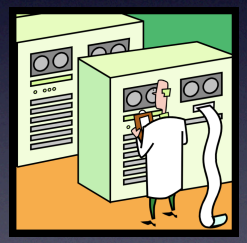
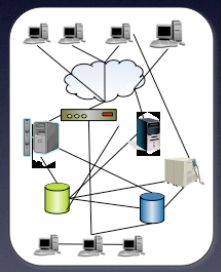
Building a Web Application Security Program

Rich Mogull
Adrian Lane
Securosis, L.L.C.

Old School, New School, Oh SH*& School





Securosis

What's Different About This Presentation

- We are focusing on the business processes of web application security.
- Although we discuss technologies, our goal is to show you how to build a program by putting the pieces together.

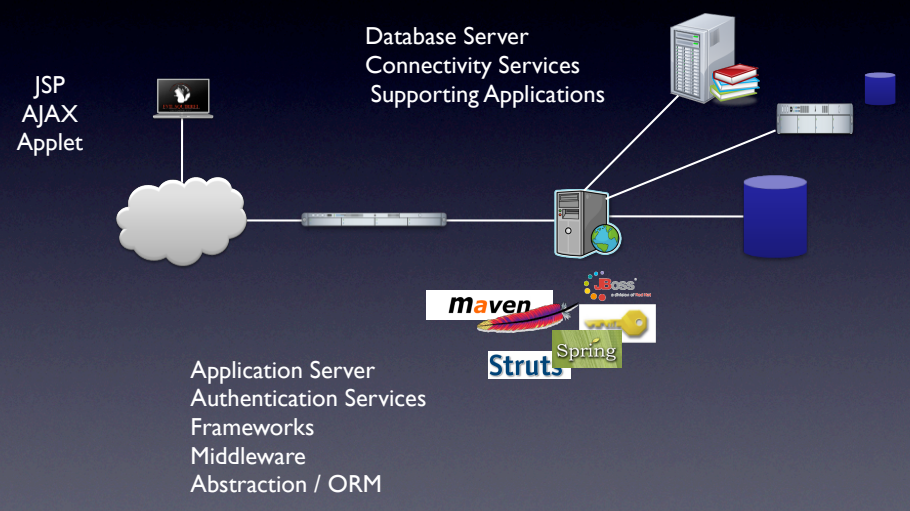
Securosis

- Before web applications, very few businesses exposed their internal transactional systems to the outside world. Even those which did expose systems to business partners on a restricted basis rarely exposed them directly to customers.
- Web applications grew organically — starting from informational websites that were little more than online catalogs, through basic services, to robust online applications connected to critical back-end systems.
- Applications developed slowly over time, with increased functionality leading to increased reliance, often without the oversight they might have gotten had they been designed as massive projects from the beginning. This transition is the classic "frog in a frying pan" problem: Drop a frog into a hot frying pan, and it will hop right out. Slowly increase the heat, and it will fry to death without noticing or trying to escape.
- Web application protocols were designed to be lightweight and flexible; and lacked privacy, integrity, and security features.
- Web application development tools and techniques evolve rapidly, but we still rely on massive amounts of legacy code. Both internal and external systems, once deployed, migrate to new systems but only rarely are we able to clean up any of the existing complex interdependencies.
- Web application threats evolve as quickly as our applications, and apply to everything we've done in the past. We're constantly discovering entirely new classes of vulnerabilities we didn't anticipate before. The web itself was never designed to securely run mission critical applications, so we are building on a platform that was not intended to support its current uses.
- Few web applications are designed securely from the ground up, or maintained with processes designed to keep them secure over time.
- When security breaches do occur, they can be difficult to detect, analyze and measure.

- In light of all these factors, web application security is typically underfunded in most organizations. And like all application development, web applications are subject to time pressures and deadlines that result in deployment tradeoffs and compromised functionality to meet business objectives. We categorize the web application security problems as follows:Few web applications were designed to be secure.
- The application is reliant on a remote web browser which is neither secure nor trusted.
- Web applications evolved to connect our most sensitive back end systems to an open, public network without inherent security controls.
- The web was not designed to be a secure platform, and is thus subject to trivial attacks on confidentiality, integrity and availability.
- Only a small percentage of security breaches are detected and noisy enough to result in measurable losses and/or gain management attention.
- We have a large volume of old application code to fix, while constantly writing new code.
- Web applications are complex combinations of platforms, tools, and services from multiple providers, each with its own security challenges.
- Many web applications are mission critical, but may not have been when they were designed.
- Due to the design of the web (and VPNs), even internal web applications are external applications.
- Web applications are custom applications; we are the vendors, and no one else will provide security fixes.
- Web application projects are funded to offer more services, easier, faster, and cheaper than before, while security tends to limit these benefits.

No single web application security tool provides effective security on its own.

Web App Dependancies



How Web Application Security is Different

- Reliant on browser for UI
- Custom code == custom vulnerabilities
- You are the vendor
- Dynamic content
- Complex platforms and frameworks

Securosis

Reliant on browser for UICustom code == custom vulnerabilitiesYou are the vendorDynamic contentComplex platforms and frameworks

Business Justifications

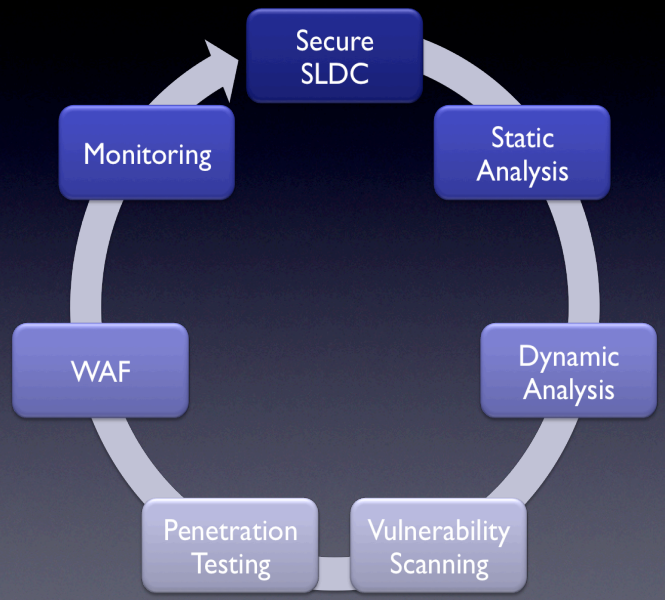
- Compliance
- Fraud Detection/ Reduction
- Cost Savings
- Reputation/ User Protection
- Reduced Breach Costs
- Availability

Securosis

Application Security

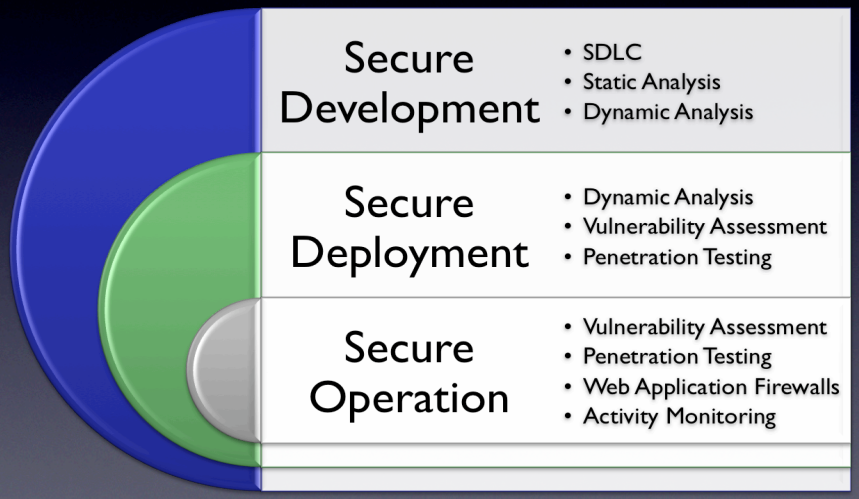
Securosis

Securing Web Applications



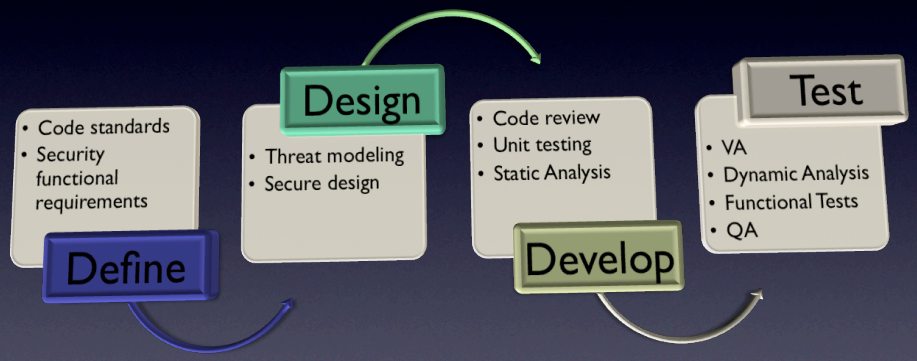
Securosis

Application Security Cycle



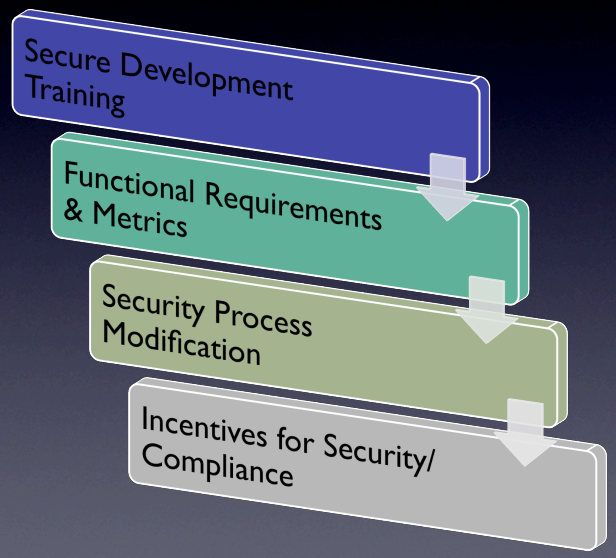
Securosis

Secure Development

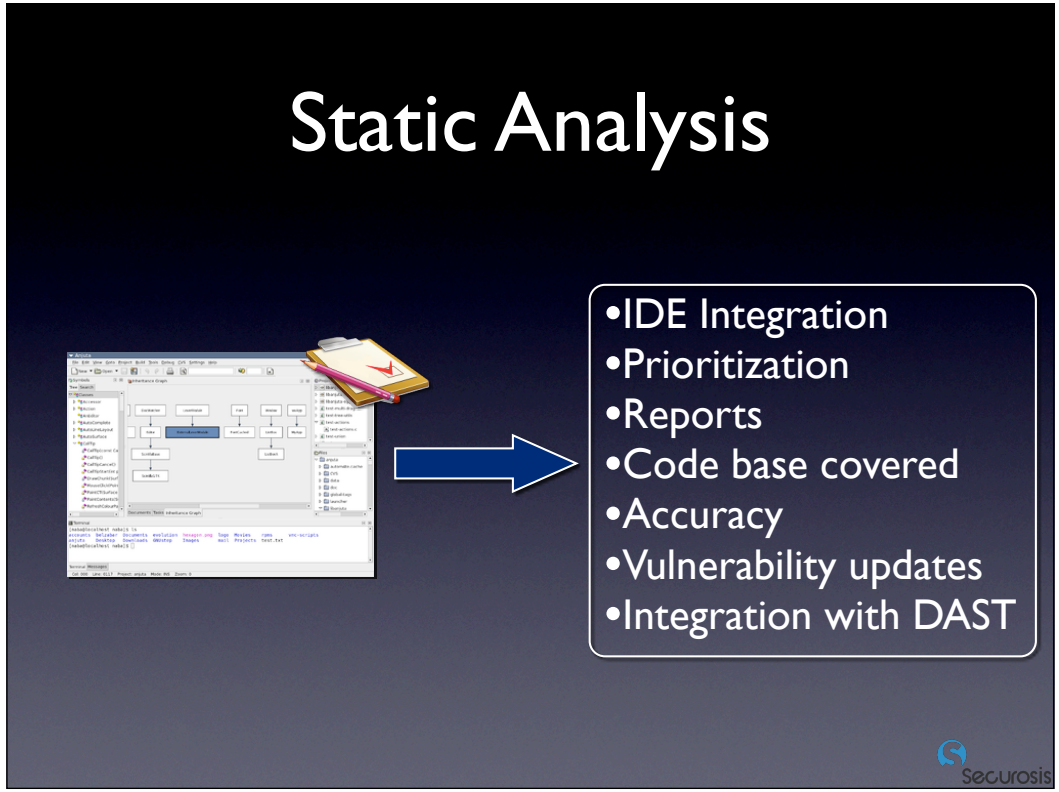


Securosis

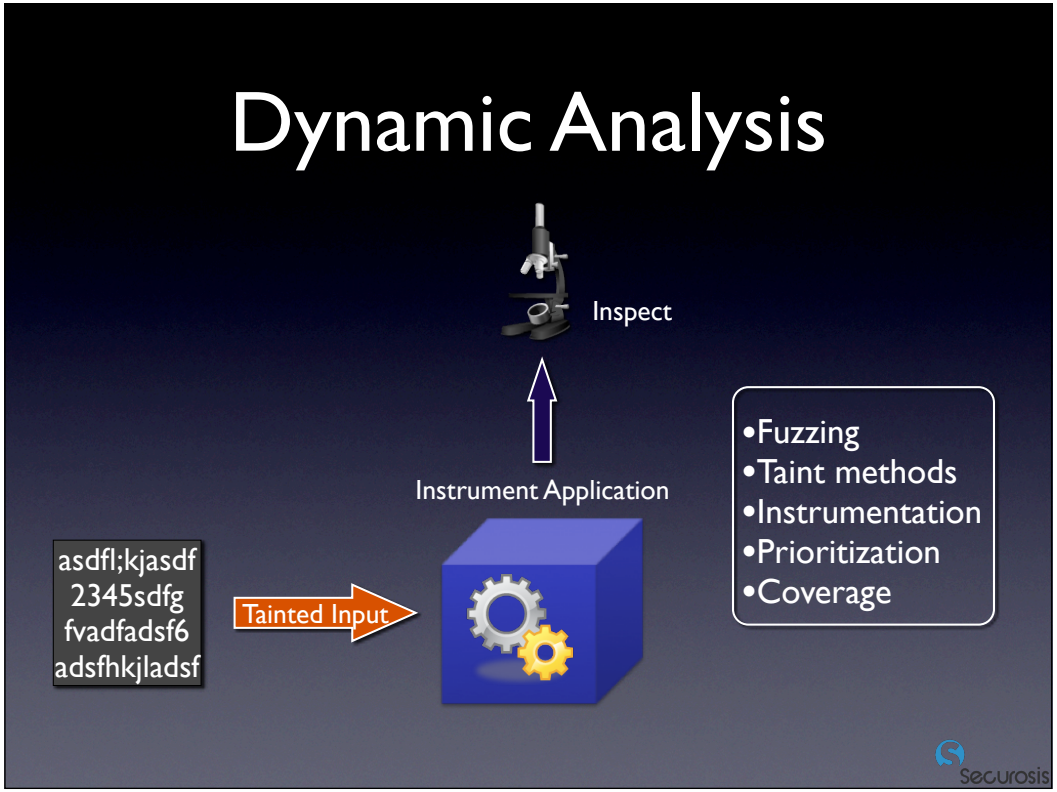
Training & SDLC



Securosis

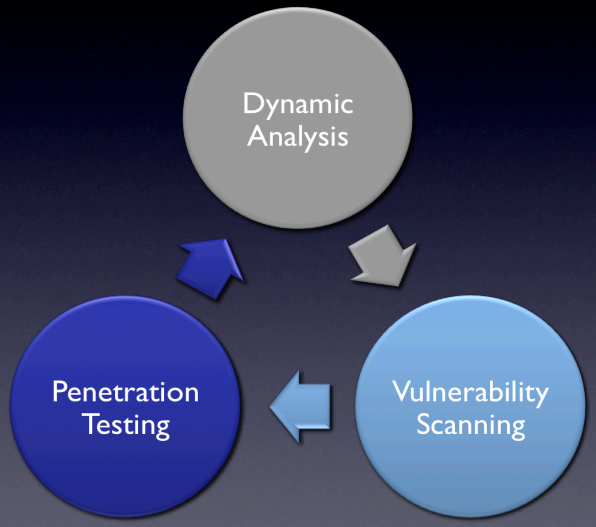


Inspection of codeGood at detecting common errors and poor programming practicesSupplements manual code inspectionDoes not account for all code pathways

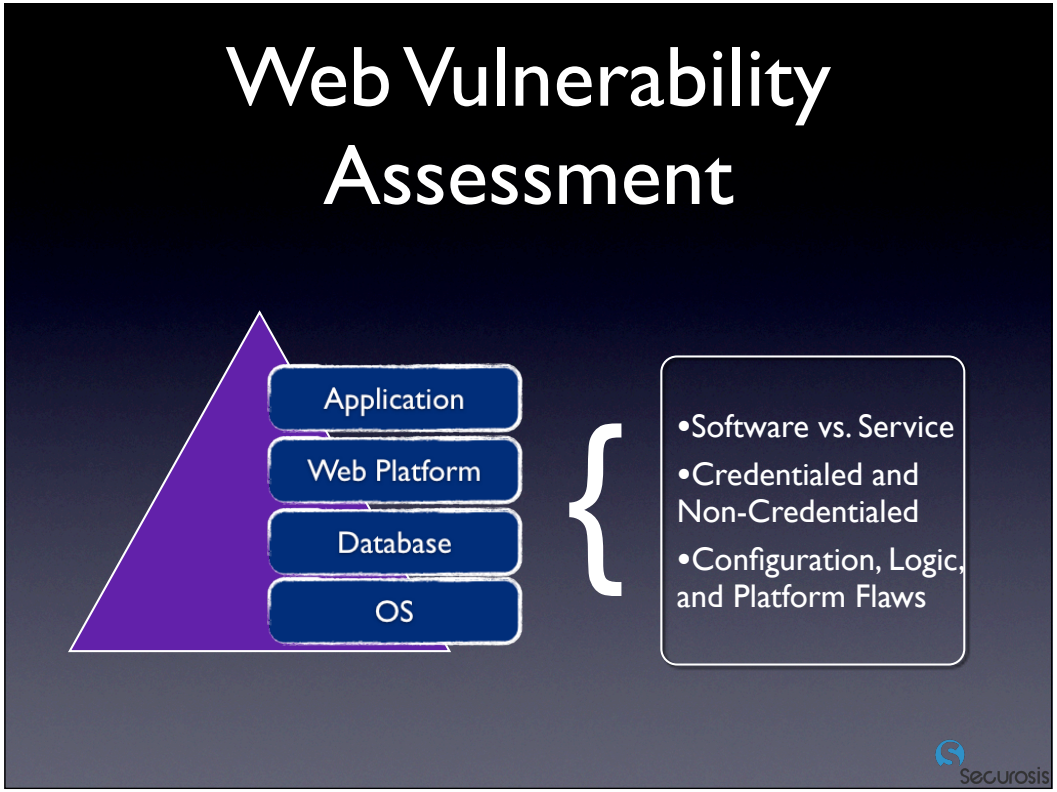


Inspect code during runtime‘Fuzzers’ send intentionally harmful inputWorks with any web applicationBehaves like attacker wouldSkill of user indicative of effectivenessNot a focused analysis

Secure Deployment



Securosis



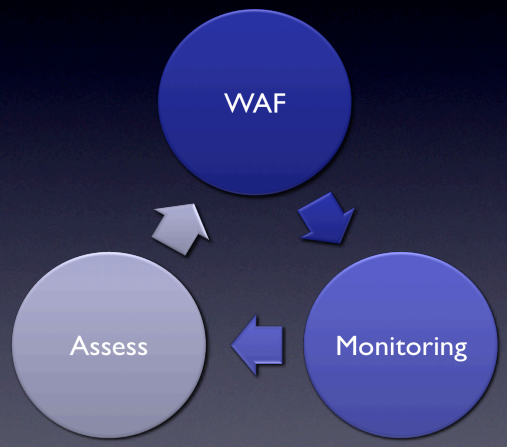
Scans for configuration weaknessesScans for known vulnerabilities
Credentialed and non-credentialed scansAvailable as software and service

Penetration Testing

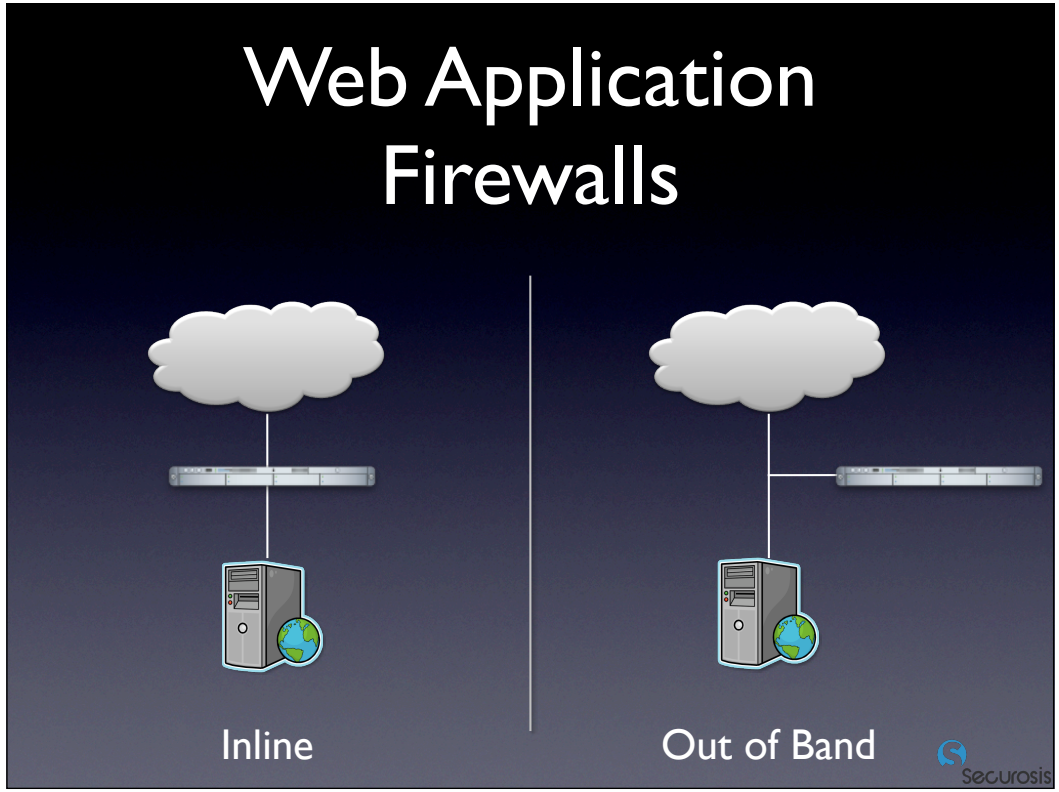
- Pen testing helps measure risk and validates vulnerabilities, it's not about "just breaking in".
- Begun testing in the development process.
- Use a combination of tools and manual process.
- Be extremely cautious about testing live apps.
- Perform periodic testing post-deployment, especially as new exploits appear.

Securosis

Secure Operations



Securosis



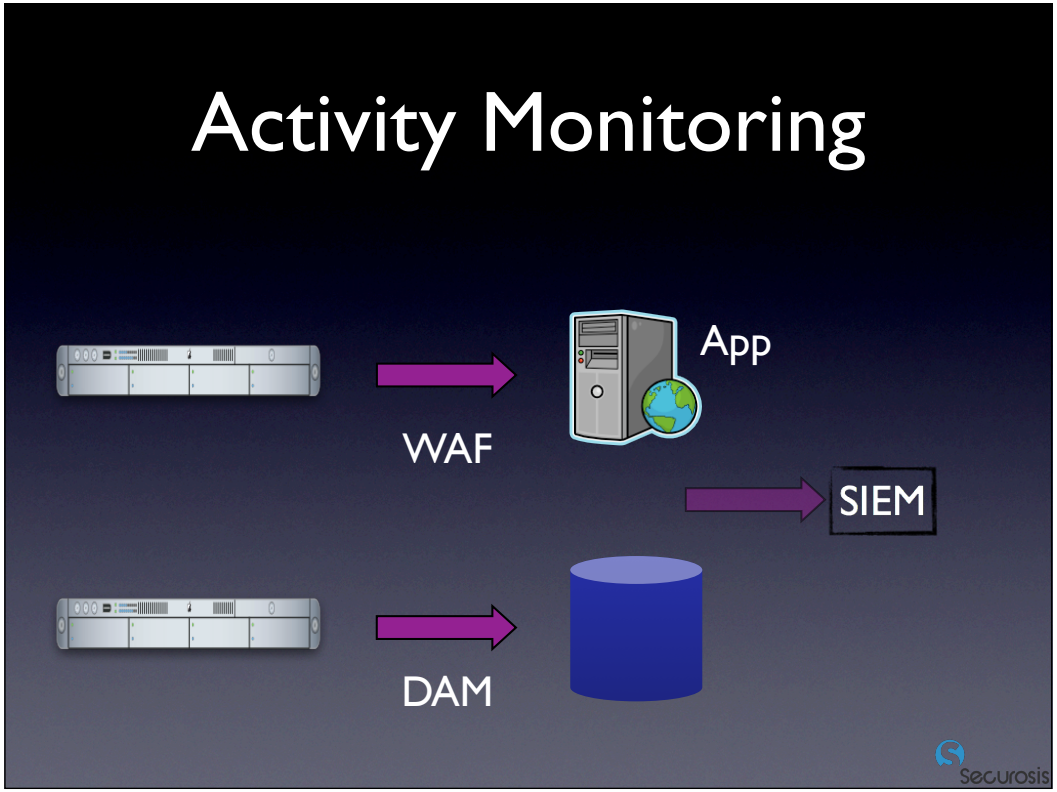
Inspects inbound requests for threatsCan monitor, block or reset connectionsGood for addressing broad classes of threatsRequires significant custom policy development

Web Application Firewalls

- No WAF can ever fully understand a custom application behind it out of the box. These are *not* the same as network firewalls.
- Require extensive tuning for good results.

Securosis

Inspects inbound requests for threatsCan monitor, block or reset connectionsGood for addressing broad classes of threatsRequires significant custom policy development



Monitors activity but does not blockMonitors database activityNo negative impact on business processCost effective

WAF + VA

Asses
- Assess Vulnerabilities
- Validate

Build
- Create a WAF Policy
- Send to WAF

Enforce
- Enforce Policy

Fix
- Patch the vulnerability
- Remove the Policy

Securosis

Use Case: Large Enterprise

- Customer WebApps provide core business functions
- Primary drivers are fraud reduction, compliance, service reliability
- Reputation and asset protection
- Lots of legacy code
- Security expertise, but not in IT or Dev

Securosis

Large Enterprise Recommendations

- Training, Education and Process Improvements
- Secure Software Development Lifecycle
- Heritage Application Plan
- External validation
- Blocking attacks and misuse

Securosis

Use case: Mid-Sized Retailer

- Need to meet PCI-DSS
- Missing basic controls and requirements
- Lack in house security expertise
- Small body of code, but core to business

Securosis

Retailer Recommendations

- Training, Education and Process Improvements
- External Assistance
- Monitoring
- Pen Testing and VA

Securosis

Internal Web Apps

- Dozens of internal web applications
- Supports HR, sales, workflow, BI and partners
- Security and integrity are major concerns
- Insider fraud and breach deterrence are primary drivers

Securosis

Internal WebApp Security Recommendations

- Vulnerability Assessment and Pen Testing
- Training, Education, and Process Improvement
- Monitoring

Securosis

Rich Mogull

Securosis, L.L.C.

rmogull@securosis.com

http://securosis.com

Twitter: rmogull

AIM: securosis

Skype: rmogull

Securosis