конференция РусКрипто'2010

Тестирование уязвимостей приложений в рамках сертификации по PA-DSS

Сапожников Антон

Информзащита
Системный интегратор

Who is who?

1. PCI DSS aka Payment Card Industry Data Security Standard

   Стандарт безопасности данных индустрии платежных карт

2. PA DSS aka Payment Application Data Security Standard

   Стандарт безопасности данных платежных приложений

3. Требования Стандарта безопасности данных платежных приложений (PA-DSS) являются производными от PCI DSS

Who is who?

4. Стандарт **PCI DSS** обычно **не распространяется** на производителей платежных приложений, поскольку большинство производителей не выполняют хранение, обработку и передачу данных платежных карт.

5. Поскольку **клиенты** должны **соблюдать** требования **PCI DSS**, то платежные **приложения** должны ему **соответствовать** и не нарушать

Who is who?

6. Стандарт **PA-DSS распространяется на производителей** программного обеспечения и другие организации, разрабатывающие платежные приложения

7. PA-DSS содержит 14 пунктов требований и множество подпунктов

8. п.5. **Необходимо разрабатывать защищенные** платежные приложения.

Who is who?

1. Стандарт **PA-DSS распространяется на производителей** программного обеспечения и другие организации, разрабатывающие платежные приложения

2. PA-DSS содержит 14 пунктов требований и множество подпунктов

3. п.5. **Необходимо разрабатывать защищенные** платежные приложения.

PA-DSS п.5

5.1.1) До внедрения в среду эксплуатации должно выполняться тестирование всех обновлений безопасности, а также изменений в конфигурации систем и программного обеспечения

5.1.1.1) Проверка всех данных, вводимых в систему (для предотвращения межсайтового скриптинга (cross-site scripting), инъекций, выполнения вредоносного кода и т. д.)

5.1.1.2) Проверка правильности обработки ошибок

….

5.1.1.5) Проверка того, что управление доступом на основе ролей выполняется правильно

….

PA-DSS п.5

5.1.7) Для идентификации потенциальных уязвимостей в разработанном программном обеспечении после внесения всех изменений должен выполняться анализ кода перед его передачей клиенту

5.1.7.a) Производитель проводит анализ кода при любом его изменении для внутренних приложений

5.1.7.b) Производитель проводит анализ кода при любом его изменении для веб-приложений

…

PA-DSS п.5

Анализ кода **не проводит его автор** и лицо, проводящее анализ, **обладает достаточными знаниями** техник пересмотра кода и безопасного программирования…

web-приложения в рамках PA-DSS

PA-DSS рекомендует следовать OWASP для веб-приложений

OWASP - Open Web Application Security Project

Рейтинг OWASP Top Ten 2010 RC1 содержит десять наиболее распространенных уязвимостей веб-приложений

OWASP Top Ten 2010

1. Injection (SQL,Shell,...)
2. Cross Site Scripting (XSS)
3. Broken Authentication and Session Management
4. Insecure Direct Object References
5. Cross Site Request Forgery (CSRF)
6. Security Misconfiguration
7. Failure to Restrict URL Access
8. Unvalidated Redirects and Forwards
9. Insecure Cryptographic Storage
10. Insufficient Transport Layer Protection

PA-DSS и !web-приложения

1. Для остальных приложений возможно применять CWE/SANS Top 25 Most Dangerous Programming Errors

2. CWE/SANS Top25 применим для любых приложений

    1. Cross site Scripting (XSS)
    2. SQL Injection
    3. Classic Buffer overflow
    4. Cross Site Request Forgery (CSRF)
    5. Improper Access Control

Выполнение анализа кода

Анализ исходного кода вручную

инструменты:

1. Редактор с подсветкой синтаксиса
2. /dev/hands
3. /dev/brain

Выполнение анализа кода

Или с использованием автоматических средств:

1. Статический анализатор кода
    1. Pixy
    2. CppCheck
    3. Rats
    4. Вменяемый web scanner
    5. …
    6. ?

Поскольку полностью автоматизировать невозможно, то комплекте обязательно идут:

1. /dev/hands
2. /dev/brain

Выполнение анализа кода

1. Не все типы ошибок возможно найти автоматически

2. Автоматизированные инструменты дают ложные срабатывания

3. Волшебной кнопки «найти все баги в этом коде» не ожидается. =(

Фаззинг

Fuzzing - процесс генерации заведомо некорректных входных данных для исследуемого продукта в надежде вызвать ошибку или сбой.

«Трехлетний ребенок путем нажатия случайных клавиш обнаружил уязвимость в Mac OS X.»

Фаззинг

Фаззинг не требует от исследователя глубокого знания и понимание всех концепций программирования и функций примененных в продукте.

Но существует проблема с реализацией алгоритма фаззинга протокола\формата

Фаззинг

Некоторые фреймворки:

1. ProxyFuzzer
2. PeachFuzzer
3. SPIKE

Если приложение использует собственный протокол, то придется:

- допилить фреймворк.
- Либо реализовать с нуля.

Фаззинг

Несмотря на то, что фаззинг применяется в основном для анализа blackbox,

в случае whitebox мы можем более точно настроить алгоритмы, модели и шаблоны фаззинга,

и в последствии обнаружить ошибки, пропущенные при анализе другими методами.

конференция
РусКрипто'2010

ВОПРОСЫ ?

Сапожников Антон

- (495) 980 23 45
- a.sapozhnikov@infosec.ru

Информзащита
Системный интегратор