МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт сервиса, туризма и дизайна (филиал) СКФУ в г. Пятигорске

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению лабораторных работ
по дисциплине

ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ

Направление подготовки	10.03.01 Информационная безопасность
Профиль	Комплексная защита объектов информатизации
Квалификация выпускника	бакалавр
Форма обучения	очная
Учебный план	2020 г.

Пятигорск, 2020 г.

ВВЕДЕНИЕ

Методические указания содержат курс лабораторных работ по дисциплине «Программно-аппаратные средства защиты информации» направленный на изучение принципов функционирования и элементной базы вычислительных систем.

Содержащиеся в данном пособии сведения теории, методические указания и рекомендации по выполнению лабораторных работ позволяют использовать его в качестве дополнительного пособия для закрепления курса лекций.

СОДЕРЖАНИЕ

Лабораторная работа 1.

Межсетевые экраны

Цель работы: Изучение межсетевых экранов. Приобретение навыков работы с Iptables и WAF.

Компетенции:

Код	Формулировка:
ОПК- 7	способностью определять информационные ресурсы, подлежащие защите, угрозы безопасности информации и возможные пути их реализации на основе анализа структуры и содержания информационных процессов и особенностей функционирования объекта защиты;
ПК-1	способностью выполнять работы по установке, настройке и обслуживанию программных, программно – аппаратных (в том числе криптографических) и технических средств защиты информации;
ПК-2	способностью применять программные средства системного, прикладного и специального назначения, инструментальные средства, языки и системы программирования для решения профессиональных задач;
ПК-3	способностью администрировать подсистемы информационной безопасности объекта защиты;
ПК-4	способностью участвовать в работах по реализации политики информационной безопасности, применять комплексный подход к обеспечению информационной безопасности объекта защиты;

Теоретическая часть.

Межсетевой экран

Скорее всего, вы уже сталкивались с таким понятием как межсетевой экран. В ядро Linux встроен свой межсетевой экран, называемый Netfilter. Управление им осуществляется с помощью утилиты Iptables.

Межсетевой экран, сетевой экран, файервол, брандмауэр — комплекс аппаратных или программных средств, осуществляющий контроль и фильтрацию проходящих через него сетевых пакетов в соответствии с заданными правилами. Основной задачей сетевого экрана является защита компьютерных сетей или отдельных узлов от несанкционированного доступа. Также сетевые экраны часто называют фильтрами, так как их основная задача — не пропускать (фильтровать) пакеты, не подходящие под критерии, определённые в конфигурации.

Рассмотрим принцип работы Netfilter. Когда сетевые пакеты попадают в сетевой интерфейс, они после ряда проверок ядром проходят последовательность так называемых цепочек. Пакет обязательно проходит через цепочку PREROUTING, после чего определяется, кому он, собственно, был адресован. Если пакет не адресован локальной системе (в нашем случае серверу), он попадает в цепочка FORWARD, а иначе — в цепочку INPUT, после прохождения которой отдается локальным демонам или процессам. После этого при необходимости формируется ответ, который направляется в цепочку OUTPUT. После цепочек OUTPUT или FORWARD пакет в очередной раз встречается с правилами маршрутизации и направляется в цепочку POSTROUTING. В результате прохождения пакетом цепочек фильтрации несколько раз, проверка его принадлежности определенным критериям осуществляется несколько раз. В соответствии с этими проверками к пакету применяется определенное действие:

- ACCEPT — пакет «принимается» и передается в следующую цепочку.

- DROP — удовлетворяющий условию пакет отбрасывается и не передается в другие таблицы или цепочки.
- REJECT — пакет отбрасывается, но при этом отправителю отправляется ICMP-сообщение, сообщающее об отказе.
- RETURN — пакет возвращается в предыдущую цепочку и продолжает её прохождение начиная со следующего правила
- SNAT — применить трансляцию источника в пакете. Используется только в цепочках POSTROUTING и OUTPUT таблицы nat.
- DNAT — применить трансляцию адреса назначения в пакете. Используется в цепочках PREROUTING и (очень редко) OUTPUT в таблице 1.

Таблица 1 -Основные команды Iptables

Параметр	Описание	Пример
–destination(-d)	IP адрес назначения пакета. Может быть определен несколькими путями (см. –source).	iptables -A INPUT –destination 192.168.1.0/24
–in-interface(-i)	Определяет интерфейс, на который прибыл пакет. Полезно для NAT и машин с несколькими сетевыми интерфейсами. Применяется в цепочках INPUT,FORWARD и PREROUTING. Возможно использование знака +, тогда подразумевается использование всех интерфейсов, начинающихся на имя+ (например eth+ - все интерфейсы eth).	iptables -t nat -A PREROUTING –in-interface eth0
–out-interface(-o)	Определяет интерфейс, с которого уйдет пакет. Полезно для NAT и машин с несколькими сетевыми интерфейсами. Применяется в цепочках OUTPUT, FORWARD и POSTROUTING. Возможно использование знака +.	iptables -t nat -A POSTROUTING –in-interface eth1
	Неявные (необщие) параметры	
-p proto -h	Вывод справки по неявным параметрам протокола proto.	iptables -p icmp -h
–source-port(–sport)	Порт источник, возможно только для протоколов –protocol tcp, или –protocol udp	iptables -A INPUT –protocol tcp –source-port 25
–destination-port(–dport)	Порт назначения, возможно только для протоколов –protocol tcp, или –protemocol udp	iptables -A INPUT –protocol udp –destination-port 67
	Явные параметры	
-m state –state (устарел)	Состояние соединения. Доступные опции: NEW (Все пакеты устанавливающие новое соединение) ESTABLISHED (Все пакеты, принадлежащие установленному соединению) RELATED (Пакеты, не принадлежащие установленному	iptables -A INPUT -m state –state NEW, ESTABLISHED iptables -A INPUT -

Параметр	Описание	Пример
он же -m conntrack –ctstate	соединению, но связанные с ним. Например - FTP в активном режиме использует разные соединения для передачи данных. Эти соединения связаны.) INVALID (Пакеты, которые не могут быть по тем или иным причинам идентифицированы).	m conntrack –ctstate NEW, ESTABLISHED
-m mac –mac-source	Задает MAC адрес сетевого узла, передавшего пакет. MAC адрес должен указываться в форме XX:XX:XX:XX:XX:XX.	-m mac –mac-source 00:00:00:00:00:0
	Дополнительные параметры	
	DNAT (Destination Network Address Translation)	
–to-destination	Указывает, какой IP адрес должен быть подставлен в качестве адреса места назначения. В примере во всех пакетах протокола tcp, пришедших на адрес 1.2.3.4, данный адрес будет заменен на 4.3.2.1.	iptables -t nat -A PREROUTING -p tcp -d 1.2.3.4 -j DNAT –to-destination 4.3.2.1
	LOG	
–log-level	Используется для задания уровня журналирования (log level). В примере установлен максимальный уровень логирования для всех tcp пакетов в таблице filter цепочки FORWARD.	iptables -A FORWARD -p tcp -j LOG –log-level debug
–log-prefix	Задает текст (префикс), которым будут предваряться все сообщения iptables. Префикс может содержать до 29 символов, включая и пробелы. В примере отправляются в syslog все tcp пакеты в таблице filter цепочки INPUT с префиксом INRUT-filter.	iptables -A INPUT -p tcp -j LOG –log-prefix INRUT-filter
–log-ip-options	Позволяет заносить в системный журнал различные сведения из заголовка IP пакета.	iptables -A FORWARD -p tcp -j

ежсетевого экрана Netfilter:
чальная обработка входящих пакетов
их пакетов, адресованных непосредственно локальному компьютеру
ршрутизируемых пакетов
ов, исходящих с локального компьютера
окончательной обработки исходящих пакетов Таблицы межсетевого экрана Netfilter:
ля маркировки пакетов, которые не должны обрабатываться системой определения состояний. Содержится в цепоч
правила модификации IP-пакетов.

• nat - предназначена для подмены адреса отправителя или получателя. Данную таблицу проходят только первые пакеты из потока - трансляция адресов или маскировка (подмена адреса отправителя или получателя) применяются ко всем последующим пакетам в потоке автоматически. Поддерживает действия DNAT, SNAT, MASQUERADE, REDIRECT. Содержится в цепочках PREROUTING, OUTPUT, и POSTROUTING.

• filter — основная таблица, используется по умолчанию если название таблицы не указано. Используется для фильтрации пакетов. Содержится в цепочках INPUT, FORWARD, и OUTPUT.

Пример создания правила для межсетевого экрана

Рассмотрим две цепочки, задающие два основных правила Iptables — PREROUTING и FORWARD.

• iptables -t nat -A PREROUTING -i eth0 -j DNAT —to-destination 192.168.57.102

• iptables -A FORWARD -d 192.168.57.102 -j ACCEPT

Первая из них определяет первоначальную обработку всех пакетов, приходящих на адаптер eth0:

• -t определяет подключаемую таблицу, в данном случае — nat — для подмены адреса отправителя или получателя

• -A — выбор цепочки

• -i — входящий интерфейс

• -j — действие с пакетами, удовлетворяющими условию — в данном случае DNAT — подмена адреса получателя

• –to-destination — выбор адреса, на оторый перенаправляются пакеты

• Вторая определяет проброс пакетов через сервер:

• -A — выбор цепочки

• -d — выбор адресата

• -j — выбор действия

Web Application Firewall

WAF (Web Application Firewall) - это межсетевые экраны, работающие на прикладном уровне и осуществляющие фильтрацию трафика Web-приложений. Эти средства не требуют изменений в исходном коде Web-приложения и, как правило, защищают Web-сервисы гораздо лучше обычных межсетевых экранов и средств обнаружения вторжений.

Основные преимущества:

• Анализ поведения пользователя в используемом приложении;

• Позволяет осуществлять мониторинг HTTP трафика и проводить анализ событий в реальном режиме времени;

• Предотвращение вредоносных запросов;

• Распознавание большинства опасных угроз;

• Дополнение сетевых средств безопасности;

• Просматривать детальные отчеты об атаках и попытках взлома.

Оборудование и материалы

Для выполнения лабораторной работы предусмотрены компьютерные классы, находящиеся в аудиториях, оснащенных ПК, а также системное программное обеспечение – ОС MS Windows 7 и приложения Office (Excel Word, Access и т.п.).

Указания по технике безопасности

Лабораторная работа проводится на ПЭВМ. Запрещается прикасаться к задней панели системного блока при включенном питании, переключать разъемы интерфейсных кабелей периферийных устройств, загромождать верхние панели устройств бумагами и

посторонними предметами, допускать попадание влаги на поверхность системного блока, монитора, клавиатуры и других устройств.

Порядок выполнения работы

Задание 1.

- Установите web-сервер <sudo apt-get install apache2>
- Просмотрите список текущих правил iptables таблицы filter

sudo iptables -L

- Вы увидите, что список содержит три цепочки по умолчанию (INPUT, OUTPUT и FORWARD), в каждой из которых установлена политика по умолчанию (на данный момент это ACCEPT).

• С помощью команды <sudo iptables -S> данный список можно просмотреть в другом формате, который отражает команды, необходимые для активации правил и политик.

• Чтобы сбросить текущие правила (если таковые есть),

наберите: sudo iptables -F

• Цепочка INPUT отвечает за входящий траффик.

• Чтобы внести локальный интерфейс выполните:

sudo iptables -A INPUT -i lo -j ACCEPT

• Чтобы заблокировать весь исходящий трафик, кроме портов для SSH и веб-сервера, нужно сначала разрешить подключения к этим портам. В цепочку ACCEPT добавьте два порта (порт SSH 22 и порт http 80), что разрешит трафик на эти порты.

sudo iptables -A INPUT -p tcp -m tcp –dport 22 -j ACCEPT

sudo iptables -A INPUT -p tcp -m tcp –dport 80 -j ACCEPT

• В данной работе мы не используем SSH. Так что удалим ненужное правило. Для этого:

sudo iptables -D INPUT -p tcp -m tcp –dport 22 -j ACCEPT

• Нужно добавить еще одно правило, которое позволит устанавливать исходящие соединения (т.е. использовать ping или запускать обновления программного обеспечения):

sudo iptables -I INPUT -m state –state ESTABLISHED,RELATED -j ACCEPT

• Создав все эти правила, можно заблокировать все остальное и разрешить все исходящие соединения.

sudo iptables -P OUTPUT ACCEPT

sudo iptables -P INPUT DROP

• Просмотрите список правил

sudo iptables -L

• Добавим еще несколько правил для блокировки наиболее распространенных атак. Для начала нужно заблокировать нулевые пакеты <sudo iptables -A INPUT -p tcp –tcp-flags ALL NONE -j DROP>.

• Следующее правило отражает атаки syn-flood <sudo iptables -A INPUT -p tcp ! –syn -m state –state NEW -j DROP>. Теперь фаервол не будет принимать входящих пакетов с tcp-флагами. Нулевые пакеты, по сути, разведывательные. они используются, чтобы выяснить настройки сервера и определить его слабые места.

• Далее нужно защитить сервер от разведывательных пакетов XMAS <sudo iptables -A INPUT -p tcp –tcp-flags ALL ALL -j DROP>. Теперь сервер защищен от некоторых общих атак, которые ищут его уязвимости.

• Со второй виртуальной машины, на которую установите nmap, проведите XMAS сканирование <sudo nmap -sX>.

• По умолчанию все несохраненные правила действуют до следующей перезагрузки сервера; сразу же после перезагрузки несохраненные правила будут потеряны. Самый простой способ загрузить пакет iptables-persistent <sudo apt-get install iptables-persistent>. Во время инсталляции пакет уточнит, нужно ли сохранить текущие правила для

дальнейшей автоматической загрузки, если текущие правила были протестированы и соответствуют всем требованиям, их можно сохранить.

Задание 2.

- Для начала понадобится LAMP(Apache, MySQL, PHP). В лабораторной работе № 8, уже было показано, как установить его, используя tasksel.
- Установите mod_security <sudo apt-get install libapache2-mod-security2>
- Выполните команду <sudo apachectl -M | grep –color security2>. Если на экране появился модуль по имени security2_module (shared), значит, все прошло успешно.
- В каталоге логов Apache можно найти новый лог-файл для mod_security. /var/log/apache2/modsec_audit.log
- Установка ModSecurity включает в себя конфигурационный файл, который нужно переименовать: <sudo mv /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity /modsecurity.conf>.
- Затем перезапустите Apache <sudo service apache2 reload>.
- Стандартный конфигурационный файл настроен на DetectionOnly, то есть, фаервол только отслеживает логи, при этом ничего не блокируя. Чтобы изменить это поведение, отредактируйте файл modsecurity.conf: <sudo nano /etc/modsecurity/modsecurity.conf>
- Найдите в файле строку: “SecRuleEngine DetectionOnly”. И измените ее так: “SecRuleEngine On”.
- Найдите “SecResponseBodyAccess On” и замените на “SecResponseBodyAccess Off”. Эта директива отвечает за буферизацию тела ответа; ее рекомендуется включать, только если требуется обнаружение и предохранение от утечки данных. Включенная директива (SecResponseBodyAccess On) не только будет использовать больше ресурсов сервера, но и увеличит размер лог-файла, следовательно, ее желательно отключить.
- По умолчанию mod_security поставляется с базовым набором правил CRS (Core Rule Set), которые находятся в /usr/share/modsecurity-crs/
- Чтобы подгрузить эти готовые правила, нужно, чтобы веб-сервер Apache читал указанные выше каталоги. Для этого отредактируйте файл mod-security.conf:

nano /etc/apache2/mods-enabled/mod-security.conf

- Между <IfModule security2_module> </IfModule> внесите следующие параметры:

```
Include "/usr/share/modsecurity-crs/activated_rules/*.conf"
```

- Директория activated_rules аналогична директории Apache mods-enabled. Правила доступны в каталогах: /usr/share/modsecurity-crs/base_rules ; /usr/share/modsecurity-crs/optional_rules ; /usr/share/modsecurity-crs/experimental_rules
- Чтобы активировать правила, нужно создавать символические ссылки в каталоге activated_rules. <cd /usr/share/modsecurity-crs/activated_rules/>
- Добавьте несколько правил, например <sudo ln -s /usr/share/modsecurity-crs/base_rules/modsecurity_crs_30_http_policy.conf> ; <sudo ln -s /usr/share/modsecurity-crs/base_rules/modsecurity_crs_49_generic_attacks.conf>
- Чтобы новые правила вступили в исполнение, нужно перезапустить Apache <sudo service apache2 reload>

Содержание отчета

1. Тема
2. Цель работы
3. Краткое описание выполненной работы.
4. Продемонстрировать данную работу на ПК, в соответствии с заданиями, оформить в программной оболочке Microsoft Word, **в**ключив в него копии экрана.
5. Сформулировать заключение и выводы
6. Ответить на контрольные вопросы.

Контрольные вопросы

1. Что такое межсетевой экран?
2. Для чего используется межсетевой экран? 3.Принцип работы Netfilter.
4. Таблицы межсетевого экрана Netfilter. Для чего они используются?
5. Что такое правила межсетевого экрана?
6. Как создавать правила для межсетевого экрана утилитой Iptables?
7. Как сохранить правила для последующей автозагрузки?
8. Что такое Web Application Firewall?
9. Как настроить правила в WAF mod_security?

Основная литература

1. Царев, Р.Ю. Программные и аппаратные средства информатики : учебник / Р.Ю. Царев, А.В. Прокопенко, А.Н. Князьков ; Министерство образования и науки Российской Федерации, Сибирский Федеральный университет. - Красноярск : Сибирский федеральный университет, 2015. - 160 с. : табл., схем., ил. - Библиогр. в кн. - ISBN 978-5-7638-3187-0 ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=435670

2. Лабораторный практикум по дисциплине Программно-аппаратные средства защиты информации [Электронный ресурс] / . — Электрон. Текстовые данные. — М. : Московский технический университет связи и информатики, 2016. — 31 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/61529.html

Дополнительная литература

1. Айдинян, А.Р. Аппаратные средства вычислительной техники : учебник / А.Р. Айдинян. - М. ; Берлин : Директ-Медиа, 2016. - 125 с. : ил., схем., табл. - Библиогр. в кн. - ISBN 978-5-4475-8443-6 ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=443412

2. Привалов, И. М. (Институт сервиса, туризма и дизайна (филиал)СКФУ в г. Пятигорске). Основы аппаратного и программного обеспечения : учеб.-метод. пособие / И.М. Привалов ; Сев.-Кав. федер. ун-т. - Ставрополь : СКФУ, 2015. - 145 с. - 144 с.

Интернет-ресурсы:

1. http://www.biblioclub.ru/ - электронная библиотека
2.http://www.uts-edu.ru/ - «Электронные курсы»

Лабораторная работа 2.
Программное восстановление данных.

Цель работы: Получение теоретических и практических навыков программного восстановления данных.

Компетенции:

Код	Формулировка:
ОПК- 7	способностью определять информационные ресурсы, подлежащие защите, угрозы безопасности информации и возможные пути их реализации на основе анализа структуры и содержания информационных процессов и особенностей функционирования объекта защиты;
ПК-1	способностью выполнять работы по установке, настройке и обслуживанию программных, программно – аппаратных (в том числе криптографических) и технических средств защиты информации;
ПК-2	способностью применять программные средства системного, прикладного и специального назначения, инструментальные средства, языки и системы программирования для решения профессиональных задач;
ПК-3	способностью администрировать подсистемы информационной безопасности объекта защиты;
ПК-4	способностью участвовать в работах по реализации политики информационной безопасности, применять комплексный подход к обеспечению информационной безопасности объекта защиты;

Теоретическая часть

1. Восстановление данных TestDisk

TestDisk — свободная программа для восстановления данных, предназначенная прежде всего для восстановления потерянных разделов на носителях информации, а также для восстановления загрузочного сектора, после программных или человеческих ошибок (например, потеря MBR).

- Установка **<sudo apt-get install testdisk>**.
- Запускаем TestDisk **<sudo testdisk>**.
- Появляется окошко приветствия TestDisk, нам предлагается вести лог работы (для выполнения данной работы лог не требуется).
- Выбираем нужный диск и нажимаем **Enter**.
- Предлагается выбрать тип таблицы разделов, обычно TestDisk определяет все правильно, так что нажимаем **Enter**.
- Выбираем **Analise**.
- Выбираем **QuickSearch**.
- Нам выводят таблицу разделов. Выбираем раздел и нажимаем **P**, чтобы вывести список файлов.
- Выбираем файлы для восстановления и нажимаем **C**.
- Выбираем папку, куда будут сохранены файлы и нажимаем **C**.

2. Восстановление данных PhotoRec

PhotoRec - это утилита, входящая в состав пакета TestDisk. Предназначена для восстановления испорченных файлов с карт памяти цифровых фотоаппаратов (CompactFlash, Secure Digital, SmartMedia, Memory Stick, Microdrive, MMC), USB flash-

дисков, жестких дисков и CD/DVD. Восстанавливает файлы большинства распространенных графических форматов, включая JPEG, аудио-файлы, включая MP3, файлы документов в форматах Microsoft Office, PDF и HTML, а также архивы, включая ZIP. Может работать с файловыми системами ext2, ext3, ext4 FAT, NTFS и HFS+, причем способна восстановить графические файлы даже в том случае, когда файловая система повреждена или отформатирована.

- Установка **<sudo apt-get install testdisk>**.
- Запускаем PhotoRec **<sudo photorec>**.
- Выбираем нужный диск и нажимаем **Enter**.
- В нижнем меню можно выбрать **File Opt**, чтобы выбрать типы файлов для восстановления (по умолчанию выбраны все).
- Чтобы начать восстановление нажмите **Enter**, выбрав **Search**.
- У нас выбрана система ext4, поэтому выбираем первый вариант [ext2/ext3].
- Если выбрать пункт **FREE**, то поиск будет произведен в пустом пространстве и в этом случае будут восстановлены только удаленные файлы, а если выбрать **WHOLE**, то поиск будет произведен на всем диске.
- Теперь нужно указать директорию, куда будем сохранять нужные нам файлы. Выбираем нужную папку и нажимаем **C**.
- Выбираем файлы для восстановления и нажимаем **C**.

3. Восстановление данных Extundelete

Extundelete – утилита, позволяющая восстанавливать файлы, которые были удалены с разделов ext3/ext4.

- Установка: **<sudo apt-get install extundelete>**.
- Как только вы поняли, что удалили нужные файлы, необходимо отмонтировать раздел:**<umount /dev/<partition> >**
- Зайдите в каталог, в который будут восстанавливаться удаленные данные. Он должен быть расположен на разделе отличном от того, на котором хранились восстанавливаемые данные: **cd /<путь_к_каталогу_куда_восстанавливать_данные>**
- Запустите **extundelete**, указав раздел, с которого будет происходить восстановление и файл, который необходимо восстановить: **sudo extundelete /dev/<partition> –restore-file /<путь к файлу>/<имя_файла>**
- Можно так же восстанавливать содержимое каталогов: **sudo extundelete /dev/<partition> –restore-directory /<путь_к_директории>**

4.Восстановление данных Foremost.

Forem

экстрагирует их из диска/образа и складывает в каталог, вместе с подробным отчётом о том, чего
/etc/foremost.conf), о которых программа не знает.

Установка: **<sudo apt-get install foremost>**

Пример использования для восстановления изображений с диска **/dev/sdb** в каталог**~/out_dir**: **<s**

Оборудование и материалы

Для выполнения лабораторной работы предусмотрены компьютерные классы, находящиеся в аудиториях, оснащенных ПК, а также системное программное обеспечение – ОС MS Windows 7 и приложения Office (Excel Word, Access и т.п.).

Указания по технике безопасности

Лабораторная работа проводится на ПЭВМ. Запрещается прикасаться к задней панели системного блока при включенном питании, переключать разъемы интерфейсных кабелей периферийных устройств, загромождать верхние панели устройств бумагами и посторонними предметами, допускать попадание влаги на поверхность системного блока, монитора, клавиатуры и других устройств.

Порядок выполнения работы

Задание 1.

- Добавьте в виртуальную машину виртуальный жесткий диск.
- Запустите виртуальную машину с Linux.
- Запустите fdisk (gdisk или parted) и создайте таблицу разделов MBR с разделами.
- Отформатируйте созданные разделы в файловую систему ext4.
- Установите TestDisk.
- Удалите MBR (или таблицу разделов) с помощью команды DD.
- Восстановите MBR (или таблицу разделов) с помощью TestDisk.
- Смонтируйте восстановленные разделы и создайте там произвольные файлы.
- Удалите созданные файлы.

Задание 2.

- С помощью TestDisk восстановите данные.
- Создайте произвольный каталог и запишите туда данные каталога /var/log/ .
- Удалите данные с созданного каталога.
- С помощью PhotoRec восстановите данные.
- Создайте произвольный каталог и запишите туда данные каталога /etc/ .
- С помощью Extundelete или Foremost восстановите данные.

Содержание отчета

1. Тема
2. Цель работы
3. Краткое описание выполненной работы.
4. Продемонстрировать данную работу на ПК, в соответствии с заданиями, оформить в программной оболочке Microsoft Word, **в**ключив в него копии экрана.
5. Сформулировать заключение и выводы
6. Ответить на контрольные вопросы.

Контрольные вопросы

1. С помощью какой из программ, используемых в этой лабораторной работе, можно восстановить таблицу разделов?
2. Какие файловые системы поддерживает PhotoRec?
3. Какие форматы поддерживает PhotoRec?
4. Как Foremost восстанавливает файлы?
5. Можно ли восстановить данные с файловой системы NTFS, используя extundelete?
6. Все ли данные скопированные с каталога /var/log/ восстановились?
7. Все ли данные скопированные с каталога /etc/ восстановились?

Основная литература

1. Царев, Р.Ю. Программные и аппаратные средства информатики : учебник / Р.Ю. Царев, А.В. Прокопенко, А.Н. Князьков ; Министерство образования и науки Российской Федерации, Сибирский Федеральный университет. - Красноярск : Сибирский федеральный университет, 2015. - 160 с. : табл., схем., ил. - Библиогр. в кн. - ISBN 978-5-7638-3187-0 ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=435670

2. Лабораторный практикум по дисциплине Программно-аппаратные средства защиты информации [Электронный ресурс] / . — Электрон. Текстовые данные. — М. : Московский технический университет связи и информатики, 2016. — 31 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/61529.html

Дополнительная литература

1. Айдинян, А.Р. Аппаратные средства вычислительной техники : учебник / А.Р. Айдинян. - М. ; Берлин : Директ-Медиа, 2016. - 125 с. : ил., схем., табл. - Библиогр. в кн. - ISBN 978-5-4475-8443-6 ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=443412

2. Привалов, И. М. (Институт сервиса, туризма и дизайна (филиал)СКФУ в г. Пятигорске). Основы аппаратного и программного обеспечения : учеб.-метод. пособие / И.М. Привалов ; Сев.-Кав. федер. ун-т. - Ставрополь : СКФУ, 2015. - 145 с. - 144 с.

Интернет-ресурсы:

1. http://www.biblioclub.ru/ - электронная библиотека
2.http://www.uts-edu.ru/ - «Электронные курсы»

Лабораторная работа 3.
Обнаружение и предотвращение вторжений.

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

Компетенции:

Код	Формулировка:
ОПК- 7	способностью определять информационные ресурсы, подлежащие защите, угрозы безопасности информации и возможные пути их реализации на основе анализа структуры и содержания информационных процессов и особенностей функционирования объекта защиты;
ПК-1	способностью выполнять работы по установке, настройке и обслуживанию программных, программно – аппаратных (в том числе криптографических) и технических средств защиты информации;
ПК-2	способностью применять программные средства системного, прикладного и специального назначения, инструментальные средства, языки и системы программирования для решения профессиональных задач;
ПК-3	способностью администрировать подсистемы информационной безопасности объекта защиты;
ПК-4	способностью участвовать в работах по реализации политики информационной безопасности, применять комплексный подход к обеспечению информационной безопасности объекта защиты;

Теоретическая часть

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.

Snort использует правила, написанные простым , но в то же время гибким и достаточно мощным языком. Существует ряд общих принципов написания, запомнить которые достаточно просто.

Большая часть правил Snort умещается в 1 строку. Это следствие того, что до версии 1.8 нельзя было использовать многострочные записи. В более поздних версиях правила можно растягивать на несколько строк, вставляя в конец строки символ “” (без кавычек).

Правила Snort состоят из двух частей: заголовка правила и параметров правила. Заголовок содержит описание действия, протокол передачи данных, IP-адреса, сетевые маски и порты источника и назначения. Параметры правила хранят предупреждающее сообщение, а также информацию о том, какую часть обнаруженного пакета нужно обработать в случае срабатывания правила.

Оборудование и материалы

Для выполнения лабораторной работы предусмотрены компьютерные классы, находящиеся в аудиториях, оснащенных ПК, а также системное программное обеспечение – ОС MS Windows 7 и приложения Office (Excel Word, Access и т.п.).

Указания по технике безопасности

Лабораторная работа проводится на ПЭВМ. Запрещается прикасаться к задней панели системного блока при включенном питании, переключать разъемы интерфейсных кабелей периферийных устройств, загромождать верхние панели устройств бумагами и посторонними предметами, допускать попадание влаги на поверхность системного блока, монитора, клавиатуры и других устройств.

Порядок выполнения работы

Задание 1.

- Узнайте свой ip адрес командой ifconfig
- Установите SNORT <sudo apt-get install snort>
- При установке будет нужно указать защищаемую сеть. Введите ..*.0/24 (Где ..* - первые три числа вашего ip-адреса, например этот будет 192.168.1.0/24, если вы используете VirtualBox и у вас в настройках сети стоит сетевой мост)
- Запустите SNORT <sudo service snort start>
- Настройка правил
- Перейдите в каталог /etc/snort/rules < cd /etc/snort/rules)
- Создайте файл с правилами <nano test.rules>

alert tcp any any -> any any (content:”https://www.google.ru/” ; msg:”Someone open Google website” ; sid: 12312313;)

- Перейдите в каталог /etc/snort <cd /etc/snort)
- Теперь нужно изменить содержимое конфигурационного файла Snort < sudo nano snort.conf>
- Найдите строчки с правилами (они начинаются с include $RULE_PATH, это в части Step 7) и добавьте файл с нашими правилами

include $RULE_PATH/test.tules

- В файле snort.conf так же укажите домашнюю сеть. В Step 1 измените строчку “ipvar HOME_NET any” , на

ipvar HOME_NET 192.168.1.0/24

- Запустите snort <sudo snort -A console -i eth0 -c snort.conf>
- Зайдите на https://www.google.ru/ и проверьте в терминале, как работает правило.
- Теперь нам понадобиться еще одна виртуальная машина, на ней должен быть установлен nmap.

Задание 2.

- Со второй ВМ используйте ping, посмотрите, как реагирует SNORT
- Используйте различные методы сканирования nmap(используйте -sS, -sT, -sN, -sU, -sX, -sF и посмотрите, как реагирует SNORT;
- В файл test.rules добавьте правило обнаружения сканирования nmap -sN (NULL Scan

) alert tcp any any -> any any (msg:”NULL Scan”; flags: 0; sid:322222;)

- Запустите snort <sudo snort -A console -i eth0 -c snort.conf>
- Со второй виртуальной машины произведите NULL сканирование <sudo nmap -sN>, проверьте, как работает правило.
- Можно загрузить обновленные правила SNORT, для этого:
- Зарегистрируйтесь на сайте https://www.snort.org/ и скачайте последнюю версию правил
- Разархивируйте скачанный архив и скопируйте каталоги rules, so_rules и preproc_rules в /etc/snort :

sudo cp -R ./rules/ /etc/snort/

sudo cp -R ./so_rules/ /etc/snort/

sudo cp -R ./preproc_rules/ /etc/snort/

Содержание отчета

1. Тема
2. Цель работы
3. Краткое описание выполненной работы.
4. Продемонстрировать данную работу на ПК, в соответствии с заданиями, оформить в программной оболочке Microsoft Word, **в**ключив в него копии экрана.
5. Сформулировать заключение и выводы
6. Ответить на контрольные вопросы.

Контрольные вопросы

1. Что такое IDS?
2. Что такое сетевая система обнаружения вторжений?
3. Чем отличаются пассивные и активные IDS?
4. Что такое SNORT?
5. Какие задачи выполняет SNORT? 6.Как работают правила SNORT?
7. Как писать правила для SNORT?
8. Зачем писать собственные правила SNORT?
9. Зачем загружать обновление правил SNORT?

Основная литература

1. Царев, Р.Ю. Программные и аппаратные средства информатики : учебник / Р.Ю. Царев, А.В. Прокопенко, А.Н. Князьков ; Министерство образования и науки Российской Федерации, Сибирский Федеральный университет. - Красноярск : Сибирский

федеральный университет, 2015. - 160 с. : табл., схем., ил. - Библиогр. в кн. - ISBN 978-5- 7638-3187-0 ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=435670

2. Лабораторный практикум по дисциплине Программно-аппаратные средства защиты информации [Электронный ресурс] / . — Электрон. Текстовые данные. — М. : Московский технический университет связи и информатики, 2016. — 31 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/61529.html

Дополнительная литература

1. Айдинян, А.Р. Аппаратные средства вычислительной техники : учебник / А.Р. Айдинян. - М. ; Берлин : Директ-Медиа, 2016. - 125 с. : ил., схем., табл. - Библиогр. в кн. - ISBN 978-5-4475-8443-6 ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=443412

2. Привалов, И. М. (Институт сервиса, туризма и дизайна (филиал)СКФУ в г. Пятигорске). Основы аппаратного и программного обеспечения : учеб.-метод. пособие / И.М. Привалов ; Сев.-Кав. федер. ун-т. - Ставрополь : СКФУ, 2015. - 145 с. - 144 с.

Интернет-ресурсы:

1. http://www.biblioclub.ru/ - электронная библиотека
2.http://www.uts-edu.ru/ - «Электронные курсы»

Лабораторная работа 4.
Электронная цифровая подпись.

Цель работы:

Ознакомиться со схемами цифровой подписи и получить навыки создания и проверки подлинности ЦП.

Компетенции:

Код	Формулировка:
ОПК- 7	способностью определять информационные ресурсы, подлежащие защите, угрозы безопасности информации и возможные пути их реализации на основе анализа структуры и содержания информационных процессов и особенностей функционирования объекта защиты;
ПК-1	способностью выполнять работы по установке, настройке и обслуживанию программных, программно – аппаратных (в том числе криптографических) и технических средств защиты информации;
ПК-2	способностью применять программные средства системного, прикладного и специального назначения, инструментальные средства, языки и системы программирования для решения профессиональных задач;
ПК-3	способностью администрировать подсистемы информационной безопасности объекта защиты;
ПК-4	способностью участвовать в работах по реализации политики информационной безопасности, применять комплексный подход к обеспечению информационной безопасности объекта защиты;

Теоретическая часть

На протяжении многих веков при ведении деловой переписки, заключении контрактов и оформлении любых других важных бумаг подпись ответственного лица или исполнителя была непременным условием признания его статуса или неоспоримым свидетельством его важности. Подобный акт преследовал две цели:

- гарантирование истинности письма путем сличения подписи с имеющимся образцом;
- гарантирование авторства документа (с юридической точки зрения).

Выполнение данных требований основывается на следующих свойствах подписи:

- подпись аутентична, то есть с ее помощью получателю документа можно доказать, что она принадлежит подписывающему;
- подпись служит доказательством, что только тот человек, чей автограф стоит на документе, мог подписать данный документ, и никто другой не смог бы этого сделать;
- подпись непереносима, то есть является частью документа и поэтому перенести ее на другой документ невозможно:
- документ с подписью является неизменяемым, то есть после подписания его невозможно изменить, оставив данный факт незамеченным;
- подпись неоспорима, то есть человек, подписавший документ, в случае признания экспертизой, что именно он засвидетельствовал данный документ, не может оспорить факт подписания;
- любое лицо, имеющее образец подписи, может удостовериться в том, что данный документ подписан владельцем подписи.

С переходом к безбумажным способам передачи и хранения данных, а также с развитием систем электронного перевода денежных средств, в основе которых –

электронный аналог бумажного платежного поручения, проблема виртуального подтверждения аутентичности документа приобрела особую остроту. Развитие любых подобных систем теперь немыслимо без существования электронных подписей под электронными документами. Однако применение и широкое распространение *электронно-цифровых подписей* (ЭЦП) повлекло целый ряд правовых проблем. Так, ЭЦП может применяться на основе договоренностей внутри какой-либо группы пользователей системы передачи данных, и в соответствии с договоренностью внутри данной группы ЭЦП должно иметь юридическую силу. Но будет ли электронная подпись иметь доказательную силу в суде, например, при оспаривании факта передачи платежного поручения?

Схема 1

Данная схема предполагает шифрование электронного документа (ЭД) на основе симметричных алгоритмов и предусматривает наличие в системе третьего лица (арбитра), пользующегося доверием участников обмена подписанными подобным образом электронными документами. Взаимодействие пользователей данной системой производится по следующей схеме (рис.1):

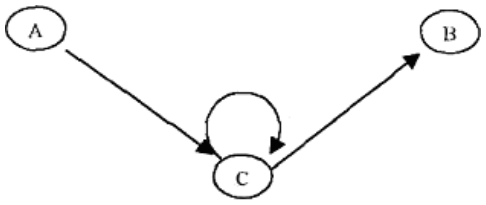

Рис. 1- Основные методы построения схем ЭЦП.

Участник А зашифровывает сообщение своим секретном ключе K_A, знание которого разделено с арбитром (С на рис. 1), затем шифрованное сообщение передается арбитру с указанием адресата данного сообщения (информация, идентифицирующая адресата, передается также в зашифрованном виде).

Арбитр расшифровывает полученное сообщение ключом K_A, производит необходимые проверки и затем зашифровывает его секретным ключом участника В (K_B). Далее зашифрованное сообщение посылается участнику В вместе с информацией, что оно пришло от участника А.

Участник В расшифровывает данное сообщение и убеждается в том, что отправителем является участник А.

Авторизацией документа в данной схеме считается сам факт шифрования электронного документа (ЭД) секретным ключом и передача зашифрованного ЭД арбитру. Основным преимуществом этой схемы является наличие третьей стороны, исключающей какие-либо спорные вопросы между участниками информационного обмена, то есть в данном случае не требуется дополнительной системы арбитража ЭЦП. Недостатком схемы является так же наличие третьей стороны и использование симметричных алгоритмов шифрования. На практике эта схема не получила широкого распространения.

Схема 2

Фактом подписания документа в данной схеме является шифрование документа секретным ключом его отправителя. Здесь используются асимметричные алгоритмы шифрования (рис.2).

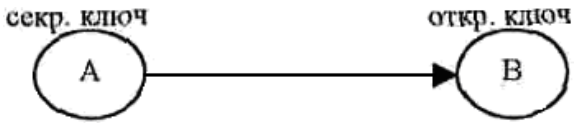

Рис. 2- Основные методы построения схем ЭЦП.

Вторая схема используется довольно редко вследствие того, что длина ЭД может оказаться очень большой (шифрование асимметричным алгоритмом может оказаться неэффективным по времени), но в этом случае в принципе не требуется наличие третьей стороны, хотя она и может выступать в роли сертификационного органа открытых ключей пользователя.

Схема 3

Наиболее распространенная схема ЭЦП использует шифрование окончательного результата обработки ЭД хэш-функцией при помощи асимметричного алгоритма. Структурная схема такого варианта построения ЭЦП представлена на рисунке 3.

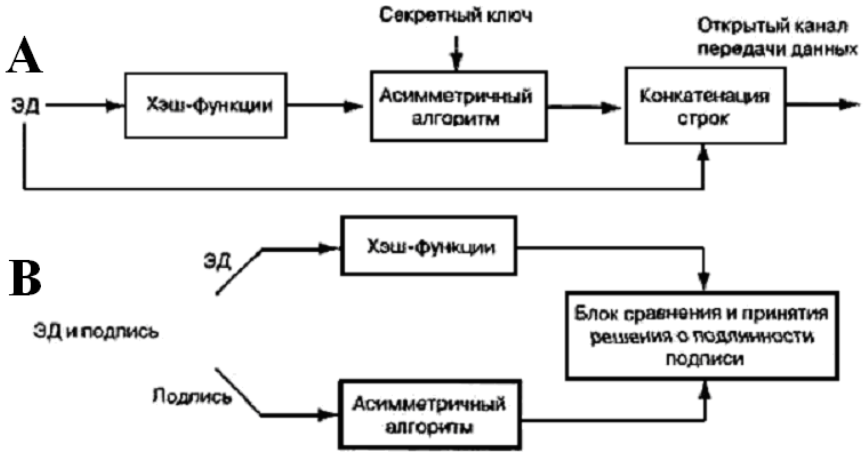

Рис.3- Основные методы построения схем ЭЦП

Процесс генерации ЭЦП происходит следующим образом. Участник А вычисляет хэш-код от ЭД. Полученный хэш-код проходит процедуру преобразования с использованием своего секретного ключа. После чего полученное значение (которое и является ЭЦП) вместе с ЭД отправляется участнику В.

Участник В должен получить ЭД с ЭЦП и сертифицированный открытый ключ участника А, а затем произвести расшифрование на нем ЭЦП, сам электронный документ (ЭД) подвергается операции хэширования, после чего результаты сравниваются, и если они совпадают, то ЭЦП признается истинной, в противном случае ложной.

В настоящее время применяются несколько алгоритмов цифровой подписи:

- RSA (наиболее популярен);
- Digital Signature Algorithm, DSA (алгоритм цифровой подписи американского правительства, который применяют в стандарте цифровой подписи (Digital Signature Standard, DSS), также используется часто);
- алгоритм Эль-Гамаля (иногда можно встретить).
- алгоритм, который применяют в стандарте ГОСТ Р34.10-94 (в основе лежит DSA и является вариацией подписи Эль-Гамаля);

• Так же существуют алгоритмы подписей, в основе которых лежит криптография эллиптических кривых; они похожи на все прочие, но в некоторых ситуациях работают эффективнее.

Электронная подпись RSA

Для осуществления подписи сообщения $m=m_1m_2m_3..m_n$ необходимо вычислить хеш-функцию $y=h(m_1m_2m_3..m_n)$, которая ставит в соответствие сообщению m число y. На следующем шаге достаточно снабдить подписью только число y, и эта подпись будет относиться ко всему сообщению m.

Далее по алгоритму RSA вычисляются ключи (e,n) и (d,n) Затем вычисляется $s = y^d \bmod n$ (d на этот раз секретная степень).

Число s это и есть цифровая подпись. Она просто добавляется к сообщению и получается подписанное сообщение <m,s>.

Теперь каждый, кто знает параметры подписавшего сообщение (т.е. числа e и n), может проверить подлинность подписи.

Для этого необходимо проверить выполнение равенства $h(m) = s^e \bmod n$.

Алгоритм Эль-Гамаля

Для генерации пары ключей сначала выбирается простое число p и два случайных числа g и x. Оба эти числа должны быть меньше p.

Чтобы подписать сообщение M, сначала выбирается случайное число k, взаимно простое с p-1. Затем вычисляется

$$a = g^k \bmod p$$

и с помощью расширенного алгоритма Евклида находится b в следующем уравнении:

$$M = (xa + kb) \bmod (p - 1)$$

Подписью является пара чисел: a и b. Случайное значение k должно храниться в секрете. Для проверки подписи нужно убедиться, что

$$y^a a^b \bmod p = g^M \bmod p$$

Открытый ключ:

p	простое число (может быть общим для группы пользователей)
g	$<p$ (может быть общим для группы пользователей)
y	$=g^x \bmod p$

Закрытый ключ:

x	$<p$

Подпись:

k	выбирается случайным образом, взаимно простое с p-1
a	(подпись) $=g^k \bmod p$
b	(подпись), такое что $M = (xa + kb) \bmod (p - 1)$

Проверка:

Подпись считается правильной, если $y^a a^b \bmod p = g^M \bmod p$

Пример (алгоритм Эль-Гамаля)

1)Пусть общие параметры для некоторого сообщества пользователей p=23 и g=5.

Пусть секретный ключ x=7. Вычислим открытый ключ y:

2) $y = 5^7 \bmod 23 = 17$

Пусть нужно поставить подпись на сообщение m=baaqab

Перейдем к вычислению подписи по алгоритму.

3)Прежде всего, вычисляется хеш-функция. Пусть её значение h(m)=h(baaqab)=M=3.

4) Затем генерируется случайное число k, например k=5. Вычисляем по формулам 5) $a = 5^5 \bmod 23 = 20$

И по расширенному алгоритму Евклида находим b

6)3=(7*20+5*b) mod 22

Такое b существует, т.к. НОД(k,p-1)=1. Получили b=21.

7)Получили подписанное сообщение в виде <baaqab,20,21>

Подписанное сообщение передается.

Полученное сообщение проверим на подлинность.

1)Прежде всего, вычисляется хеш-функция h(baaqab)=M=3.

2)Затем вычисляем левую часть

$y^a a^b \bmod p = g^M \bmod p$

$17^{20} * 20^{21} \bmod 23 = 16*15 \bmod 23 = 10$

3)и после этого правую $5^3 \bmod 23 = 10$

Так как левая часть совпала с правой, то можно сделать вывод, что подпись верна.

Оборудование и материалы

Для выполнения лабораторной работы предусмотрены компьютерные классы, находящиеся в аудиториях, оснащенных ПК, а также системное программное обеспечение – ОС MS Windows 7 и приложения Office (Excel Word, Access и т.п.).

Указания по технике безопасности

Лабораторная работа проводится на ПЭВМ. Запрещается прикасаться к задней панели системного блока при включенном питании, переключать разъемы интерфейсных кабелей периферийных устройств, загромождать верхние панели устройств бумагами и посторонними предметами, допускать попадание влаги на поверхность системного блока, монитора, клавиатуры и других устройств.

Порядок выполнения работы

Задание 1.

Реализовать приложение, позволяющие решить задачи в соответствии с вариантом:

1.Для указанных открытых ключей пользователя RSA проверить подлинность подписанных сообщений:

1) n=55, e=3: <7,28>, <22,15>,<16,36>

2) n=65, e=5: <6,42>, <10,30>,<6,41>

3) n=77, e=7: <13,41>, <11,28>,<5,26>

4) n=91, e=5: <15,71>, <11,46>,<16,74>

5) n=33, e=3: <10,14>, <24,18>,<17,8>

Задание 2.

Реализовать приложение, позволяющие решить задачи в соответствии с вариантом:

2. Абоненты некоторой сети применяют подпись Эль-Гамаля с общими параметрами p=23, g=5. Для указанных секретных параметров абонентов найти открытый ключ (y) и построить подпись для сообщения m:

1) x=11, k=3, m=15
2) x=10, k=15, m=5
3) x=3, k=13, m=8
4) x=18, k=7, m=5
5) x=9, k=19, m=15

Во всех вариантах будем предполагать, что h(m)=m для всех значений m.

№ варианта	№№ задач
1	1.1 , 2.1
2	1.2 , 2.2
3	1.3 , 2.3
4	1.4 , 2.4
5	1.5 , 2.5
6	1.4 , 2.2
7	1.3 , 2.1

Содержание отчета

1. Тема
2. Цель работы
3. Краткое описание выполненной работы.
4. Продемонстрировать данную работу на ПК, в соответствии с заданиями, оформить в программной оболочке Microsoft Word, **в**ключив в него копии экрана.
5. Сформулировать заключение и выводы
6. Ответить на контрольные вопросы.

Контрольные вопросы

1. Основные методы построения схем ЭЦП.
2. Шифрование окончательного результата обработки ЭД хэш-функцией при помощи асимметричного алгоритма.
3. Процессы генерации ЭЦП.
4. Электронная подпись RSA.
5. Алгоритм Эль-Гамаля.

Основная литература

1. Царев, Р.Ю. Программные и аппаратные средства информатики : учебник / Р.Ю. Царев, А.В. Прокопенко, А.Н. Князьков ; Министерство образования и науки Российской Федерации, Сибирский Федеральный университет. - Красноярск : Сибирский федеральный университет, 2015. - 160 с. : табл., схем., ил. - Библиогр. в кн. - ISBN 978-5-7638-3187-0 ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=435670

2. Лабораторный практикум по дисциплине Программно-аппаратные средства защиты информации [Электронный ресурс] / . — Электрон. Текстовые данные. — М. : Московский технический университет связи и информатики, 2016. — 31 с. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/61529.html

Дополнительная литература

1. Айдинян, А.Р. Аппаратные средства вычислительной техники : учебник / А.Р. Айдинян. - М. ; Берлин : Директ-Медиа, 2016. - 125 с. : ил., схем., табл. - Библиогр. в

кн. - ISBN 978-5-4475-8443-6 ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=443412

2. Привалов, И. М. (Институт сервиса, туризма и дизайна (филиал)СКФУ в г. Пятигорске). Основы аппаратного и программного обеспечения : учеб.-метод. пособие / И.М. Привалов ; Сев.-Кав. федер. ун-т. - Ставрополь : СКФУ, 2015. - 145 с. - 144 с.

Интернет-ресурсы:

1. http://www.biblioclub.ru/ - электронная библиотека
2.http://www.uts-edu.ru/ - «Электронные курсы»

Лабораторная работа 5.
Программно-аппаратное шифрование данных при их хранении.

Цель работы:

Освоить шифрование данных с использованием шифрующей системы (Encrypting File System - EFS).

Компетенции:

Код	Формулировка:
ОПК- 7	способностью определять информационные ресурсы, подлежащие защите, угрозы безопасности информации и возможные пути их реализации на основе анализа структуры и содержания информационных процессов и особенностей функционирования объекта защиты;
ПК-1	способностью выполнять работы по установке, настройке и обслуживанию программных, программно – аппаратных (в том числе криптографических) и технических средств защиты информации;
ПК-2	способностью применять программные средства системного, прикладного и специального назначения, инструментальные средства, языки и системы программирования для решения профессиональных задач;
ПК-3	способностью администрировать подсистемы информационной безопасности объекта защиты;
ПК-4	способностью участвовать в работах по реализации политики информационной безопасности, применять комплексный подход к обеспечению информационной безопасности объекта защиты;

Теоретическая часть

Шифрующая *файловая система* (Encrypting File *System* - *EFS*) появилась в операционных системах семейства *Windows*, начиная с *Windows 2000*. Она позволяет шифровать отдельные папки и файлы на томах с файловой системой *NTFS*. Рассмотрим этот механизм подробнее.

Сначала несколько слов о рисках, которые можно снизить, внедрив данный механизм. Повышение мобильности пользователей приводит к тому, что большое количество конфиденциальных данных (предприятий или личных) оказывается на дисках ноутбуков, на съемных носителях и т.д. *Вероятность* того, что подобное устройство будет украдено или временно попадет в чужие руки, существенно выше чем, например, для жесткого диска корпоративного персонального компьютера (хотя и в этом случае, возможны кражи или *копирование* содержимого накопителей). Если данные хранить в зашифрованном виде, то даже если носитель украден, *конфиденциальность* данных нарушена не будет. В этом и заключается цель использования *EFS*.

Следует учитывать, что для передачи *по* сети, зашифрованный *EFS файл* будет расшифрован, и для защиты данных в этих случаях надо использовать дополнительные *механизмы*.

Рассмотрим работу *EFS*. Пусть, у нас имеется *сервер Windows Server* 2008, входящий в *домен*, и три учетные записи, обладающие административными правами на сервере (одна из них - встроенная административная *запись Administrator*).

Пользователь **User1** хочет защитить конфиденциальные файлы. Тут надо отметить, что хотя шифровать с помощью *EFS* можно и отдельные файлы, рекомендуется применять *шифрование* целиком к папке.

User1 с помощью оснастки **Certificates** запрашивает сертификат (можно выбрать *шаблон User* или *Basic EFS*). Теперь у него появляется ключевая пара и *сертификат открытого ключа*, и можно приступать к шифрованию.

Чтобы зашифровать папку, в ее свойствах на вкладке **General** нажимаем кнопку **Advanced** и получаем *доступ* к атрибуту, указывающему на*шифрование* файла.

Работа *EFS* организована так, что одновременно сжатие и *шифрование* файлов и папок осуществляться не может. Поэтому нельзя разом установить атрибуты **Compress contents to save disk** и **Encrypt contents to secure data** (рис. 1).

Рис. 1- В свойствах папки устанавливаем шифрование

При настройках *по* умолчанию, зашифрованная *папка* выделяется в проводнике зеленым цветом. Для зашифровавшего *файл* пользователя порядок работы с ним не изменится.

Теперь выполним "переключение пользователей" и зайдем в систему под другой учетной записью, обладающей административными правами, но не являющейся встроенной административной записью. Пусть это будет **User2**.

Несмотря на то, что **User2** имеет такие же разрешения на *доступ* к файлу, что и **User1**, прочитать он его не сможет (рис. 2).

Также он не сможет его скопировать, т.к. для этого надо расшифровать *файл*. Но надо учитывать, что **User2** может удалить или переименовать *файл* или папку.

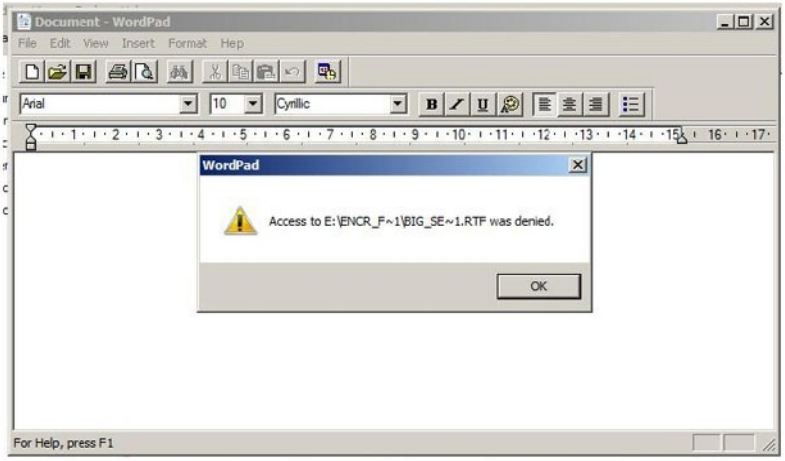

Рис. 2- Другой пользователь прочитать файл не сможет

Оборудование и материалы

Для выполнения лабораторной работы предусмотрены компьютерные классы, находящиеся в аудиториях, оснащенных ПК, а также системное программное обеспечение – ОС MS Windows 7 и приложения Office (Excel Word, Access и т.п.).

Указания по технике безопасности

Лабораторная работа проводится на ПЭВМ. Запрещается прикасаться к задней панели системного блока при включенном питании, переключать разъемы интерфейсных кабелей периферийных устройств, загромождать верхние панели устройств бумагами и посторонними предметами, допускать попадание влаги на поверхность системного блока, монитора, клавиатуры и других устройств.

Порядок выполнения работы

Задание 1.

1.Работая под первой учетной записью, запросите сертификат (если он не был получен ранее), после чего зашифруйте папку с тестовым файлом, который не жалко потерять. Проверьте, что будет происходить при добавлении файлов, переименовании папки, копировании ее на другой диск с файловой системой NFTS на том же компьютере, копировании папки на сетевой диск или диск с FAT.

2.Убедитесь, что никто другой не сможет прочитать зашифрованный файл.

3.Снова зайдите под первой учетной записью. В оснастке **Certificates**, удалите *сертификат пользователя* (несмотря на выдаваемые системой предупреждения). Завершите сессию пользователя в системе и войдите заново. Попробуйте открыть зашифрованный файл.

Как вы убедились, если сертификат и соответствующая ему ключевая пара удалены, *пользователь* не сможет прочитать зашифрованные им же данные. В частности поэтому, в *EFS* введена роль агента восстановления. Он может расшифровать зашифрованные другими пользователями данные.

Реализуется это примерно следующим образом. *Файл* шифруется с помощью симметричного криптоалгоритма на сгенерированном системой случайном ключе (назовем его **K1**). *Ключ* **K1** шифруется на открытом ключе пользователя, взятом из сертификата, и хранится вместе с зашифрованным файлом. Также хранится **K1**, зашифрованный на открытом ключе агента восстановления. Теперь либо *пользователь*, осуществлявший *шифрование*, либо *агент* восстановления могут *файл* расшифровать.

При настройке *по* умолчанию роль агента восстановления играет встроенная учетная *запись* администратора (локального, если *компьютер* не в домене, или доменная).

Задание 2.

Зайдите в систему под встроенной учетной записью администратора и расшифруйте папку. То, какой *пользователь* является агентом восстановления, задается с помощью групповых политик. Запустим оснастку **Group Policy Management**. В политике домена найдем группу **Public Key Policies** и там **Encrypting File System**, где указан сертификат агента восстановления (рис. 3). Редактируя политику (*пункт* **Edit** в контекстном *меню*, далее **Policies —> Windows Settings —> Security Settings —> Public Key Policies —> Encrypting File System**), можно отказаться от присутствия агентов восстановления в системе или наоборот, указать более одного агента (рис. 4).

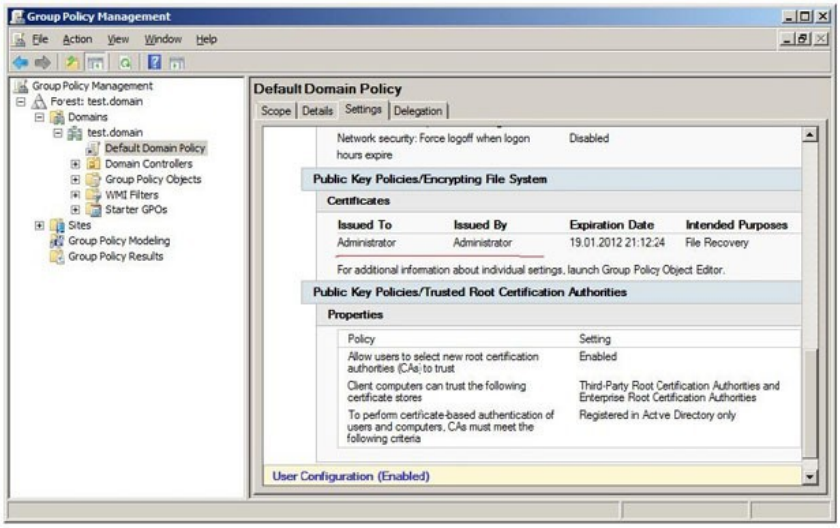

Рис. 3- Агент восстановления

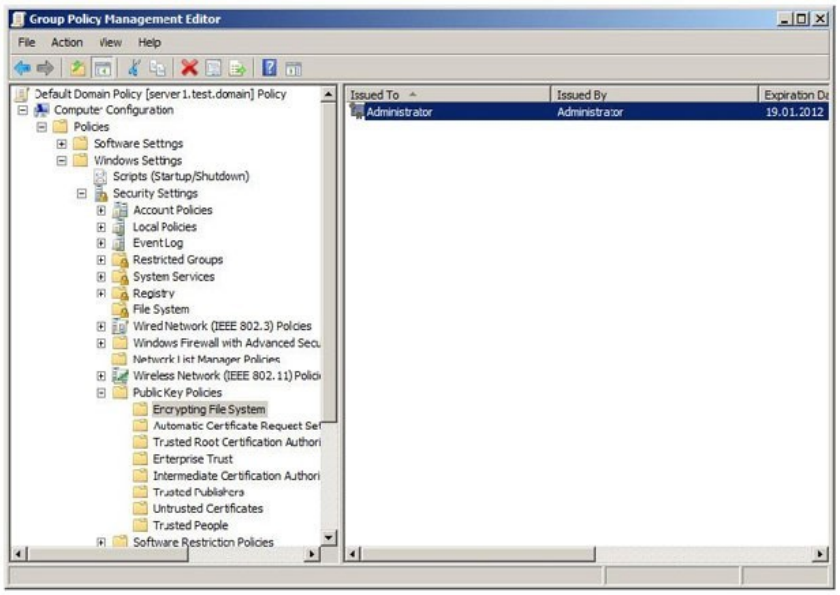

Рис.4- Изменение агента восстановления

Задание 3.

1.Отредактируйте политику таким образом, чтобы убрать из системы агента восстановления (удалите в *политике сертификат*). Выполнив команду "gpupdate /force" (меню **Start—>run—> gpupdate /force**) примените политику.

2.Повторив действия из предыдущих заданий, убедитесь, что теперь только тот пользователь, который зашифровал файл, может его расшифровать.

3.Теперь вернем в систему агента восстановления, но будем использовать новый сертификат. В редакторе политик находим политику **Encrypting File System** и в контекстном меню выбираем **Create Data Recovery Agent**. Это приведет к тому, что пользователь **Administrator** получит новый сертификат и с этого момента сможет восстанавливать шифруемые файлы.

Теперь рассмотрим, как можно предоставить *доступ* к зашифрованному файлу более чем одному пользователю. Такая настройка возможна, но делается она для каждого файла в отдельности.

В свойствах зашифрованного файла откроем окно с дополнительными параметрами, аналогичное представленному на рис.5 для папки. Если нажать кнопку **Details**, будут выведены подробности относительно того, кто может получить *доступ* к файлу. На рис. 5 видно, что в данный момент это *пользователь* **User1** и *агент* восстановления **Administrator**. Нажав кнопку **Add** можно указать сертификаты других пользователей, которым предоставляется *доступ* к файлу.

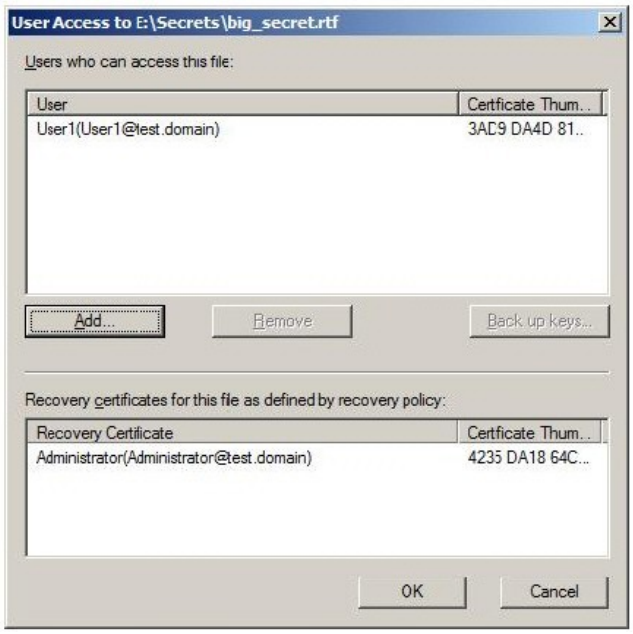

Рис. 5. Данные о пользователях, которые могут расшифровать файл

Задание 4

Зашифруйте *файл*. Предоставьте другому пользователю, не являющемуся агентом восстановления, возможность также расшифровать данный *файл*. Проверьте работу выполненных настроек.

Содержание отчета

1. Тема
2. Цель работы
3. Краткое описание выполненной работы.
4. Продемонстрировать данную работу на ПК, в соответствии с заданиями, оформить в программной оболочке Microsoft Word, **в**ключив в него копии экрана.
5. Сформулировать заключение и выводы
6. Ответить на контрольные вопросы.

Контрольные вопросы:

1. Шифрующая *файловая система* (Encrypting File *System - EFS*).
2. Работа шифрующей системы (Encrypting File System - EFS).
3. Шифрование ф*айла* с помощью симметричного криптоалгоритма.
4. Какой *пользователь* является агентом восстановления

Основная литература

1. Царев, Р.Ю. Программные и аппаратные средства информатики: учебник / Р.Ю. Царев, А.В. Прокопенко, А.Н. Князьков ; Министерство образования и науки Российской Федерации, Сибирский Федеральный университет. - Красноярск: Сибирский федеральный университет, 2015. - 160 с. : табл., схем. ил. - Библиогр. в кн. - ISBN 978-5-7638-3187-0; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=435670

2. Лабораторный практикум по дисциплине Программно-аппаратные средства защиты информации [Электронный ресурс] /. — Электрон. Текстовые данные. — М.: Московский технический университет связи и информатики, 2016. — 31 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/61529.html

Дополнительная литература

1. Айдинян, А.Р. Аппаратные средства вычислительной техники : учебник / А.Р. Айдинян. - М. ; Берлин : Директ-Медиа, 2016. - 125 с. : ил., схем., табл. - Библиогр. в кн. - ISBN 978-5-4475-8443-6 ; То же [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=443412

2. Привалов, И. М. (Институт сервиса, туризма и дизайна (филиал)СКФУ в г. Пятигорске). Основы аппаратного и программного обеспечения : учеб.-метод. пособие / И.М. Привалов ; Сев.-Кав. федер. ун-т. - Ставрополь : СКФУ, 2015. - 145 с. - 144 с.

Интернет-ресурсы:

1. http://www.biblioclub.ru/ - электронная библиотека
2.http://www.uts-edu.ru/ - «Электронные курсы»