**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ»

Институт компьютерных технологий и защиты информации

**КОНСПЕКТ ЛЕКЦИЙ ПО ДИСЦИПЛИНЕ «ЗАЩИЩЕННЫЕ WEB-ТЕХНОЛОГИИ»**

Квалификация: специалист по защите информации

Форма обучения: очная

Направление подготовки/специальность: 10.05.02 Информационная безопасность телекоммуникационных систем

Направленность (профиль, специализация, магистерская программа):

Разработка защищенных телекоммуникационных систем

Особенности обеспечения информационной безопасности WEB – ресурсов.

Классификация WEB- ресурсов.

1. По доступности
2. Открытые
3. Закрытые
4. Полуоткрытые
5. По физическому расположению
6. Локальные
7. Общедоступные
8. По цели создания
9. Коммерческие
10. Не коммерческие
11. По функциональности
12. Статические
13. Динамические
14. FLASH сайты
15. По предоставляемой пользователю возможности
16. Контент сайты
17. Сайты для общения
18. Сайты для коммерции
19. Онлайн – сервисы
20. По предоставляемой информации
21. Интернет портал
22. Информационные ресурсы
23. Интернет представительство
24. WEB – сервисы
25. Сайты для заработка

Информационные ресурсы делятся на тематические сайты и тематические порталы

Интернет представители бывают:

1. Сайт визитка
2. Промо сайт
3. Корпоративный сайт
4. Представительский сайт
5. Каталог продукции
6. Интернет магазин

WEB – сервисы:

1. Поисковики
2. Файлообменники
3. Почтовые сервисы
4. Доски объявлений
5. Блоги
6. WEB-сервисы
7. Социальные сети
8. Фото-хостинги
9. Видео-хостинги
10. Каталоги сайтов
11. Новостные порталы

Сетевые угрозы

WEB Application security consortium выделил несколько базовых классов, каждый из которых содержит несколько групп распространенных уязвимостей, использование которых может нанести ущерб компании.

1. Словарные пароли пользователей (79%)
2. Внедрение операторов SQL (63%)
3. Интерфейсы управления оборудованием доступные любому пользователю (58%)
4. Межсайтовое выполнение сценария (53%)
5. Раскрытие конфигураций WEB приложений(47%)
6. Недостаточная защита от подбора учетных данных (47%)
7. Использование открытых протоколов передачи данных (47%)
8. Выход за приделы каталога (42%)
9. Раскрытие информации об идентификаторах (32%)

Классификация атак и уязвимостей

1. Аутентификация – атаки направлены на используемые WEB- приложением методы проверки идентификатора пользователя службы или приложения
2. Подбор – автоматизированный процесс проб и ошибок используемый для того, чтобы угадать имя пользователя, пароль, ключ шифрования и номер банковской программы.
3. Недостаточная идентификация – уязвимость возникает, когда WEB- сервер позволяет получать злоумышленникам доступ к конфиденциальной информации или функциям сервера без доступа аутентификации.
4. Небезопасное восстановление паролей – возникает, когда WEB- сервер позволяет злоумышленнику несанкционированно модифицировать или восстанавливать пароли других пользователей.
5. Авторизация – атаки направлены на методы, которые используются WEB-сервером для определения того, имеет ли пользователь приложения необходимые для совершения действия разрешения. Многие WEB- сайты разрешают только определенным пользователям получить доступ к некоторому содержимому или функциям приложения. Доступ другим пользователям должен быть ограничен. Используя различные техники, злоумышленник может повысить свои привилегии и получить доступ к защищенным ресурсам.
6. Предсказуемое значение идентификатора сессии – оно позволяет перехватывать сессии других пользователей.
7. Недостаточная авторизация – возникает, когда WEB-сервер позволяет санкционировано получить доступ к конфиденциальной информации, или доступ к информации должен быть ограничен.
8. Отсутствие тайм – аута сессии – в случае если для идентификатора сессии или учетных данных не предусмотрен тайм – аут или его значение слишком велико злоумышленник может воспользоваться старыми данными по – умолчанию для авторизации.
9. Фиксация сети – используя данный класс атак, злоумышленник присваивает идентификатору сессии пользователя заданные значения.
10. Атаки на клиента или пользователя WEB-сервера. Во время посещения сайта между пользователем и сервером устанавливать доверительные отношения в плане того что пользователь ожидает что сайт предоставляет ему содержимое. Эксплуатируя это доверие, злоумышленник может использовать различные методы для проведения атак на клиентов сервера.
11. Подмена содержимого. Используя эту технику, злоумышленник заставляет пользователя поверить, что страницы сгенерированы WEB-сервером, а не переданы из внешнего источника
12. Межсайтовое выполнение сценария. Наличие этой уязвимости позволяет передать серверу исполняемый код, который будет перенаправлять браузер пользователя.
13. Расщепление HTTP запроса

При использовании данной уязвимости злоумышленник посылал серверу специальным образом сформированный «запрос – ответ» который интерпретируется целью атаки как два разных ответа.

1. Выполнение кода. Атаки направлены на выполнении кода на WEB- сервере. Все сервера используют данные переданные пользователям при обработке запросов. Часто эти данные используются при составлении команд применяемых для генерации динамического содержимого. Если при разработке не учитывать требования информационной безопасности, то злоумышленник получает возможность модифицировать исполняемые команды.
2. Переполнение буфера. Эксплуатация переполнения буфера позволяет злоумышленнику изменить путь исполнения программы путем перезаписи данных в памяти системы.
3. Атаки на функции форматирующих строк. При использовании этих атак путь исполнения этих программ модифицируется методом перезаписи областей памяти с помощью функций форматирования символами
4. Внедрение операторов LDAP. Атаки этого типа направлены на WEB- серверы создающие запросы к службе LDAP на основании данных веденных пользователем.
5. Выполнение команд операционной системы. Атаки этого класса направлены на выполнение команд операционной системы на многие сервера, путём манипуляции выходными данными.
6. Внедрение операторов SQL. Эти атаки направленные на WEB- серверы создают SQL запросы с серверами СУБД на основе данных вводимых пользователем
7. Внедрение серверных сценариев. Атаки данного класса позволяют злоумышленнику передать исполняемый код, который в дальнейшем будут выполнены на WEB-сервере.
8. Разглашение информации. Атаки данного класса направлены на получение дополнительной информации в WEB-приложениях. Используя эту уязвимость, злоумышленник может определить используемые дистрибутивы программного обеспечения, номера версии клиента и сервера и установление обновлений. В других случаях в вытекающей информации может содержаться расположение временных файлов или резервных копий. Во многих случаях эти данные не требуются для работы пользователя. Большинство серверов предоставляет доступ к чрезмерному объему данных, а также следует минимизировать объем служебной информации. Чем большими значениями приложение будет располагать злоумышленник, тем легче ему будет скомпрометировать истину.
9. Идентификация приложений. Определяя версию приложения, злоумышленник использует для получения информации об использовании сервером и клиентом операционных системах, WEB серверах, и браузеров.
10. Утечка информации. Эти уязвимости возникают в ситуациях, когда сервер публикует важную информацию, например комментарии разработчика, или сообщение об ошибке, которая может быть использована для компрометации системы.
11. Обратный путь директории. Данная техника атак направлены на получение доступа к файлам директории и командам, находящимся вне основной директории веб сервера.
12. Предсказуемое расположение ресурсов позволяет злоумышленнику получить доступ к открытым данным или функциональным возможностям.
13. Логические атаки. Атаки данного класса направляются на эксплуатацию функций приложения. Логика приложения представляет собой процесс функционирования программы при выполнении определенных действий. Например, восстановление паролей, электронную коммерцию, регистрацию данных и др. Приложение может потребовать от пользователя корректное выполнение нескольких последовательных действий для выполнения определённой задачи. Злоумышленник может обойти его и использовать эти механизмы в своих целях.
14. Отказ в обслуживании. Данный класс атак направлен на нарушение доступности веб сервера.
15. Недостаточное противодействие автоматизации. Эта уязвимость возникает, в случае если сервер позволяет автоматически выполнять операции, которые должен проводить вручную пользователь.
16. Недостаточная проверка процесса. Эта уязвимость возникает в случае, когда сервер недостаточно проверяет последовательность выполнения операций приложения.

Анализ и оценка уязвимости WEB-ресурсов.

В последние годы крупные организации все активнее стали использовать разнообразные WEB-приложения, то есть одинаковые сайты компании и систему управления ресурсами предприятия. Электронные торговые площадки, порталы государственных услуг, корпоративные приложения на основе специализированных клиентских программ обеспечения всё чаще заменяют веб-серверами. Поэтому именно уязвимости в WEB-приложении становятся одним из основных векторов атак на корпоративную информацию системы. Исследуя WEB-приложения, в большей части это WEB-приложения электронной коммерции 30 %, финансы/банки 22 % , промышленности 12 %, информативные технологии 15 %, телекоммуникаций 13 %.

Уязвимости по средствам разработчика. Наиболее уязвимые приложение на PHP. 81% систем на этом языке содержит критические опасные уязвимости. Каждое WEB-приложение на PHP в среднем содержит 11 критически опасных уязвимостей.

Уязвимость на сервере. Самая распространённая ошибка администратора WEB-сервера является идентификация пользователя. В частности данная уязвимость всплывает в 8 из 10 случаев.

Уязвимости по отраслям. Лидерами по количеству системных уязвимостей и по количеству уровня риска оказались банковские системы 89 %. По среднему количеству уязвимостей на одну систему, наименее защищенные оказались WEB-сайты промышленных предприятий, где на одно приложение приходится 18 критически опасных уязвимостей.

Уязвимости WEB-приложений и обеспечение их безопасности.

В последние годы число атак нацеленных на WEB-сервисы значительно возросло. WEB - сервера особо уязвимы в силу своей открытости, поскольку рассчитаны на обмен информации с пользователями. Обеспечение безопасности операционной системы реализуется путём установки последних обновлений системы безопасности, регулярно выпускаемых разработчиками этой операционной системы. Также следует вовремя обновлять все ПО работающее на WEB-сервере. Любые программы, не относящиеся к необходимым компонентам, например, служба удаленного рабочего стола, следует отключить или удалить. Использование антивирусного программного обеспечения является обязательным требованием для любого WEB - сервера вне зависимости от операционной системы. Сочетание с гибким межсетевым экраном антивирусное программное обеспечение становится одним из самых эффективных способов защиты от угроз безопасности. Когда WEB - сервер становится целью атаки, злоумышленник без промедления постарается загрузить инструменты взлома или вредоносное программное обеспечение, чтобы успеть использовать уязвимость, до того как она будет закрыта. При отсутствии качественного антивируса такая уязвимость системы безопасности может долгое время оставаться незамеченной. Атаки на WEB - сервер можно разделить на две категории локальные и глобальные.

Локальные обычно направлены напрямую на информацию или перенаправлены на отдельный WEB - сервере. Глобальные обычно направлены на несколько WEB - сайтов и стали ставить свои цели о закрытии всех пользователей сайта.

Наиболее опасные виды сетевых атак.

1. Фишинг - вид атаки, который начинается с рассылки пользователей сообщение содержащее ссылку на известный ресурс или имитирующей.   
2. Социальная инженерия - метод несанкционированного доступа к информации или системы управления информации без использования технических средств. Метод основан на использовании человеческого фактора и считается очень разрушительно.   
3. Серфинг - одна из разновидностей фишинга. Суть атаки заключается в том, что страница с известными «URL» поменяется на страницы злоумышленника.  
4. Троянский конь-программа записывает всю информацию и передает эти данные на машину хозяина.

Атаки на WEB - серверы. Легальные WEB - серверы взломали следующие атаки , SQL инъекции, вредоносные приложения

Метод перебора с поиском сервера, через выраженный хостинг компании, через уязвимости программы. Одним из основных средств вторжения в последнее время стали атаки типа автозагрузки, то есть загрузка файлов или скриптов без ведома хозяина. Разнообразие мультимедиа форм так же облегчает проблему пользователей, так как пользователь обычно не удивляет необходимое время загрузки новейшей версии программы для просмотра материала программа на уделяемом сервере.

CSS - является одной из наиболее распространенных средств атак, преследующих цель получения персональных данных с помощью WEB - технологий. Иногда этот вид атаки называют html инъекции.

Файлы cookie - большинство браузеров поддерживает 2 метода, которые WEB - приложения могут использовать для сохранности информации данные клиентов, обычно Файлы cookie и файлы cookie сеанса. Файл cookie это небольшой файл издаваемый браузером и хранящийся на компьютере пользователя. Его содержимое не регламентируют, но обычно в таких файлах хранится название, дата окончания срока действия и некий объём данных. Файлы cookie сеанса позволяют WEB - приложениям хранящие данные в памяти. Различие заключается в том, что обычно файл cookie сохраняешься на компьютере пользователя и остаёшься на нем до момента удаления пользователем. Файл cookie сеанса храниться только на протяжении времени работы компьютера и автоматически теряется при закрытии приложения браузера. При этом оба файла подвергаются влиянию извне.

SQL-инъекция используется для атаки WEB - сайтов работающих с базами данных. Возможность внедрения SQL - кода возникает, если в SQL запросах используется неотфильтрованные данные вводимые пользователем. Многие современные WEB - сайты используют сценарии и SQL запросы для динамичного формирования содержимого страниц. В SQL - запросах чисто используется данные вводимые пользователем. Это может привести к угрозе безопасности, поскольку злоумышленники могут просто внедрить вредоносный SQL – код.

Межсайтовый скриптинг. Данный вид атаки может быть направлен на сайты отображающие пользователю данные. Вместо конкретного кода злоумышленник пытаешься атаковать код самого WEB – сайта, внедряя в него вредоносные файлы. Многие сайты хранят имена всех пользователей в базе данных, чтобы иметь возможность отобрать при вводе соответственного пользователя. Злоумышленника может создать учетную запись, разместив при этом в поле имени вредоносный код. Подобные атаки обычно появляются с помощью вредоносных скриптов. Предполагается, что в базе данных хранится имя пользователя, но на самом деле это будет вредоносный код.

Использование уязвимостей: одной из желанных целей для создания вредоносного программного обеспечения является запуск кода без каких-либо действий пользователя. Самый распространенный способ - это использование уязвимостей приложений или ОС для получения прав на выполнение. В последние годы эта тактика применяется многочисленными семействами сетевых червей заражавших сетевой компьютер, используя сетевые уязвимости. Этот метод особенно характерен для интернет атак. Использование уязвимостей в клиентском браузере предоставляет вредоносным программам возможность выполнения кода даже в том случае если пользователь просто перешел на вредоносную страницу.

Использование уязвимостей является идеальным механизмом, с помощью которого может запускаться вредоносный код. Тем не менее, социальная инженерия остается самым надежным методом и продолжает широко применяться в почтовых системах.

Использование уязвимостей: одной из желанных целей для создания вредоносного ПО является запуск кода без каких-либо действий пользователя. Самый распространенный способ - это использование уязвимостей приложений или ОС для получения прав на выполнение. В последние годы эта тактика применяется многочисленными семействами сетевых червей заражавших сетевой компьютер, используя сетевые уязвимости. Этот метод особенно характерен для интернет атак. Использование уязвимостей в клиентском браузере предоставляет вредоносным программам возможность выполнения кода даже в том случае если пользователь просто перешел на вредоносную страницу.

Использование уязвимостей является идеальным механизмом, с помощью которого может запускаться вредоносный код. Тем не менее, социальная инженерия остается самым надежным методом и продолжает широко применяться в почтовых системах.

Использование поисковых интернет систем при выявлении потенциальных целей особенно актуально в случае, когда атаке подвергается сам WEB-ресурс или отдельное WEB-приложение. Данная техника используется в целом ряде вредоносного ПО, наиболее известным из которых является вирус червь, применяющий систему Google для выявления новых жертв, использующих определенные интернет приложения. Если раньше злоумышленники использовали для поиска уязвимостей компьютеров в сети специализированные скрытные средства, то теперь подходящие для атаки мишени можно находить с помощью поисковых систем. Предположим, что в популярном приложении для ведения форума или блога найдена уязвимость. Злоумышленники могут использовать поисковые системы для выявления сайтов, на которых имеются страницы, создаваемые популярным приложением, имеющим уязвимости, тем самым формируя потенциальных жертв.

Вредоносное ПО может размещаться на разных сайтах. Хакеры могут использовать бесплатные службы WEB-хостингов, создавать новые сайты, или захватывать существующие сайты, превращая их в базу для атаки. В большинстве случаев используются скрипты. Также как обычные скрипты зачастую опрашивают клиентский браузер, чтобы обеспечить корректное отображение страницы, вредоносные скрипты могут запрашивать браузер, чтобы загрузить подходящий эксплоит. В таких атаках обычно используются перебор списка известных эксплоитов в ходе, которого, ищется тот эксплоит, от которого клиент не защищен.

Пользователи подключаются к сайтам для использования предоставляемых ими услуг, позволяя своим браузерам соответствовать им образом отображения страницы. Пользователей обычно призывают вести списки надежных сайтов, чтобы браузер можно было настроить в соответствии с уровнем надежности просматриваемого сайта. Сама по себе идея применения более строгих правил к разрешенному контенту поступает из ненадежной области, основанной на разумных принципах. Компрометация сайта, в результате которой на странице загружается вредоносный контент, служит малозаметным средством достижения своих целей.

Более того если сайт с большим количеством регулярных пользователей, которые считают его надежным, то потенциальное число жертв будет также велико.

Язык HTML представляет множество удобных способов загрузки дополнительного контента. Чаще всего на скомпрометированных сайтах используется метод с применением тега IFRAME, который позволяет незаметно загрузить на страницу дополнительный контент. Этот тег широко используется для обычной разметки на многих WEB-сайтах. Тег IFRAME поддерживает несколько атрибутов. Чаще всего в атаках используются атрибуты ширины и высоты, с помощью которых можно задавать размер фрейма на странице, куда загружается контент. Чтобы сделать компрометацию незаметной для пользователя в большинстве вредоносных тегов задействуются весьма малые значения для длины и ширины от 0 до 10 пикселей. Подобные теги приводят к созданию на странице множества мелких полей, которые могут стать весьма заметны в случае ее многократного заражения. Аналогичные результаты обычно достигаются путем компрометации WEB страниц с помощью вредоносного скрипта, который просто вписывает тег IFRAME в страницу при ее просмотре.

Теоретически после обхода системы безопасности сайта и получении удаленного доступа на скомпрометированном сервере могут размещать все компоненты необходимые для атаки. Измененные страницы могут запускать вредоносные скрипты и вредоносное ПО, размещаемые на том же сервере.

Adobe Flash позволяет посылать заголовки HTTP запросов и сценариев использующих Flash скрипт. Возможность для злоумышленников заставить браузер жертвы посылать HTTP запросы 3-ей стороне имеет серьезные последствия для безопасности. Браузеры накладывают ограничения на размер кода потенциально вредоносного, который может быть послан внутри фальсифицированного заголовка запроса. Такое ограничение снижает риск уязвимости, но Flash позволяет злоумышленникам обойти это ограничение путем упаковки фальсифицированных запросов в Flash-файл. Специальным образом подготовленный Flash-файл доставляет код атакуемому браузеру, что увеличивает риск успешной атаки WEB-серверов.

Использование сценарных возможностей одного приложения для реализации уязвимости в другом достаточно широко используется в WEB. Потенциально это может даже привести к выявлению черных ходов в PDF файлах, поскольку PDF-файлы могут содержать в себе java script сценарий.

Сигнатурный анализ

Распознавание атак по сигнатурам становится все более проблематичным.

Во-первых, сигнатур становится настолько много, что их перебор начинает поглощать более заметные ресурсы процессора.

Во-вторых, многие вредоносные коды содержат в себе механизмы активных вариаций сигнатуры. В качестве альтернативы сигнатурному распознаванию может рассматриваться регистрация аномального поведения пользователя ЭВМ, или всей локальной сети, которая может быть сопряжена с атакой или вторжением.

Меры по защите от интернет атак

Самый простой способ проверки интернет контента - это использование какого-либо приложения, проверяющего входящий http трафик. В ходе такой проверки входящий контент обычно передается антивирусной программе, что позволяет блокировать страницы, содержащие заведомо вредоносные программы и скрипты. Проверка контента на интернет шлюзе позволяет обеспечить дополнительный уровень защиты пользователя с помощью антивирусной программы. Проверка контента и фильтрация URL адресов дополняют друг друга, обеспечивая защиту при изменении источника и метода атаки. В WEB приложении также обычно применяется какая-либо форма классификации URL адресов.

При этом запросы заведомо вредоносным адресам или доменам могут блокироваться вне зависимости от того был ли обнаружен вредоносный контент.

Безусловно, это полезно с учетом того, что хакеры активно используют автоматизацию для постоянного изменения угроз с целью избежания обнаружения.

Успех блокирования запросов к заведомо вредоносным доменам зависит от своевременного обновления списка подобных сайтов. Эффективность применения такого списка определяется рядом факторов, в числе которых релевантность данных, серверная поддержка и фильтрация URL адресов. Релевантность данных это сбор необходимой информации о вредоносных интернет программах для выявления новых атак с максимальной скоростью.

В таких решениях могут использоваться автоматические интернет боты или совместная работа с партнерами, позволяющая собрать максимальный объем данных об интернет угрозах.

Серверная поддержка

Для обработки поступающих данных об URL адресах, проверке контента и оперативной публикации необходимых данных используемых соответствующими продуктами требуются сложные системы обработки и публикации. Также системы должны быть способны отслеживать угрозы и вести анализ вредоносных интернет программ в реальном времени, гарантировать выявление всех используемых в атаке файлов и блокирование всех задействованных URL адресов.

Фильтр URL адресов

Фильтр URL адресов также может использоваться для управления типами сайтов, которые пользователю разрешается посещать. Сайты, относящиеся к игровым или развлекательным, могут быть закрыты в пределах компании или локального участка.

Одним из главных и безусловных требований является своевременное обновление всех системных и прикладных программ.

Уязвимость WEB-сервера возрастает, когда он предоставляет хостинг для размещения десятков, а иногда тысяч WEB-серверов клиентов. Особую угрозу представляют взломанные машины, которые превращаются в proxy-серверы.

Т.к. некоторые софты нельзя просмотреть, если запретить выполнение Java script, Java-апплетов, .Net приложений, Flash, разрешение работы этих программ в рамках браузера на компьютере пользователя открывает двери для вторжения в случае наличия ошибок в кодах WEB-приложений, которые принимают ввод пользователей и используют cookie. Любые WEB-приложения, которые предлагают ввод текста в диалоговом режиме потенциально могут стать средством ввода вредоносного кода и последующей пересылки его другим пользователям. Многих угроз можно избежать, если сформировать шлюз WEB-сервера, который может:

1. Фильтровать URL, препятствуя заходам на скомпрометированные WEB узлы
2. Выявлять вирусы, нежелательные программы
3. Выявлять истинный тип файла, а не полагаться на расширение в имени
4. Обеспечивать работу SSL с высоким уровнем защиты
5. Блокировать доступ в случае, если вместо имени машины используется IP адрес
6. Исполнять коды, только если они получены из проверенных URL
7. Автоматически и регулярно обновлять базу данных вредоносных кодов и вирусов и т.д.
8. Блокировать работу если число соединений слишком велико, или если загружаемый WEB модуль оказался слишком велик
9. Не доверять IP адресам присылаемым пользователям

Наиболее эффективной может считаться многоуровневая защита

1. Использование анти-спам фильтра
2. URL фильтрация
3. Сканирование на предмет наличия Java Script вставок, а так же тексты откликов на WEB-запросе
4. Блокировка пересылки из серверов любых персональных данных.
5. Фаервол - комплекс программно-аппаратных средств, осуществляющих контроль и фильтрацию проходящих через него сетевых пакетов на разных уровнях модели OSI в соответствии с заданными привилегиями.

Типовая методика анализа уязвимостей в WEB приложениях

В последнее время WEB технологии стали активно использоваться для выполнения задач критичных с точки зрения обеспечения защиты информации. Для информационных систем реализующих WEB-технологии все более актуальными становятся угрозы безопасности информации, связанные с использованием для выполнения компьютерных атак уязвимостей WEB-приложений. Для эффективного построения системы защиты информации, подобных информационных систем, на ряду с защитой от несанкционированного доступа к информации, необходима реализация защиты на уровне используемого ПО, т.е. информационные системы должны быть защищены от угроз связанных с наличием уязвимостей в ПО а также в WEB-приложении.

Контроль защищенности WEB-приложений, как правило, выполняется в рамках сертификации по требованиям безопасности информации, которая предусматривает тестирование на проникновение.

При проведении анализа уязвимостей в рамках сертификационных испытаний, экспертами в настоящее время активно используются методические рекомендации приведенные в ГОСТ Р ИСО/МЭК 18045 т.е. документа содержащего общую методологию оценки используемую при проведении испытаний по линии общих критериев. Предлагаемая этим примером методика предполагает проведение идентификации множества потенциальных уязвимостей объекта и проведение тестирования на проникновение с целью проверки выдвинутых гипотез относительно наличия потенциальных уязвимостей. В частности этот документ направлен на детализацию и пояснение действий оценщика «Идентификация потенциальной уязвимости в общедоступных источниках» и «Тестирование проникновения», которые обладают необходимой полнотой в части поиска, идентификации и тестирования потенциальной уязвим.

Типовая методика анализа уязвимостей представляет собой набор сценариев, типовых атак на приложения, которые должны быть выявлены экспертом в испытательной лаборатории. Для формирования множества типовых сценариев атак на WEB приложения используется методика предлагаемая стандартом ИСО/IEC TR 2000 модифицированной с учетом особенностей отечественной системы сертификации.

На первом этапе формируется перечень потенциальных уязвимостей WEB-приложений. При выполнении сертификации потенциальных уязвимостей WEB-приложений выполняется исследование следующих источников информации:

1. Документация на различные WEB-приложения, представленная для различных WEB-приложений
2. Модель WEB -приложений построенная с использованием диаграммы информационных потоков
3. Банк данных угроз безопасности данных ФСТЕК России
4. Базы данных уязвимостей
5. Базы данных шаблонных атак

На втором этапе для каждой потенциальной уязвимости разрабатываются типовые сценарии атаки, например: межсайтовое выполнение скриптов, внедрение SQL кода и т.д. Каждый сценарий включает в себя порядок выполнения и ожидаемые результаты.

Порядок выполнения теста

1. Авторизация на АРМ с использованием данных учетной записи администратора информационной системы.
2. Запуск браузера, выполнение доступа к WEB-приложению, путем ввода в адресной строке браузера соответствующего адреса.
3. Авторизация в WEB-приложении с использованием данных учетной записи администратора WEB-приложения
4. Поиск и идентификация частей WEB-приложения, которые обеспечивают получение, обработку и последующее использование данных полученных от пользователя WEB-приложения для генерации WEB-страниц (эти части WEB-приложения могут быть использованы для проведения атаки типа межсайтовое выполнение скрипта)
5. Для каждой идентифицированной части WEB-приложения, которая потенциально может использоваться для проведения атаки типа межсайтовое выполнение скрипта, осуществляется атака типа межсайтовое выполнение скрипта путем отправки запросов WEB-приложению, содержащих наборы данных, реализующих политики выполнения сценариев в браузере.

Безопасность приложений на основе систем облачных вычислений.

Облачные вычисления получили широкое распространение в широких областях науки и техники. На этапе распространения облачных вычислений отсутствовала полная классификация угроз. Сейчас на рынке представлены различные способы защиты в облачной архитектуре. Они ориентированы на узкий спектр решения задач, к известным типам угроз добавились сложности, связанные с контролем среды – гипервизор – программа или аппаратное средство обеспечивающая или позволяющая одновременные, параллельные выполнения нескольких ОС на одном и том же хост компьютере. Другая сложность облачной инфраструктуры – трафик между гостевыми машинами и разграничение прав доступа. Основой обеспечения безопасности является строгий контроль физического доступа к серверам и сетевой инфраструктуре. В отличие от физической безопасности сетевая безопасность представляет собой представление надежной модели угроз, включающей в себя защиту от вторжений и применение межсетевых экранов с целью разграничения внутри сети центра обработки данных на подсети с разным уровнем доверия. Следует обратить особое внимание на механизмы регулирования подключения к облаку. Это может быть контроль доступа через определенные IP-адреса, обязательное применение VPN, доступ также может регулироваться с помощью специальных узлов – шлюзов безопасности, которые играют роль посредника между клиентами и облачными платформами. Классификация существующих угроз облачных вычислений с учетом особенностей виртуальной среды:

1. Динамичность виртуальных машин. Они могут быть перемещены между физическими серверами. Такая изменчивость влияет на целостность всей системы безопасности, однако уязвимости ОС или приложений в виртуальной среде распространяется бесконтрольно, и часто проявляются после произвольного промежутка времени, поэтому очень важно надежно зафиксировать состояние защиты системы, не зависимо от ее место положения.
2. Разграничение сети и защита периметра. При использовании облачных вычислений периметр сети размывается или исчезает, это приводит к тому, что защита более уязвимой части облака определяет общий уровень защищенности. Для разграничения сегментов разными уровнями доверия, виртуальные машины должны сами обеспечивать себя защитой, перемещая сетевой периметр в самой виртуальной машине.
3. Уязвимости внутри гипервизора. Сервер облачных вычислений и локальный сервер используют одни и те же ОС и приложения. Для ПО на облачных платформах очень высока угроза индивидуального взлома или заражения. Вызвать переполнение буфера и инициировать запуск произвольного кода могут определенные ошибки в гипервизоре. Ошибки могут содержаться как на стороне управления виртуальной инфраструктурой, когда их эксплуатация проводится снаружи, так и со стороны виртуальной машины.
4. Защита данных и приложений в традиционных инфраструктурах защиты данных строится на основе физической защиты доступа к программно аппаратным ресурсам. В облачной среде такой подход теряет смысл, т.к. методы защиты должны представлять собой единое целое. Доступ может получить только тот, кто обладает нужными правами в нужное время и в нужном месте. Предприятия должны обладать возможностью проверить, что ресурсам не нанесен вред и система не скомпрометирована, особенно в ситуации, когда они размещаются в разделяемой физической среде.
5. Доступ системных администраторов, к серверам и приложениям. Одна из основных характеристик облачных платформ, это самообслуживание. Каждый пользователь может получить доступ через интернет к управлению вычислительной мощностью. Критически важными становятся: строгий контроль доступа для администраторов, а так же обеспечение контроля и прозрачности изменений на системном уровне.
6. Влияние традиционной безопасности на производительность. К большинству существующих решений безопасности создавалось для внутренней инфраструктуры и проектировалось без учета работы в виртуальной среде. В облачной системе, где виртуальные машины разделяют общие программные и аппаратные ресурсы, единовременный запуск приведет к катастрофическому снижению производительности. В настоящее время облачные услуги по защите информации предлагает большое число компаний.
7. Атаки на системы управления. Большое кол-во виртуальных машин используемых в облачной инфраструктуре требует наличия систем управления способных надежно контролировать создание, перенос и удаление виртуальных машин. Появление новых угроз и методов атак, направленных непосредственно на платформу виртуализации требует внедрения ряда специализированных защитных механизмов, которую можно разделить на два класса. К первому классу относятся системы, поставляемые в виде готовых аппаратных решений или в виде виртуальных устройств. Преимущества такого подхода это быстрая скорость развертывания и ввода в эксплуатацию, использование существующих аппаратных мощностей, экономия ресурсов. Второй класс образует средства, предназначенные непосредственно для защиты виртуальных машин и контроля коммуникаций в виртуальной среде