OWASP

The Open Web Application Security Project

OWASP
The Open Web Application Security Project

Geraint Williams

16:45 Wednesday 25th, 2014

PCI DSS AND SECURE APPLICATIONS

About Me

- Geraint Williams
  - Senior Consultant and QSA for IT Governance
    - Lead Technical Services Team
    - QSA
    - CREST Registered Tester
  - Visiting Fellow for University of Bedfordshire
    - Subject matter expert
    - Research on wireless and Internet of Things

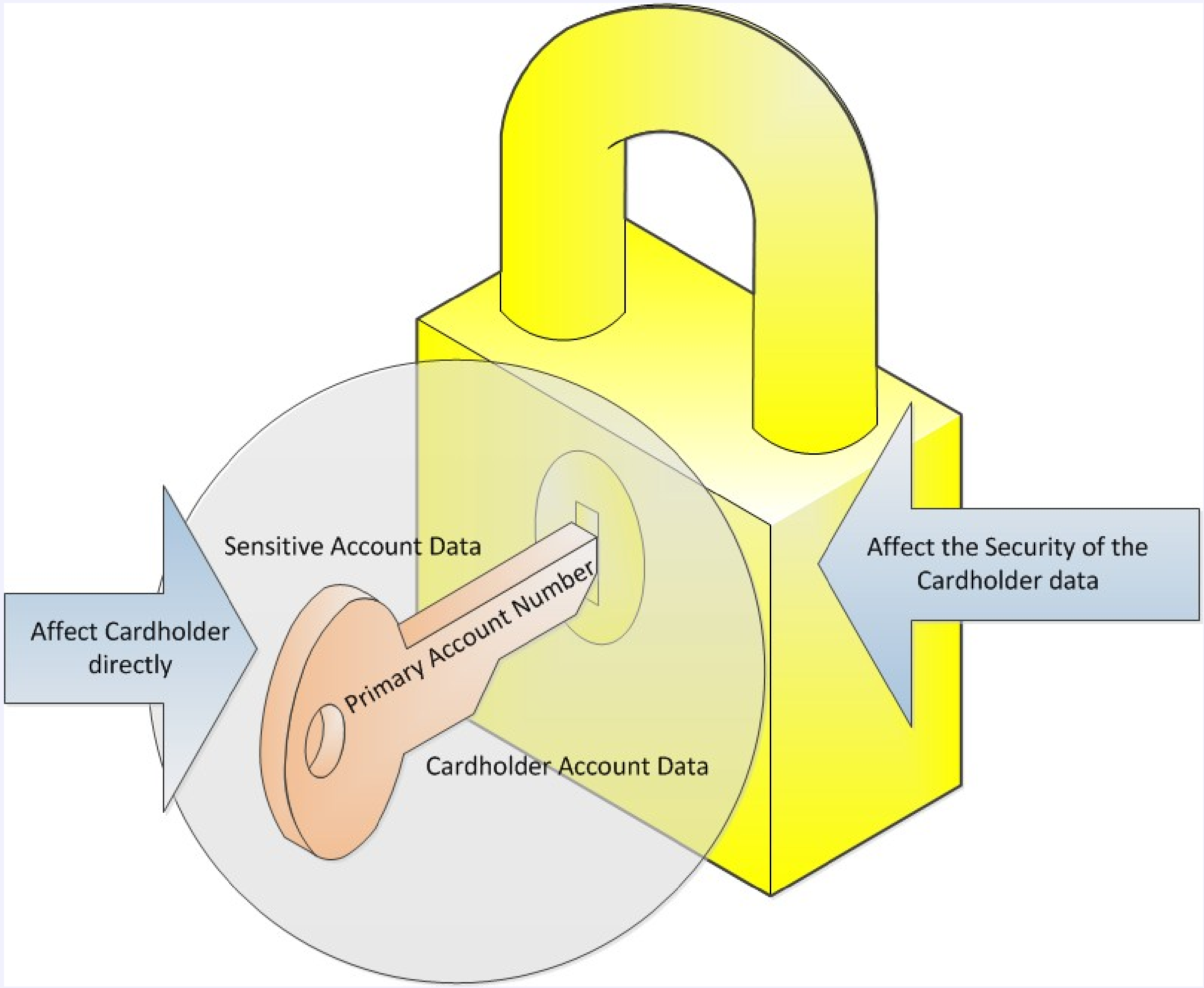
OBJECTIVE

Objective

- Examining the PCI DSS requirements as they apply to software developers
- Explain what a QSA is going to be looking for when examining software development
- Help software developers meet the certification requirements of the PCI DSS

APPLICABILITY OF THE PCI DSS

Applicability


Sensitive Account Data
Affect Cardholder directly
Primary Account Number
Affect the Security of the Cardholder data
Cardholder Account Data

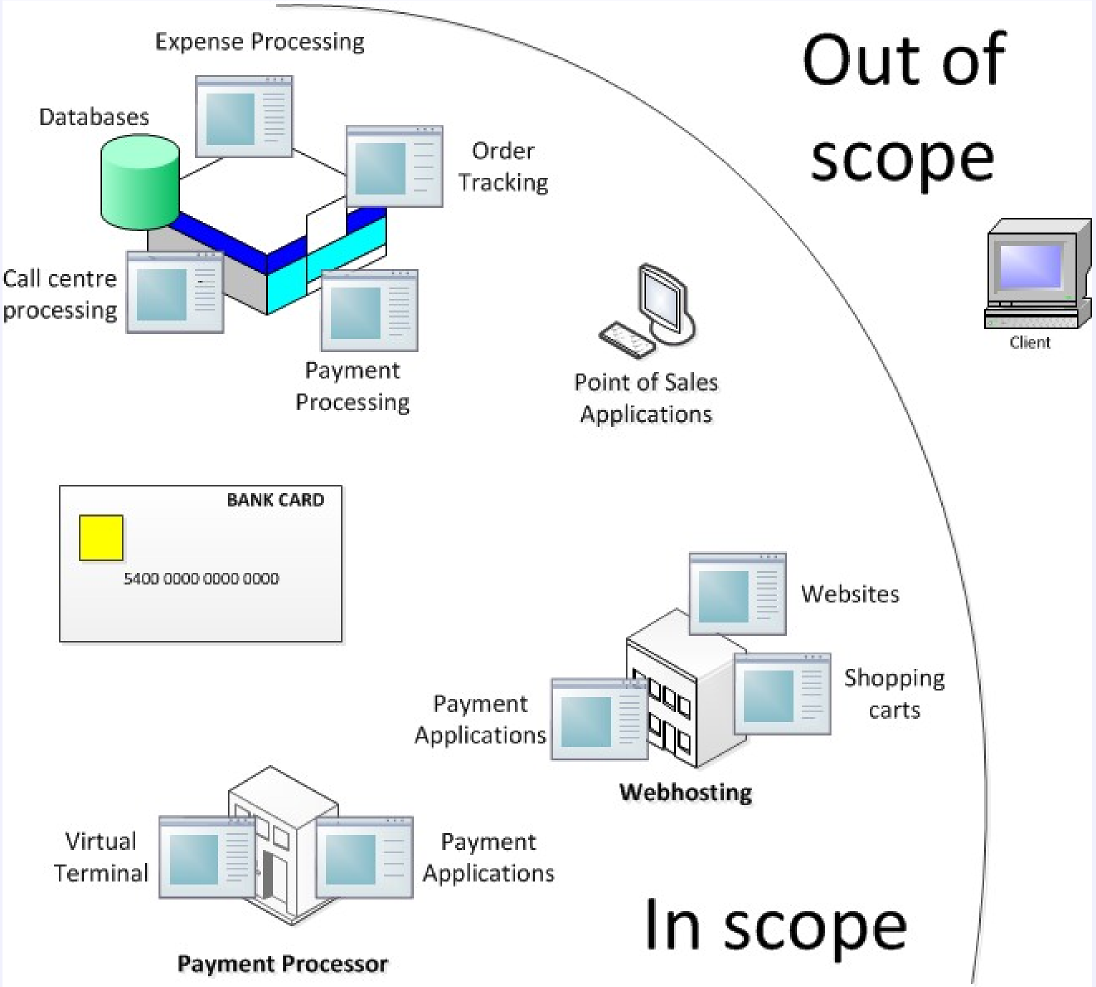
Scope

OWASP
The Open Web Application Security Project

Scope

- The PCI DSS security requirements apply to all system components included in or connected to the cardholder data environment
- The cardholder data environment (CDE) is comprised of people, processes and technologies that store, process, or transmit cardholder data or sensitive authentication data
- The assessed entity determines the cardholder data environment and retains documentation that shows how PCI DSS scope was determined
- The assessor is required to validate that the scope of the assessment is accurately defined and documented.

Scope (Cont)


Out of scope
Expense Processing
Databases
Order Tracking
Call centre processing
Payment Processing
Point of Sales Applications
Client
BANK CARD
5400 0000 0000 0000
Websites
Shopping carts
Payment Applications
Webhosting
Virtual Terminal
Payment Applications
Payment Processor
In scope

Significance of PAN

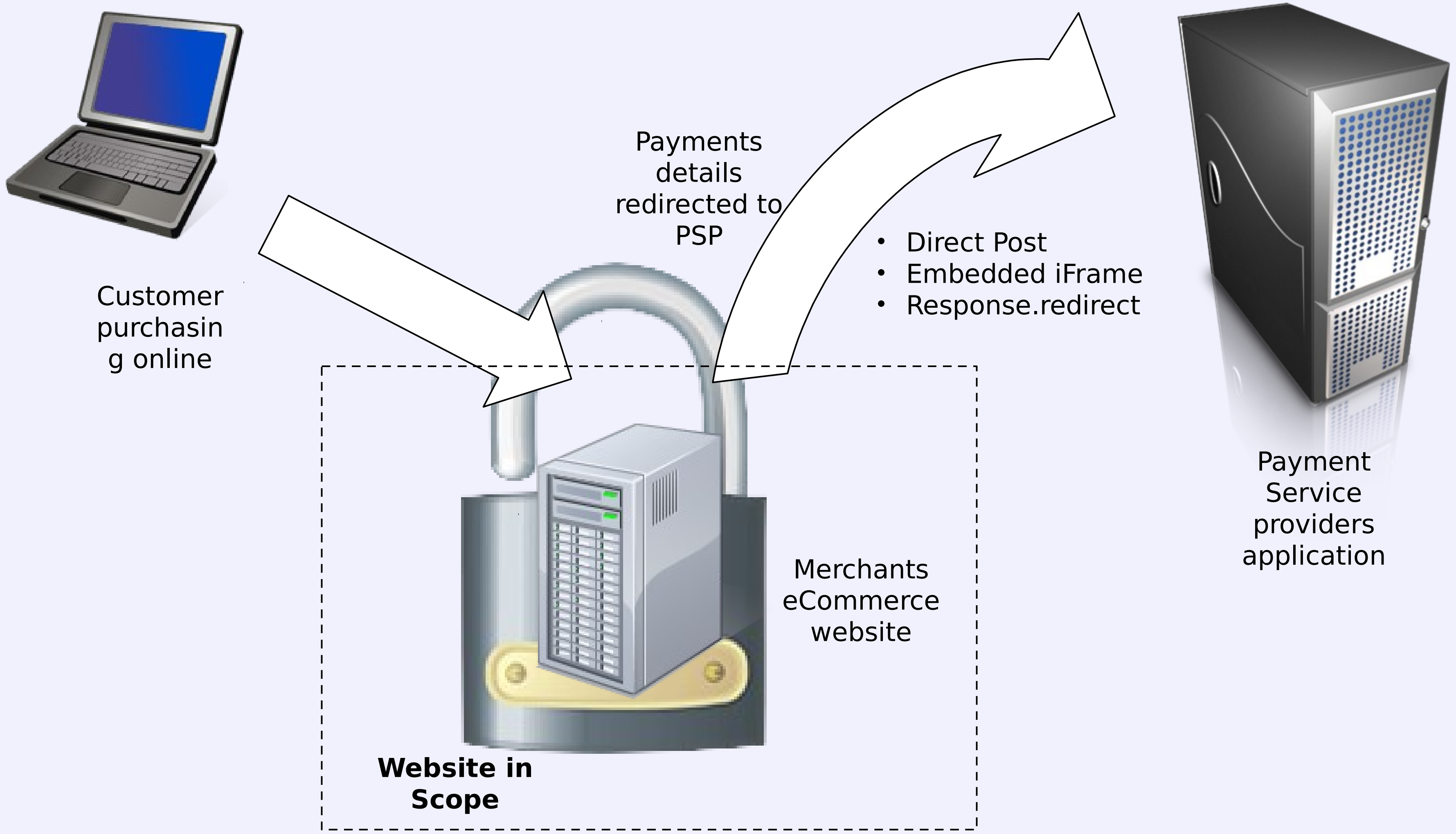
- payment card details captured within
  - expense tracking systems
  - corporate card management
  - etc…
- *Anywhere the PAN is captured, stored, processed or transmitted, even when not directly involved in a payment transaction, the PCI DSS still applies or effects the security of the PAN as it is captured, stored, processed or transmitted*

PA-DSS requirements

Software Development

- PA-DSS Applications
  - ~~sold and installed “off the shelf”~~
  - ~~payment applications provided in modules,~~
- Non PA-DSS Application
  - payment applications offered by application or service providers only as a services
  - non-payment applications modules
  - payment application developed for and sold to a single customer
  - payment applications developed by merchants and service providers if used only in-house

Websites in Scope



Customer
purchasin
g online

Payments
details
redirected to
PSP

- Direct Post
- Embedded iFrame
- Response.redirect

Payment
Service
providers
application

Merchants
eCommerce
website

**Website in
Scope**

PCI DSS REQUIREMENTS V3

PCI DSS v3 Software development requirements

OWASP
The Open Web Application Security Project

Requirement No.	PCI DSS Requirements	Testing Procedures
6.3	3	7
6.4	10	15
6.5	11	14
6.6	1	1
6.7	1	1

Requirement 6.3

6.3 Develop internal and external software applications (including web-based administrative access to applications) securely, as follows:

- In accordance with PCI DSS (for example, secure authentication and logging)
- Based on industry standards and/or best practices.
- Incorporating information security throughout the software-development life cycle

Strong Cryptography

Examples of industry-tested and accepted standards and algorithms for encryption include:

- AES (128 bits and higher)
- TDES (minimum triple-length keys)
- RSA (2048 bits and higher)
- ECC (160 bits and higher), and
- ElGamal (2048 bits and higher)

Authentication

8.3 Implement two-factor authentication for remote access

8.4 Render all passwords unreadable during storage and transmission, by using strong cryptography.

8.5 Ensure proper user identification and authentication management for non-consumer users and administrators.

Requirement 10: Logging

10.1 Establish a process for linking all access to system components to each individual user – especially access done with administrative privileges.

10.2 Implement automated audit trails for all system components for reconstructing these events: all individual user accesses to cardholder data; all actions taken by any individual with root or administrative privileges; access to all audit trails; invalid logical access attempts; use of identification and authentication mechanisms; initialization of the audit logs; creation and deletion of system-level objects.

10.3 Record audit trail entries for all system components for each event, including at a minimum: user identification, type of event, date and time, success or failure indication, origination of event, and identity or name of affected data, system component or resource.

10.5 Secure audit trails so they cannot be altered.

Requirement 6.4

6.4 Follow change control processes and procedures for all changes to system components.

Requirement 6.5

6.5 Address common coding vulnerabilities in software-development processes as follows:

- Train developers in secure coding techniques, including how to avoid common coding vulnerabilities, and understanding how sensitive data is handled in memory.
- Develop applications based on secure coding guidelines.

Requirement 6.6

OWASP
The Open Web Application Security Project

6.6 For public-facing web applications, address new threats and vulnerabilities on an ongoing basis and ensure these applications are protected against known attacks by either of the following methods:

- Reviewing public-facing web applications via manual or automated application vulnerability security assessment tools or methods, at least annually and after any changes
- Installing an automated technical solution that detects and prevents web-based attacks (for example, a web-application firewall) in front of public-facing web applications, to continually check all traffic.

Requirement 6.7

6.7 Ensure that security policies and operational procedures for developing and maintaining secure systems and applications are documented, in use, and known to all affected parties.

KEY PRACTICES

Key practices

OWASP
The Open Web Application Security Project

- Secure software development lifecycle practices that ensure the inclusion of security during the requirements definition, design, analysis, and testing phases of software development.
- Requiring developers to understand how cardholder data is handled in memory, and how modern malware will scrape memory to retrieve sensitive data.
- The use of separate development, testing and production environments; including separation of duties for developers, testers and production administrators.
- The need to remove test account credentials and test data from application before it is released to the production environment.

Key practices (cont)

OWASP
The Open Web Application Security Project

- Prohibition of the use of ‘live’ data for testing or development purposes.
- The use of change control mechanisms to ensure all changes to system components are reviewed and authorised.
- Software developers are trained in secure coding techniques and develop applications on secure coding guidelines.
- The testing of applications to ensure they do not suffer from known vulnerabilities.
- Public facing web applications are protected against known attacks.

Key practices vs requirements

	6.3	6.4	6.5	6.6	6.7
Secure software development lifecycle practices that ensure the inclusion of security during the requirements definition, design, analysis, and testing phases of software development.	✓				✓
The use of separate development, testing and production environments; including separation of duties for developers, testers and production administrators	✓				✓
The need to remove test account credentials and test data from application before it is released to the production environment.	✓				✓
The testing of applications to ensure they do not suffer from known vulnerabilities.	✓	✓			✓
Prohibition of the use of ‘live’ data for testing or development purposes.		✓			✓
The use of change control mechanisms to ensure all changes to system components are reviewed and authorised.		✓			✓
Public facing web applications are protected against known attacks.				✓	✓
Requiring developers to understand how cardholder data is handled in memory, and how modern malware will scrape memory to retrieve sensitive data.			✓		✓
Software developers are trained in secure coding techniques and develop applications on secure coding guidelines.			✓		✓

Process Maturity

Competencies

OWASP
The Open Web Application Security Project

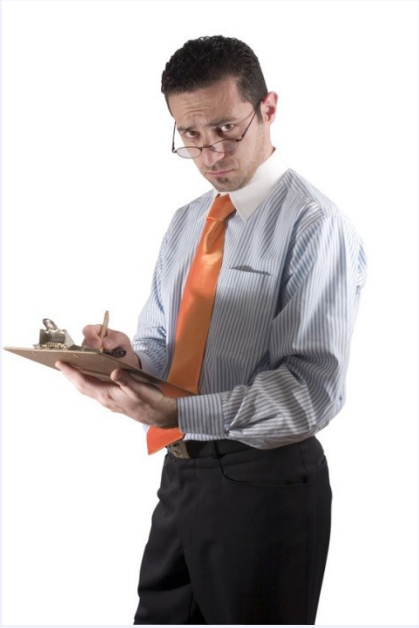
	6.3	6.4	6.5	6.6	6.7
SAQ A	□	□	□	□	□
SAQ A-EP	□	☑	☑	☑	☑
SAQ B	□	□	□	□	□
SAQ B-IP	□	□	□	□	□
SAQ C-VT	□	□	□	□	□
SAQ C	□	□	□	□	□
SAQ P2PE-HW	□	□	□	□	□
SAQ D (Merchant & Service Provider)	☑	☑	☑	☑	☑

QSA

QSA

OWASP
The Open Web Application Security Project

- Qualified Security Assessor



Knowledge

Certification

Experience

Industry Best Practice

- Industry best practices
  - OWASP Guide
  - SANS CWE Top 25
  - CERT Secure Coding
  - etc.

Key Practice 1

- Secure software development lifecycle practices that ensure the inclusion of security during the requirements definition, design, analysis, and testing phases of software development.

  - *Formal mature design methodology*
  - *Specific policies and procedures*
  - *Evidence*

Key Practice 2

- Requiring developers to understand how cardholder data is handled in memory, and how modern malware will scrape memory to retrieve sensitive data.

- *Competence of Developers*
- *Continuous professional development*

Key Practice 3

- The use of separate development, testing and production environments; including separation of duties for developers, testers and production administrators.
  - *Specific policies and procedures*
  - *Physical & logical segregation*
  - *Formal approval procedure*
  - *Sign off by management*
  - *Competence of project managers*

Key Practice 4

- The need to remove test account credentials and test data from application before it is released to the production environment.

  - *Formal mature design methodology*
    - *Specific policies and procedures*
      - *Evidence*

Key Practice 5

- Prohibition of the use of ‘live’ data for testing or development purposes.

- *Formal mature design methodology*
- *Specific policies and procedures*
- *Source of ‘test’ data*

Key Practice 6

- The use of change control mechanisms to ensure all changes to system components are reviewed and authorised.

  - *Formal mature design change*
  - *Sign off by management*
  - *Recording of evidence*
  - *Documentation*

Key Practice 7

- Software developers are trained in secure coding techniques and develop applications on secure coding guidelines.

- *Competence of Developers*
- *Continuous professional development*
- *Methodology*
- *Tools*

Key Practice 8

- The testing of applications to ensure they do not suffer from known vulnerabilities.

  - *Competence of testers*
  - *Segregation of testers*
  - *Methodology*
  - *Tools*

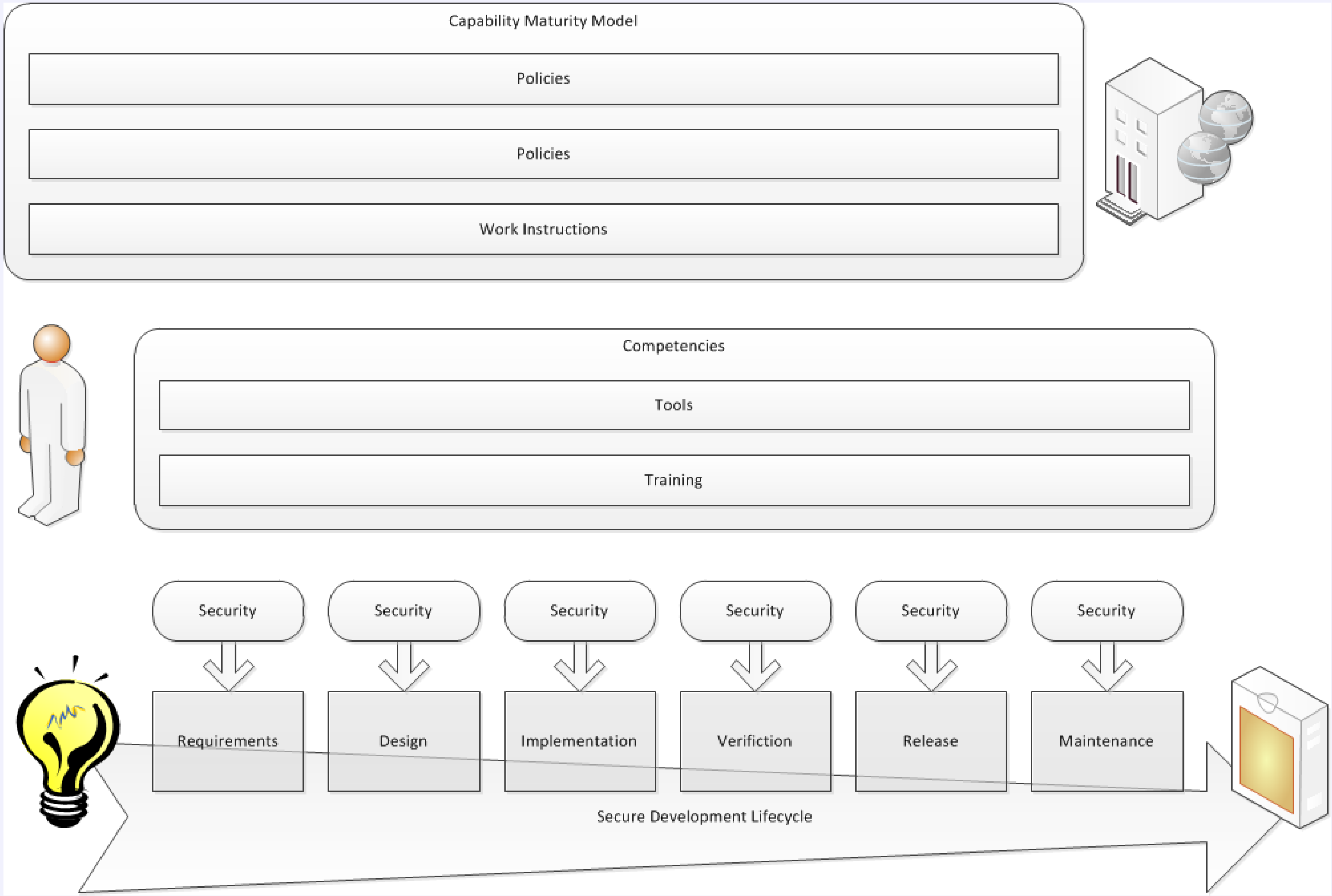
Key Practice 9

- Public facing web applications are protected against known attacks.

  - *Methodology*
  - *Competence of testers*
  - *Segregation of testers*
  - *Tools*

Secure Development Lifecycle

OWASP
The Open Web Application Security Project



Demonstrating competence

- (ISC)2 exam - CSSLP
- GIAC Exams – GSSP
- EC-Council – Certified Secure Programmer

- OWASP Certification Project - DEPRECATED

OWASP Projects

OWASP
The Open Web Application Security Project

- OpenSAMM
- Software Assurance Maturity Model (SAMM)
  - Evaluating an organization’s existing software security practices
  - Building a balanced software security assurance program in well-defined iterations
  - Demonstrating concrete improvements to a security assurance program
  - Defining and measuring security-related activities throughout an organization

- OWASP Developer Guide
- is a "first principles" book
- The major themes in the Developer Guide include:
  - Foundation
  - Architecture
  - Design
  - Build
  - Configure
  - Operate

- OWASP Code Review Guide
- this guide focuses on the mechanics of reviewing code for certain vulnerabilities, and provides limited guidance on how the effort should be structured and executed

- OWASP Secure Coding Practices - Quick Reference Guide
- is a technology agnostic set of general software security coding practices, in a comprehensive checklist format, that can be integrated into the development lifecycle.

OWASP
The Open Web Application Security Project

- OWASP Testing Guide
- The aim of the project is to help people understand the what, why, when, where, and how of testing web applications.
- The project has delivered a complete testing framework, not merely a simple checklist or prescription of issues that should be addressed.

- OWASP PCI Project
- The PCI toolkit is based on a decision tree assessment methodology, which helps you identify if your web applications are part of the PCI-DSS scope and how to apply the PCI-DSS requirements.

CONCLUSION

- Can train developers, but need them to put it into practice
- Good practice is often not documented or evidence generated
- QSA’s need to be able to understand software development
- Specialist QSA’s for the PA-DSS

QUESTIONS

Any Questions ?

OWASP
The Open Web Application Security Project

Contact details

Blogs

geraintw.blogspot.co.uk
wirelessmscresearch.blogspot.co.uk
blog.itgovernance.co.uk/author/geraint-williams/

Linkedin

uk.linkedin.com/in/geraintpwilliams

Twitter

twitter.com/#!/GeraintW

Personal website

www.geraintw.co.uk