ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Петербургский государственный университет путей сообщения Императора Александра I»
(ФГБОУ ВО ПГУПС)

Петрозаводский филиал ПГУПС

ОДОБРЕНО
на заседании цикловой комиссии
протокол № 11 от 23.06.2017
Председатель цикловой комиссии:
_____ (Комиссаров)

УТВЕРЖДАЮ
Начальник УМО
_____ А.В. Калько
«23» 06 2017 г.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по организации и проведению лабораторных работ

По МДК 03.02. Безопасность функционирования информационных систем

Специальность: 09.02.02 Компьютерные сети

Разработчик:
Зав. УВЦ Капоровский В.Е.

Введение

Методическое пособие по проведению лабораторных работ по МДК 03.02. «Безопасность функционирования информационных систем» ПМ.03 «Эксплуатация объектов сетевой инфраструктуры» разработаны для студентов курса специальности 09.02.02 «Компьютерные сети» в соответствии с требованиями Федерального государственного образовательного стандарта по специальности среднего профессионального образования (далее СПО) 09.02.02 «Компьютерные сети».

Данное пособие содержит теоретические основы, описание хода работы, алгоритмы действий в процессе выполнения, решения задач, а также при необходимости контрольные вопросы и задания по проверке освоения материала.

В пособие даны руководства по следующим темам:

- Основы информационной безопасности;
- Угрозы информационной безопасности;
- Защита информационных систем;
- Основы сетевой безопасности;
- Организация безопасного доступа к локальным и глобальным сетям;
- Межсетевые экраны.

Лабораторные работы по МДК.03.02 «Безопасность функционирования информационных систем» направлена на:

- приобретение студентами профессиональных навыков и первоначального опыта в профессиональной деятельности;
- формирование основных профессиональных компетенций, соответствующих виду профессиональной деятельности (ВПД): Эксплуатация объектов сетевой инфраструктуры;
- воспитание сознательной трудовой и производственной дисциплины.

Результатом освоения МДК 03.02. «Безопасность функционирования информационных систем» является овладение обучающимися видом профессиональной деятельности (ВПД) Эксплуатация объектов сетевой инфраструктуры, в том числе профессиональными (ПК) и общими (ОК) компетенциями:

Код	**Наименование результата обучения**
ПК 3.1.	Устанавливать, настраивать, эксплуатировать и обслуживать технические и программно-аппаратные средства компьютерных сетей.
ПК 3.2.	Проводить профилактические работы на объектах сетевой инфраструктуры и рабочих станциях.
ПК 3.3.	Эксплуатация сетевых конфигураций
ПК 3.4.	Участвовать в разработке схемы послеаварийного восстановления работоспособности компьютерной сети, выполнять восстановление и резервное копирование информации.
ПК 3.5	Организовывать инвентаризацию технических средств сетевой инфраструктуры, осуществлять контроль оборудования после его ремонта
ПК 3.6	Выполнять замену расходных материалов и мелкий ремонт периферийного оборудования, определять устаревшее оборудование и программные средства сетевой инфраструктуры.
ОК 1.	Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес
ОК 2.	Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.
ОК 3.	Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.
ОК 4.	Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.

ОК 5.	Использовать информационно-коммуникационные технологии в профессиональной деятельности.
ОК 6.	Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.
ОК 7.	Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.
ОК 8.	Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.
ОК 9.	Ориентироваться в условиях частой смены технологий в профессиональной деятельности.

Правила охраны труда при проведении лабораторных работ.

1. Общие требования охраны труда.
   1.1. К работе в учебном кабинете допускаются студенты, прошедшие инструктаж по охране труда, знающие правила пожарной безопасности.
   1.2. При работе в кабинете должны соблюдаться правила поведения, расписание учебных занятий, установленный режим труда и отдыха.
   1.3. При проведении занятий возможно воздействие на студентов следующих опасных факторов:
   - нарушение осанки, искривление позвоночника, развитие близорукости при неправильном подборе мебели;
   - нарушение остроты зрения при недостаточной освещенности в кабинете;
   - поражение электрическим током при неисправном оборудовании кабинета;
   1.4. В процессе занятий студенты должны соблюдать правила личной гигиены, содержать в чистоте рабочее место.
2. Требования безопасности перед началом занятия.
   2.1. Включить полностью освещение в кабинете, убедиться в правильности работы светильников. Наименьшая освещенность в кабинете должна быть не менее 300Лк ($20ВТ/м^2$) при люминесцентных лампах.
   2.2. Убедиться в исправности электрооборудования кабинета: коммуникационные коробки выключателей и розеток не должны иметь трещин, сколов, а также оголенных контактов.
   2.3. Проверить санитарное состояние кабинета, убедиться в целостности стекол в окнах и провести сквозное проветривание кабинета.
3. Требование безопасности во время занятия.
   3.1. Используемые в кабинете демонстрационные электрические приборы должны быть исправны и иметь заземление и зануление.
4. Требования безопасности в аварийных ситуациях.
   4.1. При возникновении аварийных ситуаций немедленно эвакуировать студентов и сообщить администрации учреждения.
5. Требования безопасности по окончании занятия.
   5.1. Выключить демонстрационные электрические приборы;
   5.2. Закрыть окна и выключить свет

Перечень лабораторных работ

по МДК 03.02. Безопасность функционирования информационных систем

Лабораторные работы

1. Применение симметричных криптосистем: шифрование, дешифрование.
2. Применение асимметричных криптосистем: шифрование, дешифрование, установка и проверка ЭЦП.
3. Применение хэш-функций для алгоритмов аутентификации и защиты данных.
4. Изучение идентификации, аутентификации и авторизации в ActiveDirectory. Настройка аудита средствами групповой политики.
5. Проектирование защиты от сбоев электропитания для организации.
6. Проектирование защиты от потери данных для организации.
7. Создание точек восстановления в автоматическом и ручном режимах. Восстановление ОС из созданных точек.
8. Изучение журналов системы антивирусной защиты. Анализ обнаруженных угроз.
9. Составление таблицы разграничения доступа организации.
10. Защита файловых объектов.
11. Организация общего доступа к ресурсам файловой системы.
12. Защита трафика туннелированием SSH.
13. Изучение механизма шифрования IPSec.
14. Мониторинг трафика. Утилиты командной строки.
15. Установка, настройка и использование программных сетевых анализаторов и сканеров безопасности. Анализ уязвимостей вычислительной системы.
16. Настройка коммутатора, поддерживающего VLAN.
17. Настройка маршрутизации. Проверка сетевых соединений. Включение службы маршрутизации. Добавление маршрутов. Таблицы маршрутизации.
18. Настройка соединений виртуальных частных сетей. Внедрение политик удаленного доступа. Настройка и проверка работы службы преобразования сетевых адресов.
19. Установка, настройка и использование программных брандмауэров. Защита от сетевых атак. Имитация сетевой атаки на сетевые службы. Анализ журналов.
20. Установка настройка и использование систем обнаружения вторжений. Имитация сетевых атак. Анализ работы системы обнаружения сетевых вторжений.

Лабораторная работа №1.

Применение симметричных криптосистем: шифрование, дешифрование.

Цель: Изучить симметричные методы шифрования и научиться их применять, выполнять шифрование и дешифрование с использованием этого метода.

Теоретические сведения.

Одним из методов защиты данных от нежелательного доступа являются криптографические методы.

Открытый текст - информация, которую может быть понятна любому субъекту.

Шифрование - Процесс преобразования открытого текста с целью сделать непонятным его смысл.

В результате шифрования получается шифротекст. Процесс обратного преобразования шифротекста в открытый текст называется расшифрованием.

Криптографические методы делятся на два основных типа: симметричные (шифрование секретным ключом) и асимметричные (шифрование открытым ключом).

k – ключ шифрования, k’ – ключ расшифрования

В симметричных методах k= k’, т.е для шифрования и расшифровывания используется один и тот же секретный ключ

Криптография - совокупность методов преобразования данных, направленных на то, чтобы сделать эти данные бесполезными для злоумышленника.

Эти преобразования позволяют решить две проблемы защиты информации: обеспечение конфиденциальности – путем лишения злоумышленника возможности извлечь информацию из каналов связи; обеспечение целостности – путем лишения злоумышленника возможности изменить сообщение так, чтобы изменился ее смысл или ввести ложную информацию в канал связи.

Процесс получения открытого текста из шифротекста без знания ключа расшифрования называют дешифрованием (или взломом шифра), а науку о методах дешифрования-криптоанализом.

Раздел науки, объединяющий криптографию и криптоанализ, называется криптологией.

Симметричные методы шифрования.

Главным принципом в них является условие, что отправитель и получатель заранее знают алгоритм шифрования, а также ключ к сообщению.

К основным способам симметричного шифрования относятся:

- перестановки
- замены (подстановки)

Методы перестановки:

1. Простая перестановка
2. Одиночная перестановка по ключу
3. Двойная перестановка
4. Перестановка "Магический квадрат

1. Простая перестановка

1	2	3	4	5	6
4	2	3	5	1	6

Простая перестановка без ключа — один из самых простых методов шифрования.

Сообщение записывается в таблицу по столбцам. После того, как открытый текст записан колонками, для образования шифровки он считывается по строкам. Для использования этого шифра отправителю и получателю нужно договориться об общем ключе в виде размера таблицы.

Приезжаю сегодня-встречай на вокзале

п	а	о	с	й	о
р	ю	д	т		к

и	-	н	р	н	з
е	с	я	е	а	а
з	е		ч		л
ж	г	в	а	в	е

Ппаосйорюдт ки-нрнзесяеаазе ч лжгваве

2. Одиночная перестановка по ключу.

Он отличается от предыдущего лишь тем, что колонки таблицы переставляются по ключевому слову, фразе или набору чисел длиной в строку таблицы.

Пример.

Открытый текст: «*Приезжаю сегодня. Встречай на вокзале. Я*». Длина текста = 40

Пусть ключ К= {3,5,4,1,2} (длина ключа = 5).

Кол-во столбцов = длине ключа = 5

Кол-во строк = 40/5=8.

П		.	а	к
р	с		й	з
и	е	В		а
е	г	с	н	л
з	о	т	а	е
ж	д	р		.
а	н	е	в	
ю	я	ч	о	Я

Теперь переставляем столбцы согласно ключу. {3,5,4,1,2};3 столбик ставим на 1 место; 5 – на 2-е; 4 – на 3-е; 1 – на 4-е; 2 – на 5-е

.	к	а	П	
	з	й	р	с
В	а		и	е
с	л	н	е	г
т	е	а	з	о
р	.		ж	д
е		в	а	н
ч	Я	о	ю	я

.кап зйрсва иеслнегтеазор. жде ванчяоюя

В качестве ключа можно использовать последовательность символов. (некий пароль).

Для использования его в методе перестановки необходимо символьный ключ преобразовать.

Пусть ключом будет слово *«тайна».*

- отсортируем символы ключа в лексографическом порядке.:

1	2	3	4	5
а	а	й	н	т

- заменим символы ключа целым числом равным номеру его позиции в отсортированном ключе: {5,1,3,4,2}

3. Двойная перестановка.

Для дополнительной скрытности можно повторно шифровать сообщение, которое уже было зашифровано. Этот способ известен под названием двойная перестановка. Для этого размер второй таблицы подбирают так, чтобы длины ее строк и столбцов были другие, чем в первой таблице. Кроме того, в первой таблице можно переставлять столбцы, а во второй строки. Наконец, можно заполнять таблицу зигзагом, змейкой, по спирали или каким-то другим способом. Такие способы заполнения таблицы если и не усиливают стойкость шифра, то делают процесс шифрования гораздо более занимательным.

4.Перестановка «Магический квадрат»

Магическими квадратами называются квадратные таблицы со вписанными в их клетки последовательными натуральными числами от 1, которые дают в сумме по каждому столбцу, каждой строке и каждой диагонали одно и то же число. Подобные квадраты широко применялись для вписывания шифруемого текста по приведенной в них нумерации. Если потом

выписать содержимое таблицы по строкам, то получалась шифровка перестановкой букв. На первый взгляд кажется, будто магических квадратов очень мало.

Тем не менее, их число очень быстро возрастает с увеличением размера квадрата. число магических квадратов размером 5 х 5 около 250000.

Пример

В квадрат размером 4 на 4 вписывались числа от 1 до 16. Сумма равна 34.

16 3 2 13
5 10 11 8
9 6 7 12
4 15 14 1

Например, требуется зашифровать фразу: «*ПриезжаюСегодня*.». Буквы этой фразы вписываются последовательно в квадрат согласно записанным в них числам: позиция буквы в предложении соответствует порядковому числу. В пустые клетки ставится точка.

16	.	3	и	2	р	13	д
5	з	10	е	11	г	8	ю
9	С	6	ж	7	а	12	о
4	е	15	я	14	н	1	П

После этого шифрованный текст записывается в строку (считывание производится слева направо, построчно): *.ирдзегюСжаоеянП*

Метод замены (подстановки).

Шифром замены называется алгоритм шифрования, который производит замены каждой буквы открытого текста, на какой-либо символ шифрованного текста. Получатель сообщения расшифровывает его путем обратной замены.

В классической криптографии различают четыре разновидности шифров замены:

1. *Простая замена* (одноалфавитный шифр). Каждая буква открытого текста заменяется на один и тот же символ шифротекста.
2. *Блочная замена* – шифрование открытого текста производится блоками. Например, блоку «АБА» соответствует блок «РНР», а блоку «АББ» – «СЛЛ» и т. д.
3. *Пропорциональные замены*. Замена, аналогичная простой замене с единственным отличием: каждой букве открытого текста ставится в соответствие несколько символов шифротекста:

А→ 5, 13, 25, 57.

Б→ 7, 19, 31, 43. Ключом являются шифровальные таблицы.

4. *Многоалфавитная замена* состоит из нескольких шифров простой замены. Например, могут использоваться 5 шифров простой замены, а кокой из них применяется для шифрования конкретного символа в открытом тексте зависит от его положения в тексте.
5. *Гаммирование.*

1. Одноалфавитные шифры.

Устанавливается однозначное соответствие между каждым знаком исходного алфавита и соответствующим знаком зашифрованного текста.

Пример: азбука Морзе.

При таком методе шифрования ключом является используемая таблица замен.

Блочная замена

Такие шифры обладают более высокой надежностью шифрования.

“ - ” большой объем таблицы замен. (*спросить*?)

Пропорциональные шифры

Каждой букве открытого текста ставится в соответствие несколько символов шифротекста:

Для знаков, встречающихся часто, используется относительно большое число возможных эквивалентов. Для менее используемых исходных знаков может оказаться достаточным одного или двух эквивалентов. При шифровании замена для символа открытого текста выбирается либо случайным, либо определенным образом (например, по порядку).

Например, поставим в соответствие буквам русского языка трехзначные числа

символ	Варианты замены		
А	760	128	255
Б	121		
В	234	205	

Например для зашифровки слова автор – может быть шифр 128 234 ….

Пропорциональные шифры более сложны для вскрытия

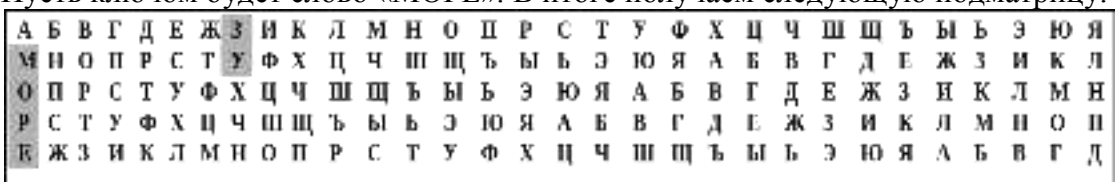
2. Многоалфавитные шифры.

Такая схема шифрования основывается на т.н. *таблице Вижинера* и называется *подстановка Вижинера*.

Таблица представляет собой квадратную матрицу с числом элементов S, где S – количество символов в алфавите. В первой строке матрицы записываются буквы в порядке очередности их в алфавите, во второй – та же последовательность букв, но с сдвигом влево на одну позицию, в третьей – с сдвигом на две позиции и т. д. Освободившиеся места справа заполняются вытесненными влево буквами,

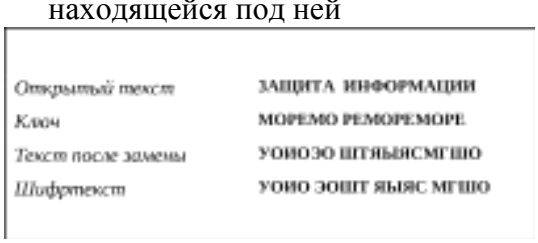
Для шифрования текста устанавливается *ключ*, *представляющий собой некоторое слово или набор букв.* Далее из полной матрицы (см. рис.5.) выбирается подматрица шифрования, включающая, например, первую строку и строки матрицы, первым символом (буквой) которой являются последовательно буквы ключа.

Пусть ключом будет слово «МОРЕ». В итоге получаем следующую подматрицу:



Процесс шифрования включает следующую последовательность действий:

1) под каждой буквой шифруемого текста записываются буквы ключа, повторяющие ключ требуемое число раз;
2) шифруемый текст по подматрице заменяется буквами, расположенными на пересечениях линий, соединяющих буквы текста первой строки подматрицы и буквы ключа, находящейся под ней



Для *дешифрования* необходимо знать ключ.(*можно попробовать*)

Условия получения абсолютно невзламываемого шифра (Шифр Вернама)

1) длина ключа будет равна длине открытого текста
2) ключ будет использован только один раз
3) ключ должен представлять собой случайную последовательность букв.

3. Гаммирование

Суть метода гаммирования состоит в том, что символы шифруемого текста последовательно складываются с символами некоторой специальной последовательности, которая называется гаммой. Отсюда и название метода – «гаммирование».

Как правило, для создания гаммы используют датчик псевдослучайной последовательности (ПСП), который на основе ключа пользователя генерирует некоторую рабочую ключевую последовательность (гамму).

Процесс шифрования заключается в следующем.

- Символы исходного текста и гаммы представляются в виде двоичного кода;
- соответствующие разряды складываются;
- полученная последовательность двоичных знаков шифрованного текста заменяется символами алфавита в соответствии с выбранным кодом.

Метод гаммирования бессилен, если криптоаналитику противника (злоумышленнику) становится известен фрагмент открытого текста и соответствующая ему криптограмма (шифртекст)

Большинство современных симметричных криптосистем относятся к разряду блочных шифров. Открытый текст разбивается на блоки и к каждому блоку применяется функция шифрования, использующая многократное повторение операций подстановки и гаммирования.

Ход работы:

1. Зашифровать произвольную фразу с помощью произвольного ключа методом «*Одиночная перестановка с ключом*»
2. Получить у товарища по группе Шифрограмму из п.1 и ключ. Провести расшифровку.
3. Зашифровать произвольную фразу с помощью произвольного ключа методом «*подстановка Вижинера*»
4. Получить у товарища по группе Шифрограмму из п.2 и ключ. Провести расшифровку.

Лабораторная работа №2.

Применение асимметричных криптосистем: шифрование, дешифрование, установка и проверка ЭЦП.

Цель: Изучить ассиметричные методы шифрования и научиться их применять, выполнять шифрование и дешифрование с использованием этого метода.

Теоретические сведения.

Одним из методов защиты данных от нежелательного доступа являются криптографические методы.

Открытый текст - информация, которую может быть понятна любому субъекту.

Шифрование - Процесс преобразования открытого текста с целью сделать непонятным его смысл.

В результате шифрования получается шифротекст. Процесс обратного преобразования шифротекста в открытый текст называется расшифрованием.

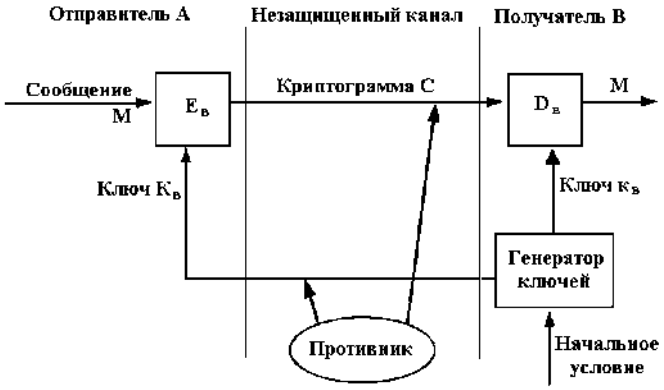
Криптографические методы делятся на два основных типа: симметричные (шифрование секретным ключом) и асимметричные (шифрование открытым ключом).

k – ключ шифрования, k’ – ключ расшифрования

В ассимитричных $k \neq k'$. Один из ключей остается секретным (secret key), а другой открытым (public key).

Главное достижение асимметричного шифрования в том, что оно позволяет людям, не имеющим договорённости о безопасности, обмениваться секретными сообщениями. Необходимость отправителю и получателю согласовывать тайный ключ по специальному защищённому каналу полностью отпала. Все коммуникации затрагивают только открытые ключи, тогда как закрытые хранятся в безопасности.

Асимметричная криптография изначально задумана как средство передачи сообщений от одного объекта к другому (а не для конфиденциального хранения информации, которое обеспечивают только симметричные алгоритмы).



Характерные особенности асимметричных криптосистем:

1. Алгоритмы шифрования и расшифрования

Ee: M - C,
Db: C - M,
являются открытыми.
2. Открытый ключ **e** и криптограмма C могут быть отправлены по незащищенным каналам, т.е. противнику известны e и C.
Защита информации в асимметричной криптосистеме основана на секретности ключа d.
У.Диффи и М.Хеллман сформулировали требования, выполнение которых обеспечивает безопасность асимметричной криптосистемы:
1. Вычисление пары ключей (e, d) получателем B на основе начального условия должно быть простым.
2. Отправитель A, зная открытый ключ e и сообщение M, может легко вычислить криптограмму по алгоритму E.

$$Ee(M) = C$$

3. Получатель B, используя секретный ключ d и криптограмму C, может легко восстановить исходное сообщение.

$$Dd(C) = M$$

$$M = Dd(Ee(P)) = M$$

4. Противник, зная открытый ключ e, при попытке вычислить секретный ключ d наталкивается на непреодолимую вычислительную проблему.
5. Противник, зная пару (e, C), при попытке вычислить исходное сообщение M наталкивается на непреодолимую вычислительную проблему.

Особенности асимметричной криптосистемы:

1. Большая продолжительность процедур шифрования/расшифрования (в 1000 раз больше чем в симметричных)
2. Использование более длинных ключей. (по криптостойкости симметричный ключ длиной 56 бита соответствует ассимитричный ключ 384 бита)

Алгоритм RSA.

В 1977 году 3 учёными из Массачусетского технологического института был разработан алгоритм шифрования, основанный на проблеме о разложении на множители. Система была названа по первым буквам их фамилий (RSA — Rivest, Shamir, Adleman).

RSA стал первым алгоритмом, пригодным и для шифрования, и для цифровой подписи.

Первым этапом любого асимметричного алгоритма является создание пары ключей : открытого и закрытого и распространение открытого ключа "по всему миру". Для алгоритма RSA этап создания ключей состоит из следующих операций :

1. Выбираются два простых (!) числа p и q
2. Вычисляется их произведение n=p*q
3. Выбирается произвольное число e (e<n), такое, что НОД(e,(p-1)(q-1))=1, то есть e должно быть взаимно простым с числом (p-1)(q-1).
4. Найдем такое d, что (e *d)mod((p-1)(q-1))=1.
5. (n; e) – открытый (public) ключ
   (n; d) – секретный (private) ключ.

Число d хранится в строжайшем секрете – это и есть закрытый ключ, который позволит читать все послания, зашифрованные с помощью пары чисел

Если бы существовали эффективные методы разложения на сомножители (факторинга), то, разложив n на сомножители (факторы) p и q, можно было бы получить секретный (private) ключ d. Таким образом надежность криптосистемы RSA основана на трудноразрешимой – практически неразрешимой – задаче разложения n на сомножители (то есть на невозможности факторинга n) так как в настоящее время эффективного способа поиска сомножителей не существует (e,n).

Шифрование: $C = M^e \{\mathrm{mod}\ n\}$
Расшифрование: $M = C^d \{\mathrm{mod}\ n\}$

Ход работы.

1. Зашифровать сообщение “АБС” с помощью алгоритма RSA, зная что символы можно представить в виде числовой последовательности. Буква A =1, B=2, C=3.

2. Расшифровать полученную криптограмму.

Лабораторная работа №3.

Применение хэш-функций для алгоритмов аутентификации и защиты данных.

Цель: Изучить применение хеш-функций для алгоритмов аутентификации и защиты данных. Познакомиться с работой утилиты OpenSSL

Теоретические сведения.

При разработке любого криптоалгоритма следует учитывать, что в половине случаев конечным пользователем системы является человек, а не автоматическая система. Это ставит вопрос о том, удобно, и вообще реально ли человеку запомнить 128-битный ключ (32 шестнадцатеричные цифры). На самом деле предел запоминаемости лежит на границе 8-12 подобных символов, а, следовательно, если мы будем заставлять пользователя оперировать именно ключом, тем самым мы практически вынудим его к записи ключа на каком-либо листке бумаги или электронном носителе, например, в текстовом файле. Это, естественно, резко снижает защищенность системы.

Для решения проблемы запоминания ключа были разработаны методы, преобразующие произносимую, осмысленную строку произвольной длины – пароль, в указанный ключ заранее заданной длины. В подавляющем большинстве случаев для этой операции используются так называемые хеш-функции (от англ. hashing – мелкая нарезка и перемешивание).

Хеширование— преобразование входного массива данных произвольной длины в выходную битовую строку фиксированной длины. Такие преобразования также называются хеш-функциями или функциями свёртки, а их результаты называют хешем, хеш-кодом или сводкой сообщения.

В общем случае однозначного соответствия между исходными данными и хеш-кодом нет в силу того, что количество значений хеш-функций меньше, чем вариантов входного массива. Существует множество массивов с разным содержимым, но дающих одинаковые хеш-коды— так называемые коллизии. Вероятность возникновения коллизий играет немаловажную роль в оценке качества хеш-функций.

Хеширование применяется например при выработке электронной подписи (на практике часто подписывается не само сообщение, а его хеш-образ).

Существует множество алгоритмов хеширования с различными свойствами (разрядность, вычислительная сложность, криптостойкость и т. п.). Выбор той или иной хеш-функции определяется спецификой решаемой задачи. Простейшими примерами хеш-функций могут служить контрольная сумма или CRC.

Общие требования к хеш-функциям:

Хорошая хеш-функция должна удовлетворять двум свойствам:

1. Быстро вычисляться;
2. Минимизировать количество коллизий

Аутентификация пользователей.

Одна из функций защиты – идентификация пользователя (сообщение системе имени). Если процедура идентификации закончилась успешно, то пользователь является законным, так как он имеет некоторый признак (идентификатор), зарегистрированный в системе.

Следующий шаг – это проверка подлинности пользователя, то есть его аутентификация: устанавливается, является ли данный пользователь тем, кем он себя объявляет.

Если аутентификация прошла успешно, и подтверждена подлинность пользователя, можно установить доступные ему ресурсы – это предоставление полномочий (авторизация).

При передаче данных, после того как соединение установлено, необходимо обеспечить требования:

а) получатель д. б. уверен в подлинности источника данных;

б) получатель д. б. уверен в подлинности передаваемых данных;

в) отправитель д. б. уверен в доставке данных получателю;

д) отправитель д. б. уверен в подлинности доставленных данных.

Для требований (а) и (б) средство защиты – цифровая подпись; для требований (в) и (д) – отправитель должен получить уведомление о вручении с помощь удостоверяющей почты (certified mail).

Средства защиты в такой процедуре **–** цифровая подпись ответного сообщения**.**

Аутентификация отправителя обеспечивается цифровой подписью.

Подпись сообщения в асимметричной криптосистеме выполняется путем шифрования на секретном ключе отправителя.

Передаваемое сообщение состоит из содержательной информации отправителя (в открытом виде) с добавленной к ней цифровой подписью.

Получатель, зная открытый ключ отправителя, может выполнить дешифрование и тем самым осуществить Аутентификацию Источника по результату сравнения принятой и вычисленной получателем цифровой подписи.

Не зная секретного ключа отправителя, невозможно создать ложное сообщение с заданной цифровой подписью.

Использование хэш-функции в технологии цифровой подписи позволяет избежать удвоения размера передаваемого сообщения, когда размер цифровой подписи будет равен размеру исходного сообщения (в символах сообщения).

Таким образом, процедура вычисления подписи сводится к последовательному вычислению значения хэш-функции от исходного сообщения и шифрованию полученного значения на секретном ключе отправителя (или дешифрованию на открытом ключе при проверке подписи).

Если отправитель и получатель знают один и тот же сеансовый ключ, аутентичность сообщения можно обеспечить, вычислив значение хэш-функции от объединения (конкатенации) передаваемого сообщения и сеансового ключа.

Результат этого вычисления называется кодом аутентификации сообщения (КАС)

КАС нужен для защиты от навязывания ложных сообщений. Для защиты от подделки КАС не передается в открытом виде, а объединяется с открытым текстом (конкатенация).

Полученный в результате объединения блок шифруется затем на сеансовом ключе.

В асимметричной криптосистеме никто, кроме отправителя, не знает секретного ключа. Это позволяет однозначно доказывать принадлежность при отказе отправителя (получателя) от ранее переданного (принятого) сообщения.

Также, получатель, не зная секретного ключа, не может подписать сообщение от лица отправителя.

Популярные стандарты хеширования.

Рассмотрим теперь то, в каких популярных стандартах могут быть представлены хэш-функции. В числе таковых — CRC. Данный алгоритм представляет собой циклический код, называемый также контрольной суммой. Данный стандарт характеризуется простотой и в то же время универсальностью — посредством него можно хешировать самый широкий спектр данных. CRC — один из самых распространенных алгоритмов, не относящихся к криптографическим.

В свою очередь, при шифровании достаточно широкое применение находят стандарты MD4 и MD5. Еще один популярный криптографический алгоритм — SHA-1. В частности, он характеризуется размером хэша 160 бит, что больше, чем у MD5 — данный стандарт поддерживает 128 бит. Есть российские стандарты, регулирующие использование хэш-функций, — ГОСТ Р 34.11-94, а также заменивший его ГОСТ Р 34.11-2012. Можно отметить, что величина хэша, предусмотренная алгоритмами, принятыми в РФ, составляет 256 бит.

Стандарты, о которых идет речь, могут быть классифицированы по различным основаниям. Например, есть те, что задействуют алгоритмы блочные и специализированные. Простота вычислений на основе стандартов первого типа часто сопровождается их невысокой скоростью. Поэтому в качестве альтернативы блочным алгоритмам могут задействоваться те, что предполагают меньший объем необходимых вычислительных операций. К быстродействующим стандартам принято относить, в частности, отмеченные выше MD4, MD5, а также SHA.

Ход работы.

Работа с OpenSSL

OpenSSL – это библиотека программного кода, написанная на языке C, которая реализует основные криптографические операции, такие как симметрическое и ассиметрическое шифрование, цифровую подпись, хэширование, итд…

Утилита OpenSSL имеет интерфейс командной строки, которая написана с использованием этой библиотеки.

1. Для выполнения работы требуется установить программу Win32OpenSSL_Light (или другую версию OpenSSL). Запустить openssl.exe (папка \bin).

Интерфейс утилиты OpenSSL имеет след. Структуру:

openssl команда [опции команды] [аргументы команды]

openssl list-standart-commands – список доступных команд

openssl rand –unknown -option – просмотр всех опций команды rand

2. Хеширование.

Сравнение файлов

Создадим текстовый файл с любым содержимым text1.txt и его копию text.txt

Для вычисления хешей используется команда openssl dgst –[аргумент] или краткая форма openssl –[аргумент]

(dgst может также выполнять манипуляции с ЭЦП).

Вычисляется хеш сообщения фиксированной длины в виде одной строки или, если указана опция -c, строки, разделённой на пары HEX чисел двоеточием. Из алгоритмов хеширования могут применяться: md2 (128 бит), md4 (128 бит), md5 (128 бит), mdc2 (128 бит), sha (160 бит), sha1 (160 бит), ripemd160 (160 бит). Результат можно сохранить в файл если использовать опцию –out

Пример:

Вычисление md5 хеш файла:

md5 -c text1.txt

MD5(text1.txt)= 81:fd:20:ff:db:06:d5:2d:c3:55:b5:7d:3f:37:ac:94

SHA1 хеш этого же файла:

sha1 file

SHA1(file)= 13f2b3abd8a7add2f3025d89593a0327a8eb83a

Задача: Вычислите хэши с помощью алгоритм хэширования md5 и sha1 2-х файлов text и text1 и проверьте что они одинаковы.

3.С помощью подкоманды passwd можно генерировать хэши паролей.

Опция -1, создает хеш с помощью алгоритма md5

Пример:

passwd -1 MySecret

1sXiKzkus$haDZ9JpVrRHBznY5OxB82.

Задание:

- Создайте небольшой произвольный текстовый файл.
- Вычислите 2 хеша с помощью разных функций хеширования.
- Вычислите хеш для произвольного пароля и сравните его с хешем пароля отличающегося на один символ

4. Симметричное шифрование/расшифрование. Различные режимы.

Иногда может возникнуть необходимость зашифровать файл без развёртывания инфраструктуры ключей и сертификатов, а пользуясь одним только паролем.

Для расшифрование требуется знать пароль и алгоритм шифрования.

Список поддерживаемых шифров можно узнать у самой программы openssl с помощью команды list-cipher-commands

Доступен широкий выбор алгоритмов шифрования которые поддерживает OpenSSL — Blowfish, Camellia, DES, RC2, RC4, RC5, IDEA, AES и другие. Помимо разнообразных алгоритмов также доступны разные режимы шифрования — ECB, CBC, CFB, OFB.

Некоторые режимы шифрования можно использовать с разной разрядностью.

Пример:

Зашифруем файл file.txt, используя алгоритм des3 и с

охраним его в

openssl des3 -in file.txt -out des3.txt

расшифруем полученный файл и сохраним его в file-d.txt

openssl des3 -d -in des3.txt -out des3-d.txt

Задание:

- Создайте небольшой произвольный текстовый файл.
- Зашифруйте шифром des в 4-х разных режимах. (des-ecb, des-cbc, des-cfb, des-ofb

- Расшифруйте их, сравните результат с исходным текстом.
- Внесите изменения в один байт в зашифрованные файлы и проанализируйте результаты расшифровки.

5. Ассиметричное шифрование/расшифрование. Генерация ключей.

Как сгенерировать секретный ключ RSA?
Использовать подкоманду genrsa:

```
# по умолчанию длина ключа 512 бит; ключ выводится на стандартный поток
openssl genrsa
# ключ 1024 бита, сохраняется в файл mykey.pem
openssl genrsa -out mykey.pem 1024
# то же, что выше, только зашифрован алгоритмом DES с помощью парольной фразы
openssl genrsa -des3 -out mykey.pem 1024  (вводится запрос на ключ и повтор ключа)
```

Как сгенерировать открытый ключ RSA?
С помощью подкоманды rsa можно создать открытую версию для закрытого ключа RSA:

```
openssl rsa -in mykey.pem –pubout –out pubkey.pem
```

Шифрации/расшифрации RSA алгоритмом
С помощью подкоманды rsautl можно провести шифрацию и дешифрацию открытым и закрытым ключом
Данная утилита имеет также возможность подписывать и проверять подпись сообщений (однако работать все равно приходится с хешем сообщения, т.к.подписывать можно только небольшой объем данных, по этой причине лучше применять dgst).
Для шифрации/дешифрации используется следующий синтаксис:

```
openssl rsautl  -in file -out file.cr  -inkey pubkey.pem –pubin -encrypt
```

(Шифрация "file" с использованием публичного ключа "pubkey.pem")

```
openssl rsautl -in file.cr -out file -keyin secretkey.pem -decrypt
```

(Дешифрация "file.cr" с использованием секретного ключа "secretkey.pem")
Подписывания сообщения секретным ключом и проверки ЭЦП публичным ключом
утилита dgst может использоваться для подписывания сообщения секретным ключом и проверки ЭЦП публичным ключом. Для этого используется следующий синтаксис:

```
openssl dgst -sign private_key -out signature –md5 file[s]
```

Подписывание file с помощью секретного ключа "private_key", используя алгоритм хеширования "hasalg" (обычно применяются sha1 или md5).

```
openssl dgst -signature signature –md5 -verify public_key file[s]
```

Проверка подписи в "file", используя публичный ключ "public_key" и ЭЦП "signature". Данная программа выводит «Verification OK» при правильной подписи или «Verification Failure» в любом другом случае.
Учтите, что ЭЦП в таком случае хранится отдельно от файла, который ею подписан (причем в каком-то кривом формате).
Задание.

- Создать закрытый и открытый ключи.
- Зашифровать открытым ключом сообщение
- Расшифровать закрытым ключом
- внести изменения в зашифрованный файл и попытаться его расшифровать.
- Создать ЭЦП к файлу.
- проверить ЭЦП
- Внести изменения в файл и проверить ЭЦП.

Лабораторная работа №4.

Изучение идентификации, аутентификации и авторизации в ActiveDirectory. Настройка аудита средствами групповой политики.

Цель: изучить идентификацию, аутентификацию и авторизацию в ActiveDirectory.

Безопасность сети – ключевая проблема, стоящая перед ИТ-службами. Решение формируется из комплекса элементов, один из них – безопасная аутентификация.

Терминология

Когда идет речь о защите информации, одним из важнейших аспектов является защита от несанкционированного доступа к ресурсам нашей сети. Разумеется, крайне важным вопросом является обеспечение процедуры безопасной аутентификации. Совершенно очевидно, что любое разграничение полномочий, настройка прав доступа на ресурсы системы имеет смысл только, если мы уверены в том, что тот, кто пытается получить доступ к нашим ресурсам, является легальным пользователем. В данной лекции мы рассмотрим некоторые аспекты обеспечения безопасности, а именно внедрение двухфакторной аутентификации в службе каталога Active Directory Domain Services (AD DS). Вопросы безопасной аутентификации являются весьма актуальными при попытке обеспечения безопасности организации в целом.

Прежде всего целесообразно разобраться с терминологией. Иногда даже сотрудники ИТ-отделов путают термины «идентификация», «аутентификация» и «авторизация». В чем тут разница? Процесс регистрации пользователя в системе состоит из трех взаимосвязанных, последовательно выполняемых процедур:

Идентификация– это процедура распознавания субъекта по его идентификатору. В процессе регистрации выполняется предъявление идентификатора системе, и она проверяет его наличие в своей базе данных. Только субъекты с известными системе идентификаторами считаются легальными.

•Аутентификация – процедура проверки подлинности, позволяющая достоверно убедиться в том, что предъявивший свой идентификатор на самом деле является именно тем, за кого он себя выдает. Для этого он должен подтвердить факт обладания некоторой информацией, которая может быть доступна только ему одному (пароль, ключ и т.п.).

•Авторизация – процедура предоставления определенных прав доступа к ресурсам системы после прохождения им процедуры аутентификации. Для каждого субъекта в системе определяется набор прав, которые он может использовать при обращении к ее ресурсам.

В данном случае под субъектом подразумевается любой участник безопасности, например, учетная запись пользователя, созданная в службе каталога AD DS.

Для того чтобы обеспечить управление и контроль над данными процедурами, дополнительно используются процессы администрирования и аудита.

• Администрирование – процесс управления доступом к ресурсам системы. Этот процесс включает в себя:

- создание идентификатора (создание учетной записи пользователя) в системе;
- управление данными пользователя, применяемыми для его аутентификации (смена пароля, издание сертификата и т.п.);
- управление правами доступа к ресурсам системы.

• Аудит – процесс контроля доступа к ресурсам системы, включающий протоколирование действий при доступе к ресурсам системы для обеспечения возможности обнаружения попыток несанкционированных действий.

Для подтверждения своей подлинности необходимо предоставить некоторую секретную информацию. Существуют различные виды такой информации, которые можно обозначить одним термином «фактор аутентификации».

• Фактор аутентификации – определенный вид информации, предоставляемый субъектом системе при его аутентификации. Данная процедура может быть реализована с использованием одного или нескольких аутентификационных факторов. Например, у пользователя может быть запрошен пароль либо потребуется предоставить отпечаток пальца.

• Однофакторная аутентификация – процесс, в котором используется только один тип аутентификационных факторов.

• Многофакторная аутентификация – процесс, в котором используется несколько факторов. Например, при регистрации пользователь должен использовать смарт-карту и пароль.

Наиболее распространено использование сочетания двух типов аутентификационных факторов. Характерным примером является работа с банкоматом. Нам требуется одновременно использовать карту с магнитной полосой и PIN-код.

Виды классификация типов факторов аутентификации согласно NCSC-TG-017

Какие же существуют факторы аутентификации?

На основе знания чего-либо (Authentication by Knowledge):

- пароль или парольная фраза;
- PIN.

На основе обладания чем-либо (Authentication by Ownership):

- физический ключ;
- карта с магнитной полосой;
- OTP-токен, генерирующий одноразовый пароль.

На основе биометрии (Authentication by Characteristic):

- отпечаток пальца;
- рисунок сетчатки глаза;
- голос.

В некоторых компаниях организуется строгий контроль доступа в помещение, то есть в определенные помещения доступ предоставляется только ограниченному числу лиц. Например, в серверную комнату может войти только администратор. Если при этом установить для всех компьютеров, находящихся в этих помещениях, строго определенные IP-адреса, то появляется возможность усиления безопасности при доступе сотрудников к ресурсам компьютерной сети. Им предоставляется доступ к определенным действиям или данным только в том случае, если они это делают в строго определенном помещении и, соответственно, с определенных компьютеров, имеющих определенные IP-адреса.

В этом случае иногда говорят об использовании четвертого типа фактора аутентификации – на основе места проведения процедуры, однако это не считается дополнительным типом факторов аутентификации, так как он не может использоваться отдельно от других. Поскольку порой весьма затруднительно обеспечить эффективную работу определенного сотрудника на строго определенном рабочем месте (компьютере), данный «фактор» нельзя выделять как дополнительный тип фактора аутентификации.

Особенности аутентификации по паролю. Риски парольной аутентификации и методы борьбы с ними

Ну что же, мы разобрались с основной терминологией и теперь поговорим о практике применения. Какой вид аутентификации используется чаще всего? Для пользователей наиболее широко используется аутентификация по секретной информации, которая неизвестна непосвященным людям. При «некомпьютерном» использовании это может быть произносимая голосом фраза или запоминаемая комбинация для замка. В случае вычислительных систем это может быть вводимый с помощью клавиатуры набор символов.

Чем длиннее пароль или идентификационная фраза, тем он более устойчив к взлому (сложнее поддается подбору, перебору или другим типам атак). Хорошим вариантом считается идентификационная фраза длиной от 25 до 100 символов. К сожалению, длинные пароли обладают другими недостатками:

• Их сложнее запомнить, а стало быть, пользователи их будут записывать, и есть высокая вероятность, что будут развешивать в виде стикера на мониторах и в других легкодоступных местах, что, несомненно, снижает безопасность. Кроме того, внедрение рекомендаций регулярной смены паролей приводит к тому, что запоминать что-то новое надо будет часто, что также приводит к усложнению работы пользователей, с которым не все готовы согласиться.

• Их медленнее набирают – соответственно их проще подсмотреть.

• Стремясь упростить для себя процесс запоминания, пользователи часто используют в качестве кодового слова осмысленные фразы (фамилии, имена, адреса, памятные даты и т.д.).

Парольная аутентификация является наиболее простым методом аутентификации с точки зрения сложности реализации (не требуется внедрять инфраструктуру удостоверяющих центров) и по умолчанию присутствует в большинстве операционных систем

Типовые атаки на пароль

Давайте рассмотрим некоторые типовые виды взлома пароля, которые используются чаще всего.

Методы перебора паролей. Атака со словарем

Злоумышленник, перебирая пароли, производит в специальном файле поиск, используя слова из большого заранее подготовленного им словаря, а также зашифровывает каждое пробное значение с помощью того же алгоритма, что и программа регистрации.

Борьба с этим видом взлома не является слишком сложной, в этом случае следует использовать сложные длинные пароли, которые содержат различные типы символов вместо осмысленных фраз. Кроме этого, можно заблокировать учетную запись при неоднократном неправильном вводе пароля. И наконец, используя аудит, выявить источник атаки.

Социотехника, угадывание, подглядывание

Злоумышленник представляется администратором и вынуждает пользователя или открыть свой пароль, или сменить его на указанный взломщиком.

Здесь метод борьбы также понятен, пользователи должны быть проинформированы о недопущении разглашения своих учетных данных кому бы то ни было. В организации должны быть разработаны административные процедуры, запрещающие разглашение паролей другим лицам при любых обстоятельствах. Следует также извещать пользователей о том, что администратор никогда не обратится к ним с таким требованием.

Более изощренный вариант такой атаки нацелен на администраторов, а не на пользователей. Злоумышленник представляется законным пользователем и просит администратора заменить пароль. Также он может представиться одним из руководителей и попросить заменить пароль, расширить полномочия и т.п.

Решение этой проблемы тоже лежит в организационной плоскости. Должна действовать корпоративная политика, согласно которой администратор меняет пароль пользователя только при условии, что он может установить его личность и передать новый вариант пользователю безопасным способом. Средства самостоятельного управления паролями могут удовлетворять обоим критериям.

Есть и другие варианты атак, например, попробовать навести справки в отделе кадров, посмотреть на столе, покопаться в мусоре. И собрав информацию о личности жертвы, попробовать угадать пароль. Иногда девичьей фамилии жены вполне достаточно для доступа к почте, размещенной на бесплатном почтовом сервере. А дальше, проанализировав переписку пользователя, взломщик может получить достаточно информации и для получения доступа к внутрикорпоративным данным. То есть злоумышленник, исходя из знаний личных данных жертвы, пытается войти в систему с помощью имени пользователя и одного или нескольких паролей, которые могли бы быть использованы. Этот способ атаки, как ни удивительно, часто оказывается эффективным и против административных учетных записей. Пароли P@ssw0rd, QWERTY123 и т.д. по-прежнему еще можно встретить у пользователей, чьей задачей как раз и является недопущение подобного в информационных сис-темах.

Для защиты от такой атаки следует использовать идентификационные фразы, не содержащие очевидных ассоциаций, например, RQ12#lm25 гораздо лучше, чем Margo200576. Ну и опять нам на помощь приходит блокировка учетной записи при неоднократном неверном вводе пароля.

Что касается подглядывания при вводе пароля, то здесь нам отчасти могут помочь административные процедуры, запрещающие вводить свои учетные данные в присутствии других лиц, и регулярная смена пароля пользователем, требования обязательной блокировки рабочей станции и т.п.

«Троянский конь»

Злоумышленник скрытно устанавливает ПО, имитирующее обычную регистрационную программу и собирающее имена пользователей и пароли при попытках пользователей войти в систему. Впрочем, такое программное обеспечение может быть установлено не только злоумышленником.

На компьютере подобные программы могут появиться и в результате заражения вирусами, или могут быть установлены самим пользователем, когда он пытается использовать какой-либо продукт нелегально, скачивая генератор серийных номеров и т.д., что в своем коде может содержать нежелательную функциональность.

Для защиты от этого вида атак следует использовать антивирусное программное обеспечение, программное обеспечение по оценке целостности файлов, ограничение на запуск несанкционированных приложений.

Принуждение

В этом случае чтобы заставить пользователя открыть свой пароль, злоумышленник использует угрозы или физическое воздействие. В некоторых системах предусматривается возможность для пользователя подать сигнал о том, что вход осуществляется под принуждением. Обычно это реализуется посредством использования специального пароля при входе в систему – пароль «вход под принуждением».

Для защиты от «троянских коней» следует использовать антивирусные средства и блокировку несанкционированного программного обеспечения. Для ограничения возможностей пользователей по внесению вирусов в информационную систему оправданы: настройка запрета на работу с внешними устройствами (CD, DVD, Flash), строгий режим работы UAC, использование отдельно стоящих интернет-киосков на базе компьютеров, не входящих в состав рабочей сети. И наконец, внедрение строгих регламентов работы, определяющих правила работы пользователей в корпоративной сети (запрет передачи своих учетных данных кому бы то ни было, запрет оставлять свои учетные данные в доступных местах, требования обязательной блокировки рабочей станции при оставлении рабочего места и т.п.). В результате мы сможем добиться уменьшение рисков, связанных с нарушением безопасности компании.

Предположим, все это сделано. Тем не менее говорить о том, что нам удалось обеспечить безопасную аутентификацию в своей системе, пока преждевременно.

Человеческий фактор – самая большая угроза

Существуют еще угрозы, справиться с которыми нам не удалось. Одна из наиболее существенных – человеческий фактор. Пользователи наших информационных систем не всегда достаточно сознательны и, несмотря на разъяснения администраторов безопасности, записывают свои учетные данные (имя пользователя и пароль) и не заботятся о секретности этой конфиденциальной информации.

Как мы можем видеть, в системе внедрены длинные и сложные пароли, и ассоциативный ряд явно не просматривается. Тем не менее пользователи нашли «эффективный» способ запоминания и хранения учетных данных… Вы видите, что в этом случае как раз и сработала особенность, о которой я говорил выше: длинные и сложные пароли записываются и могут храниться ненадлежащим образом.

Инсайдинг

Еще одна существенная угроза безопасности заключается в потенциальной возможности физического доступа злоумышленника к рабочей станции легального пользователя и передача конфиденциальной информации третьим лицам. Речь идет о ситуации, когда внутри компании существует сотрудник, похищающий информацию у своих коллег.

Разумеется, кто попало не пройдет в офис компании, однако всем известно, что самым опасным является внутренний шпион. У него уже есть физический доступ к вашей системе, и разместить клавиатурный шпион не составит труда, тем более эти устройства доступны широкому кругу лиц.

Всегда ли пользователь блокирует свою рабочую станцию, покидая свое рабочее место? Удается ли администратору информационной системы добиться, чтобы пользователю не назначались избыточные полномочия, особенно при необходимости использования старых программных продуктов? Всегда ли администратор, а особенно в небольшой компании, обладает достаточной квалификацией, чтобы внедрить рекомендации производителей программного и аппаратного обеспечения по построению безопасных информационных систем?

Таким образом, можно сделать вывод о ненадежности парольной аутентификации в принципе. Следовательно, требуется аутентификация многофакторная, при этом такого вида, чтобы пароль пользователя не набирался на клавиатуре.

Что нам может помочь?

Имеет смысл рассмотреть двухфакторную аутентификацию: 1-й фактор – обладание паролем, 2-й – знание PIN-кода. Перехват доменного пароля чреват возможностью входа, перехват PIN-кода не так опасен, так как дополнительно требуется смарт-карта.

На это можно возразить, что пользователь вполне может оставить свою карту в считывателе, а PIN-код написать на стикере, как и раньше. Однако существуют системы контроля, которые могут заблокировать оставленную карту так, как это реализовано в банкоматах

Кроме того, современные решения для двухфакторной аутентификации предполагают не только возможность аутентификации в AD. Использование смарт-карт и USB-ключей помогает и во многих других случаях, например, при доступе к публичной электронной почте, в интернет-

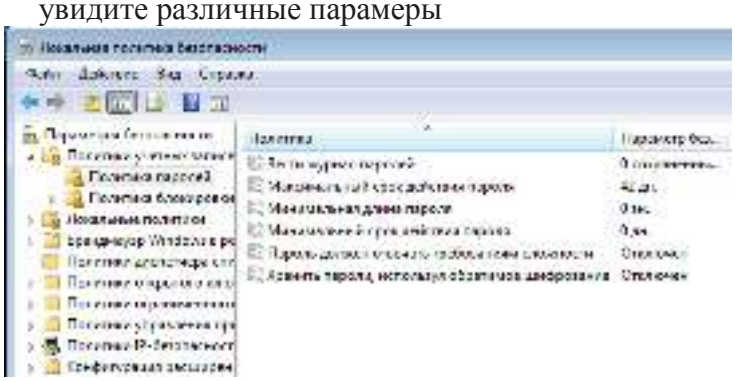
магазины, где требуется регистрация, к приложениям, имеющим свою собственную службу каталога и т.д.
Таким образом можно получить практически универсальное средство аутентификации.
Внедрение двухфакторной аутентификации на основе асимметричной криптографии в AD.
Служба каталога Active Directory поддерживает возможность аутентификации с помощью смарт-карт, начиная с Windows 2000.
Разумеется, речь идет о компьютерах в составе домена. Если же есть необходимость прибегнуть к двухфакторной аутентификации при работе в рабочей группе или при использовании более ранних версий операционных систем, то нам придется обратиться к программному обеспечению третьих фирм. Например, к SafeNet (Aladdin) eToken Network Logon 5.1.
Вход в систему может быть обеспечен как при использовании службы каталога домена, так и локальной службы каталога. При этом пароль пользователя не набирается на клавиатуре, а передается из защищенного хранилища на смарт-карте.

Ход работы.

1) Методы защиты при использовании аутентификации по паролю

Для защиты паролей от взлома следует настроить соответствующую политику. Настроим политику паролей в ОС Windows 7(в ActiveDirectory политика аналогичная) :

1. Откроем оснастку Локальная политика безопасности. Для ее запуска, откройте меню Пуск, в строке поиска введите *secpol.msc* и нажмите Ввод.
2. В открывшемся окне Локальная политика безопасности перейдите в раздел Параметры безопасности \ Параметры учетной записи \ Параметры паролей. В данном разделе вы увидите различные парамеры



3. Установите самостоятельно следующие параметры и обоснуйте:
   - минимальную длину пароля, что позволит нам избежать коротких паролей
   - Для того чтобы пользователь задавал сложные пароли, следует включить требование сложности
   - Для обеспечения регулярной смены пароля нужно задать его максимальный срок жизни
   - Ну и наконец, для того чтобы пользователь не менял свой пароль на старый путем многократной смены паролей, нужно задать минимальный срок, в течение которого пароль нельзя поменять (**Minimum password age).**

2) Для активизации аудита на изолированном компьютере (в ActiveDirectory оснастка – Групповая политика безопасности):
   1. Запустите оснастку Локальная политика безопасности (Можно так открыть Пуск › Программы › Администрирование › Локальная политика безопасности.)
   2. Последовательно раскройте узлы Локальные политики (Local Policies), Политика аудита (Audit Policy).
3. На правой панели появится список политик аудита. По умолчанию все они имеют значение Нет аудита (No Auditing). Для включения аудита следует изменить значения нужных параметров.
4. Выполните двойной щелчок на устанавливаемой политике аудита. Появится окно диалога, с помощью которого можно разрешить аудит. В группе Вести аудит следующих попыток доступа (Audit these attempts) установите флажки Успех (Success) или Отказ (Failure), или оба.
5. Нажмите кнопку ОК.

Лабораторная работа №5.

Проектирование защиты от сбоев электропитания для организации.

Цель: научиться защищать организацию от сбоев электропитания

Теоретические сведения:

Наиболее надежным средством предотвращения потерь информации при кратковременном отключении электроэнергии в настоящее время является установка источников бесперебойного питания (UPS). Задача: обеспечить питание всей локальной сети или отдельной компьютера в течение какого-то промежутка времени, достаточного для восстановления подачи напряжения, корректного завершения работы, для сохранения информации.

Умные UPS умеют следующее – компьютер получает сигнал, что UPS перешел на работу от собственных аккумуляторов и время такой автономной работы ограничено. Тогда компьютер выполняет действия по корректному завершению всех выполняющихся программ и отключается (команда SHUTDOWN). Большинство источников бесперебойного питания одновременно выполняет функции и стабилизатора напряжения, является дополнительной защитой от скачков напряжения в сети. Есть современные сетевые устройства, которые оснащены собственными дублированными системами электропитания.

Крупные организации имеют собственные аварийные электрогенераторы или резервные линии электропитания. Эти линии подключены к разным подстанциям, и при выходе из строя одной них электроснабжение осуществляется с резервной подстанции.

Большинство сбоев сетевого напряжения можно классифицировать следующим образом:

Высокочастотные помехи (радио помехами), появляются в сети в результате работы самих же потребителей. Это могут быть мощные бытовые инструменты, например электродрели, а так же различные импульсные устройства. Частота подобного сигнала может варьироваться от единиц килогерц до нескольких десятков мегагерц. Этот тип помехи - один из самых безопасных, поскольку лишь в редких случаях причиняет значительный вред.. При достаточно сильной амплитуде помех некоторая плохо защищенная техника может начать работать со сбоями, однако выход из строя маловероятен. Защита в этом случае состоит в использовании простого сетевого фильтра.

Импульсные помехи

Импульсные помехи являются гораздо более опасными. Они представляют собой короткие всплески напряжения. Продолжительность их действия небольшая и составляет несколько миллисекунд, но амплитуда напряжения может достигать десятков киловольт. Причиной могут явиться природные катаклизмы (гроза, например) или техногенные факторы (на подстанциях). Сильный импульс с большой вероятностью может привести к выходу из строя любой современной техники. Защита в этом случае состоит в использовании простого сетевого фильтра

Кратковременные провалы и всплески напряжения могут быть вызваны множеством причин, и считаются нормальным явлением для любой сети, если, конечно, время их действия и изменение амплитуды не большое. Провалы встречаются более часто, поскольку они вызываются включением мощных потребителей. Если такие проблемы долговременны или присутствуют постоянно, то это не очень хорошо влияет на работу оборудования. Максимальное долговременное отклонение от стандарта не должно превышать ±10%, т.е. напряжение может колебаться от 207 до 253, и приборы рассчитаны на это. Однако иногда допустимые 10% не выполняются, и если при отклонении в минус блок питания просто выключит аппаратуру, то при отклонении в плюс может произойти что-то менее приятное. Понятно, что в таких ситуациях необходимо использовать какие-то регуляторы напряжения. Устройства, используемые для этих целей, называются "автоматический регулятор напряжения", или AVR. Отсутствие напряжения может быть вызвано аварией или отключением по каким-то причинам. Эта ситуация неприятная, т.к. отсутствие амплитуды или ее падение до предельно низкого значения ведет к выключению техники,. В этом случае спасет только автономное электроснабжение, которое обеспечивается источниками бесперебойного питания. И самый редкий случай – сильное искажение формы сигнала или частоты. Это возможно только из-за проблем организации, осуществляющей энергоснабжение. Современные блоки питания к этому не сильно требовательны, однако, если

искажения слишком сильны, то опять же приходится прибегать к помощи источников бесперебойного питания.

Источники бесперебойного питания, фильтры, конструкции и типы.

Основная характеристика источника бесперебойного питания — это мощность, указываемая обычно в вольт-амперах (VA). Для того, чтобы узнать мощность в ваттах, можно умножить значение мощности в вольт-амперах на 0,6 (хотя, точное значение этого коэффициента зависит от конкретного оборудования). Источники бесперебойного питания различаются и другими показателями. В итоге, стоимость устройств, имеющих одинаковую мощность (например, 700 VA), но использующих различные технологии, может довольно существенно, в несколько раз, различаться.

Для того чтобы понять, какой именно ИБП вам нужен для того, чтобы защитить ваше оборудование (обычно к ИБП подключают монитор и компьютер), нужно узнать мощность, потребляемую вашим оборудованием и подбирать устройство, мощность которого немного больше. Дополнительная мощность, в любом случае, не помешает.

Основная задача источника бесперебойного питания — стабилизация параметров электрического тока, и, в случае серьезных отклонений параметров сетевого питания от нормы, перевод подключенного к ИБП оборудования на питание от батарей. Обычно срок автономного питания (если, например, речь идет об ИБП мощностью 700 VA и о питании оборудования, мощностью, не превышающей это значение) устройств от батарей ИБП не превышает 5-15 минут. Этого времени должно хватить на то, чтобы корректно завершить работу и выключить компьютер. Некоторые ИБП обладают возможностью организации обратной связи с компьютерами — если питание отключится в отсутствие пользователя, ИБП может дать компьютеру команду на отключение. И отключение будет выполнено корректно, а не аварийно.

Если по каким-то причинам вы не используете ИБП, воспользуйтесь, хотя бы, сетевым фильтром. Сетевой фильтр внешне похож на обычный удлинитель, но он может стабилизировать перепады напряжения (в определенных пределах, естественно) и фильтровать помехи. Пожалуй, сетевым фильтром можно обойтись в том случае, если там, где вы живете, проблемы с электропитанием бывают крайне редко.

Стабилизаторы.

Предназначенные для компьютеров стабилизаторы сочетают в себе функции сетевого фильтра и стабилизатора напряжения. Они не только отфильтровывают импульсные помехи, но и выдерживают стабильное напряжение на выходе (например, 230 В) при колебаниях (понижениях и повышениях) входного напряжения на 30-40%

Сетевые фильтры.

Анализ защитного оборудования наиболее целесообразно начать с рассмотрения фильтров-удлинителей. Чем же они отличаются от обычных удлинителей? По своей природе эти устройства могут защитить оборудование от импульсных и высокочастотных помех и перенапряжения. В основе импульсной защиты находится использование варисторов.

Варисторы

Варистор характеризуется нелинейной зависимостью тока от приложенного напряжения. То есть, пока напряжение не превышает некого допуска, через варистор проходит низкий ток. Как только амплитуда превышает этот установленный порог, через варистор начинает протекать огромный ток. Перед варистором находится предохранитель, который почти во всех современных конструкциях является автоматическим и многоразовым, и, как только ток превосходит номинальное значение (обычно это 10А), предохранитель размыкает цепь и отключает оборудование от сети. Такая защита действенна, но имеет несколько минусов. Во-первых, техника просто жестко выключается во время работы. Во-вторых, при сильном импульсе варисторы могут сгореть, оборудование не повредится, но фильтр со сгоревшими элементами уже не обеспечит защиты.Самый простой фильтр-удлинитель использует как минимум одни варистор и предохранитель, модели получше оборудованы как минимум тремя варисторами, которые включены треугольником между основными линиями (фаза, ноль и земля).

Источники бесперебойного питания

Источники бесперебойного питания способны защитить от большинства вышеупомянутых проблем. Изначально эти устройства создавались для компьютеров. Но в последнее время производители стали выпускать модели с евро-розетками, для того, чтобы к ним можно было подключить и бытовую технику.

Есть три типа ИБП:

- Резервный (Off-Line, Standby)
- Линейно-интерактивный (Line-Interactive)
- Непрерывного действия (online)

Резервный ИБП (Off-Line, Standby) / Back UPS

На фотографии резервный источник бесперебойного питания APC Back-UPS ES 700VA (BE700G-RS)

Самый простой и недорогой тип ИБП. Потребляет минимальное количество электроэнергии и практически бесшумный.

Состоит из аккумулятора (от которого происходит питание во время сбоя электроэнергии), зарядного устройства (питающего батарею аккумулятора) и инвертора (преобразователя напряжения с постоянного в переменный ток — 12 — 220 вольт).

Аккумулятор имеет такие технические характеристики: 12В и от 7 Ah до 9 Ah (Чем выше Ah, тем соответственно мощнее аккумулятор).

Линейно-интерактивный ИБП (Line-Interactive) / Smart UPS

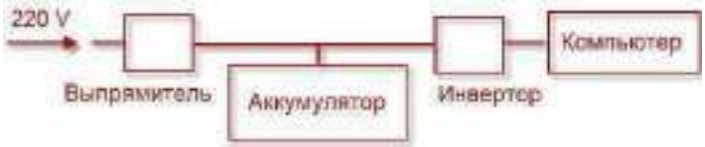
Отличается от резервного ИБП только тем, что имеет дополнительный ступенчатый стабилизатор напряжения на основе автотрансформатора, благодаря которому происходит выравнивание входящего напряжения до 220V без обращения к батарее.

ИБП этого типа обладают большой мощностью, за счет чего дороже по цене. Подходят в тех случаях, когда колебания напряжения в сети значительно превышают нормы: от 160 до 290 Вольт.

ИБП Непрерывного действия (online)

На фотографии источник бесперебойного питания непрерывного действия APC Smart-UPS RT 1000VA

Самый дорогой тип ИБП. Обеспечивает наилучшее резервное питание за счёт постоянной и высокой точности поддерживаемого выходного напряжения, а также качественной и быстрой фильтрации всех помех. Входное переменное напряжение в данном типе устройств сначала преобразуется в постоянное, а затем в переменное, после чего подается на выход.



Из недостатков этого типа источника бесперебойного питания можно отметить большое выделение тепла, требующее отвода, и шума.

Основные технические характеристики ИБП

Мощность. Указывается в Вольт-Амперах (ВА/VA) и Ваттах (Вт).
Обращая внимание на этот параметр, нужно определиться, какую технику будет защищать ИБП и рассчитать ее электропотребление.
Расчет необходимой мощности ИПБ.
Если по какой-либо причине в документации к ИПБ мощность указана только в Воль-Амперах (ВА), то пересчитать ее на Ватты можно по следующей формуле:
ВА х 0.7 = Вт
Возьмем за пример самый простой и распространенный источник бесперебойного питания APC Back-UPS ES 700VA. Его мощность составляет 700 VA. Переводим в Ватты:
700 х 0.7 = 490 Вт
Теперь мы знаем, что к этому ИПБ можно подключать максимальную нагрузку в 490 Вт (или 700 VA).
Самое время узнать мощность нагрузки компьютера. Рассчитать ее можно, сложив значения мощности каждого из комплектующих. Получить эти значения можно из технических характеристик устройств, воспользовавшись поиском по интернету или в прилагающихся документациях. Существует также большое количество онлайн-сервисов для автоматического подсчета суммарной мощности системного блока.
Нужно заметить, что не все онлайн-калькуляторы считают одинаково. Иногда разница между значениями мощности в двух разных таких сервисах может варьироваться в пределах 100 Вт. Поэтому рекомендуется приплюсовывать запас, т.е. взять по возможности максимальное значение. Не забудьте, что в сумму мощности компьютера помимо системного блока, также должны входить мощность монитора и внешней периферии (принтер, сканер, внешний жесткий диск и т.д.).
Если вы знаете количество потребляемой энергии компьютера в Ватах и желаете пересчитать эту сумму в Вольт-Амперы, то такой перерасчет можно сделать по следующей формуле:
Вт / 0.7 = ВА (VA)
Время автономной работы.
Все современные ИБП обеспечивают до 30 и более минут работы. Выбирайте этот параметр исходя опять же из пожеланий. Хотите вы продолжать работу, когда нет электроэнергии в течении долгого времени, или же вам просто необходимо некоторое время, чтобы безопасно завершить работу компьютера и сохранить данные. В некоторых моделях ИБП есть возможность подключать дополнительные батареи и соответственно расширить этим самым время автономной работы.
Розетки.
Обратите внимание на количество и тип розеток. ИБП имеющий в наличии «обычные» — «евро» розетки будет более универсален. Но тут исходите из нужд. В крайнем случае, вы всегда сможете докупить к «компьютерным» розеткам специальный переходник -сетевой фильтр.
Защита дополнительного оборудования.
Скачки напряжения могут происходить не только в электросети, но и в телефонных линиях, сетевых или телевизионных кабелях. Поэтому если необходимо, убедитесь, чтобы ИБП имел соответствующую защиту.
Вывод.
Перед выбором типа ИБП стоит понаблюдать за состоянием электросети. При частых отключениях электроэнергии лучше взять резервный ИБП. При нестабильном напряжении стоит выбрать линейно-интерактивный ИБП. Определитесь с количеством и типом розеток. И напоследок, самое главное, старайтесь выбирать источник бесперебойного питания с запасом мощности.

Резервное питание

Резервное электропитание применяют в основном для серверной.

Система электропитания серверной состоит из подсистемы гарантированного электропитания (ПГЭ), подсистемы бесперебойного электропитания (ПБЭ), подсистемы распределения электропитания (ПРЭ)

Подсистема гарантированного электропитания (ПГЭ) включает в себя три источника электроэнергии: два ввода электропитания от разных подстанций и автономную дизельную электроподстанцию (АДЭ). Каждый источник должен обеспечить мощность, равную суммарной потребляемой мощности оборудования серверной. Автомат ввода резерва (АВР) автоматически переключает в случае перебоев с электропитанием на основном.

Подсистема бесперебойного электропитания (ПБЭ) – ИБП, которых следует иметь два — основной и резервный. Каждый должен быть рассчитан на суммарную мощность всего оборудования и иметь хотя бы 30% запас мощности. Задача ИБП - обеспечить работу оборудования и подсистем на определенное рассчитанное время плюс время, необходимое для перехода на резервные линии, АДЭ и обратно.

В подсистему распределения электропитания (ПРЭ) входят распределительные щиты и кабели питания, ведущие как к оборудованию, так и к рабочим местам пользователей. Для того, чтобы при проведение ремонтных, профилактических и других работ не пришлось отключать общую систему электропитания, всех её потребители следует разделить на группы, причём, каждая группа должна иметь свой автомат защиты сети (АЗС). Помимо этого у отдельного АЗС (если установлен у отдельного потребителя) номинал его не должен превышать номинал основного АЗС группы.

К каждой стойке или телекоммуникационному шкафу должно быть подведено два кабеля от источников бесперебойного питания – основного и резервного. Внутри шкафов или стоек необходимо установить модули распределения питания.

В серверной должна быть предусмотрена подсистема технологического заземления (ПТЗ), отдельная от защитного заземления здания. Её подсоединение к заземлению здания производится непосредственно у защитных электродов, расположенных в грунте. Заземлению должны подвергаться все металлические элементы и конструкции серверной, каждый шкаф или стойка заземляются отдельным проводником.

Расчет времени автономной работы

Усредненный (приблизительный) расчет времени автономной работы по формулам расчет можно осуществить по упрощенной формуле, для этого: Емкость аккумулятора в Ампер-часах, умножаем на напряжение аккумуляторов, в вольтах, делим на постоянную нагрузку в Вт, и получаем = Количество часов непрерывной работы.

Например, телевизор, который потребляет 80В, с аккумулятором на 50 Ампер-часов будет непрерывно работать в течении 7,5 часов (50*12/80).

Ход работы.

Рассчитать время автономной работы ПК, который находится на вашем рабочем месте, если предположить, что он подключен к ИБП APC Back-UPS 500 BA.

Лабораторная работа №6.

Проектирование защиты от потери данных для организации.

Цель: построить защиту от потери данных для организации

Резервирование и архивирование

Какая из обязанностей системного администратора самая неприятная и неинтересная? По мнению многих, таковой является обеспечение непрерывного цикла резервного копирования и архивирования данных.

Немало администраторов пускают резервное копирование на самотек, производя его от случая к случаю в надежде на авось. Действительно, зачем напрягаться каждый день, если техника работает надежно. Аварии происходят не чаще одного-двух раз в год, а то и реже. Тем не менее, когда это случается (а надо сказать, авария так же неизбежна, как смена года, хотя ее и нельзя предсказать заранее), администратор горько сожалеет (хотя бы про себя), что он своевременно не провел резервное копирование, и обычно дает себе зарок со следующего понедельника осуществлять резервное копирование регулярно. Но спустя некоторое время, когда ситуация нормализуется, он забывает о тяжелых испытаниях, и все опять возвращается на круги своя.

Проблему усугубляет то обстоятельство, что большинство руководителей предприятий не отдают себе отчета в важности резервного копирования и архивирования данных. Выбить у начальства серьезное программно-аппаратное обеспечение в состоянии только системный администратор с железными нервами и мертвой хваткой. Все остальные вынуждены использовать самое простое (и, соответственно, самое неудобное) обеспечение, из-за чего у администратора возникает множество дополнительных забот, таких, например, как ручная смена картриджей или ежедневное составление расписания резервного копирования. Разумеется, это не добавляет администраторам энтузиазма для организации надлежащей схемы резервного копирования и архивирования данных.

Программное обеспечения.

Терминология, применяемые при сохранении данных.

1. Резервное копирование (backup, restore. recovery). Как следует из названия, резервное копирование предназначено для хранения информации (на внешних носителях) с тем, чтобы ее можно было восстановить при авариях или сбоях в информационных системах.
2. Архивирование (archive). Архивирование - для обеспечения долгосрочного сохранение наработанной информации. Часто такая информация уже не требуется для текущей работы, но тем не менее может понадобиться через какое то время.
3. Системы иерархического хранения данных (Hierarchical Storage Management, HSM). Внешние накопители могут быть использованы для оперативного и интерактивного хранения информации, аналогично тому, как используются винчестеры. В системах HSM медленные, но емкие внешние накопители могут выступать в качестве второго (магнитооптика) или третьего (магнитные ленты) уровня хранения. Файлы, к которым пользователи давно не обращались, переносятся (мигрируют) с винчестеров на накопители второго или третьего уровня. При обращении файл снова автоматически перемещается на винчестер.

Особенности:

При резервном копировании целью является сохранение текущего состояния системы, причем предыдущее состояние хранить совершенно необязательно. При архивировании задача состоит в долгосрочном хранении информации, чтобы данные можно было извлечь, даже если они созданы и месяц, и год назад.

В дальнейшем под терминами «хранение» и «копирование» мы будем понимать как резервное копирование, так и архивирование на внешние носители.

Резервное копирование и архивирование осуществляются в соответствии с тремя основными программными методами записи на внешние носители: полным, инкрементальным и дифференциальным.

1) При полном методе каждый раз производится копирование всего набора выбранной информации, например копируется целиком файловая система, база данных или отдельный каталог на диске. Данный метод занимает много времени при записи и ведет к большому расходу носителей. С другой стороны, в этом случае восстановление информации осуществляется быстрее, чем при любом другом методе, для этого требуется только один образ (один набор

носителей). Полное копирование является наиболее привлекательным решением при резервном копировании системной информации и служит отправной точкой для других методов.
2) Инкрементальный метод представляет собой поэтапный способ записи информации. При таком методе первая запись на носитель является полной копией. При каждой последующей записи на носитель помещаются только модифицированные файлы (т. е. те, у которых изменились содержание, атрибуты или права доступа). По истечении заданного администратором времени цикл повторяется, т. е. опять сначала делается полная копия, а затем инкрементальные копии. С точки зрения копирования на носитель данный метод является самым быстрым и ведет к минимальному расходу носителя. Однако восстановление информации занимает много времени: информацию сначала требуется восстановить с полной копии, а затем по порядку со всех последующих. Тем не менее это самый популярный метод архивирования и даже резервного копирования, поскольку восстановление/разархивирование — достаточно редкая в информационной среде процедура.
3) При дифференциальном методе первая запись на носитель также является полной копией. На последующих этапах копируются только изменения источника копирования с последнего полного бэкапа (первое копирование). Опять же, после окончания цикла вся процедура вновь начинается с полной копии.
Дифференциальная копия содержит изменения, произошедшие с последнего полного бэкапа, а инкрементальная – с предыдущего (полного или инкрементального) бэкапа. Дифференциальные копии имеют больший объем, чем инкрементальные. Однако восстановление данных происходит быстрее, поскольку каждая из дифференциальных копий может полностью заменить предыдущую. Тогда как восстановление из инкрементальной копии требует последовательного восстановления всех сделанных копий.

-Главной проблемой инкрементального и дифференциального копирования является проблема выбора надежного критерия для установления факта модификации файла - время создания/модификации файлов, размер файла или контрольная сумма содержимого файла. К сожалению, все они имеют те или иные недостатки, связанные с особенностями обработки атрибутов
-Все программы резервного копирования/архивирования можно условно разделить на три категории.
1. Системы начального уровня, включаемые в состав операционных систем. К ним можно также отнести большинство бесплатных и условно-бесплатных программ резервного копирования. Эти программы не могут похвастаться богатством функциональных возможностей и предназначены для самых тривиальных ситуаций.
2. В настоящее время на рынке доминируют системы среднего уровня, поскольку при относительно невысокой цене они обладают широкими возможностями по резервному копированию и архивированию.
3. Системы старшего уровня предназначены для резервного копирования и архивирования в сложных гетерогенных средах. При этом они поддерживают самые разнообразные аппаратные платформы, операционные системы,.
За исключением программ начального уровня, все системы резервного копирования/архивирования реализованы в архитектуре клиент-сервер (см. Рисунок 1). Серверный компонент системы резервного копирования/архивирования устанавливается на один из серверов (это может быть NetWare, Windows NT, UNIX, MVS и т. д.). К этому же серверу подключаются внешние накопители, например стримеры или библиотеки магнитных лент. Именно сервер системы резервирования выполняет реальную работу по резервному копированию и архивированию на ленты.

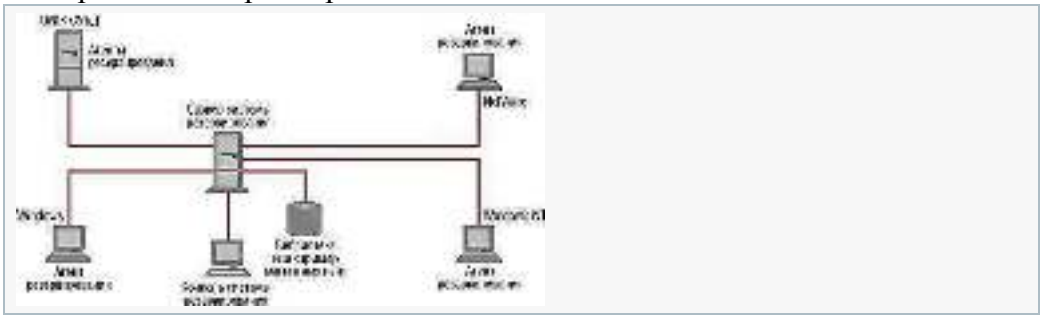


Рисунок 1. Архитектура системы резервного копирования/архивирования.

Управление системой осуществляется с консоли системы резервирования (она может располагаться как на отдельной машине, так и на сервере системы резервирования — все зависит от конкретной ситуации). На компьютерах, данные с которых подлежат резервному копированию и архивированию, размещаются программные агенты резервирования.

Один из важных вопросов, который предстоит решить, — какой вариант построения системы резервирования лучше использовать: централизованный или децентрализованный. При централизованном подходе все средства резервного копирования и архивирования сосредоточены в одном месте и управляются из единого центра. Соответственно, децентрализованное размещение предполагает, что отдельные подразделения предприятия имеют собственные и независимые средства резервирования данных.

Что лучше? К сожалению, однозначного ответа на этот вопрос нет, все зависит от конкретной ситуации. Плюсом централизованного размещения является то, что оно позволяет значительно снизить затраты на установку и эксплуатацию системы резервного копирования и архивирования. Однако централизованный вариант годится только для случаев, когда все объекты резервного копирования соединены высокопроизводительной сетью. Если счет подлежащих резервированию серверов и рабочих станций идет на десятки и сотни, то это неблагоприятно сказывается на времени, необходимом для резервирования.

Децентрализованное размещение — единственно возможный вариант при наличии филиалов, подключенных по медленным каналам связи.

Причины, приводящих к потере или порче данных.

1. Аппаратные поломки, особенно поломка дисководов. Несмотря на то что за последнее время надежность аппаратных средств значительно увеличилась, аварии и сбои тем не менее по-прежнему происходят.
2. Ошибки и сбои в операционных системах и прикладном программном обеспечении. В любом ПО всегда имеются ошибки, при определенных условиях они могут привести к порче данных.
3. Вирусы и «троянские кони». Очень распространенная причина потери информации, особенно в системах Windows.
4. Непреднамеренное уничтожение данных. По статистике — самая распространенная причина потери информации (по некоторым источникам, до 75% информации теряется в результате ошибок пользователей или обслуживающего персонала).
5. Преднамеренное уничтожение информации в результате атак злоумышленников.

Технические проблемы резервного копирования связаны не столько с самим резервным копированием, сколько с процессом восстановления данных после аварии. Как справедливо отмечает Куртис Престон, написавший одну из самых толковых книг по резервному копированию UNIX, основной недостаток литературы по резервному копированию связан с тем, что только 10% объема книг отводится процедуре восстановления данных, а остальная часть посвящена самому процессу резервного копирования. Таким образом, даже при наличии резервных копий в процессе восстановления информации администратор сталкивается с массой проблем.

Восстановление ОС

Особо тяжелый случай представляет восстановление операционной системы целиком

Мероприятия по подготовке и проведению резервного копирования

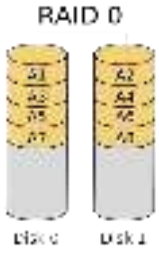
Поскольку восстановление работы компьютеров после аварии или сбоев в большинстве случаев является далеко не простой процедурой, то от администраторов требуется заранее расписать мероприятия по подготовке и проведению резервного копирования, а также по действиям персонала в момент нештатной ситуации. Необходимо соблюдать следующие правила.

1. Резервное копирование проводить на периодической основе. В случае обновления или модернизации ПО рекомендуется выполнить внеочередное копирование.
2. Резервные копии должны снабжаться сопроводительной документацией, где необходимо указать, когда и где проведено резервное копирование, какая машина подверглась резервированию, какие диски, и т. д..
3. Резервные копии и сопроводительная документация должны храниться в защищенном месте, вдали от сервера резервирования. В случае пожара или наводнения это позволит уменьшить количество проблем.

4. Особо ценные резервные копии (например, для сервера резервирования) следует дублировать.
5 Администратору необходимо продумать и оформить на бумаге действия обслуживающего персонала (план мероприятий) на случай аварии или сбоя. Этот документ надо хранить вместе с резервными копиями., Чтобы не только администратор, но и другие специалисты по ИТ могли осуществить восстановление. Тем более что спустя некоторое время сам администратор вряд ли вспомнит, как восстанавливать систему, если у него не будет под рукой плана.
6. Перед составлением плана мероприятий процедуру восстановления следует в обязательном порядке тестировать на специально выделенном компьютере, иначе толку от плана может быть мало: в жизни многое оказывается не таким простым, как это написано в документации на систему резервирования.
Копирование СУБД и пользовательских файлов.
Проблемы могут возникнуть и при резервировании и архивировании баз данных. Резервирование базы данных лучше всего проводить, когда перед резервированием БД закрывается. Такое резервирование лучше всего выполнять в ночное время, когда пользователей можно отключать от базы. Однако во многих случаях этот вариант неприемлем. Во-первых, базы данных сейчас нередко достигают в объеме сотен и тысяч гигабайт, поэтому их копирование требует слишком много времени, и даже ночи для этого не хватит.
Наиболее популярный подход к резервированию активных БД заключается в том, что в определенный момент создается полная копия базы. Все последующие обращения к базе (в момент резервирования) либо кэшируются, либо заносятся на диск с помощью переадресации. После завершения копирования эти обновления вносятся в БД. Иногда кэшируются не обновления, а старые данные. Очевидно, чтобы сохранить целостность данных, БД должна устойчиво функционировать в момент резервирования.
Определенные проблемы может доставить резервное копирование обычных пользовательских файлов, если в момент резервирования они блокированы (открыты для записи). Большинство систем резервирования нижнего уровня не могут обрабатывать их и пропускают эти файлы. Однако в настоящее время многие системы среднего и старшего уровня имеют модули, с помощью которых они могут копировать открытые файлы. Технология резервирования открытых файлов аналогична тому, как это реализовано для СУБД, т. е. за счет кэширования старых данных или обновлений.
Требования по резервированию.
Системы резервирования должны комплектоваться средствами антивирусной защиты, либо такие средства должны присутствовать на всех узлах сети, подвергаемых резервированию. На ленты могут копироваться лишь данные, свободные от вирусов. Иначе все меры борьбы с неполадками могут оказаться бесполезны. Например, сервер мог выйти из строя в результате атаки вируса, а на ленте этот вирус уже сохранен. Восстановление данных снова приведет к инициализации вируса.
Среди других очевидных требований можно назвать наличие фильтров для копирования и восстановления данных, а также поддержку основных схем ротации лент.

Защита от потери данных. RAID системы.

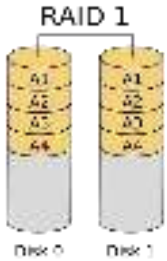
Raid 0



RAID 0 (*striping — «чередование»*) — дисковый массив из двух или более жёстких дисков без резервирования. Информация разбивается на блоки данных (A_i) фиксированной длины и записывается на оба/несколько дисков одновременно.
Этот уровень не предоставляет какой-бы то ни было защиты данных, т.е. не является отказоустойчивым, если умирает какой-либо диск в этом рэйде умирает и вся информация которая

на нём хранилась, а так же возможно и часть информации на здоровом диске. Основное назначение этого уровня - повышение скорости, для организации такого рэйда нужно минимум 2 диска. Система видит эти два диска как один, за счёт этого происходит и прирост производительности, контроллер может записывать к примеру крупный файл сразу на два диска, в результате чисто теоритически скорость чтения и записи может увеличиваться вдвое, но и надёжность также ниже чем у обычного диска. Объём жестких дисков складывается т.е. если у вас есть четыре диска по 1Tb то объединив их в массив вы получите 4Tb

Raid 1



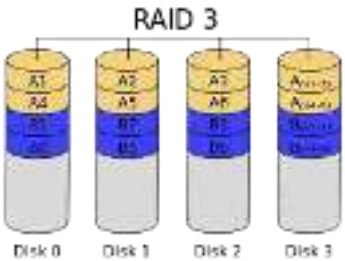
RAID 1 (*mirroring — «зеркалирование»*) — массив из двух дисков, являющихся полными копиями друг друга.

(+):Вся информация с одного жёсткого полностью дублируется на втором диске, за счёт этого достигается отказоустойчивость т.е. если выходит из строя один жёсткий диск, информация у вас не потеряется. Скорость записи обычно ниже чем у одного жесткого диска, но скорость чтения выше.

С любым уровнем RAID (кроме нулевого) рекомендуют использовать диски горячего резерва.

(-): Недостаток RAID 1 в том, что по цене двух жестких дисков пользователь фактически получает объем лишь одного.

Raid 3



В массиве RAID 3 из n дисков данные разбиваются на куски размером меньше сектора (разбиваются на байты или блоки) и распределяются по $n-1$ дискам. Ещё один диск используется для хранения блоков чётности. В RAID 2 для этой цели применялся $n-1$ диск, но большая часть информации на контрольных дисках использовалась для коррекции ошибок на лету, в то время как большинство пользователей удовлетворяет простое восстановление информации в случае поломки диска, для чего хватает информации, умещающейся на одном выделенном жёстком диске.

Отличия RAID 3 от RAID 2: невозможность коррекции ошибок на лету и меньшая избыточность.

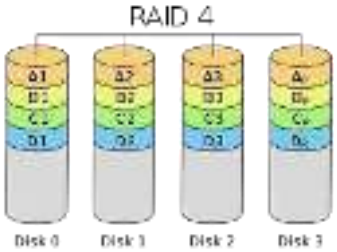
Достоинства:

- высокая скорость чтения и записи данных;
- минимальное количество дисков для создания массива равно трём.

Недостатки:

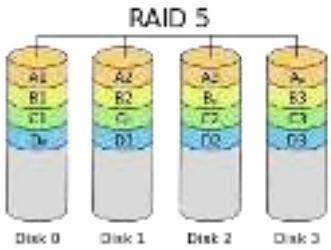
- массив этого типа хорош только для однозадачной работы с большими файлами, так как время доступа к отдельному сектору, разбитому по дискам, равно максимальному из интервалов доступа к секторам каждого из дисков. Для блоков малого размера время доступа намного больше времени чтения.
- большая нагрузка на контрольный диск, и, как следствие, его надёжность сильно падает по сравнению с дисками, хранящими данные.

Raid 4.



RAID 4 похож на RAID 3, но отличается от него тем, что данные разбиваются на блоки, а не на байты. Таким образом, удалось отчасти «победить» проблему низкой скорости передачи данных небольшого объёма. Запись же производится медленно из-за того, что чётность для блока генерируется при записи и записывается на единственный диск.

Raid 5.



Основным недостатком уровней RAID от 2-го до 4-го является невозможность производить параллельные операции записи, так как для хранения информации о чётности используется отдельный контрольный диск. RAID 5 не имеет этого недостатка. Блоки данных и контрольные суммы циклически записываются на все диски массива. Под контрольными суммами подразумевается результат операции XOR (исключающее или). *Xor* обладает особенностью, которая даёт возможность заменить любой операнд результатом, и, применив алгоритм *xor*, получить в результате недостающий операнд. Например: $a\ xor\ b = c$ (где a, b, c — три диска рейд-массива), в случае если a откажет, мы можем получить его, поставив на его место c и проведя *xor* между c и b: $c\ xor\ b = a$. Это применимо вне зависимости от количества операндов: $a\ xor\ b\ xor\ c\ xor\ d = e$. Если отказывает c тогда e встаёт на его место и проведя *xor* в результате получаем c: $a\ xor\ b\ xor\ e\ xor\ d = c$. Этот метод по сути обеспечивает отказоустойчивость 5 версии. Для хранения результата xor требуется всего 1 диск, размер которого равен размеру любого другого диска в raid.

Достоинства

RAID5 получил широкое распространение, в первую очередь, благодаря своей экономичности. Объём дискового массива RAID5 рассчитывается по формуле (n-1)*hddsize, где n — число дисков в массиве, а hddsize — размер наименьшего диска. Например, для массива из четырех дисков по 80 гигабайт общий объём будет (4 − 1) * 80 = 240 гигабайт. На запись информации на том RAID 5 тратятся дополнительные ресурсы и падает производительность, так как требуются дополнительные вычисления и операции записи, зато при чтении (по сравнению с отдельным винчестером) имеется выигрыш, потому что потоки данных с нескольких дисков массива могут обрабатываться параллельно.

Недостатки

Производительность RAID 5 заметно ниже, в особенности на операциях типа Random Write (записи в произвольном порядке), при которых производительность падает на 10-25 % от производительности RAID 0 (или RAID 10), так как требует большего количества операций с дисками Недостатки RAID 5 проявляются при выходе из строя одного из дисков — весь том переходит в критический режим (degrade), все операции записи и чтения сопровождаются дополнительными манипуляциями, резко падает производительность. При этом уровень надежности снижается до надежности RAID-0 с соответствующим количеством дисков (то есть в n раз ниже надежности одиночного диска). Если до полного восстановления массива произойдет выход из строя, или возникнет невосстановимая ошибка чтения хотя бы на еще одном диске, то массив разрушается, и данные на нем восстановлению обычными методами не подлежат. Следует также принять во внимание, что процесс RAID Reconstruction (восстановления данных

RAID за счет избыточности) после выхода из строя диска вызывает интенсивную нагрузку чтения с дисков на протяжении многих часов непрерывно, что может спровоцировать выход какого-либо из оставшихся дисков из строя в этот наименее защищенный период работы RAID, а также выявить ранее необнаруженные сбои чтения в массивах cold data (данных, к которым не обращаются при обычной работе массива, архивные и малоактивные данные), что повышает риск сбоя при восстановлении данных.
Минимальное количество используемых дисков равно трём.

Комбинированные уровни

Помимо базовых, существуют комбинированные уровни с названиями вида «RAID α+β» или «RAID αβ», что обычно означает «*RAID β, составленный из нескольких RAID α*» (иногда производители интерпретируют это по-своему).
Например:

- RAID 10 (или 1+0) — это RAID 0, составленный из нескольких (или хотя бы двух) RAID 1 (*зеркалированных* пар).
- RAID 51 — RAID 1, зеркалирующий два RAID 5 .

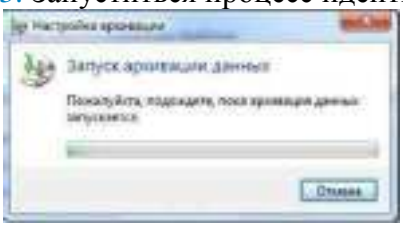
Ход работы.

1) Резервирование средствами Windows 7

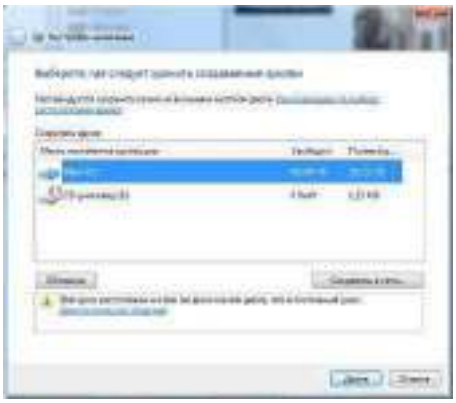
1. Для вызова главного окна настроек процесса резервирования данных кликните по кнопке “Пуск” в левом нижнем углу –> “Все программы” –> выберите пункт “Обслуживание” –> “Архивация и восстановление”.

2. В открывшемся окошке кликните по ссылке “Настроить резервное копирование” для настройки расписания архивации либо создания архива системы/файлов в ручном режиме самостоятельно.

3. Запуститься процесс идентификации подключенных носителей информации

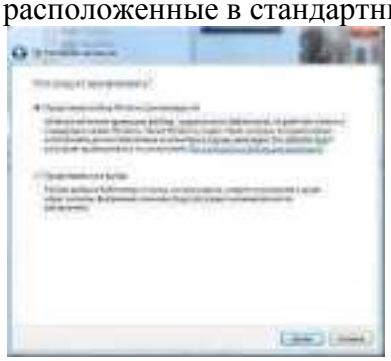


после чего откроется окошко, где необходимо будет выбрать место хранения создаваемого архива.



Если вы укажите раздел жесткого диска, расположенный на одном физическом носителе с системным разделом, то Мастер архивации вас об этом предупредит. Этот вариант не рекомендуемый, т.к. при выходе из строя жесткого диска вы потеряете и оригинал, и архивную копию (резервную копию).

4. Следующим шагом необходимо выбрать вариант резервирования “в ручном режиме” (самостоятельно), либо предоставить выбор каталогов и файлов резервирования Мастеру архивации. В последнем случае в архив попадут системные файлы и пользовательские данные, расположенные в стандартных каталогах операционной системы (вроде “Мои документы” и т.д.).

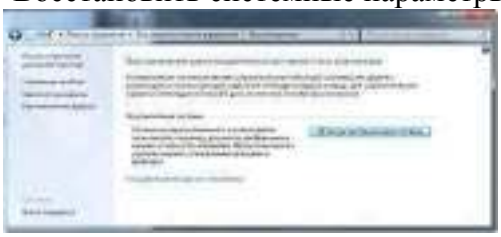


Если же вы выбрали вариант самостоятельно выбора каталогов, то в следующем окне Мастер предоставит вам возможность выбрать объекты для резервирования.

Помните, при выборе файлов и каталогов для архивации, учитывайте размер резервного хранилища.

5. В следующем окне Мастер архивации покажет общую сводку выставленных настроек резервирования и предоставит возможность изменить расписание автоматической архивации. Далее кликните по кнопке “Сохранить параметры и запустить архивацию”. Ход процесса будет отображаться в отдельном окне.

6. За восстановления системы, как нетрудно догадаться, отвечает одноименная ссылка “Восстановить системные параметры или компьютер”.



2.)Резервирование средствами Cobian backup

1.Установить программу Cobian backup.

2.Ознакомиться с функционалом данного программного продукта.

3.Выполнить инкрементальное, дифференциальное резервное копирование.

4.Сравнить с функционалом средства резервного копирования в Windows 7.

3) Сравнить уровни RAID -0,1,5 по следующим показателям – количество дисков, эффективная емкость, допустимое количество дисков вышедших из строя, надежность, скорость чтения, скорость записи. Оформить в виде таблицы.

Уровень	Количество	Эффективная	Допустимое количество	Надёжность	Скорость	Скорость

	дисков	ёмкость*	вышедших из строя дисков		чтения	записи
0	от 2	S * N	нет	очень низкая	высокая	высокая
1	2	S	1 диск	высокая	высокая	низкая
5	от 3	S * (N − 1)	1 диск	*средняя*	высокая	*средняя*

Лабораторная работа №7.

Создание точек восстановления в автоматическом и ручном режимах. Восстановление ОС из созданных точек.

Цель: Научиться создавать точки восстановления в автоматическом, ручном и с помощью планировщика задач. Выполнять восстановление из созданных точек.

Теоретические сведения.

Восстановление системы позволяет выполнить откат состояния операционной системы к одной из точек восстановления, фиксирующих состояние на момент, когда система стабильно работала. Преимуществом данной функции заключается в том, что она предоставляет возможность быстрого восстановления ("отката" состояния системы к состоянию, в котором она находилась в один из предыдущих моментов во времени) без переустановки системы, а также не подвергает риску случайного перезаписывания рабочих файлов пользователей. Возможно выполнение отката к любому из следующих типов контрольных точек и точек восстановления.

- Начальная контрольная точка (initial system checkpoint) системы создается при первом запуске компьютера с вновь установленной ОС.
- Точки восстановления для автоматических обновлений (Automatic update restore points) создаются, когда инсталлируются обновления, которые загружаются с помощью Windows Update.
- Точки восстановления при восстановлении с резервной копии (Backup recovery restore points) создаются, когда пользователь использует мастер архивации или восстановления (Backup or Restore Wizard).
- Точки восстановления при инсталляции программ (Program name installation restore points) создаются, при установке программного обеспечения.
- Точки восстановления для операции восстановления (Restore operation restore points) создаются каждый раз, когда пользователь осуществляете какое-либо восстановление.
- Системные контрольные точки (System checkpoints)- это запланированные точки восстановления, которые создаются компьютером регулярно, даже если пользователь не вносил никаких изменений в систему.
- Точки восстановления для неопознанного устройства (Unsigned device driver restore points) создаются, когда устанавливается драйвер устройства, который не был опознан или сертифицирован.
- Пользователь может создавать свои собственные точки восстановления вручную ("ручные" контрольные точки - manual checkpoints) в любой момент с помощью мастера восстановления системы (System Restore Wizard).
- Пользователь может создавать свои собственные точки восстановления по расписанию.

Количество контрольных точек восстановления, доступных в любой заданный момент времени, ограничено объемом пространства, которое выделено пользователем для работы системы восстановления. Максимальный размер пространства, которое можно выделить, составляет приблизительно 12 процентов.

В ходе процедуры восстановления происходит восстановление ОС и программ, инсталлированных на компьютере, к состоянию, в котором они находились на момент выбранной контрольной точки восстановления. Этот процесс не затрагивает личные файлы пользователя (включая сохраненные документы, сообщения электронной почты, адресную книгу, список Избранные (Favorites) и список Журнал (History) Интернет Explorer).

Все изменения, внесенные утилитой Восстановление системы (System Restore), полностью обратимы, и если пользователя не удовлетворяют результаты, то можно восстановить предыдущие настройки и выполнить все снова.

Ход работы.

Задание 1. Создание точки восстановления автоматически

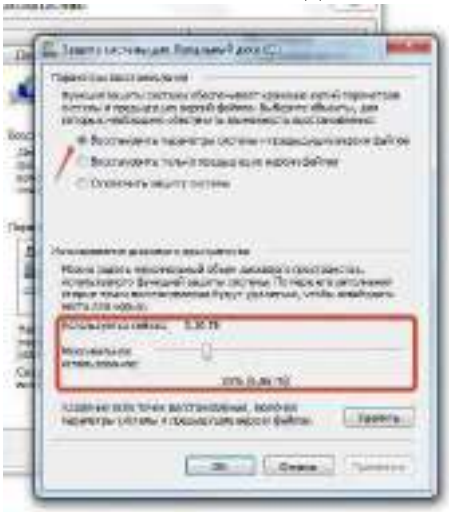
Первым делом проверьте, запущена ли в системе функция отката. Если нет, то её нужно включить для диска, на котором хранятся файлы Windows. Можно включить защиту и на других разделах винчестера, чтобы запустить процедуру создания теневых копий файлов и папок.

1. Кликом правой кнопки по значку «Компьютер» вызовите контекстное меню и запустите свойства.

2. Перейдите в раздел «Защита системы».

3. Выберите диск и нажмите «Настроить».

Настройка заключается в выборе параметров восстановления и указании объема выделяемого пространства на жестком диске. Рекомендуется выбрать первый режим работы, при котором восстановлению подлежат параметры системы и предыдущие версии файлов



.

С помощью ползунка «Максимальное использование» можно выбрать, сколько места вы готовы выделить для хранения контрольных точек. Оптимальное значение для настройки – 15%. Можно указать меньше или больше в зависимости от объема винчестера. Однако помните, что с течением времени старые метки удаляются, заменяясь новыми контрольными датами.

Задание 2. Создание точки восстановления вручную

1. Кликните ПКМ по ярлыку «Мой компьютер». Его можно найти в меню «Пуск», на рабочем столе, а также на панели быстрого доступа.
2. В открывшемся ниспадающем меню выберите пункт с названием «Свойства».
3. Откроется окно настроек нашего персонального компьютера. В левой его части находится навигационное меню. Выберите раздел «Защита системы».
4. В категории «Параметры защиты» вам необходимо выбрать жесткий диск, с которым вы хотите работать.

5. Кликните по кнопке «Создать» в самом низу окна.
6. Перед вами будет открыто всплывающее окно, в котором вы можете указать комментарий для созданного сохранения. Время и дату указывать нет необходимости, они выставляются системой автоматически.

7. Еще раз щелкните по кнопке «Создать».

Подождите некоторое время, пока новая точка восстановления будет создана.

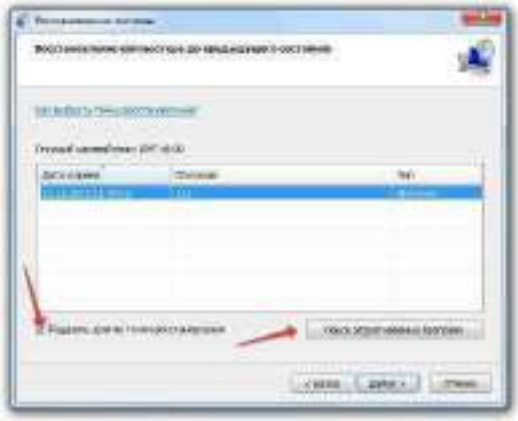
Задание 3.Восстановление ОС.

Защита системы была включена, контрольные точки созданы, возникла необходимость откатиться до предыдущего состояния. Как это сделать:

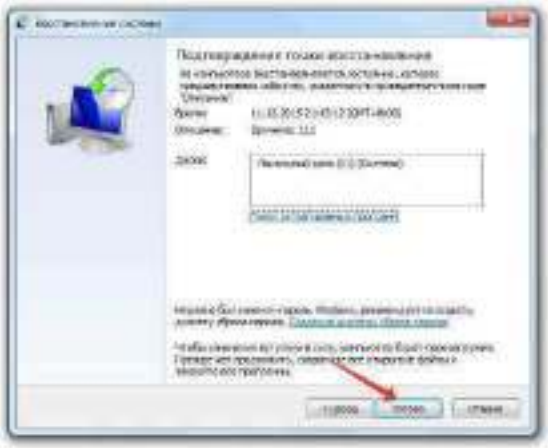
1. С помощью поиска в меню «Пуск» найдите и запустите «Восстановление системы».

2. Отметьте «Показать другие», чтобы вывести список всех сохраненных точек. Можете запустить «Поиск затрагиваемых программ» – вам будут показаны приложения, которые

будут удалены и восстановлены в результате отката.



3. Выберите контрольную дату и нажмите «Далее».
4. Проверьте правильность установленных параметров и щелкните «Готово», чтобы включить восстановление.



Во время отката компьютер перезагрузится. Прерывать процедуру восстановления нельзя, так как это чревато повреждением системных файлов. Ваши личные данные (документы, музыка и т.д.) не удалятся, но программы, установленные после контрольной даты, будут деинсталлированы.

Если Windows не загружается, то запустить восстановление можно с установочного диска. После его загрузки нужно выбрать раздел «Восстановление системы» – среди инструментов для возврата Windows в работоспособное состояние будет одноименный раздел, запуск которого предоставляет доступ к созданным контрольным точкам. Вам нужно сделать то же, что и в среде Windows — выбрать дату и откатиться до неё.

Задание 4. Предыдущие версии файлов

Откат Windows не восстанавливает удаленные файлы. Но если вы включили восстановление Windows, то система автоматически будет создавать теневые копии данных, которые можно использовать для возврата удаленной информации. Как это сделать:

1. Кликните правой кнопкой по папке, в которой были удаленные файлы. Выберите «Восстановить прежнюю версию».
2. С помощью даты изменения найдите версию папки, когда в ней были нужные файлы. Нажмите «Открыть», чтобы проверить содержимое каталога. Если вы нашли нужную версию, щелкните «Восстановить» или «Копировать». Так вы можете заново сохранить на диске те файлы, что были с него стерты.

Если вовремя сделать контрольную точку и запустить восстановление предыдущей версии, то можно обойтись без специальных программ для возврата удаленных данных. Но теневые копии не хранятся вечно – они тем объемом диска, что был выделен при настройке восстановления системы. Если поставить больший объем, то и теневые копии будут храниться дольше, поэтому лучше не жалеть места для функции защиты системы.

Задание 5. Настройка периодичности создания точек восстановления

- Создайте задачу, которая будет создавать точки восстановления со следующими условиями: 1)точки восстановления создаются каждую пятницу в 19.00 в течение месяца

Для того чтобы изменить периодичность создания точек восстановления системы проделайте следующие действия:

1. Запустите **Планировщик заданий**. Для этого откройте **Пуск —> Панель управления —> Администрирование —> Планировщик заданий**
2. В дереве слева откройте **Библиотека планировщика задач -> Microsoft -> Windows -> SystemRestore**
3. Щелкните правой кнопкой мыши в верхнем среднем окне на файл **SR** и выберите **Свойства**.
4. В открывшемся окне перейдите на вкладку **Триггеры**.
5. Нажмите **Создать**
6. И задайте необходимые Вам временные параметры.
7. нажмите **Изменить**. Для удаления созданых триггеров выберите его и нажмите **Удалить**.
8. Нажмите на ОК для принятия изменений.

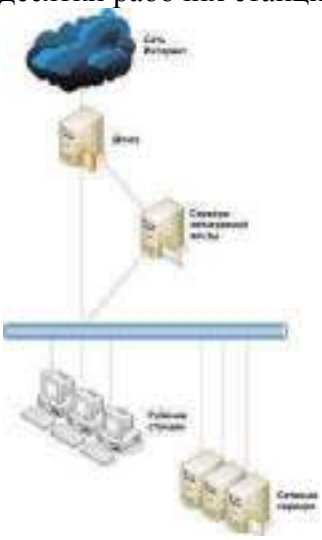
Лабораторная работа №8.

Антивирусная защита компьютерной сети. Централизованное управление антивирусной защитой

Теоретические сведения.

Основы построения локальной компьютерной сети

Большинство существующих сегодня в мире компьютерных сетей - это сети среднего масштаба, имеющие в своем составе один *шлюз*, который отвечает за *связь* с *Интернет*, один почтовый *сервер*, принимающий и пересылающий электронные письма, несколько сетевых серверов и десятки рабочих станций.



Каждый элемент локальной компьютерной сети можно охарактеризовать списком доступных для него способов обмена информацией и в соответствии с этими данными разделить все компьютеры на такие *сегменты*:

- **Рабочие станции и сетевые сервера** - обмен файлами по сети и с помощью мобильных носителей
- **Почтовые сервера** - прием и отправка электронных писем, иногда обмен файлами по сети и с помощью мобильных носителей
- **Шлюз** - организация обмена файлов между компьютерами локальной сети и более глобальной сетью, например Интернет. Дополнительно возможен обмен файлами по сети и с помощью мобильных носителей

Естественно, рабочие станции также могут принимать электронную почту, однако фактически они ее копируют либо с почтового сервера, либо со шлюза, что можно приравнять к внутрисетевому обмену данными.

Уровень защиты шлюзов

В большинстве случаев антивирусная защита на уровне шлюза играет вспомогательную роль в общей системе антивирусной безопасности сети. Это происходит потому, что задача такого антивирусного комплекса - только проверка поступающей извне информации на наличие в ней вредоносных программ. Однако даже если вирус проникнет сквозь шлюз, заразить ни один компьютер ему не удастся: его перехватит антивирус на локальной машине, а в случае инфицированного почтового сообщения - он будет остановлен еще на почтовом сервере.

Однако такой сценарий реализуется только при исправно и бесперебойно работающей системе антивирусной защиты сети, в частности на уровне защиты рабочих станций и сетевых серверов. На практике же часто встречаются сбои. Причем чем больше локальная сеть, тем больше вероятность, что такой инцидент может случиться. Несмотря на то, что распределенная система защиты рабочих станций и сетевых серверов не даст в любом случае такому вирусу распространиться далее по сети и он будет локализован на одной инфицированной машине, это все равно не очень хорошо, потому что на ней тоже могут храниться очень важные документы и при отсутствии защиты шлюза вирус сможет, например, произвести несанкционированную рассылку или позволить злоумышленнику похитить конфиденциальную информацию.

Поэтому антивирусная защита шлюза позволяет существенно увеличить надежность антивирусной защиты в целом.
Дополнительно, в случае вирусной эпидемии в Интернет, именно система защиты шлюза прореагирует и уведомит администратора первой, что позволит ему оперативно принять меры по повышению уровня защиты, например, провести срочное внеочередное обновление антивирусных баз или даже отключить отдельные особо важные или секретные компьютеры от сети.
По определению шлюз - это компьютер с установленной программой, реализующий механизм передачи данных от одной сети к другой. Обычно под этим подразумевается переход локальная сеть - сеть Интернет и все за редким обоснованным исключением компьютеры сети связываются с Интернет только через шлюз.
Основной функционал шлюза состоит в передаче запросов от одного сегмента, в другой. Например, если внутреннему пользователю нужно загрузить информацию с внешнего веб-сайта, он направляет соответствующий запрос на шлюз, который на основе этих данных запрашивает удаленный веб-сервер, получает с него требуемую информацию и передает ее пользователю. Шлюз также может работать и в обратном направлении - когда веб-сайт находится внутри локальной сети, а запрос идет от внешнего пользователя. В случае корпоративной электронной почты в роли пользователя выступает почтовый сервер.
Аналогично защите почты, на уровне защиты шлюза используется антивирусный комплекс для защиты шлюзов. Он отвечает только за проверку проходящих через него данных, а за чистоту файловой системы ответственен комплекс по защите сетевых серверов. Поэтому программный комплекс для защиты шлюзов должен содержать только фильтры для проходящих через него потоков. Обычно это HTTP, FTP и SMTP

Централизованное управление антивирусной защитой

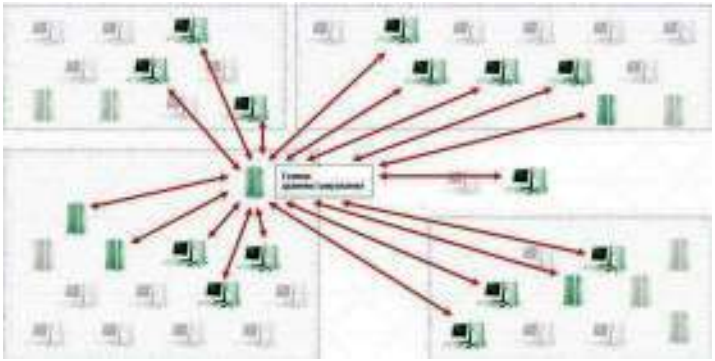
Для локальной сети, насчитывающей десятки или больше компьютеров, использование системы удаленного централизованного управления антивирусной защитой оказывается очень полезным. Она позволяет администратору не вставая из-за своего рабочего места обслуживать все входящие в его ведение рабочие станции и сетевые сервера:

1. установка и обновление антивирусных программ, а также антивирусных баз данных;
2. централизованная дистанционная установка и настройка антивирусов;
3. удаленно настраивать политики антивирусной безопасности, разрешать или запрещать пользователям самим менять какие-либо настройки;
4. автоматическое обнаружение новых рабочих станций, подключенных к корпоративной сети, с последующей автоматической установкой на эти станции антивирусных программ;
5. планирование заданий для немедленного или отложенного запуска (таких как обновление программ, антивирусной базы данных, сканирование файлов и т.п.) на любых компьютерах сети;
6. отображение в реальном времени процесса работы антивирусов на рабочих станциях и серверах сети.

Все перечисленные выше функции или многие из них реализованы в сетевых центрах управления ведущих корпоративных антивирусных продуктов, созданных компаниями Symantec (http://www.symantec.ru), NOD32 и Лаборатория Касперского .

Логическая сеть

В отношении системы удаленного администрирования употребляется термин **логической сети**, под которым понимается группа компьютеров, управление антивирусной защитой которых может вестись из одного источника.



Таким образом, если в организации, которой принадлежит локальная сеть, есть несколько связанных только через Интернет филиалов, то управление антивирусной защитой может осуществляться полностью централизованно одним администратором или же может быть разбито на отдельные сегменты. Часто в больших компаниях можно встретить смешанный вариант - вся локальная сеть разбита на связанные между собой подсети разных масштабов и за каждой из них следит отдельный человек, но существует главный администратор, который может со своего рабочего места в любой момент вмешаться в работу своих подчиненных и взять управление на себя.

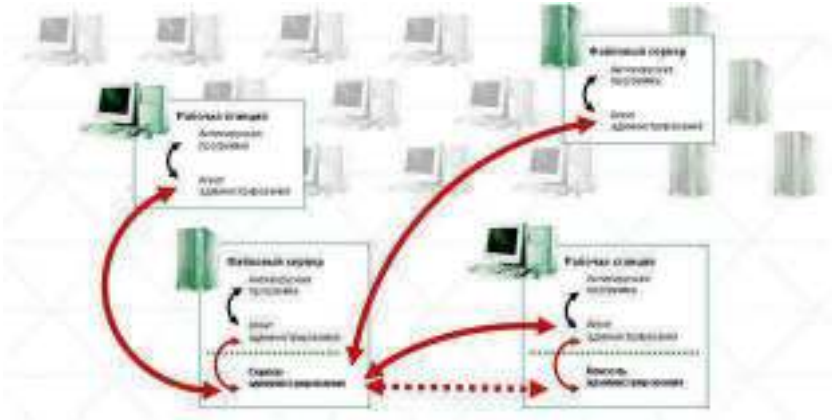
Компоненты

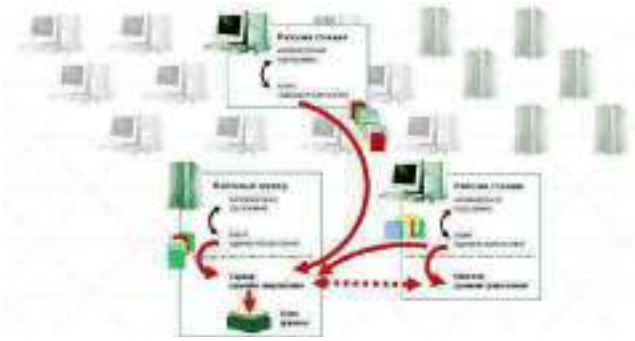
Система удаленного централизованного управления обычно состоит из таких отдельных программных компонентов:

- **Клиентской антивирусной программы**, то есть антивирусного комплекса для рабочих станций или сетевых серверов.
- **Сервера администрирования** - так называется программа, которая собирает, обрабатывает и хранит все настройки, информацию обо всех событиях и инцидентах, имевших место в сети, рассылает уведомления и отчеты. Для полноценного функционирования необходима база данных для хранения всей собранной информации. Сервер администрирования и база данных могут устанавливаться как на отдельном выделенном для этого компьютере, так и на рабочем месте администратора, на одной машине или на разных.
- **Агента администрирования**, который устанавливается на все компьютеры, входящие в логическую сеть системы антивирусной защиты. Его задача - обеспечить связь клиентской программы с сервером администрирования и оперативно передать ему информацию о состоянии антивирусной защиты на этой машине, получить новые антивирусные базы или другие указания и команды.
- **Консоли администрирования**, устанавливаемой на рабочем месте администратора. Это небольшая программа, которая позволяет в приятном и удобном виде вывести данные с сервера администрирования, на их основе построить графики и диаграммы, создать отчеты, произвести настройку клиентских компьютеров, удаленно запустить проверку или обновить антивирусные базы одновременно на нескольких машинах. Возможности той или иной консоли полностью зависят от заложенных в нее фирмой-производителем функций.

Ход работы

1) Нарисуйте схему взаимодействия между 2 рабочими станциями (на одной из которых будет сидеть администратор), 2 файловыми серверами (один из которых, будет обрабатывать информацию обо всех событиях (антивирусной защите)), а другой будет хранить всю собранную информацию). Укажите все компоненты Системы удаленного централизованного управления, которые будут храниться на каждом из них. Укажите стрелками взаимосвязь между ними.





2) Ознакомиться с основными параметрами настройки программы Антивируса Касперского 6.0 и настроить ее так, чтобы она обнаруживала тестовый вирус.

Во время работы в системной панели операционной системы появляется иконка антивируса.

Для ознакомления с интерфейсом пользователя нужно будет поочередно вызвать четыре окна интерфейса Антивируса Касперского 6.0.

Рис. 1. Главное окно программы «Антивирус Касперского 6.0»

В верхней правой части окна размещено две ссылки: **Настройка** и **Справка**. Первая используется для настройки антивируса, вторая - для вывода справочной системы.

Рис. 2. Окно настроек программы

Антивирусный монитор обеспечивает защиту в режиме реального времени, т. е. постоянно проверяет файлы, которым происходит обращение. В терминах «Антивируса Касперского 6.0» такая функциональность носит название «защита» и делится на защиту файловой системы, проверку электронной почты (протоколы SMTP, POP3, IMAP), веб-антивирус (проверка HTTP трафика), проактивную защиту (противостояние неизвестным вирусам, контроль запуска программ, обращений к реестру Windows).

Рисунок 3. Окно настроек файлового антивируса

Антивирусный сканер (в терминах «Антивируса Касперского 6.0» - поиск вирусов) выполняет сканирование ресурсов компьютера в целях поиска вирусов. Сканирование может быть запущено пользователем вручную или по заранее установленному расписанию.

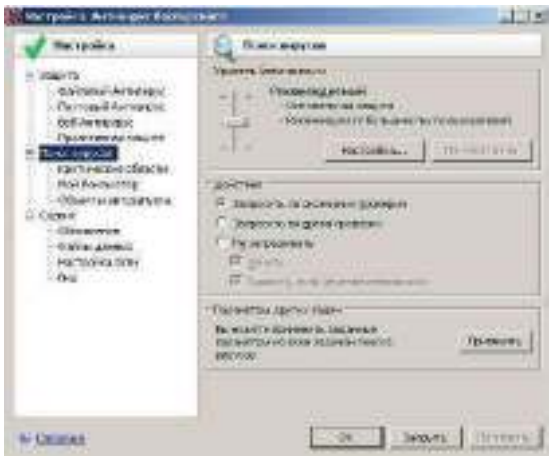


Рис. 4. Окно настроек поиска вирусов

В узле «Сервис» располагаются средства настройки обновления антивирусных баз, ведения файлов отчетов, параметров уведомлений, настройки сети и внешнего вида программы.

Рис. 5. Окно настроек сервисных функций

Использование тестового вируса EICAR

Тестовый вирус EICAR (European Institute for Computer Antivirus Research) разработан Европейским институтом компьютерных антивирусных исследований.

EICAR – это небольшой 68 байтный файл, который при запуске на незащищенном компьютере вызывает показ уведомления "EICAR-STANDARD-ANTIVIRUS-TEST-FILE!". Иных, свойственных вирусам проявлений он не несет. Однако если на компьютере стоит и исправно работает антивирус, EICAR будет заблокирован. Это происходит потому, что все ведущие производители антивирусных программ договорились между собой - считать EICAR вирусом и применять к нему все правила и действия, применяемые к настоящим вредоносным программам.

Для создания антивируса необходимо открыть текстовый редактор и ввести следующую строку символов:

X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*

После этого следует сохранить файл с расширением .com.

Для более подробного тестирования можно применять другие расширения. Например, если указать .txt, можно проверить проверяются ли текстовые файлы. Для проверки будут ли обнаруживаться вирусы в архивах, EICAR можно заархивировать.

1. Модификация тестового вируса EICAR

Суть EICAR такова, что он оказывается неизлечимым. Это происходит потому, что антивирус идентифицирует EICAR как вирус по наличию в нем упомянутых 68 символов. Если их удалить - то от файла ничего не останется. Следовательно, с помощью EICAR можно тестировать только основную функцию антивируса - обнаружение.

2 Создать файл CURE-EICAR. Файл создается в текстовом редакторе путем добавления в начало вируса символов “CURE-” и сохранения файла с расширением .com. Обнаружив такой файл антивирус «вылечит» его, сократив размер файла до 4 байт (символы «CURE»).

Рис. 6. Модификация вируса CURE-EICAR

3 Создать файл DELE-EICAR. Файл создается в текстовом редакторе путем добавления в начало вируса символов “DELE-” и сохранения файла с расширением .com. Обнаружив такой

файл, антивирус определяет его как неизлечимый или троянскую программу и удаляет. По результатам проверки файл должен остаться только в резервном хранилище.

Рис. 7. Модификация вируса DELE-EICAR

4 Создать файл CORR-EICAR. Файл создается в текстовом редакторе путем добавления в начало вируса символов “CORR-” и сохранения файла с расширением .com. Обнаружив такой файл, антивирус определяет его как файл с поврежденной структурой, вследствие чего проверить его на наличие вирусов невозможно. Такой файл признается условно чистым.

Рис. 8. Модификация вируса CORR-EICAR

5 Создать файл ERRO-EICAR. Файл создается в текстовом редакторе путем добавления в начало вируса символов “ERRO-” и сохранения файла с расширением .com. При сканировании такого файла, антивирус обнаружит ошибку при анализе его содержимого (например, при нарушении целостности при проверке многотомного архива). Такой файл признается условно чистым.

Рис. 9. Модификация вируса ERRO-EICAR

6 Создать файл SUSP-EICAR. Файл создается в текстовом редакторе путем добавления в начало вируса символов “SUSP-” и сохранения файла с расширением .com. При сканировании такого файла антивирус считает его подозрительным, а именно зараженным неизвестным вирусом. Такой файл должен быть помещен на карантин или удален.

Рис. 10. Модификация вируса SUSP-EICAR

1.2.8 Создать файл WARN-EICAR. Файл создается в текстовом редакторе путем добавления в начало вируса символов “WARN-” и сохранения файла с расширением .com. Такой файл также признается подозрительным, но не неизвестным вирусом, а модификацией известного.

Сохранить отчет с результатами работы антивируса в текстовый файл. Для этого в главном окне программы выбрать раздел «Защита» и в нем контейнер «Статистика»

Лабораторная работа №9.
Составление таблицы разграничения доступа организации.

Цель: Составление таблицы разграничения доступа организации

Теоретические сведения.
Многие системные администраторы и специалисты по безопасности сталкиваются с трудностями при настройке систем ограничения доступа в своих «вотчинах». Это может быть связано со многими причинами, например: недостаток опыта у системного администратора, недостаток документации, нечёткая постановка задачи заказчиком . Попытаемся разработать, уточнить и быстро внедрить систему разграничения доступа на предприятии. Ее называют «матрицей доступа», так как основной его инструмент представляет собой двухмерную таблицу.

Ресурсы и мандаты
То, к чему тот или иной пользователь может получить доступ, мы назовём ***ресурсом***. Ресурсом может являться как область хранилища (например, каталог или файл на сервере), так и некая совокупность ресурсов в сети Интернет (например, «социальные сети», «видеохостинги»). Отдельным видом ресурсов является также право полного доступа к областям хранилища или ресурсам сети. В матрице доступа в рамках этой статьи ресурсы будут представлять собой столбцы таблицы.
Мандатом условимся называть некий признак, имеющийся у пользователя или группы, на основании которого пользователь получает доступ к ресурсам. Некоторым образом понятие мандата пересекается с понятием прав доступа для группы пользователя, что и можно использовать при внедрении спланированной политики. В рамках этой статьи мандаты будут представлять собой строки таблицы.
Мандат — понятие разрешительного типа, то есть он не может являться запретом на доступ, а только лишь в некоторых случаях частичным разрешением (например, разрешение на чтение файла).

Матрица доступа
Как понятно из вышеизложенного, она представляет собой таблицу, в заголовках столбцов которой перечислены ресурсы, а в заголовках строк — мандаты. На пересечении столбца и строки мы ставим условный знак, определяющий тип доступа (если таковое понятие применимо). Пустая клетка означает отсутствие доступа. В рамках данной статьи условимся о следующих знаках:•
+ — наличие доступа, если бессмысленно говорить о его типе (например, в случае доступа в Интернет), а также полный доступ (чтение, запись, создание файла)
•R — доступ только на чтение
•M — доступ на чтение и запись, без возможности создания нового файла
Выстроив таким образом таблицу, мы сопоставляем мандаты с ресурсами и получаем эскиз политики безопасности.
Следующий этап — **внедрение**. Для этого следует мандаты интерпретировать в сущности используемой вами платформы — то есть расставить соответствующие права группам, настроить объекты групповых политик (если речь идет об Active Directory), создать списки доступа на маршрутизаторах, разнести пользователей по группам, присвоить им IP-адреса из соответствующих диапазонов и так далее. Конкретные шаги зависят от используемых платформ, оборудования, опыта системного администратора и т.п.

Ход работы:

Задача1: Составить матрицу доступа.
Условия: В компании имеются отделы: продаж, сметный, производственный. У отдела продаж свои документы на сервере. Их должны иметь право просматривать и редактировать бухгалтерия (но не создавать новые файлы) и только просматривать сметный отдел. У сметчиков тоже, их должна иметь право смотреть бухгалтерия. Производственники — аналогично. Руководству надо

видеть всё. И опять же — своя папка на сервере, чтоб никто не видел. Что касается интернета — продажникам придётся разрешить всё, кроме видео. Руководству тоже. Сметчикам и инженерам на производстве надо запретить «одноклассники», «вконтакты» и прочую дребедень, видео. Хотя вот этому инженеру видео надо разрешить — он там какие-то технологические процессы смотрит. На кассе интернета вообще быть не должно. Бухгалтерии что-то запрещать — себе дороже выйдет, но стажёру тоже зарезать всё баловство.

Анализ и рассуждения

О мандатах. Мы можем явно выделить мандаты отделов, а также явно добавляется руководство. С бухгалтерией сложнее — там есть стажёр и не стажёры. Впрочем, для инженера в производственном отделе тоже надо предусмотреть мандат. Ещё неявно объявляется сам системный администратор. Для ограниченного доступа лучше выбирать низший общий уровень, а потом избирательно повышать. Таким образом для бухгалтерии у нас образуется три мандата: «бухгалтерия», «доступ к соцсетям» и «доступ к видео». Для производственников — «производство» и опять же «доступ к видео». Для кассы (относящейся к бухгалтерии) придётся снова сделать понижение уровня мандатов: бухгалтерии вообще отключаем интернет и вводим отдельный мандат «интернет». Руководству и системному администратору разделение доступа не нужно, для оптимизации мы можем ввести в матрицу отдельный ресурс.
О ресурсах. Хранилища отделов — очевидны. О доступе к интернету — у нас есть практически 5 вариантов: без интернета, интернет без соцсетей и видео, интернет без соцсетей, интернет без видео, и полный интернет. Логично таким образом сделать интернет с ограничениями плюс два отдельных ресурса: соцсети и видео. В общем-то, некоторые ресурсы в данном примере выродились в отдельные мандаты, но зато мы их видим в общей картине.

Матрица доступа

	Хранилище руководства	Хранилище бухгалтерии	Хранилище продажников	Хранилище сметчиков	Хранилище производственников	Интернет без соцсетей и видео	Соцсети в интернете	Видео в интернете	Неограниченный интернет
Руководитель	+	R	R	R	R				+
Сис.админ.	+	+	+	+	+				+
Бухгалтерия		+	M	R	R				
Продажники			+			+	+		
Сметчики			R	+		+			
Производство					+	+			
Интернет						+			
Соцсети							+		
Видео								+	

Задача 2.

1. Придумать организацию.
2. Составить список пользователей и распределить их по группам

ФИО	Должность	Группы, через запятую

3. Составить список информационных ресурсов – Интернет, Shares, Приказы, 1С итд. Всего 10 ресурсов.
4. Составить матрицу доступа

	Группа1	Группа2	…..
Ресурс 1			

Ресурс 2			

5. Составить график работы информационных ресурсов – по времени обеда, ночное время, отпуска…
6. Вывод.

Лабораторная работа №10.
Защита файловых объектов.

Цель: изучить модель безопасности операционной системы Windows, получить навыки практического использования ее средств обеспечения безопасности.

Теоретические сведения.

1. **Основные сведения**

1. **Классификация защиты семейства ОС Windows**

Защита конфиденциальных данных от несанкционированного доступа является важнейшим фактором успешного функционирования любой многопользовательской системы.

1.2. Идентификация пользователей

Для защиты данных Windows использует следующие основные механизмы: аутентификация и авторизация пользователей, аудит событий в системе, шифрование данных, поддержка инфраструктуры открытых ключей, встроенные средства сетевой защиты.

Защита объектов и аудит действий с ними в ОС Windows организованы на основе избирательного (дискреционного) доступа, когда права доступа (чтение, запись, удаление, изменение атрибутов) субъекта к объекту задается явно в специальной матрице доступа. Для укрупнения матрицы пользователи могут объединяться в группы. При попытке субъекта (одного из потоков процесса, запущенного от его имени) получить доступ к объекту указываются, какие операции пользователь собирается выполнять с объектом. Если подобный тип доступа разрешен, поток получает описатель (дескриптор) объекта и все потоки процесса могут выполнять операции с ним. Подобная схема доступа, очевидно, требует аутентификации каждого пользователя, получающего доступ к ресурсам и его надежную идентификацию в системе, а также механизмов описания прав пользователей и групп пользователей в системе, описания и проверки дискреционных прав доступа пользователей к объектам. Рассмотрим, как в ОС Windows организована аутентификация и авторизация пользователей.

S – R – I – S0 - S1 - … - Sn – RID

Все действующие в системе объекты (пользователи, группы, локальные компьютеры, домены) идентифицируются в Windows не по именам, уникальность которых не всегда удается достичь, а по **идентификаторам защиты (Security Identifiers, SID)**. SID представляет собой числовое значение переменной длины:

S - неизменный идентификатор строкового представления SID;

R – уровень ревизии (версия). На сегодня 1.

I - (identifier-authority) идентификатор полномочий . Представляет собой 48-битную строку, идентифицирующую компьютер или сеть, который(ая) выдал SID объекту. Возможные значения:

- 0 (SECURITY_NULL_SID_AUTHORITY) — используются для сравнений, когда неизвестны полномочия идентификатора;
- 1 (SECURITY_WORLD_SID_AUTHORITY) — применяются для конструирования идентификаторов SID, которые представляют всех пользователей. Например, идентификатор SID для группы *Everyone* (Все пользователи) — это *S-1-1-0*;
- 2 (SECURITY_LOCAL_SID_AUTHORITY) — используются для построения идентификаторов SID, представляющих пользователей, которые входят на локальный терминал;
- 5 (SECURITY_NT_AUTHORITY) — сама операционная система. То есть, данный идентификатор выпущен компьютером или доменом.

Sn – 32-битные коды (колчеством 0 и более) субагентов, которым было передано право выдать SID. Значение первых подчиненных полномочий общеизвестно. Они могут иметь значение:

- 5 — идентификаторы SID присваиваются сеансам регистрации для выдачи прав любому приложению, запускаемому во время определенного сеанса регистрации. У таких идентификаторов SID первые подчиненные полномочия установлены как 5 и принимают форму *S-1-5-5-x-y*;
- 6 —когда процесс регистрируется как служба, он получает специальный идентификатор SID в свой маркер для обозначения данного действия. Этот идентификатор SID имеет подчиненные полномочия 6 и всегда будет *S-1-5-6*;

- 21 (SECURITY_NT_NON_UNIQUE) — обозначают идентификатор SID пользователя и идентификатор SID компьютера, которые не являются уникальными в глобальном масштабе;
- 32 (SECURITY_BUILTIN_DOMAIN_RID) — обозначают встроенные идентификаторы SID. Например, известный идентификатор SID для встроенной группы администраторов *S-1-5-32-544*;
- 80 (SECURITY_SERVICE_ID_BASE_RID) — обозначают идентификатор SID, который принадлежит службе.

Остальные подчиненные полномочия идентификатора совместно обозначают домен или компьютер, который издал идентификатор SID.

RID – 32-битный относительный идентификатор. Он является является идентификатором уникального объекта безопасности в области, для которой был определен SID. Например, 500 — обозначает встроенную учетную запись *Administrator,* 501 — обозначает встроенную учетную запись *Guest, a* 502 — RID для билета на получение билетов протокола Kerberos .

При генерации SID Windows использует генератор случайных чисел, чтобы обеспечить уникальность SID для каждого пользователя. Для некоторого произвольного пользователя SID может выглядеть так:

S-1-5-21-789336058-484763869-725345543-1003

Предопределенным пользователям и группам Windows выдает характерные SID, состоящие из SID компьютера или домена и предопределенного RID. В таблице 1 приведен перечень некоторых общеизвестных SID.

Таблица 1. Общеизвестные SID Windows

SID	Название	Описание
S-1-1-0	Все	Группа, в которую входят все пользователи
S-1-5-2	Сеть	Группа, в которую входят все пользователи, зарегистрировавшиеся в системе из сети
S-1-5-7	Анонимный вход	Группа, в которую входят все пользователи, вошедшие в систему анонимно
S-1-5-*домен*-500	Администратор	Учетная запись администратора системы. По умолчанию только эта запись обеспечивает полный контроль системы
S-1-5-*домен*-501	Гость	Учетная запись пользователя-гостя

Полный список общеизвестных SID можно посмотреть в документации Platform SDK. Узнать SID конкретного пользователя в системе, а также SID групп, в которые он включен, можно, используя консольную команду **whoami**:

whoami /user

Соответствие имени пользователя и его SID можно отследить также в ключе реестра **HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ ProfileList.**

После аутентификации пользователя процессом Winlogon, все процессы, запущенные от имени этого пользователя будут идентифицироваться специальным объектом, называемым **маркером доступа (access token)**. Если процесс пользователя запускает дочерний процесс, то его маркер наследуются, поэтому маркер доступа олицетворяет пользователя для системы в каждом запущенном от его имени процессе. Основные элементы маркера представлены на рис. 1.

SID пользователя	**SID1 … SIDn Идентификаторы групп пользователя**	**DACL по умолчанию**	**Привилегии**	**Прочие параметры**

Рисунок 1. Обобщенная структура маркера доступа.

Маркер доступа содержит идентификатор доступа самого пользователя и всех групп, в которые он включен. В маркер включен также DACL по умолчанию - первоначальный список прав доступа, который присоединяется к создаваемым пользователем объектам. Еще одна важная для определения прав пользователя в системе часть маркера – список его привилегий. Привилегии - это права доверенного объекта на совершение каких-либо действий по отношению ко всей

системе. В таблице 2 перечислены некоторые привилегии, которые могут быть предоставлены пользователю.

Таблица 2. Привилегии, которыми могут быть наделены пользователи

Имя и идентификатор привилегии	Описание привилегии
Увеличение приоритета диспетчирования SeIncreaseBasePriorityPrivilege	Пользователь, обладающий данной привилегией может изменять приоритет диспетчирования процесса с помощью интерфейса Диспетчера задач
Закрепление страниц в памяти SeLockMemoryPrivilege	Процесс получает возможность хранить данные в физической памяти, не прибегая к кэшированию данных в виртуальной памяти на диске.
Управление аудитом и журналом безопасности SeAuditPrivilege	Пользователь получает возможность указывать параметры аудита доступа к объекту для отдельных ресурсов, таких как файлы, объекты Active Directory и разделы реестра.
Овладение файлами или иными объектами SeTakeOwnershipPrivilege	Пользователь получает возможность становиться владельцем любых объектов безопасности системы, включая объекты Active Directory, файлы и папки NTFS, принтеры, разделы реестра, службы, процессы и потоки
Завершение работы системы SeShutdownPrivilege	Пользователь получает возможность завершать работу операционной системы на локальном компьютере
Обход перекрестной проверки SeChangeNotifyPrivilege	Используется для обхода проверки разрешений для промежуточных каталогов при проходе многоуровневых каталогов

Управление привилегиями пользователей осуществляется в оснастке **«Групповая политика»**, раздел **Конфигурация Windows/Локальные политики/Назначение прав пользователя**.

Чтобы посмотреть привилегии пользователя, можно также использовать команду

whoami /all

Остальные параметры маркера носят информационный характер и определяют, например, какая подсистема создала маркер, уникальный идентификатор маркера, время его действия. Необходимо также отметить возможность создания ограниченных маркеров (restricted token), которые отличаются от обычных тем, что из них удаляются некоторые привилегии и его SID-идетификаторы проверяются только на запрещающие правила. Создать ограниченный маркер можно программно, используя API-функцию **CreateRestrictedToken**, а можно запустить процесс с ограниченным маркером, используя пункт контекстного меню Windows **“Запуск от имени…”** и отметив пункт **“Защитить компьютер от несанкционированных действий этой программы”** (рис.2).

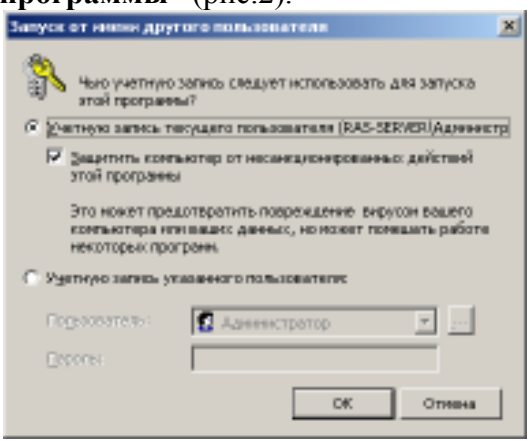


Рисунок 2. Запуск процесса с ограниченным маркером

Ограниченные маркеры используются для процессов, подменяющих клиента и выполняющих небезопасный код.

Маркер доступа может быть создан не только при первоначальном входе пользователя в систему. Windows предоставляет возможность запуска процессов от имени других пользователей, создавая для этих процессов соответствующий маркер. Для этих целей можно использовать:
- API-функции **CreateProcessAsUser, CreateProcessWithLogon**;
- оконный интерфейс (рис.2), инициализирующийся при выборе пункта контекстного меню **“Запуск от имени…”**;
- консольную команду **runas**:
runas /user:*имя_пользователя program* ,
где *имя_пользователя* - имя учетной записи пользователя, которая будет использована для запуска программы в формате *пользователь@домен* или *домен\пользова-тель*;
program – команда или программа, которая будет запущена с помощью учетной записи, указанной в параметре **/user.**
В любом варианте запуска процесса от имени другой учетной записи необходимо задать ее пароль.

1.3. Защита объектов системы.

Маркер доступа идентифицирует субъектов-пользователей системы. С другой стороны, каждый объект системы, требующий защиты, содержит описание прав доступа к нему пользователей. Для этих целей используется **дескриптор безопасности (Security Descriptor, SD)**. Каждому объекту системы,включая файлы, принтеры, сетевые службы, контейнеры Active Directory и другие, присваивается дескриптор безопасности, который определяет права доступа к объекту и содержит следующие основные атрибуты (рис.3):
- SID владельца, идентифицирующий учетную запись пользователя-владельца объекта;
- пользовательский список управления доступом (Discretionary Access Control List, **DACL**), который позволяет отслеживать права и ограничения, установленные владельцем данного объекта. DACL может быть изменен пользователем, который указан как текущий владелец объекта.
- системный список управления доступом (System Access Control List, SACL), определяющий перечень действий над объектом, подлежащих аудиту;
- флаги, задающие атрибуты объекта.
Авторизация Windows основана на сопоставлении маркера доступа субъекта с дескриптором безопасности объекта. Управляя свойствами объекта, администраторы могут устанавливать разрешения, назначать право владения и отслеживать доступ пользователей.

Рисунок 3. Структура дескриптора безопасности объекта Windows

Список управления доступом содержит набор элементов (Access Control Entries, ACE). В **DACL** каждый ACE состоит из четырех частей: в первой указываются пользователи или группы, к которым относится данная запись, во второй – права доступа, а третья информирует о том, предоставляются эти права или отбираются. Четвертая часть представляет собой набор флагов, определяющих, как данная запись будет наследоваться вложенными объектами (актуально, например, для папок файловой системы, разделов реестра).
Если список ACE в DACL пуст, к нему нет доступа ни у одного пользователя (только у владельца на изменение DACL). Если отсутствует сам DACL в SD объекта – полный доступ к нему имеют все пользователи.
Если какой-либо поток запросил доступ к объекту, подсистема SRM осуществляет проверку прав пользователя, запустившего поток, на данный объект, просматривая его список DACL. Проверка осуществляется до появления разрешающих прав **на все** запрошенные операции. Если встретится запрещающее правило хотя бы **на одну** запрошенную операцию, доступ не будет предоставлен.

Рассмотрим пример на рис.4. Процесс пытается получить доступ к объекту с заданным DACL. В маркере процесса указаны SID запустившего его пользователя, а также SID групп, в которые он входит. В списке DACL объекта присутствуют разрешающие правила на чтение для пользователя с SID=100, и на запись для группы с SID=205. Однако, в доступе пользователю будет отказано, поскольку раньше встречается запрещающее запись правило для группы с SID=201.

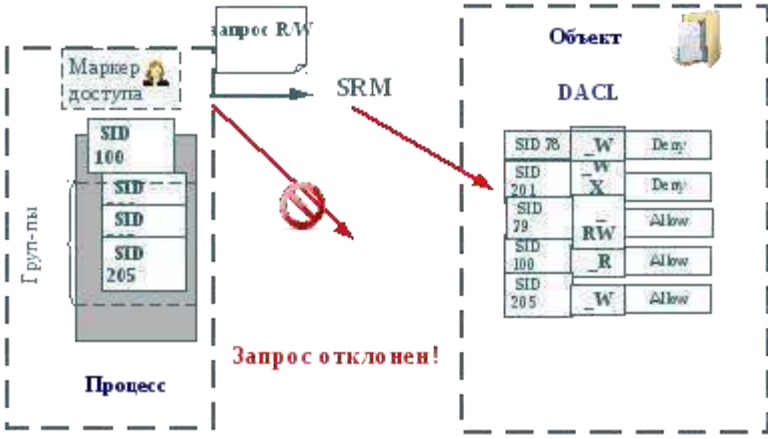


Рисунок 4. Проверка прав доступа пользователя к объекту

Необходимо отметить, что запрещающее правило помещено в списке DACL на рисунке не случайно. Запрещающие правила **всегда** размещаются перед разрешающими, то есть являются доминирующими при проверке прав доступа.

Для определения и просмотра прав доступа пользователей к ресурсам можно использовать как графические средства контроля, так и консольные команды. Стандартное окно свойств объекта файловой системы (диска, папки, файла) на вкладке **Безопасность** (рис. 5) позволяет просмотреть текущие разрешения для пользователей и групп пользователей, редактировать их, создавать новые или удалять существующие.

Рисунок 6. Определение параметров наследования прав доступа к объектам

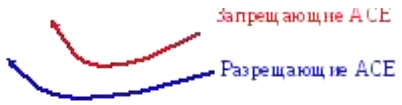


Рисунок 5. GUI-интерфейс Windows для изменения прав доступа к объектам

При определении прав доступа к объектам можно задать правила их наследования в дочерних контейнерах. В окне дополнительных параметров безопасности на вкладке **Разрешения** при выборе опции «**Наследовать от родительского объекта применимых к дочерним объектам разрешения, добавляя их к явно заданным в этом окне»** (рис. 6) можно унаследовать разрешения и ограничения, заданные для родительского контейнера, текущему объекту.

При выборе опции **«Заменить разрешения для всех дочерних объектов заданными здесь разрешениями, применимыми к дочерним объектам»** разрешается передача определенных для объекта-контейнера правил доступа его дочерним объектам.

В этом же окне на вкладке **Владелец** допустимо узнать владельца объекта и заменить его. Владелец объекта имеет право на изменение списка его DACL, даже если к нему запрещен любой тип доступа. Администратор имеет право становиться владельцем любого объекта.

С учетом возможности вхождения пользователя в различные группы и независимости определения прав доступа к объектам для групп и пользователей, зачастую бывает сложно определить конечные права пользователя на доступ к объекту: требуется просмотреть запрещающие правила, определенные для самого объекта, для всех групп, в которые он включен, затем то же проделать для разрешающих правил. Автоматизировать процесс определения разрешенных пользователю видов доступа к объекту можно с использованием вкладки **«Действующие разрешения»** окна дополнительных параметров безопасности объекта (рис. 7).

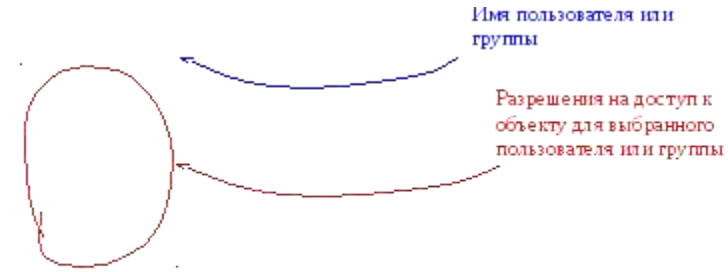


Рисунок 7. Определение эффективных прав доступа пользователя (группы) к объекту

Для просмотра и изменения прав доступа к объектам в режиме командной строки предназначена команда **cacls** (**icacls** в Windows Vista и Widows 7).

cacls *имя_файла* [**/t**] [**/e**] [**/c**] [**/g** *пользователь*:*разрешение*] [**/r** *пользователь* [...]] [**/p** *пользователь*:*разрешение* [...]] [**/d** *пользователь* [...]]

Назначения параметров команды приведены в таблице 3.

Таблица 3. Параметры команды cacls

<имя файла>	Задаёт файл или папку, права доступа к которой необходимо просмотреть/изменить (допустимо использовать шаблоны с символами ***** и **?**).
/t	Изменение избирательных таблиц контроля доступа (DACL) указанных файлов в текущем каталоге и всех подкаталогах
/e	Редактирование избирательной таблицы управления доступом (DACL) вместо ее замены
/c	Заставляет команду продолжить изменение прав доступа при возникновении ошибки, связанной с нарушениями прав доступа
/g <пользователь | группа: разрешение>	Предоставление прав доступа указанному пользователю
/r <пользователь | группа>	Отнимает права доступа указанного пользователя.
/p <пользователь | группа: разрешение>	Заменяет права доступа указанного пользователя
/d <пользователь | группа>	Отказывает в праве доступа указанному пользователю или группе

Для указания добавляемых или отнимаемых прав используются следующие значения:
F - полный доступ;
C - изменение (запись);
W - запись;
R - чтение;
N - нет доступа.
Рассмотрим несколько примеров.
cacls d:\test
Выдаст список DACL для папки test.
cacls d:\test /d ИмяКомпьютера\ИмяПользователя /e
Запретит доступ к объекту для указанного пользователя.
cacls d:\test /p ИмяКомпьютера\ИмяГруппы:f /e /t
Предоставит полный доступ к папке d:\test и ее подпапках всем для членов указанной группы.
Для программного просмотра и изменения списков DACL можно использовать API-функции AddAccessAllowedAce, AddAccessDeniedAce, SetSecurityInfo. Подробнее с этими функциями и примерами их использования можно ознакомиться в [пособие].
1.4. Подсистема аудита.
Важный элемент политики безопасности – аудит событий в системе. ОС Windows ведет аудит событий по 9 категориям:
1. Аудит событий входа в систему.
2. Аудит управления учетными записями.
3. Аудит доступа к службе каталогов.
4. Аудит входа в систему.

5. Аудит доступа к объектам.
6. Аудит изменения политики.
7. Аудит использования привилегий.
8. Аудит отслеживания процессов.
9. Аудит системных событий.

Рассмотрим более подробно, какие события отслеживает каждая из категорий.

Аудит событий входа в систему

Аудит попыток пользователя войти в систему с другого компьютера или выйти из нее, при условии, что этот компьютер используется для проверки подлинности учетной записи.

Аудит управления учетными записями

Аудит событий, связанных с управлением учетными записями на компьютере: создание, изменение или удаление учетной записи пользователя или группы; переименование, отключение или включение учетной записи пользователя; задание или изменение пароля.

Аудит доступа к службе каталогов

Аудит событий доступа пользователя к объекту каталога Active Directory, для которого задана собственная системная таблица управления доступом (SACL).

Аудит входа в систему

Аудит попыток пользователя войти в систему с компьютера или выйти из нее.

Аудит доступа к объектам

Аудит событий доступа пользователя к объекту – например, к файлу, папке, разделу реестра, принтеру и т. п., - для которого задана собственная системная таблица управления доступом (SACL).

Аудит изменения политики

Аудит фактов изменения политик назначения прав пользователей, политик аудита или политик доверительных отношений.

Аудит использования привилегий

Аудит попыток пользователя воспользоваться предоставленным ему правом.

Аудит отслеживания процессов

Аудиту таких событий, как активизация программы, завершение процесса, повторение дескрипторов и косвенный доступ к объекту.

Аудит системных событий

Аудит событий перезагрузки или отключения компьютера, а также событий, влияющих на системную безопасность или на журнал безопасности.

Решения об аудите конкретного типа событий безопасности принимаются в соответствии с политикой аудита локальной системы. Политика аудита, также называемая локальной политикой безопасности (local security policy), является частью политики безопасности, поддерживаемой LSASS в локальной системе, и настраивается с помощью редактора локальной политики безопасности (Оснастка **gpedit.msc, Конфигурация компьютера - Конфигурация Windows – Параметры безопасности – Локальные политики – Политика аудита,** рис. 8).

Рисунок 8. Конфигурация политики аудита редактора локальной политики безопасности

Для каждого объекта в SD содержится список SACL, состоящий из записей ACE, регламентирующих запись в журнал аудита удачных или неудачных попыток доступа к объекту. Эти ACE определяют, какие операции, выполняемые над объектами конкретными пользователями или группами, подлежат аудиту. Информация аудита хранится в системном журнале аудита. Аудиту могут подлежать как успешные, так и неудачные операции. Подобно записям ACE DACL, правила аудита объектов могут наследоваться дочерними объектами. Процедура наследования определяются набором флагов, являющихся частью структуры ACE.

Рисунок 9. Интерфейс редактирования правил аудита для объекта

Настройка списка SACL может быть осуществлена в окне дополнительных свойств объекта (**пункт “Дополнительно”, закладка “Аудит”,** рис. 9).

Для программного просмотра и изменения списков SACL можно использовать API-функции **GetSecutityInfo** и **SetSecutityInfo**.

При инициализации системы и изменении политики LSASS посылает SRM сообщения,

информирующие его о текущей политике аудита. LSASS отвечает за прием записей аудита, генерируемых на основе событий аудита от SRM, их редактирование и передачу Event Logger (регистратору событий). SRM посылает записи аудита LSASS через свое LPC-соединение. После этого Event Logger заносит записи в журнал безопасности.

События аудита записываются в журналы следующих типов:

1. **Журнал приложений.** В журнале приложений содержатся данные, относящиеся к работе приложений и программ.
2. **Журнал безопасности.** Журнал безопасности содержит записи о таких событиях, как успешные и безуспешные попытки доступа в систему, а также о событиях, относящихся к использованию ресурсов.
3. **Журнал системы.** В журнале системы содержатся события системных компонентов Windows. Например, в журнале системы регистрируются сбои при загрузке драйвера или других системных компонентов при запуске системы.

4. Журнал службы каталогов. В журнале службы каталогов содержатся события, заносимые службой каталогов Windows (на контроллере домена AD).

5. Журнал службы репликации. В журнале службы репликации файлов содержатся события, заносимые службой репликации файлов Windows (на контроллере домена AD).

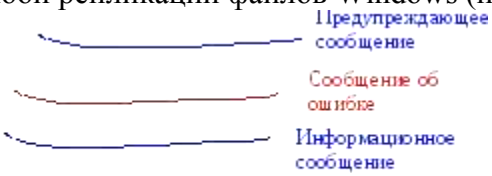


Рисунок 10. Оснастка Windows «Просмотр событий»

Просмотр журнала безопасности осуществляется в оснастке «Просмотр событий» (**eventvwr.msc,** рис. 10). Сами журналы хранятся в файлах **SysEvent.evt, SecEvent.evt, AppEvent.evt** в папке **%WinDir%\system32\config.**

В журнал записываются события 3 основных видов:

1. **Информационные сообщения о событиях.**

Описывают успешное выполнение операций, таких как запуск или некоторое действие системной службы.

2. Предупреждающие сообщения о событиях.

Описывают неожиданные действия, означающие проблему, или указывают на проблему, которая возникнет в будущем, если не будет устранена сейчас .

3. Сообщения о событиях ошибок.

Описывают ошибки, возникшие из-за неудачного выполнения задач.

1.5. Шифрующая файловая система.

Начиная с версии Windows 2000, в операционных системах семейства Windows NT поддерживается шифрование данных на разделах файловой системы NTFS с использованием *шифрующей файловой системы* (**Encrypted File System, EFS**). Основное ее достоинст во заключается в обеспечении конфиденциальности данных на дисках компьютера за счет использования надежных симметричных алгоритмов для шифрования данных в реальном режиме времени.

Для шифрации данных EFS использует симметричный алгоритм шифрования (AES или DESX) со случайным ключом для каждого файла (**File Encryption Key, FEK**). По умолчанию данные шифруются в Windows 2000 и Windows XP по алгоритму DESX, а в Windows XP с Service Pack 1 (или выше) и Windows Server 2003 — по алгоритму AES. В версиях Windows, разрешенных к экспорту за пределы США, драйвер EFS реализует 56-битный ключ шифрования DESX, тогда как в версии, подлежащей использованию только в США, и в версиях с пакетом для 128-битного шифрования длина ключа DESX равна 128 битам. Алгоритм AES в Windows использует 256-битные ключи.

При этом для обеспечения секретности самого ключа FEK шифруется асимметричным алгоритмом RSA открытым ключом пользователя, результат шифрации FEK – **Data Decryption Field, DDF** – добавляется в заголовок зашифрованного файла (рис. 11). Такой подход обеспечивает надежное шифрование без потери эффективности процесса шифрования: данные шифруются быстрым симметричным алгоритмом, а для гарантии секретности симметричного ключа используется асимметричный алгоритм шифрования.

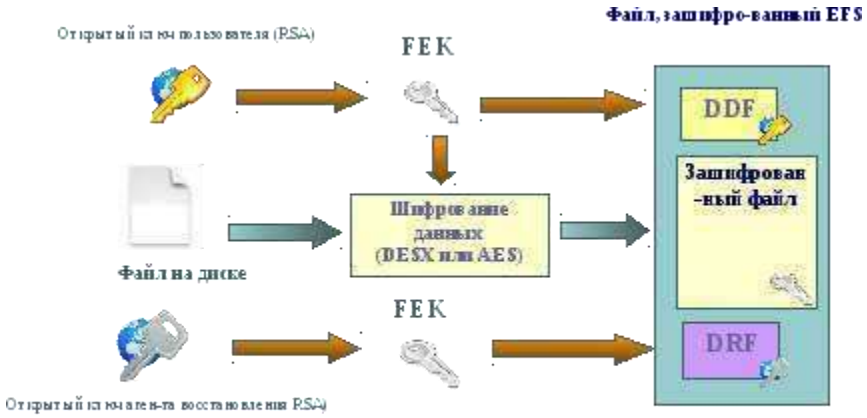


Рисунок 11. Схема шифрации файла в EFS

Для шифрации файлов с использованием EFS можно использовать графический интерфейс или команду **cipher**.

Графический интерфейс доступен в стандартном окне свойств объекта по нажатию кнопки **«Дополнительно»** (рис. 12). Зашифрованные объекты в стандартном интерфейсе Windows Explorer отображаются зеленым цветом.

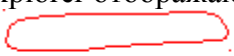


Рисунок 12. Графический интерфейс шифрования файла с использованием EFS

Необходимо отметить, что EFS позволяет разделять зашифрованный файл между несколькими пользователями. В этом случае FEK шифруется открытыми ключами всех пользователей, которым разрешен доступ к файлу, и каждый результат шифрации добавляется в DDF.

Шифрование файла с использованием EFS защищает файл комплексно: пользователю, не имеющему права на дешифрацию файла, недопустимы, в том числе, такие операции, как удаление, переименование и копирование файла. Необходимо помнить, что EFS является частью файловой системы NTFS, и в случае копирования защищенного файла авторизованным пользователем на другой том с файловой системой, на поддерживающей EFS (например, FAT32), он будет дешифрован и сохранен на целевом томе в открытом виде.

Консольная команда **cipher** может быть использована для шифрации/дешифрации файлов из командной строки или в bat-сценарии.

cipher [{**/e**|**/d**}] [**/s:***каталог*] [**/a**] [**/i**] [**/f**] [**/q**] [**/h**] [**/k**] [**/u**[**/n**]] [*путь* [...]] | [**/r:***имя_файла_без_расширения*]

Назначения параметров команды приведены в таблице 4.

Таблица 4. Параметры команды **cipher**

/e	Шифрует указанные папки. Папки помечаются таким образом, чтобы файлы, которые будут добавляться в папку позже, также шифровались.
/d	Расшифровывает указанные папки. Папки помечаются таким образом, чтобы файлы, которые будут добавляться в папку позже, не будут шифроваться
/s: ***каталог***	Выполняет выбранную операцию над указанной папкой и всеми подпапками в ней.
/a	Выполняет операцию над файлами и каталогами
/i	Продолжение выполнения указанной операции даже после возникновения ошибок. По умолчанию выполнение **cipher** прекращается после возникновения ошибки
/f	Выполнение повторного шифрования или расшифровывания указанных объектов. По умолчанию уже зашифрованные или расшифрованные файлы пропускаются командой **cipher**
/k	Создание ключа шифрования файла для пользователя, выполнившего команду **cipher**. Если используется данный параметр, все остальные параметры команды **cipher** не учитываются.

/u Обновление ключа шифрования файла пользователя или ключа агента восстановления на текущие ключи во всех зашифрованных файлах на локальном диске (если эти ключи были изменены). Этот параметр используется только вместе с параметром **/n**.

/n Запрещение обновления ключей. Данный параметр служит для поиска всех зашифрованных файлов на локальных дисках. Этот параметр используется только вместе с параметром **/u**.

путь Указывает шаблон, файл или папку.

/r: ***имя_файла*** Создание нового сертификата агента восстановления и закрытого ключа с последующей их записью в файлах с именем, указанным в параметре *имя_файла* (без расширения). Если используется данный параметр, все остальные параметры команды **cipher** не учитываются.

Например, чтобы определить, зашифрована ли какая-либо папка, необходимо использовать команду:

cipher *путь\имя_папки*

Команда **cipher** без параметров выводит статус (зашифрован или нет) для всех объектов текущей папки.

Для шифрации файла необходимо использовать команду

cipher /e /a *путь\имя_файла*

Для дешифрации файла, соответственно, используется команда

cipher /d /a *путь\имя_файла*

Допустима шифрация/дешифрация группы файлов по шаблону:

cipher /e /a d:\work*.doc

Пара открытый и закрытый ключ для шифрации FEK создаются для пользователя автоматически при первой шифрации файла с использованием EFS.

Если некоторый пользователь или группа пользователей зашифровали файл с использованием EFS, то его содержимое доступно только им. Это приводит к рискам утери доступа к данным в зашифрованных файлах в случае утраты пароля данным пользователем (работник забыл пароль, уволился и т.п.). Для предотвращения подобных проблем администратор может определить некоторые учетные записи в качестве агентов восстановления.

Агенты восстановления (Recovery Agents) определяются в политике безопасности **Encrypted Data Recovery Agents (Агенты восстановления шифрованных данных)** на локальном компьютере или в домене. Эта политика доступна через оснастку **Групповая политика (gpedit.msc)** раздел **«Параметры безопасности»-> «Политика открытого ключа»-> «Файловая система EFS».** Пункт меню **«Действие»-> «Добавить агент восстановления** данных» открывает мастер добавления нового агента.

Добавляя агентов восстановления можно указать, какие криптографические пары (обозначенные их сертификатами) могут использовать эти агенты для восстановления шифрованных данных (рис. 13). Сертификаты для агентов восстановления создаются командой **cipher** с ключом **/r** (см. табл. 4). Для пользователя, который будет агентом восстановления, необходимо импортировать закрытый ключ агента восстановления из сертификата, созданного командой **cipher**. Это можно сделать в мастере импорта сертификатов, который автоматически загружается при двойном щелчке по файлу *.pfx.

Рисунок 13. Добавление нового агента восстановления EFS

EFS создает – DRF **(Data Recovery Field)**-элементы ключей для каждого агента восстановления, используя провайдер криптографических сервисов, зарегистрированный для EFS-восстановления. DRF добавляется в зашифрованный файл и может быть использован как альтернативное средство извлечения FEK для дешифрации содержимого файла.

Windows хранит закрытые ключи в подкаталоге **Application Data\Micro-soft\Crypto\RSA** каталога профиля пользователя. Для защиты закрытых ключей Windows шифрует все файлы в папке RSA на основе симметричного ключа, генерируемого случайным образом; такой ключ называется мастер-ключом пользователя. Мастер-ключ имеет длину в 64 байта и создается стойким генератором случайных чисел. Мастер-ключ также хранится в профиле пользователя в каталоге Application Data\Microsoft\Protect и зашифровывается по алгоритму 3DES с помощью

ключа, который отчасти основан на пароле пользователя. Когда пользователь меняет свой пароль, мастер-ключи автоматически расшифровываются, а затем заново зашифровываются с учетом нового пароля.

Для расшифровки FEK EFS использует функции Microsoft CryptoAPl (CAPI). CryptoAPI состоит из DLL провайдеров криптографических сервисов (cryptographic service providers, CSP), которые обеспечивают приложениям доступ к различным криптографическим сервисам (шифрованию, дешифрованию и хэшированию). EFS опирается на алгоритмы шифрования RSA, предоставляемые провайдером **Microsoft Enhanced Cryptographic Provider (\Windows\ System32\Rsaenh.dll).**

Шифрацию и дешифрацию файлов можно осуществлять программно, используя API-функции **EncryptFile** и **DecryptFile**.

2. **Порядок выполнения работы.**

2.1. Ознакомьтесь с теоретическими основами защиты информации в ОС семейства Windows в настоящих указаниях и конспектах лекций.

2.2. Выполните задания 2.2.1-2.2.8

1. Запустите в программе **Oracle VM Virtualbox** виртуальную машину WinXP. Войдите в систему под учетной записью администратора, пароль узнайте у преподавателя. Все действия в пп 2.2.1-2.2.8 выполняйте в системе, работающей на виртуальной машине.
2. Создайте учетную запись нового пользователя **testUser** в оснастке **«Управление компьютером» (compmgmt.msc)**. При создании новой учетной записи запретите пользователю смену пароля и снимите ограничение на срок действия его пароля. Создайте новую группу **"testGroup"** и включите в нее нового пользователя. Удалите пользователя из других групп. Создайте на диске **C:** папку **forTesting**. Создайте или скопируйте в эту папку несколько текстовых файлов (*.txt).
3. С помощью команды **runas** запустите сеанс командной строки (**cmd.exe**) от имени вновь созданного пользователя. Командой **whoami** посмотрите SID пользователя и всех его групп, а также текущие привилегии пользователя. Строку запуска и результат работы этой и ***всех*** следующих консольных команд копируйте в файл протокола лабораторной работы.
4. Убедитесь в соответствии имени пользователя и полученного SID в реестре Windows. Найдите в реестре, какому пользователю в системе присвоен SID **S-1-5-21-1957994488-492894223-170857768-1004** (Используйте ключ реестра **HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList**).
5. Командой **whoami** определите перечень текущих привилегий пользователя **testUser**. В сеансе командной строки пользователя попробуйте изменить системное время командой **time**. Чтобы предоставить пользователю подобную привилегию, запустите оснастку **«Локальные параметры безопасности» (secpol.msc).** Добавьте пользователя в список параметров политики **«Изменение системного времени»** раздела **Локальные политики -> Назначение прав пользователя**. После этого перезапустите сеанс командной строки от имени пользователя, убедитесь, что в списке привилегий добавилась **SeSystemtimePriviege**. Попробуйте изменить системное время командой **time**.

Убедитесь, что привилегия **«Завершение работы системы» (SeShutdownPrivilege)** предоставлена пользователю testUser . После этого попробуйте завершить работу системы из сеанса командной строки пользователя командой **shutdown –s**. Добавть ему привелегию **«Принудительное удаленное завершение» (SeRemoteShutdownPrivilege)**. Попробуйте завершить работу консольной командой еще раз (отменить команду завершения до ее непосредственного выполнения можно командой **shutdown –a**).

2.2.6. Ознакомьтесь с справкой по консольной команде **cacls.** Используя эту команду, просмотрите разрешения на папку **c:\forTesting**. Объясните все обозначения в описаниях прав пользователей и групп в выдаче команды.

а) Разрешите пользователю **testUse**r запись в папку **forTesting**, но запретите запись для группы **testGroup**. Попробуйте записать файлы или папки в **forTesting** от имени пользователя **testUser**. Объясните результат. Посмотрите эффективные разрешения пользователя **testUser** к папке **forTesting** в окне свойств папки.

б) Используя стандартное окно свойств папки, задайте для пользователя **testUser** такие права доступа к папке, чтобы он мог записывать информацию в папку **forTesting**, но не мог просматривать ее содержимое. Проверьте, что папка **forTesting** является теперь для пользователя

testUser “слепой”, запустив, например, от его имени файловый менеджер и попробовав записать файлы в папку, просмотреть ее содержимое, удалить файл из папки.
в) Для вложенной папки **forTesting\Docs** отмените наследование ACL от родителя и разрешите пользователю просмотр, чтение и запись в папку. Проверьте, что для пользователя папка **forTesting\Docs** перестала быть “слепой” (например, сделайте ее текущей в сеансе работы файлового менеджера от имени пользователя и создайте в ней новый файл).
г) Снимите запрет на чтение папки **forTesting** для пользователя **testUser**. Используя команду **cacls** запретите этому пользователю доступ к файлам с расширением txt в папке **forTesting**. Убедитесь в недоступности файлов для пользователя.
д) Командой **cacls** запретите пользователю все права на доступ к папке **forTesting** и разреште полный доступ к вложенной папке **forTesting\Docs**. Убедитесь в доступности папки **forTesting\Docs** для пользователя. Удалите у пользователя **testUser** привилегию **SeChangeNotifyPrivilege**. Попробуйте получить доступ к папке **forTesting\Docs**. Объясните результат.
е) Запустите файловый менеджер от имени пользователя **testUser** и создайте в нем папку **newFolder** на диске C. Для папки **newFolder** очистите весь список ACL командой **cacls**. Попробуйте теперь получить доступ к папке от имени администратора и от имени пользователя. Кто и как теперь может вернуть доступ к папке? Верните полный доступ к папке для всех пользователей.
ж) Создайте в разделе **HKLM\Software** реестра раздел **testKey**. Запретите пользователю **testUser** создание новых разделов в этом разделе реестра. Создайте для раздела **HKLM\Software\testKey** SACL, позволяющий протоколировать отказы при создании новых подразделов, а также успехи при перечислении подразделов и запросе значений (предварительно проверьте, что в локальной политике безопасности соответствующий тип аудита включен). Попробуйте от имени пользователя **testUser** запустить **regedit.exe** и создать раздел в **HKLM\Software**. Убедитесь, что записи аудита были размещены в журнале безопасности (**eventvwr.msc**).

7. Шифрование файлов и папок средствами EFS.

а) От имени пользователя **testUser** зашифруйте какой-нибудь файл на диске. Убедитесь, что после этого был создан сертификат пользователя, запустив оснастку **certmgr.msc** от имени пользователя (раздел **Личные**). Просмотрите основные параметры сертификата открытого ключа пользователя **testUser** (срок действия, используемые алгоритмы). Установите доверие к этому сертификату в вашей системе.
б) Создайте в папке **forTesting** новую папку **Encrypt**. В папке **Encrypt** создайте или скопируйте в нее текстовый файл. Зашифруйте папку **Encrypt** и все ее содержимое из меню свойств папки от имени администратора. Попробуйте просмотреть или скопировать какой-нибудь файл этой папки от имени пользователя **testUser**. Объясните результат. Скопируйте зашифрованный файл в незашифрованную папку (например, **forTesting**). Убедитесь что он остался зашифрованным. Добавьте пользователя **testUser** в список имеющих доступа к файлу пользователей в окне свойств шифрования файла. Повторите попытку получить доступ к файлу от имени пользователя **testUser**.
в) Создайте учетную запись нового пользователя **agentUser**, сделайте его членом группы Администраторы. Определите для пользователя **agentUser** роль агента восстановления EFS. Создайте в папке **forTesting** новый текстовый файл с произвольным содержимым. Зашифруйте этот файл от имени пользователя **testUser**. Убедитесь в окне подробностей шифрования файла, что пользователь **agentUser** является агентом восстановления для данного файла. Попробуйте прочитать содержимое файла от имени администратора и от имени пользователя **agentUser**. Объясните результат.
г) Зашифруйте все текстовые файлы папки **forTesting** с использованием консольной команды шифрования **cipher** от имени пользователя **testUser** (предварительно снимите запрет на доступ к этим файлам, установленный в задании 2.2.6г).
д) Убедитесь, что при копировании зашифрованных файлов на том с файловой системой, не поддерживающей EFS (например, FAT32 на флеш-накопителе), содержимое файла дешифруется.
2.2.8. После демонстрации результатов работы преподавателю восстановите исходное состояние системы: удалите созданные папки и файлы, разделы реестра, удалите учетную запись созданного пользователя и его группы, снимите с пользователя **agentUse**r роль агента восстановления.
2.2.9. Представьте отчёт по лабораторной работе преподавателю и отчитайте работу.

2.3. Содержание отчета

Отчет по лабораторной работе должен содержать следующие сведения:

- название и цель работы;
- протокол выполнения лабораторной работы, содержащий список консольных команд, составленных при выполнении работы, и результаты их выполнения.

2. **Контрольные вопросы**

1. К какому классу безопасности относится ОС Windows по различным критериям оценки?
2. Каким образом пользователи идентифицируются в ОС Windows?
3. Что такое списки DACL и SACL?
4. Перечислите, каким образом можно запустить процесс от имени другого пользователя.
5. Как происходит проверка прав доступа пользователя к ресурсам в ОС Windows?
6. Что такое маркер безопасности, и какова его роль в модели безопасности Windows?
7. Как с использованием команды cacls добавить права на запись для всех файлов заданной папки?
8. Какие события подлежат аудиту в ОС Windows?
9. Каким образом шифруются файлы в файловой системе EFS? Что такое FEK? DDF? DDR?
10. Какие алгоритмы шифрования используются в EFS?

Лабораторная работа №11. Организация общего доступа к ресурсам файловой системы.

Цель: Освоение навыков управления доступом пользователей к файлам и папкам с целью защиты информации от несанкционированного доступа

Теоретические сведения

Файловые системы современных операционных систем при соответствующей настройке эффективно обеспечивают безопасность и надежность хранения данных на дисковых накопителях. Для операционных систем Windows стандартной является файловая система NTFS.

Устанавливая для пользователей определенные разрешения для файлов и каталогов (папок), администраторы могут защитить информацию от несанкционированного доступа. Каждый пользователь должен иметь определенный набор разрешений на доступ к конкретному объекту файловой системы. Кроме того, он может быть владельцем файла или папки, если сам их создает. Администратор может назначить себя владельцем любого объекта файловой системы, но обратная передача владения от администратора к пользователю невозможна.

Назначение разрешений производится для пользователей или групп. Так как рекомендуется выполнять настройки безопасности для групп, то необходимо, чтобы пользователь был членом хотя бы одной группы на компьютере или в домене.

Разрешения могут быть установлены для различных объектов компьютерной системы, однако в этой работе будут рассмотрены разрешения для файлов и папок. Другие задачи, например разрешения для принтеров, решаются аналогичным образом.

Указания к проведению лабораторной работы

Для назначения разрешений для файла или папки администратор выбирает данный файл или папку и при нажатии правой кнопки мыши использует команду Свойства (Properties) и в появившемся окне переходит на вкладку Безопасность (Security). Пример для папки с именем Авиатор приведен на рисунке 1.

В зоне Имя (Name) имеется список групп и пользователей, которым уже назначены разрешения для данного файла или папки.

Рисунок 1 Вкладка Безопасность окна свойств папки Авиатор

Для добавления пользователя или группы нажмите кнопку Добавить (Add) или Удалить (Remove). При добавлении появится диалог Выбор: Пользователи, Компьютеры или Группы (SelectUsers,Computers,orGroups). Добавив пользователя или группу мы увидим этот объект в зоне Имя и выделив его, можем задать необходимые разрешения с помощью установки флажков Разрешить (Allow) или Запретить (Deny) в зоне Разрешения (Permissions).

Стандартные разрешения для файлов:

- Полный доступ (Full Control);
- Изменить (Modify);
- Чтение и выполнение (Read&Execute);
- Чтение (Read);
- Запись (Write).

Стандартные разрешения для папок:

- Полный доступ (Full Control);
- Изменить (Modify);

- Чтение и выполнение (Read&Execute);
- Список содержимого папки
- Чтение (Read);
- Запись (Write).

Разрешение Чтение позволяет просматривать файлы и папки и их атрибуты.
Разрешение Запись позволяет создавать новые файлы и папки внутри папок, изменять атрибуты и просматривать владельцев и разрешения.
Разрешение Список содержимого папки позволяет просматривать имена файлов и папок.
Разрешение Чтение и выполнение для папок позволяет перемещаться по структуре других папок и служит для того, чтобы разрешить пользователю открывать папку, даже если он не имеет прав доступа к ней, для поиска других файлов или вложенных папок. Разрешены все действия, право на которые дают разрешения Чтение и Список содержимого папки. Это же разрешение для файлов позволяет запускать файлы программ и выполнять действия, право на которые дает разрешение Чтение.
Разрешение Изменить позволяет удалять папки, файлы и выполнять все действия, право на которые дают разрешения Запись и Чтение и выполнение.
Разрешение Полный доступ позволяет изменять разрешения, менять владельца, удалять файлы и папки и выполнять все действия, на которые дают право все остальные разрешения NTFS.
Разрешения для папок распространяются на их содержимое: подпапки и файлы.
При назначении пользователю или группе разрешения на доступ к файлу или папке руководствуются тем уровнем доступа, который достаточен для данной группы или пользователя при выполнении им своих рабочих обязанностей.
Рассмотренные разрешения относятся к пользователям данного компьютера, совершившим вход локально непосредственно на данную машину. Такие разрешения называются разрешениями файловой системы.
Так как файловая система Windows называется NTFS, то разрешения файловой системы для Windows называют разрешениями NTFS.
Разрешения для пользователей, получившим доступ к папке или файлу через сеть, регулируются отдельно с помощью так называемых разрешений общего доступа. Эти разрешения распространяются только на папки, к которым предоставлен общий доступ через сеть и действуют только для пользователей, обращающихся к папке через сеть. Возможности пользователя задаются разрешениями, представленными ниже:

- Полный доступ (Full Control);
- Изменить (Change);
- Чтение (Read);

Доступ к средствам настройки разрешений общего доступа выполняется через свойства папки, предоставленной в общий доступ (рисунок 4.2)

Рисунок 4.2 Разрешения общего доступа для папки WS_DEMO

Разрешения общего доступа являются средством обеспечения безопасности данных при коллективной работе с документами и поэтому должны устанавливаться очень тщательно и обоснованно. При этом администратору рекомендуется действовать следующим образом.

- Для каждого ресурса общего доступа определить, каким группам пользователей необходим доступ к нему и какой требуется уровень доступа;
- Для упрощения администрирования назначайте разрешения группам, а не отдельным пользователям;

- Устанавливайте максимально строгие разрешения, которые, однако, должны позволять пользователям совершать необходимые действия;
- Организуйте ресурсы общего доступа таким образом, чтобы папки с одинаковым уровнем требований безопасности находились в одной папке. Затем установите общий доступ только к ней, все вложенные папки наследуют настройки безопасности;
- Для папок общего доступа применяйте интуитивно понятные пользователям имена, корректно отображаемые всеми клиентскими операционными системами, используемыми на предприятии.
- Если в общих папках предполагается хранить программы-приложения, то целесообразно поместить их в одну папку – единое место хранения и обновлении я приложений;

Несколько общих папок, доступных членам группы Администраторы, так называемые скрытые Административные общие папки, создаются операционной системой автоматически. Имена этих папок заканчиваются знаком $. Это корневые каталоги каждого тома на жестком диске (C$,D$ и т.д.), папка Admin$ для доступа к системному каталогу, папка Print$ для доступа к файлам драйверов принтеров.

Кроме того, скрытую папку с общим доступом можно создать с целью доступа к ней только тех пользователей, которые будут знать имя скрытой папки.

Получить доступ к общим папкам других компьютеров можно используя компоненты Сетевое окружение, Мой компьютер, Мастер добавления в сетевое окружении и команду выполнить (Run).

Соединение с общей папкой через Сетевое окружение выполняется двойным щелчком по ресурсу, к которому необходимо получить доступ. Если общий ресурс отсутствует в списке доступных, выберите значок Добавить новый элемент в сетевое окружение и укажите адрес подключаемого ресурса.

Соединение с общей папкой через компонент Мой компьютер выполняется через меню Сервис этого компонента в пункте Подключить сетевой диск при указании пути к общему ресурсу. Если необходимо пользоваться этим соединением постоянно, нужно чтобы флажок Восстанавливать при входе в систему был установлен. Соединение будет доступно в разделе Сетевые диске окна Мой компьютер.

Для соединения с общей папкой с помощью команды Выполнить щелкните Пуск, затем Выполнить и введите путь к папке в формате UNC(\\имя_ компьютера\имя_общей папки).

Рассмотрим, как пользоваться средствами установки разрешений файловой системы и общего доступа.

После выбора объекта, для которого будет выполняться настройка разрешений файловой системы, в диалоговом окне свойств файла или папки необходимо выбрать вкладку Безопасность, показанную на рисунке 2.

В данном случае показано, что для папки Авиатор для группы Администраторы установлены разрешения уровня Полный доступ, а для группы Все разрешения ограничены на уровне Чтение.

При установке разрешений в списке групп можно заметить имена так называемых встроенных системных групп, невидимых при использовании оснасток для управления группами и пользователями. Эти группы не имеют определенных членств, которые можно назначить или изменить, но в них система включает различных пользователей в различное время, в зависимости от того, каким образом пользователь получает доступ к системе или ресурсам.

В данном случае имеется в виду группа Все, в которую во время своей работы входят все, кто получил доступ к компьютеру или домену.

Разрешения можно не только устанавливать, но запрещать. Запрет имеет больший приоритет, чем разрешение. Запрет разрешений как метод контроля ресурсов Microsoft применять не рекомендует, и он используется, в основном, для дополнительной настройки разрешений конкретным пользователям, в отличие от разрешений для других пользователей группы.

Рассмотренные разрешения называются стандартными и позволяют решить большинство задач, связанных с регулированием уровня доступа групп к ресурсам.

Рисунок 2 Установка разрешений для группы Все

Кнопка Дополнительно служит для задания специальных разрешений. Каждое стандартное разрешение состоит из нескольких специальных, например стандартное разрешение Запись состоит из шести специальных разрешений: создание файлов/запись данных, Создание папок/дозапись данных, запись атрибутов, Запись дополнительных атрибутов, чтение разрешений, синхронизация. Специальные разрешения можно использовать для более тонкой настройки в нестандартных ситуациях.

В окне специальных разрешений имеются закладки Аудит, Владелец и Эффективные разрешения.

Аудит - это процесс, позволяющий фиксировать события, происходящие в системе и имеющие отношения к безопасности. На данной вкладке производится выбор пользователя или группы, для которых данная папка (или файл) будет объектом аудита.

Закладка Владелец обеспечивает такое свойство безопасности, как право владения объектом файловой системы. Администратор всегда может стать владельцем любого объекта файловой системы, любой пользователь является владельцем созданных им объектов и, если локальные или доменные политики безопасности разрешат, пользователь может назначать себя владельцем других файлов и папок.

Подробное рассмотрение вопросов владения выходит за рамки данного пособия, однако отметим, что многие операции с файлами и папками, например смена разрешений, шифрование и дешифрование привязаны к факту владения данным объектом.

Список управления доступом (ACL) хранится на диске NTFS для каждого файла или папки. В нем перечислены пользователи и группы, для которых установлены разрешения для файла или папки, а также сами назначенные разрешения.

Каждому пользователю или группе могут быть установлены множественные разрешения через участие в нескольких группах с разным набором разрешений. В этом случае действуют эффективные разрешения – пользователь обладает всеми назначенными ему разрешениями.

Действует приоритет разрешений для файлов над разрешениями для папок и приоритет запрещения над разрешением.

Разрешения, назначенные родительской папке, по умолчанию наследуются всеми подпапками и файлами, содержащимися в папках. Однако есть возможность предотвратить наследование для любой вложенной папки и в этом случае эта папка сама становится родительской для вложенных в нее папок.

Если папка предоставлена для общего доступа, то на нее распространяются разрешения двух видов:

- разрешения файловой системы, установленные для пользователей данного компьютера;
- разрешения общего доступа, объявленные для пользователей, получивших доступ через сеть.

Обычно для папок общего доступа задают разрешения полного доступа, а ограничения вводят установкой разрешений NTFS.

В этом случае действует объединение разрешений NTFS и разрешений для общей папки, при котором наиболее строгое разрешение имеет приоритет над другими.

Задание

1. Создайте папку, в которую поместите текстовый файл и приложение в виде файла с расширением exe. Например, одну из стандартных программ Windows, такую как notepad.exe(Блокнот).
2. Установите для этой папки разрешения полного доступа для одного из пользователей группы администраторы, и ограниченные разрешения для пользователя с ограниченной учетной записью.
3. Выполните различные действия с папкой и файлами для обеих учетных записей и установите, как действуют ограничения, связанные с назначением уровня доступа ниже, чем полный доступ.
4. Установите общий доступ к папке и подключитесь к ней через сеть с другого компьютера.
5. Установите разрешения общего доступа так, чтобы администратор не имел ограничений, а пользователь имел ограниченный уровень доступа.
6. Экспериментально убедитесь в правилах объединения разрешений NTFS и разрешений общего доступа.
7. Составьте отчет о проведенных экспериментах.
8. Разработайте стратегию регулирования безопасности при коллективном доступе к общим папкам для различных групп пользователей.

Контрольные вопросы

1. Какое из следующих разрешений NTFS для папок позволяет вам удалять папку?

- Чтение
- Чтение и выполнение
- Изменение
- Администрирование

2. Какое разрешение NTFS для файлов следует установить для файла, если вы позволяете пользователям удалять файл, но не позволяете становиться владельцами файла?
3. Какие объекты по умолчанию наследуют разрешения, установленные для родительской папки?
4. Кто может устанавливать разрешения для отдельных пользователей и групп? (выберите все правильные ответы)

- Члены группы Администраторы
- Члены группы Опытные пользователи
- Пользователи, обладающие разрешением Полный доступ
- Владельцы файлов и папок

5. Какой из следующих вкладок диалогового окна свойств файла или папки следует воспользоваться для установки или изменений разрешений NTFS:

- Дополнительно
- Разрешения
- Безопасность
- Общие

6. Если вы хотите, чтобы пользователь или группа не имела доступа к определенной папке или файлу, следует ли запретить разрешения для этой папки или файла?

Лабораторная работа №12.
Защита трафика туннелированием SSH.

Теоретические сведения.

SSH-туннелирование — это метод транспортировки произвольных сетевых данных по зашифрованному SSH-соединению. Его можно использовать для добавления шифрования в устаревшие приложения. Он также может использоваться для реализации VPN (виртуальных частных сетей) и доступа к службам интрасети (частная корпоративная сеть) через брандмауэры.

SHH-туннель Windows использует порт 22, чтобы обеспечить шифрование данных, передаваемых через общедоступную сеть (например, Интернет), тем самым предоставляя функции VPN.

Туннель через SSH является стандартом для безопасного удаленного входа в систему и передачи файлов по ненадежным сетям.

Трафик направлен внутри зашифрованного SSH-соединения, чтобы он не мог прослушиваться или перехватываться, пока находится в пути. SSH-туннелирование позволяет добавить сетевую безопасность к устаревшим приложениям, которые не поддерживают шифрование.

Безопасное соединение по ненадежной сети устанавливается между клиентом SSH и SSH-сервером. Это SSH-соединение зашифровывается, защищает конфиденциальность и целостность и аутентифицирует связующие стороны.

Соединение SSH используется приложением для подключения к серверу приложений. При активированном туннелировании приложение связывается с портом на локальном хосте, который слушает клиент SSH. Затем клиент SSH перенаправляет приложение поверх своего зашифрованного туннеля на сервер. Последний подключается к фактическому серверу приложений — обычно на том же компьютере или в том же центре обработки данных, что и сервер SSH. Таким образом, связь приложения защищена без необходимости изменения рабочих процессов приложений или конечных пользователей.

Типы переадресации портов

Переадресация портов — это широко поддерживаемая функция, обнаруживаемая во всех основных клиентах и серверах SSH. С функцией перенаправления портов SSH можно осуществлять передачу различных типов интернет-трафика через сеть. Это используется для того, чтобы избежать сетевой слежки или с целью обхода неправильно настроенных маршрутизаторов в Интернете.

Существует **три типа переадресации** портов с SSH:

- локальная — соединения с SSH-клиента перенаправляются на SSH-сервер, а затем на целевой сервер;
- удаленная — соединения с SSH-сервера перенаправляются через SSH-клиент, а затем на целевой сервер;
- динамическая — соединения из различных программ пересылаются через SSH-клиент, затем через SSH-сервер и, наконец, на несколько целевых серверов.

Локальная переадресация портов является наиболее распространенным типом и, в частности, позволяет обойти брандмауэр компании, который блокирует "Википедию".

Неоспоримое преимущество SSH-туннелей заключается в том, что они зашифрованы. Никто не увидит, какие сайты вы посещаете — будут видны только SSH-соединения с сервером

Удаленная переадресация портов встречается реже. Позволяет подключиться с вашего SSH-сервера к компьютеру в интрасети вашей компании.

Динамическая переадресация портов используется также нечасто. Позволяет обойти брандмауэр компании, который полностью блокирует доступ к Интернету. Требует много работы для настройки, и обычно проще использовать локальную переадресацию портов для определенных сайтов, к которым вы хотите получить доступ.

Удаленная переадресация портов

Теперь объясним на реальном примере работу удаленной переадресации. Допустим, вы разрабатываете приложение Rails на своей локальной машине, и хотите показать его другу. К сожалению, ваш интернет-провайдер не предоставил вам публичный IP-адрес, поэтому невозможно напрямую подключиться к ПК через Интернет.

Иногда это можно решить, настроив NAT (трансляция сетевых адресов) на вашем маршрутизаторе, но это не всегда работает, и для этого требуется изменить конфигурацию вашего маршрутизатора, что не всегда желательно. Это решение также не работает, если у вас нет доступа администратора в вашей сети.

Чтобы устранить эту проблему, понадобится другой компьютер, который является общедоступным и имеет доступ к SSH. Это может быть любой сервер в Интернете, если вы можете подключиться к нему. Мы создадим SSH-туннель, который откроет новый порт на сервере и подключит его к локальному порту на вашем компьютере:

Риски

В качестве полезной вещи, несомненно, выступает SSH-туннелирование. Оно включает риски, которые необходимо решать корпоративным ИТ-отделам безопасности. Соединения бесплатных SSH-туннелей защищены сильным шифрованием. Это делает их содержимое невидимым для большинства развертываемых решений сетевого мониторинга и фильтрации трафика. Эта невидимость несет значительный риск, если она используется для вредоносных целей, таких как фильтрация данных. Киберпреступники или вредоносное ПО могут использовать SSH-туннели для скрытия своих несанкционированных сообщений или для извлечения похищенных данных из целевой сети

SSH сервер и клиент

Популярный SSH сервер

- Free SSHd

Как только вы запустили SSH сервер, вам понадобится SSH клиент для Windows. Вот несколько самых популярных SSH клиентов для Windows:

- PuTTY
- Van Dyke - SecureCRT (коммерческий)
-

Ход работы:

1) установить FreeSSHd - SSH сервер в Windows 7?

1. Скачать сервер freeSSHd v.1.3.1 с официального сайта http://www.freesshd.com/.

Принять все параметры по умолчанию для установки и нажать «Установить», чтобы начать процесс.

2. Выбрать создание частных ключей для SSH сервера.

3. Запустить SSHd в качестве службы.

Запустив FreeSSHd в качестве службы, он будет доступен независимо от того, вошли вы в консоль или нет.

4. Запустите ярлык FreeSSHd на рабочем столе, чтобы настроить SSH сервер.

Заходим в свойства (settings) и выполняем основные настройки. Выглядит окно настроек следующим образом:

5. Приложение FreeSSHd может предложить следующее:

- Возможности SSH сервера и Telnet сервера
- Опции запуска SSHd только в определенных интерфейсах
- Различные способы аутентификации, включая интегрированную NTLM аутентификацию на Windows AD
- Различные способы шифрования, включая AES 128, AES 256, 3DES, Blowfish и т.д.
- Опция создания защищенного туннеля в соединении

- Опциональный Secure FTP (sFTP) – для безопасного FTP, смотреть FreeFTPd website
- Возможность управлять пользователями и ограничивать доступ к безопасной оболочке, безопасному туннелю и безопасному FTP
- Возможность предоставлять доступ только определенным узлам или подсетям
- Возможность регистрировать в журнал все подключения и команды, выполненные через FreeSSHd
- Просмотр пользователей, подключенных в данный момент
- Автоматическое обновление FreeSSHd

6. Далее нам нужно создать пользователя, для этого переходим на вкладкуUsers.

Я решил установить логин для моей учетной записи локального администратора Windows. Я установил авторизацию на NTLM. Так в базе данных FreeSSHd нет локального пароля, и если пароль администратора измениться в базе данных локальной учетной записи Windows, вам не придется менять пароль в базе данных учтенной записи FreeSSHd.

Разрешим пользователю только возможность использовать Shell.

7. Далее, что нужно было сделать для того, чтобы позволить себе вход, это открыть исключения брандмауэра Windows. Хотя можно вообще отключить брандмауэр Windows вместо того, чтобы открывать порты, конечно, самой безопасной опцией было оставить брандмауэр включенным и разрешить исключение для SSH ' TCP порт 22.
   Добавьте в исключение брандмауэра порт 22.

2) Протестируем подключение использую клиента SSH PuTTY.
   Он бесплатный, достаточно простой в использовании и содержит весь необходимый функционал

   1. Запускаем программу и видим следующее:

   2. Здесь вам необходимо ввести имя сервера или его ip-адрес в поле Host Name и название сессии в Saved Sessions. Можете назвать сессию аналогично имени сервера, чтобы не путаться.

После того как данные будут введены, нажмите кнопку Save, чтобы сохранить их, и двойным щелчком мыши по названию сессии начните подключение к серверу.

   3. На следующем шаге вас попросят ввести логин и пароль от входа на ваш сервер. По пользовательским данным вас могут и не пустить. Если вы столкнулись с аналогичной ситуацией, то зайдите на свой сервер и отредактируйте права пользователя. Вам требуется поставить галочку в пункте «shell», если она отсутствует.

3) Через интерфейс PuTTY можно зайти в любой каталог на сервере, создать, удалить, переместить или загрузить файл, создать новую папку и многое другое. Но для этого вам потребуется знание определенных команд.

SSH команды

Всего SSH команд достаточно много — около 50-60 точно. Основные из них:

ls — отобразить файлы и папки;

- cd — перейти в корневой каталог;
- cd .. — перейти в каталог уровнем выше;
- cd papka1/papka2 — перейти в указанную папку;
- pwd — показать путь к текущему расположению;
- cp — копировать файл;
- rm — удалить файл;
- mv — переместить файл;
- mkdir — создать новую папку;
- rmdir — удалить папку;
- get — загрузить файл на локальный компьютер;
- put — загрузить файл на удаленный компьютер;
- exit — завершить сессию и выйти из программы;
- help — список всех команд.

Выполните эти действия.

Лабораторная работа №13.
Изучение механизма шифрования IPSec.

Теоретические сведения.

Необходимость защиты данных

В конце шестидесятых годов американское агентство перспективных исследований в обороне DARPA приняло решение о создании экспериментальной сети под названием ARPANet. В семидесятых годах ARPANet стала считаться действующей сетью США, и через эту сеть можно было получить доступ к ведущим университетским и научным центрам США. В начале восьмидесятых годов началась стандартизация языков программирования, а затем и протоколов взаимодействия сетей. Результатом этой работы стала разработка семиуровневой модели сетевого взаимодействия ISO/OSI и семейства протоколов TCP/IP, которое стало основой для построения как локальных, так и глобальных сетей.

Базовые механизмы информационного обмена в сетях TCP/IP были в целом сформированы в начале восьмидесятых годов, и были направлены прежде всего на обеспечение доставки пакетов данных между различными операционными системами с использованием разнородных каналов связи. Несмотря на то, что идея создания сети ARPANet (впоследствии превратившейся в современный Интернет) принадлежала правительственной оборонной организации, фактически сеть зародилась в исследовательском мире, и наследовала традиции открытости академического сообщества. Ещё до коммерциализации Интернета (которая произошла в середине девяностых годов) многие авторитетные исследователи отмечали проблемы, связанные с безопасностью стека протоколов TCP/IP. Основные концепции протоколов TCP/IP не полностью удовлетворяют (а в ряде случаев и противоречат) современным представлениям о компьютерной безопасности.

До недавнего времени сеть Интернет использовалась в основном для обработки информации по относительно простым протоколам: электронная почта, передача файлов, удалённый доступ. Сегодня, благодаря широкому распространению технологий WWW, всё активнее применяются средства распределённой обработки мультимедийной информации. Одновременно с этим растёт объём данных, обрабатываемых в средах клиент/сервер и предназначенных для одновременного коллективного доступа большого числа абонентов. Разработано несколько протоколов прикладного уровня, обеспечивающих информационную безопасность таких приложений, как электронная почта (PEM, PGP и т.п.), WWW (Secure HTTP, SSL и т.п.), сетевое управление (SNMPv2 и т.п.). Однако наличие средств обеспечения безопасности в базовых протоколах семейства TCP/IP позволит осуществлять информационный обмен между широким спектром различных приложений и сервисных служб.

Краткая историческая справка появления протокола

В 1994 году Совет по архитектуре Интернет (IAB) выпустил отчет "Безопасность архитектуры Интернет". В этом документе описывались основные области применения дополнительных средств безопасности в сети Интернет, а именно защита от несанкционированного мониторинга, подмены пакетов и управления потоками данных. В числе первоочередных и наиболее важных защитных мер указывалась необходимость разработки концепции и основных механизмов обеспечения целостности и конфиденциальности потоков данных. Поскольку изменение базовых протоколов семейства TCP/IP вызвало бы полную перестройку сети Интернет, была поставлена задача обеспечения безопасности информационного обмена в открытых телекоммуникационных сетях на базе существующих протоколов. Таким образом, начала создаваться спецификация Secure IP, дополнительная по отношению к протоколам IPv4 и IPv6.

Архитектура IPSec

IP Security — это комплект протоколов, касающихся вопросов шифрования, аутентификации и обеспечения защиты при транспортировке IP-пакетов; в его состав сейчас входят почти 20 предложений по стандартам и 18 RFC.

Спецификация IP Security (известная сегодня как IPsec) разрабатывается Рабочей группой IP Security Protocol IETF. Первоначально IPsec включал в себя 3 алгоритмо-независимые базовые спецификации, опубликованные в качестве RFC-документов "Архитектура безопасности IP", "Аутентифицирующий заголовок (AH)", "Инкапсуляция зашифрованных данных (ESP)" (RFC1825, 1826 и 1827). Необходимо заметить, что в ноябре 1998 года Рабочая группа IP Security Protocol предложила новые версии этих спецификаций, имеющие в настоящее время статус предварительных стандартов, это RFC2401 — RFC2412. Отметим, что RFC1825-27 на протяжении уже нескольких лет считаются устаревшими и реально не используются. Кроме этого, существуют несколько алгоритмо-зависимых спецификаций, использующих протоколы MD5, SHA, DES.

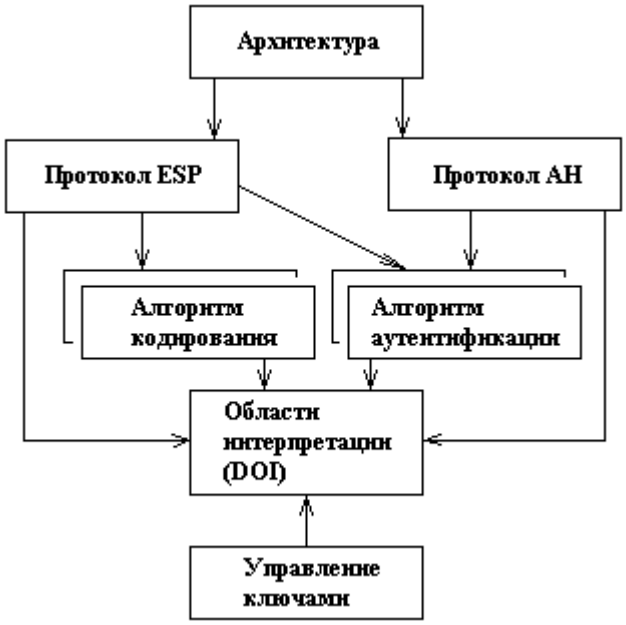


Рис. 1 – Архитектура IPSec.

Рабочая группа IP Security Protocol разрабатывает также и протоколы управления ключевой информацией. В задачу этой группы входит разработка Internet Key Management Protocol (IKMP), протокола управления ключами прикладного уровня, не зависящего от используемых протоколов обеспечения безопасности. В настоящее время рассматриваются концепции управления ключами с использованием спецификации Internet Security Association and Key Management Protocol (ISAKMP) и протокола Oakley Key Determination Protocol. Спецификация ISAKMP описывает механизмы согласования атрибутов используемых протоколов, в то время как протокол Oakley позволяет устанавливать сессионные ключи на компьютеры сети Интернет. Ранее рассматривались также возможности использования механизмов управления ключами протокола SKIP, однако сейчас такие возможности реально практически нигде не используются. Создаваемые стандарты управления ключевой информацией, возможно, будут поддерживать Центры распределения ключей, аналогичные используемым в системе Kerberos. Протоколами ключевого управления для IPSec на основе Kerberos сейчас занимается относительно новая рабочая группа KINK (Kerberized Internet Negotiation of Keys).

Гарантии целостности и конфиденциальности данных в спецификации IPsec обеспечиваются за счет использования механизмов аутентификации и шифрования соответственно. Последние, в свою очередь, основаны на предварительном согласовании сторонами информационного обмена

т.н. "контекста безопасности" – применяемых криптографических алгоритмов, алгоритмов управления ключевой информацией и их параметров. Спецификация IPsec предусматривает возможность поддержки сторонами информационного обмена различных протоколов и параметров аутентификации и шифрования пакетов данных, а также различных схем распределения ключей. При этом результатом согласования контекста безопасности является установление индекса параметров безопасности (SPI), представляющего собой указатель на определенный элемент внутренней структуры стороны информационного обмена, описывающей возможные наборы параметров безопасности.

По сути, IPSec, который станет составной частью IPv6, работает на третьем уровне, т. е. на сетевом уровне. В результате передаваемые IP-пакеты будут защищены прозрачным для сетевых приложений и инфраструктуры образом. В отличие от SSL (Secure Socket Layer), который работает на четвертом (т. е. транспортном) уровне и теснее связан с более высокими уровнями модели OSI, IPSec призван обеспечить низкоуровневую защиту.

Уровни TCP/IP	Уровни ISO/OSI
4. Прикладных программ	7. Прикладных программ 6. Представление данных
3. Транспортный	5. Сеансовый 4. Транспортный
2. Межсетевой	3. Сетевой
1. Доступа к сети	2. Канальный 1. Физический

Рис. 2 — Модель OSI/ISO.

К IP-данным, готовым к передаче по виртуальной частной сети, IPSec добавляет заголовок для идентификации защищенных пакетов. Перед передачей по Internet эти пакеты инкапсулируются в другие IP-пакеты. IPSec поддерживает несколько типов шифрования, в том числе Data Encryption Standard (DES) и Message Digest 5 (MD5).

Чтобы установить защищенное соединение, оба участника сеанса должны иметь возможность быстро согласовать параметры защиты, такие как алгоритмы аутентификации и ключи. IPSec поддерживает два типа схем управления ключами, с помощью которых участники могут согласовать параметры сеанса. Эта двойная поддержка в свое время вызвала определенные трения в IETF Working Group.

С текущей версией IP, IPv4, могут быть использованы или Internet Secure Association Key Management Protocol (ISAKMP), или Simple Key Management for Internet Protocol. С новой версией IP, IPv6, придется использовать ISAKMP, известный сейчас как IKE, хотя не исключается возможность использования SKIP. Однако, следует иметь в виду, что SKIP уже давно не рассматривается как кандидат управления ключами, и был исключён из списка возможных кандидатов ещё в 1997 г.

Заголовок AH

Аутентифицирующий заголовок (AH) является обычным опциональным заголовком и, как правило, располагается между основным заголовком пакета IP и полем данных. Наличие AH никак не влияет на процесс передачи информации транспортного и более высокого уровней. Основным и единственным назначением AH является обеспечение защиты от атак, связанных с несанкционированным изменением содержимого пакета, и в том числе от подмены исходного

адреса сетевого уровня. Протоколы более высокого уровня должны быть модифицированы в целях осуществления проверки аутентичности полученных данных.

Формат AH достаточно прост и состоит из 96-битового заголовка и данных переменной длины, состоящих из 32-битовых слов. Названия полей достаточно ясно отражают их содержимое: Next Header указывает на следующий заголовок, Payload Len представляет длину пакета, SPI является указателем на контекст безопасности и Sequence Number Field содержит последовательный номер пакета.

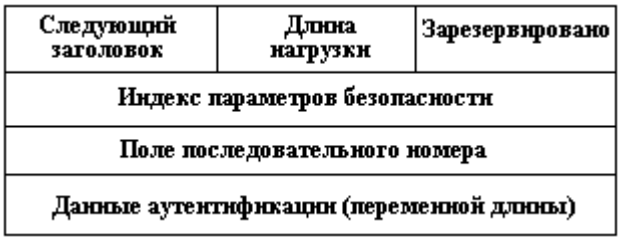


Рис. 3 — Формат заголовка AH.

Последовательный номер пакета был введен в AH в 1997 году в ходе процесса пересмотра спецификации IPsec. Значение этого поля формируется отправителем и служит для защиты от атак, связанных с повторным использованием данных процесса аутентификации. Поскольку сеть Интернет не гарантирует порядок доставки пакетов, получатель должен хранить информацию о максимальном последовательном номере пакета, прошедшего успешную аутентификацию, и о получении некоторого числа пакетов, содержащих предыдущие последовательные номера (обычно это число равно 64).

В отличие от алгоритмов вычисления контрольной суммы, применяемых в протоколах передачи информации по коммутируемым линиям связи или по каналам локальных сетей и ориентированных на исправление случайных ошибок среды передачи, механизмы обеспечения целостности данных в открытых телекоммуникационных сетях должны иметь средства защиты от внесения целенаправленных изменений. Одним из таких механизмов является специальное применение алгоритма MD5: в процессе формирования AH последовательно вычисляется хэш-функция от объединения самого пакета и некоторого предварительно согласованного ключа, а затем от объединения полученного результата и преобразованного ключа. Данный механизм применяется по умолчанию в целях обеспечения всех реализаций IPv6, по крайней мере, одним общим алгоритмом, не подверженным экспортным ограничениям.

Заголовок ESP

В случае использования инкапсуляции зашифрованных данных заголовок ESP является последним в ряду опциональных заголовков, "видимых" в пакете. Поскольку основной целью ESP является обеспечение конфиденциальности данных, разные виды информации могут требовать применения существенно различных алгоритмов шифрования. Следовательно, формат ESP может претерпевать значительные изменения в зависимости от используемых криптографических алгоритмов. Тем не менее, можно выделить следующие обязательные поля: SPI, указывающее на контекст безопасности и Sequence Number Field, содержащее последовательный номер пакета. Поле "ESP Authentication Data" (контрольная сумма), не является обязательным в заголовке ESP. Получатель пакета ESP расшифровывает ESP заголовок и использует параметры и данные применяемого алгоритма шифрования для декодирования информации транспортного уровня.

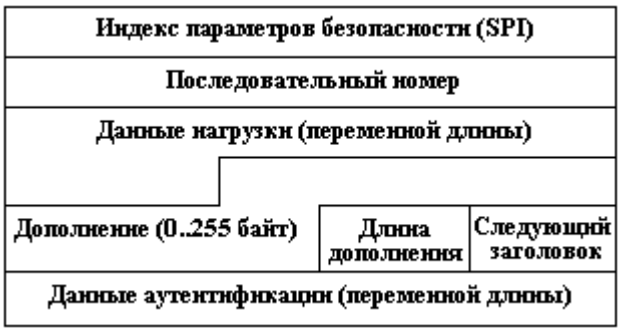


Рис. 4 — Формат заголовка ESP.

Различают два режима применения ESP и AH (а также их комбинации) — транспортный и туннельный.

Транспортный режим

Транспортный режим используется для шифрования поля данных IP пакета, содержащего протоколы транспортного уровня (TCP, UDP, ICMP), которое, в свою очередь, содержит информацию прикладных служб. Примером применения транспортного режима является передача электронной почты. Все промежуточные узлы на маршруте пакета от отправителя к получателю используют только открытую информацию сетевого уровня и, возможно, некоторые опциональные заголовки пакета (в IPv6). Недостатком транспортного режима является отсутствие механизмов скрытия конкретных отправителя и получателя пакета, а также возможность проведения анализа трафика. Результатом такого анализа может стать информация об объемах и направлениях передачи информации, области интересов абонентов, расположение руководителей.

Туннельный режим

Туннельный режим предполагает шифрование всего пакета, включая заголовок сетевого уровня. Туннельный режим применяется в случае необходимости скрытия информационного обмена организации с внешним миром. При этом, адресные поля заголовка сетевого уровня пакета, использующего туннельный режим, заполняются межсетевым экраном организации и не содержат информации о конкретном отправителе пакета. При передаче информации из внешнего мира в локальную сеть организации в качестве адреса назначения используется сетевой адрес межсетевого экрана. После расшифровки межсетевым экраном начального заголовка сетевого уровня пакет направляется получателю.

Security Associations

Security Association (SA) — это соединение, которое предоставляет службы обеспечения безопасности трафика, который передаётся через него. Два компьютера на каждой стороне SA хранят режим, протокол, алгоритмы и ключи, используемые в SA. Каждый SA используется только в одном направлении. Для двунаправленной связи требуется два SA. Каждый SA реализует один режим и протокол; таким образом, если для одного пакета необходимо использовать два протокола (как например AH и ESP), то требуется два SA.

Политика безопасности

Политика безопасности хранится в SPD (База данных политики безопасности). SPD может указать для пакета данных одно из трёх действий: отбросить пакет, не обрабатывать пакет с помощью IPSec, обработать пакет с помощью IPSec. В последнем случае SPD также указывает, какой SA необходимо использовать (если, конечно, подходящий SA уже был создан) или указывает, с какими параметрами должен быть создан новый SA.

SPD является очень гибким механизмом управления, который допускает очень хорошее управление обработкой каждого пакета. Пакеты классифицируются по большому числу полей, и SPD может проверять некоторые или все поля для того, чтобы определить соответствующее действие. Это может привести к тому, что весь трафик между двумя машинами будет передаваться при помощи одного SA, либо отдельные SA будут использоваться для каждого приложения, или даже для каждого TCP соединения.

ISAKMP/Oakley

Протокол ISAKMP определяет общую структуру протоколов, которые используются для установления SA и для выполнения других функций управления ключами. ISAKMP поддерживает несколько Областей Интерпретации (DOI), одной из которых является IPSec-DOI. ISAKMP не определяет законченный протокол, а предоставляет "строительные блоки" для различных DOI и протоколов обмена ключами.

Протокол Oakley — это протокол определения ключа, использующий алгоритм замены ключа Диффи-Хеллмана. Протокол Oakley поддерживает идеальную прямую безопасность (Perfect Forward Secrecy — PFS). Наличие PFS означает невозможность расшифровки всего траффика при компрометации любого ключа в системе.

IKE

IKE — протокол обмена ключами по умолчанию для ISAKMP, на данный момент являющийся единственным. IKE находится на вершине ISAKMP и выполняет, собственно, установление как ISAKMP SA, так и IPSec SA. IKE поддерживает набор различных примитивных функций для использования в протоколах. Среди них можно выделить хэш-функцию и псевдослучайную функцию (PRF).

Хэш-функция — это функция, устойчивая к коллизиям. Под устойчивостью к коллизиям понимается тот факт, что невозможно найти два разных сообщения m_1 и m_2, таких, что $H(m_1)=H(m_2)$, где H — хэш функция.

Что касается псеводслучайных функций, то в настоящее время вместо специальных PRF используется хэш функция в конструкции HMAC (HMAC — механизм аутентификации сообщений с использованием хэш функций). Для определения HMAC нам понадобится криптографическая хэш функция (обозначим её как H) и секретный ключ K. Мы предполагаем, что H является хэш функцией, где данные хэшируются с помощью процедуры сжатия, последовательно применяемой к последовательности блоков данных. Мы обозначим за B длину таких блоков в байтах, а длину блоков, полученных в результате хэширования — как L (L<B). Ключ K может иметь длину, меньшую или равную B. Если приложение использует ключи большей длины, сначала мы должны хэшировать сам ключ с использованием H, и только после этого использовать полученную строку длиной L байт, как ключ в HMAC. B обоих случаях рекомендуемая минимальная длина для K составляет L байт. Определим две следующие различные строки фиксированной длины:

ipad = байт 0x36, повторённый B раз;

opad = байт 0x5C, повторённый B раз.

Для вычисления HMAC от данных 'text' необходимо выполнить следующую операцию:

H(K XOR opad, H(K XOR ipad, text))

Из описания следует, что IKE использует для аутентификации сторон HASH величины. Отметим, что под HASH в данном случае подразумевается исключительно название Payload в ISAKMP, и это название не имеет ничего общего со своим содержимым.

Атаки на AH, ESP и IKE.

Все виды атак на компоненты IPSec можно разделить на следующие группы: атаки, эксплуатирующие конечность ресурсов системы (типичный пример — атака "Отказ в обслуживании", Denial-of-service или DOS-атака), атаки, использующие особенности и ошибки конкретной реализации IPSec и, наконец, атаки, основанные на слабостях самих протоколов. AH и ESP. Чисто криптографические атаки можно не рассматривать — оба протокола определяют понятие "трансформ", куда скрывают всю криптографию. Если используемый криптоалгоритм стоек, а определенный с ним трансформ не вносит дополнительных слабостей (это не всегда так, поэтому правильнее рассматривать стойкость всей системы — Протокол-Трансформ-Алгоритм), то с этой стороны все нормально. Что остается? Replay Attack — нивелируется за счет использования Sequence Number (в одном единственном случае это не работает — при использовании ESP без аутентификации и без AH). Далее, порядок выполнения действий (сначала шифрация, потом аутентификация) гарантирует быструю отбраковку "плохих" пакетов (более того, согласно последним исследованиям в мире криптографии, именно такой порядок действий наиболее безопасен, обратный порядок в некоторых, правда очень частных случаях, может привести к потенциальным дырам в безопасности; к счастью, ни SSL, ни IKE, ни другие распространенные протоколы с порядком действий "сначала аутентифицировать, потом зашифровать", к этим частным случаям не относятся, и, стало быть, этих дыр не имеют). Остается Denial-Of-Service атака. Как известно, это атака, от которой не существует полной защиты. Тем не менее, быстрая отбраковка плохих пакетов и отсутствие какой-либо внешней реакции на них (согласно RFC) позволяют более-менее хорошо справляться с этой атакой. В принципе, большинству (если не всем) известным сетевым атакам (sniffing, spoofing, hijacking и т.п.) AH и ESP при правильном их применении успешно противостоят. С IKE несколько сложнее. Протокол очень сложный, тяжел для анализа. Кроме того, в силу опечаток (в формуле вычисления HASH_R) при его написании и не совсем удачных решений (тот же HASH_R и HASH_I) он содержит несколько потенциальных "дыр" (в частности, в первой фазе не все Payload в сообщении аутентифицируются), впрочем, они не очень серьезные и ведут, максимум, к отказу в установлении соединения.От атак типа replay, spoofing, sniffing, hijacking IKE более-менее успешно защищается. С криптографией несколько сложнее, — она не вынесена, как в AH и ESP, отдельно, а реализована в самом протоколе. Тем не менее, при использовании стойких алгоритмов и примитивов (PRF), проблем быть не должно. В какой-то степени можно рассматривать как слабость IPsec то, что в качестве единственного обязательного к реализации криптоалгоритма в нынешних спецификациях указывается DES (это справедливо и для ESP, и для IKE), 56 бит ключа которого уже не считаются достаточными. Тем не менее, это чисто формальная слабость — сами спецификации являются алгоритмо-независимыми, и практически все известные вендоры давно реализовали 3DES (а некоторые уже и AES).Таким образом, при правильной реализации, наиболее "опасной" атакой остается Denial-Of-Service.

Оценка протокола

Протокол IPSec получил неоднозначную оценку со стороны специалистов. С одной стороны, отмечается, что протокол IPSec является лучшим среди всех других протоколов защиты передаваемых по сети данных, разработанных ранее (включая разработанный Microsoft PPTP). По мнению другой стороны, присутствует чрезмерная сложность и избыточность протокола. Так, Niels Ferguson и Bruce Schneier в своей работе "A Cryptographic Evaluation of IPsec" отмечают, что они обнаружили серьёзные проблемы безопасности практически во всех главных компонентах IPsec. Эти авторы также отмечают, что набор протоколов требует серьёзной доработки для того, чтобы он обеспечивал хороший уровень безопасности. В работе приведено

описание ряда атак, использующих как слабости общей схемы обработки данных, так и слабости криптографических алгоритмов.

Задание.

На основе пройденного материала составить отчѐт, содержащий ответы на следующие вопросы:

1. С какой целью был разработан IPSec?
2. Возможности IPSec.
3. Протоколы IPSec.
4. Какая информация хранится в базах данных IPSec?

Лабораторная работа №14.
Мониторинг трафика. Утилиты командной строки.

Цель работы: Изучение основных сетевых утилит и механизма их работы, приобретение навыков использования сетевых утилит для получения необходимой информации о состоянии сети

Теоретические сведения

Утилиты Ping и Traceroute

При работе в Интернет время от времени возникают ситуации, когда нужно определить, работоспособен ли тот или иной канал или узел, а в случае работы с динамическими протоколами маршрутизации выяснить, по какому из каналов вы в данный момент работаете. Используется эта процедура и для оценки вероятности потери пакетов в заданных сегментах сети или каналах. Для решения этих задач удобна программа Ping.

Ping - это процедура, которая базируется на ICMP- и UDP-протоколах пересылки дейтограмм и служит для трассировки маршрутов и проверки работоспособности каналов и узлов (в некоторых программных пакетах эта команда имеет имена trace, hopcheck, tracert или traceroute). Для решения поставленной задачи PING использует отклики протокола ICMP. Применяется PING и при отладке сетевых продуктов. Трассировка может выполняться, например, посредством команды ping -q (пакет PCTCP). При выполнении этой команды ЭВМ сообщит вам Internet адреса всех промежуточных узлов, их имена и время распространения отклика от указанного вами узла. Следует иметь в виду, что трассировка осуществляется непосредственно с помощью IP-протокола (опция записи адресов промежуточных узлов). Ниже приведен пример использования команды Ping. Если вы просто напечатаете команду ping -?, то ЭВМ выдаст на экран справочную таблицу по использованию этой команды:

Формат команды:

ping [-t] [-a] [-n число] [-l размер] [-f] [-i TTL] [-v TOS] [-r число] [-s число]
[[-j список Узлов]: [-k список Узлов]] [-w таймаут] **host_name**

Параметры команды:

-t	Отправка пакетов на указанный узел до команды прерывания. Для вывода статистики и продолжения нажмите «Ctrl+ «Break», для прекращения – «Ctrl+«C»
-a	Определение адресов по именам узлов
-n	Число отправляемых эхо-пакетов
-l	Размер буфера отправки
-f	Установка флага, запрещающего фрагментацию пакета
-i TTL	Задание срока жизни пакета
-v TOS	Задание типа службы
-r	Запись маршрута для указанного числа переходов
-s	Штамп времени для указанного числа переходов
-j	Свободный выбор маршрута по списку узлов
-k	Жесткий выбор маршрута по списку узлов
-w	Таймаут каждого ответа в миллисекундах

При запуске команды посылаются эхо-запросы. Номер последовательности начинается с 0 и увеличивается на единицу каждый раз когда посылается следующий эхо запрос. ping печатает номер последовательности каждого возвращенного пакета, позволяя нам увидеть, потерялся ли пакет, поменялась ли последовательность движения пакетов и был ли пакет продублирован. Так как IP является ненадежным сервисом доставки датаграмм, любое из трех вышеперечисленных условий может появиться при работе программы ping.

Исторически сложилось так, что программа ping посылает эхо запрос один раз в секунду, печатая каждый эхо отклик в момент его возвращения. Однако новые разработки требуют указания опции -s, чтобы программа работала подобным образом. По умолчанию новые реализации посылают только один эхо запрос и выдают сообщение "host is alive" (хост доступен), если эхо отклик получен, или "no answer" (не отвечает), если отклик не получен в течение 20 секунд

Работа программы в глобальных сетях

При работе в глобальных сетях результат может значительно отличаться. Следующий пример был получен в рабочий день после полудня, время, когда Internet обычно довольно загружен:

ping vangogh.cs.berkeley.edu PING vangogh.cs.berkeley.edu: 56 data bytes 64 bytes from (128.32.130.2): icmp_seq=0. time=660. ms 64 bytes from (128.32.130.2): icmp_seq=5. time=1780. ms 64 bytes from (128.32.130.2): icmp_seq=7. time=380. ms 64 bytes from (128.32.130.2): icmp_seq=8. time=420. ms 64 bytes from (128.32.130.2): icmp_seq=9. time=390. ms 64 bytes from (128.32.130.2): icmp_seq=14. time=110. ms 64 bytes from (128.32.130.2): icmp_seq=15. time=170. ms 64 bytes from (128.32.130.2): icmp_seq=16. time=100. ms ^? вводим символ прерывания ---- vangogh.CS.Berkeley.EDU PING Statistics ---- 17 packets transmitted, 8 packets received, 52% packet loss round-trip (ms) min/avg/max = 100/501/1780

При работе в глобальных сетях можно встретиться с дублированием пакетов (один и тот же номер последовательности появляется дважды или несколько раз), также может возникнуть перемешивание номеров последовательности (номер последовательности N+1 появляется перед номером последовательности N).

Когда принимается ICMP эхо отклик, печатается номер последовательности, затем параметр время жизни (TTL) и рассчитанное время возврата. Как видно из примера, приведенного выше, эхо отклики возвращаются в том же порядке, в котором были отправлены (0, 1, 2 и так далее). Эхо запросы и эхо отклики с номерами последовательности 1, 2, 3, 4, 6, 10, 11, 12 и 13 были потеряны. ping может рассчитать время возврата, так как он сохраняет время, когда был отправлен эхо запрос, в разделе данных ICMP сообщения. Когда отклик возвращается, эти данные извлекаются и сравниваются с текущим временем.

Обратите внимание на значительную разницу между величинами времен возврата. (Количество потерянных пакетов, а именно 52%, является ненормальным. Это неприемлимо для Internet даже в рабочие дни после полудня.)

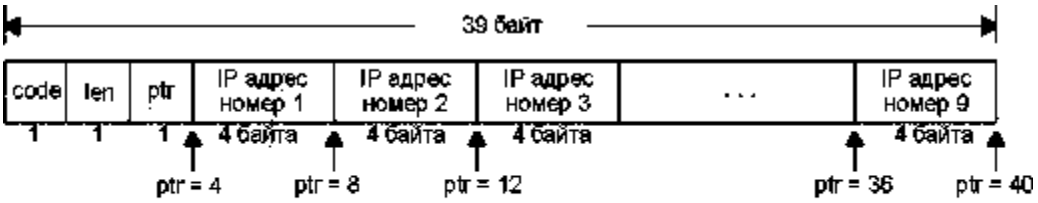
Первая строка вывода содержит IP адрес хоста назначения, даже если было указано имя (vangogh.cs.berkeley.edu). Это означает, что имя было преобразовано в IP адрес. После запуска программы ping проходит несколько секунд, перед тем как появляется первая строка вывода с напечатанным IP адресом, это время необходимо DNS, чтобы определить IP адрес, соответствующий имени хоста.

Опция записи ip маршрута

Программа ping предоставляет возможность просмотреть IP опцию записи маршрута (RR). В большинстве версий программы ping присутствует опция -R, которая включает характеристику записи маршрута. При использовании этой опции ping устанавливает опцию IP записи маршрута (RR) в исходящих датаграммах (которые содержат ICMP эхо запрос). При этом каждый маршрутизатор, который обрабатывает датаграмму, добавляет свой IP адрес в список, находящийся в дополнительном поле. Когда датаграмма достигает конечного пункта назначения, список IP адресов копируется в исходящий ICMP эхо отклик, а все маршрутизаторы на обратном пути также добавляют свои IP адреса в список. Когда ping принимает эхо отклик, печатает список IP адресов.

Как бы просто это не звучало, в действительности, запись маршрута - достаточно сложный процесс. Генерация IP опции RR хостом источником, обработка опции RR промежуточными маршрутизаторами и отражение входящего списка RR из ICMP эхо запроса в исходящий ICMP эхо отклик все это дополнительные и необязательные характеристики. Большинство систем в настоящее время поддерживают эти дополнительные характеристики, однако некоторые системы не отображают список IP адресов.

Самая большая проблема, однако, заключается в ограниченном размере IP заголовка, в который должен поместиться список IP адресов. Из рисунка видно, что поле длины заголовка (header length) в IP заголовке составляет 4 бита, что ограничивает размер IP заголовка в пределах пятнадцати 32-битных слов (60 байт). Так как фиксированный размер IP заголовка составляет 20 байт, а RR опция использует 3 байта для своей установки (что мы опишем ниже), то остается 37 байт (60-20-3) на список адресов, а это, в свою очередь, позволяет поместить туда до 9 IP адресов. На рисунке показан общий формат опции записи маршрута в IP датаграмме.



где,
code - код
len - длина
ptr - указатель

Рисунок 1. Общий формат опции маршрута в IP заголовке

Код (code) - однобайтовое поле, содержащее тип IP опции. Для опции RR установлено значение 7. Длина (len) - это полный размер в байтах опции RR, в данном случае 39. (Несмотря на то, что существует возможность указать опцию RR с размером меньше максимального, ping всегда предоставляет поле опции размером 39 байт, что позволяет записать до 9 IP адресов. Несмотря на то, что существует ограничение в размере опций в IP заголовке, оно, тем не менее, позволяет указать размер меньше максимального.)
Указатель (ptr) - это индекс в 39-байтной опции, который указывает на то, где хранится следующий IP адрес. Его минимальное значение 4, что указывает на первый IP адрес. Когда следующий IP адрес записывается в список, значение ptr меняется следующим образом: 8, 12, 16 и так до 36. После того как записан девятый адрес, ptr устанавливается в значение 40, указывая на то, что список полон. А теперь давайте зададим себе такой вопрос.
Когда маршрутизатор (который по определению имеет несколько интерфейсов) записывает свой IP адрес в список, какой IP адрес он записывает? Это должен быть адрес либо входящего интерфейса, либо исходящего. RFC 791 [Postel 1981a] указывает, что маршрутизатор записывает IP адрес исходящего интерфейса. Исходный хост, на примере ниже, (хост, запустивший ping) получает ICMP эхо отклик с включенной опцией RR, он вносит в список IP адрес своего входящего интерфейса.
Обычный пример
Давайте попробуем запустить программу ping с опцией RR. Мы запустили ping с хоста svr4 на хост slip. Промежуточный роутер (bsdi) обрабатывает датаграмму, следующий вывод будет получен от svr4:

svr4 % ping -R slip PING slip (140.252.13.65): 56 data bytes 64 bytes from 140.252.13.65: icmp_seq=0 ttl=254 time=280 ms RR: bsdi (140.252.13.66) slip (140.252.13.65) bsdi (140.252.13.35) svr4 (140.252.13.34) 64 bytes from 140.252.13.65: icmp_seq=1 ttl=254 time=280 ms (same route) 64 bytes from 140.252.13.65: icmp_seq=2 ttl=254 time=280 ms (same route) ^? --- slip ping statistics --- 3 packets transmitted, 3 packets received, 0% packet loss round-trip min/avg/max = 270/276/280 ms

На рисунке 2 показаны 4 пересылки, через которые проходит пакет (по две в каждом направлении), а также IP адреса добавляемые к списку RR при каждой пересылке.

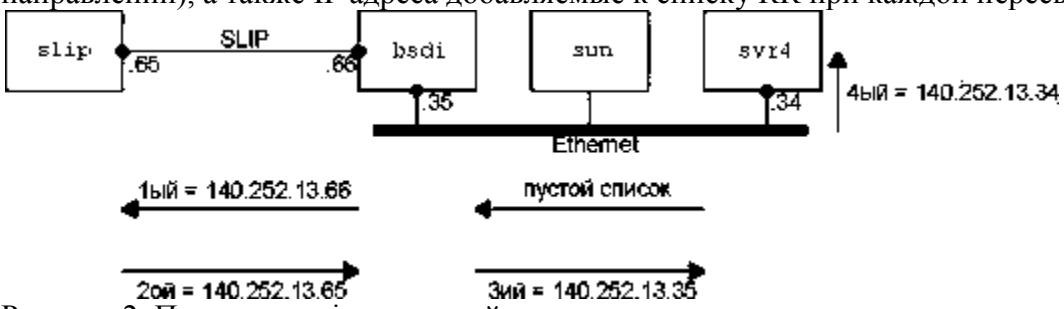


Рисунок 2. Программа ping с опцией записи маршрутизации

Маршрутизатор bsdi добавляет в список разные IP адреса в зависимости от направления движения датаграммы. Он всегда добавляет IP адрес исходящего интерфейса. Однако, когда ICMP эхо отклик достигает системы, которая инициировала запрос (svr4), она добавляет в список IP адрес входящего интерфейса.

Трассировка маршрута

Программа **Traceroute**, написанная Van Jacobson, - отладочное средство, которое позволяет лучше понять устройство протоколов TCP/IP. Обычно две последовательные датаграммы

отправленные от одного и того же источника к одному и тому же пункту назначения проходят по одному и тому же маршруту, однако гарантировать этого невозможно. Traceroute позволяет нам посмотреть маршрут, по которому двигаются IP датаграммы от одного хоста к другому. С помощью Tracert можно воспользоваться IP опцией маршрутизации от источника.
В предыдущей команде была уже рассмотрена опция трассировки маршрута. Зачем нужна отдельная программа?
Во-первых, исторически не все маршрутизаторы поддерживают опцию записи маршрута, из чего следует, что некоторые маршруты становятся неиспользуемыми. (Tracert не требует каких-либо специальных характеристик на промежуточных маршрутизаторах.)
Во-вторых, запись маршрута обычно осуществляется в одном направлении. Отправитель включает опцию, а получатель должен вставить все значения из принятого IP заголовка и каким-либо образом вернуть их отправителю. Большинство реализаций сервера Ping (функция ICMP эхо отклика, встроенная в ядро) отображают входящий RR список, однако при этом удваивается количество записанных IP адресов (путь туда и обратно), помимо этого существует еще несколько ограничений. (Tracert требует только того, чтобы на пункте назначения присутствовал работающий UDP модуль - никаких специальных серверных приложений не требуется.)
Третья и основная причина заключается в том, что размер, предоставляемый для опций в IP заголовке, недостаточен для того, чтобы обработать большинство маршрутов. В поле опций IP заголовка входит всего 9 IP адресов. Если во времена ARPANET этого хватало, на сегодняшний день этого слишком мало.

Формат команды:

tracert [-d] [-h максимальное число] [-j список Узлов] [-w интервал] **host_name**

Праметры:

-d	Без разрешения в имена узлов
-h	Максимальное число прыжков при поиске узла
-j	Свободный выбор маршрута по списку узлов.
-w	Интервал ожидания каждого ответа в миллисекундах.

Tracert использует ICMP и поле TTL в IP заголовке. Поле TTL (время жизни) это 8-битное поле, которое отправитель устанавливает в какое-либо значение. Рекомендуемое исходное значение указано в Assigned Numbers RFC и в настоящее время равно 64. Более старые системы устанавливают это значение в 15 или 32. На примерах работы программы Ping видно,что ICMP эхо отклики часто отправляются с TTL, установленным в максимальное значение - 255.
Каждый маршрутизатор, который обрабатывает датаграмму, уменьшает значение TTL на единицу или на количество секунд, в течение которых маршрутизатор обрабатывал датаграмму. Так как большинство маршрутизаторов задерживает датаграмму меньше чем секунду, поле TTL, как правило, уменьшается на единицу и довольно точно соответствует количеству пересылок.
С помощью поля TTL предотвращается зацикливание датаграммы в петлях маршрутизации. Например, если маршрутизатор вышел из строя или соединение между двумя маршрутизаторами потеряно, может потребоваться некоторое время (от нескольких секунд до нескольких минут), для того чтобы определить, что маршрут потерян и что его необходимо обойти. В это время существует вероятность, что датаграмма будет уничтожена в петле маршрутизации. Чтобы предотвратить потерю датаграммы, поле TTL устанавливается в максимальную величину.
Когда маршрутизатор получает IP датаграмму с TTL равным либо 0, либо 1, он не должен отправлять эту датаграмму дальше. (Хост приемник должен доставить подобную датаграмму в приложение, так как датаграмма не может быть смаршрутизировна. Как правило, системы не должны получать датаграммы с TTL равным 0.) Если такую датаграмму получает маршрутизатор, он уничтожает ее и посылает хосту, который ее отправил ICMP сообщение "время истекло" (time exceeded). Принцип работы Tracert заключается в том, что IP датаграмма, содержащая это ICMP сообщение, имеет в качестве адреса источника IP адрес маршрутизатора.
При работе Tracert на хост назначения отправляется IP датаграмма с TTL, установленным в единицу. Первый маршрутизатор, который должен обработать датаграмму, уничтожает ее (так как TTL равно 1) и отправляет ICMP сообщение об истечении времени (time exceeded). Таким образом, определяется первый маршрутизатор в маршруте. Затем Tracert отправляет датаграмму с TTL равным 2, что позволяет получить IP адрес второго маршрутизатора. Это продолжается до тех пор, пока датаграмма не достигнет хоста назначения. Однако, если датаграмма прибыла

именно на хост назначения, он не уничтожит ее и не сгенерирует ICMP сообщение об истечении времени, так как датаграмма достигла своего конечного назначения. Как можно определить, что датаграмма достигла конечного пункта назначения?
В UDP датаграммах, которые посылает Tracert, устанавливается несуществующий номер UDP порта (больше чем 30000), что делает невозможным обработку этой датаграммы каким-либо приложением. Поэтому когда прибывает подобная датаграмма, UDP модуль хоста назначения генерирует ICMP сообщение "порт недоступен" (port unreachable). Все что необходимо в этом случае, Tracert это определить тип принятого ICMP сообщения - либо об истечении времени, либо о недоступности порта - именно таким образом мы узнаем, доставлена ли датаграмма в пункт назначения.
На рисунке 3 показано как отправляется запрос от sun к NIC.

```
sun>tracertnic.ddn.miltraceroutetonic.ddn.mil(192.112.36.5), 30hopsmax, 40bytepackets1netb.tuc.noao.edu(140.252.1.183) 218ms 227ms 233ms2gateway.tuc.noao.edu(140.252.1.4) 233ms 229ms 204ms3butch.telcom.arizona.edu(140.252.104.2) 204ms 228ms 234ms4Gabby.Telcom.Arizona.EDU(128.196.128.1) 234ms 228ms 204ms5NSIgate.Telcom.Arizona.EDU(192.80.43.3) 233 ms 228ms 234ms6JPL1.NSN.NASA.GOV(128.161.88.2) 234ms 590ms 262ms7JPL3.NSN.NASA.GOV(192.100.15.3) 238ms 223ms 234ms8GSFC3.NSN.NASA.GOV(128.161.3.33) 293ms 318ms 324ms9GSFC8.NSN.NASA.GOV(192.100.13.8) 294ms 318ms 294ms10SURA2.NSN.NASA.GOV(128.161.166.2) 323ms 319ms 294ms11nsn-FIX-pe.sura.net(192.80.214.253) 294ms 318ms 294ms12GSI.NSN.NASA.GOV(128.161.252.2) 293ms 318 ms 324ms13NIC.DDN.MIL(192.112.36.5) 324ms 321ms 324ms
```

Рисунок 3. tracert от sun к nic.ddn.mil.
Когда датаграмма выходит из сети tuc.noao.edu, она попадает в сеть telcom.arizona.edu. Затем она попадает в сеть nsn.nasa.gov. Второе RTT для TTL равного 6 (590) почти в два раза больше, чем два другие RTT (234 и 262). Это показывает динамику IP маршрутизации. Подобное может произойти где-нибудь по пути от источника к маршрутизатору если какой-нибудь промежуточный маршрутизатор задержал датаграмму.
RTT для первой попытки с TTL равным 3 (204) меньше, чем RTT для первой попытки с TTL равной 2 (233). Так как каждое полученное RTT является полным временем прохода от посылающего хоста к маршрутизатору, это вполне объснимо.
Первая строка, без номера содержит имя и IP адрес пункта назначения и указывает на то, что величина TTL не может быть больше 30. Размер датаграммы установлен в 40 байт, из которых 20 байт отводится на IP заголовок, 8 байт на UDP заголовок и 12 байт на пользовательские данные. (В 12 байтах пользовательских данных содержится номер последовательности, который увеличивается на единицу при отправке каждой следующей датаграммы, копия исходящего TTL и время, когда датаграмма была отправлена.)
Следующие две строки вывода начинаются с TTL, после чего следует имя хоста или маршрутизатора и их IP адреса. Для каждого значения TTL отправляется 3 датаграммы. Для каждого возвращенного ICMP сообщения рассчитывается и печатается время возврата (round-trip). Если ответ не получен в течении пяти секунд на любую из трех датаграмм, печатается звездочка, после чего отправляется следующая датаграмма.

Изучение других утилит

Изучение утилиты ipconfig. Утилита предназначена для просмотра параметров TCP/IP на компьютере и управления IP-адресами, полученными интерфейсом через DHCP.
Запустите сеанс DOS. В ответ на приглашение DOS введите:

```
> ipconfig /?
```

и нажмите клавишу [Enter]. Изучите режимы работы утилиты ipconfig, запуская ее с различными параметрами.
Изучение утилиты arp. С помощью протокола ARP (Address Resolution Protocol) TCP/IP-компьютер преобразует IP-адреса в аппаратные, необходимые протоколам сетевого уровня для отправки кадров. IP использует ARP для определения аппаратного адреса, по которому нужно передавать дейтаграммы. Чтобы сократить объем сетевого трафика, генерируемого ARP, компьютер сохраняет разрешенные аппаратные адреса в КЭШе (на срок от 2 до 10 минут) на тот случай, если компьютеру понадобится отправить по этому же адресу дополнительные пакеты. В

комплект Windows входит утилита командной строки arp.exe, с помощью которой можно управлять содержимым кэша ARP, например, добавлять в него аппаратные адреса компьютеров, к которым Вы часто обращаетесь, чтобы сэкономить немного времени и сократить сетевой трафик.
Запустите сеанс DOS. В ответ на приглашение DOS введите:
> arp /?
и нажмите клавишу [Enter]. Изучите режимы работы утилиты arp, запуская ее с различными параметрами.
Изучение утилиты netstat. Netstat – утилита командной строки, отображающая информацию о текущих сетевых подключениях TCP/IP-компьютера и о трафике, генерируемом различными протоколами TCP/IP. На компьютерах с UNIX эта программа называется netstat, на компьютерах с Windows – netstat.exe.
Запустите сеанс DOS. В ответ на приглашение DOS введите:
> netstat /?
и нажмите клавишу [Enter]. Изучите режимы работы утилиты arp, запуская ее с различными параметрами.
Изучение утилиты nslookup. Утилита клмандной строки nslookup (на UNIX-системах) или nslookup.exe (на компьютерах с Windows NT/2000) позволяет генерировать запросы DNS и передавать их конкретному DNS-серверу.
С помощью программы nslookup Вы можете проверить работоспособность и производительность конкретного DNS-сервера.
Запустите сеанс DOS. В ответ на приглашение DOS введите:
> nslookup
и нажмите клавишу [Enter]. Утилита nslookup будет запущена в интерактивном режиме. В ответ на приглашение nslookup введите:
> help
и нажмите клавишу [Enter]. Изучите режимы работы утилиты nslookup, запуская ее с различными параметрами.
Изучение утилиты nbtstat. Nbtstat – утилита командной строки, отображающая статистику протокола и текущие сетевые подключения TCP/IP-компьютера.
Запустите сеанс DOS. В ответ на приглашение DOS введите:
> nbtstat /?
и нажмите клавишу [Enter]. Изучите режимы работы утилиты nbtstat, запуская ее с различными параметрами
Дополнительные утилиты на компьютерах с Windows XP/Wista. При установке служб finger, rexec и rsh на удаленных узлах, можно получить информацию о пользователях удаленной системы или подключиться к удаленному узлу.
Утилита hostname выводит имя текущего узла.
Утилита route. Обработка таблиц всех сетевых маршрутов. Изучите всевозможные параметры данной утилиты и выведите информацию по установленным маршрутам:
>route –p PRINT
Определите по статистике IP адрес шлюза вашей сети и основного шлюза.
Утилита pathping. Трассировка маршрута с дополнительными параметрами. Изучите всевозможные параметры данной утилиты.
>pathping –P –p 10 mrsu.ru
Определите узлы, которые проходит ваш пакет до указанного адреса. Определите что выводится в таблице статистики при запуске данной команды.
Утилита nslookup. Утилита выводит IP адреса по указанному DNS имени, используя службу имен.
Пример ввода:
>nslookup mrsu.ru
Перечисленные утилиты используются для выявления проблем на определенных уровнях при работе в сети или реконфигурации сетевых настроек.

Задание.
1. Выполните команды ping и tracert до конкретного компьютера в сети (например):
tracert s301-10 или

tracert 192.168.32.10 или
tracert mrsu.ru
Команды запускаются в командной строке (Пуск->Программы->Стандартные->Командная строка или Пуск->Выполнить->cmd). Команды надо запустить в нескольких режимах и расшифровать полученную статистику.
2. Определите параметры TCP/IP.
3. Определите в сети текущие сетевые подключения TCP/IP – компьютера и таблицу маршрутов.
4. Просмотрите статистические данные по всем протоколам TCP/IP.

Лабораторная работа №15.
Установка, настройка и использование программных сетевых анализаторов и сканеров безопасности. Анализ уязвимостей вычислительной системы.

Цели работы:
· Изучить основные принципы анализа сетевого трафика.
· Изучить сетевые протоколы, форматы кадров и пакетов, передаваемых по сети.
· Провести анализ сетевых угроз информационной безопасности.

Теоретические сведения.
Sniffer (от англ. to sniff – нюхать) – это сетевой анализатор трафика, программа или программно-аппаратное устройство, предназначенное для перехвата и последующего анализа, либо только анализа сетевого трафика, предназначенного для других узлов.
Перехват трафика может осуществляться:

- обычным «прослушиванием» сетевого интерфейса (метод эффективен при использовании в сегменте концентраторов (хабов) вместо коммутаторов (свичей), в противном случае метод малоэффективен, поскольку на сниффер попадают лишь отдельные фреймы);
- подключением сниффера в разрыв канала;
- ответвлением (программным или аппаратным) трафика и направлением его копии на сниффер;
- через анализ побочных электромагнитных излучений и восстановление таким образом прослушиваемого трафика;
- через атаку на канальном (2-й) или сетевом (3-й) уровне, приводящую к перенаправлению трафика жертвы или всего трафика сегмента на сниффер с последующим возвращением трафика в надлежащий адрес.

В начале 1990-х широко применялся хакерами для захвата пользовательских логинов и паролей. Широкое распространение хабов позволяло захватывать трафик без больших усилий в больших сегментах сети. Снифферы применяются как в благих, так и в деструктивных целях. Анализ прошедшего через сниффер трафика, позволяет:

- Отслеживать сетевую активность приложений.
- Отлаживать протоколы сетевых приложений.
- Локализовать неисправность или ошибку конфигурации.
- Обнаружить паразитный, вирусный и закольцованный трафик, наличие которого увеличивает нагрузку сетевого оборудования и каналов связи.
- Выявить в сети вредоносное и несанкционированное ПО, например, сетевые сканеры, флудеры, троянские программы, клиенты пиринговых сетей и другие.
- Перехватить любой незашифрованный (а порой и зашифрованный) пользовательский трафик с целью узнавания паролей и другой информации.

Постепенно из инструментов, предназначенных только для диагностики, сниферы постепенно превратились в средства для исследований и обучения. Например, они постоянно используются для изучения динамики и взаимодействий в сетях. В частности, они позволяют легко и наглядно изучать тонкости сетевых протоколов. Наблюдая за данными, которые посылает протокол, вы можете глубже понять его функционирование на практике, а заодно увидеть, когда некоторая конкретная реализация работает не в соответствии со спецификацией. На сегодняшний момент существует достаточно большое количество хороших реализаций снифферов
Сниффер Wireshark
Программа Wireshark является одной из самых удобных реализаций снифферов. Портирована на большое количество платформ. Распространяется абсолютно бесплатно. В дальнейшем на лабораторных работах мы будем постоянно использовать данный сниффер для изучения и анализа сетевых протоколов. Но для начала рассмотрим базовый принцип работы сниффера как раз на примере Wireshark.^
Базовый принцип работы снифферов
Давайте рассмотрим с вами рис. 1.

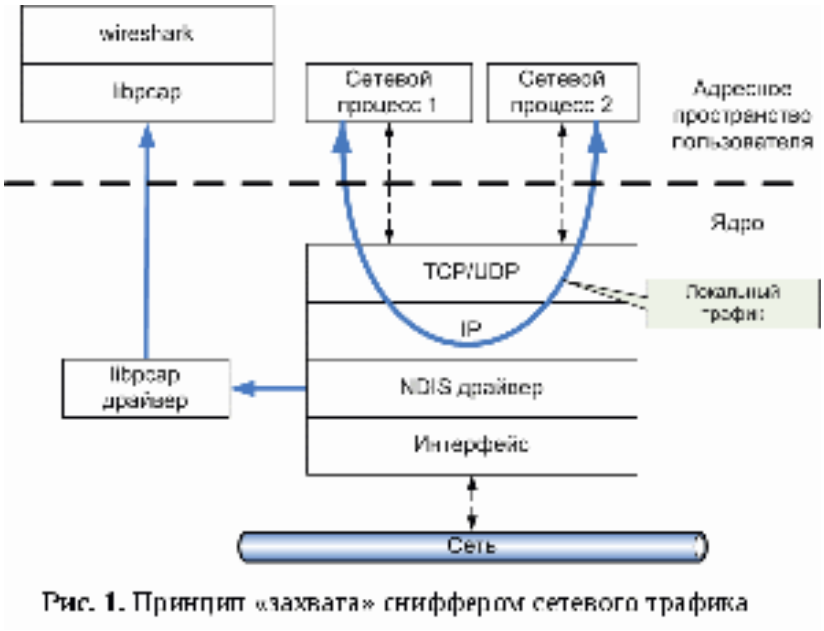


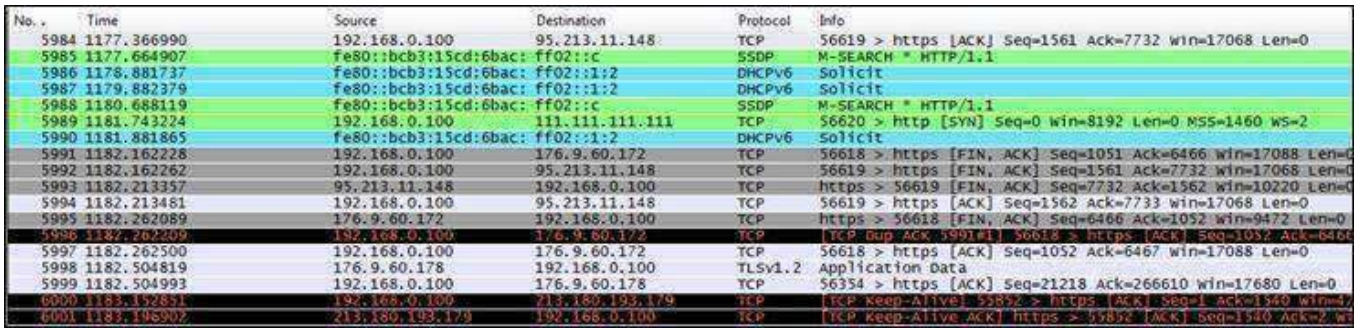
Рис. 1. Принцип «захвата» сниффером сетевого трафика

На нем изображена схематично структура сетевой подсистемы ОС. Вся базовая инфраструктура реализована в виде драйверов и работает в режиме ядра. Пользовательские процессы и реализации прикладных протоколов, в частности интерфейс сниффера работают в пользовательском режиме. На рисунке отображены 2 пользовательских процесса («сетевой процесс 1» и «сетевой процесс 2»). Основными компонентами сниффера являются: драйвер для захвата пакетов (libpcap драйвер), интерфейсная библиотека (libpcap) и интерфейс пользователя (Wireshark). Библиотека libpcap (реализация под ОС Windows носит название WinPcap - http://www.winpcap.org) – универсальная сетевая библиотека, самостоятельно реализующая большое количество сетевых протоколов и работающая непосредственно с NDIS (Network Driver Interface Specification) драйверами сетевых устройств. На базе данной библиотеки реализовано большое количество сетевых программ, в частности сниффер Wireshark. Сниффер использует библиотеку в режиме «захватывата» пакетов, т.е. может получать копию ВСЕХ данных проходящих через драйвер сетевого интерфейса. Изменения в сами данные не вносятся! Основной нюанс использования сниффера заключается в том, что он не позволяет производить анализ локального трафика, т.к. он не проходит через драйвер сетевого устройства (см. рис 1.). Т.е., если вы захотите проанализировать сниффером трафик между 2-ми сетевыми процессами на локальной машине (например, ftp-сервер и ftp-клиент), то у вас ждет разочарование. Однако, например при использовании виртуальных машин, сниффер будет работать без проблем, т.к. виртуальные машины эмулируют реальную среду и сетевые адаптеры, поэтому трафик идет через драйвера как и в нормальной ситуации при взаимодействии с другими физическими сетевыми машинами. Также к недостаткам большинства снифферов стоит отнести и тот факт, что, позволяя анализировать трафик, проходящий через сетевой интерфейс, они не могут указать, какое именно приложение генерирует или получает его. Это объясняется те, что информация об этом хранится на сетевом (например, IP) уровне сетевого стека, а большинство снифферов использует собственную реализацию стека протоколов (например, библиотеку WinPcap), которая (как уже было показано) работает непосредственно с драйверами устройств. Стоит также отметить, что снифферы вносят дополнительную нагрузку на процессор, т.к. могут обрабатывать достаточно объемный сетевой трафик, в особенности для высокоскоростных соединений (Fast Ethernet, Gigabit Ethernet и др.).

Ход работы:

1. Запустим программу *Wireshark* и получим список всех *Ethernet*-адаптеров, а также их *mac*-адреса, включим режим прослушивания.

Все захваченные пакеты отображаются в окне пакетов. Каждая строка в



No.	Time	Source	Destination	Protocol	Info
5984	1177.366990	192.168.0.100	95.213.11.148	TCP	56619 > https [ACK] Seq=1561 Ack=7732 Win=17068 Len=0
5985	1177.664907	fe80::bcb3:15cd:6bac:	ff02::c	SSDP	M-SEARCH * HTTP/1.1
5986	1178.881737	fe80::bcb3:15cd:6bac:	ff02::1:2	DHCPv6	Solicit
5987	1179.882379	fe80::bcb3:15cd:6bac:	ff02::1:2	DHCPv6	Solicit
5988	1180.688119	fe80::bcb3:15cd:6bac:	ff02::c	SSDP	M-SEARCH * HTTP/1.1
5989	1181.743224	192.168.0.100	111.111.111.111	TCP	56620 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
5990	1181.881865	fe80::bcb3:15cd:6bac:	ff02::1:2	DHCPv6	Solicit
5991	1182.162228	192.168.0.100	176.9.60.172	TCP	56618 > https [FIN, ACK] Seq=1051 Ack=6466 Win=17088 Len=0
5992	1182.162262	192.168.0.100	95.213.11.148	TCP	56619 > https [FIN, ACK] Seq=1561 Ack=7732 Win=17068 Len=0
5993	1182.213357	95.213.11.148	192.168.0.100	TCP	https > 56619 [FIN, ACK] Seq=7732 Ack=1562 Win=10220 Len=0
5994	1182.213481	192.168.0.100	95.213.11.148	TCP	56619 > https [ACK] Seq=1562 Ack=7733 Win=17068 Len=0
5995	1182.262089	176.9.60.172	192.168.0.100	TCP	https > 56618 [FIN, ACK] Seq=6466 Ack=1052 Win=9472 Len=0
5996	1182.262209	192.168.0.100	176.9.60.172	TCP	[TCP Dup ACK 5991#1] 56618 > https [ACK] Seq=1052 Ack=6466
5997	1182.262500	192.168.0.100	176.9.60.172	TCP	56618 > https [ACK] Seq=1052 Ack=6467 Win=17088 Len=0
5998	1182.504819	176.9.60.178	192.168.0.100	TLSv1.2	Application Data
5999	1182.504993	192.168.0.100	176.9.60.178	TCP	56354 > https [ACK] Seq=21218 Ack=266610 Win=17680 Len=0
6000	1183.152851	192.168.0.100	213.180.193.179	TCP	[TCP Keep-Alive] 55852 > https [ACK] Seq=1 Ack=1340 Win=4
6001	1183.196902	213.180.193.179	192.168.0.100	TCP	[TCP Keep-Alive ACK] https > 55852 [ACK] Seq=1340 Ack=2 Wi

данном окне связана с одним пакетом. При выборе строки дополнительная информация будет отображена в окне детализации.

Рисунок 1 – Окно пакетов, в котором отображаются все захваченные пакеты

2. При помощи утилиты *ipconfig* определим настройку протокола *IP* (адрес подсети, *IP*-адрес компьютера, а также диапазон адресов, используемых в подсети).

```
C:\Windows\system32\cmd.exe
Настройка протокола IP для Windows

   Имя компьютера . . . . . . . . . . : MaX-PC
   Основной DNS суффикс . . . . . . . :
   Тип узла. . . . . . . . . . . . . . : Гибридный
   IP-маршрутизация включена . . . . . : Нет
   WINS-прокси включен . . . . . . . . : Нет

Адаптер беспроводной локальной сети Беспроводное сетевое соединение 2:

   Состояние среды. . . . . . . . : Среда передачи недоступна.
   DNS-суффикс подключения . . . . . :
   Описание. . . . . . . . . . . . . : Microsoft Virtual WiFi Miniport Adapter
   Физический адрес. . . . . . . . . : 3E-4B-D6-9D-9A-3E
   DHCP включен. . . . . . . . . . . : Да
   Автонастройка включена. . . . . . : Да

Ethernet adapter Сетевое подключение Bluetooth:

   Состояние среды. . . . . . . . : Среда передачи недоступна.
   DNS-суффикс подключения . . . . . :
   Описание. . . . . . . . . . . . . : Устройства Bluetooth (личной сети)
   Физический адрес. . . . . . . . . : 1C-4B-D6-85-D2-FB
   DHCP включен. . . . . . . . . . . : Да
   Автонастройка включена. . . . . . : Да

Адаптер беспроводной локальной сети Беспроводное сетевое соединение:

   DNS-суффикс подключения . . . . . :
   Описание. . . . . . . . . . . . . : Atheros AR9285 Wireless Network Adapter
   Физический адрес. . . . . . . . . : 1C-4B-D6-9D-9A-3E
   DHCP включен. . . . . . . . . . . : Да
   Автонастройка включена. . . . . . : Да
   Локальный IPv6-адрес канала . . . : fe80::bcb3:15cd:6bac:3643%11(Основной)
   IPv4-адрес. . . . . . . . . . . . : 192.168.0.100(Основной)
   Маска подсети . . . . . . . . . . : 255.255.255.0
   Аренда получена. . . . . . . . . . : 8 ноября 2016 г. 5:53:14
   Срок аренды истекает. . . . . . . . . . : 17 ноября 2016 г. 5:00:28
   Основной шлюз. . . . . . . . . : 192.168.0.1
   DHCP-сервер. . . . . . . . . . . : 192.168.0.1
   IAID DHCPv6 . . . . . . . . . . . : 219958230
   DUID клиента DHCPv6 . . . . . . . : 00-01-00-01-17-F9-E8-96-48-5B-39-88-DA-94

   DNS-серверы. . . . . . . . . . . : fec0:0:0:1:1:1:1:1
                                       192.168.0.1
   NetBios через TCP/IP. . . . . . . : Включен

Ethernet adapter Подключение по локальной сети:

   Состояние среды. . . . . . . . : Среда передачи недоступна.
   DNS-суффикс подключения . . . . . :
   Описание. . . . . . . . . . . . . : Realtek PCIe GBE Family Controller
   Физический адрес. . . . . . . . . : 48-5B-39-88-DA-94
   DHCP включен. . . . . . . . . . . : Да
   Автонастройка включена. . . . . . : Да
```

В данном пункте в командной строке мы вводим *ipconfig /all*, что позволяет вывести полную конфигурацию *TCP/IP* для всех адаптеров. Без параметра */all* команда*ipconfig* выводит только *IP*-адреса, маску подсети и основной шлюз для каждого адаптера.

Рисунок 2 – Полная конфигурация *TCP/IP* для всех адаптеров

```
C:\Users\MaX>arp -a

Интерфейс: 192.168.0.100 --- 0xb
  адрес в Интернете      Физический адрес      Тип
  192.168.0.1           1c-bd-b9-34-7d-cb     динамический
  192.168.0.255         ff-ff-ff-ff-ff-ff     статический
  224.0.0.22            01-00-5e-00-00-16     статический
  224.0.0.251           01-00-5e-00-00-fb     статический
  224.0.0.252           01-00-5e-00-00-fc     статический
  239.255.255.250       01-00-5e-7f-ff-fa     статический
  255.255.255.255       ff-ff-ff-ff-ff-ff     статический
```

3. Осуществим вывод таблиц протокола*ARP* для всех интерфейсов при помощи команды *arp – a.* Вывод таблиц производился на домашнем компьютере, дальнейшие скриншоты и описание следуют из работы в аудитории.

Рисунок 3 – Таблица протокола *ARP* для всех интерфейсов

4. Пользуясь программой сканером, отследим пакеты протокола *ARP*, сформированные при соединении с компьютером-корреспондентом.

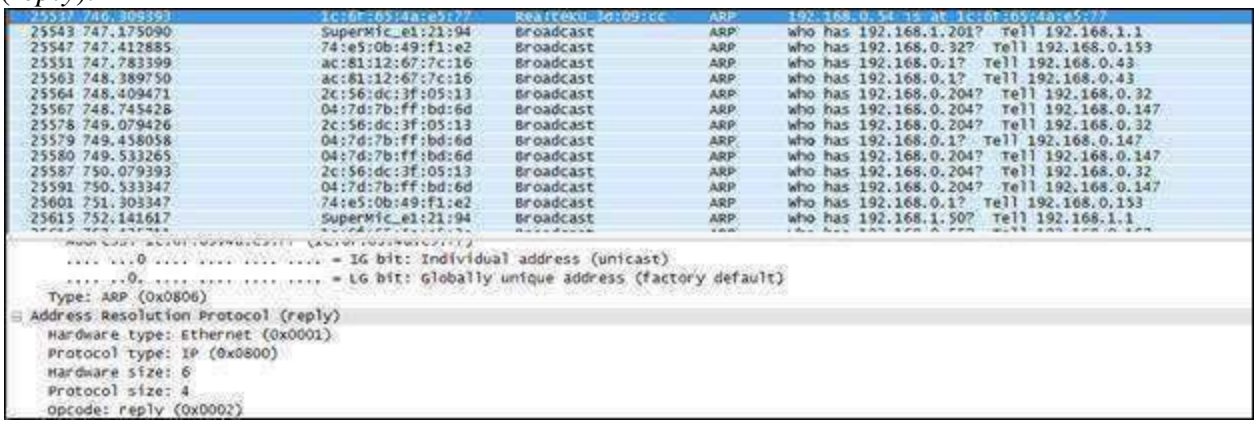
Adress Resolution Protocol – протокол разрешения адреса.

Передавать фреймы в сети *Ethernet* можно, используя аппаратные (далее *MAC*) адреса. Это не сетевой (*IP*) адрес вида 192.168.0.1, а нечто более сложное, вида 00:00:00:c1:d9:26.

Принцип работы следующий. Компьютер, которому требуется передать данные в сети, имеет *IP* адрес (свой собственный), а также *IP* адрес получателя. Однако передавать данные можно, только имея *MAC*-адреса. Поэтому компьютер, инициирующий соединение, посылает широковещательный *ARP* запрос вида:

Who has <IP-addr>? Tell <Self IP-addr>

где спрашивает, “ У кого IP такой-то? Сообщите мне”, и этот пакет посылается на ff:ff:ff:ff:ff:ff (широковещательно), то есть всем узлам в сети. Каждый компьютер в сети получает этот пакет и смотрит, принадлежит ли указанный *IP* какому-либо из его интерфейсов. Если принадлежит, то на *MAC*-адрес отправителя (он известен из заголовка пакета) отправляется пакет с ответом (*reply*).



Компьютер, получивший ответ, для передачи на целевой *IP* будет посылать данные на *MAC*-адрес отправителя пакета ответа.

Рисунок 4 – Отслеживание пакетов протокола *ARP*, сформированных при соединении с компьютером-корреспондентом

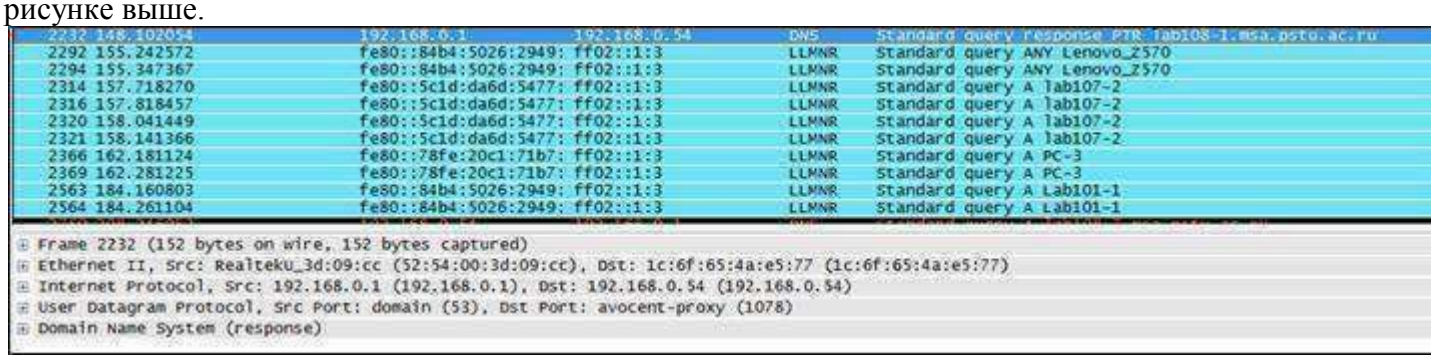
Очень распространена атака вида ***MITM*** *(Man in the Middle)*, когда компьютер злоумышленника отвечает вместо целевого компьютера (раньше его) и обмен между двумя

узлами происходит через посредника, прослушивающего весь трафик. На рисунке ниже показана атака *MITM.*

Рисунок 5 – Атака «человек посередине» *(MITM)*

Перед тем, как узел “А” посылает кадр данных для “Б”, он должен получить физический адрес узла “Б” с известным *IP*-адресом. Посылается широковещательный запрос и ожидается ответ от “Б” обратно к “А”, что “ У меня такой адрес”. Задача злоумышленника ответить узлу “А” раньше истинного узла “Б”, что он, якобы, и есть узел “Б”.

Тогда узел “А” будет посылать пакет злоумышленнику вместо истинного “Б”, полагая, что посылает верно. Прочитав данные, злоумышленник пошлёт пакет истинному “Б” дальше, и наоборот. Иначе говоря, злоумышленник «встраивается» между ними, как это изображено на рисунке выше.



5. *DNS* – трафик. Протокол*DNS* служит для преобразования понятного людям символьного имени, такого как, к примеру, *litl-admin.ru* в *IP*-адрес, на который выполняется запрос.

Рисунок 6 – Образец трафика *DNS*

Протокол *DNS* (в выделенной зоне он самый нижний – *domain name system*) является надстройкой в протокол *UDP*, лежащий на 4-ом уровне. То есть без установки постоянного соединения, простой отправкой пакетов (дэйтаграмм). Как и*TCP*, протокол *UDP* имеет порты. Для *DNS* это порт с номером 53, что легко прослеживается в дереве протоколов.

Спускаясь ещё ниже, мы видим, что *UDP* протокол опирается на *IP* – протокол передачи между сетями. В настоящее время*IP*-протокол - основной протокол передачи данных в сети *Internet*. На этом уровне имеется информация об *IP*-адресах источника, а также назначения и сведения об инкапсулированном пакете *UDP*.

В качестве адреса источника будет использоваться наша машина, запросившая информацию по *DNS*, а в качестве IP-адреса назначения – сам *DNS* сервер, который (подразумевается) должен иметь базу данных таких соответствий.

Как видим, если развернуть дерево информации в анализе пакета, то станет понятно, что поступил запрос (*Queries*) получения стандартной записи хоста (Type: PTR) по имени (Name: 54.0.168.192.in-addr.arpa).

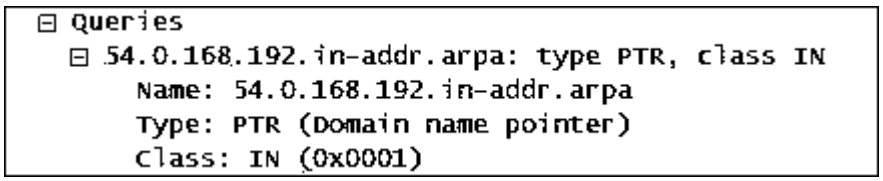


Рисунок 7 – Стандартный запрос

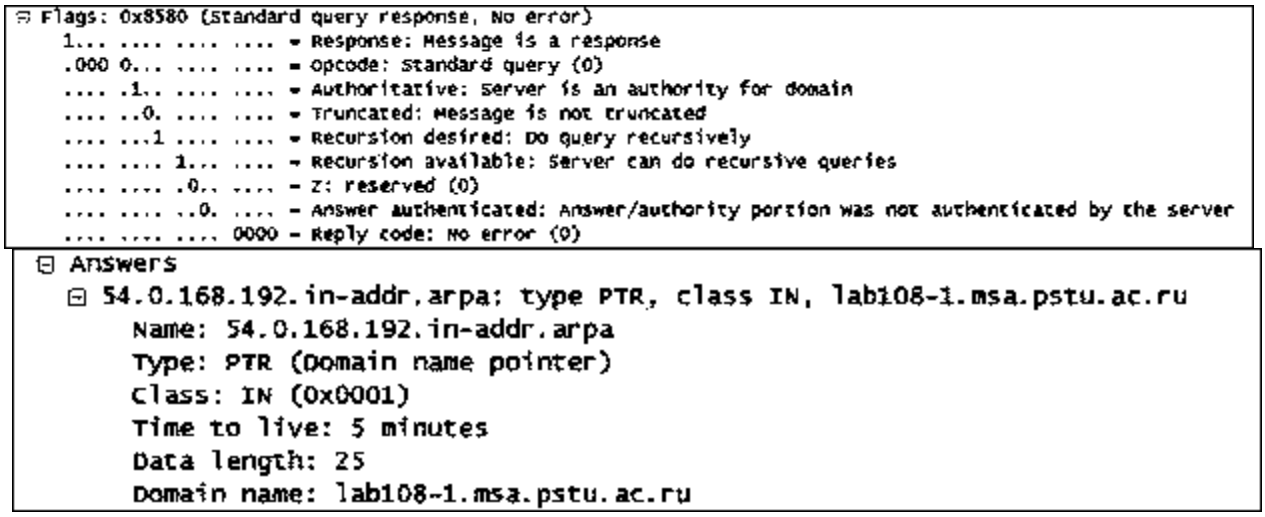


Рисунок 8 – Ответ на запрос

Рисунок 9 – Флаги ответа

6. Пользуясь программой сканером на примере одного из активных соединений, отследим получаемый трафик по протоколу *TCP*, а на примере одного из пакетов - вложенность протоколов. Выполним анализ информации, содержащейся в заголовках сегмента *TCP*.

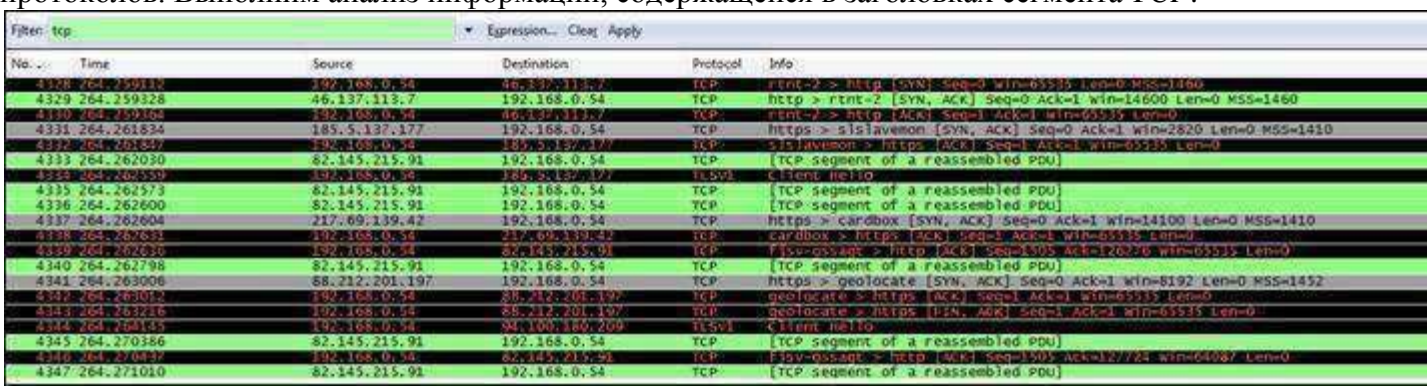
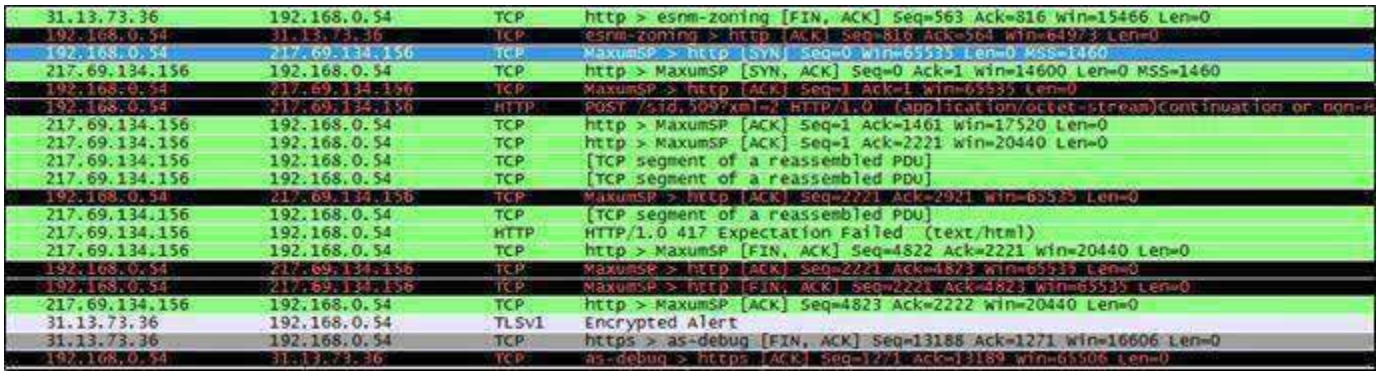


Рисунок 10 – Трафик по протоколу *TCP*

Предположим, узел А желает установить соединение с узлом В. Первый отправляемый из А в В *TCP*-сегмент не содержит полезных данных, а служит для установления соединения. В его заголовке (в поле *Flags*) установлен бит *SYN*, означающий запрос связи. В ответ на получение такого сегмента узел В откликается посылкой *TCP*-сегмента, в заголовке которого установлен бит *ACK*, подтверждающий установление соединения для получения данных от узла А. Так как протокол *TCP* обеспечивает полнодуплексную передачу данных, то узел В в этом же сегменте устанавливает бит *SYN*, означающий запрос связи для передачи данных от В к А. Полезных данных этот сегмент также не содержит. Третий *TCP*-сегмент в сеансе посылается из А в В в ответ на сегмент, полученный из В. Так как соединение А ->В можно считать установленным (получено подтверждение от В), то узел А включает в свой сегмент полезные данные.

Сеанс обмена данными заканчивается процедурой разрыва соединения, которая аналогична процедуре установки, с той разницей, что вместо *SYN* для разрыва используется служебный бит *FIN* (“данных для отправки больше не имею”), который устанавливается в заголовке последнего сегмента с данными, отправляемого узлом.

После применения фильтра *TCP* первые три кадра на панели списка пакетов (верхний раздел) отображают протокол транспортного уровня *TCP*, создающий надёжный сеанс связи. Последовательность *[SYN], [SYN, ACK]* и *[ACK]*иллюстрирует трёхстороннее рукопожатие.



Протокол *TCP* регулярно используется во время сеанса связи для контроля доставки датаграмм, проверки их поступления и управления размером окна. Для каждого обмена данными между *FTP*- клиентом и *FTP*-сервером запускается новый сеанс *TCP*. По завершении передачи данных сеанс *TCP* закрывается. По завершении сеанса *FTP* протокол *TCP* выполняет плановое отключение и прекращение работы. Программа *Wireshark* отображает подробные данные *TCP* на панели сведений о пакетах (средний раздел).

Рисунок 11 – Сеанс обмена данными

7. Пользуясь программой сканером, проследим процесс установления соединения по протоколу *FTP* с любым *FTP* сервером. Проанализируем защищенность процедуры аутентификации с *FTP* сервером.

В поле фильтра следует просто написать "*ftp*". Незамедлительно после этого будет выделен трафик, относящийся к сессии *FTP*, и в качестве потрясающей демонстрации проблем с безопасностью увидим имя пользователя и пароль, который ввели ранее. Результат представлен на рисунке ниже.

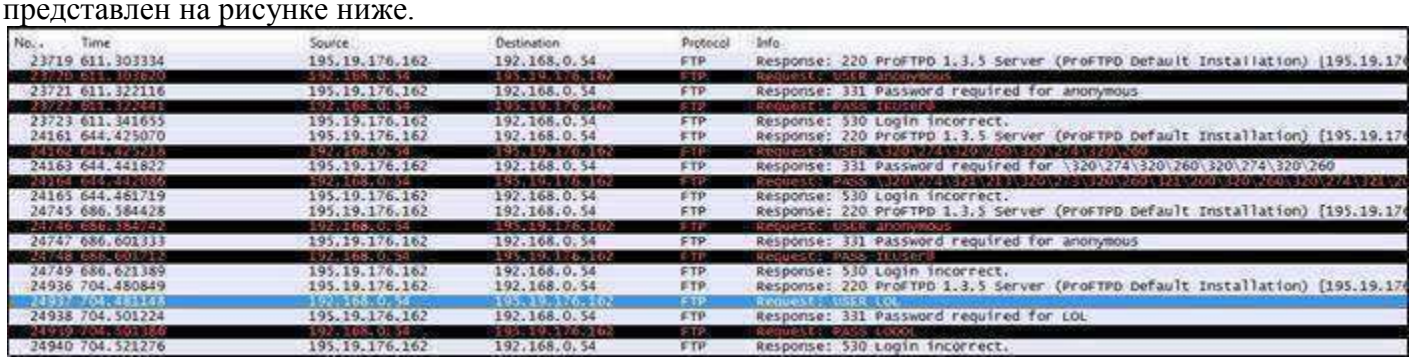


Рисунок 12 – Трафик по протоколу *FTP*

Вывод:в ходе выполненной лабораторной работы были изучены основные принципы анализа сетевого трафика, сетевые протоколы, форматы кадров и пакетов, передаваемых по сети, а также проведён анализ сетевых угроз информационной безопасности.

Лабораторная работа №16.
Настройка коммутатора, поддерживающего VLAN.

Теоретические сведения

VLAN (аббр. от англ. Virtual Local Area Network) — логическая ("виртуальная") локальная компьютерная сеть, представляет собой группу хостов с общим набором требований, которые взаимодействуют так, как если бы они были подключены к широковещательному домену, независимо от их физического местонахождения. VLAN имеет те же свойства, что и физическая локальная сеть, но позволяет конечным станциям группироваться вместе, даже если они не находятся в одной физической сети.

VLAN'ы могут быть настроены на коммутаторах, маршрутизаторах, других сетевых устройствах.

Преимущества:

1- Облегчается перемещение, добавление устройств и изменение их соединений друг с другом.
2- Достигается большая степень административного контроля вследствие наличия устройства, осуществляющего между сетями VLAN маршрутизацию на 3-м уровне.
3- Уменьшается потребление полосы пропускания по сравнению с ситуацией одного широковещательного домена.
4- Сокращается непроизводственное использование CPU за счет сокращения пересылки широковещательных сообщений.
5 - Предотвращение широковещательных штормов и предотвращение петель.

В данной работе рассматривается настройка VLAN на коммутаторе фирмы Cisco на его портах доступа.

Ход работы.

Создайте сеть, логическая топология которой представлена на рис.1. Компьютеры соединены коммутатором Cisco 2960-24TT. В таблице 1 приведены адреса компьютеров.

Задача данной работы – сделать две независимые группы компьютеров: ПК0, ПК1 и ПК2 должны быть доступны только друг для друга, вторая независимая группа - компьютеры ПК3 и ПК4. Для этого создадим два отдельных VLAN (рис.1).

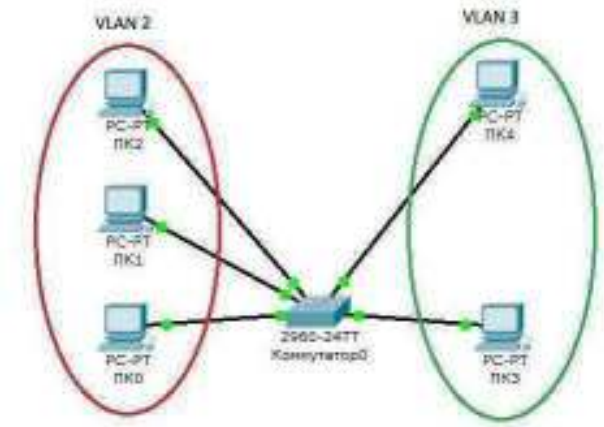


Рис. 1. Схема сети с одним коммутатором.

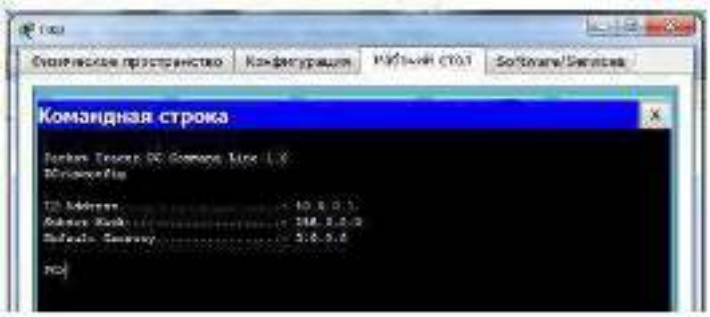
Таблица 1.

Компьютер	IP адрес	Порт коммутатора
ПК0	10.0.0.1/8	1
ПК1	10.0.0.2/8	2
ПК2	10.0.0.3/8	3
ПК3	10.0.0.4/8	4
ПК4	10.0.0.5/8	5

Далее будем считать, что ПК0, ПК1 и ПК2 находятся в VLAN 2, а ПК3 и ПК4 находятся в VLAN 3. Для проверки конфигурации хоста ПК0 выполним команду ipconfig. Результат выполнения

команды на рисунке 2. При желании можно выполнить аналогичную проверку на остальных хостах.

Рис. 2. Проверка конфигурации хоста



Используя команду PING проверим связь между всеми компьютерами. Сейчас они в одной сети и все доступны друг для друга

Теперь займемся настройкой VLAN 2 и VLAN3, чтобы структурировать сети на коммутаторе и навести в них порядок.

Далее перейдем к настройке коммутатора. Откроем его консоль. Для того чтобы это выполнить в Packet Tracer дважды щелкните левой кнопкой мыши по коммутатору в рабочей области.

В открывшемся окне перейдите на вкладку CLI. Вы увидите окно консоли. Нажмите Enter, чтобы приступить к вводу команд. Информация, которая в данный момент отражена на консоли, свидетельствует о том что интерфейсы FasteEthernet0/1 – FasteEthernet0/5 находятся в рабочем состоянии.

Перейдем в привилегированный режим выполнив команду enable:

Switch>en

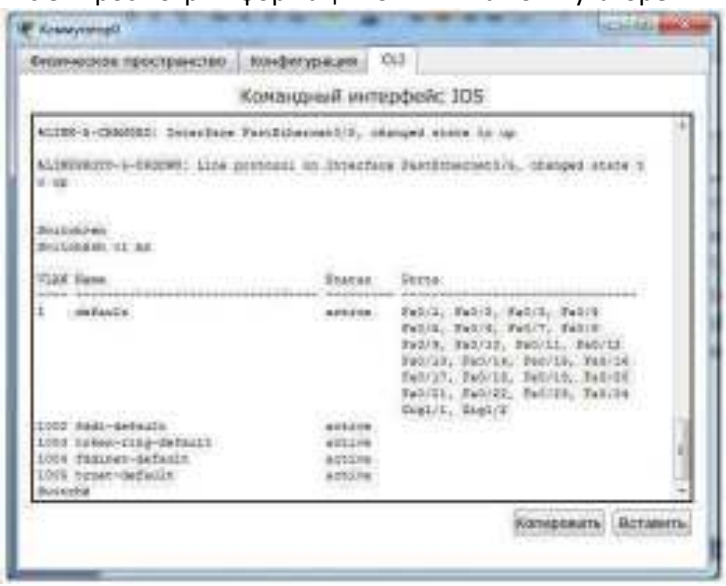
Switch#

Просмотрим информацию о существующих на коммутаторе VLAN-ах (рис.3).

Для этого выполним следующую команду:

Switch#sh vl br

Рис.3. Просмотр информации о VLAN на коммутаторе.



В результате выполнения команды на экране появится: номера VLAN – первый столбец, название VLAN - второй столбец, состояние VLAN (работает он в данный момент или нет) – третий столбец, порты принадлежащие к данному VLAN – четвертый столбец. Как мы видим по умолчанию на коммутаторе существует пятьVLAN-ов. Все порты коммутатора по умолчанию принадлежат VLAN 1. Остальные четыре VLAN являются служебными и используются не очень часто.

Для реализации сети, которую мы запланировали сделать, создадим на коммутаторе еще два VLAN. Для этого в привилегированном режиме выполните следующую команду:

```
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#
```

для перехода в режим конфигурации. Вводим команду VLAN 2. Данной командой вы создадите на коммутаторе VLAN с номером 2. Указатель ввода Switch(config)# изменится на Switch(config-vlan)# это свидетельствует о том, что вы конфигурируете уже не весь коммутатор в целом, а только отдельный VLAN, в данном случае VLAN номер 2. Если вы используете команду «vlan x», где x номер VLAN, когда VLAN x еще не создан на коммутаторе, то он будет автоматически создан и вы перейдете к его конфигурированию. Когда вы находитесь в режиме конфигурирования VLAN, возможно изменение параметров выбранной виртуальной сети, например можно изменить ее имя с помощью команды name.
Для достижения поставленной в данном посте задачи, сконфигурируем VLAN 2 следующим образом:

```
Switch(config)#vlan 2
Switch(config-vlan)#name subnet_10
Switch(config)#interface range fastEthernet 0/1-3
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 2
```

Разберем данную конфигурацию. Как уже говорилось ранее командой VLAN 2, мы создаем на коммутаторе новый VLAN с номером 2. Команда name subnet_10 присваивает имя subnet_10 виртуальной сети номер 2. Выполняя команду interface range fastEthernet 0/1-3 мы переходим к конфигурированию интерфейсов fastEthernet0/1, fastEthernet0/2 и fastEthernet0/3 коммутатора. Ключевое слово range в данной команде, указывает на то, что мы будем конфигурировать не один единственный порт, а целый диапазон портов, в принципе ее можно не использовать, но тогда последние три строки придется заменить на:

```
Switch(config)#interface fastEthernet 0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 2
Switch(config)#interface fastEthernet 0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 2
Switch(config)#interface fastEthernet 0/3
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 2
```

Команда switchport mode access конфигурирует выбранный порт коммутатора, как порт доступа (аксесс порт). Команда switchport access vlan 2 указывает, что данный порт является портом доступа для VLAN номер 2.
Выйдите из режима конфигурирования, дважды набрав команду exit и просмотрите результат конфигурирования (рис.4), выполнив уже знакомую нам команду sh vl br еще раз:

Рис.4. Распределение портов на VLAN.

```
Switch(config-if-range)#switchport access vlan 2
Switch(config-if-range)#exit
Switch(config)#exit
Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch#sh vl br

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/4, Fa0/5, Fa0/6, Fa0/7
                                                 Fa0/8, Fa0/9, Fa0/10, Fa0/11
                                                 Fa0/12, Fa0/13, Fa0/14, Fa0/15
                                                 Fa0/16, Fa0/17, Fa0/18, Fa0/19
                                                 Fa0/20, Fa0/21, Fa0/22, Fa0/23
                                                 Fa0/24, Gig1/1, Gig1/2
2    subnet_10                         active    Fa0/1, Fa0/2, Fa0/3
1002 fddi-default                      active
1003 token-ring-default                active
1004 fddinet-default                   active
1005 trnet-default                     active
Switch#
```

На коммутаторе появился еще один VLAN с номером 2 и именем subnet_10, портами доступа которого являются fastEthernet0/1, fastEthernet0/2 и fastEthernet0/3.

Далее аналогичным образом создадим VLAN 3 с именем subnet_192 и сделаем его портами доступа интерфейсы fastEthernet0/4 и fastEthernet0/5. Результат должен получиться следующим (рис.5):

Рис.5. Распределение портов на VLAN.

```
Switch#sh vl br

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                                Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                                Fa0/22, Fa0/23, Fa0/24, Gig1/1
                                                Gig1/2
2    subnet_10                        active    Fa0/1, Fa0/2, Fa0/3
3    subnet_192                       active    Fa0/4, Fa0/5
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
Switch#
```

В принципе уже все готово и наша сеть настроена. Осталось лишь ее немного протестировать. Перейдите в консоль компьютера ПК0. Пропингуйте с него остальные компьютеры сети. Компьютеры ПК1 и ПК2 доступны, а компьютеры ПК3 и ПК4 не доступны. Все пять компьютеров теоретически должны находится в одной подсети 10.0.0.0/8 и видеть друг друга, на практике они находятся в разных виртуальных локальных сетях и поэтому не могут взаимодействовать между собой.

Лабораторная работа №17.
Настройка маршрутизации. Проверка сетевых соединений. Включение службы маршрутизации. Добавление маршрутов. Таблицы маршрутизации

Теоретические сведения

Протоколы маршрутизации - это правила, по которым осуществляется обмен информации о путях передачи пакетов между маршрутизаторами. Протоколы характеризуются временем сходимости, потерями и масштабируемостью. В настоящее время используется несколько протоколов маршрутизации.

Одна из главных задач маршрутизатора состоит в определении наилучшего пути к заданному адресату. Маршрутизатор определяет пути (маршруты) к адресатам или из статической конфигурации, введённой администратором, или динамически на основании маршрутной информации, полученной от других маршрутизаторов. Маршрутизаторы обмениваются маршрутной информацией с помощью протоколов маршрутизации.

Маршрутизатор хранит таблицы маршрутов в оперативной памяти. Таблица маршрутов это список наилучших известных доступных маршрутов. Маршрутизатор использует эту таблицу для принятия решения куда направлять пакет.

В случае статической маршрутизации администратор вручную определяет маршруты к сетям назначения.

В случае динамической маршрутизации – маршрутизаторы следуют правилам, определяемым протоколами маршрутизации для обмена информацией о маршрутах и выбора лучшего пути.

Статические маршруты не меняются самим маршрутизатором. Динамические маршруты изменяются самим маршрутизатором автоматически при получении информации о смене маршрутов от соседних маршрутизаторов. Статическая маршрутизация потребляет мало вычислительных ресурсов и полезна в сетях, которые не имеют нескольких путей к адресату назначения. Если от маршрутизатора к маршрутизатору есть только один путь, то часто используют статическую маршрутизацию.

Ход работы.

Проведем настройку статической маршрутизации с помощью графических мастеров интерфейса Cisco Packet Tracer. Создайте схему сети, представленную на рис.1.

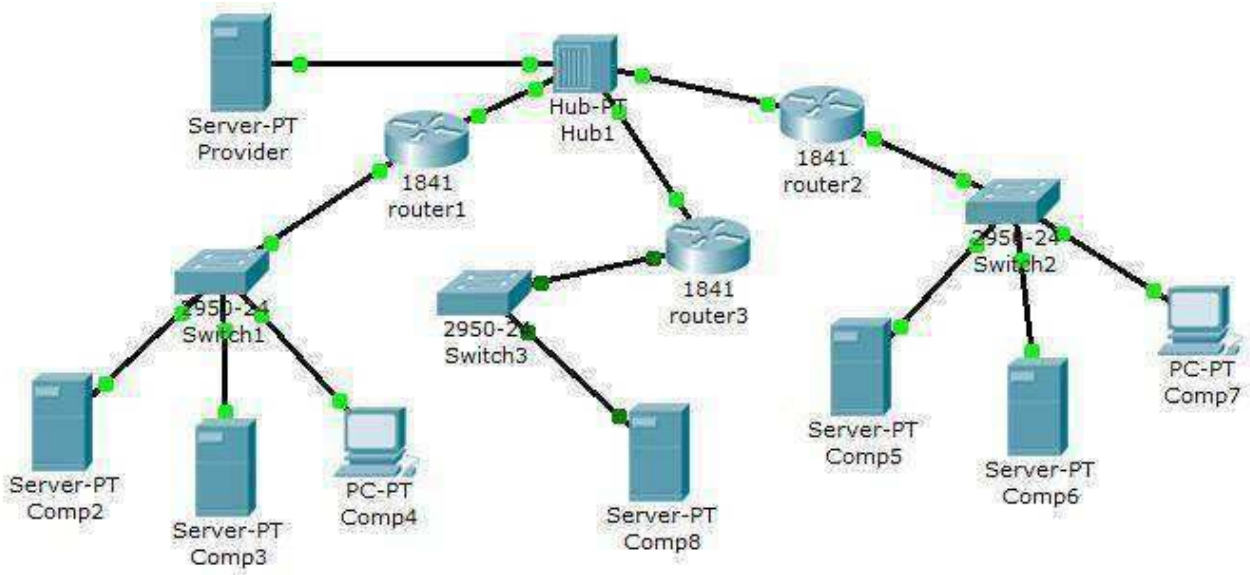


Рис.1. Схема сети.

На данной схеме представлена корпоративная сеть, состоящая из следующих компонентов:
Сеть 1 – на Switch1 замыкается сеть первой организации (таблица 1):

Таблица 1. Сеть первой организации.

компьютер	**IP адрес**	**Функции**

Comp2	192.168.1.2/24	DNS и HTTP сервер
Comp3	192.168.1.3/24	DHCP сервер
Comp4	Получен с DHCP сервера	Клиент сети

В данной сети на Comp2 установлен DNS и Web сервер с сайтом организации.
На Comp3 установлен DHCP сервер. Компьютер Comp4 получает с DHCP сервера IP адрес, адрес DNS сервера провайдера (сервер Provider) и шлюз. Шлюз в сети – 192.168.1.1/24.

Сеть 2 – на Switch2 замыкается сеть второй организации (таблица 2):

Таблица 5.2. Сеть второй организации.

компьютер	IP адрес	Функции
Comp5	10.0.0.5/8	DNS и HTTP сервер
Comp6	10.0.0.6/8	DHCP сервер
Comp7	Получен с DHCP сервера	Клиент сети

В данной сети на Comp5 установлен DNS и Web сервер с сайтом организации.

На Comp4 установлен DHCP сервер. Компьютер Comp7 получает с DHCP сервера IP адрес, адрес DNS сервера провайдера (сервер Provider) и шлюз. Шлюз в сети – 10.0.0.1/8.

Сеть 3 – на Hub1 замыкается городская сеть 200.200.200.0/24. В сети установлен DNS сервер провайдера (комьютер Provaider с IP адресом -200.200.200.10/24), содержащий данные по всем сайтам сети (Comp2, Comp5, Comp8).

Сеть 4 – маршрутизатор Router3 выводит городскую сеть в интернет через коммутатор Switch3 (сеть 210.210.210.0/24). На Comp8 (IP адрес 210.210.210.8/24, шлюз 210.210.210.3/24.) установлен DNS и Web сервер с сайтом.

Маршрутизаторы имеют по два интерфейса:

Router1 – 192.168.1.1/24 и 200.200.200.1/24.
Router2 – 10.0.0.1/8 и 200.200.200.2/24.
Router3 – 210.210.210.3/24 и 200.200.200.3/24.

Задача:
1 – настроить сети организаций;
2 – настроить DNS сервер провайдера;
3 – настроить статические таблицы маршрутизации на роутерах;
4 – проверить работу сети – на каждом из компьютеров - Comp4, Comp7 и Comp8. С каждого из них должны открываться все три сайта корпоративной сети.
В предыдущих лабораторных работах рассматривалась настройка сетевых служб и DNS сервера. Приступим к настройке статической маршрутизации на роутерах. Поскольку на представленной схеме четыре сети, то таблицы маршрутизации как минимум должны содержать записи к каждой из этих сетей – т.е. четыре записи. На роутерах Cisco в таблицах маршрутизации как правило не прописываются пути к сетям, к которым подсоединены интерфейсы роутера. Поэтому на каждом роутере необходимо внести по две записи.

Настройте первый роутер.

Для этого войдите в конфигурацию маршрутизатора и в интерфейсах установите IP адрес и маску подсети. Затем в разделе МАРШРУТИЗАЦИЯ откройте вкладку СТАТИЧЕСКАЯ, внесите данные (рис.2) и нажмите кнопку ДОБАВИТЬ:

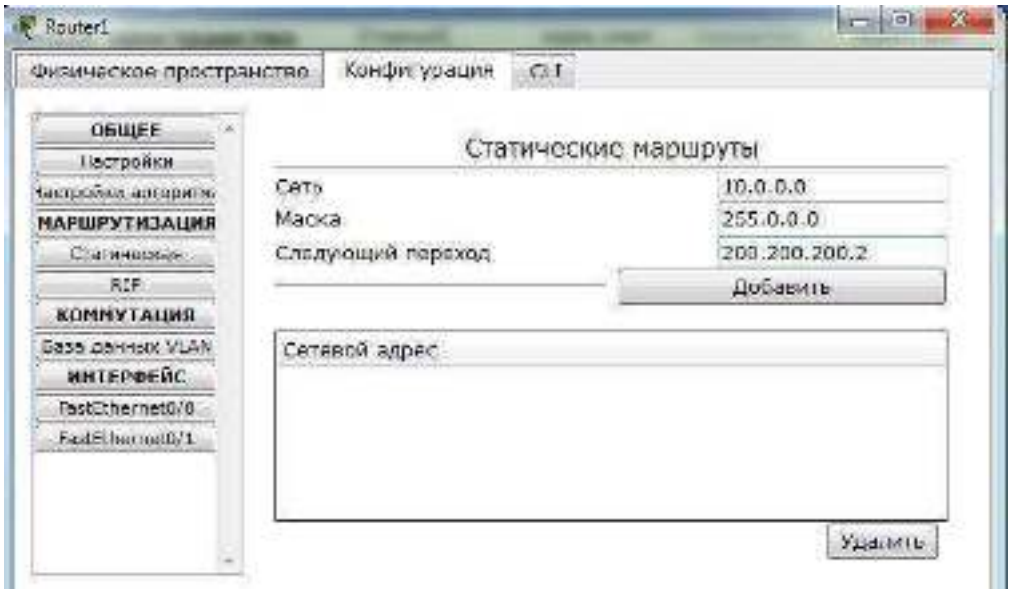


Рис.2. Данные для сети 10.0.0.0/8.

В результате у вас должны появиться две записи в таблице маршрутизации (рис.3):

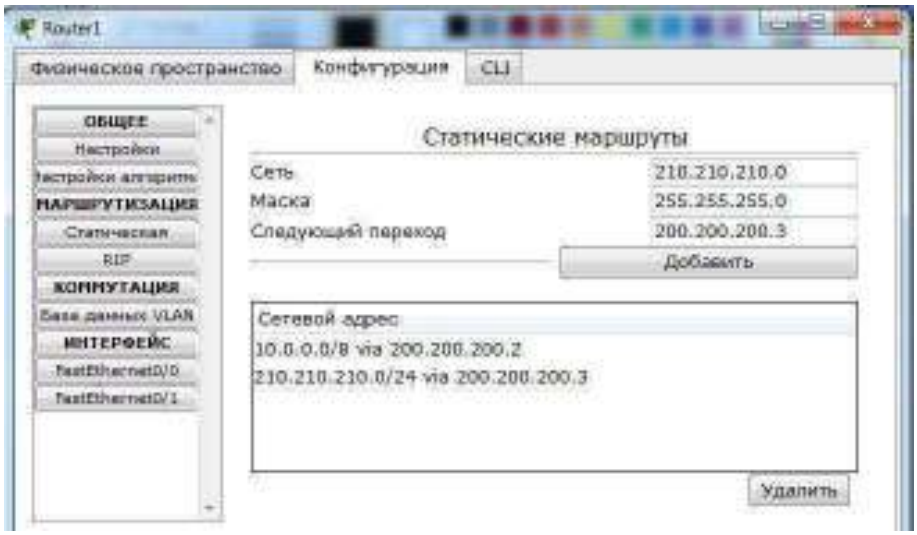


Рис.5.3. Формирование статической таблицы маршрутизации.

Чтобы посмотреть полную настройку таблицы маршрутизации, выберите в боковом графическом меню инструмент ПРОВЕРКА (пиктограмма лупы), щелкните в схеме на роутере и выберите в раскрывающемся меню пункт ТАБЛИЦА МАРШРУТИЗАЦИИ.

После настройки всех роутеров в вашей сети станут доступны IP адреса любого компьютера и вы сможете открыть любой сайт с компьютеров Comp4, Comp7 и Comp8.

Лабораторная работа №18.

Настройка соединений виртуальных частных сетей. Внедрение политик удаленного доступа. Настройка и проверка работы службы преобразования сетевых адресов.

Цель занятия:

- научиться организовывать соединение между двумя сетевыми узлами;
- изучить степень защищенности, передаваемой по туннельному соединению информации с использованием анализатора сетевого трафика

Для выполнения работы студент должен знать:

- технологии виртуальных частных сетей
- протокол IPSec

Оборудование: ПК Pentium (R) Dual-Core E 6700, ОС Windows 7, Ms Word, программы Oracle VirtualBox, Ethernet.

Ход работы

1. Настроить виртуальную сеть между основной ОС и виртуальной машиной Windows 2000. Для этого выполнить следующие действия.
2. В общих настройках виртуальной сети включить адаптер VMnet1 (опция «Enable adapter», рис. 1).

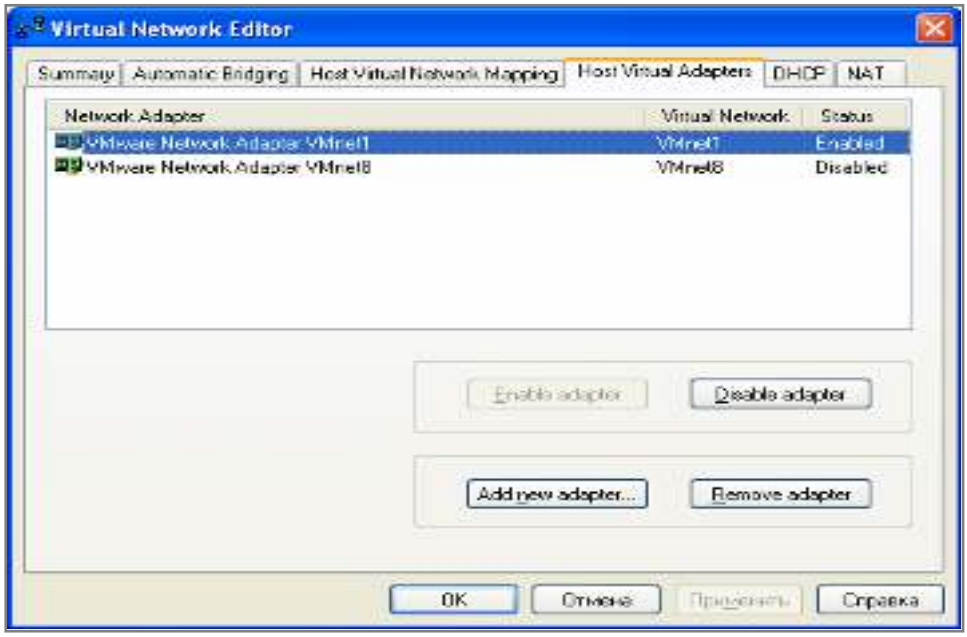
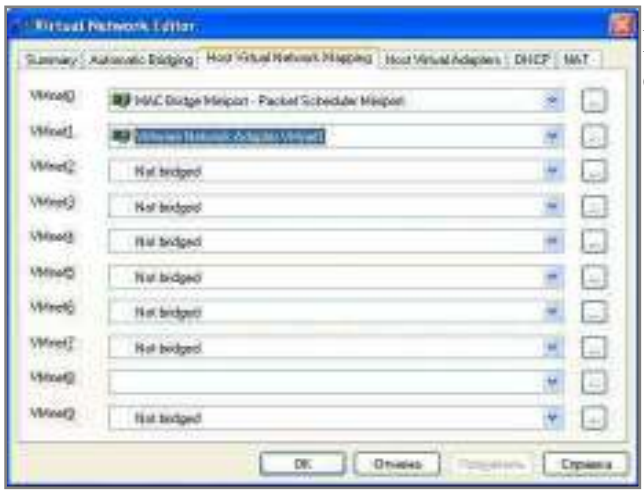


Рис.1. Активация адаптера VMnet1



В разделе «Host Virtual Network Mapping» настроить свойства адаптера VMnet1, указав подсеть 192.168.200.0 (рис. 2).

Рис. 2. Настройка подсети адаптера VMnet1

4. В настройках загружаемой виртуальной машины указать подключение к адаптеру VMnet1 (рис. 3).
5. Установить IP-адрес виртуальной машины 192.168.200.2.
6. Установить IP-адрес адаптера VMnet1 основной ОС (VMware Network Adapter VMnet1) 192.168.200.1.
7. Подключение по локальной сети основной ОС настроить на IP-адрес 192.168.1.128.
8. Добавить в основной ОС входящее подключение VPN, для чего в свойст-вах «Сетевого окружения» запустить «Мастер новых подключений». С по-мощью мастера последовательно установить следующие параметры: «Ус-тановить прямое подключение к другому компьютеру»; «Принимать вхо-дящие подключения»; «Разрешить виртуальные частные подключения»; указать учетную запись, которая будет использована для подключения.
9. Настроить в основной ОС входящее подключение VPN в разделах: «Общие» ⇒ «Разрешить другим пользователям устанавливать частное подключение к моему компьютеру с помощью туннеля в Интернете или другой сети» (установлен).
«Пользователи» ⇒ «Все пользователи должны держать в секрете свои па-роли и данные» (сброшен)
«Сеть» ⇒ «Протокол Интернета (TCP/IP)» ⇒ «Разрешить звонящим дос-туп к локальной сети» (установлен)
«Сеть» ⇒ «Протокол Интернета (TCP/IP)» ⇒ «Указать IP-адреса явным образом» (192.168.1.128 — 192.168.1.254)
«Сеть» ⇒ «Клиент для сетей Microsoft» (установлен)
«Сеть» ⇒ «Служба доступа к файлам и принтерам сетей Microsoft» (уста-новлен)
Остальные параметры оставить по умолчанию.

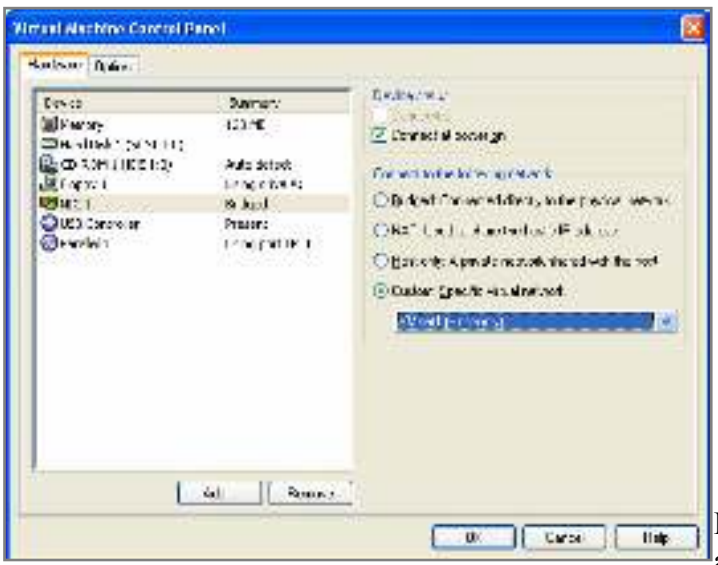


Рис. 3. Настройка адаптера виртуальной машины на адаптер VMnet1

10. Добавить в ОС виртуальной машины подключение к виртуальной частной сети через Интернет со следующими параметрами:
«IP-адрес компьютера, к которому осуществляется подключение» (IP-адрес назначения): 192.168.200.1
«Безопасность» ⇒ «Требуется шифрование данных» (сброшен)
«Сеть» ⇒ «Тип вызываемого сервера VPN» ⇒ «Туннельный протокол точка-точка (PPTP)»

«Сеть» ⇒ «Тип вызываемого сервера VPN» ⇒ «Настройка» ⇒ «Программное сжатие данных» (сброшен)
«Сеть» ⇒ «Клиент для сетей Microsoft» (установлен)
«Сеть» ⇒ «Служба доступа к файлам и принтерам сетей Microsoft» (установлен)

11. Чтобы предотвратить возможность сетевого доступа к файлам и каталогам основной ОС с виртуальной машины в обход туннеля VPN, необходимо дополнительно установить следующие параметры для соединения VMnet1 в основной ОС:
«Общие» ⇒ «Протокол Интернета (TCP/IP)» ⇒ «Дополнительно» ⇒ «WINS» ⇒ «Отключить NetBIOS через TCP/IP»
«Общие» ⇒ «Клиент для сетей Microsoft» (сброшен)
«Общие» ⇒ «Служба доступа к файлам и принтерам сетей Microsoft» (сброшен)
Аналогичные параметры должны быть установлены для подключения к локальной сети в ОС виртуальной машины (тоже, фактически, VMnet1):
«Общие» ⇒ «Протокол Интернета (TCP/IP)» ⇒ «Дополнительно» ⇒ «WINS» ⇒ «Отключить NetBIOS через TCP/IP»
«Общие» ⇒ «Клиент для сетей Microsoft» (сброшен)
«Общие» ⇒ «Служба доступа к файлам и принтерам сетей Microsoft» (сброшен)

Установить виртуальное частное подключение. Выяснить адрес, выделенный клиенту, а также адрес сервера. При установленном параметре «Раз-решить звонящим доступ к локальной сети» подключившийся таким образом клиент становится узлом локальной сети, но только на сетевом уровне модели OSI и выше.

Анализ защищенности передаваемой информации

13. На втором рабочем месте запустить произвольный web-сервер.
14. Запустить анализатор трафика и настроить его на перехват пакетов, пере-даваемых виртуальным сетевым адаптером VMnet1.
15. Отправить из ОС виртуальной машины несколько ECHO-запросов в адрес сервера двумя способами: сначала напрямую через сеть VMnet1 (адрес сер-вера 192.168.200.1), а затем через туннельное соединение (адрес сервера необходимо выяснить при помощи диалогового окна состояния соединения). Обратите внимание, что пакеты, посылаемые через туннельное соединение, не опознаются как ICMP-пакеты. Поскольку шифрование пере-даваемой информации и программное сжатие отключены, то содержимое исходного IP-пакета сохраняется в первоначальном виде. Изменения в передаваемой информации заключаются только в том, что к исходному пакету добавляется заголовок протокола PPTP, который затем снимается при выходе пакета из туннеля.
16. Перевести IP-адреса источников и приемников ECHO-запросов (всего 4 различных адреса) в шестнадцатеричную систему исчисления. Найти эти адреса в перехваченных пакетах. Убедиться, что при туннелировании IP-адреса остаются неизменными и могут быть восстановлены в случае пере-хвата трафика. Привести пакеты ECHO-запросов, отправленных напрямую и через туннель, и выделить в них соответствующие IP-адреса.
17. Запустить на виртуальной машине Internet Explorer и подключиться к за-пущенному в локальной сети web-серверу. При помощи анализатора трафика посмотреть пакеты, передаваемые через интерфейс VMnet1. Найти HTTP-запросы, отправляемые на 80 (50h) порт web-сервера, а также ответы сервера, отправляемые с 80 порта. Текст HTTP-запроса начинается со слова GET, следующего за ним пробела и далее URL запрашиваемого ресурса. Сравнить эти пакеты с пакетами, передаваемыми по локальной сети. В чем выражено отличие этих пакетов?
18. Разорвать виртуальное соединение.
19. Включить шифрование передаваемой информации, для этого в свойствах соединения в ОС виртуальной машины установить следующий параметр:
Безопасность ⇒ Шифрование данных

- Установить виртуальное соединение. Отправить из ОС виртуальной машины несколько ECHO-запросов через туннельное соединение. Просмотреть перехваченный трафик, есть ли возможность установить, пакеты какого содержания передавались?
  - Зашифрованы ли поля заголовков?
- Какая информация может быть перехвачена злоумышленником в случае его подключения к линии связи?

Контрольные вопросы

1. Каким образом технология VPN обеспечивает конфиденциальность данных?
2. Каким образом технология VPN обеспечивает целостность данных?
3. Почему при использовании технологии VPN IP-адреса внутренней сети недоступны внешней сети?

Лабораторная работа №19.

Установка, настройка и использование программных брандмауэров. Защита от сетевых атак. Имитация сетевой атаки на сетевые службы. Анализ журналов.

Цель : научиться защищать сетевой компьютер и настраивать брандмауэр.

Ход работы

Задание 1. Подготовьте компьютер для выполнения лабораторной работы:

- Запустите виртуальную машину VirtualBox. Перейдите в полноэкранный режим работы. Выполняйте остальные задания лабораторной работы в виртуальной машине.

Задание 2. Создайте новую политику IP-безопасности на локальном компьютере:

Откройте оснастку Управление политикой безопасности IP:

1. откройте диалоговое окно Запуск программ (Пуск/Выполнить);
2. введите команду mmc и нажмите клавишу ENTER;
3. выполните команду меню Консоль/Добавить или удалить оснастку;
4. откройте окно с доступными оснастками с помощью кнопки Добавить;
5. выберите в списке элемент Управление политикой безопасности IP и добавьте его с помощью кнопки Добавить;
6. завершите добавление оснастки кнопкой Готово;
7. закройте диалоговое окно Добавить изолированную оснастку;
8. закройте диалоговое окно Добавить/Удалить оснастку с помощью кнопки OK.

Активизируйте оснастку Политика безопасности IP на «Локальный компьютер». Справа отобразятся установленные по умолчанию политики. Запустите мастер создания политик безопасности:

1. вызовите контекстное меню оснастки Политика безопасности IP на «Локальный компьютер»
2. выполните команду Создать политику безопасности IP….

Ознакомьтесь с информацией мастера и щелкните по кнопке Далее.

Установите Имя политики безопасности IP:

1 введите в поле Имя – My_politic.

2 введите в поле Описание – Это политика IP безопасности локального компьютера и щелкните по кнопке Далее.

Настройте политику безопасного соединения. Для этого установите флажок Использовать правило по умолчанию и щелкните по кнопке Далее.

Установите Способ проверки подлинности правила отклика по умолчанию:

1 активизируйте Использовать данную строку для защиты обмена ключами;

2 введите в нижнее поле 123456789;

3 закройте окно кнопкой Далее.

Закройте мастера создания политики безопасности кнопкой Готово. Откроется диалоговое окно Свойства: My_politic.

Запустите Мастер правил безопасности и настройте правила безопасности:

1. запустите мастер кнопкой Добавить;
2. ознакомьтесь с описанием мастера и - Далее;
3. выберите Это правило не определяет туннель и щелкните Далее;
4. выберите Локальные сетевые подключения и щелкните Далее;
5. выберите Использовать сертификат данного центра сертификации (ЦС);
6. щелкните Обзор и выберите любой сертификат, кнопка Далее;
7. в списке фильтров IP выберите Полный IP трафик и щелкните Далее;
8. добавьте новое действие фильтра:
   - щелкните по кнопке Добавить;
   - ознакомьтесь с описанием запустившегося мастера и - Далее;
   - введите в поле Имя – My_filter и щелкните по кнопке Далее;
   - выберите Разрешить и щелкните по кнопке Далее;
   - завершите добавление нового действия кнопкой Готово.
9. активизируйте созданное вами действие и измените его параметры:

- щелкните по кнопке Изменить;
- выберите Согласовать безопасность;
- щелкните по кнопке Добавить и выберите Шифрование и обеспечение целостности;
- установите флажок Принимать небезопасную связь, но отвечать с помощью IPSEC и щелкните по кнопке Далее;

10. завершите работу мастер кнопкой Готово.

Добавьте в политику фильтр для блокировки всех входящих подключений:

- отключите использование мастера (флажок Использовать мастер);
- откройте диалоговое окно Созданий новых правил кнопкой Добавить;
- откройте диалоговое окно Добавление фильтра кнопкой Добавить;
- добавьте новый фильтр:

1. сбросьте флажок Использовать мастер;
2. откройте диалоговое окно Свойства: Фильтр кнопкой Добавить;
3. в поле Адрес источника пакетов выберите Любой адрес IP;
4. в поле Адрес назначения пакетов выберите Мой IP адрес;
5. установите флажок Отраженный для блокировки приходящих пакетов;
6. установите протокол TCP для фильтрации (вкладка Протокол\раскрывающийся список Выберите протокол);
7. завершите настройку нового фильтра кнопкой ОК;

11. закройте диалоговое окно Список фильтров кнопкой ОК.
12. завершите добавление нового правила кнопкой ОК.

Закройте диалоговое окно Свойства: My_politic.

Активизируйте выбранную политику (контекстное меню созданной политики/Назначить).

Проверьте работу политики, воспользовавшись утилитой ping на другом компьютере. Если политика настроена верно, то утилита ping выдаст сведения о том что данный компьютер недоступен.

Задание 3.Настройте фильтрацию IP -трафика.

- Откройте диалоговое окно свойств Подключения по локальной сети (Пуск/Панель управления/Сетевые подключения).
- Откройте диалоговое окно Свойства: Протокол Интернета (TCP/IP) и щелкните по кнопке Дополнительно.
- Перейдите на вкладку Параметры.
- Откройте окно Фильтрация TCP/IP с помощью кнопки Свойства.
- Установите TCP-порты, которые можно использовать:
  - выберите в разделе TCP-порты переключатель Только и щелкните по кнопке Добавить;
  - введите номер порта для протокола HTTPS – 443;
  - аналогично добавьте порты
    - для протокола отправки почты SMTP – 25;
    - для протокола получения почты POP3 – 110;
    - протокол FTP – 21;
    - протокол Telnet - 23.
  - Щелкните ОК для применения параметров.
- Запретите использование протокола Telnet.
- Закройте окно Дополнительные параметры TCP/IP кнопкой ОК.
- Закройте окно Свойства: Протокол Интернета (TCP/IP) кнопкой ОК.
- Проверьте настроенную фильтрацию. Для этого подключитесь по протоколу Telnet с другого компьютера (программа Telnet входит в состав ОС Windows и используется для работы на удаленном компьютере в командной строке).

Задание 4. Настройте брандмауэр Windows:

1. Откройте настройки брандмауэра (Пуск/Панель управления/Центр обеспечения безопасности/Брандмауэр Windows).
2. Разрешите доступ браузеру Internet Explorer к Интернету:
   1. перейдите на вкладку Исключения и щелкните по кнопке Добавить программу.
   2. выберите в списке Internet Explorer и щелкните по кнопке ОК.
3. Включите ведение журнала безопасности:
   1. перейдите на вкладку Дополнительно;

2. щелкните по кнопке Параметры в разделе Ведение журнала безопасности;
3. включите запись пропущенных и успешных пакетов;
4. сохраните сделанные изменения кнопкой ОК.

4. Завершите конфигурирование брандмауэра кнопкой ОК.
5. Подключитесь к сети Интернет с помощью браузера Internet Explorer Если все настроено правильно, то вы сможете выйти в Интернет, в противном случае брандмауэр выдаст сообщение о том, что какая-то программа пытается получить доступ в Интернет.

Оформить отчет по работе.

Контрольные вопросы

1. Дайте определение межсетевого экрана.
2. Перечислите функции экранирования, которые выполняет межсетевой экран.
3. Перечислите функции брандмауэра?
4. По каким признакам классифицируются межсетевые экраны?

Лабораторная работа №20.

Установка настройка и использование систем обнаружения вторжений. Имитация сетевых атак. Анализ работы системы обнаружения сетевых вторжений.

Цель: Получить сведения о том, как осуществляется защита с помощью систем обнаружения и предотвращения вторжений. Научиться использовать SNORT.

Теоретические сведения

Система обнаружения вторжений (IDS) — программное или аппаратное средство, предназначенное для выявления фактов неавторизованного доступа в компьютерную систему или сеть либо несанкционированного управления ими в основном через Интернет.

Сетевая система обнаружения вторжений (англ. network intrusion detection system, NIDS) — система обнаружения вторжений, которая отслеживает такие виды вредоносной деятельности, как DoS атаки, сканирование портов или даже попытки проникновения в сеть.

В пассивной IDS при обнаружении нарушения безопасности, информация о нарушении записывается в лог приложения, а также сигналы опасности отправляются на консоль и/или администратору системы по определенному каналу связи. В активной системе, также известной как Система Предотвращения Вторжений (IPS — Intrusion Prevention system (англ.)), IDS ведет ответные действия на нарушение, сбрасывая соединение или перенастраивая межсетевой экран для блокирования трафика от злоумышленника. Ответные действия могут проводиться автоматически либо по команде оператора.

Обнаружение проникновения позволяет организациям защищать свои системы от угроз, которые связаны с возрастанием сетевой активности и важностью информационных систем. При понимании уровня и природы современных угроз сетевой безопасности, вопрос не в том, следует ли использовать системы обнаружения проникновений, а в том, какие возможности и особенности систем обнаружения проникновений следует использовать.

Snort — свободная сетевая система предотвращения вторжений (IPS) и обнаружения вторжений (IDS) с открытым исходным кодом, способная выполнять регистрацию пакетов и в реальном времени осуществлять анализ трафика в IP-сетях.

Выполняет протоколирование, анализ, поиск по содержимому, а также широко используется для активного блокирования или пассивного обнаружения целого ряда нападений и зондирований, таких как попытки атак на переполнение буфера, скрытое сканирование портов, атаки на веб-приложения, SMB-зондирование и попытки определения операционной системы. Программное обеспечение в основном используется для предотвращения проникновения, блокирования атак, если они имеют место.

Доступны версии программы, работающие под управлением операционных систем Windows NT, Linux, BSD, Mac OS X, а также некоторых других. В соответствии с предложенной выше классификацией, Snort является сетевой СОА, основанной на сигнатурном анализе. Сигнатуры атак описываются при помощи правил — специальных синтаксических конструкций, позволяющих выявлять интересующую администратора информацию в полях заголовков и содержимом передаваемых по сети пакетов. Кроме того, в Snort реализовано несколько препроцессоров, выполняющих более сложные операции по анализу трафика, такие, например, как дефрагментация IP-пакетов,отслеживание TCPсоединений и выявление попыток сканирования портов.

Ход работы:

1) Установка и запуск программы

1. Для обеспечения возможности перехвата сетевых пакетов программой Snort необходима предварительная установка и запуск службы WinPcap (WinPcap_3_1.exe).

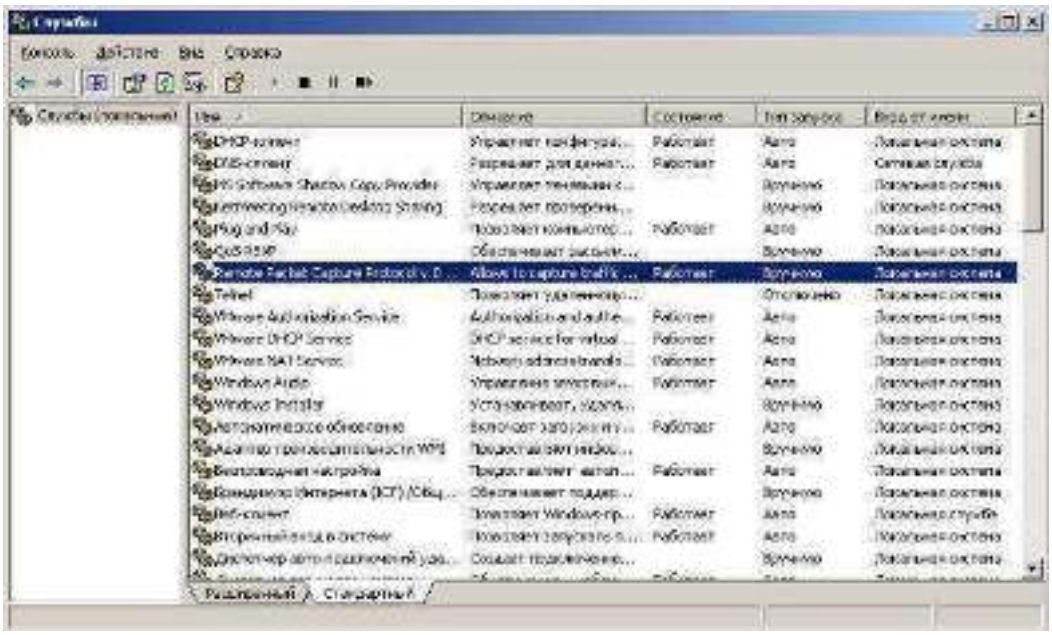


Рис. 4.1. Служба WinPcap в перечне служб

2. Для установки СОА Snort версии 2.4.3 необходимо запустить файл «Snort_243_Installer.exe» и ответить на интересующие программу-инсталлятор вопросы. По умолчанию Snort устанавливается в каталог «C:\snort», а его исполняемый файл располагается в каталоге «C:\snort\bin». Запуск СОА Snort осуществляется из командной строки, в табл. 4.1 приведен неполный список параметров, с которыми может производиться запуск. Чтобы вывести этот список на экран во время работы, необходимо выполнить команду snort -?

Таблица 4.1

Список параметров COA Snort

Параметр	Описание
-c <rules>	Использовать файл правил <rules>.
-E	Добавлять предупреждения (alerts) в журнал регистрации событий Windows NT (не создавая log-файла).
-h<hn>	Задать домашнюю сеть (home network) <hn>.
-i<if>	Подключиться к сетевому интерфейсу номер <if> (список интерфейсов можно получить при помощи команды snort -W).
-I	Добавлять к предупреждению наименование интерфейса.
-K<mode>	Режим регистрации предупреждений: pcap (по умолчанию) — в двоичном формате, ascii — в текстовом формате,none — без регистрации.
-l<ld>	Сохранять результаты регистрации в каталог <ld>.
-L<file>	Сохранять результаты регистрации в указанный файл формата tcpdump (будет располагаться в каталоге, предварительно указанном параметром -l).
-n<cnt>	Завершить работу программы после получения <cnt> пакетов.
-N	Не сохранять в файлах регистрации содержимого пакетов (сохраняется лишь текст предупреждений).
-p	Анализировать только пакеты, отправленные на локальный адрес, и широковещательные пакеты.
-r<tf>	Прочитать и обработать файл <tf>, записанный в формате tcpdump.
-S<n=v>	Установить значение переменной n файла правил, равной v.
-U	Записывать временные метки в универсальном скоординированном времени (UTC).
-v	Отображать на экране заголовки всех перехваченных пакетов.
-V	Показать версию Snort.
-W	Показать доступные сетевые интерфейсы.
-X	Сохранять в файле журнала регистрации событий содержимое перехваченных пакетов в «сыром» виде, начиная с уровня link модели OSI.

Параметр	Описание
-y	Добавить месяц, день и год в отображаемые и сохраняемые временные отметки
-?	Показать помощь по параметрам командной строки

3. Для проверки работоспособности COA Snort рекомендуется выполнить следующие действия.

Задание

1.Установить COA Snort.

2.Вывести на экран список доступных сетевых интерфейсов командой

snort –W

3.Запустить Snort на выбранном интерфейсе в режиме анализатора пакетов с выводом информации на экран, указав программе завершить работу после приема третьего пакета:

snort –v–i1–n3

4.Выполнить любые действия, которые приведут к отправке или приему сетевых пакетов (например, отправить эхо-запросна любойIP-адрескомандой ping). Убедиться, что пакеты перехватываются и отображаются на экране.

4. Описание языка правил

Рассмотрим краткое описание языка правил, на котором задаются сигнатуры атак, обнаруживаемых СОА Snort. Полное описание языка правил содержится в файле документации «c:\snort\doc\snort_manual.pdf»

Правила записываются в одну строку, если возникает необходимость перенести текст правила на следующую строку, необходимо добавить в конце строки символ обратной косой черты «\».

Правила состоят из двух частей: заголовка и набора атрибутов. Заголовок, в свою очередь, состоит из:

1.Указания действия, которое необходимо выполнить (alert,log,pass и др.).

2.Протокола (tcp,udp,icmp,ip).

3.IP-адресаи маски подсети источника и приемника информации, а также информации о портах источника и приемника.

Действие alert заключается в генерации предупреждающего события и сохранении содержимого пакета для дальнейшего анализа. Действие log предполагает сохранение пакета без генерации предупреждения. Действие pass означает пропуск пакета (его игнорирование). Существует также ряд более сложных действий, которые здесь не рассматриваются.

Текст атрибутов располагается в скобках, каждая пара атрибут — значение имеет вид <атрибут>: <значение>;. Значения строковых атрибутов записываются в кавычках.

Рассмотрим пример простого правила (одна строка):

alert <протокол> <адрес_подсети1>[/маска_подсети1] <порт1> <направление> <адрес_подсети2>[/маска_подсети2] <порт2> ([msg:"Текст сообщения";] [другие_атрибуты])

где

−alert — действие, которое необходимо выполнить при обнаружении пакета, удовлетворяющего данному правилу, и которое заключается в генерации «предупреждения» — записи в журнале регистрации

−<протокол> — наименование протокола (tcp, udp, icmp, ip)

−<адрес_подсети>[/маска_подсети] — IP-адрес и маска подсети,

либо IP-адресузла участника обмена в формате:192.168.247.0/24, либо

192.168.247.1

−<порт> — номер порта либо диапазон портов в формате1:1024 для обозначения диапазона портов от 1 до 1024,1024: — с номерами больше или равными 1024, или:1024 — меньше или равными 1024 соответственно

−<направление> — обозначение направления в виде->,<- или<>

Вместо IP-адресови номеров портов могут использоваться псевдонимыany, являющиеся заменителем любого значения.

Атрибуты являются наиболее значимой частью правил, так как позволяют искать интересующую информацию в полях заголовков и содержимом пакетов. Существует четыре категории атрибутов:

−meta-data— предоставляют информацию о правиле, но не влияют на процесс обнаружения;

−payload — атрибуты данного типа предназначены для поиска информации в «полезной нагрузке» (содержимом) пакета;

−non-payload— предназначены для поиска информации в заголовках паке-

тов;

−post-detection— определяют поведение системы после обнаружения пакета, удовлетворяющего правилу.

Втабл. 4.2 приведен неполный список атрибутов, которые могут быть использованы при написании правил.

Если известно местонахождение интересующей информации в пакете, то целесообразно ограничить область поиска при помощи модификаторов

offset idepth, так как это существенно сократит время, затрачиваемое на анализ пакета.

Таблица 4

Список атрибутов COA Snort

Атрибут	Описание
meta-data	
`msg: "<текст>";`	Сообщение, которое добавляется в журнал регистрации при активации правила.
`sid: <идентификатор>;`	Уникальный номер, используемый для идентификации правил. Идентификаторы от 100 до 1 000 000 используются для правил, включенных в дистрибутив Snort. Для локальных правил следует использовать значения больше 1 000 000.
`rev: <номер_редакции>;`	Целое число, служащее для обозначения номера редакции правила.
`classtype: <имя_класса>;`	Используется для обозначения класса атаки. Полный список классов приведен в документации по Snort..
`priority: <приоритет>;`	Целое число, используемое для переопределения приоритета, задаваемого указанным ранее классом атаки, или для назначения приоритета новому правилу. Наивысший приоритет — 1, типично значение атрибута составляет от 1 до 4.
payload	
`content: [!]"<строка>";`	Позволяет искать заданную подстроку в содержимом полезной нагрузки пакета. Восклицательный знак означает отсутствие указанной информации в пакете. По умолчанию данный атрибут является чувствительным к регистру. Для обозначения двоичных данных следует использовать шестнадцатеричные значения, отделенные вертикальными чертами: `|00 5C|`. Атрибут `content` имеет несколько модификаторов, которые могут располагаться
`nocase;`	следом за ним. Модифицирует стоящий ранее атрибут
`depth: <число_байт>;`	`content`, делая его нечувствительным к регистру. Значение атрибута (в байтах) определяет, как

Атрибут	Описание
offset: <число_байт>;	Значение атрибута определяет, сколько байтов полезной нагрузки следует пропустить. Поиск будет вестись, начиная с число_байт+1-гобайта.
distance: <число_байт>;	Атрибут похож на depth, но указывает, сколько байт необходимо пропустить после предыдущей совпавшей подстроки перед продолжением поиска.
within: <число_байт>;	Атрибут указывает системе искать совпадения лишь в первых число_байт, начиная с конца предыдущей совпавшей подстроки.
non-payload	
dsize: <размер>;	Сравнить размер полезной нагрузки с заданным. Возможно указание диапазонов значений с использованием знаков >,< и<>. Например:>128, 300<>500 (от 300 до 500).
flags: [!|*|+]<флаги>	Проверить, установлены ли указанные флаги в принятом TCP-пакете.Флаги записываются подряд без пробелов и обозначаются следующим образом: F — FIN (LSB в байте флагов) S — SYN R — RST P — PSH A — ACK U — URG 1 — Резерв 1 (MSB в байте флагов) 2 — Резерв 2 0 — флаги не установлены Могут быть дополнительно использованы следующие модификаторы: ! — указанные флаги не установлены * — установлен хотя бы один из указанных + — установлены указанные и любые другие
itype: <тип>;	Сравнить тип ICMP сообщения с указанным. Возможно указание диапазонов значений с использованием знаков >,< и<> (см. выше).
icode: <код>;	Сравнить код ICMP сообщения с указанным.

Вместо IP-адресовмогут использоваться переменные, заданные выше по тексту следующим образом:

var <имя_переменной> <значение_переменной>

Чтобы сослаться на переменную далее в тексте, перед ее именем следует поставить знак доллара $.

В текст файла правил можно включать комментарии, которые отделяются знаком #. Вся информация справа от этого знака и до конца строки считается комментарием и не интерпретируется системой:

#<комментарий>

Рассмотрим пример задания двух переменных последующего их использования в правиле, фильтрующем входящие ICMP-пакетыECHO (тип 8):

```
#Глобальные переменные
var HOME_NET
192.168.247.1 var
EXTERNAL_NET !$HOME_NET
#Обнаружение эхо-запросов (ping'ов)
alert icmp $EXTERNAL_NET
any ->$HOME_NET any (msg:"Incoming ECHO REQUEST"; itype: 8;)
```

Использование СОА Snort

СОА Snort можно использовать как анализатор трафика, обладающий значительными возможностями по фильтрации пакетов. Например, можно создать файл с правилами, использующими исключительно действия типа log. В результате из входящего потока данных будут отобраны и сохранены пакеты, удовлетворяющие указанным правилам. Так как по

умолчанию журнал ведется в двоичном формате tcpdump, он может быть импортирован почти всеми специализированными программами анализа трафика. Обычно эти программы позволяют наглядно отображать содержимое пакетов, но не обладают такими возможностями по их фильтрации, как Snort.

Для запуска Snort в режиме анализатора трафика, как и для запуска его в режиме системы обнаружения атак, необходимо выполнить следующую команду в командной строке Windows:

snort -i<интерфейс>-c<файл_конфигурации>-l<путь_к_журналу>

где

<интерфейс> — номер интерфейса, полученный в результате выполнения команды snort –W

<файл_конфигурации> — путь к файлу, в котором хранятся настройки программы и правила обнаружения

<путь_к_журналу> — путь к каталогу, в котором необходимо сохранить файл журнала

Пример:

snort -i3-c ../etc/my.conf-l ../log

Следует обратить внимание, что при записи пути используются не обратные, а прямые «косые черты».

Для завершения работы СОА Snort, необходимо нажать клавиши <Ctrl+C>.

Рассмотрим несколько правил (табл. 4.3), которые позволят обнаруживать атаки, описанные в разделе 2. Текст правил должен записываться в одну строку.

Таблица 4.3

Примеры правил COA Snort

№	Описание	Правило
1	Обнаружение входящих ECHO-запросов(ping'ов)	alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg: "Incoming ECHO REQUEST"; itype: 8;)
2	Обнаружение исходящих ECHO-ответов	alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg: "Outgoing ECHO REPLY"; itype: 0;)
3	Обнаружение больших ICMP-пакетов (атака «Ping of Death»)	alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg: "Incoming large ICMP packet"; dsize: >800;)
4	DoS-атакаWinnuke	alert tcp $EXTERNAL_NET any -> $HOME_NET 135:139 (msg: "DoS Winnuke attack"; flags: U+;)
5	Запрос на подключение к 139 порту (служба SMB) из внешней сети (два варианта)	alert tcp $EXTERNAL_NET any -> $HOME_NET 139 (msg: "NETBIOS SMB IPC$ share access"; flags: A+; content: "|00|"; offset: 0; depth: 1; content: "|FF|SMB|75|"; offset: 4; depth: 5; content: "\\IPC$|00|"; nocase;) alert tcp $EXTERNAL_NET any -> $HOME_NET 139 (msg:"NETBIOS SMB IPC$ share access (unicode)"; flags: A+; content: "|00|"; offset: 0; depth: 1; content: "|FF|SMB|75|"; offset: 4; depth: 5; content:

№	Описание	Правило
		"|5c00|I|00|P|00|C|00|$| 00|"; nocase;)
6	Запрос на подключение к 445 порту (служба SMB) из внешней сети (два варианта)	alert tcp $EXTERNAL_NET any -> $HOME_NET 445 (msg: "NETBIOS SMB IPC$ share access"; flags: A+; content: "|00|"; offset: 0; depth: 1; content: "|FF|SMB|75|"; offset: 4; depth: 5; content: "\\IPC$|00|"; nocase;) alert tcp $EXTERNAL_NET any -> $HOME_NET 445 (msg:"NETBIOS SMB IPC$ share access (unicode)"; flags: A+; content: "|00|"; offset: 0; depth: 1; content: "|FF|SMB|75|"; offset: 4; depth: 5; content: "|5c00|I|00|P|00|C|00|$|00|"; nocase;)
6	Обнаружение сканирования портов методом NULL.	alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"NULL port scanning"; flags:!FSRPAU;)
7	Обнаружение сканирования портов методом XMAS.	alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"XMAS port scanning"; flags:FPU+;)

Задание:

5.Создать в каталоге «\snort\etc» файл «my.conf», содержащий следующие строки:
var HOME_NET <IP-адрес_COA>var EXTERNAL_NET !$HOME_NET

6.Добавить в файл «my.conf» правило, позволяющее обнаруживать входящие ECHO-запросы.Проверить, происходит ли обнаружение, запустив COA из каталога «\snort\bin» следующей командой (из «командной строки»):
snort -i<интерфейс>-c ../etc/my.conf-l ../log
Для проверки выполнить несколько ECHO-запросовс другого компьютера, используя команду:
ping <IP-адрес_COA>

7.Дополнить файл «my.conf» правилами, указанными в табл. 4.3. Для проверки обнаружения подключений к службе SMB использовать команду:
net use \\<IP-адрес_COA>\IPC$"" /user:""

К какому порту производится подключение? Как это зависит от используемой операционной системы?

Выявление факта сканирования портов

ВСОА Snort встроен программный модуль, позволяющий выявлять сканирование портов защищаемой системы. Алгоритм, обнаруживающий сканирование, основан на том, что при сканировании портов существенно увеличивается количество исходящих TCP-пакетовс установленным флагом RST. Установка этого флага на отправляемом в ответ пакете означает, что порт, к которому производилось обращение, закрыт. Таким образом, анализируя количество пакетов с установленным флагом RST, можно обнаружить факт сканирования портов системы.

Программный модуль, или препроцессор, как его называют разработчики Snort, инициализируется из файла конфигурации следующим образом. В конфигурационный файл следует добавить следующие строки:

preprocessor flow: stats_interval 0 hash 2

preprocessor sfportscan: proto { <протокол> } scan_type

{<тип_сканирования> } sense_level

{<чувствительность> } logfile { <файл_с_отчетом> }

Первая строка предназначена для инициализации препроцессора Flow, без которого модуль обнаружения сканирования не работает. Вторая строка инициализирует препроцессор sfPortscan, при этом задаются следующие параметры (жирным шрифтом показаны рекомендуемые значения):

<протокол> — анализируемый протокол (tcp,udp,icmp,ip_proto,all)

<тип_сканирования> — выявляемые типы сканирования (portscan,portsweep,decoy_portscan,distributed_portscan,all)

<чувствительность> — чувствительность (low, medium, high)

<файл_с_отчетом> — имя файла, в который будет помещен отчет об обнаруженных попытках сканирования портов

Файл с отчетом будет располагаться в каталоге, указанном параметром -c при запуске Snort. Чувствительность определяет перечень анализируемой информации и в итоге сказывается на вероятности «ложной тревоги» (для high она наибольшая). За более подробной информацией о параметрах модуля sfPortscan следует обращаться к документации на Snort.

Приведем пример настройки модуля sfPortscan: preprocessor flow: stats_interval 0 hash 2

preprocessor sfportscan: proto { all } scan_type { all } sense_level { medium } logfile { portscan.log }

При данной настройке Snort будет выявлять все описанные в разделе 2 методы сканирования портов. Необходимо отметить, что две и больше процедуры сканирования портов, выполненные во время одного сеанса работы Snort, будут отражены в файле регистрации событий только один раз. Это остается справедливым даже в том случае, когда используются разные методы сканирования. Таким образом, для поверки возможности обнаружения сканирования, выполняемого разными методами, Snort необходимо закрывать и запускать снова.

Задание

8.Дополнить файл «my.conf» приведенными выше строками для настройки препроцессора sfPortscan. С использованием утилиты nmap проверить, происходит ли обнаружение попыток сканирования портов защищаемого узла. Использовать следующие команды для запуска сканирования:

nmap <IP-адрес_СОА> -v -sT -p <диапазон_портов> — для сканирования методом с полным циклом подключения (метод Connect)nmap <IP-адрес_СОА> -v -sS -p <диапазон_портов> — для сканирования с неполным циклом подключения (метод SYN)

nmap <IP-адрес_СОА> -v -sN -p <диапазон_портов> — для сканирования при помощиTCP-пакетасо сброшенными флагами (метод NULL)

nmap <IP-адрес_СОА> -v -sX -p <диапазон_портов> — для сканирования при помощиTCP-пакетасо всеми установленными флагами

(метод XMAS)

9.Какие методы сканирования позволяют практически выявлять наличие открытых портов? Как это зависит от используемой операционной системы? Все ли указанные методы сканирования обнаруживает COA Snort?