Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)**

УТВЕРЖДАЮ
Заведующий кафедрой ИСПИ

_____ И.Е. Жигалов
«20» марта 2025 г.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ МЕЖДИСЦИПЛИНАРНОГО КУРСА

«ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ РЕСУРСОВ»

В РАМКАХ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

«ТЕХНИЧЕСКАЯ ПОДДЕРЖКА И АДМИНИСТРИРОВАНИЕ ИНФОРМАЦИОННЫХ РЕСУРСОВ»

09.02.09 Веб-разработка
Разработчик веб приложений

Владимир, 2025

Методические указания к лабораторным работам междисциплинарного курса «Обеспечение безопасности информационных ресурсов» разработал доцент кафедры ИЗИ Полянский Д.А.

Методические указания к лабораторным работам рассмотрены и одобрены на заседании УМК специальности 09.02.09 Веб-разработка
протокол № 1 от «10» марта 2025 г.

Председатель УМК специальности _____ И.Е. Жигалов

Методические указания к лабораторным работам рассмотрены и одобрены на заседании кафедры ИСПИ
протокол № 7а от «12» марта 2025 г.

Рецензент от работодателя:
руководитель группы обеспечения
качества программного обеспечения
ООО «БСЦ МСК» _____ С.С. Смирнова

ЛАБОРАТОРНЫЙ ПРАКТИКУМ
по дисциплине
«Обеспечение безопасности информационных ресурсов»

СЕМЕСТР 5
Лабораторная работа № 1
«Антивирусные программы»

Задание

Выполнить сканирование и лечение локального диска с помощью не менее чем двух антивирусных программ.

Использовать все доступные режимы сканирования этих программ.

Порядок выполнения и защиты работы

1. Скопировать файлы из тестируемого локального диска (или директории) на другой диск (директорию).
2. Протестировать локальный диск (или директорию с количеством файлов не менее 1000) с функцией "лечение" во всех режимах работы не менее чем двумя антивирусными программами. Перед очередным тестированием и лечением восстанавливать файлы с резервной копии.
3. Оформить отчёт о проделанной работе.
4. Изучить контрольные вопросы.
5. Подготовить и представить преподавателю отчёт о выполнении работы. Отчёт может быть представлен в электронной форме.
6. Ответить на вопросы преподавателя из списка контрольных.

Содержание отчёта о выполнении лабораторной работы

1. Цель работы и общее задание
2. Общие характеристики компьютера, на котором выполнялось сканирование, в т.ч. локального диска. Количество и общий размер файлов из тестируемой директории.
3. Названия и версии антивирусных программ.
4. Краткое описание режимов работы антивирусных программ.
5. Сводная таблица результатов сканирования, содержащая следующую информацию для каждого из режимов сканирования:
   - название режима сканирования,
   - время сканирования,
   - количество проверенных объектов,
   - количество обнаруженных вирусов,
   - количество заражённых файлов,
   - количество вылеченных файлов.

6. Сравнительная характеристика использованных антивирусных программ и их режимов работы по различным показателям:
   - время работы,
   - количество обнаруженных вирусов,
   - количество вылеченных файлов,
   - количество удалённых файлов…
7. Выводы о режимах работы и эффективности использования исследуемых антивирусных программ.

Контрольные вопросы

1. Что такое компьютерный вирус?
2. Классифицируйте вирусы по среде обитания.
3. Классифицируйте вирусы по способу заражения.
4. Классифицируйте вирусы по характеру воздействия.
5. Классифицируйте вирусы по особенностям алгоритма.
6. Порядок функционирования загрузочного вируса.
7. Порядок функционирования файлового вируса.
8. Порядок функционирования загрузочно-файлового вируса.
9. Порядок функционирования стелс-вируса.
10. Порядок функционирования сетевого вируса.
11. Порядок функционирования полиморфного вируса.
12. Порядок функционирования макро-вируса.
13. Уровни полиморфизма вирусов.
14. Пути проникновения вирусов в КС.
15. Признаки появления вирусов.
16. Общие методы защиты от вирусов.
17. Классифицируйте средства защиты от вирусов по способу функционирования.
18. Работа программ-детекторов.
19. Работа программ-ревизоров.
20. Работа программ-фильтров.
21. Работа программ-иммунизаторов.

Лабораторная работа № 2
«Управление учётными записями Windows»

Теоретические сведения

1. Учётные записи пользователей

Основой системы разграничения доступа в ОС является понятие **учётной записи**. Для каждого зарегистрированного пользователя система создаёт свою учётную запись. Учётная запись – это запись в специальной базе данных системы, содержащая информацию о пользователе, а также данные для аутентификации пользователя. Каждый раз при аутентификации пользователя, происходит сравнение введённых им аутентификационных данных с данными из базы, и при совпадении пользователь получает соответствующий доступ к ОС.

Примечание. *Если рабочая станция входит в состав локальной вычислительной сети (ЛВС) на основе домена, то помимо локальных учётных записей, существуют доменные учётные записи, которые хранятся не на локальной системе, а на сервере (контроллере домена). Проверку аутентификационных данных таких учётных записей осуществляет контроллер домена.*

Все учётные записи пользователей ОС можно условно разделить на четыре категории:

- **Системные (специальные) учётные записи** создаются автоматически при установке ОС. Они отличаются от обычных пользовательских учётных записей, и возможности по управлению ими ограничены (например, их нельзя удалить). Такие учётные записи, как правило, играют особую роль в ОС и необходимы для выполнения различных специальных функций (например, загрузки критических системных процессов). Интерактивный вход под такими учётными записями чаще всего невозможен.
- **Встроенные учётные записи пользователей** создаются автоматически при установке ОС. Они практически не отличаются от обычных пользовательских учётных записей, но возможности по управлению ими, как правило, ограничены (например, их нельзя удалить). Такие учётные записи необходимы самой ОС для функционирования СРД «по умолчанию».
- **Стандартные учётные записи** пользователей также создаются автоматически при установке ОС. Однако они не являются обязательными и выполняют вспомогательные функции. Перечень таких учётных записей может сильно варьироваться в различных ОС линейки Windows и даже в рамках различных service packs. Возможности по управлению такими учётными записями не ограничены.

– **Пользовательские учётные записи** – зарегистрированные пользователи ОС. Такие учётные записи создаются и управляются администратором системы или пользователем, имеющим соответствующие права.

Ниже приведены **системные (специальные) учётные записи ОС Windows** и их назначение:

Учётная запись	Описание
SYSTEM (Локальная система)	Это **системная учётная запись**. Обладает **неограниченными правами** в системе. Используется самой ОС для загрузки компонентов ядра, системных сервисов (служб) и библиотек. Недоступна для интерактивного входа. Изменение атрибутов учётной записи также невозможно.
LOCAL SERVICE (Локальная служба)	Это ограниченная системная учётная запись. Обычно используется для загрузки локальных сервисов (служб) системы. Недоступна для интерактивного входа. Изменение атрибутов учётной записи также невозможно.
NETWORK SERVICE (Сетевая служба)	Это ограниченная системная учётная запись. Обычно используется для загрузки сетевых сервисов (служб) системы. Недоступна для интерактивного входа. Изменение атрибутов учётной записи также невозможно.

Ниже приведены **встроенные учётные записи ОС Windows** и их назначение:

Учётная запись	Описание
Administrator (Администратор)	Учётная запись **администратора** системы. Необходима для выполнения многих административных задач (управление настройками системы, управление другими пользователями и т.п.). Обычно обладает наивысшими правами среди доступных для интерактивного входа учётных записей. Пароль задается при установке системы.
Guest (Гость)	Гостевая учётная запись. Обычно **обладает минимальными правами** и предназначена для входа анонимных пользователей. Отключена по умолчанию. Как правило, на учётной записи установлен пароль по умолчанию “guest”.

Ниже приведены примеры **стандартных учётных записей ОС Windows** и их назначение:

Учётная запись	Описание
HelpAssistant	Учётная запись для предоставления удаленной помощи
SUPPORT_388945a0	Это учётная запись поставщика для службы справки и поддержки

В Windows вся информация об учётных записях хранится в специальном разделе системного реестра. На жестком диске соответствующий раздел находится в файле **%SystemRoot%\system32\config\sam**. Доступ к данному файлу (и соответствующему разделу системного реестра) имеет только **учётная запись SYSTEM**. Даже **Administrator** не имеет прямого доступа к базе учётных записей ОС.

2. Группы пользователей

Каждая учётная запись обладает определёнными правами доступа и привилегиями в системе. Эти права могут выставляться администратором для каждой учётной записи отдельно. Однако это не всегда удобно, т.к. многие пользователи обладают одинаковыми правами доступа, и приходится для соответствующих учётных записей выставлять одни и те же права. Поэтому ещё одним инструментом управления разграничением доступа в ОС являются **группы.**

Группа – это совокупность учётных записей, обладающих одинаковыми правами. Каждая отдельная учётная запись может принадлежать к одной или нескольким группам, и, следовательно, обладать правами группы.

Все группы можно условно разделить на две категории:

– **Стандартные группы** пользователей создаются автоматически при установке ОС.

– **Пользовательские группы** – зарегистрированные группы ОС. Такие группы создаются и управляются администратором или пользователем, имеющим соответствующие права.

Далее перечислим и охарактеризуем основные **стандартные группы** ОС Windows.

Администраторы - членство в этой группе по умолчанию предоставляет самый широкий набор прав и возможность изменять собственные права. По умолчанию членом этой группы является только встроенная учётная запись Администратора. Права Администратора в системе практически неограниченны, хотя учётная запись SYSTEM обладает ещё более высокими привилегиями.

В целях безопасности рекомендуется использовать административный доступ только для выполнения следующих действий:

- установки операционной системы и её компонентов (например, драйверов устройств, системных служб и так далее);
- установки пакетов обновления;
- обновления операционной системы;
- восстановления операционной системы;
- настройки важнейших параметров операционной системы (политики паролей, управления доступом, политики аудита, настройки драйверов в режиме ядра и так далее);
- вступления во владение файлами, ставшими недоступными;
- управления журналами безопасности и аудита;
- архивирования и восстановления системы.

Опытные пользователи - эта группа поддерживается, в основном, для совместимости с предыдущими версиями и для выполнения несертифицированных приложений. Разрешения по умолчанию, предоставленные этой группе, позволяют членам группы изменять параметры ОС. Члены группы Опытные пользователи имеют больше прав, чем члены группы Пользователи, и меньше, чем члены группы Администраторы. Опытные пользователи могут выполнять любые задачи с ОС, кроме задач, зарезервированных для группы Администраторы (например, установка служб и драйверов).

Опытные пользователи могут:

- устанавливать программы, не изменяющие файлы операционной системы, и системные службы;
- настраивать ресурсы на уровне системы, включая принтеры, дату и время, параметры электропитания и другие ресурсы панели управления.

Опытные пользователи не могут добавлять себя в группу Администраторы. Они не имеют доступа к данным других пользователей на томе NTFS, если соответствующие разрешения этих пользователей не получены.

Пользователи - членами этой группы обычно являются рядовые пользователи системы. Группа Пользователи предоставляет самую безопасную среду для выполнения программ. На разделах с файловой системой NTFS параметры безопасности по умолчанию разработаны, чтобы предотвратить нарушение целостности операционной системы и установленных программ членами этой группы. Пользователи не могут изменять параметры реестра на уровне системы, файлы операционной системы или программы. Они не могут организовывать общий доступ к каталогам или создавать локальные принтеры. Пользователи имеют полный доступ только к своим файлам данных и только к своей части реестра (HKEY_CURRENT_USER). Права на уровне пользователя часто не допускают выполнение пользователем различных приложений. Учётные записи, входящие в группу Пользователи, не могут устанавливать новые приложения в систему и гарантированно могут запускать только

сертифицированные приложения. **Именно в эту группу по умолчанию попадают вновь созданные учётные записи.**

Операторы архива - члены этой группы могут архивировать и восстанавливать файлы на компьютере независимо от всех разрешений, которыми защищены эти файлы. Они могут также входить в систему и завершать работу компьютера.

Гости - члены этой группы по умолчанию имеют минимальные права в системе.

Операторы настройки сети - члены этой группы могут иметь некоторые административные права для управления настройкой сетевых параметров.

Пользователи удалённого рабочего стола - члены этой группы имеют право на выполнение удалённого входа в систему.

3. Управление учётными записями и группами

Управление учётными записями в Windows можно осуществлять, используя следующие средства:

1. Оснастка Локальные пользователи и группы (lusrmgr.msc)
2. Оснастка Управление компьютером (compmgmt.msc)
3. Панель управления > Учётные записи пользователей (nusrmgr.cpl)
4. Командная строка (NET USER, NET LOCALGROUP)

Наиболее мощным и удобным средством управления является оснастка «Локальные пользователи и группы». Однако средства командной строки незаменимы для автоматизации задач управления учётными записями.

Порядок выполнения работы

1. Изучить теоретический материал лабораторной работы, в том числе самостоятельно команды NET USER и NET LOCALGROUP.
2. Изучить управление учётными записями в Windows через оснастку **Локальные пользователи и группы.**

Для **вызова оснастки** необходимо:

а) Нажать **Пуск > Выполнить**

б) Набрать *lusrmgr.msc* и нажать ОК

3. Изучить управление учётными записями в Windows через консоль.

Для **управления учётными записями через консоль** необходимо:

a. Нажать **Пуск > Выполнить**
b. Набрать *cmd* и нажать ОК
c. Использовать команды **net user** и **net localgroup**

4. Создать учётные записи, показанные в следующей таблице. Затем протестировать процесс входа в систему, используя одну из созданных учётных записей

Имя пользователя	**Полное имя**	**Пароль**	**Менять пароль**

Userl	User One	Нет пароля	Должен
User2	User Two	Нет пароля	Не должен
User3	User Three	User3	Должен
User4	User Four	User4	Не должен

5. Создать две локальные группы Group1 и Group2 и добавить в них учётные записи пользователей.

Имя группы	**Описание группы**	**Члены группы**
Group1	пользователи	user1, user2
Group2	пользователи	user3, user4

a. Затем следует удалить пользователей из первой группы и добавить его во вторую группу
b. Добавить пользователей с правами администратора в первую группу
c. После этого удалить обе созданные группы.

Содержание отчёта о выполнении лабораторной работы

8. Цель работы и общее задание
9. Общие характеристики компьютера, на котором выполнялась работа.
10. Использованные в процессе работы команды (полностью с конкретными ключами и параметрами.
11. Скриншоты каждого этапа выполнения задания.
12. Выводы о проделанной работе.

Контрольные вопросы

22. В чем разница между доменными и локальными учётными записями?
23. Какие виды учётных записей вам известны?
24. Какая информация необходима для создания локальной учётной записи?
25. Как создать учётные записи пользователей в системе?
26. Зачем нужно использовать группы?
27. Как создать локальную группу?
28. Каковы последствия удаления группы?
29. Каковы различия между встроенными и обычными локальными группами?

30. Перечислите встроенные группы пользователей

Лабораторная работа № 3
по дисциплине
«Установка и настройка программного межсетевого экрана»

Теоретические сведения

Межсетевым экраном называется программно-аппаратный или программный элемент, контролирующий на основе заданных параметров сетевой трафик, а в случае необходимости и фильтрующий его. Также может называться фаейрволом (Firewall) или брандмауэром.

Назначение межсетевых экранов

Сетевой экран используется для защиты отдельных сегментов сети или хостов от возможного несанкционированного проникновения через уязвимости программного обеспечения, установленного на ПК, или протоколов сети. Работа межсетевого крана заключается в сравнении характеристик проходящего сквозь него трафика с шаблонами уже известного вредоносного кода.

Наиболее часто сетевой экран инсталлируется на границе периметра локальной сети, где он выполняет защиту внутренних узлов.

Это стало причиной, по которой брандмауэры стали устанавливать не только на границе сети, но и между её сегментами, что значительно повышает степень безопасности сети.

Фильтрация трафика

Трафик фильтруется на основе заданных правил – ruleset. По сути, межсетевой экран представляет собой последовательность анализирующих и обрабатываемых трафик фильтров согласно данному пакету конфигураций.

У каждого фильтра своё назначение; причём, последовательность правил может значительно влиять на производительность экрана.

К примеру, большинство файрволов при анализе трафика последовательно сравнивают его с известными шаблонами из списка – очевидно, что наиболее популярные виды должны располагаться как можно выше.

Принципов, по которому осуществляется обработка входящего трафика, бывает два.

Согласно первому разрешаются любые пакеты данных, кроме запрещённых, поэтому если он не попал ни под какое ограничение из списка конфигураций, он передается далее.

Согласно второму принципу, разрешаются только те данные, которые не запрещены – такой метод обеспечивает самую высокую степень защищенности, однако существенно нагружает администратора.

Межсетевой экран выполняет две функции: deny, запрет данных – и allow – разрешение на дальнейшую передачу пакет.

Некоторые брандмауэры способны выполнять также операцию reject – запретить трафик, но сообщить отправителю о недоступности сервиса, чего не происходит при выполнении операции deny, обеспечивающей таким образом большую защиту хоста.

Типы межсетевых экранов (Firewall)

Чаще всего межсетевые экраны классифицируют по поддерживаемому уровню сетевой модели OSI. Различают:

- Управляемые коммутаторы;
- Пакетные фильтры;
- Шлюзы сеансового уровня;
- Посредники прикладного уровня;
- Инспекторы состояния.

Управляемые коммутаторы

Нередко причисляются к классу межсетевых экранов, но осуществляют свою функцию на канальном уровне, поэтому не способны обработать внешний трафик.

Некоторые производители (ZyXEL, Cisco) добавили в свой продукт возможность обработки данных на основе MAC-адресов, которые содержатся в заголовках фреймов.

Тем не менее, даже этот метод не всегда приносит ожидаемый результат, так как мак-адрес можно легко изменить с помощью специальных программ.

Виртуальные локальные сети позволяют организовывать группы хостов, в которые данные стопроцентно изолированы от внешних серверов сети.

В рамках корпоративных сетей управляемые коммутаторы могут стать весьма эффективным и сравнительно недорогим решением. Главным их минусом является неспособность обрабатывать протоколы более высоких уровней.

Пакетные фильтры

Пакетные фильтры используются на сетевом уровне, осуществляя контроль трафика на основе данных из заголовка пакетов.

Нередко способны обрабатывать также заголовки протоколов и более высокого уровня – транспортного (UDP, TCP), Пакетные фильтры стали самыми первыми межсетевыми экранами, остаются самыми популярными и на сегодняшний день. При получении входящего трафика анализируются такие данные, как: IP получателя и отправителя, тип протокола, порты получателя и источника, служебные заголовки сетевого и транспортного протоколов.

Уязвимость пакетных фильтров заключается в том, что они могут пропустить вредоносный код, если он разделен на сегменты: пакеты выдают себя за часть другого, разрешённого контента.

Решение этой проблемы заключается в блокировании фрагментированных данных, некоторые экраны способны также дефрагментировать их на собственном шлюзе – до отправки в основной узел сети.

Тем не менее, даже в этом случае межсетевой экран может стать жертвой DDos-атаки.

Пакетные фильтры отличаются высокой скоростью анализа пакетов, отлично выполняют свои функции на границах с сетями низкой степени доверия.

Тем не менее, они неспособны анализировать высокие уровни протоколов и легко могут жертвами атак, при которых подделывается сетевой адрес.

Шлюзы сеансового уровня

Использование сетевого экрана позволяет исключить прямое взаимодействие внешних серверов с узлом – в данном случае он играет роль посредника, называемого прокси.

Он проверяет каждый входящий пакет, не пропуская те, что не принадлежат установленному ранее соединению.

Те пакеты, которые выдают себя за пакеты уже завершённого соединения, отбрасываются.

Тем не менее, даже у этого решения есть значительный минус: ввиду отсутствия возможности проверки содержания поля данных хакер относительно легко может передать в защищаемую сеть трояны.

Посредники прикладного уровня

Как и шлюзы сеансового уровня, фаейрволы прикладного уровня осуществляют посредничество между двумя узлами, но отличаются существенным преимуществом – способностью анализировать контекст передаваемых данных. Сетевой экран подобного типа может определять и блокировать нежелательные и несуществующие последовательности команд (подобное часто означает ДОС-атаку), а также запрещать некоторые из них вообще.

Посредники прикладного уровня определяют и тип передаваемой информации – ярким примером являются почтовые службы, запрещающие передачу исполняемых файлов. Кроме этого они могут осуществлять аутентификацию пользователя, наличие у SSL-сертификатов подписи от конкретного центра.

Главным минусом такого типа сетевого экрана является долгий анализ пакетов, требующий серьёзных временных затрат. Помимо этого, у

посредников прикладного уровня нет автоподключения поддержки новых протоколов и сетевых приложений.

Инспекторы состояния

Создатели инспекторов состояния поставили перед собой цель собрать воедино преимущества каждого их выше перечисленных типов сетевых экранов, получив таким образом брандмауэр, способный обрабатывать трафик как на сетевом, так и на прикладном уровнях.

Инспекторы состояния осуществляют контроль:

- всех сессий – основываясь на таблице состояний,
- всех передаваемых пакетов данных – на основе заданной таблицы правил,
- всех приложений, на основе разработанных посредников.

Фильтрация трафика инспектора состояния происходит тем же образом, что и при использовании шлюзов сеансового уровня, благодаря чему его производительность гораздо выше, чем у посредников прикладного уровня. Инспекторы состояния отличаются удобным и понятным интерфейсом, лёгкой настройкой, обладают широкими возможностями расширения.

Реализация межсетевых экранов

Межсетевые экраны (Firewall) могут быть либо программно-аппаратными, ибо программными. Первые могут быть выполнены как в виде отдельного модуля в маршрутизаторе или коммутаторе, так и специального устройства.

Чаще всего пользователи выбирают исключительно программные межсетевые экраны – по той причине, что для их использования достаточно лишь установки специального софта.

Тем не менее, в организациях нередко найти свободный компьютер под заданную цель, бывает затруднительно – к тому же, отвечающий всем техническим требованиям, зачастую довольно высоким.

Именно поэтому крупные компании предпочитают установку специализированных программно-аппаратных комплексов, получивших название «security appliance». Работают они чаще всего на основе систем Linux или же FreeBSD, ограниченных функционалом для выполнения заданной функции.

Такое решение имеет следующие преимущества:

- Лёгкое и просто управление: контроль работы программно-аппаратного комплекса осуществляется с любого стандартного протокола (Telnet, SNMP) – или защищённого (SSL, SSH).
- Высокая производительность: работа операционной системы направлена на одну единственную функцию, из неё исключены любые посторонние сервисы.

- Отказоустойчивость: программно-аппаратные комплексы эффективно выполняют свою задачу, вероятность сбоя практически исключена.

Ограничения межсетевого экрана (Firewall-a)

Сетевой экран не проводит фильтрацию тех данных, которые не может интерпретировать.

Пользователь сам настраивает, что делать с нераспознанными данными – в файле конфигураций, согласно которым и осуществляется обработка такого трафика.

К таким пакетам данным относятся трафик из протоколов SRTP, IPsec, SSH, TLS, которые используют криптографию для скрытия содержимого, протоколы, шифрующие данные прикладного уровня (S/MIME и OpenPGP).

Как работает межсетевой экран

Фильтрация трафика происходит на основе заранее установленных правил безопасности.

Для этого создается специальная таблица, куда заносится описание допустимых и недопустимым к передаче данных.

Межсетевой экран не пропускает трафик, если одно из запрещающих правил из таблицы срабатывает.

Файрволы могут запрещать или разрешать доступ, основываясь на разных параметрах: IP-адресах, доменных именах, протоколах и номерах портов, а также комбинировать их.

- IP-адреса. Каждое устройство, использующее протокол IP, обладает уникальным адресом. Вы можете задать определенный адрес или диапазон, чтобы пресечь попытки получения пакетов. Или наоборот — дать доступ только определенному кругу IP-адресов.
- Порты. Это точки, которые дают приложениям доступ к инфраструктуре сети. К примеру, протокол ftp пользуется портом 21, а порт 80 предназначен для приложений, используемых для просмотра сайтов. Таким образом, мы получаем возможность воспрепятствовать доступу к определенным приложениям и сервисам.
- Доменное имя. Адрес ресурса в интернете также является параметром для фильтрации. Можно запретить пропускать трафик с одного или нескольких сайтов. Пользователь будет огражден от неприемлемого контента, а сеть от пагубного воздействия.
- Протокол. Файрвол настраивается так, чтобы пропускать трафик одного протокола или блокировать доступ к одному из них. Тип протокола указывает на набор параметров защиты и задачу, которую выполняет используемое им приложение.

Недостатки МЭ

Межсетевые экраны обороняют сеть от злоумышленников. Однако необходимо серьезно отнестись к их настройке.

Будьте внимательны: ошибившись при настройке параметров доступа, вы нанесете вред и файрвол будет останавливать нужный и ненужный трафик, а сеть станет неработоспособной.

Применение межсетевого экрана может стать причиной падения производительности сети. Помните, что они перехватывают весь входящий трафик для проверки.

При крупных размерах сети чрезмерное стремление обеспечить безопасность и введение большего числа правил приведет к тому, что сеть станет работать медленно.

Зачастую одного файрвола недостаточно, чтобы полностью обезопасить сеть от внешних угроз. Поэтому его применяют вместе с другими программами, такими как антивирус.

Примеры ПМЭ

Межсетевые экраны обороняют сеть от злоумышленников. Однако необходимо серьезно отнестись к их настройке.

Comodo Firewall – безплатный фаервол, который обеспечивает высокий уровень защиты от сетевых угроз, блокирует вредоносное ПО и защищает компьютер от хакерских атак. Сканирует все процессы и соединения во время интернет сёрфинга, извещая пользователя о подозрительных операциях.

Avast! Internet Security – комплексное решение для защиты операционной системы от сетевых угроз в режиме реального времени. Эта безплатная программа имеет широкий набор инструментов для многоуровневой защиты ПК от вредоносных программ и рекламного ПО.

Outpost Firewall Pro – кастомизируемый фаервол с проактивной защитой. Имеет удобный компактный интерфейс с русским языком и позволяет свободно использовать все функции приложения без чтения справок.

ZoneAlarm Free Firewall – простой, но эффективный фаервол, который контролирует приложения и процессы на компьютере, обеспечивая защиту от хакерских атак и вредоносного ПО. Благодаря двусторонней защите, компьютер пользователя становится невидимым в сети.

Kerio WinRoute Firewall – полный пакет для обеспечения сетевой безопасности компьютера, который разрабатывался специально как корпоративный брендмауэр и может предложить большое количество функций, предназначенных именно для корпоративного использования. Помимо стандартных функций антивируса и фаервола, имеет встроенный VPN-сервер, удобное управление доступом к сайтам, фильтрации контента, поддержкой протоколов VoIP и UPnP.

Порядок выполнения и защиты работы

1. Выбрать программный межсетевой экран ПМЭ из представленных выше или любой другой с модульным принципом построения и изучить его возможности по управлению доступом к ресурсам защищаемого компьютера.
2. Выполнить установку и настройку ПМЭ в соответствии с заданием. Наименования модулей в разных ПМЭ могут отличаться, но модули должны иметь указанный функционал. Представленные в задании и контрольных вопросах наименования и функции даны для МПЭ **Outpost Firewall Pro**.
3. Подготовить и представить преподавателю отчёт о выполнении работы. Отчёт может быть представлен в электронной форме.

Задание

1. Выполнить установку программного межсетевого экрана ПМЭ.
2. Выполнить настройку ПМЭ:
   - Установить режим работы ПМЭ, позволяющий изменять настройки во время работы при возникновении нештатных ситуаций.
   - Настроить конфигурацию доверенной зоны (при наличии локальной сети; при её отсутствии в отчёте описать возможности настройки).
   - Разработать не менее одного нового правила работы для одного или нескольких сетевых приложений.
   - Настроить подключаемые модули фильтрации HTML-страниц:
     - установить все возможности блокирование баннеров и рекламной информации,
     - настроить блокирование Web-сайтов по определенным адресам,
     - ограничить использование элементов cookies и всплывающих окон для всех Web-страниц,
   - Настроить модуль детектора атак: установить режимы работы данного модуля, в соответствии с возможностями ПМЭ, ограничивающие возможности удаленного доступа к защищаемому компьютеру.
   - Настроить другие подключаемые модули, входящие в комплект ПМЭ в соответствие с их возможностями и общими рекомендациями, даваемыми разработчиком ПМЭ.
   - Определить общие настройки, ограничивающие доступ к настройке параметров ПМЭ локальным злоумышленникам.

Содержание отчёта о выполнении лабораторной работы

13. Цель работы и общее задание
14. Общее описание режимов работы ПМЭ.

15. Краткие пояснения по каждому установленному параметру ПМЭ (назначение, список возможных значений).
16. Выводы о проделанной работе.

Контрольные вопросы

1. Что такое ПМЭ?
2. Перечислите основные возможности ПМЭ.
3. Что такое модули и модульный принцип организации ПМЭ?
4. Какие существуют политики работы с сетью?
5. Что такое режим обучения ПМЭ?
6. На какие три группы можно разделить все приложения с точки зрения ПМЭ?
7. Назовите основные угрозы, возникающие при работе с сетью.
8. Какие возможности ПМЭ можно использовать для защиты от проникновения на компьютер посторонних программ?
9. Какие возможности ПМЭ можно использовать для защиты от попыток получения доступа к информации на компьютере?
10. Какие возможности ПМЭ можно использовать для защиты от ненужной информации?
11. Что такое доверенная зона?
12. Каким образом предопределённые правила разделены на группы?
13. Из каких элементов состоит правило для работы приложения?
14. Перечислите несколько событий, используемых при построении правил.
15. Какие стеки протоколов используют при построении правил?
16. Какие существуют направления сетевых подключений?
17. Какие действия может выполнить система при выполнении условий правила?
18. В чём отличие обычного режима от режима невидимости при обработке запроса на соединение?
19. В чём заключается назначение модуля работы с DNS?
20. В чём заключается назначение модулей фильтрации содержимого Web-страниц?
21. В чём заключается назначение модуля защиты файлов?
22. В чём заключается назначение модуля детектора атак?
23. Какие существуют уровни тревоги?
24. Какие существуют виды блокировки при обнаружении атаки?
25. Что такое конфигурация и какие возможности работы с ней предоставляет ПМЭ?

СЕМЕСТР 6

Лабораторная работа № 1

«Сбор информации о веб-приложении»

Краткие теоретические сведения

Одним из первых этапов анализа защищенности любой компьютерной системы является сбор информации. В зависимости от используемой методологии анализа защищенности веб-приложения можно применять различные методы и средства сбора информации. Стоит отметить, что сбор информации характерен для методологии тестирования на возможность проникновения.

Методы сбора информации делят на активные и пассивные.

Активные методы требуют непосредственного взаимодействия с исследуемым приложением путем отправки ему запросов и анализа соответствующих ответов, а пассивные методы используют информацию, отправляемую сервером веб-приложения его клиентам (например, HTTP-заголовки X-Frame-Options, Strict-Transport-Security и т.д.) без отправки запросов. При анализе веб-приложений, как правило, используют только активные методы.

Активные методы делят на методы с подключением к приложению (например, идентификация веб-сервера с помощью сканера Httprecon) и методы без подключения (например, сбор информации о приложении поисковыми роботами, сканерами Интернет, и т.д.).

В результате проведения сбора информации о веб-приложении могут быть получены:

- имена и IP-адреса сетевых узлов, на которых размещены веб-приложение и его компоненты;
- логины и пароли технологических учетных записей;
- комментарии разработчиков;
- данные о системном и прикладном ПО, применяемых средствах защиты и конфигурации веб-приложения;
- адреса электронной почты разработчиков приложения;
- исходный код серверной части веб-приложения;
- конфиденциальные файлы.

Программными средствами получения необходимой информации являются:

- поисковые системы (например, Google, Shodan, Bing);
- специализированные сканеры уязвимостей Интернет (например, http://un1c0rn.net/);
- инструментальные средства анализа защищенности сетей общего назначения (Nmap, Xprobe2, XSpider);
- инструментальные средства анализа защищенности сетей веб-приложений (AppScan, Acunetix, Burp Suite, ZAP, W3AF и т.д.).

Задание

Выполнить сбор информации об анализируемом веб-приложении. В качестве такового можно использовать любое доступное веб-приложение (в том числе разработанное ранее в рамках других учебных курсов) либо из перечня в Приложении 1.

Поскольку ряд запросов, реализуемых в лабораторной работе, могут быть отклонены правилами политики информационной безопасности локальной сети, рекомендовано выполнять такие задания на виртуальной машине.

Порядок выполнения работы

7. В адресной строке браузера перейти по адресу <веб-адрес тестируемого приложения>/robots.txt. Проанализировать содержимое файла. Сделать выводы о наличии «скрытых» директорий.
8. В адресной строке браузера перейти по адресу http://<веб-адрес тестируемого приложения>/crossdomain.xml и, затем, по адресу http://<веб-адрес тестируемого приложения>/clientaccesspolicy.xml. Проанализировать содержимое файлов. Сделать выводы о корректности конфигурации политики междоменного взаимодействия.
9. Перейти по адресу http://www.google.com (или другой поисковый сервис). Задать поисковые запросы, определяемые анализируемым приложением, например:
   - site:<веб-адрес тестируемого приложения> filetype:docx confidential
   - site:<веб-адрес тестируемого приложения> filetype:doc secret
   - site:<веб-адрес тестируемого приложения> inurl:admin
   - site:<веб-адрес тестируемого приложения> filetype:sql
   - site:<веб-адрес тестируемого приложения> intext: "Access denied"

   Проанализировать логику запросов и полученные данные. Построить свои запросы, используя примеры из баз запросов (например, http://www.exploit- db.com/google-dorks).
10. Перейти по адресу приложения. Задать следующий поисковый запрос:
    hostname:<веб-адрес тестируемого приложения>
11. Данный тест выполняется только для приложений, размещенных в локальной сети (либо на виртуальной машине). С помощью сетевого сканера Nmap выполнить идентификацию ОС веб-сервера:
    # nmap –O <веб-адрес тестируемого приложения> –vv
12. Подключиться к веб-серверу, используя утилиту Netcat:
    # nc <веб-адрес тестируемого приложения> 80
    Отправить следующий GET запрос:
    GET / HTTP/1.1
    Host: <веб-адрес тестируемого приложения>
    \r\n

    По заголовкам Server и X-Powered-By определить программное обеспечение, реализующее веб-сервер и framework веб-приложения.

В браузере установить расширение Wappalyzer, перейти по адресу веб-приложения и проанализировать информацию о компонентах веб-приложения полученную через Wappalyzer.

13. Выполнить тесты по идентификации поддерживаемых веб-сервером HTTP-методов. Для этого необходимо отправить с помощью Burp Suite или Netcat запрос следующего вида:

    OPTIONS / HTTP/1.1
    Host: <веб-адрес тестируемого приложения>
    \r\n

    Проверить, поддерживает ли сервер обработку запросов с произвольными методами:

    DOGS / HTTP/1.1
    Host: <веб-адрес тестируемого приложения>
    \r\n

    Если веб-сервер поддерживает метод TRACE, то это может привести к уязвимости к атаке Cross-Site Tracing (XST). Для проверки поддержки веб-сервером методы TRACE отправить запрос

    TRACE / HTTP/1.1
    Host: <веб-адрес тестируемого приложения>
    \r\n

    Веб-сервер поддерживает метод TRACE и потенциально уязвим к атаке XST, если получен ответа вида

    HTTP/1.1 200 OK
    Connection: close
    Content-Length: 39
    TRACE / HTTP/1.1
    Host: <веб-адрес тестируемого приложения>

Бонусное задание (для выполнения вне лаборатории). Известно, что адрес веб-интерфейса системы VMWare Horizon View HTML Access содержит строку
portal/webclient/views/mainUI.html.

Найти такие системы, доступные из сети Интернет.

Содержание отчёта о выполнении лабораторной работы

17. Цель работы и общее задание
18. Краткое описание тестируемых веб-приложений.
19. Названия и версии средств тестирования (утилит, расширений и т.д.).
20. Использованные дополнительные запросы (по п.3) с краткой характеристикой.
21. Полученные результаты тестирования.
22. Выводы.

ПРИЛОЖЕНИЕ 1

Перечень веб-приложений, которые могут быть использованы для тестирования и являющиеся небезопасными

1. http://testphp.vulnweb.com
2. http://testaspnet.vulnweb.com
3. http://testasp.vulnweb.com
4. http://testhtml5.vulnweb.com
5. http://crackme.cenzic.com
6. http://demo.testfire.net
7. http://aspnet.testsparker.com
8. http://php.testsparker.com/
9. http://www.webscantest.com/
10. https://hack.me
11. http://pentesteracademylab.appspot.com
12. http://zero.webappsecurity.com
13. https://code.google.com/p/webgoat/
14. https://owasp.codeplex.com
15. https://github.com/owasp/railsgoat
16. https://community.rapid7.com/docs/DOC-1875
17. https://www.pentesterlab.com/exercises
18. http://downloads.phdays.com/phdays_ibank_vm.zip
19. http://code.google.com/p/owaspbwa/wiki/ProjectSummary
20. http://www.mcafee.com/us/downloads/free-tools/hacme-bank.aspx
21. http://www.mcafee.com/us/downloads/free-tools/hacmebooks.aspx
22. http://www.mcafee.com/us/downloads/free-tools/hacmetravel.aspx

Лабораторная работа № 2
«Тестирование защищённости транспортного уровня»

Цель работы

Освоение методов и средств тестирования защищённости служб SSL/TLS.

Краткие теоретические сведения

Защита транспортного уровня веб-приложения основана на использовании протоколов семейства SSL/TLS, имеющих много механизмов, функций и параметров защиты, реализация и конфигурация которых определяет в конечном итоге уровень защищённости веб-приложения.

Несмотря на то, что в настоящее время известно много автоматизированных средств тестирования защищённости SSL/TLS (например, сервис www.ssllabs.com, программы SSLscan и SSLyze), детали их реализации, как правило, неизвестны или требуют дополнительного исследования, что иногда не позволяет полностью доверять результатам их работы. Одним из низкоуровневых и надёжных средств тестирования защищённости служб SSL/TLS является клиент OpenSSL.

Задание

Протестировать защищённость служб SSL/TLS вебсервера тестируемого приложения (см лабораторную работу №1) средствами OpenSSL.

Порядок выполнения работы

1. Установить последнюю пакет OpenSSL.
2. Выполнить базовые проверки SSL/TLS. Запустить сетевой анализатор Wireshark (имеет свободную лицензию). Выполнить тестовое подключение к серверу:

openssl s_client –connect <веб-адрес тестируемого приложения>:443

Просмотреть трассировку установки соединения в Wireshark. Определить следующие параметры: версию протокола SSL/TLS, используемый криптографический набор (cipher suite), длину открытого ключа сервера, включение механизма сжатия данных. Отправить следующий HTTP-запрос и убедиться в получении ответа от сервера:

GET / HTTP/1.1

Host: <веб-адрес тестируемого приложения>

Проверить поддержку сервером механизма «Sever Name Indication» (SNI):

openssl s_client –connect <веб-адрес тестируемого приложения>:443

–servername <веб-адрес тестируемого приложения>

Просмотреть трассировку установки соединения в Wireshark в этом случае. Поддержка расширения SNI идентифицируется путем установки

соединения с опцией SNI и без нее. Если в ответ получены различные сертификаты, то SNI поддерживается сервером.

Если указанное в опции SNI имя неизвестно, то клиент выводит сообщение об ошибке или предупреждение.

3. Идентифицировать все поддерживаемые протоколы SSL/TLS, выполнив последовательно команды:

```
# openssl s_client –connect <веб-адрес тестируемого приложения>:443
–ssl2
# openssl s_client –connect <веб-адрес тестируемого приложения>:443
–ssl3
# openssl s_client –connect <веб-адрес тестируемого приложения>:443
–tls1
# openssl s_client –connect <веб-адрес тестируемого приложения>:443
–tls1_1
# openssl s_client –connect <веб-адрес тестируемого приложения>:443
–tls1_2
```

Просмотреть трассировку сканирования. Найти отличия в структуре сетевых сообщений для разных версий протокола.

4. Идентифицировать криптографические наборы (cipher suite), поддерживаемые сервером. Для получения всех поддерживаемых клиентом криптографических наборов выполнить команду

```
# openssl ciphers -v
```

Для проверки поддержки, например, набора AES256-SHA выполнить следующую команду:

```
# openssl s_client –connect <веб-адрес тестируемого приложения>:443
–cipher AES256-SHA
```

Проверить поддержку криптографического набора, содержащего шифр RC4:

```
# openssl s_client –connect <веб-адрес тестируемого приложения>:443
–cipher RC4-SHA
```

Проверить, что при установке защищенного соединения криптографический набор выбирается в порядке, определяемом настройками сервера, а не клиента. Для этого из списка поддерживаемых сервером криптографических наборов выбрать три произвольных и выполнить команды, например:

```
# openssl s_client –connect <веб-адрес тестируемого приложения>:443
–cipher 'AES256-SHA256,AES128-SHA,DES-CBC-SHA'
# openssl s_client –connect <веб-адрес тестируемого приложения>:443
–cipher 'AES128-SHA256,AES256-SHA,DES-CBC-SHA'
# openssl s_client –connect <веб-адрес тестируемого приложения>:443
–cipher 'DES-CBC-SHA,AES128-SHA,AES256-SHA256'
```

При корректной настройке во всех случаях должен быть выбран набор AES256-SHA256.

Проверить поддержку сервером Forward Secrecy на основе DHE и ECDHE:

openssl s_client –connect
<веб-адрес тестируемого приложения>:443
–cipher 'ECDHE—RSA-AES256-SHA384'
openssl s_client –connect <веб-адрес тестируемого приложения>:443
–cipher 'DHE—RSA-AES256-SHA256'

5. Для определения поддержки сервером механизма «Session Resumption» выполнить команду:

openssl s_client –connect <веб-адрес тестируемого приложения>:443
–reconnect

или её менее информативный вариант

openssl s_client –connect <веб-адрес тестируемого приложения>:443
–reconnect | grep 'New\|Reuse'

6. Для идентификации механизма «Secure Renegotiation» выполнить команду:

openssl s_client –connect <веб-адрес тестируемого приложения>:443 |
grep 'Secure Renegotiation'

Просмотреть трассировку сканирования и убедиться в поддержке данного механизма. Поддержка механизма «Secure Renegotiation» сервером определяется по наличию расширения «renegotiation_info» в сообщение ServerHello или путём просмотра вывода команды:

openssl s_client –connect <веб-адрес тестируемого приложения>:443
–tlsextdebug

Для идентификации поддержки сервером механизма «ClientInitiated Renegotiation» необходимо подключиться к веб-серверу по SSL/TLS с помощью клиента OpenSSL

openssl s_client –connect <веб-адрес тестируемого приложения>:443

и затем отправить запрос:

HEAD / HTTP/1.1
Host: <веб-адрес тестируемого приложения> R

или

GET / HTTP/1.1
Host: <веб-адрес тестируемого приложения>
R

Если сервер не поддерживает «Client-Initiated Renegotiation», то будет выведено сообщение об ошибке. Если сервер поддерживает данный механизм, то сервер отправит клиенту снова свои сертификаты.

Поддержка механизма «Client-Initiated Renegotiation» может быть использована для реализации в отношении веб-сервера DoSатаки, так как при каждом установлении соединения сервер вынужден тратить существенно больше вычислительных ресурсов чем клиент.

Чтобы проверить возможность реализации DoS-атаки, можно проверить, сколько раз клиент может инициировать пересогласование (renegotiation) криптографических параметров:

GET / HTTP/1.1
Host: <веб-адрес тестируемого приложения>

R
R
R
R
R

7. Проверить наличие уязвимости к атаке «BEAST». Данная атака использует недостатки блочных шифров, работающих в режиме CBC, и существует во всех версиях протоколов SSL/TLS до версии TLS 1.1. Для того чтобы защититься от атаки BEAST, необходимо использовать шифр RC4 или протокол TLS версии 1.1 и старше. С другой стороны, шифр RC4 в настоящее время считается небезопасным, поэтому его использование нежелательно. Для защиты от атаки BEAST на практике предлагается два подхода:

- первый из них носит название «Строгое ослабление» (Strict mitigation) и предполагает использование протокола TLS версии 1.1 и старше со всеми клиентами, которые его поддерживают;
- второй подход называется «Приоритезация RC4» (RC4 prioritization) и заключается в повышении приоритета шифра RC4 для клиентов, поддерживающих только протоколы SSL 2.0, SSL 3.0 и TLS 1.0. Таким образом, необходимо убедиться, что клиенты SSL 3.0 или TLS 1.0, не поддерживающие шифр RC4, не смогут установить соединение:

openssl s_client –connect <веб-адрес тестируемого приложения>:443 –no_ssl2 –no_tls1_1
–no_tls1_2 –cipher 'ALL:!RC4'

или что клиенты, поддерживающие шифр RC4, установят соединение, используя его

openssl s_client –connect <веб-адрес тестируемого приложения>:443 –no_ssl2 –no_tls1_1
–no_tls1_2 –cipher 'ALL:+RC4'

8. Проверить наличие уязвимости к атаке «Heartbleed» по косвенным признакам, а также путём использования активных тестов в Metasploit Framework.

Проверить поддержку сервером протокола «Heartbeat» через OpenSSL путём выполнения команды:

openssl s_client –connect <веб-адрес тестируемого приложения>:443
–tlsextdebug

Если сервер не возвращает в сообщениях данные о расширении «Heartbeat», то он не уязвим к данной атаке.

Для того чтобы проверить, отвечает ли сервер на запросы Heartbeat, выполнить команды

openssl s_client –connect <веб-адрес тестируемого приложения>:443
–msg

Для проверки уязвимости клиента (например, веб-браузера) к Heartbleed атаке можно установить соединение с любым сервером и просмотреть трассировку соединения.

Рассмотрим вариант активного тестирования (выполнение атаки с использованием уязвимости) в среде Metasploit Framework.

Для тестирования клиента выполнить следующие команды:

msfconsole
> use auxiliary/server/openssl_heartbeat_client_memory
> show options
> run

В веб-браузере открыть ресурс Metasploit, отвечающий за тестирование на наличие уязвимости к атаке Heartbleed и просмотреть информацию о результате тестирования клиента.

Для тестирования сервера в среде Metasploit Framework выполнить следующие команды:

msfconsole
> use auxiliary/scanner/ssl/openssl_heartbleed
> show options
> set RHOSTS <веб-адрес тестируемого приложения>
> set RPORT 443
> set VERBOSE true
> run

9. Проверить наличие уязвимости к атаке «CRIME» по косвенным признакам. Данная атака основана на сжатии данных на уровне SSL/TLS. Для проверки достаточно выполнить команду:

openssl s_client –connect <веб-адрес тестируемого приложения>:443 –reconnect | grep 'Compression'

10. Проверить наличие HTTP-заголовков Strict-TransportSecurity, устанавливаемых на стороне веб-сервера. Отправить следующий HTTP-запрос к веб-приложению в программе Burp Suite

GET / HTTP/1.1
Host: <веб-адрес тестируемого приложения>
\r\n

HTTP-ответ должен содержать заголовок следующего вида:

Strict-Transport-Security: max-age=31536000;
includeSubDomains

Проверить наличие страниц со смешанным контентом (mixedcontent pages) – страниц, доступных по HTTPS, но содержащих ресурсы (картинки, скрипты JavaScript, файлы CSS, медиа-контент), доступные по протоколу HTTP. Для этого следует сконфигурировать браузер для работы с тестируемым веб-приложением через веб-прокси Burp Suite, в процессе работы с веб-приложением необходимо во вкладке HTTP History просмотреть историю и удостовериться, что все запросы к серверу отправляются только по протоколу HTTPS.

Проверить, что cookie, содержащие чувствительную информацию, имеют атрибут secure, например:

Set-Cookie: SessionId=371d2sm6cbn3d31a;path=/;secure

Проверить, что чувствительный контент не кэшируется на стороне клиента. Запрещение кэширования определяется наличием HTTP-заголовков Pragma, Cache-Control и Expires со следующими рекомендованными значениями:

Pragma: no-cache

Cache-Control: no-cache, no-store, must-revalidate, max-age=0

Expires: 0

Проверить, что приложение защищено от атаки «SSL Stripping». Для этого необходимо убедиться, что веб-приложение доступно только по протоколу HTTPS и не доступно опционально по протоколу HTTP или в веб-приложении используется заголовок Strict-Transport-Security (при условии того, что пользователь гарантированно попадёт на сайт по протоколу HTTPS).

Бонусное задание (для выполнения вне лаборатории). Выполнить тестирование SSL/TLS с использованием сервиса ssllabs.com. В веб-браузере перейти по адресу https://www.ssllabs.com/ssltest/index.html, ввести доменное имя сервера, выполнить сканирование, просмотреть и проанализировать полученные результаты. В качестве тестируемых серверов можно использовать веб-серверы различных компаний, новостных сайтов и т.п.

Выполнить тестирование нескольких веб-клиентов (например, веб-браузеров Internet Explorer, Mozilla Firefox, Google Chrome, WhiteHat Security Aviator, Яндекс Браузер, Apple Safari, Opera и т.п.). Для этого в каждом тестируемом браузере открыть страницу https://www.ssllabs.com/ssltest/index.html, выполнить сканирование, просмотреть и проанализировать результаты.

Содержание отчёта о выполнении лабораторной работы

1. Цель работы и общее задание
2. Краткое описание тестируемых веб-приложений.
3. Названия и версии средств тестирования (утилит, расширений и т.д.).
4. Полученные результаты тестирования.
5. Выводы.

Лабораторная работа № 3
«Тестирование защищенности механизма управления доступом»

Цель работы

Освоение методов и средств тестирования защищенности механизма управления доступом в веб-приложениях.

Краткие теоретические сведения

Одним из основных механизмов защиты современных веб-приложений является механизм управления доступом. Выделяют следующие этапы управления доступом:

− идентификация (установление идентификационных данных);

− аутентификация (подтвержденное установления идентификационных данных);

− авторизация (назначение прав идентификационным данным).

При входе в веб-приложение (sign in, log in) пользователь идентифицируется (сообщает свой идентификатор) и аутентифицируется (доказывает, что он именно тот пользователь, чей идентификатор был сообщен).

Большинство веб-приложений используют аутентификацию по паролю. В веб-приложениях с высоким уровнем защищённости также применяются протоколы двухфакторной аутентификации. Очевидным недостатком аутентификации по паролю является возможность использования паролей с плохими статистическими характеристиками. Хранение пароля или его передача по каналам связи в открытом или даже зашифрованном виде потенциально несёт угрозу раскрытия пароля.

Тем не менее, современные защищенные веб-приложения в большинстве случаев используют передачу пароля в зашифрованном виде с помощью протоколов семейства SSL/TLS, а хранение пароля в хешированном виде. При этом для хранения паролей пользователей рекомендуется использовать не криптографические хэш-функции общего назначения (например, SHA или MD5), а специализированные функции PBKDF2, bcrypt, scrypt и т.п.

Авторизация в веб-приложениях может быть определена как процесс проверки того, разрешён или запрещён запрос на получение доступа пользователя к ресурсу в соответствии с заданной политикой безопасности. Как правило, в веб-приложениях реализуется ролевая (RBAC) или атрибутная (ABAC) политика логического управления доступом.

Одним из методов тестирования возможности получения привилегий другого пользователя является дифференциальный анализ. Его идея заключается в идентификации всех возможных запросов и соответствующих им URL, которые может выполнить данный пользователь. Затем все полученные запросы выполняются от имени другого пользователя веб-приложения.

Механизм авторизации рекомендуется реализовывать на уровнях представления, бизнес-логики и данных веб-приложения. Уровень представления – не отображает функционал (например, формы, фреймы, ссылки, кнопки), на который пользователь не имеет прав доступа. Уровень бизнес-логики обеспечивает выполнение проверки наличия соответствующих прав доступа до выполнения запроса в веб-приложении, т.е. никакие функции не могут быть выполнены до авторизации (например, если пользователь отправляет запрос на удаление учётной записи, то веб-приложение должно убедиться, что пользователь имеет право на удаление учётной записи и не выполнять никаких функций до того, как это будет установлено). Уровень данных обеспечивает проверку наличия прав доступа пользователя к данным, а не только к функционалу обработки данных (например, пользователь, используя URL вида /delete?record=1, должен удалять только те записи в базе данных, на которые он имеет право доступа DELETE).

Задание

Протестировать защищенность механизма управления доступом исследуемого веб-приложения (см лабораторную работу №1).

Порядок выполнения работы

1. Настроить работу браузера через штатный прокси-сервер Burp Suite. В веб-браузере открыть главную страницу тестируемого веб-приложения.
2. Зарегистрироваться в веб-приложении. Получить идентификатор учётной записи и пароль доступа к веб-приложению.

Проанализировать предсказуемость идентификаторов пользователей и, если это возможно, алгоритм назначения идентификаторов.

Проанализировать реализованную в веб-приложении парольную политику. Оценить доступную сложность выбора паролей пользователями.

3. Перейти по ссылке для аутентификации в приложении.

При этом необходимо убедиться, что форма аутентификации доступна только по протоколу HTTPS. Убедиться, что вводимые пользователем логин и пароль отправляются в зашифрованном виде по протоколу HTTPS. Убедиться, что логин и пароль не отправляются с помощью HTTP-метода GET.

4. Проверить, что в веб-приложении изменены стандартные пароли для встроенных учётных записей. Проверить, что новые учётные записи создаются с различными паролями.
5. Проверить возможность идентификации пользователей веб-приложения через формы регистрации, входа и восстановления пароля.

Для этого следует ввести несуществующее имя пользователя и произвольный пароль, а затем имя существующего пользователя и произвольный, но неправильный пароль. В обоих случаях должно быть выведено одно и то же сообщение об ошибке вида «Ошибка в имени

пользователя или неверный пароль». Также оба HTTP-ответа должны совпадать с точностью до изменяемых параметров и быть получены за одно и то же время. В противном случае веб-приложение имеет скрытый канал (оракул), позволяющий идентифицировать его пользователей.

6. Проверить возможность реализации атаки подбора пароля пользователя. Ввести имя пользователя. Ввести несколько раз неправильный пароль (5-10 раз). После этого ввести правильный пароль для этой учётной записи. Ввести одинаковый пароль для разных учётных записей.

Проверить возможность доступа к веб-приложению. Блокирование учётных записей пользователя после нескольких неудачных попыток входа создает условие для реализации DoS-атаки и не должно использоваться в механизмах защиты от атак подбора паролей. Вместо этого необходимо использовать возрастающие временные задержки или средства анти-автоматизации (например, CAPTCHA).

7. Проверить, что чувствительный контент (например, страницы с введёнными номерами кредитных карт, счетов, адресов) не доступен через механизм History веб-браузера, а также не кэшируется им. Войти под учётной записью пользователя, перейти на страницу с чувствительным контентом. Ввести новые данные.

Выйти из приложения. Пользователь не должен иметь возможность выполнять новые запросы (при корректной реализации управления сессиями). Если при этом пользователю доступны ранее запрашиваемые страницы, то это означает, что серверная часть веб-приложения не запретила веб-браузеру сохранять данные в истории.

Запрещение кэширования определяется наличием HTTP-заголовков Pragma, Cache-Control и Expires со следующими рекомендованными значениями:

Pragma: no-cache

Cache-Control: no-cache, no-store, must-revalidate, max-age=0

Expires: -1

8. Запустить веб-приложение Web Goat. Ввести логин: «guest», пароль: «guest».

Перейти по ссылке Access Control Flaws → Bypass a Path Based Access Control. Изучить условия задачи. Используя FireBug (или любой аналогичный инструмент), изменить значение AccessControlMatrix.html на ../../main.jsp. Нажать кнопку «View File».

Перейти по ссылке LAB: Role Based Access Control → Stage 1. Изучить условия задачи. Войти под пользователем Tom (пароль: Tom). Можно видеть, что от пользователя скрыта кнопка «DeleteProfile», так как он не должен иметь возможности удалять учётные записи. Нажать кнопку «View Profile». В Burp Suite просмотреть запрос. Используя FireBug (или любой аналогичный инструмент), изменить HTML-размету, заменив элемент

<input type="submit" value="ViewProfile" name="action">

на элемент

<input type="submit" value="DeleteProfile" name="action">

Нажать кнопку «DeleteProfile». Просмотреть отправленный запрос в Burp Suite. Профиль пользователя будет удален.

Опционально решить задачу LAB: Role Based Access Control → Stage 2.

Перейти по ссылке LAB: Role Based Access Control → Stage 3. Изучить условия задачи. Войти под пользователем Tom (пароль: Tom). Нажать кнопку «View Profile». В Burp Suite просмотреть запрос. Можно видеть, что пользователю доступны данные своего профиля. Используя FireBug (или любой аналогичный инструмент), изменить HTML-размету, заменив элемент
<option value="105" selected="">Tom Cat (employee)</option>
на элемент
<option value="103" selected="">Tom Cat (employee)</option>

Нажать кнопку «ViewProfile». Просмотреть отправленный запрос в Burp Suite. Будут выведены данные профиля пользователя Curly Stooge.

Опционально решить задачу LAB: Role Based Access Control → Stage 4.

Перейти по ссылке «Remote admin access». Изучить условия задачи. Просмотреть подменю «Admin Functions». Перейти по ссылке WebGoat/attack?Screen=86&menu=200&admin=true. Просмотреть подменю «Admin Functions».

Содержание отчёта о выполнении лабораторной работы

1. Цель работы и общее задание
2. Краткое описание тестируемых веб-приложений.
3. Названия и версии средств тестирования (утилит, расширений и т.д.).
4. Полученные результаты тестирования.
5. Выводы.

СПИСОК ЛИТЕРАТУРЫ

1. Ковган, Н. М. Компьютерные сети учеб. пособие / Н. М. Ковган - Минск : РИПО. URL : https://www.studentlibrary.ru/book/ISBN9789855033746.html (дата обращения: 10.03.2025).
2. Магда, Ю. С. Raspberry Pi. Руководство по настройке и применению / Ю. С. Магда. - 2-е изд. - М : ДМК Пресс. URL : https://www.studentlibrary.ru/book/ISBN9785898183950.html (дата обращения: 10.03.2025).
3. Скрыпников, А. В. Защита Web-приложений : учеб. пособие / А. В. Скрыпников, Д. В. Арапов, В. В. Денисенко, Т. Д. Герасимова. - Воронеж : ВГУИТ. URL : https://www.studentlibrary.ru/book/ISBN9785000324691.html (дата обращения: 10.03.2025).