# КРПС

## Безопасность

Лекция №11

(версия 1.0)

# Вирусы

| Год | События, цифры, факты |
|---|---|
| 21.11. 1988 | Вирус Морриса на 24 часа вывел из строя сеть ARPANET. Ущерб составил 98 млн. долларов. |
| 1988 | Зафиксировано 200 попыток заражения вирусами (150 - успешных) глобальной компьютерной сети NASA. Причем 16 мая в течение 7 часов было заражено 70 ЭВМ. |
| с 20.03. по 9.09. 1988 | Промышленная ассоциация компьютерных вирусов (Computer Vi-rus Industry Association - CVIA) зарегистрировала 61795 случаев заражения вирусами различных информационных систем по всему миру. |
| 1986 - 1989 | Зарегистрировано 450 случаев попыток НСД и заражения вирусами (220 - успешные) сети МО США DDN. Длительность цикла проникновения и выборки информации не превышала 1 мин. |
| 1992 | В США было заражено чуть более одного из каждых десяти офисных компьютеров (данные для более, чем 60000 ПЭВМ фирм Mac, Atari, Amiga, PC) |
| 1994 | Национальная аудиторская служба Великобритании (National Audit Office - NAO) зарегистрировала 562 случая заражения вирусами компьютерных систем британских правительственных организаций, что в 3.5 раза превышает уровень 1993 г. |

# Хакеры

Уязвимость в мобильном приложении WhatsApp позволяет получать полный доступ к переписке пользователя злоумышленникам из сторонних приложений, установленных на мобильное устройство. Инструкцию по взлому написал и опубликовал в своем блоге независимый специалист по информационной безопасности и главный технический директор DoubleThink Бас Босхерт (Bas Bosschert).

Проблема касается только устройств на базе Android, в которых WhatsApp сохраняет резервную копию истории на карте памяти. Большинство приложений, устанавливаемых на устройство, имеют доступ к карте памяти. Таким образом, они могут легко получить доступ к файлу базы данных WhatsApp.

Для того чтобы расшифровать файл, Босхерт написал несложный скрипт на языке Python, текст которого опубликовал в своем блоге. Ключ шифрования автор взял из приложения WhatsApp Xtract, которое позволяет просматривать и хранить историю переписки из WhatsApp на персональном компьютере. WhatsApp использует один и тот же ключ для шифрования сообщений всех пользователей, отметил эксперт.

Статья 159.6 УК РФ, Мошенничество в сфере компьютерной информации устанавливает максимальное наказание (в случае, если хищение совершено организованной группой или в особо крупном размере) в виде 10 лет лишения свободы.

# Программисты

## 1-Й ХАКЕР СССР ОСТАНОВИЛ КОНВЕЙЕР ВАЗА И ОСТАЛСЯ НА СВОБОДЕ

23.12.2011 / 00:30

**В 1983 году в СССР было совершенно первое в истории преступление в сфере высоких технологий – хакнули ПО на АВТОВАЗе, в результате чего конвейер встал на три дня. Возник прецедент: совершено преступление, за которое не предусмотрено наказание.**

# Защищенная разработка

Защищенная разработка – это комплекс мер, добавляющихся к обычному процессу создания программного обеспечения, направленных на обеспечения высокого уровня безопасности информации, обрабатываемой разрабатываемым программным обеспечением.

Существует множество методик защищенной разработки, самые известные из них созданы крупнейшими разработчиками программного обеспечения и различными общественными и государственными организациями. Методики защищенной разработки описывают необходимые действия на всех шагах жизненного цикла программного обеспечения.

# Microsoft Security Development Lifecycle (SDL)

| Этап разработки | Действие |
|---|---|
| Обучение | Обучение основам безопасности |
| Требования | Задание требований безопасности |
| | Создание контрольных условий качества и панелей ошибок |
| | Оценка рисков безопасности и конфиденциальности |
| Проектирование | Задание требований проектирования |
| | Анализ возможных направлений для злоумышленных атак |
| | Моделирование рисков |
| Реализация | Применение утвержденных инструментов |
| | Отказ от небезопасных функций |
| | Статический анализ |

# Microsoft Security Development Lifecycle (SDL)

| | |
|---|---|
| Проверка | Динамический анализ |
| | Нечеткое тестирование (фаззинг) |
| | Проверка возможных направлений для злоумышленных атак |
| Выпуск | Планирование реагирования на инциденты |
| | Окончательная проверка безопасности |
| | Сертификация и архивация выпуска |
| Реагирование | Выполнение плана реагирования на инциденты |

# Build Security In

Архитектура и проектирование

- Оценка рисков при создании архитектуры ПО;
- Рекомендации при проектировании;
- Принципы проектирования защищенного ПО;
- Описание утилит для моделирования.

Разработка требований

- Создание требований к защищенному ПО;
- Шаблоны возможных атак на ПО.

Разработка

- Анализ исходных текстов ПО;
- Советы и практики по разработке;
- Правила разработки.

Управление

- Управление процессом разработки защищенного ПО;
- Оценка рисков.

# Build Security In

Тестирование:

- Тестирование на безопасность;
- Тестирование по принципу «белого ящика»;
- Тестирование по шаблонам атак.

Рекомендации по системам в целом:

- Применение, интеграция и развитие;
- Внедрение и обслуживание;
- Управление инцидентами;
- Тестирование на проникновение в систему;
- Тестирование по принципу «черного ящика».

Фундаментальные основы:

- Метрики для измерений защищенности ПО;
- Обучение и поощрение сотрудников;
- Применение процессов защищенной разработки;
- Моделирование бизнес-кейсов.

# Software Assurance Maturity Model (SAMM)

| Этап разработки | Действие |
|---|---|
| Управление | Стратегия и метрики для оценки |
| | Правила и соответствия требованиям |
| | Обучение специалистов и разработка руководств |
| Разработка | Оценка угроз |
| | Разработка требований по защищенности |
| | Разработка архитектуры по защищенности |
| Проверка | Проверка архитектуры |
| | Проверка исходных текстов ПО |
| | Проверка на защищенность |
| Выпуск | Управление уязвимостями и реагирование на их обнаружение |
| | Усиление защиты с помощью всей инфраструктуры в целом |
| | Обеспечение обратной связи |

# Building Security In Maturity Model (BSIMM)

Этапы и действия аналогичны SAMM за некоторым исключением в описании процессов. BSIMM имеет те же плюсы и минусы, что и SAMM.

# Security Considerations in the System Development Life Cycle (SDLC)

Инициирование;

Разработка;

Внедрение;

Обслуживание;

Вывод из эксплуатации.

# Cisco Security Development Lifecycle (CSDL)

| Этап разработки | Действие |
|---|---|
| Концепция | Задание требований по безопасности |
| | Задание требований по процессам |
| | Задание требований по функциональности |
| Планирование | Задание требований по проектированию |
| | Моделирование рисков |
| Разработка | Использование наиболее безопасных библиотек |
| | Статический анализ |
| | Реализация требований по безопасности |
| | Проверка используемого стороннего ПО |
| Проверка | Тестирование на уязвимости |
| | Нечеткое тестирование (фаззинг) |
| | Проверка выполнений требований по безопасности |
| Выпуск | Проверка соответствия требованиям CSDL |
| Реагирование | Реагирование на инциденты |
| | Мониторинг уязвимостей в использованном стороннем ПО |

# Рекомендации

- Защита от пользователей
- Защита от стороннего ПО
- Защита от разработчиков

# Источники


- Защищенная разработка программного обеспечения ( http://daily.sec.ru/publication.cfm?pid=42547)
- Казарин О.В. Безопасность программного обеспечения компьютерных систем. ( http://citforum.ru/security/articles/kazarin/)
- 1-й хакер СССР остановил конвейер ВАЗа и остался на свободе ( http://ponedelnik.info/society/1-y-khaker-sssr-ostanovil-konveyer-vaza-i-ostalsya-na-svobode )