Семинар RISSPA по безопасности данных индустрии платежных карт
27 февраля 2014, «Холидей Инн Санкт-Петербург Московские Ворота»

Информзащита
Системный интегратор

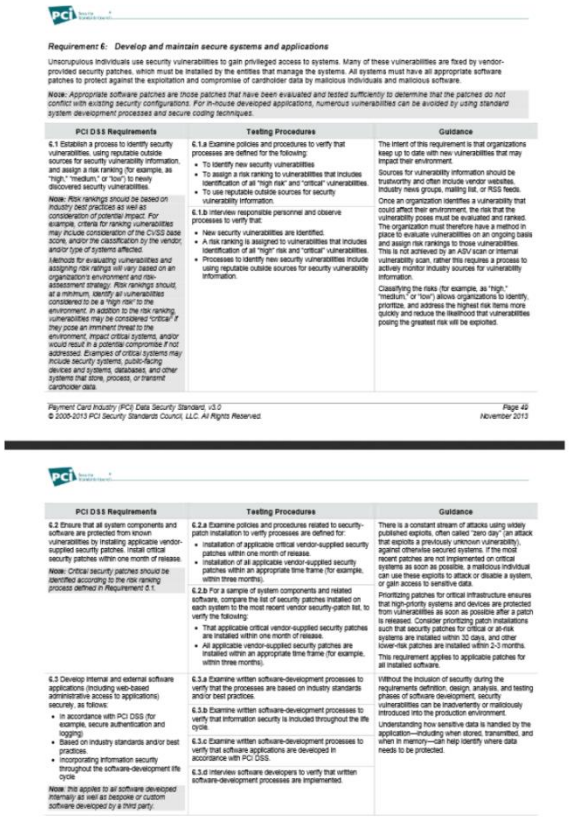
Разработка ПО в рамках PCI DSS

как ее видит жуткий зануда

Алексей Бабенко
руководитель направления, PA&PCI QSA

«Стандартный» взгляд

- Набор требований, которые должны выполнить программисты?
- 6.3-6.5 стандарта?
- Применимо только при наличии отдельного подразделения разработчиков?
- Отдельный процесс, осуществляемый разработчиками



Requirement 6: Develop and maintain secure systems and applications

Unscrupulous individuals use security vulnerabilities to gain privileged access to systems. Many of these vulnerabilities are fixed by vendor-provided security patches, which must be installed by the entities that manage the systems. All systems must have all appropriate software patches to protect against the exploitation and compromise of cardholder data by malicious individuals and malicious software.

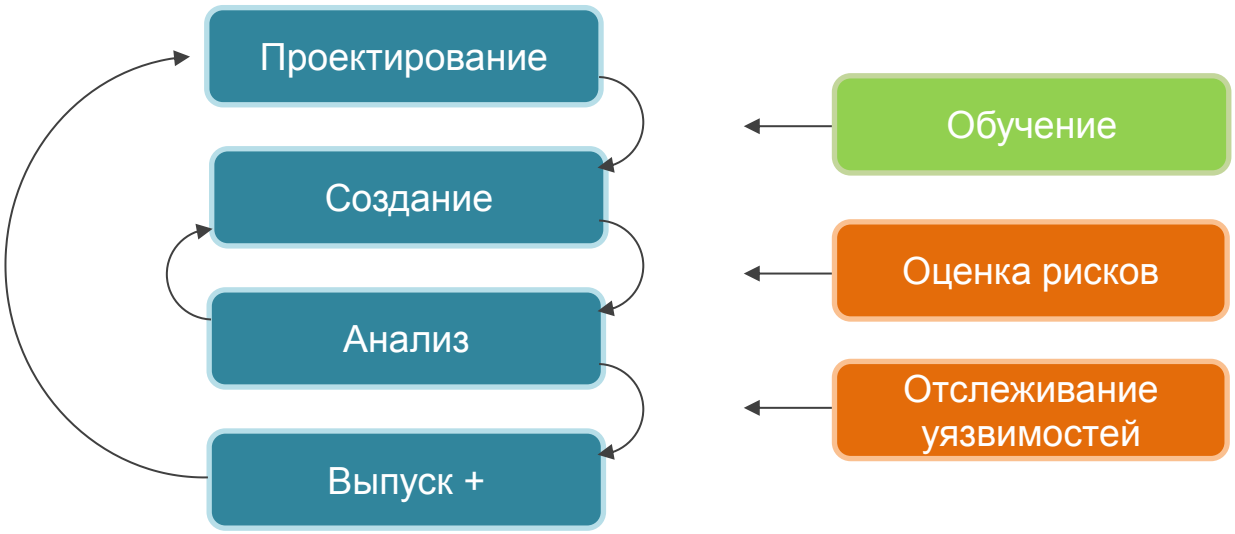
Note: Appropriate software patches are those patches that have been evaluated and tested sufficiently to determine that the patches do not conflict with existing security configurations. For in-house developed applications, numerous vulnerabilities can be avoided by using standard system development processes and secure coding techniques.

PCI DSS Requirements	Testing Procedures	Guidance
6.1 Establish a process to identify security vulnerabilities, using reputable outside sources for security vulnerability information, and assign a risk ranking (for example, as "high," "medium," or "low") to newly discovered security vulnerabilities. *Note: Risk rankings should be based on industry best practices as well as consideration of potential impact. For example, criteria for ranking vulnerabilities may include consideration of the CVSS base score, and/or the classification by the vendor, and/or type of systems affected.* *Methods for evaluating vulnerabilities and assigning risk ratings will vary based on an organization's environment and risk-assessment strategy. Risk rankings should, at a minimum, identify all vulnerabilities considered to be a "high risk" to the environment. In addition to the risk ranking, vulnerabilities may be considered "critical" if they pose an imminent threat to the environment, impact critical systems, and/or would result in a potential compromise if not addressed. Examples of critical systems may include security systems, public-facing devices and systems, databases, and other systems that store, process, or transmit cardholder data.*	6.1.a Examine policies and procedures to verify that processes are defined for the following: • To identify new security vulnerabilities • To assign a risk ranking to vulnerabilities that includes identification of all "high risk" and "critical" vulnerabilities. • To use reputable outside sources for security vulnerability information. 6.1.b Interview responsible personnel and observe processes to verify that: • New security vulnerabilities are identified. • A risk ranking is assigned to vulnerabilities that includes identification of all "high" risk and "critical" vulnerabilities. • Processes to identify new security vulnerabilities include using reputable outside sources for security vulnerability information.	The intent of this requirement is that organizations keep up to date with new vulnerabilities that may impact their environment. Sources for vulnerability information should be trustworthy and often include vendor websites, industry news groups, mailing list, or RSS feeds. Once an organization identifies a vulnerability that could affect their environment, the risk that the vulnerability poses must be evaluated and ranked. The organization must therefore have a method in place to evaluate vulnerabilities on an ongoing basis and assign risk rankings to those vulnerabilities. This is not achieved by an ASV scan or internal vulnerability scan, rather this requires a process to actively monitor industry sources for vulnerability information. Classifying the risks (for example, as "high," "medium," or "low") allows organizations to identify, prioritize, and address the highest risk items more quickly and reduce the likelihood that vulnerabilities posing the greatest risk will be exploited.

PCI DSS Requirements	Testing Procedures	Guidance
6.2 Ensure that all system components and software are protected from known vulnerabilities by installing applicable vendor-supplied security patches. Install critical security patches within one month of release. *Note: Critical security patches should be identified according to the risk ranking process defined in Requirement 6.1.*	6.2.a Examine policies and procedures related to security-patch installation to verify processes are defined for: • Installation of applicable critical vendor-supplied security patches within one month of release. • Installation of all applicable vendor-supplied security patches within an appropriate time frame (for example, within three months). 6.2.b For a sample of system components and related software, compare the list of security patches installed on each system to the most recent vendor security-patch list, to verify the following: • That applicable critical vendor-supplied security patches are installed within one month of release. • All applicable vendor-supplied security patches are installed within an appropriate time frame (for example, within three months).	There is a constant stream of attacks using widely published exploits, often called "zero day" (an attack that exploits a previously unknown vulnerability), against otherwise secured systems. If the most recent patches are not implemented on critical systems as soon as possible, a malicious individual can use these exploits to attack or disable a system, or gain access to sensitive data. Prioritizing patches for critical infrastructure ensures that high-priority systems and devices are protected from vulnerabilities as soon as possible after a patch is released. Consider prioritizing patch installations such that security patches for critical or at-risk systems are installed within 30 days, and other lower-risk patches are installed within 2-3 months. This requirement applies to applicable patches for all installed software.
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. *Note: This applies to all software developed internally as well as bespoke or custom software developed by a third party.*	6.3.a Examine written software-development processes to verify that the processes are based on industry standards and/or best practices. 6.3.b Examine written software-development processes to verify that information security is included throughout the life cycle. 6.3.c Examine written software-development processes to verify that software applications are developed in accordance with PCI DSS. 6.3.d Interview software developers to verify that written software-development processes are implemented.	Without the inclusion of security during the requirements definition, design, analysis, and testing phases of software development, security vulnerabilities can be inadvertently or maliciously introduced into the production environment. Understanding how sensitive data is handled by the application—including when stored, transmitted, and when in memory—can help identify where data needs to be protected.

Информзащита
Системный интегратор

Процессный подход



Информзащита
Системный интегратор

Обучение

- Знать – уметь – использовать
- Основные темы:
  - требования PCI DSS
  - приемы безопасной разработки (OWASP, CWE), проектирования, тестирования, пр.
  - лучшие практики отрасли
- Форма обучения:
  - самостоятельное
  - внутреннее
  - внешнее
- Периодические обучение – хорошо, но лучше непрерывный процесс

Информзащита
Системный интегратор

Проектирование

- Не всегда использование номеров карт целесообразно
- Формирование требований:
  - Требования PCI DSS (ПО = системный компонент)
  - Учет лучших практик
  - Учет внутренних требований по ИБ
  - Учет оценки рисков и информации об уязвимостях
- Проектирование с учетом сформированных требований

Информзащита
Системный интегратор

Создание

- Требования к разработке:
  - Использование методов безопасного программирования
  - Применение лучших практик
  - Учет требований PCI DSS
  - Учет оценки рисков и анализа угроз
- Разделение сред и обязанностей
- Корректное использование тестовых данных

Информзащита
Системный интегратор

Анализ

- Анализ кода (белый ящик)
  - Использование приемов безопасной разработки
  - Автоматизированный анализ только в качестве инструмента в руках специалиста
  - Проводит специалист с опытом в данной сфере не являющийся автором кода
  - Анализ для всех изменений на предмет безопасности
- Тестирование безопасности (черный ящик):
  - Общефункциональное с учетом выявленных уязвимостей
  - Тестирование безопасности:
    - Реализация требований PCI DSS
    - Меры по защите от известных уязвимостей
    - Меры предотвращения недостатков выявленных в модели угроз и при анализе новых уязвимостей
  - Тестирование по методике – хорошо, с использованием контрольных чек-листов – лучше.

Выпуск

- Решение за выпуск релиза:
  - Предыдущие этапы были выполнены успешно
  - Требования стандарта учтены
  - Разработаны процедуры «отката»
  - Оценка воздействия на затрагиваемый процесс
- Ответственный за выпуск релиза – отвечает за факт выполнения процедур, ответственность за качество выполнения на соответствующих специалистах
- Релиз – не конец проекта

Информзащита
Системный интегратор

Выявление уязвимостей

- Выявляем все что может влиять на безопасность ПО:
  - Новые уязвимости и слабости в функциях программирования
  - Уязвимости компиляторов
  - Уязвимости используемых сторонних библиотек, компонент, сервисов, протоколов
- Корректная работа с последними обновлениями системных компонент
- Учет результатов при проектировании, разработке, анализов

Информзащита
Системный интегратор

Анализ угроз

- В рамках общего анализа рисков
- Стандарт – минимальный набор требований, каждое ПО уникально и может иметь специфичные угрозы
- Моделирование угроз – STRIDE, OWASP, CERT
- Учет результатов анализа при проектировании, разработке, анализе

Информзащита
Системный интегратор

Следующий шаг

- Выполнение требований PA-DSS
- Создание аналога руководства по применению PA-DSS – документа, описывающего как ПО выполняет требования стандарта и какими настройками это обеспечивается
- Проведение внешнего обучения
- Проведение независимой оценки ПО
- Выделение отдельного подразделение – обеспечения безопасности ПО

Информзащита
Системный интегратор

Сравнение SDL

PCI DSS Development Lifecycle	Cisco Secure Development Lifecycle	Microsoft Security Development Lifecycle
Обучение	Secure Design	Training
Выявление уязвимостей	3rd Party Security	Implementation (частично)
Проектирование	Product Security Requirements	Requirements
Оценка рисков	Secure Design	Design
Создание	Secure Coding	Implementation
Анализ кода	Secure Analysis	Implementation
Тестирование безопасности	Vulnerability Testing	Verification, Release
Выпуск	-	Release
Поддержка	-	Response

ВОПРОСЫ?

Алексей Бабенко
руководитель направления, PA&PCI QSA

a.babenko@infosec.ru
+ 7 (495) 980-23-45 #458
+ 7 905 991-99-19

skype: arekusux
arekusux.blogspot.com

www.infosec.ru

Информзащита
Системный интегратор

Полезные ссылки

- Безопасное программирование
  - http://cwe.mitre.org
  - http://owasp.org
- Общие базы данных уязвимостей
  - http://www.securityfocus.com
  - http://nvd.nist.gov
  - http://secunia.com
- Информация по внешнему обучению
  - http://itsecurity.ru/catalog/kp75
  - http://www.sans.org/security-training.php
  - https://www.owasp.org/index.php/Category:OWASP_AppSec_Conference
  - http://www.giac.org/certification/gssp-java
- Материалы для организации внутреннего обучения:
  - https://www.owasp.org/index.php/Category:OWASP_Code_Review_Project
  - https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
  - http://www.sans.org/top25-software-errors
  - http://projects.webappsec.org/w/page/13246978/Threat-Classification
  - http://www.cert.org/secure-coding
  - http://cwe.mitre.org/data/graphs/699.html
- Материалы с сайта консула:
  - https://www.pcisecuritystandards.org/pdfs/PCI_DSS_v2_eCommerce_Guidelines.pdf
  - https://www.pcisecuritystandards.org/documents/information_supplement_6.6.pdf
  - https://www.pcisecuritystandards.org/documents/Mobile_Payment_Security_Guidelines_Developers_v1.pdf