ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

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

Петрозаводский филиал ПГУПС

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

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

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по организации и проведению практических занятий

По УП 03.04. Безопасность сетей

Специальность: 09.02.02 Компьютерные сети

Разработчик:
Зав.УВЦ Капоровский В.Е.

2017г

Введение

Методическое пособие по проведению практических работ по УП 03.04. «Безопасность сете й»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».

Настоящее методическое пособие рассчитано на самостоятельную работу обучающихся в учебном кабинете под руководством преподавателя, а также является руководством для преподавателей при подготовке к проведению учебной практики.

Для успешного прохождения учебной практики могут быть использованы теоретические знания полученные обучающимися при прохождении ПМ.03 «Эксплуатация объектов сетевой инфраструктуры».

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.

УП.03.04 «Безопасность сетей»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;
- усвоение студентами основ законодательства об охране труда, системы стандартов безопасности труда, требований правил гигиены труда и производственной санитарии, противопожарной защиты, охраны окружающей среды в соответствии с новыми нормативными и законодательными актами.

Программа учебной практики УП.03.04 «Безопасность сетей», является частью программы подготовки специалистов среднего звена в соответствии с ФГОС СПО по специальности 09.02.02 Компьютерные сети (базовой подготовки) в части освоения основного вида профессиональной деятельности (ВПД): Эксплуатация объектов сетевой инфраструктурыи формирования следующих профессиональных компетенций (ПК):

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.

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

Практическое занятие №1

Применение симметричных криптосистем: шифрование, дешифрование. Использование программ для симметричного шифрования/дешифрования

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*

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 K= {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}

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,

А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А
В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б
Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В
						.	.	.	.	.	.	.	.	.	.	.															
Я	А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю

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:

А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	К	Л
О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П
Е	Ж	З	И	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д

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

Открытый текст	**ЗАЩИТА ИНФОРМАЦИИ**
Ключ	**МОРЕМО РЕМОРЕМОРЕ**
Текст после замены	**УОИОЭО ШТЯЫЯСМГШО**
Шифртекст	**УОИО ЭОШТ ЯЫЯС МГШО**

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

.

Работа с OpenSSL

OpenSSL – это библиотека программного кода, написанная на языке С, 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.

Интерфейс утилиты OpenSSL имеет след. Структуру:

openssl команда [опции команды] [аргументы команды]

openssl list-standart-commands – списокдоступныхкоманд

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 сохранимего

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:

1)

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.

2)-

1. Установить программу Win32OpenSSL_Light (или другую версию OpenSSL). Запустить openssl.exe (папка \bin).
2. 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 зашифрованные файлы и проанализируйте результаты расшифровки.

Практическое занятие №2.

Исследование электронно-цифровой подписи (ЭЦП) на основе алгоритма RSA.

Теоретические сведения.

Технология применения системы ЭЦП предполагает наличие сети абонентов, обменивающихся подписанными электронными документами. При обмене электронными документами по сети значительно снижаются затраты, связанные с их обработкой, хранением и поиском.

Одновременно при этом возникает проблема, как аутентификации автора электронного документа, так и самого документа, т.е. установление подлинности автора и отсутствия изменений в полученном электронном сообщении.

В алгоритмах ЭЦП как и в асимметричных системах шифрования используются однонаправленные функции. ЭЦП используется для аутентификации текстов, передаваемых по телекоммуникационным каналам.

ЭЦП представляет собой относительно небольшой объём дополнительной цифровой информации, передаваемой вместе с подписанным текстом.

Концепция формирования ЭЦП основана на обратимости асимметричных шифров, а также на взаимосвязанности содержимого сообщения, самой подписи и пары ключей. Изменение хотя бы одного из этих элементов сделает невозможным подтверждение подлинности подписи, которая реализуется при помощи асимметричных алгоритмов шифрования и хэш-функций.

Система ЭЦП включает две процедуры: формирование цифровой подписи; проверку цифровой подписи.

В процедуре формирования подписи используется секретный ключ отправителя сообщения, в процедуре проверки подписи - открытый ключ отправителя.

Безопасность системы RSA определяется вычислительной трудностью разложения на множители больших целых чисел. Недостатком алгоритма цифровой подписи RSA является уязвимость её к мультипликативной атаке. Другими словами, алгоритм ЭЦП RSA позволяет хакеру без знания секретного ключа сформировать подписи под теми документами, в которых результат хэширования можно вычислить как произведение результата хэширования уже подписанных документов.

. Алгоритм электронной цифровой подписи (ЭЦП) RSA.

Определение открытого «e» и секретного «d» ключей
(действия отправителя).

1) Выбор двух взаимно простых больших чисел ***p*** и ***q.***
2) Определение их произведения ***n=p·q***.
3) Определение: ***n1= (p-1)(q-1)***.
4) Выбор секретного ключа ***d*** с учетом условий: ***1<d<n1***;
НОД(d,n1)) = 1.
5) Определение значения открытого ключа ***e: e<n,***
e·d = 1(mod(n1)).

Формирование ЭЦП

1) Вычисление хэш-значения сообщения ***M*: *m=h(M).***

2) Для получения ЭЦП шифруем хэш-значение***m***с помощью секретного ключа***d***и отправляем получателю цифровую подпись:$S=m^d(modn)$и открытый текст сообщения***M.***

Аутентификация сообщения –проверка подлинности подписи

1) Расшифровка цифровой подписи***S***с помощью открытого ключа***e***и вычисление её хэш-значения$m'=S(modn)$

2) Вычисление хэш-значения принятого открытого текста***M:***

$m=h(M)$

3) Сравнение хэш-значений***m***и***m'***если***m=m'***то цифровая подпись***S***-достоверна.

Порядок выполнения работы соответствует, приведённому выше алгоритму формирования ЭЦП по схеме RSA.

Процедуру формирования ЭЦП сообщения***M***рассмотрим на следующем простом примере:

Пример вычисления хэш-значения сообщения M: m=h(M) и проверки подлинности ЭЦП.

1) Хешируемое сообщение***M***представим, например, как последовательность целых чисел***312.***В соответствии с приведённым выше алгоритмом формирования ЭЦП RSA выбираем два взаимно простых числа$p=3,q=11,$вычисляем значение$n=p\cdot q=3\cdot 11=33,$выбираем значение секретного ключа$d=7$и вычисляем значение открытого ключа$e=3.$Вектор инициализацииH_0выбираем равным 6(выбирается случайным образом). Хэш-код сообщения $M=312$формируется следующим образом:

$H_1=(M_1+H_0)^2(mod\ n)=(3+6)^2(mod\ 33)=81\ (mod\ 33)=15;$

$H_2=(M_2+H_1)^2\ (mod\ n)=(1+15)^2(mod\ 33)=256\ (mod\ 33)=25;$

$H_3=(M_3+H_2)^2\ (mod\ n)=(2+25)^2(mod\ 33)=729\ (mod\ 33)=3.$

Итоговое хэш-значение:

$m=3.$

2) Для получения ЭЦП шифруем хэш-значение***m***с помощью секретного ключа$d=7$и отправляем получателю цифровую подпись:

$S=m^d(modn)$и открытый текст сообщения***M***

$S=3^7(mod\ 33)=2187\ (mod\ 33)=9$

2.4.3. Проверка подлинности ЭЦП

Расшифровка***S***(т. е. вычисление её хэш-значения***m')*** производится с помощью открытого ключа***e***:

$m'=S^e(mod\ n)=9^3(mod\ 33)=729\ (mod\ 33)=3$

2.4.4. Если сравнение хэш-значений***m'*** и***m*** показывает их равенство, т.е.$m=m'$ то подпись достоверна.

Задание

Для выполнения работы группа студентов разбивается на пары. Каждый студент генерирует свой открытый и секретный ключ. Далее, участники из каждой пары, выполнив хэширование своего сообщения и подписав его каждый своим секретным ключом, обмениваются результатами***(M,S)***и проверяют подлинность полученных сообщений.

Содержание сообщения***M***и числа***p***и***q***выбираются по таблице 1. Вариант выдает преподаватель. Например, для варианта 61 шифруемое слово будет – «область», а начальные параметры для генерации ключей –$p=5,q=7$.

Таблица 1

Исходные данные для выполнения работы

Первая цифра номера варианта										
Шифруемое слово	0	1	2	3	4	5	6	7	8	9
	доступ	добыча	загадка	анализ	защита	подход	область	уровень	сервер	система
H_0	4	5	6	5	4	7	5	6	4	5
Вторая цифра номера варианта										
Параметры шифрования	0	1	2	3	4	5	6	7	8	9
p	7	5	3	11	13	3	5	7	11	5
q	17	7	11	3	17	7	11	13	17	13

Практическое занятие №3
Вычисление хеш-функций, создание и проверка ЭЦП.

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

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.

Работа с OpenSSL

OpenSSL – это библиотека программного кода, написанная на языке С, 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.

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

Ассиметричное шифрование/расшифрование. Генерация ключей.

Как сгенерировать секретный ключ RSA?

Использовать подкоманду genrsa:

по умолчанию длина ключа 512 бит; ключ выводится на стандартный поток

opensslgenrsa

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

opensslgenrsa -des3 -outmykey.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")

С помощью подкоманды passwd можно генерировать хэши паролей.

Утилита dgst используется для подписывания секретным ключом и проверки ЭЦП публичным ключом.

Задание.

1. Установить программу Win32OpenSSL_Light (или другую версию OpenSSL). Запустить openssl.exe (папка \bin).
2. Хеширование.Сравнение файлов

Создайте текстовый файл с любым содержимым text1.txt и его копию text2.txt

Вычислите хэши с помощью алгоритмовхешированиямd4 и sha 2-х файлов text1 и text2.

Проверьте будут ли они одинаковы.

3. Вычислите хеш для произвольного пароля и сравните его с хешем пароля отличающегося от этого пароля на один символ

4. Создайте небольшой произвольный текстовый файл.

С помощью алгоритма RSA:

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

5. Для произвольного файла с помощьюутилиты dgst:

- Создать ЭЦП к файлу.
- проверить ЭЦП
- Внести изменения в файл и проверить ЭЦП.

Практическое занятие №4

Защита от компьютерных вирусов. Настройка и применение программ для защиты компьютера.

Теоретические сведения

Вирусом называется специально созданная программа, которая способна самостоятельно распространяться в компьютерной среде.

В мире не существует единой классификации вирусов, однако можно выделить три группы вирусов:

 файловые вирусы;

 загрузочные вирусы;

 комбинированные файлово-загрузочные вирусы.

Кроме того, вирусы бывают макрокомандные, резидентные и нерезидентные, полиморфные и маскирующиеся (стелс-вирусы).

Файловые вирусы

Как нетрудно догадаться из названия, областью обитания файловых вирусов являются файлы. Вирусы записывают свой код в тело программного файла таким образом, что при запуске программы вирус первым получает управление. Сделав свое черное дело, вирус передает управление зараженной программе, так что пользователь ничего не замечает. При запуске вирус сканирует локальные диски компьютера и сетевые каталоги в поисках очередной жертвы. После того как подходящий программный файл будет найден, вирус записывает в него свой код.

Самый простой способ гарантированно удалить вирусы с компьютера заключается в том, чтобы после форматирования диска компьютера на низком уровне установить операционную систему и прикладные программы с лицензионных дистрибутивов

Загрузочные вирусы

Вторая большая группа вирусов - это так называемые загрузочные вирусы. Распространение и активизация этих вирусов происходит в момент загрузки операционной системы, еще до того, как пользователь успел запустить какую-либо антивирусную программу.

Простые и полиморфные вирусы

Обычные компьютерные вирусы обнаружить достаточно легко, так как в процессе заражения они записывают в заражаемый файл или системную область диска свой собственный код. Автору антивирусной программы достаточно выделить из этого кода уникальную последовательность команд или байт, характерную именно для данного вируса. Такая последовательность носит название сигнатуры.

Затем антивирусная программа уже в автоматическом режиме просматривает все файлы и системные области дисков в поиске сигнатур известных вирусов. Естественно, что проблем с обнаружением таких вирусов нет.

Очень скоро авторы вирусов догадались использовать в своих вирусах алгоритмы шифрования, затрудняющие их обнаружение и выделение сигнатуры. Такие вирусы, получившие название шифрующихся, при заражении новых файлов и системных областей диска шифруют собственный код, пользуясь для этого случайными паролями (ключами). Когда вирус получает управление, он первым делом расшифровывает собственный код.

Сложность обнаружения таких вирусов состоит в том, что код вируса случайным образом изменяется при каждом новом заражении и, соответственно, автору антивируса сложнее выделить сигнатуру такого вируса. Однако, так как шифрующийся вирус все же должен содержать неизменную процедуру расшифровки, то сигнатуру получить можно. Даже простые антивирусные программы способны успешно обнаруживать и удалять вирусы, применяющие алгоритм шифровки.

Вслед за шифрующимися вирусами появилась еще более сложная разновидность вирусов, получившая страшное название вирусов-мутантов. Более научное название вирусов-мутантов - полиморфные вирусы. От шифрующихся вирусов они отличаются тем, что даже процедура расшифровки меняется у разных особей одного вируса. Каждый раз когда вирус заражает новый файл или системную область диска, он полностью изменяется, поэтому из полиморфных вирусов невозможно выделить сигнатуру

Стелс-вирусы

В ходе проверки компьютера антивирусные программы считывают данные - файлы и системные области с жестких дисков и дискет, пользуясь средствами операционной системы и базовой системы ввода/вывода BIOS. Ряд вирусов, после запуска оставляют в оперативной памяти компьютера специальные модули, перехватывающие обращение программ к дисковой подсистеме компьютера. Если такой модуль обнаруживает, что программа пытается прочитать зараженный файл или системную область диска, он на ходу подменяет читаемые данные, как будто вируса на диске нет
Стелс-вирусы обманывают антивирусные программы и в результате остаются незамеченными. Тем не менее, существует простой способ отключить механизм маскировки стелс-вирусов. Достаточно загрузить компьютер с не зараженной системной флешки и сразу, не запуская других программ с диска компьютера (которые также могут оказаться зараженными), проверить компьютер антивирусной программой.

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*

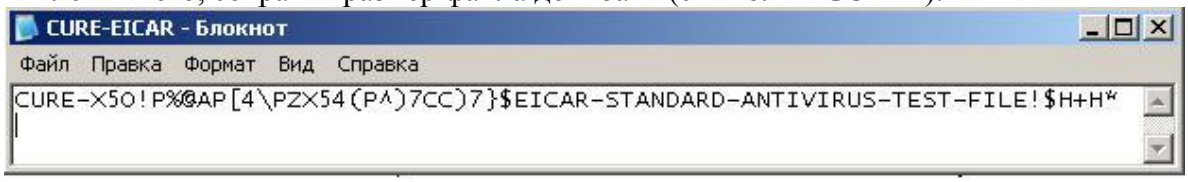
После этого следует сохранить файл с расширением .com.

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 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»).

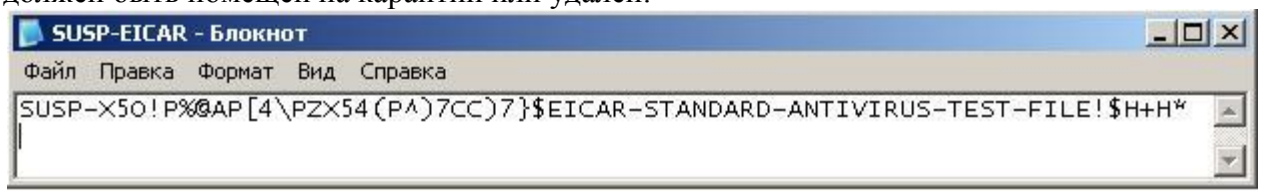


3 файл DELE-EICAR. Файл создается в текстовом редакторе путем добавления в начало вируса символов “DELE-” и сохранения файла с расширением .com. Обнаружив такой файл, антивирус определяет его как неизлечимый или троянскую программу и удаляет. По результатам проверки файл должен остаться только в резервном хранилище.

4 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.

5 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.

6 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.



7. 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.

Задание

1) Ознакомиться с антивирусной программой установленной на компьютере. С настройками.

2) Использование тестового вируса EICAR

- Создать файл EICAR

- Создать файл CURE-EICAR

- Создать файл DELE-EICAR

- Создать файл CORR-EICAR

- Создать файл ERRO-EICAR

- Создать файл SUSP-EICAR

- Создать файл WARN-EICAR

Протестировать работу антивируса на обнаружение тестовых вирусов.

3) Установить другую антивирусную программу, которую предоставил преподаватель на тестовом компьютере. Протестировать ее на обнаружение тестовых вирусов.

Сохранить отчет с результатами работы антивируса

Практическое занятие №5

Проектирование и испытание защиты от сбоев электропитания.

Теоретические сведения:

Наиболее надежным средством предотвращения потерь информациипри 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 автономнойработыограничено. 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 напряжениямогут 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 напряженияможет 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)

РезервныйИБП (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

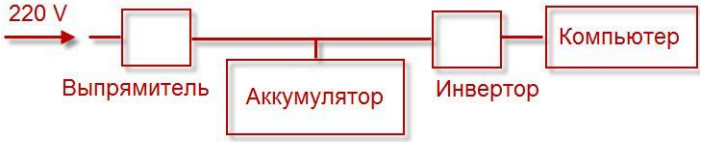
Отличается от резервного ИБП только тем, что имеет дополнительный ступенчатый стабилизатор напряжения на основе автотрансформатора, благодаря которому происходит выравнивание входящего напряжения до 220V без обращения к батарее.

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.

Вподсистему 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.

Всерверной 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.

1) Рассчитать время автономной работы оборудования серверного шкафа, если предположить, что оно подключен к ИБП 3000 ВА. Оборудование серверного шкафа состоит из двух коммутаторов, одного маршрутизатора, одного сервера. Модели оборудования взять любые.
2) Продумать система электропитания для серверной, в которой будет находиться серверный шкаф. Объяснить.

Практическое занятие №6.

Использование средств восстановления ОС.

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 восстановлениядля 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.

Задание

1) Создайте диск аварийного восстановления для ОС
2) Как можно использовать следующие инструменты для восстановления ОС:

- журналы событий; - программу просмотра сведений о системе, - программу настройки системы, -

3) Создать точку восстановления ОС:
   -автоматически
   -вручную
   -с помощью планировщика задач
4) Выполнить восстановление ОС с помощью точки восстановления, созданной с помощью планировщика задач.
5) Использование предыдущих версий файлов.

- создайте папку на рабочем столе, а в ней создайте текстовый файл 1.txt
-настройте создание теневых копий для данной папки
-удалите файл
-восстановите этот файл с помощью предыдущих версий.

Практическое занятие №7

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.

Мандатомусловимся 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.

Задача

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 20 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.

Практическое занятие №8

Защита файловых объектов.

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) идентификаторполномочий .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

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 системеУправление 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

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).

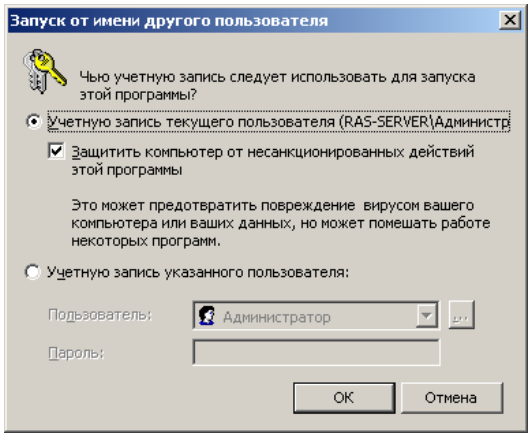


Рисунок 2. Запуск процесса с ограниченным маркером

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, инициализирующийся при выборе пункта контекстного меню **“Запуск от имени…”**;
- консольную команду **runas**:

runas /user:*имя_пользователяprogram* ,

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.

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.

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** позволяет просмотреть текущие разрешения для пользователей и групп пользователей, редактировать их, создавать новые или удалять существующие.
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»** 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.
Автоматизировать процесс определения разрешенных пользователю видов доступа к объекту можно с использованием вкладки **«Действующие разрешения»** окна дополнительных параметров безопасности объекта **.**
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* [...]]
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].
. 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.
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.

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.

Настройка списка SACL может быть осуществлена в окне дополнительных свойств объекта (**пункт "Дополнительно", закладка "Аудит"**.

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.

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**

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»**. Зашифрованные объекты в стандартном интерфейсе Windows Explorer отображаются зеленым цветом.

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*]

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.

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**.

Задание.

Запустите в программе **Oracle VMVirtualbox** виртуальную машину Windows 7. Войдите в систему под учетной записью администратора, пароль узнайте у преподавателя. Все действия выполняйте в системе, работающей на виртуальной машине.

1)

1. Создайте учетную запись нового пользователя **testUser.** 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 в этой папке несколько текстовых файлов (*.txt).
2. С помощью команды **runas** запустите сеанс командной строки (**cmd.exe**) от имени вновь созданного пользователя.
3. 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 работы.
4. Убедитесь в соответствии имени пользователя и полученного SID в реестре Windows.

5. Командой **whoami** определите перечень текущих привилегий пользователя **testUser**. В сеансе командной строки измените системное время командой **time.**
6. Убедитесь, что привилегия **«Завершение работы системы» (SeShutdownPrivilege)** предоставлена пользователю testUser. Добавьте ему привелегию **«Принудительное удаленное завершение» (SeRemoteShutdownPrivilege).**
7. Используя команду **cacls**, просмотрите разрешения на папку **c:\forTesting**. Объясните все обозначения в описаниях прав пользователей и групп в выдаче команды.
2)
1. 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.
2. 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.
3 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).
4 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.
5 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.
6 Запустите файловый менеджер от имени пользователя **testUser** и создайте в нем папку **newFolder** на диске С. 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.
7 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**).
3)Шифрование файлов и папок средствами EFS.
1. 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.
2 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.**
3 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.

4 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**).**

5 Убедитесь, что при копировании зашифрованных файлов на том с файловой системой, не поддерживающей EFS (например, FAT32 на флеш-накопителе), содержимое файла дешифруется.

Сделайте отчёт по работе.

Практическое занятие №9.
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 Windowsстандартной является файловая системаNTFS.

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.

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

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);

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.), папкаAdmin$ для доступа к системному каталогу, папкаPrint$ 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.

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.

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.

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 NTFSи разрешений для общей папки, при котором наиболее строгое разрешение имеет приоритет над другими.

Задание

1. Создайте папку TEST, в которую поместите 3 вложенные папки – test1, test2, test3. Во все папки положите по одному произвольному текстовому файлу. В папку test –файл test.txt, в папку test1 – файл test1.txt, в папку test2 – файл test2.txt, в папку test3 – файл test3.txt,
2. Установите для папки TEST разрешения полного доступа для одного из пользователей группы администраторы, и ограниченные разрешения для пользователя с ограниченной учетной записью.
3. Установите наследование разрешений для всех остальных папок, кроме папки test2.
4. В настройках для разрешений найдите “Применять эти разрешения к объектам и контейнерам только внутри этого контейнера”. Как это можно использовать? Создайте практически такую ситуацию, когда эта настройка могла бы использоваться.
5. Установите общий доступ к папке и подключитесь к ней через сеть с другого компьютера.
6. Установите разрешения общего доступа так, чтобы администратор не имел ограничений, а пользователь имел ограниченный уровень доступа.
7. Экспериментально убедитесь в правилах объединения разрешений NTFS и разрешений общего доступа. Сделайте вывод
8. Составьте отчет о проведенных экспериментах.
9. Разработайте стратегию безопасности при коллективном доступе к 2-м общим папкам для 3-х различных групп пользователей.

Практическое занятие №10

Защита трафика туннелированием SSH. Использование IPSec.

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)

Задание:

1) установить FreeSSHd - SSH сервер в Windows 7.
2) Запустить SSHd в качестве службы.
3) Создать пользователя. Установить авторизацию - NTLM.
4) Оставить брандмауэр включенным и разрешить исключение для SSH ' TCP порт 22.
5) Установить PuTTY
6) Через интерфейс PuTTY зайти в любой каталог на сервере.
   - создать текстовый файл
   - загрузить файл на локальный компьютер
7) Подключиться через PuTTY к соседу.
   - загрузить файл на удаленный компьютер
8) На выбор выполните еще 5 SSH команд.

Оформите отчет.

Практическое занятие № 11

Создание самоподписанных SSL сертификатов при помощи программы XCA.

Теоретические сведения

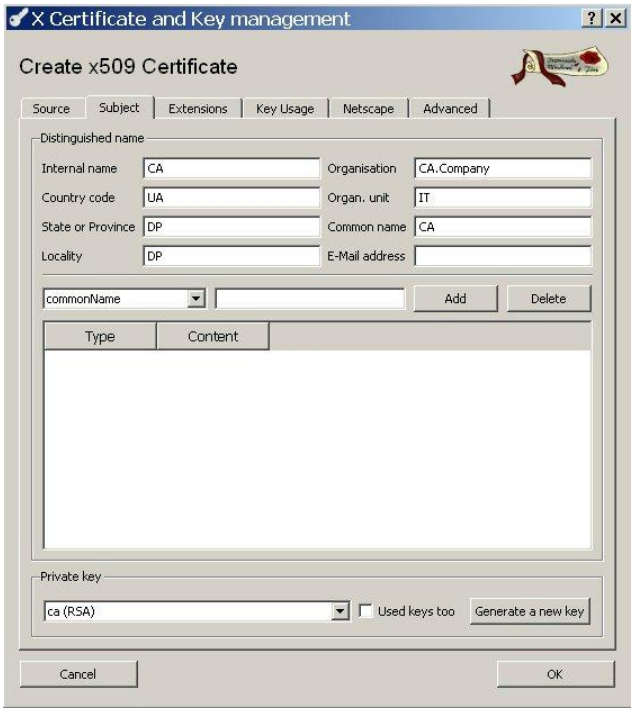
Рано или поздно ко всем администраторам сети приходит руководство, и просит сделать доступ к внутренним ресурсам сети компании через Интернет. Вот и становиться задача организации доступа.

Воспользуемся методом создания самоподписанных SSL сертификатов при помощи open-source программы **XCA.** ***Самозаверенный (самоподписанный) сертификат*** *— специальный тип сертификата, подписанный самим его создателем. Технически данный тип ничем не отличается от сертификата, заверенного подписью удостоверяющего центра (УЦ), только вместо передачи на подпись в УЦ пользователь создаёт свою собственную сигнатуру.*

Суть технологии: Мы создаем свой центр сертификации. С помощью него создаем секретные ключи и сертификаты, подписанные нашим центром сертификации. Корневой сертификат помещается на сервер. Клиентский ключ и сертификат заносится в браузер клиента. При попытке подключиться к защищенному ресурсу, происходит проверка ключевой пары. Если все хорошо, то создается защищенный канал между сервером и браузером, в котором данные «упаковываются» в криптографический протокол SSL или TLS, тем самым обеспечивая защиту этих данных.

Программа **XCA** поддерживает: • все 32-bit MS Windows (95/98/NT/2000/XP);• все BSD платформы (FreeBSD/NetBSD/OpenBSD/Apple Mac OS X);• все POSIX (Linux/BSD/UNIX-like OSes), OS X, FreeBSD, Linux.

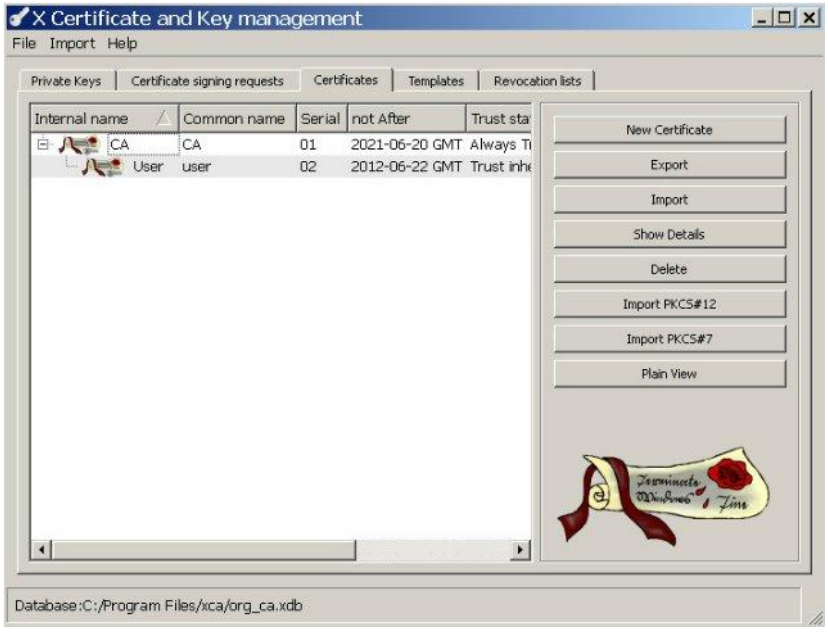
Создание корневого сертификат и ключа нашего центра сертификации: Вкладка *Private Keys-> New Keys.* Создание сертификата к нашему ключу - Вкладка *Certificates -> New certificates*, в открывшемся окне в поле *«Template for the new certificate»* выбираем *«[default] CA»*. Этим мы выбрали шаблон для создания самоподписанного корневого сертификата. Далее переходим на вкладку *Subjects*. Нам необходимо заполнить поля, которые будут занесены в корневой сертификат. Пример заполнения представлен на скриншоте



Обратите внимание на поле «Private key». В нем должно быть указано имя нашего секретного ключа, на который мы делаем корневой сертификат. Переходим на вкладку *Advanced* и нажимаем кнопку *«Validate»*, далее кнопку *«OK»*.Теперь у нас есть свой маленький центр сертификации.

Мы только что создали запрос в наш центр сертификации, который видим на вкладке *Certificate signing request*. Нажимаем на нем правой кнопкой мышки и выбираем *Sign*. В появившемся окне в

поле *Signing* выбираем *Use this Certificate for signing*. В поле ввода видим наш корневой сертификат CA, нажимаем *«OK»*. Итак, наши клиентский ключ и сертификат подписаны центром сертификации.



Задача:

1) Установить программу XCA.
2) Создать базу данных для будущего центра сертификации.
3) Создайте секретный ключ и сертификат к этому ключу.
4) Создайте ключи и сертификаты пользователя и сервера.
5) Подписать клиентский ключ и сертификат центром сертификации.

Практическое занятие № 12

Создание защищённого канала передачи данных при помощи программыStunnel.

Теоретические сведения.

Рано или поздно ко всем администраторам сети приходит руководство, и просит сделать доступ к внутренним ресурсам сети компании через Интернет. Вот и становиться задача организации доступа.

Почитав литературу, я остановился на методе создания самоподписанных SSL сертификатов и программе **Stunnel**. ***Самозаверенный (самоподписанный) сертификат*** *— специальный тип сертификата, подписанный самим его создателем. Технически данный тип ничем не отличается от сертификата, заверенного подписью удостоверяющего центра (УЦ), только вместо передачи на подпись в УЦ пользователь создаёт свою собственную сигнатуру.*

Суть технологии: Мы создаем свой центр сертификации. С помощью него создаем секретные ключи и сертификаты, подписанные нашим центром сертификации. Корневой сертификат помещается на сервер. Клиентский ключ и сертификат заносится в браузер клиента. При попытке подключиться к защищенному ресурсу, происходит проверка ключевой пары. Если все хорошо, то создается защищенный канал между сервером и браузером, в котором данные «упаковываются» в криптографический протокол SSL или TLS, тем самым обеспечивая защиту этих данных.

Сертификаты мы будем создавать при помощи open-source программы **XCA**. Программа **XCA** поддерживает: • все 32-bit MS Windows (95/98/NT/2000/XP);• все BSD платформы (FreeBSD/NetBSD/OpenBSD/Apple Mac OS X);• все POSIX (Linux/BSD/UNIX-like OSes), OS X, FreeBSD, Linux.

Настройка **Stunnel**.

```
; Certificate/key is needed in server mode and optional in client mode
; Пути к секретному ключу и сертификату сервера
cert = /usr/local/etc/stunnel/server.crt
key = /usr/local/etc/stunnel/server.pem

; Protocol version (all, SSLv2, SSLv3, TLSv1)
sslVersion = SSLv3

; Some security enhancements for UNIX systems - comment them out on Win32
chroot = /var/tmp/stunnel
setuid = stunnel
setgid = nogroup
pid = /stunnel.pid

; Some performance tunings
socket = l:TCP_NODELAY=1
socket = r:TCP_NODELAY=1

; Authentication stuff
verify = 2
; CApath is located inside chroot jail
CApath = /certs
; It's often easier to use CAfile
; Путь к корневому сертификату центра сертификации
CAfile = /usr/local/etc/stunnel/ca.crt
; Some debugging stuff useful for troubleshooting
debug = 7
output = /var/log/stunnel.log

; Настройка
[https]
accept = 443
connect = 192.168.1.1:80
```

Задача.

1) С помощью программы XCA создаем ключ и сертификат для сервера Stunnel
2) Экспортируем секретные ключи сервера и пользователя в формате *PEM*. Сертификат пользователя экспортируем в формате *«PKCS #12»*.
   При экспорте программа запросит пароль на файл.
3) Сертификат нашего центра сертификации и сервера экспортируем в формат «CRT».
4) Описание всех остальных вкладок и полей можно найти в документации к программе. В итоге у вас должно появиться:

   • *CA.crt* – Корневой сертификат центра сертификации;
   • *Server.crt* – Сертификат сервера;
   • *Server.pem* – Секретный ключ сервера;
   • *User.pkcs12* – секретный ключ + сертификат клиента.

5) Настроить **Stunnel**.
6) Импортируйте *User.pkcs12* и *CA.crt* в браузер. В итоги получите защищенный канал передачи данных между браузером и нашим ресурсом в локальной сети.

Практическое занятие № 13

Создание VPN на базе PPTP.Настройка VPN

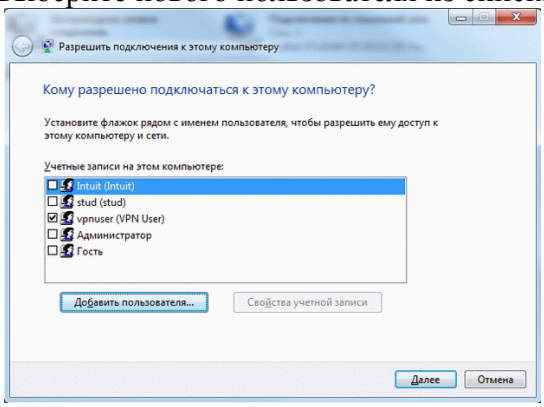
Теоретические сведения

VPN (Virtual Private Network) – это технология, которую используют для доступа к защищенной сети (сетям) посредством общедоступной сети Интернет. При помощи VPN-канала можно защитить свою информацию, зашифровав ее и передав внутри VPN-сессии. Кроме этого VPN является дешевой альтернативой дорогостоящему выделенному каналу связи.
Примечание. Windows 7 поддерживает только одно входящее подключение, и только использование

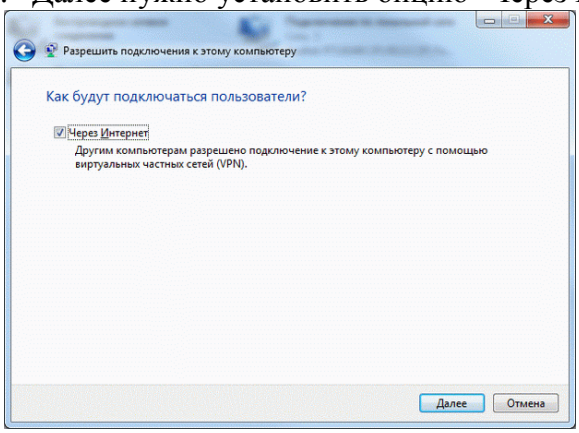
протокола PPTP.

Настройка:

1. Что бы создать VPN сервер, перейдите в "Центр управления сетями и общим доступом".
2. В открывшемся окне выберите пункт "Изменение параметров адаптера".
3. Откроется окно с сетевыми интерфейсами Вашего компьютера.
4. Нажмите кнопку "ALT " что бы стала доступна панель меню.
5. На панели выберите пункт "Файл" и подпункт "Новое входящее подключение…"
6. Откроется окно мастера настройки входящих соединений. Первым шагом необходимо или создать нового пользователя или отметить существующего (по умолчанию список учетных записей состоит из учетных записей Windows). Нажмите кнопку "Добавить пользователя..." для создания нового пользователя.
7. Введите нового пользователя и нажмите кнопку "OK".
8. Выберите нового пользователя из списка и нажмите кнопку "Далее".



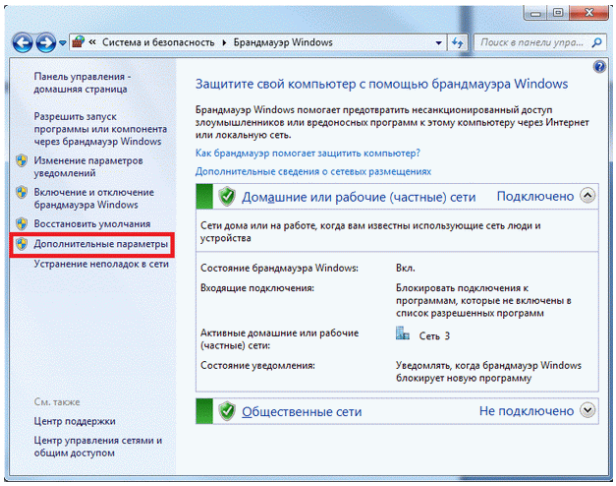
9. Далее нужно установить опцию "Через Интернет" и нажать кнопку "Далее".



10. Далее нужно настроить "Протокол Интернета версии 4". Для этого выберите соответствующий пункт меню из списка и нажмите кнопку "Свойства".
11. Откроется окно свойств входящих вызовов, где нужно будет установить опцию "Разрешить звонящим доступ к локальной сети". Так же выберите опцию "Указать IP – адреса явным образом" и установите диапазон IP –адресов, которые будут присваиваться клиентам, главное

чтобы созданная подсеть не входила в адресное пространство интернета или вашей локальной сети. Нажмите кнопку "ОК", что бы подтвердить изменения.

12. После того, как все параметры входящих подключений будут настроены. Нажмите кнопку "Разрешить доступ".
13. В результате в списке сетевых интерфейсов появится "Входящие подключения".
14. Следующим шагом необходимо добавить в список исключений, брандмауэра Windows 7, порт VPN (PPTP), для того, что бы входящие подключения клиентов не блокировались брандмауэром сервера. Для этого перейдите в "Панель управления" и выберите "Система и безопасность".
15. Выберите из списка "Брандмауэр Windows".
16. Выберите на панели слева, пункт меню "Дополнительные параметры".



17. Теперь необходимо создать новое правило для входящих подключений. Выберите на панели справа, пункт "Создать правило…"
18. Откроется мастер созданий правил. Выберите опцию "Для порта" и нажмите кнопку "Далее".
19. Следующим шагом необходимо указать протоколы и порты для входящих VPN соединений, для PPTP это "TCP 1723"
20. На следующем шаге, работы мастера, выберите опцию "Разрешить подключение" и нажмите кнопку "Далее".
21. Далее необходимо выбрать профили, для которых будет использовано правило. Существует всего три правила:
    - **Доменный.** Профиль домена применяется к сети, если для домена, в который входит локальный компьютер, обнаружен контроллер домена. При установке этого флажка правило применяется к сетевому трафику, проходящему через подключенный к этой сети сетевой адаптер.
    - **Частный.** Частный профиль применяется к сети, если она помечена администратором компьютера как частная и не является доменной сетью.
    - **Общий (Публичный).**Общий профиль применяется к сети, если компьютер непосредственно подключен к публичной сети, например в аэропорту или кафе. Параметры общего профиля должны быть самыми строгими, поскольку компьютер подключен к публичной сети, в которой безопасность нельзя контролировать так же строго, как в ограниченной информационной среде.

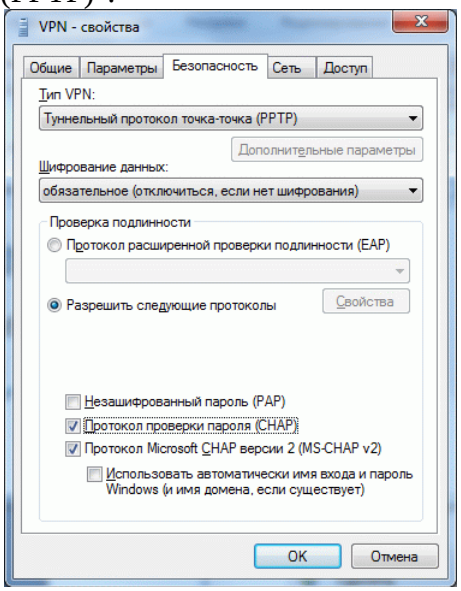
    Выберите нужные Вам профили и нажмите кнопку "Далее"
22. Введите уникальное имя правила и нажмите кнопку "Готово".

Создание VPN клиента в Windows 7

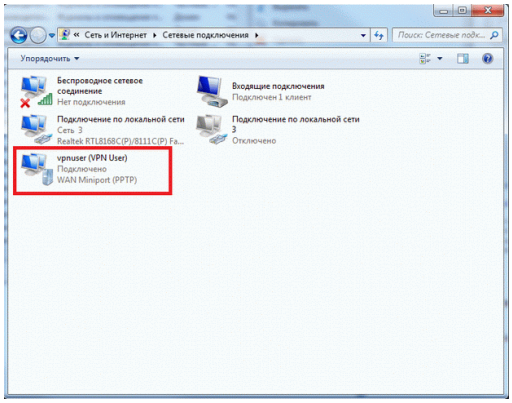
Теперь необходимо настроить на клиент-машине VPNсоединение с сервером.

1. Перейдите в "Центр управления сетями и общим доступом"
2. В открывшемся окне выберите пункт "Настройка нового подключения или сети"

3. В появившемся окне мастера, создания новых подключений, выберите пункт "Подключение к рабочему месту".
4. После выберите пункт "Использовать мое подключение к Интернету (VPN)"
5. На следующем шаге введите IP – адрес VPN сервера и имя соединения (имя местоположения), так же отметьте опцию "Не подключаться сейчас, только выполнить установку для подключения в будущем" и нажмите кнопку "Далее".
6. После чего введите имя и пароль пользователя и нажмите на кнопку "Подключить".
7. В списке сетевых интерфейсов должно появиться VPN – соединения.
8. Теперь необходимо настроить VPN подключение. Для этого щелкните правой кнопкой мыши на созданном VPN соединении и выберите пункт "Свойства".
9. Перейдите на вкладку безопасность и выберите тип VPN:"Туннельный протокол точка-точка (PPTP)".



10. Перейдите во вкладку сеть, выберите из списка "Протокол Интернета версии 4" и нажмите кнопку "Свойства".
11. В открывшемся окне щелкните на кнопку "Дополнительно".
12. Уберите флажок с пункта "Использовать основной шлюз в удаленной сети". Это необходимо для того, что бы у компьютера-клиента был доступ в сеть Internet через машину- сервера, в противном случае VPN-соединение ограничится, только ресурсами машины-сервера. Нажмите кнопку "ОК" что бы применить изменения.
13. Откройте Ваше VPN соединение и введите логин и пароль. Нажмите на кнопку "Подключение".
14. После соединения с сервером – VPN подключение появится в списке текущих подключений.
15. На машине сервере, также появится подключенный пользователь.



Задание.

1) Создайте VPN сервер в Windows 7
2) Создайте VPN клиент в Windows 7

Практическое занятие № 14

Настройка VLAN на двух коммутаторах Cisco.

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.

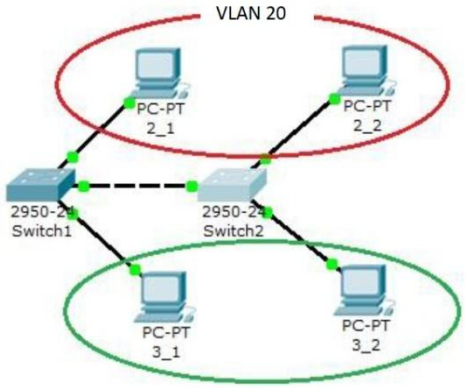
Преимущества:

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

5 - Предотвращение широковещательных штормов и предотвращение петель.

Задание

Создайте сеть, логическая топология которой представлена на рис.. Компьютеры соединены коммутатором Cisco 2950-24. В таблице приведены адреса компьютеров



VLAN 30

Компьютер	IP адрес	Коммутатор	Порт коммутатора	Вилан
2_1	10.0.0.1/8	Switch1	1	VLAN 20
2_2	10.0.0.3/8	Switch2	1	VLAN 20
3_1	10.0.0.2/8	Switch1	2	VLAN 30
3_2	10.0.0.4/8	Switch2	2	VLAN 30

1) Проверьте связность получившейся сети, пока в сети нет разделения на VLAN . Пропингуете с 2_1 все остальные компьютеры. Поскольку пока в сети нет разделения на VLAN, то все компьютеры должны быть доступны.
2) Настройте VLAN 20 и VLAN30, чтобы структурировать сети на коммутаторах.
3) Теперь организуйте магистраль обмена между коммутаторами. Для этого настройте порт на каждом коммутаторе как транковый.
4) Проверьте , что компьютеры, входящие в один виллан пингуются У вас должна появиться связь между компьютерами 2_1 и 2_2, а так же между 3_1 и 3_2. Но компьютеры в другом виллане будут недоступны.

Сохраните схему сети.

Практическое занятие № 15

Настройка 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.

Задание.

1) Создайте сеть, логическая топология которой представлена на рис

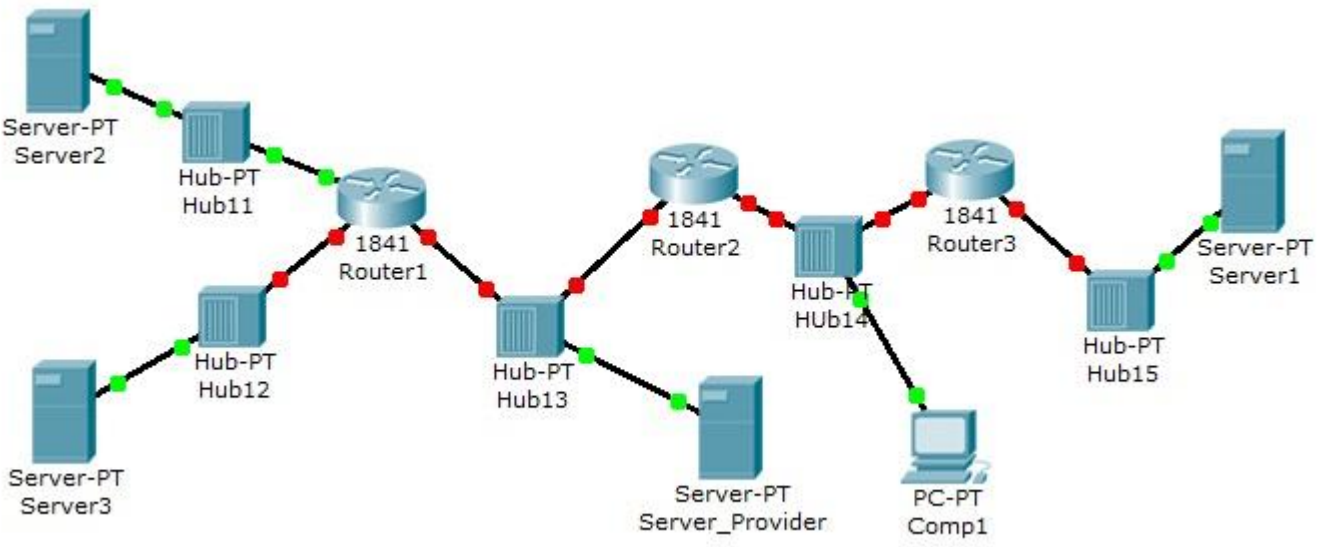


Рис. Схема сети.

Пять концентраторов представляют следующие пять сетей:

Hub11 – сеть 11.0.0.0

Hub12 – сеть 12.0.0.0

Hub13 – сеть 13.0.0.0

Hub14 – сеть 14.0.0.0

Hub15 – сеть 15.0.0.0

В сети три Web узла на Server1, Server2 и Server3.

Сервера и компьютер имеют произвольные IP адреса со шлюзами своих роутеров.
Интерфейсы роутеров определяются сетью на концентраторе и номером роутера.
Например для Router3: 15.0.0.3 и 14.0.0.3

2) Для компьютера Comp1 должны быть доступны все сервера корпоративной сети.

Практическое занятие № 16

Мониторинг сетевого трафика. Утилиты командной строки.

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 запросов.
-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.

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.

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.

Изучение других утилит

Изучение утилиты ipconfig. Утилита предназначена для просмотра параметров TCP/IP на компьютере и управления IP-адресами, полученными интерфейсом через DHCP.

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.

Изучение утилиты netstat. Netstat – утилита командной строки, отображающая информацию о текущих сетевых подключениях TCP/IP-компыотера и о трафике, генерируемом различными протоколами TCP/IP. На компьютерах с UNIX эта программа называется netstat, на компьютерах с Windows – netstat.exe.

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.

Изучение утилиты nbtstat. Nbtstat – утилита командной строки, отображающая статистику протокола и текущие сетевые подключения TCP/IP-компьютера.

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:

Утилита pathping. Трассировка маршрута с дополнительными параметрами. Изучите всевозможные параметры данной утилиты..

Задание.

1. Выполните команды ping и tracert до конкретного компьютера в сети. Что такое TTL?
2. Изучите режимы работы утилиты ipconfig, запуская ее с различными параметрами.
3. Изучите режимы работы утилиты arp, запуская ее с различными параметрами.
4. Изучите режимы работы утилиты netstat, запуская ее с различными параметрами.
5. Изучите режимы работы утилиты nslookup, запуская ее с различными параметрами.
6. Изучите режимы работы утилиты nbtstat, запуская ее с различными параметрами
7. Изучите режимы работы утилиты route, запуская ее с различными параметрами

Определите по статистике IP адрес шлюза вашей сети и основного шлюза.

8. Изучите режимы работы утилиты pathping, запуская ее с различными параметрами
9. Определите в сети текущие сетевые подключения TCP/IP – компьютера и таблицу маршрутов.

Практическое занятие № 17

Установка, настройка и использование программных сетевых анализаторов и сканеров безопасности. Анализ уязвимостей вычислительной системы.

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.

Задание.

1.Установите программу *Wireshark*

2.Запустите программу *Wireshark* и получите список всех *Ethernet*-адаптеров, а также их *mac*-адреса, включите режим прослушивания.

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

4. Осуществите вывод таблиц протокола *ARP* для всех интерфейсов/

5. Пользуясь программой сканером на примере одного из активных соединений, отследите получаемый трафик по протоколу *TCP*, а на примере одного из пакетов - вложенность протоколов. Выполните анализ информации, содержащейся в заголовках сегмента *TCP*.
6. Пользуясь программой сканером, проследите процесс установления соединения по протоколу *FTP* с любым *FTP* сервером.
7. Сделайте вывод о выполненной работе.

Практическое занятие №18

Настройка и использование программных брандмауэров, систем обнаружения вторжений

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 дефрагментацияIP-пакетов,отслеживание TCPсоединений и выявление попыток сканирования портов

Задание

1) Настройте брандмауэр Windows. Запретите доступ браузеру Internet Explorer к Интернету
2) Установите СОА Snort
3) Выведите на экран список доступных сетевых интерфейсов с помощью СОА Snort
4) Запустить Snort на выбранном интерфейсе в режиме анализатора пакетов с выводом информации на экран, указав программе завершить работу после приема третьего пакета:
5).Выполнить любые действия, которые приведут к отправке или приему сетевых пакетов (например, отправить эхо-запросы на любой IP-адрес командой ping). Убедиться, что пакеты перехватываются и отображаются на экране.
6) Получить у преподавателя задание по выявлению факта сканирования портов.