Тестирование безопасности

Введение

- Очень часто современные программные продукты разрабатываются в сжатые сроки и при ограниченных бюджетах проектов.
- Разработчики зачастую игнорируют необходимость обеспечения защищённости своих продуктов, подвергая тем самым пользователей неоправданному риску.
- например, системы Интернет-платежей для виртуальных магазинов, как правило, разрабатываются для каждого магазина отдельно, практически «с нуля», используя технические средства без учёта степени их защищённости. В результате не удивительным становится такое количество сообщений о различных видах Интернет-мошенничества. Разработкой подобных критичных систем занимаются программисты, возможно прекрасно знающие языки и методы программирования, но не имеющие достаточных как практических, так и теоретических знаний в области информационной безопасности. У программистов нет времени для мониторинга новых уязвимостей используемых технологий, — у них есть задачи, и есть сроки для их выполнения.

Зачем разработчику беспокоиться по поводу защищённости и безопасности своих программ?

- Во-первых, существует ряд категорий информации и сведений, защита которых требуется национальными нормативно-правовыми актами страны, где ПП разрабатывается или используется. Практически каждая страна имеет такие категории защищаемой информации (тайны), как: государственная, банковская, личная, коммерческая и т.д.
- Во-вторых, сегодняшний пользователь весьма образован и имеет высокие требования при выборе ПП, которые он использует в своей работе. В случае наличия выбора между решениями нескольких разработчиков, пользователь, несомненно, учтёт также и степень безопасности и защищённости того или иного ПП.
- В-третьих, защищённый и безопасный ПП необходим самому разработчику. Защищённый ПП защищает интересы не только пользователя, но и разработчика (право продажи и распространения ПП и т.д.). Так, например, если ПП представляет собой мультимедийную электронную энциклопедию, то защита содержимого этой энциклопедии позволяет разработчику иметь уверенность, что в ближайшее время на рынке не появится электронных энциклопедий сторонних фирм, дублирующих содержание его собственной разработки. Аналогично, функции защиты от копирования ПП позволяют разработчику снизить величину ущерба, который приносят ему «пиратские» копии ПП.

Определения тестирования безопасности

- Существует много определений тестирования безопасности. Рассмотрим основные:
- 1. Тестирование программного обеспечения с целью обнаружения уязвимостей в области безопасности.
- 2. Процесс определения того факта, что особенности системы, отвечающие за её безопасность, реализованы согласно тому, как они были спроектированы, и что они являются адекватными в контексте окружения, в котором приложение выполняется. Этот процесс включает ручное функциональное тестирование, тестирование вторжений, верификацию.

Терминология тестирования безопасности

- ***Программный продукт*** – множество компьютерных программ, процедур вместе с соответствующей документацией и данными
- ***Программа*** – данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определенного алгоритма
- ***Защищённость программного продукта*** – способность противодействовать несанкционированному вмешательству в нормальный процесс его функционирования а также попыткам хищения, незаконной модификации, использования, копирования или разрушения самого продукта, его составляющих, данных и информации, входящих в состав продукта, доступных ему в процессе выполнения или заложенных в него во время разработки.

Терминология тестирования безопасности

- ***Безопасность программного продукта*** – способность продукта достигать приемлемого уровня риска для здоровья и наследственности людей, их бизнеса, компьютерных систем (в том числе, находящимся в них данным и программного обеспечения), имущества или окружающей среды в отсутствие нарушений законов и норм права при данном способе (контексте) применения.

Терминология тестирования безопасности

- В первую очередь, говоря об информационной безопасности, речь ведут о защите информации.
- ***Информация*** – сведения о лицах, предметах, фактах, событиях, явлениях и процессах независимо от формы их представления
- А вот научное определение того, что такое информация
- ***Информация*** – это фундаментальный генерализационноединый безначально-бесконечный законопроцесс резонансно-сотового, частотно-квантового и волнового отношения, взаимодействия, взаимопревращения и взаимосохранения (в пространстве и времени) энергии, движения, массы и антимассы на основе материализации и дематериализации в микро- и макро-структурах Вселенной.

Источники проблем информационной безопасности

- 1. Конфликт интересов между разработчиком продукта и его пользователями. Интересы разработчика – максимизация прибыли от продажи разработанного продукта, минимизация усилий при разработке продукта, обеспечение гарантий своего бизнеса. Интересы пользователей – минимизация расходов и максимизация эффекта от использования продукта, а также обеспечение гарантий своего бизнеса. В результате, в области эксплуатации продукта появляются различного рода угрозы, как со стороны разработчика, так и со стороны пользователя.
- Со стороны разработчика: неумышленное внесение ошибок при разработке продукта, ведущих к нарушению свойств безопасности данных пользователей; умышленное внесение программных «закладок», способных нанести ущерб пользователю (мониторинг предпочтений пользователя, сбор информации личного характера и т.п.)
- Со стороны пользователя: нелегальное использование продукта (эксплуатация, копирование, тиражирование, распространение); взлом продукта и его составляющих – кража интеллектуальной собственности разработчика, заложенной в продукт при его разработке.

Источники проблем информационной безопасности

- 2. Конфликт интересов между пользователями продукта, а также между пользователями продукта и пользователями компьютерной системы, в которой эксплуатируется продукт. В данном случае требуется, чтобы в продукте и в компьютерной системе были реализованы функции идентификации и авторизации пользователей, а также функции разграничения доступа к данным. Сам же продукт не должен подвергать риску как саму компьютерную систему, так и её пользователей (т.е. использование продукта не должно снижать общий уровень информационной безопасности компьютерной системы).

Источники проблем информационной безопасности

- 3. Форс-мажорные обстоятельства (перебои с электропитанием, ошибки в работе аппаратного обеспечения, ошибки в реализации используемых технологий и т.д.). В данном случае проблема обеспечения безопасности и защищённости очень тесно пересекается с проблемой обеспечения надёжности и восстанавливаемости продукта, его данных и компонентов. При этом следует отметить, что разработчик прикладного продукта при выборе технологий, при помощи которых он собирается разрабатывать свой продукт, сам становится потребителем информационных технологий. Т.е. проблема обеспечения безопасности и защищённости информационных технологий в данном случае беспокоит его как пользователя этих технологий. Т.е. на разработчика прикладного продукта также могут распространяться такие угрозы со стороны разработчиков информационных технологий, как умышленное и неумышленное внесение «закладок»/ошибок, а со стороны разработчика прикладного продукта – нелегальное использование и кража интеллектуальной собственности разработчиков ИТ.

Принципы безопасности и защищённости

- Основными принципами безопасности и защищённости программных продуктов являются:
- ***Конфиденциальность***. Предотвращение несанкционированного доступа к информации или программе.
- ***Целостность***. Предотвращение повреждения, искажения или изменения информации или программы.
- ***Проверка подлинности (аутентификация)***. Удостоверение субъектов (пользователей или процессов) в сети перед предоставлением доступа к данным, а также удостоверение объектов перед их использованием.

Принципы безопасности и защищённости

- ***Проверка полномочий (авторизация).*** Обеспечение доступа к данным только тем субъектам, которые были надлежащим образом авторизованы и имеют соответствующие права (на чтение, запись, изменение и т. д.).
- ***Доступность.*** Обеспечение необходимой работоспособности и доступности данных и программ.
- ***Подотчётность*** (в некоторых источниках — доказательство причастности, контроль). Установление связи между пользователем (или объектом) и действием (например, в случае электронных платёжных систем – подтверждение факта, что отправитель послал, а адресат получил некоторую сумму денег, а также опровержение этого факта, если транзакция, по каким-то причинам не состоялась).

Категории объектов защиты

- В качестве основных категорий объектов защиты программного продукта выделяют:
  - данные;
  - информацию;
  - функции ПП.

Категории объектов защиты

- Под данными подразумевается информация, представленная в виде файлов программы (файлы данных, исполняемые файлы самой программы), а также обрабатываемая информация, полученная от пользователей, из переменных операционной системы (реестра и т.д.), по сети или в результате взаимодействия между потоками и процессами операционных систем.
- Под информацией в качестве объекта защиты понимаются различного рода сведения (передаваемые, хранимые и обрабатываемые в электронном виде), доступные продукту и имеющие определённую ценность, факт нарушения конфиденциальности, целостности или доступности которых может вести к какому либо виду ущерба (недополученной прибыли), либо ущемлению прав пользователей.

Категории объектов защиты

- Доступ к функциям программного продукта также должен быть ограничен таким образом, чтобы исключить возможные варианты ущербов, что решается путём реализации функций авторизации и предоставления прав доступа. Задача обеспечения целостности и подлинности функций продукта тесно перекликается с задачей обеспечения безопасности данных продукта, т.к. носителями функций продукта являются исполняемые файлы и библиотеки. Так, путём внесения изменений в исполняемые модули продукта, злоумышленник может изменить логику работы его функций, заставить их выполнять то, что ему требуется. Следует отметить, что важными атрибутами защищённого и безопасного продукта являются: категории обрабатываемой информации (её стоимость, важность) и критичность функций продукта для нужд и задач бизнеса пользователя. Соответственно, в зависимости от требований к продукту и уточняются задачи защиты информации.

Уровни информационной безопасности

- Уровни представления программных продуктов с позиции информационной безопасности таковы:
  Набор данных (как это диктует определение ГОСТ, т.е. рассмотрение продукта как набора файлов, из которых он состоит).
- **Множество потоков данных**, когда продукт выполняется (обычно такие потоки описываются в виде data-flow диаграмм на этапе разработки ПП). Необходимо следить, чтобы ни что не препятствовало нормальному ходу потоков, а также, чтобы неавторизованные субъекты или процессы не смогли читать или модифицировать (удалять) данные из потоков, а авторизованным пользователям и процессам не было отказано в доступе к ним (требование также относится и к данным, которые находятся в процессе пересылки).

Уровни информационной безопасности

- **Набор функций** (сервисов). Необходимо обеспечить разграничение доступа к функциям продукта, а также обеспечить меры по сохранению доступности, правильности и надёжности работы этих функций.
- **Предмет договорных отношений между разработчиком и пользователем** (разработчик может внести в продукт некоторые функции, которые отслеживают выполнение условий договора со стороны пользователей, а пользователи могут пытаться обойти или отключить эти функции).
- **Часть компьютерной системы**, в которой выполняется продукт. Продукт должен быть безопасным для самой системы и её пользователей и, в то же время, продукт и его данные должны быть защищёны от неавторизованных процессов и пользователей в рамках данной системы.

Поиск уязвимостей

- При словах «поиск уязвимостей в программе» большинство представляет себе гения-одиночку, выполняющего загадочные тесты над беспомощной программой. Это не так. Поиск уязвимостей проводится гораздо более методично.
  Это необходимо потому, что жизненный цикл разработки безопасности (Security Development Lifecycle – SDL) и его изначальная фокусировка на безопасной разработке и развёртывании уменьшает число скрытых дефектов, делая задачу поиска уязвимостей в ходе тестирования гораздо более сложной.
  Тестирование безопасности программ слишком важно, чтобы предоставить его маленькой группе виртуозов. Оно должно быть методичным, повторяемым и усваиваемым, чтобы его можно было применять в широком спектре ситуаций.

Типы тестов уязвимости

- Суть тестирования состоит в изменении – поиске того в программе и ее среде, что может меняться, внесении изменений и наблюдении за реакцией программы. Цель состоит в обеспечении надежной и безопасной работы программы в разумных и даже не слишком разумных рабочих сценариях. Таким образом, наиболее фундаментальным принципом планирования, которое может выполнить тестер, является понимание того, что может быть изменено и какими способами это изменение необходимо готовить к тестированию.

Типы тестов уязвимости

- С точки зрения безопасности, среда, ввод пользователя и внутренние данные с логикой являются основными местами, где такие изменения могут открыть проблемы с безопасностью. Среда состоит из файлов, приложений, ресурсов системы и других локальных, либо сетевых ресурсов, используемых приложением. Любой из них может послужить точкой входа атаки.

Типы тестов уязвимости

- Ввод пользователя – это данные, происходящие от внешних (обычно недоверенных) сущностей, которые анализируются и используются программой. Внутренние данные и логика – это внутренне хранящиеся переменные и логические пути, имеющие неограниченное число потенциальных нумераций.
- Варьируя информацию в среде программы, домене ввода и данных/логических путях можно выполнять атаки. Я подробно раскрою каждую из этих трех категорий.

Атаки через среду

- Программы не работают в изоляции. Они полагаются на множества двоичных файлов и эквивалентных коду модулей, таких как сценарии и подключаемые модули. Они также могут использовать информации о конфигурации из реестра файловой системы, а также базы данных и службы, могущие находиться где угодно. Каждое из этих интеграций сред может быть источником серьезной уязвимости безопасности

Атаки через среду

- Также имеется ряд важных вопросов, которые необходимо задать о степени доверия приложения этим взаимодействиям, включая следующие: Насколько приложение доверяет своей локальной среде и удаленным ресурсам? Помещает ли приложение конфиденциальную информацию в ресурс (скажем, реестр), который могут читать другие приложения? Доверяет ли оно каждому загружаемому файлу или библиотеке, не проверяя их содержимое? Может ли злоумышленник воспользоваться этим доверием, чтобы заставить приложение выполнить свой приказ?

Атаки через среду

- Вдобавок к вопросам доверия, тестеры уязвимостей должны высматривать DLL, которые могут оказаться неисправными, либо были заменены (или изменены) злоумышленником, двоичные файлы, или файлы, с которыми взаимодействует приложение и которые не полностью защищены списками контроля доступа (access control list – ACL), либо иначе открыты для атаки. Тестеры также могут высматривать приложения, получающие доступ к общим ресурсам памяти или хранящие конфиденциальную информацию в реестре, либо во временных файлах. Наконец, тестеры должны оценить факторы, могущие оказать давление на систему, такие как медленная сеть, недостаток памяти, итп и оценить воздействие этих факторов на безопасность.

Атаки через среду

- Атаки через среду чаще всего выполняются путем подготовки небезопасной среды и затем выполнения приложения в этой среде, чтобы увидеть, как оно ответит. Это непрямая форма тестирования – атаки проводятся против среды, в которой работает приложение. Теперь давайте взглянем на прямое тестирование.

Атаки через ввод

- При тестировании уязвимости наиболее важно подмножество вводов, происходящее из недоверенных источников. Они включают пути сообщений, такие как сетевые протоколы и разъемы, открытые внешние функции, такие как DCOM, удаленные вызовы процедур (RPC) и веб-служб, файлы данных (двоичные или текстовые), временные файлы, созданные в ходе исполнения и файлы управления, такие как сценарии и XML – все они подвержены злонамеренным изменениям. Наконец, элементы управления интерфейса пользователя, позволяющие прямой ввод пользователя, включая экраны входа в систему, пользовательские веб-интерфейсы и прочее, также должны проверяться.

Атаки через ввод

- Конкретнее говоря, необходимо определить, контролируется ли ввод должным образом: допускается ли верный ввод и блокируется ли неверный (такой как долгие строки, деформированные пакеты, итп)? Адекватная проверка ввода и анализ файлов крайне важны.
- Чтобы увидеть, может ли опасный ввод попасть в элементы управления интерфейса пользователя и обнаружить, что происходит в таком случае, необходимо тестирование. Это включает специальные символы, закодированный ввод, фрагменты кода, строки формата, escape-последовательности, итп. Необходимо определить, смогут ли преодолеть защиту ли длинные строки, вставленные в пакеты, поля или файлы и способные вызвать переполнение памяти. Испорченные пакеты в потоках протоколов также являются проблемой. Необходимо следить за сбоями и зависаниями и проверять стек на предмет эксплуатируемых сбоев памяти. Наконец, необходимо обеспечить то, что проверки и сообщения об ошибках будут случаться в нужном месте (на стороне клиента, а не сервера), в качестве защиты от плохого ввода.

Атаки через ввод

- Атаки через ввод напоминают метание гранат в приложение. Некоторые из них будут должным образом отбиты, некоторые приведут к взрыву программы. Задачей группы тестирования уязвимости является определение, которые из них которые и указание на необходимость исправлений.

Атаки через данные и логику

- Некоторые опасности происходят из внутреннего механизма хранения данных и логики алгоритмов приложения. В таких случаях очевидно наличие ошибок в дизайне и программном коде, где разработчик либо ориентировался на благонамеренного пользователя, либо не учел некоторых путей кода, по которым может пройти пользователь.
- Отказ в обслуживании является основным примером из этой категории, но определенно не самым опасным. Атаки типа «отказ в обслуживании» могут быть успешны, когда разработчики забыли подготовиться к большому числу пользователей (или подключений, файлов, в общем того ввода, который может подвести какой-либо ресурс к пределу его возможностей).

Атаки через данные и логику

- Однако, существуют куда более коварные логические дефекты, которые необходимо искать тестированием. Например, входящие данные, управляющие сообщениями об ошибках и другими создаваемыми выходящими данными, могут открыть злоумышленнику информацию, которой тот может воспользоваться. Один из реальных примеров таких данных, которые всегда следует удалять – это любые жестко закодированные тестовые учетные записи или тестовые API-интерфейсы (которые часто входят во внутренние сборки, чтобы помочь в автоматизации тестов). Они могут предоставить взломщику легкий доступ. Следует выполнить еще два теста – ввести фальшивые учетные данные, чтобы определить, являются ли надежными внутренние механизмы проверки подлинности и выбрать входящие данные, с варьирующимися путями кода. Часто бывает, что один путь кода безопасен, но к тем же функциям возможен доступ другим путем, с непреднамеренным обходом какой-либо важной проверки.

Программное обеспечение для автоматизации тестирования безопасности

- Перед вами перечень лишь категорий программного обеспечения, используемого специалистами в области безопасности для тестирования компьютерных систем:
- **Сетевые утилиты:** SSH клиенты, инвентаризация сети, программы для поиска и проверки прокси-серверов, подсчет трафика, удаленное управление, сканеры портов, мониторинг сети, обнаружение и анализ Wi-Fi сетей, взлом беспроводного шифрования, HTTP снифферы, анализатора протоколов, парольные снифферы, снифферы беспроводных сетей, пакетные снифферы.

Программное обеспечение для автоматизации тестирования безопасности

- **Системы обнаружения вторжения:** защита Windows системы, защита Linux систем, сканеры уязвимостей, уклонение от атак, подбор и восстановление паролей, активное противодействие атакам.
- **Управление доступом:** сетевые экраны, системы контроля FS, эксплоиты и шеллкоды, анализ активности.

STORMNET

Мифы безопасности

- **МИФ ПЕРВЫЙ**: «защита информации – это только лишь криптография». Этот миф, видимо, связан с тем, что с самого начала своего развития системы информационной безопасности разрабатывались для военных ведомств. Разглашение такой информации могло привести к огромным жертвам, в том числе и человеческим. Поэтому конфиденциальности (т. е. неразглашению информации) в первых системах безопасности уделялось особое внимание.
  Очевидно, что надёжно защитить сообщения и данные от подглядывания и перехвата может только полное их шифрование. Видимо из-за этого начальный этап развития компьютерной безопасности прочно связан с криптошифрами.
  Однако сегодня информация имеет уже не столь «убойную» силу, и задача сохранения её в секрете потеряла былую актуальность.
  Сейчас главные условия безопасности информации – её доступность и целостность. Другими словами, пользователь может в любое время затребовать необходимый ему сервис, а система безопасности должна гарантировать его правильную работу. Любой файл или ресурс системы должен быть доступен в любое время (при соблюдении прав доступа). Если какой-то ресурс недоступен, то он бесполезен.

Миф первый

- Другая задача защиты – обеспечение неизменности информации во время её хранения или передачи. Это так называемое условие целостности. Таким образом, конфиденциальность информации, обеспечиваемая криптографией, не является главным требованием при проектировании защищённых систем. Выполнение процедур криптокодирования и декодирования может замедлить передачу данных и уменьшить их доступность, так как пользователь будет слишком долго ждать свои «надёжно защищённые» данные, а это недопустимо в некоторых современных компьютерных системах. Поэтому система безопасности должна в первую очередь гарантировать доступность и целостность информации, а затем уже (если необходимо) её конфиденциальность.
  Принцип современной защиты информации можно выразить так поиск оптимального соотношения между доступностью и безопасностью.

Миф второй

- **МИФ ВТОРОЙ**: «во всём виноваты хакеры». Этот миф поддерживают средства массовой информации, которые со всеми ужасающими подробностями описывают различные взломы. Однако редко упоминается то, что хакеры чаще всего используют некомпетентность и халатность обслуживающего персонала.
  Хакер – диагност. Именно некомпетентность пользователей можно считать главной угрозой безопасности. Также серьёзную угрозу представляют служащие, которые чем-либо недовольны, например, заработной платой.

STORMNET

Миф второй

- Одна из проблем подобного рода – слабые пароли. Пользователи для лучшего запоминания выбирают легко угадываемые пароли. Причём проконтролировать сложность пароля невозможно. Другая проблема – пренебрежение требованиями безопасности. Обычно пользователь сам «приглашает» в систему вирусы и троянских коней. Кроме того много неприятностей может принести любая неправильно набранная команда. Таким образом, лучшая защита от нападения – его недопущение. Обучение пользователей правилам сетевой безопасности может предотвратить нападения. Другими словами, защита информации включает в себя кроме технических мер ещё и обучение или правильный подбор обслуживающего персонала.

Миф третий

- **МИФ ТРЕТИЙ**: «абсолютная защита». Абсолютной защиты быть не может. Распространено такое мнение – «установил защиту и можно ни о чём не беспокоиться».
  Полностью защищённый компьютер – это тот, который стоит под замком в бронированной комнате в сейфе, не подключен ни к какой сети (даже электрической) и выключен. Такой компьютер имеет абсолютную защиту, однако использовать его нельзя. В этом примере не выполняется требование доступности информации. «Абсолютности» защиты мешает не только необходимость пользоваться защищаемыми данными, но и усложнение защищаемых систем.

Миф третий

- Использование постоянных, не развивающихся механизмов защиты опасно, и для этого есть несколько причин. Одна из них – развитие сети. Ведь защитные свойства электронных систем безопасности во многом зависят от конфигурации сети и используемых в ней программ. Даже если не менять топологию сети, всё равно придётся когда-нибудь использовать новые версии ранее установленных продуктов. Однако может случиться так, что новые возможности этого продукта пробьют брешь в защите.
  Кроме того, нельзя забывать о развитии и совершенствовании средств нападения. Техника так быстро меняется, что трудно определить, какое новое устройство или программное обеспечение, используемое для нападения, может обмануть вашу защиту. Компьютерная защита — это постоянная борьба с глупостью пользователей и интеллектом хакеров.