**Вопросы безопасности протоколов TCP/IP**

|  |  |
| --- | --- |
| **Атака** | **Реализация** |
| **Прослушивание** | Осуществляется переводом интерфейса персонального компьютера в режим прослушивания, при этом возможен отбор трафика по установленным критериям (tcpdump, noop). Наиболее подвержены сети Ethernet, FDDI, радиосети, из- за сложности физического доступа несколько сложнее реализация для выделенных телефонных и коммутируемых линий. |
| **Сканирование сети** | Сканирование сети - выявление подключенных к сети компьютеров и определение работающих на них сетевых сервисов  1.1 Сканирование подключенных к сети компьютеров осуществляется с помощью программы ping посылкой ICMP-сообщений Echo по широковещательному адресу, на которое ответят все компьютеры, поддерживающие обработку данных сообщений.  1.2. Обратное сканирование реализуется посылкой RST-сегментов, ответы на несуществующие DNS-запросы и т.д. Если ответное сообщение не получено, то узел существует, подключен к сети и работает.  2.1 Выявление подключенных компьютеров позволяет определить помощью программ сканеров открытые порты, для реализации атак на уровне приложений. Например, с помощью стандартной программы telnet возможна попытка установления TCP соединения с помощью функции connect, позволяющей указать произвольный номер порта для установления соединения.  2.2 Сканирование в режиме половинного открытия.  Для этого на сканируемый порт направляется SYN-сегмент, получение ответного сегмента с битами SYN и ACK означает, что порт открыт, сегмент с битом RST –порт закрыт. В случае если попытка открытия порта осуществилась, то злоумышленник немедленно ликвидирует попытку соединения отправляя сегмент с битом RST.  2.3 Сканирование с помощью FIN сегментов. На сканируемый порт посылается пакет с установленным битом FIN (PSH,URG), получение ответного сегмента с битами RST означает, что порт закрыт, в противном случае порт открыт. |
| **Перехват данных**  1. Ложные ARP-ответы | Используется для перехвата трафика между узлами А и В, с помощью рассылки сфальсифицированных ARP-сообщений, при котором каждый из атакуемых узлов считает МАС- адрес злоумышленника адресом собеседника. Для этого:   1. С помощью программы ping злоумышленник X определяет МАС адреса узлов А и В. 2. На компьютере злоумышленника конфигурируется дополнительные логические IP интерфейсы, с именами адресов A и В, и отключенным для них протоколом ARP. Устанавливается статическая ARP-таблица и разрешается ретрансляция дейтаграмм. 3. На МАС-адреса узла А отправляется сфабрикованный ARP-ответ, в котором сообщается, что IP-адресу В соответствует МАС –адрес Х. Аналогичный ответ отправляется узлу В. В связи с тем, что ARP-протокол без сохранения состояния, то узлы А и В примут ARP-ответ, даже если они не посылали запрос. 4. Для поддержания в «неведении» узлы с интервалом 20-40 с. посылаются ARP-ответы, для обновления сфабрикованных записей в ARP- таблицах узлов А и В.   В случае если узел В является шлюзом сети, в котором находится узел А, то возможен перехват всего трафика между узлом А и Интернетом.  Злоумышленник Х может выступать в роли узла В, фабрикуя ответы от его имени, и отсылая их в А. |
| 2.Навязывание ложного маршрутизатора  *Ложное сообщение ICMP Redirect* | Навязывание узлу А адреса Х, в качестве адреса маршрутизатора. При этом данные от узла А перехватываются Х, анализируются и отправляются настоящему маршрутизатору, который пересылает их узлу В. Для этого:   1. Х должен находится в одной IP-сети с узлом А и знать адрес маршрутизатора, через который А отправляет дейтограммы В. 2. Х формируем IP-дейтограмму, в которой в качестве отправителя указан IP адрес шлюза, а получателя IP адрес узла А. В дейтограмму помещается сообщение Redirect, где в поле адрес нового маршрутизатора указан IP- адрес Х. Сформированное сообщение отправляется А. 3. Узел А получив дейтограмму, считает, что Redirect является реакцией на ранее отправленную дейтограмму узлу В. Из возвращенного заголовка и первых 64 бита узел А определяет, что необходимо сделать перенаправление для дейтограмм и вносит в свою таблицу маршрутов частный маршрут через маршрутизатор, указанный в сообщении Redirect. 4. Новый маршрут действует несколько минут, поэтому Х необходимо периодически повторять действия 1-3. |
| *Атака на протоколы маршрутизации* | Злоумышленник может попытаться ввести в заблуждение маршрутизаторы. Для этого формируются подложные сообщения протокола маршрутизации с целью переключения требуемых маршрутов на себя.   1. Злоумышленник Х расположен между двумя сетями А и В. 2. Перехватив сообщения RIP- между сетями А и В, злоумышленник отправляет сфальсифицированное сообщение узлу А, где в векторе расстояния указывает расстояние от себя до узла А. То же производится с узлом В. Тем самым заставляя маршрутизаторы изменять свою таблицу маршрутизации. 3. В случае, если расстояние от маршрутизатора А к В =1, то злоумышленник формирует маршрутизатору А сообщение от имени В с вектором =16, заставляя тем самым удалить данный маршрут из своей таблицы. Затем злоумышленник формирует новое сообщение от своего имени вектором=1, с указанием доступности сети В через X. |
| **Несанкционированный обмен данными** | Туннелирование  Злоумышленник хочет отправить данные с узла X узлу А, однако правила фильтрации запрещают отправку дейтограмм узлу А, но разрешают узлу В. Для осуществления данной атаки злоумышленник должен:   1. Для этого злоумышленником формируется дейтограмма, заголовок которой заполняется стандартным способом. Данная дейтограмма инкапсулируется в новую IP-дейтограмму, при этом в поле протокола указывается, инкапсуляция и адрес промежуточного узла B. 2. Получив данную дейтограмму узел В анализирует первый заголовок видит, что дейтограмма адресована не узлу A и отправляет ее по назначению, снабдив новым заголовком 3. В данном случае фильтрующий узел видит, что дейтограмма пришла с разрешенного узла и пропускает ее.   Атака крошечными фрагментами   1. Злоумышленник формирует датаграмму TCP-сегмента, состоящую из 2 фрагментов: в первом фрагменте датаграммы равной 8 октетам, находятся номера портов отправителя и получателя и поле Sequence Number, однако значения флагов не попадает. 2. Маршрутизатор, используя правила фильтрации проверяет первый фрагмент датаграммы, на наличие флага SYN, так как он отсутствует, то дейтограмма пропускается в сеть. 3. Злоумышленник формирует 2 фрагмент дейтограммы, помещая оставшуюся часть заголовка TCP- сегмента, с установленным флагом SYN 4. Так как первый фрагмент дейтограммы проверен, остальные фрагменты маршрутизатором проверяться не будут, и дайтограмма будет пропущена в сеть. 5. Происходит сборка дейтограммы на узле получателе и соединение с компьютером злоумышленника   Накладывающие фрагменты  1. Злоумышленник формирует датаграмму TCP-сегмента, состоящую из 2 фрагментов: в первом фрагменте равном 8 октетам находится полный TCP-заголовок, с установленным флагом ACK.  2. Маршрутизатор, используя правила фильтрации проверяет первый фрагмент датаграммы, на наличие флага SYN, так как он отсутствует, то дейтограмма пропускается в сеть.  3. Злоумышленник формирует 2 фрагмент дейтограммы, начиная с 9 октета с установленным флагом SYN  4. Так как первый фрагмент дейтограммы проверен, остальные фрагменты маршрутизатором проверяться не будут, и дайтограмма будет пропущена в сеть.  5. Происходит сборка дейтограммы на узле получателе. При этом второй фрагмент накладывается на первый. При сборке такой датаграммы в TCP-заголовке переписываются поля начиная с ACK SN в соответствии со значениями из второго фрагмента, и в итоге получается SYN-сегмент. |
| **Имперсонализация (spoofing)**  Узел злоумышленника выдает себя за другой узел | Десинхронизация  Злоумышленник X находится в одном сегменте сети с узлами A и В или на пути между ними.  Каждый из узлов А и В при установлении связи знает номера дейтограмм, которые может прислать собеседник, а также размер окна. Если дейтограмма не попадает в окно, то она уничтожается, а собеседнику отправляется сообщение с номером ожидаемой дейтограммы. Для реализации атаки:   1. Злоумышленник сбивает показания счетчиков узлов А и В. 2. Если узел В отправляет узлу А дейтограмму с номером SN , не попадающем в окно А ,то узел А уничтожает дейтограмму и отправляет сообщение с указанием ожидаемого SN. Но данный SN не попадает в окно B. (ACK-шторм) 3. В данном случае, злоумышленник выступает в качестве посредника и может не только подслушивать, но активно подменять трафик. |
| **Отказ в обслуживании**  **Dos**  Приведение атакуемого узла или сети в такое состояние, когда передача данных другому узлу становиться невозможной | Smurf  От имени атакуемого узла производится рассылка широковещательных Echo-сообщений. Если число возвращений велико, то атакуемый узел и сети-усилители могут быть заблокированы шквалом ответных сообщений  SYN-flood  Состоит в посылке злоумышленником SYN-запросов на атакуемый узел в количестве большем, чем тот может обработать. Для этого:   1. Злоумышленник от имени несуществующего отправителя посылает SYN–запросы атакуемому узлу с определенной частотой. 2. При получении SYN-запроса модуль TCP создает блок обслуживания и отправляет подтверждение в виде SYN, на которое он никогда не получит ответ. В конце концов ресурсы узла истощаются и происходит отказ. |

|  |  |
| --- | --- |
| **На механизмы аутентификации (Authentication)** | **Подбор** (Brute Force)- процесс проб и ошибок (возможно автоматизированный), угадывания ключевой информации (имя пользователя, пароль, номер кредитной карточки, ключ шифрования и т.д.) Возникает в случае, когда система позволяет пользователям использовать слабые пароли или ключи шифрования. Существует прямой подбор используются различные варианты пароля для одного имени пользователя и обратный–пароль остается неизменным, а перебирает различные имена пользователей. |
| **Недостаточная аутентификация** (Insufficient Authentication) возможность получения доступа к информации или функциям сервера без должной аутентификации (доступ к ресурсам, сокрытым по определенному адресу, не указанному на основных страницах сервера или других общедоступных ресурсах).  Некоторые Web-приложения по умолчанию используют для административного доступа ссылку в корневой директории сервера (/admin/). Так как, разработчик предполагает, что воспользоваться этой страницей невозможно, в связи с тем, что ссылки на неё отсутствует, то реализацией аутентификации пренебрегают. Необходимый URL находится перебором файлов и директорий с использованием сообщений об ошибках, журналов перекрестных ссылок или путем простого чтения документации. |
| **Небезопасное восстановление паролей** (Weak Password Recovery Validation)–возможность несанкционированного получения, модификации или восстановления паролей других пользователей.  Функция восстановления пароля применяется Web-серверами, при утрате пароля. Из существующих методов восстановления пароля используется метод "секретного вопроса", ответ на который указывается в процессе регистрации; "подсказка", помогающая вспомнить пароль; часть персональных данных идентифицирующих пользователя (ИНН, домашний адрес почтовый индекс и т.д.). После прохождения процедуры идентичности, система отобразит пароль или перешлет его по почте. Уязвимость имеет место, если атакующий получает возможность использовать данный механизм. Например, использование секретного вопроса или указание email пользователя в комбинации с домашним адресом и номером телефона приводит к тому, что данную информацию нарушитель может получить из сетевых справочников и легко воспользоваться ею. Использование подсказок помогает в реализации подбора паролей, стойкость пароля "221277King" нарушается использованием подсказки "д-р+люб писатель", что помогает сформировать относительно короткий словарь для атаки путем перебора. |
| **На механизмы авторизации (Authorization)** | **Предсказуемое значение идентификатора** сессии (Credential/Session Prediction)– позволяет перехватывать сессии других пользователей (см атаку ????)  Доступ некоторых серверов предполагает аутентификацию пользователя в виде указания имени и пароля при первом обращении и дальнейшее отслеживание его сессии, для чего Web-сервер генерирует уникальный идентификатор, присвоенный сессии пользователя, позволяющий обращаться к серверу без повторной аутентификации. Идентификатор сессии храниться в cookie, скрытых полях форм или URL. Если нарушитель сможет определить алгоритм, используемый для генерации идентификатора сессии, то возможно:  1) подключение к серверу, используя текущий идентификатор сессии для доступа к ресурсам или реализации новой атаки ;  2) вычисление или подбор следующего идентификатора сессии;  3) присвоение полученного значения идентификатора cookie/скрытому полю формы/URL. |
| **Недостаточная авторизация** (Insufficient Authorization)– возможность получения информации или функций Web-сервера, доступ к которым должен быть ограничен.  Процедура авторизации определяет, какие действия может совершать пользователь, служба или приложение. Некоторые серверы, после аутентификации, сохраняют в cookie или скрытых полях идентификатор "роли" пользователя в рамках Web-приложения. Если разграничение доступа основывается на проверке данного параметра без верификации принадлежности к роли при каждом запросе, нарушитель может повысить свои привилегии, просто модифицировав значение cookie |
| **Отсутствие таймаута сессии** (Insufficient Session Expiration)– использование нарушителем перехваченного идентификатора, необходимого для авторизации.  Отсутствие таймаута сессии позволяет нарушителю воспользоваться историей браузера для просмотра страниц пользователя. Если функция выхода из системы перенаправляет на основную страницу Web-сервера, а не завершает сессию, страницы, посещенные пользователем, могут быть просмотрены злоумышленником, что, обеспечивает нарушителя доступом к страницам сервера без повторной аутентификации.  Большое значение таймаута увеличивает шансы подбора действующего идентификатора. |
| **Фиксация сессии** (Session Fixation)– присваивание идентификатору сессии пользователя заданное значение, необходимое нарушителю.  Выделяют два типа систем управления сессиями:"разрешающий"– позволяет браузеру указывать любой идентификатор; "строгий"–обрабатывает только идентификаторы, сгенерированные сервером, при этом нарушителю приходится поддерживать "сессию-заглушку" и периодически соединяться с сервером для избежание закрытия сессии по таймауту. В отличие от кражи идентификатора, данная атака предоставляет злоумышленнику гораздо больший простор для творчества. Атаки, направленные на фиксацию сессии обычно проходят в три этапа.  1) **Установление сессии** -заглушки на атакуемом сервере и получение или выбор идентификатора  2) **Фиксация сессии–**передача значение идентификатора сессии-заглушки браузеру пользователя и фиксация его идентификатора сессии (например, установив значение cookie в браузере с помощью XSS).  3**) Подключение к сессии–** после аутентификации пользователя на сервере нарушитель подключается к серверу, с зафиксированным идентификатором пользователя, и получает доступ к сессии пользователя. |
| **Атаки на клиентов (Client-side Attacks)** | **Подмена содержимого** (Content Spoofing)– подмена нарушителем страниц сгенерированных Web-сервером, на свои.  Некоторые Web-страницы создаются с использованием динамических источников HTML-кода. Например, фрейм <frame src="http://foo.example/1.html" передается в параметре URL строкой http://foo.example/page?frame\_src=http://foo.example/1.html. Если нарушитель заменить значение параметра "frame\_src" на "frame\_src= <http://attacker.example/spoof.html>", то будет отображаться результирующая страница, загруженная с сервера нарушителя, хотя в строке адреса браузера пользователя будет отображаться адрес сервера (foo.example). Эта атака так же может использована для создания ложных форм ввода пароля, пресс-релизов и т.д |
| **Межсайтовое выполнение сценариев** (Cross-site Scripting, XSS)– передача атакуемому серверу исполняемого кода, который будет перенаправлен браузеру пользователя.  Код создается на языках HTML/JavaScript, VBScript, ActiveX, Java, Flash, и т.д. Переданный код в URL, в заголовках HTTP запроса, значениях полей форм и т.д. исполняется на уязвимом сервере и получает возможность читать, модифицировать или передавать важные данные, доступные с помощью браузера. У атакованного пользователя может быть скомпрометирован аккакунт (кража cookie), его браузер может быть перенаправлен на другой сервер или осуществлена подмена содержимого сервера, возможно использование браузером жертвы для просмотра страниц сайта от имени атакуемого пользователя. |
| **Расщепление HTTP-запроса** (HTTP Response Splitting)– посылка нарушителем специальным образом сформированный запрос к серверу, ответ на который интерпретируется целью атаки как два разных ответа  Для реализации атаки необходимо как минимум три стороны: Web-сервер, содержащий уязвимость; цель атаки, взаимодействующая с Web-сервером под управлением нарушителя (кэширующий сервер-посредник или кэш браузера);сам нарушитель.  Атака осуществляется при возвращении сервером данных, предоставленных пользователем в заголовках HTTP–ответа (при перенаправлении пользователя на другую страницу или сохранение данных пользователя в cookie). Основой расщепления HTTP-запроса является внедрение символов перевода строки (CR,LF) для закрытия первой (стандартной) транзакции и формирования второй в виде вопроса–ответа, полностью контролируемую нарушителем для:  - межсайтового выполнение сценариев;  –модификации данных кэша прокси–сервера, подделанные нарушителем данные в ответ на запрос пользователя сохраняются на жестком диске и на последующие запросы пользователей по данному адресу возвращают кэшированные данные.  - межпользовательская атака–некоторые серверы-посредники разделяют одно TCP-соединение к серверу между несколькими пользователями, при этом второй и последующие пользователи получают в ответ страницу, сформированную нарушителем.  - Перехват страниц, содержащих пользовательские данные для получения доступа к важной или конфиденциальной информации. |
| **Выполнение кода (Command Execution)** | **Переполнение буфера** (Buffer Overflow)– изменение пути исполнения программы путем перезаписи данных в памяти системы.  Переполнение буфера возникает, если объем данных превышает размер выделенного под них буфер, данные переписывают другие области памяти, что приводит к возникновению ошибки. Если нарушитель имеет возможность управлять процессом переполнения, то может вызвать отказы в обслуживании, изменить путь исполнения программы и выполнить в её контексте различные действия.  Для этого используя переполнение буфера, нарушитель перезаписывает служебные области памяти, например, адрес возврата из функций в стеке или значения переменных в программе.  Данная атака обычно возникает при создании программ на языках C и C++. |
| **Атака на функции форматирования строк** (Format String Attack)– с помощью функций форматирования символьных переменных путь исполнения программы модифицируется методом перезаписи областей памяти  Уязвимость возникает, если пользовательские данные применяются в качестве аргументов функций форматирования строк (printf, setproctitle, syslog и т.д.) При передаче приложению строки, содержащей символы форматирования ("%f", "%p", "%n" и т.д.), у нарушителя появляется возможность:  - выполнить произвольный код на сервере;  - считать значения из стека;  - вызывать ошибки в программе/отказ в обслуживании.  Например, Web-приложение хранит переменную emailAddress, используемую в качестве аргумента функции printf для каждого пользователя. Если значение переменной содержит символы форматирования, функция printf будет обрабатывать их согласно заложенной в неё логики. Поскольку дополнительных значений этой функции не передано, будут использованы значения стека, хранящие другие данные |
| **Внедрение операторов LDAP** (LDAP Injection)– атаки направленные на Web-серверы, создающие запросы к службе LDAP на основе данных, вводимых пользователем. Если информация, полученная от клиента, должным образом не верифицируется, атакующий получает возможность модифицировать LDAP-запрос, выполняемый с тем же уровнем привилегий, с каким работает компонент приложения, выполняющий запрос (сервер СУБД, Web-сервер и т.д). Например, передача серверу строки: [http://example/ldapsearch.asp?user=\*](http://example/ldapsearch.asp?user=*) приводит к формированию запроса с фильтром uid=\*, что приводит к отображением всех объектов, имеющих атрибут uid. |
| **Выполнение команд ОС** (OS Commanding)– выполнение команд ОС на Web-сервере путем манипуляции входными данными.  Если информация, полученная от клиента, должным образом не верифицируется, атакующий имеет возможность выполнить команды ОС через компонент приложения. Например, некоторые языки сценариев позволяют запускать команды ОС, используя варианты функции exec. Если данные, передаются этой функции без проверки, нарушитель может выполнить команды ОС удаленно:  exec("ls -la $dir",$lines,$rc);  Используя символ ";" (Unix) в параметре dir можно выполнить команду ОС и просмотреть содержимое файла /etc/passwd:  http://example/directory.php?dir=%3Bcat%20/etc/passwd. |
| **Внедрение операторов SQL** (SQL Injection) создание SQL запросы к серверам СУБД на основе данных, вводимых пользователем.  Если информация, полученная от клиента, должным образом не верифицируется, атакующий получает возможность модифицировать запрос к SQL-серверу, отправляемый приложением, с возможностью полного контроля на сервером СУБД и даже его ОС.  Выделяют два метода эксплуатации внедрения операторов SQL: обычная атака–нарушитель подбирает параметры запроса, используя информацию об ошибках, генерируемую Web-приложением;  атака вслепую – сервер возвращает понятную для пользователя информацию о неправильном вводе. |
| **Внедрение серверных расширений** (SSI Injection)– передача исполняемого кода, который в дальнейшем будет выполнен на Web-сервере.  Перед генерацией HTML страницы сервер может выполнять сценарии, например Server-site Includes (SSI). В некоторых ситуациях исходный код страниц генерируется на основе данных, предоставленных пользователем. При передаче серверу операторов SSI, можно осуществить выполнение команд ОС или включить в страницу запрещенное содержимое при следующем отображении.  Например, выражение < !--#exec cmd="/bin/ls /" -- > может быть интерпретировано в качестве команды, просматривающей содержимое каталога сервера в Unix системах. |
| **Внедрение операторов XPath** (XPath Injection)– направлены на Web-серверы, создающие запросы на языке XPath на основе данных, вводимых пользователем.  Язык XPath предоставляет возможность обращения к частям документа на языке XML. Синтаксис XPath близок к языку SQL запросов. Если запросы XPath генерируются во время исполнения на основе пользовательского ввода, у атакующего появляется возможность модифицировать запрос с целью обхода логики работы программы. |
| **Разглашение информации (Information Disclosure)** | **Индексирование директорий** (Directory Indexin)–возможность получения информации о наличии файлов в Web каталоге, которые недоступны при обычной навигации по Web сайту.  На запрос пользователя начальной страницы сайта, сервер просматривает основную папку, находит в ней файл, используемый по умолчанию, и на его основе генерирует ответ. При отсутствии данной страницы, Web-сервер предоставляет список файлов в директории. В этой ситуации нарушитель может получить доступ к данным, не предназначенным для свободного доступа (например, администратор, предполагает, что если гиперссылка на документ отсутствует, то он недоступен). Однако современные сканеры уязвимостей могут найти спрятанное содержимое или другие файлы через:  –ошибки конфигурации возникают при настройке сложных конфигураций, где некоторые папки должны быть доступны для просмотра, которым может воспользоваться нарушитель.  –ошибки реализации, когда сервер генерирует список файлов при получении определенного запроса.  -базы данных поисковых машин, которые могут содержать кэш старых вариантов сервера, включая списки файлов |
| **Идентификация приложений** (Web Server/Application Fingerprinting)– определение версий приложений для получения информации об используемых сервером и клиентом ОС, Web-северах и браузерах. Обычно подобные атаки осуществляются путем анализа информации, предоставляемой Web-сервером, например: заголовками HTTP-ответов, расширением файлов (.asp, jsp), сообщений об ошибках. Наличие данной информации необходимо для реализации атаки, так как она специфична для каждого варианта ОС или приложения. Детальная информация об инфраструктуре позволяет снизить количество ошибок, и как следствие - общий «шум», производимый атакующим. |
| **Утечка информации** (Information Leakage)– уязвимость возникающая, при публикации на сервере информацию, которая может быть использована для компрометации системы (комментарии разработчиков или сообщения об ошибках). Анализ данной информации позволяет злоумышленнику произвести разведку и получить представление о структуре директорий сервера, используемых SQL запросах, названиях ключевых процессов и программ сервера. |
| **Обратный путь в директориях** (Path Traversal) получение доступа к файлам, директориям и командам, находящимся вне основной директории Web-сервера.  Большинство базовых атак, эксплуатирующих обратный путь, основаны на внедрении в URL символов "../", для изменения расположения ресурса, который будет обрабатываться сервером. При фильтрации этой последовательности сервером, злоумышленник может воспользоваться альтернативными кодировками для представления символов перехода по директориям (например, Unicode ("..%u2216" или "..%c0%af"), использование обратного слеша ("..\")). Возможность использования обратного пути в каталогах довольно часто возникает в приложениях, использующих механизмы шаблонов, а также в сценариях или CGI-программах. |
| **Предсказуемое расположение ресурсов** (Predictable Resource Location)– получение доступа к скрытым данным или функциональным возможностям  Нарушитель получает доступ к содержимому, не предназначенному для публичного просмотра (временные файлы, файлы резервных копий, конфигурации) путем подбора, который может быть оптимизирован использованием стандартного соглашения об именах файлов и директорий сервера. Получаемые файлы могут содержать информацию о дизайне приложения, информацию из баз данных, имена машин или пароли, пути к директориям, уязвимости, отсутствующие в основном приложении. Например наличие или отсутствие ресурса определяется по коду ошибки (404 в случае отсутствия папки или 403 в случае её наличия на сервере). |
| **Логические атаки (Logical Attacks**) | **Злоупотребление функциональными возможностями** (Abuse of Functionality)– использование функций Web-приложения для обхода механизмов разграничения доступа, например  - использование функций поиска для получения доступа к файлам за пределами корневой директории Web-сервера;  - использование функции загрузки файлов на сервер для перезаписи файлов конфигурации или внедрения серверных сценариев;  - реализация отказа в обслуживании путем использования функции блокировки учетной записи при многократном вводе неправильного пароля.  –использование программы "FormMail" для передачи данных из HTML-формы на указанный почтовый адрес может предоставить нарушителю возможность передавать почтовые сообщения в том числе и спам любому почтовому пользователю, при этом оставаясь полностью анонимным. |
| **Отказ в обслуживании** (Denial of Service) нарушение доступности Web-сервера  Атаки, направленные на отказ в обслуживании реализуются как на сетевом, так и на прикладном уровнях. Цель нарушителя использовать функции Web-приложения, для исчерпывания критичных ресурсов системы (вычислительные мощности, ОП, дисковое пространство, пропускная способность каналов связи), или прекращение функционирования системы. Атаки могут быть направлены на любой из компонентов Web-приложения: сервер СУБД, сервер аутентификации и т.д. |
| **Недостаточное противодействие автоматизации**(Insufficient Anti-automation)– эта уязвимость возникает, когда сервер позволяет автоматически выполнять операции, которые должны проводиться вручную  К автоматизированным программам относятся роботы поисковых систем, системы автоматизированного поиска уязвимостей и регистрации учетных записей. Подобные программы генерируют тысячи запросов в минуту, что может привести к падению производительности всего приложения. |
| **Недостаточная проверка процесса** (Insufficient Process Validation) –недостаточная проверка сервером последовательности выполнения операций приложения  В процессе доступа к некоторым функциям приложения ожидается, что пользователь выполнит ряд действий в определенном порядке, при не выполнении данного условия возникает ошибка, приводящая к нарушению целостности.  Например, система электронной торговли предлагает скидку на продукт B, в случае покупки продукта A. Пользователь, не желая покупать продукт A, заполняет заказ на покупку обоих продуктов, получая скидку. После чего возвращается к форме подтверждения заказа и удаляет продукт A из покупаемых, путем модификации значений в форме. Если сервер повторно не проверит возможность покупки продукта B по указанной цене без продукта A, будет осуществлена закупка по низкой цене. |

**Атаки на оборудование**

|  |  |
| --- | --- |
| Тип атак | Описание |
| MAC - spoofing | Атака канального уровня. На сетевой карте изменяется MAC-адрес, что заставляет коммутатор отправлять на порт, к которому подключен злоумышленник, пакеты, которые до этого он видеть не мог. |
| ARP - spoofing | Подмена ARP (Address Resolution Protocol) позволит злоумышленнику перехватывать кадры в локальной сети, изменять трафик, останавливать трафик или просто перехватывать весь трафик. |
| IP - spoofing | Используются IP – пакеты, отправляемые жертве, IP – адреса хоста, которому она доверяет; легко осуществима в UDP, в некоторых случаях возможна в TCP – соединениях.  Как правило, при осуществлении IP-спуфинга мошенники вставляют ложную информацию или вредоносные команды в обычный поток данных, которыми обмениваются клиентские и серверные приложения. Для обеспечения двухсторонней связи хакер изменяет все таблицы маршрутизации, чтобы направить трафик на ложный IP-адрес.  Обычно IP-спуфинг используется для сокрытия истинного адреса атакующего, а также с целью вызова ответного пакета данных на нужный адрес и пр. |
| DNS - spoofing | * Подмена DNS ID (DNS ID Spoofing)   Заголовок пакета DNS-протокола содержит идентификационное поле для соответствия запросов и ответов. Целью подмены DNS ID является посылка своего ответа на DNS-запрос до того, как ответит настоящий DNS-сервер. Для выполнения этого, нужно спрогнозировать идентификатор запроса. Локально это реализуется простым прослушиванием сетевого трафика. Однако, удаленно выполнить эту задачу гораздо сложнее. Существуют различные методы:   * Заражение кэша DNS – сервера жертвы ложной записью. |
| TCP Hijacking (перехват TCP-сеанса) | Разновидность атаки «Человек посередине». Атакующий способен просматривать пакеты участников сети и посылать свои собственные пакеты в сеть. |

|  |  |
| --- | --- |
| **атаки истощения** ресурсов *DHCP*-сервера | Для истощения ресурсов хакер организует рассылку множества ложных запросов на DHCP-сервер, который выдает адреса виртуальным ложным пользователям, а легальным - адресов не достается. Таким образом, организуется отказ в обслуживании легальных пользователей. |
| **атака подмены** легального *DHCP*-сервера ложным | Подмена легального *DHCP*-сервера ложным (DHCP-спуфинг) обычно реализуется после атаки истощения. Ложный DHCP-серверначинает раздавать клиентам ложные адреса, в том числе ложный адрес службы доменных имен *DNS*. Тем самым злоумышленникполучает доступ к передаваемой по сети информации |
| Спуфинг коммутатора | . Сетевой злоумышленник настраивает систему для имитации коммутатора. Используя этот метод, сетевой злоумышленник может заставить систему выглядеть как коммутатор с магистральным портом. В случае успеха атакующая система становится членом всех VLAN |
| Двойная маркировка (тегирование) | эта версия атаки включает в себя маркировку передаваемых кадров двумя идентификационными заголовками для пересылки кадров в неправильную VLAN |
| VLAN Hopping | смысл атаки заключается в том, что злоумышленник может получить доступ к другому устройству в сети не из своего VLAN.  Цель данной атаки в том, что пользователь может попытаться передать данные в другой VLAN. Давайте рассмотрим, как это сделать. Как подсказывает логика, это возможно, когда атакующий может инициировать trunk-режим.  В коммутаторах Cisco Catalyst по умолчанию порт работает не в режиме mode access и не в режиме mode trunk, таким образом, на порту работает протокол DTP (Dynamic Trunk Protocol).  В такой ситуации стоит атакующему «притвориться» коммутатором, как между ними будет установленно транковое соединение, и соответственно будут доступны VLAN, сконфигурированные на коммутаторе, и после чего передать данные в другой VLAN не составит труда.  Атака VLAN Hopping возможна только в тех случаях, когда интерфейсы настроены в режиме транка. |
| Атака на STP | злоумышленник с помощью программных средств может отправить коммутатору пакет BPDU (Bridge Protocol Data Unit), в котором указать высокий приоритет и меньший MAC-адрес и тем самым стать «корневым коммутатором» с целью перехвата сетевого трафика |