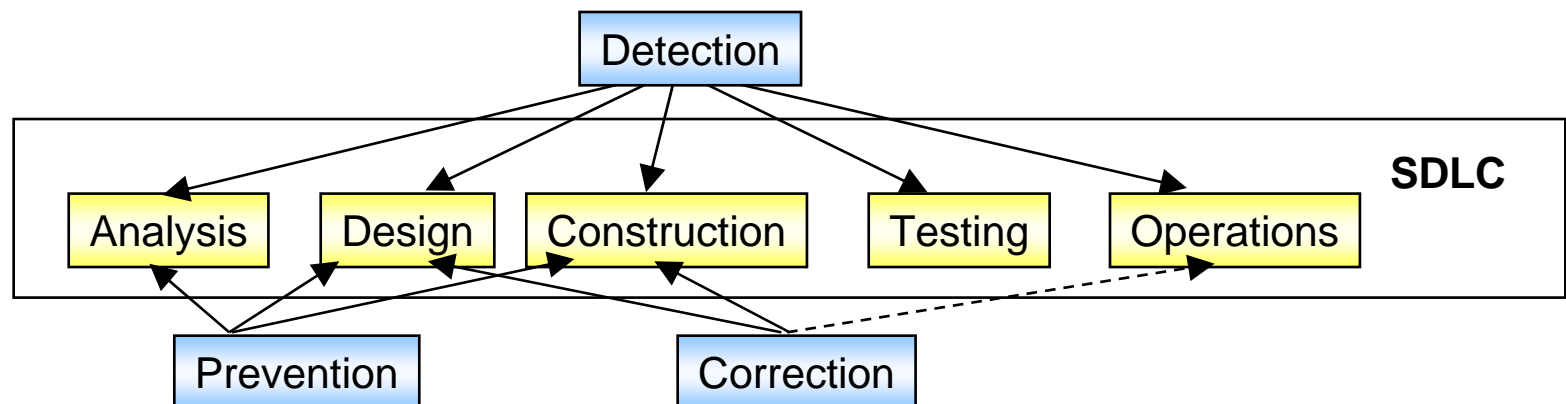
Firewalls, IDSs, VPNs, SSL and Authentication Are Not (Sufficient) Application Protectors

- Ports 80 and 443 always open for e-business (that is, path to applications is opened)
- SSL and data encryption protect data transmission, not applications
- Firewalls control traffic, not applications
- VPNs control traffic, not applications
- Authentication verifies only users of applications
- None of the above protects against internal threats

Therefore: Applications need protection through separate, specific security measures

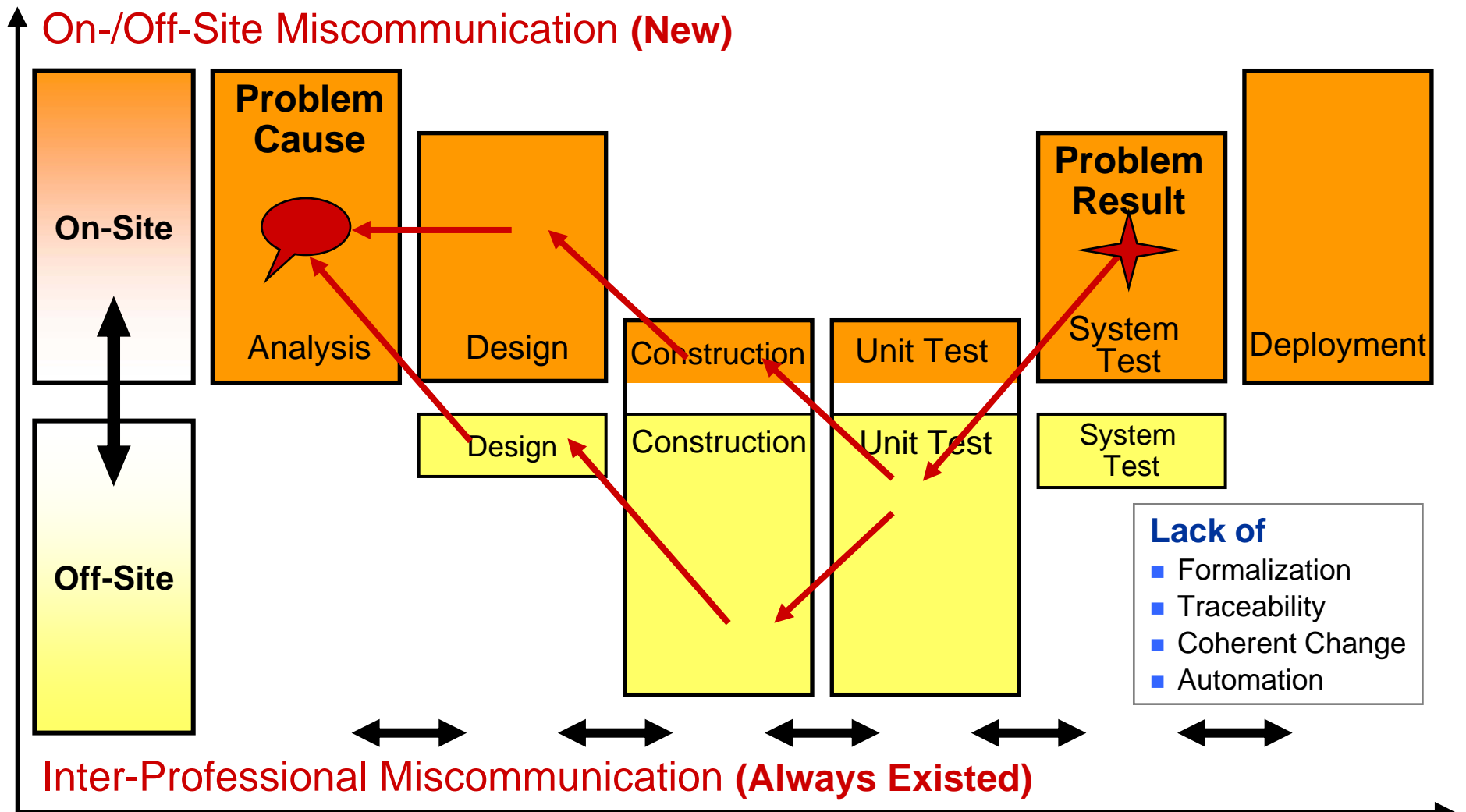
Defining Application Security

- **Security** is a set of measures for protecting an application against unforeseen actions (intentional or unintentional) that cause its cease of functioning or exploitation.
- **Security** is a set of measures for protecting application **quality** under attack.
- **Security** should provide assurance through detection, prevention and correction.



- **Security** should be an integral part of the SDLC.

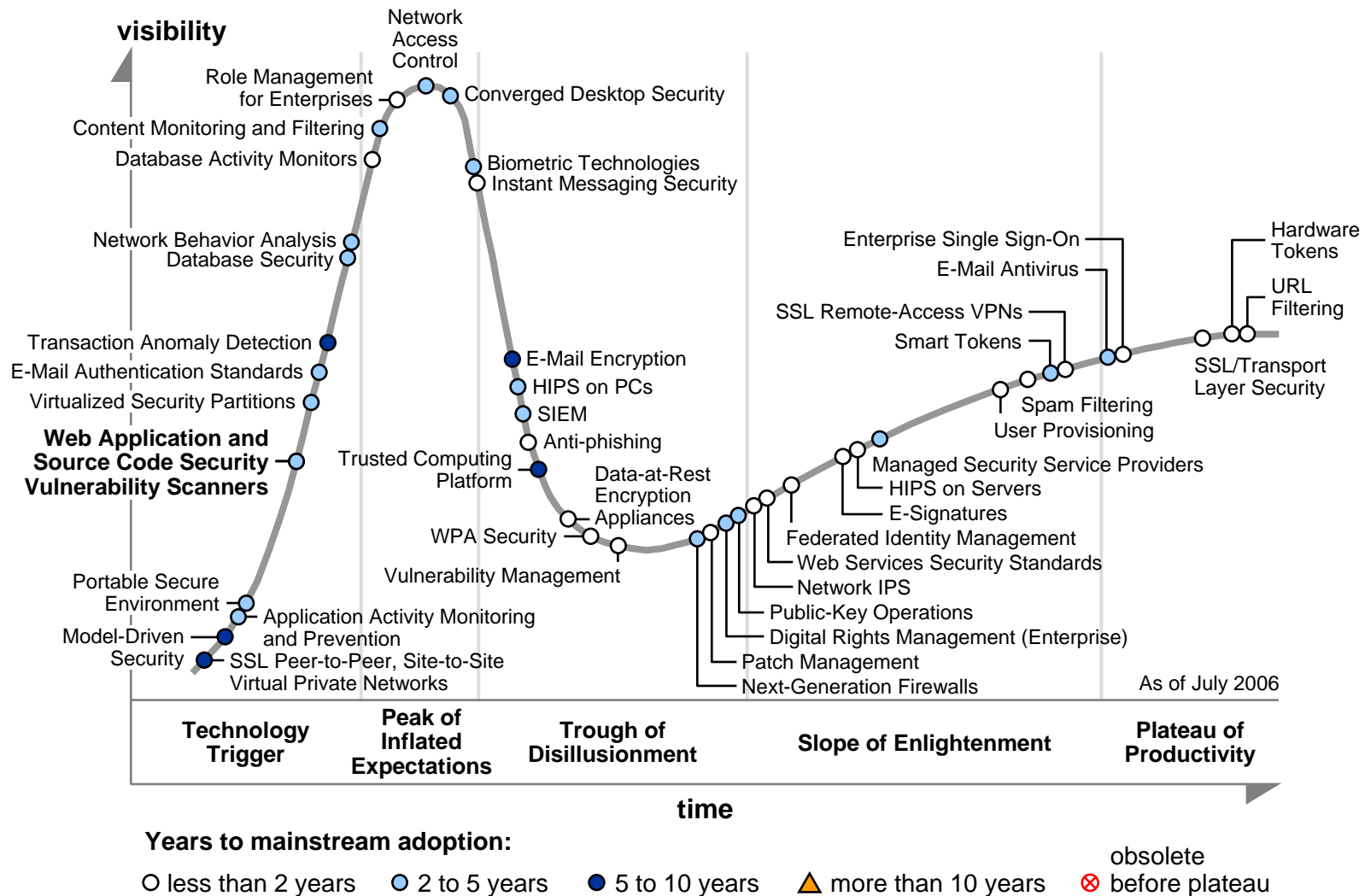
You Cannot Make Chaos Secure



Security Checks Along SDLC

Vulnerability Checks	SDLC Phases	Maturity of Tools, Practices	Injected Vulnerabilities (Not Necessarily Security)
Vulnerabilities in requirements, business processes flow, algorithms	Analysis	Embryonic	15%
Vulnerabilities caused by interrelations of modules and (Web) services, logic and data flow	Design	Embryonic	40%
Vulnerabilities in language instructions, implementation of logic and data flow	Construct	Low	35%
Vulnerabilities in executables, UI. Assembly of secure services could be insecure	Testing	Low	10%
Missing patches, administrative errors, misconfiguration. If vulnerability found — back to analysis	Operations	Low-Medium	

Hype Cycle for Information Security, 2006

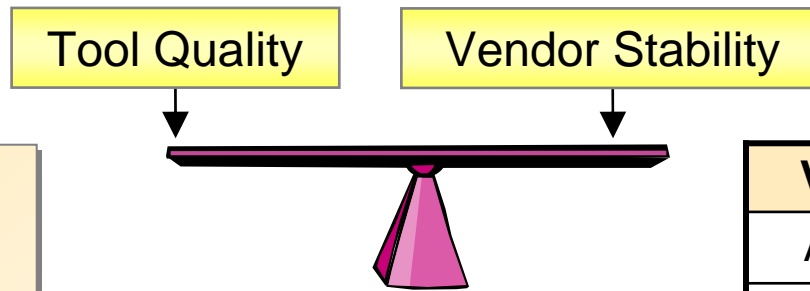


(From "Hype Cycle for Information Security, 2006," 10 July 2006)

Web Application Security Vulnerability Scanners

Need — Strategic

Acquisition — Tactical through 2009



Technology Criteria

- Vulnerability detection
- Continuous, prompt database update
- Reporting, analysis
- Integration with SDLC processes, platforms
- Compliance analysis
- Managed services

Typical Attacks

- Buffer overflow
- SQL injection
- Cross-site scripting
- Parameter injection

Business Criteria

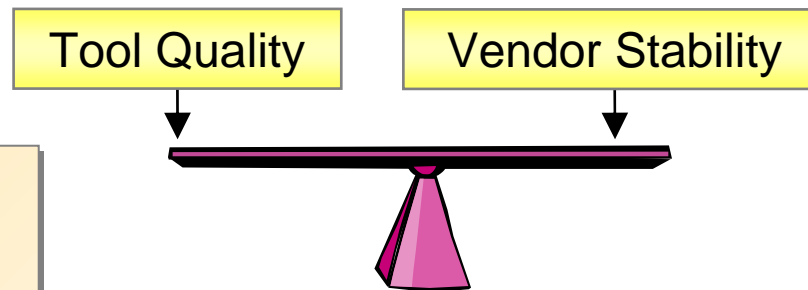
- Market understanding
- Innovation
- Company viability

Vendors	Rating
Acunetix	Caution
Cenzic	Promising
Compuware	Promising
SPI Dynamics	Positive
Watchfire	Positive
WhiteHat	Promising

Source Code Security Vulnerability Scanners

Need — Strategic

Acquisition — Tactical through 2009



Technology Criteria

- Vulnerability detection
- Ability to rate confidence and severity
- Reporting, analysis
- Integration with SDLC processes, platforms
- Performance
- Scalability

Typical Attacks

- Buffer overflow
- Privilege escalations
- Race conditions
- SQL injection

Business Criteria

- Market understanding
- Innovation
- Company viability
- Managed services

Vendors

Compuware

Coverity

Fortify

Klocwork

Ounce Labs

Reasoning Systems

Secure Software

Strategic Planning Assumptions

- By 2008, leading WASVS and SCSVS vendors will combine features of Web application vulnerability detection with source code vulnerability detection (0.8 probability).
- By 2008, 80 percent of the major Software Life Cycle (SLC) vendors will offer WASVS and SCSVS as part of their SLC platform (0.8 probability).
- By 2009, 40 percent of organizations will use a single vendor that provides both source code security scanning and web application security scanning features along SLC (0.7 probability).
- By 2009, 60 percent of IT organizations will make security vulnerability detection an integral part of their SLC processes (0.7 probability).

Security Scanners: Strategic and Tactical Imperatives

- Strategic imperative — WASVS and SCSVS should become an integral part of Software Life Cycle process. A need of having and applying WASVS and SCSVS is strategic and immediate.
- Tactical guideline — Enterprises considering WASVS and SCSVS should expect substantial market and product consolidation within the next 18 months.
- Tactical guideline — Because of market immaturity, when reviewing vendors, enterprises should find a balance between the immediate need to become more secure, vendor stability, and quality and comprehensiveness of their technologies.

Technological Criteria for WASVS selection

- **Integration with SDLC processes and platforms**
- **Ability to conduct WASVS and SCSVS and correlate results**
- **Manageability, scalability**
- **Compliance analysis**
- **Scanning as a service**

Business Criteria for WASVS selection

- **Product/Services**
- **Overall financial and organizational viability**
- **Market understanding**
- **Innovation**

Gap Between Software and Security Professionals

Developers	Security Specialists
Objective — Deliver functionality	Objective — Defend enterprise's perimeter against attacks
Approach — Design, program, test and deploy applications. Security is not our business	Approach — Install software (packages, patches) and hardware: firewalls, IDS, VPN, antiviruses
Make sure they are compliant with user requirements	Make sure only authorized users have access in and out of the perimeter, antiviruses installed
Hackers know security better	Hackers know AD better
Methodology. Various levels of AD methodology maturities. Little/no security engineering methodology	No AD methodology
Result: False sense of security	Result: False sense of security

Recommendation

- Do not expect that Firewall, IDS, VPN will adequately protect your applications. Acquire other tools, starting with Application/Code Security Scanners.
- You will not secure all vulnerabilities. Weight Threats, Vulnerabilities, Impact and Risk to prioritize security measures.
- Security testing should be counter-intuitive: not to prove that that application is functioning properly, but to find that it could function improperly
- AD/IT professionals should accept that app security is their responsibility
- Make app security an integral part of SDLC. Start security with user requirements' analysis, not with operations

Building Secure Applications



Joseph Feiman

Gartner[®]