UNISYS | Securing Your Tomorrow™

# Application Security Program: Protect Against Data Breaches

**Point of View**

*Gideon T. Rasmussen is a CISO Advisor, available for consulting engagements through Unisys. He has 20 years of experience in corporate and military organizations. Gideon has designed and led programs including PCI - Payment Card Security, Supplier Assessment, Application Security and Information Risk Management. Gideon has authored over 30 information security articles and is a veteran of the United States Air Force.*

*Contact him* ***firstname.lastname@unisys.com*** *or connect with him at* ***LinkedIn****.*

Data breaches are common in today's headlines. Criminal enterprises and hostile nation states have the resources to penetrate infrastructure controls and access data through web application vulnerabilities. Therefore, it is necessary to have an Application Security program in place to protect applications and prevent business impact.

## Awareness and Training

Developers are the first line of defense. Provide them with Secure Code Training. The cost to remediate a code defect increases as it passes into production through the Software Development Life Cycle (SDLC). Anyone who writes code should be required to complete training. Establish a Security Awareness Program with Software Developers and Application Managers as the audience. They need to be apprised of threats to their applications and required controls such as scanning. The program should also communicate application security standards and resources to help make them successful.

## Application Inventory

Register each application within an inventory. Document security related details such as where sensitive data is present, Internet exposure and in-place controls. Use the inventory to drive a risk score for deployment of application security controls. Include primary and alternate contacts to help ensure continuity. Establish a questionnaire to populate the Inventory. Leverage tools and automation to populate data fields where possible, in keeping with the old adage "Trust but Verify".

## Security within the SDLC

Have at least one Application Security Professional assigned to the program. Integrate security checkpoints into SDLC processes. Include security requirements within the application design phase and within projects. That reduces risk and improves quality from a security perspective.

*Have a proactive approach to Application Risk Management. This article provides a mid-level overview of an Application Security Program. It is necessary to establish supporting processes, standards and technologies. Start with a Multi-Generational Plan. Communicate to affected audiences and implement controls in a thoughtful and deliberate manner.*

## Scans and Assessments

Scan and assess code in pre-production environments. Conduct scans when newly developed code is mature, to leave time for remediation before the production release date.

**Static Application Security Testing (SAST):** SAST scans raw software code for security defects. False positives should be eliminated by an Application Security Professional before findings are presented to the Developer.

**Dynamic Application Security Testing (DAST):** DAST scans applications while they are running in a test environment. The Assessor enters the application URL into the DAST tool to configure the scan.

**Penetration (Pen) Testing:** The use of SAST and DAST alone may fail to detect significant vulnerabilities. Pen Testing is used to address that gap and is commonly used for high risk applications. Ethical Hackers leverage suites of security tools to identify vulnerabilities and use that information to gain access. They also use custom scripts, conduct manual tests and strive to exploit business logic. If your organization has a high risk application, it makes sense to test it with hacking techniques.

## WAFs and DBFWs

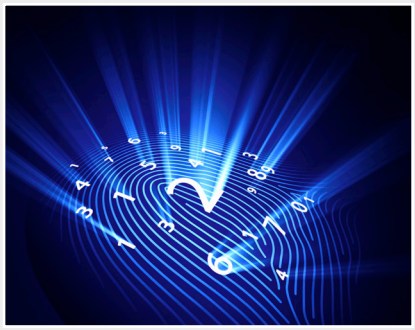
Web Application Firewalls (WAFs) mitigate common attacks such as SQL Injection and Cross-Site Scripting. WAFs provide protection at the application layer, beyond what a network firewall can address. WAFs should be used to protect Internet facing applications that store, process or transmit sensitive data. Attackers target databases directly to access where the sensitive data is stored. Use Database Firewalls (DBFWs) as a preventive control. Use of a DBFW has unanticipated benefits, such detecting a Database Administrator making changes in production during business hours, outside of change management.

## Deployment Model

Establish risk ratings for applications with a set of controls for each tier. The output may be as simple as: Low risk applications are evaluated by SAST. Medium risk applications are subjected to SAST and DAST. High risk applications must pass SAST, DAST and a Pen Test. Account for how applications are evaluated within the SDLC and whether there should be a separate annual requirement as well.

## Remediation

Remediate significant code defects before production. If there is a business priority to implement code with a vulnerability in place, ensure the risk is clearly communicated and the appropriate level of executive sign-off is obtained. Remediate production issues with a sense of urgency, in accordance with policies and standards. Include a feedback loop into Root Cause Analysis (RCA) and the awareness program.



## Risk Transparency

Track security issues within a system of record. Establish reporting and metrics that provide visibility into remediation activity. Provide trending to track remediation progress over time. Establish clear accountability from the Issue Respondent to the Application Manager, up the chain to the Executive. Establish a process to escalate security issues that have exceeded remediation standards.

## Sensitive Data

Limit the presence of sensitive data to reduce risk. Evaluate each application to determine if sensitive data can be removed from it. Replace SSNs and payment card numbers with a customer number or another unique identifier. Truncate SSNs and card numbers, storing only a portion of the number. Delete or archive sensitive information that is no longer required, such as closed accounts.

## Secure Code

Harden applications against attack. Establish a secure coding practices leveraging guidance from the OWASP Proactive Controls List. Provide code that can be used as a preventive control such as a filter that prevents a vulnerability from being exploited. Consider mandates for hardening high risk applications against attack.

## Attack Aware Applications

Work with the security team to monitor for application attacks such as session fixation and attempts to compromise authentication. Start with logging and alerts. Implement automated corrective actions once false positives have been eliminated. Monitor for fraud by an authorized user such as accessing many customer accounts in an hour to commit Identity Theft.

## Developer Belt Program

Improve the maturity of your program by providing your developers with enhanced training and security expertise. Leverage Adobe's ASSET Software Security Certification Program for inspiration. Certifications include white, green, brown and black belt. The white and green belts are based on completion of a series of training modules. The brown belt is achieved by leading a security project such as hardening an application against attack. The black belt is coveted and can only be earned by supporting/leading a group of brown belt candidates. There could be linkage into career advancement based upon belt attainment, given the validation of project management and leadership. Establish requirements for annual recertification through an online refresher course with an assessment at the end.

## Open Source Code

Establish a process to keep Open Source Code on a current version. Custom applications are built from 80% open source libraries and frameworks. If that code is left with known vulnerabilities in place, risk to the application increases. Heartbleed and Shellshock are examples of open source vulnerabilities.


## BSIMM Assessment

BSIMM is an abbreviation for Building Security In Maturity Model. It is a software security framework organized into 113 activities. A BSIMM assessment demonstrates how far your program has matured and provides a snapshot of current state. BSIMM findings can be leveraged to gain funding to remediate control gaps.

## Meetings and Conferences

Participate in OWASP meetings and conferences to learn more about application security and to share your knowledge. BSIMM conferences are also worthwhile events. There is an established BSIMM community. Information is shared between members before and after the events.



02/17 17-0062