МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Пятигорский институт (филиал) СКФУ

Методические указания

по выполнению лабораторных работ
по дисциплине
«ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ»
для направления подготовки **10.03.01 Информационная безопасность**
направленность (профиль) **Безопасность компьютерных систем**

Пятигорск
2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

1. Цель и задачи изучения дисциплины

Цель дисциплины: формирование набора универсальных и общепрофессиональных компетенций будущего бакалавра (специалиста) по направлению подготовки 10.03.01 Информационная безопасность.

Задачами дисциплины являются: дать основы о методах и средствах защиты информации в компьютерных системах; дать основы правил разграничения доступа и основных функций СЗИ, его обеспечивающих; дать основы практических аспектов построения систем ограничения доступа и других СЗИ; дать основы аппаратной реализации различных средств защиты информации; дать основы о защитных механизмах, реализованных в средствах защиты компьютерных систем от несанкционированного доступа (НСД); дать основы вопросов защиты ПО от несанкционированного использования; дать основы о применении средств криптографической защиты информации и средств защиты от НСД для решения задач обеспечения информационной безопасности; дать основы методов защиты от РПВ; дать основы методов и особенностей защиты объектов ОС; дать основы принципов построения файловой системы и моделей разграничения доступа к объектам.

2. Оборудование и материалы

Для проведения практических занятий необходимо следующее материально-техническое обеспечение: персональный компьютер; проектор; возможность выхода в сеть Интернет для поиска по образовательным сайтам и порталам; интерактивная доска.

3. Наименование лабораторных работ

№ Темы дисциплины	Наименование тем дисциплины, их краткое содержание	Объем часов
5 семестр		
13	**Лабораторная работа 1.** Программно-аппаратные средства защиты от DDoS атак.	5.4
13	**Лабораторная работа 2.** Межсетевые экраны.	5.4
13	**Лабораторная работа 3.** Построение и управление RAID – массивами и логическими томами.	5.4
13	**Лабораторная работа 4.** Программное восстановление данных.	5.4
17	**Лабораторная работа 5.** Обнаружение и предотвращение вторжений.	5.4
	Итого за 5 семестр	**27**
6 семестр		
18	**Лабораторная работа 6** Электронная цифровая подпись.	6
18	**Лабораторная работа 7.** Программно-аппаратное шифрование данных при их хранении	6
19	**Лабораторная работа 8** Защита программ от НСИ с помощью USB-ключей	6
22	**Лабораторная работа 9.** Sandbox – файловые антивирусы.	6
	Итого за 6 семестр	24

		Итого	**51**

4. Содержание лабораторных работ

ЛАБОРАТОРНЫЕ РАБОТЫ

1.Лабораторная работа №1. Протокол Kerberos.

Цель работы: Получить теоретические и практические навыки защиты от DDoS атак.

1. Теоретическая часть

DoS (от англ. Denial of Service — отказ в обслуживании) — хакерская атака на вычислительную систему с целью довести её до отказа, то есть создание таких условий, при которых легальные пользователи системы не могут получить доступ к предоставляемым системным ресурсам (серверам), либо этот доступ затруднён. Отказ «вражеской» системы может быть и шагом к овладению системой (если в нештатной ситуации ПО выдаёт какую-либо критическую информацию — например, версию, часть программного кода и т.д.). Но чаще это мера экономического давления: простой службы, приносящей доход, счета от провайдера и меры по уходу от атаки ощутимо бьют «цель» по карману. В настоящее время DoS и DDoS-атаки наиболее популярны, так как позволяют довести до отказа практически любую систему, не оставляя юридически значимых улик.

Если атака выполняется одновременно с большого числа компьютеров, говорят о **DDoS-атаке** (от англ. Distributed Denial of Service, распределённая атака типа «отказ в обслуживании»).

«Медленные атаки». Атаки типа SlowLoris. Атаки этого типа открывают множественные соединения к целевому web серверу и держат их открытыми как можно дольше путем посылки незавершенных HTTP запросов. Атакуемый сервер ждет завершения запроса оставляя соединение открытым. Периодически, SlowLoris посылает последующие заголовки HTTP для каждого запроса, но не пытается завершить соединение. В итоге пул подключений сервера заполняется и попытки доступа новых (легитимных) соединений отклоняются. Для успешной реализации атаки необходимо наличие уязвимой для атаки веб-службы (apache или сервис с похожей реализацией многопоточности) и отсутствие альтернативных средств влияющих на успешное проведение атаки (например, балансировщиков нагрузки).

TCP connection flood. Атаки этого типа инициируют большое количество соединений, но не пытаются завершить эти соединения в течение длительного времени, что задействует ресурсы сервера. Одним из подходов к реализации атаки является посылка большого количества SYN запросов, что, похоже, на атаки типа SYN flood, за исключением того, что используются реальные IP адреса и соединение корректно устанавливается. От таких атак можно защититься с помощью установки ограничения на количество соединений с IP адреса.

Однако новые методы атаки – «медленные» используют низкоскоростное соединение, в котором устанавливают соединение с сервером спустя несколько секунд после инициации и поддерживают соединение длительный период времени, регулярно посылая пакеты с данными формально соответствующими протоколу к серверу. Каждый IP адрес с которого производится атака может поддерживать сотни таких соединений, в то время как количество соединений сервера ограничено, что может привести к отказу в обслуживании. В качестве примера, ярким представителем такого класса атак является Slow Read attack.

«Произвольные пакеты» (случайные данные в пакетах по соответствующим протоколам). HTTP Flood Attack. Типичная работа веб браузера заключается в посылке HTTP запросов, в первую очередь GET и POST. Запрос GET служит для получения статичных ресурсов, таких как изображения; запрос POST используется для организации доступа к динамическому содержимому страниц. Атака наиболее эффективна в случаях когда вынуждает сервис выделить максимально доступный ресурс в ответ на простой единичный запрос. Атаки, использующие POST запросы, зачастую более ресурсоемки, инициируют сложную обработку на стороне сервера. С другой стороны атаки использующие запросы GET просты в создании и легко масштабируемы.

Атака случайными DNS-запросами на несуществующие домены. В атаках этого типа атакующий пытается переполнить количество одновременно-возможных DNS запросов, наводняя сервер запросами к множеству несуществующих доменов.

UDP Flood. Атаки этого типа используют особенности UDP протокола, где атакующий может создавать пакеты большого размера произвольного содержимого. Когда атакуемый узел получает такие пакеты на определенный порт, то выполняется проверка приложения, которое должно «слушать» этот порт. Когда выясняется отсутствие ассоциированного приложения, то выполняется посылка ответа в виде ICMP пакета Destination Unreachable. Таким образом, при большом количестве UDP пакетов, атакуемая система будет занята посылкой множества ICMP пакетов. Кроме того, UDP Flood может эффективно применяться для атаки на канал.

«SSL». Атака некорректными SSL/TLS пакетами. Атаки этого посылают произвольные данные к целевому серверу. SSL/TLS протоколы требуют выделения на стороне сервера значимого количества ресурсов, которые затрачиваются на попытки распознать произвольные данные в качестве легитимного запроса на инициацию SSL “рукопожатия”

Атака медленными SSL/TLS пакетами. Атаки этого типа, как и прочие «медленные» требуют небольшого количества пакетов для достижения отказа в обслуживании на относительно большом сервере. Атакующий в этом случае должен инициировать обычное SSL «рукопожатие» затем запросить пересмотр ключа шифрования, периодически повторяя этот ресурсоемкий для стороны сервера запрос, атакующий может исчерпать ресурсы атакуемого узла.

Атака SSL/TLS renegotiation. Атаки этого типа эксплуатируют слабость алгоритма «рукопожатия» SSL/TLS, где потребности в мощности процессора на стороне сервера на порядок превышают таковые на стороне клиента. Таким образом, если вычислительная мощность клиента и сервера равны с точки зрения реализации функций алгоритма RSA, то клиент может подавить сервер, посылая ему множество запросов инициации переговоров.

«Некорректные пакеты/фрагменты». Фрагментированные пакеты (UDP Fragment Flood). Атаки этого типа посылают UDP датаграммы, которые произвольно ссылаются на другие, отсутствующие в потоке датаграммы, что приводит к повышенному потреблению памяти на стороне атакуемого узла.

Некорректные IP фрагменты. Большой класс атак ориентируется на эксплуатацию уязвимостей в поддержке фрагментации пакетов в протоколе IP. Атаки этого типа могут приводить к отказу в обслуживании, или использоваться для обхода функций безопасности системы обнаружения вторжений. Одной из популярных атак этого типа является пересечение IP фрагментов. Атака может быть выполнена в случае, когда два фрагмента в одной и той же IP датаграмме имеют смещения, указывающие что они перекрывают друг друга при позиционировании. Некоторые системы неспособны корректно обработать подобные ситуации.

Неверные значения в заголовках пакетов. Атаки этого типа ориентируются на отдельные операционные системы и приложения, которые ошибочно обрабатывают некорректные значения в пакетах. Таким образом атаки этого типа могут приводить к

отказу в обслуживании, или использоваться для обхода функций безопасности системы обнаружения вторжений. Широко известным примером такого типа атаки может быть Land attack. В которой атакующий использует подмену IP адреса в заголовке пакета в комбинации с созданием TCP соединения. Причем устанавливается один и тот же адрес для источника и приемника, что приводит к попыткам установить соединение с самим собой.

«SMTP». SMTP Flood. В атаках этого типа злоумышленник пытается установить соединение с почтовым сервером и отправляет произвольных писем на сгенерированные случайным образом адреса, либо бездействует до истечения таймаута, удерживая соединение открытым. Каждое SMTP соединение утилизирует часть ресурсов сервера, тем самым атакующий пытается вызвать отказ в обслуживании.

1.2.Программы для выполнения лабораторной работы

hping3:

Установка:

sudo apt-get install hping3

Использование:

sudo hping3 -i u1 -S -p 80 192.168.1.1

где S - SYN, p 80 - порт 80, i u1 - Ждать 1 микро секунду между каждым пакетом

Можно также указать количество пакетов параметром - c :

sudo hping3 -i u1 -S -p 80 -c 10 192.168.1.1

slowhttptest:

Установка:

Скачать последнюю версию можно здесь:

https://code.google.com/p/slowhttptest/downloads/list

tar -xzvf slowhttptest-x.x.tar.gz

cd slowhttptest-x.x

./configure --prefix=PREFIX

make

sudo make install

Здесь PREFIX должен быть заменён на абсолютный путь, где инструмент slowhttptest должен быть установлен.

У вас должна быть установлена libssl-dev <sudo apt-get install libssl-dev> для успешной компиляции этого инструмента. Большинство систем должны иметь его.

Использование. Режим SlowLoris:

slowhttptest -c 1000 -H -i 10 -r 200 -t GET -u http://192.168.1.37/info.php -x 24 -p 3

THC-SSL-DOS:

Установка:

sudo apt-get install libssl-dev

Скачивайте отсюда http://www.thc.org/thc-ssl-dos/

Разархивируйте, переходите в папку, выполните:

./configure

sudo make all install

Использование:

thc-ssl-dos 192.168.1.202 --accept

ddosim:

Установка:

sudo apt-get install libcap-dev
sudo apt-get install build-essential
Скачать ddosim можно здесь:
http://sourceforge.net/projects/ddosim/
Разархивируйте, перейдите в папку, выполните:
sudo ./configure
sudo make
sudo make install
Использование:
invalid HTTP requests:
./ddosim -d 192.168.1.2 -p 80 -c 10 -r HTTP_INVALID -i eth0
TCP connection flood:
./ddosim -d 192.168.1.2 -p 80 -c 0 -i eth0
HTTP valid requests:
./ddosim -d 192.168.1.2 -p 80 -c 0 -w 0 -t 10 -r HTTP_VALID -i eth0
SMTP:
./ddosim -d 192.168.1.2 -p 25 -k 10.4.4.0 -c 0 -r SMTP_EHLO -i eth0

GoldenEye:
Установка:
mkdir goldeneye
cd GoldenEye
wget https://github.com/jseidl/GoldenEye/archive/master.zip
unzip master.zip
cd GoldenEye-master/
Использование:
./goldeneye.py <URl>

Http Unbearable Load King:
Скачать можно здесь:
https://packetstormsecurity.com/files/112856/HULK-Http-Unbearable-Load-King.html
Запуск
python ./hulk.py 192.168.1.113 safe

Fudp - UDP flood:
Установка:
Скачать можно здесь
http://www.olszewski.cc/fudp/
Переходите в папку, которую только что извлекли из архива и выпоняете <sudo make>
Использование.
Команда для создания вывода:
tcpdump -qt -i lo -n proto UDP
Запуск командой:
./fudp -l *цель*

1.3.Защита от DDoS

Защита от SYN-флуда
Защита строится на отключении очереди «полуоткрытых» TCP-соединений:
sudo sysctl -w net.ipv4.tcp_max_syn_backlog=1024

Включение механизма TCP syncookies:

- sudo sysctl -w net.ipv4.tcp_syncookies=1

Ограничение максимального числа «полуоткрытых» соединений с одного IP к конкретному порту:

- sudo iptables -I INPUT -p tcp --syn --dport 80 -m iplimit --iplimit-above 10 -j DROP

Защита от UDP-флуда:

- Так как UDP-пакеты отсылаются на различные UDP-сервисы, то достаточно просто отключить их от внешнего мира и установить ограничение на количество соединений к DNS-серверу:

- sudo iptables -I INPUT -p udp --dport 53 -j DROP -m iplimit --iplimit-above 1

Защита от ICMP-флуда:

• Для того, чтобы защититься от ICMP-флуда нужно отключить ответы на запросы ICMP ECHO:

- sudo sysctl net.ipv4.icmp_echo_ignore_all=1

или с помощью брандмауэра:

- sudo iptables -A INPUT -p icmp -j DROP --icmp-type 8

Действия в начале DoS:

- Для определения SYN-флуда необходимо установить число «полуоткрытых» соединений:

- grep ":80\ " | grep SYN_RCVD

В идеале их не должно быть совсем (максимум 1-3). Если это не так, то можно говорить о наличии DoS-атаки. Сложнее обстоит дело с HTTP-флудом. В самом начале необходимо подсчитать количество подключений на 80 порт и количество процессов Apache. Если они значительно превышают среднестатистические, то следует задуматься.

- grep httpd | wc -l grep ":80\ " | wc -l

Список IP-адресов, с которых идут запросы на подключение можно посмотреть следующей командой:

- grep ":80\ " | sort | uniq -c | sort -nr | less

Далее следует провести анализ пакетов с помощью команды tcpdump:

- sudo tcpdump -n -i eth0 -s 0 -w output.txt dst port 80 and host IP-сервера

Если наблюдается большой поток однообразных пакетов с разных IP-адресов, которые направлены на один порт, можно смело предполагать, что имеет место DoS-атака. Далее необходимо сбросить все эти соединения:

- sudo iptables -A INPUT -s xxx.xxx.xxx.xxx-p tcp – destination-port http -j DROP

2.3.Задания к лабораторной работе

• Для выполнения данной лабораторной работы нужны две виртуальные машины, 1 - с установленным web-сервер, и 2 - которая будет проводить DoS-атаку. На вторую тоже нужно поставить web-сервер, так как она тоже может быть целью некоторых программ (slowhttptest), выполняющих DoS атаку.

• Произведите DDoS атаки следующих типов (В теоретической части есть описание того, как устанавливать и использовать каждую из программ для проведения атак):

1. SYN flood используйте hping3
2. SlowLoris используйте slowhttptest
3. SSL используйте THC-SSL-DOS
4. TCP connection flood используйте ddosim
5. HTTP DDoS with valid requests используйте ddosim
6. HTTP DDoS with invalid requests используйте ddosim
7. Для HTTP flood используйте Goldeneye или Http Unbearable Load King
8. Для UDP flood используйте Fudp

• Защитите сервер от DDoS.

1.4. Вопросы к лабораторной работе

1. В чем отличие DDoS от DoS?
2. Что такое UDP-флуд?
3. Как защититься от DDoS?
4. Что такое SYN-флуд?
5. Как определить что используют SYN-флуд?
6. Как защититься от HTTP-флуда?
7. Как работают атаки типа SlowLoris?
8. Как совершить DDoS-атаку на почтовый сервер?
9. Как работают атаки типа SSL?

Составьте отчет о выполнении лабораторной работы. Включите в него копии экрана и ответы на вопросы лабораторной работы.

Контроль выполнения задания производится по окончании занятия и на консультациях в форме защиты выполненной работы, предоставленной в электронном и в бумажном виде в форме «Отчет по лабораторной работе».

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-2	1-3	1-3

Оценочные средства: собеседование, отчет.

2.Лабораторная работа №2. Межсетевые экраны (Iptables, WEB APLICATION FIREWALL).

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

1.1. Теоретическая часть.

Межсетевой экран

Скорее всего, ранее 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 nat.

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

Основные цепочки межсетевого экрана Netfilter:

• PREROUTING — изначальная обработка входящих пакетов

• INPUT — для входящих пакетов, адресованных непосредственно локальному компьютеру

• FORWARD — для маршрутизируемых пакетов

• OUTPUT — для пакетов, исходящих с локального компьютера

• POSTROUTING — для окончательной обработки исходящих пакетов

Таблицы межсетевого экрана Netfilter:

• raw - используется для маркировки пакетов, которые не должны обрабатываться системой определения состояний. Содержится в цепочках PREROUTING и OUTPUT.

• mangle — содержит 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.
-

Задания к лабораторной работе
Часть 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.

Часть 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/*.conf"
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>

Вопросы к лабораторной работе

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?

Составьте отчет о выполнении лабораторной работы. Включите в него копии экрана и ответы на вопросы лабораторной работы.

Контроль выполнения задания производится по окончании занятия и на консультациях в форме защиты выполненной работы, предоставленной в электронном и в бумажном виде в форме «Отчет по лабораторной работе».

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-2	1-3	1-3

Оценочные средства: собеседование, отчет.

3.Лабораторная работа № 3 Построение и управление RAID – массивами и логическими томами.

Цель работы: Получение теоретических и практических навыков построения и управления RAID массивами и логическими томами.

3.1. Теоретическая часть

Консольные команды:

- mdadm <параметры> - Консольная программа управления программными RAID массивами в Linux.
- lvm <параметры> - Консольная программа управления логическими томами LVM.
- parted <параметры> - Консольная программа для управления дисками
- watch <параметры> - Консольная программа, которая позволяет следить за изменениями в выводе команды.

RAID

RAID (Redundant Array of Independent Disks - избыточный массив независимых жестких дисков) - массив, состоящий из нескольких дисков, управляемых программным или аппаратным контроллером, связанных между собой и воспринимаемых как единое целое. В зависимости от того, какой тип массива используется, может обеспечивать различные степени быстродействия и отказоустойчивости. Служит для повышения надежности хранения данных и/или для повышения скорости чтения/записи информации.

Калифорнийский университет в Беркли предложил следующие уровни спецификации RAID, которые являются стандартом во всем мире:

- RAID 0 представлен как дисковый массив повышенной производительности, без отказоустойчивости. (Требуется минимум 2 диска)
- RAID 1 определен как зеркальный дисковый массив. (Требуется минимум 2 диска)
- RAID 2 массивы, в которых применяется код Хемминга. (Требуется минимум 7 дисков, для рационального использования)
- RAID 3 и 4 используют массив дисков с чередованием и выделенным диском четности. (Требуется минимум 4 диска)
- RAID 5 используют массив дисков с чередованием и “невыделенным диском четности”. (Требуется минимум 3 диска)
- RAID 6 используют массив дисков с чередованием и двумя независимыми “четностями” блоков. (Требуется минимум 4 диска)
- RAID 10 - RAID 0, построенный из RAID 1 массивов. (Требуется минимум 4 диска, четное количество)
- RAID 50 - RAID 0, построенный из RAID 5 массивов. (Требуется минимум 6 дисков, четное количество)
- RAID 60 - RAID 0, построенный из RAID 6 массивов. (Требуется минимум 8 дисков, четное количество)

Пример создания RAID 10

Проверим наличие виртуальных дисков.

```
sit@sit:~$ sudo parted -l
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sda: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type      File system  Flags
 1      1049kB  256MB   255MB   primary   ext2         boot
 2      257MB   21.5GB  21.2GB  extended
 5      257MB   21.5GB  21.2GB  logical               lvm

Error: /dev/sdb: unrecognised disk label
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 8590MB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:

Error: /dev/sdc: unrecognised disk label
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdc: 8590MB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:

Error: /dev/sdd: unrecognised disk label
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdd: 8590MB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:

Error: /dev/sde: unrecognised disk label
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sde: 8590MB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:

Error: /dev/sdf: unrecognised disk label
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdf: 8590MB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:

Model: Linux device-mapper (linear) (dm)
Disk /dev/mapper/sit--vg-swap_1: 533MB
```

```
Sector size (logical/physical): 512B/512B
Partition Table: loop
Disk Flags:

Number  Start  End    Size   File system     Flags
 1      0.00B  533MB  533MB  linux-swap(v1)


Model: Linux device-mapper (linear) (dm)
Disk /dev/mapper/sit--vg-root: 20.7GB
Sector size (logical/physical): 512B/512B
Partition Table: loop
Disk Flags:
Number  Start  End    Size   File system  Flags
 1      0.00B  20.7GB  20.7GB  ext4
sit@sit:~$
```

Примечание

Как видно из листинга, у нас присутствуют диски sda (на котором установлена операционная система Linux),sdb,sdc,sdd,sde,sdf. Теперь можно построить массив RAID 10 из дисков sdb, sdc, sdd и sde, а диск sdf пометим как диск горячей замены (применяется для горячей замены в случае отказа одного из дисков RAID массива).

Предупреждение

Необходимо открыть два терминала. В одном создается RAID массив, в другом осуществляется процесс наблюдения за созданием RAID массива.

Запустим процесс отслеживания состояния RAID массивов в терминале №1:

```
sit@sit:~$ sudo watch -n1 cat /proc/mdstat
```

Создадим RAID 10 в отдельном терминале №2:

```
sit@sit:~$ sudo mdadm -C /dev/md0 -l 10 -n 4 -x 1 /dev/sd[b-f]
[sudo] password for sit:
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
sit@sit:~$
```

В терминале №1 наблюдаем процесс создания RAID 10:

```
Every 1.0s: cat /proc/mdstat      Wed Sep 23 18:02:03 2015
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid10 sdf[4](S) sde[3] sdd[2] sdc[1] sdb[0]
16760832 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]
  [==========>.......]      resync  =  61.3%  (10286144/16760832)  finish=0.5min
speed=201781K/sec
unused devices: <none>
```

Создадим раздел в 1GB с файловой системой ext4 на созданном RAID 10:

```
sit@sit:~$ sudo parted /dev/md0
[sudo] password for sit:
GNU Parted 3.2
Using /dev/md0
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mklabel
New disk label type? GPT
Warning: The existing disk label on /dev/md0 will be destroyed and all data on this disk
will be lost. Do you want to continue?
Yes/No? yes
(parted) mkpart
```

Partition name? []?
File system type? [ext2]? ext4
Start? 0
End? 1GB
Warning: The resulting partition is not properly aligned for best performance.
Ignore/Cancel? Ignore
(parted) print
Model: Linux Software RAID Array (md)
Disk /dev/md0: 17.2GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number Start   End   Size   File system Name Flags
 1   17.4kB 1000MB 1000MB ext4
(parted)

Отформатируем созданный раздел в файловую систему ext4:

sit@sit:~$ sudo mkfs.ext4 /dev/md0p1

Смонтируем созданный раздел:

sudo mount -t ext4 /dev/md0p1 /mnt/

Скопируем файлы на раздел с файловой системой ext4:

sudo cp -R /var/log/* /mnt/

Разрушим один диск и проверим целостность данных.:

Наблюдаем процесс как диск горячей замены встает на место сбойного диска

Every 1.0s: cat /proc/   Wed Sep 23 19:52:04 2015

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid10 sdf[4] sde[3] sdd[2] sdc[1] sdb[0](F)
16760832 blocks super 1.2 512K chunks 2 near-copies [4/3] [_UUU]
[=====>.............] recovery = 21.8% (1832192/8380416) finish=0.4min speed=229024K/sec
unused devices: <none>

Убедимся в целостности данных на разделе:

sit@sit:~$ ls -la /mnt/
total 968
drwxr-xr-x 9 root root   4096 Sep 23 19:34 .
drwxr-xr-x 22 root root   4096 Sep 19 14:26 ..
-rw-r--r-- 1 root root 18625 Sep 23 19:34 alternatives.log
drwxr-xr-x 2 root root   4096 Sep 23 19:34 apt
-rw-r----- 1 root root 41820 Sep 23 19:34 auth.log
-rw-r--r-- 1 root root 63653 Sep 23 19:34 bootstrap.log
-rw------- 1 root root   0 Sep 23 19:34 btmp
drwxr-xr-x 2 root root   4096 Sep 23 19:34 dist-upgrade
-rw-r----- 1 root root   31 Sep 23 19:34 dmesg
-rw-r--r-- 1 root root 339677 Sep 23 19:34 dpkg.log
-rw-r--r-- 1 root root 32032 Sep 23 19:34 faillog
drwxr-xr-x 2 root root   4096 Sep 23 19:34 fsck
drwxr-xr-x 3 root root   4096 Sep 23 19:34 installer
-rw-r----- 1 root root 189514 Sep 23 19:34 kern.log
drwxr-xr-x 2 root root   4096 Sep 23 19:34 landscape
-rw-r--r-- 1 root root 292292 Sep 23 19:34 lastlog
drwx------ 8 root root 16384 Sep 23 19:32 lost+found

-rw-r----- 1 root root 173386 Sep 23 19:34 syslog
-rw-r----- 1 root root   3090 Sep 23 19:34 syslog.1
-rw-r----- 1 root root   591 Sep 23 19:34 syslog.2.gz
-rw-r----- 1 root root 30788 Sep 23 19:34 syslog.3.gz
drwxr-x--- 2 root root   4096 Sep 23 19:34 unattended-upgrades
-rw-r--r-- 1 root root   8832 Sep 23 19:34 wtmp
sit@sit:~$ sudo head -n 10 /mnt/auth.log
Sep 19 14:38:02 sit systemd-logind [506]: Watching system buttons on /dev/input/event0 (Power Button)
Sep 19 14:38:02 sit systemd-logind[506]: Watching system buttons on /dev/input/event1 (Sleep Button)
Sep 19 14:38:02 sit systemd-logind[506]: Watching system buttons on /dev/input/event5 (Video Bus)
Sep 19 14:38:02 sit systemd-logind[506]: New seat seat0.
Sep 19 14:40:10 sit systemd-logind[508]: Watching system buttons on /dev/input/event0 (Power Button)
Sep 19 14:40:10 sit systemd-logind[508]: Watching system buttons on /dev/input/event1 (Sleep Button)
Sep 19 14:40:10 sit systemd-logind[508]: Watching system buttons on /dev/input/event6 (Video Bus)
Sep 19 14:40:10 sit systemd-logind[508]: New seat seat0.
Sep 19 14:40:27 sit login[529]: pam_unix(login:session): session opened for user sit by LOGIN(uid=0)
Sep 19 14:40:27 sit systemd-logind[508]: New session c1 of user sit.

sit@sit:~$ sudo head -n 10 /mnt/syslog
Sep 23 07:17:01 sit CRON[2263]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)
Sep 23 08:17:01 sit CRON[2266]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)
Sep 23 09:17:01 sit CRON[2269]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)
Sep 23 10:17:01 sit CRON[2272]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)
Sep 23 10:46:05 sit dhclient: DHCPREQUEST of 10.0.2.15 on eth0 to 10.0.2.2 port 67 (xid=0x6a9a8b24)
Sep 23 10:46:05 sit dhclient: DHCPACK of 10.0.2.15 from 10.0.2.2
Sep 23 10:46:05 sit dhclient: bound to 10.0.2.15 -- renewal in 42505 seconds.
Sep 23 11:17:01 sit CRON[2285]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)
Sep 23 12:17:01 sit CRON[2288]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)
Sep 23 13:17:01 sit CRON[2291]: (root) CMD (cd / && run-parts --report /etc/cron.hourly)

Сделаем имитацию замены извлечением и вставки нового диска.:

sit@sit:~$ sudo mdadm /dev/md0 -r /dev/sdb
mdadm: hot removed /dev/sdb from /dev/md0
sit@sit:~$ sudo mdadm /dev/md0 -a /dev/sdb
mdadm: added /dev/sdb
sit@sit:~$

Наблюдаем что диск sdb пометился как диск горячей замены.:

Every 1.0s: cat /proc/ Wed Sep 23 19:59:09 2015

```
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid10 sdb[5](S) sdf[4] sde[3] sdd[2] sdc[1]
      16760832 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]

unused devices: <none>
```

Примечание

Для того чтобы остановить RAID используется параметр **–stop** команды mdadm.

Для очистки записи принадлежности к программному RAID используется параметр **–zero-superblock** команды mdadm.

LVM

LVM (Logical Volume Manager) - менеджер логических томов является уникальной системой управления дисковым пространством. Она позволяет с легкостью использовать и эффективно управлять дисковым пространством. Уменьшает общую нагруженность и сложность существующей системы. У логических томов, которые созданы через LVM, можно легко изменять размер, а названия, которые им даны, помогут в дальнейшем определить назначение тома.

- PV, Physical Volume или физический том. Чаще всего это раздел на диске или весь диск. К ним относят устройства программного и аппаратного RAID массивов (которые могут включать в себя еще несколько физических дисков). Физические тома объединяются и образуют группы томов.

- VG, Volume Group или группа томов. Это самый верхний уровень модели представления, которая используется в LVM. С одной стороны группа томов может состоять из физических томов, с другой- из логических томов и представлять собой единую структуру.

- LV, Logical Volume или логический том. Раздел в группе томов, тоже самое, что раздел диска в не-LVM системе. Является блочным устройством и, как следствие, может содержать файловую систему.

- PE, Physical Extent или физический экстент. Каждый физический том делится на блоки данных - физические экстенты. Они имеют размеры как и у логических экстентов.

- LE, Logical Extent или логический экстент. Каждый логический том также делится на блоки данных - логические экстенты. Размеры логических экстентов не меняются в рамках группы томов.

Инициализация дисков и разделов

Перед тем, как начать использовать диск или раздел в качестве физического тома, важно его проинициализировать. Осуществляется это с помощью команды **pvcreate**. Данная команда создаст в начале диска или раздела дескриптор группы томов.

Для диска:

```
sit@sit:~$ sudo pvcreate /dev/sdb
[sudo] password for sit:
  Physical volume "/dev/sdb" successfully created
```

Для разделов:

```
sit@sit:~$ sudo pvcreate /dev/sdb1
[sudo] password for sit:
  Physical volume "/dev/sdb1" successfully created
```

Примечание

Повторяем данную операцию для всех дисков или разделов которые необходимо поменять как физические тома LVM.

В нашем случае это - sdb, sdc , sde, sdd, sdf.

Предупреждение

Если появилась ошибка инициализации диска с таблицей разделов, проверьте, что работаете с нужным диском. Убедившись в этом выполните следующие команды:

sudo dd if=/dev/zero of=/dev/sd* bs=1k count=1

sudo blockdev -rereadpt /dev/sd*

*Данные команды уничтожат существующую таблицу разделов на диске sd**. Для разделов воспользуйтесь утилитой fdisk (parted или gdisk) и установите тип раздела в 0x8e (LVM).

Просмотреть диски (разделы) которые помечены как физические тома LVM можно с помощью команды **pvdisplay**.

```
sit@sit:~$ sudo pvdisplay
  --- Physical volume ---
  PV Name               /dev/sdb
  VG Name               storage
  PV Size               8.00 GiB / not usable 4.00 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              2047
  Free PE               2047
  Allocated PE          0
  PV UUID               dt4vrH-xpIo-IOAR-4sZD-Q9cT-St7Q-dRKInS

  --- Physical volume ---
  PV Name               /dev/sdc
  VG Name               storage
  PV Size               8.00 GiB / not usable 4.00 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              2047
  Free PE               2047
  Allocated PE          0
  PV UUID               TD4x9x-t6dp-vrJ9-GnKk-eX1J-bU06-L17fnt

  --- Physical volume ---
  PV Name               /dev/sdd
  VG Name               storage
  PV Size               8.00 GiB / not usable 4.00 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              2047
  Free PE               2047
  Allocated PE          0
  PV UUID               qgJYg6-fNAu-9P2v-lBvt-u1H5-lfml-Pb186U

  --- Physical volume ---
  PV Name               /dev/sde
  VG Name               storage
  PV Size               8.00 GiB / not usable 4.00 MiB
  Allocatable           yes
```

```
PE Size               4.00 MiB
Total PE              2047
Free PE               2047
Allocated PE          0
PV UUID               bKGRsE-ZNNV-XtqW-bXpn-yOI1-DMdC-8rANuv

--- Physical volume ---
PV Name               /dev/sdf
VG Name               storage
PV Size               8.00 GiB / not usable 4.00 MiB
Allocatable           yes
PE Size               4.00 MiB
Total PE              2047
Free PE               2047
Allocated PE          0
PV UUID               W6TBLw-3Yt6-ZJE2-lcOb-PMni-F95G-lxmyHW
```

Создание группы томов.

Для создания группы томов необходимо воспользоваться командой **vgcreate**. На вход программы необходимо указать имя группы и диски (разделы) которые необходимо добавить в данную группу.

```
sit@sit:~$ sudo vgcreate storage /dev/sd[b-f]
  Volume group "storage" successfully created
```

Просмотреть группы томов в системе можно с помощью команды **vgdisplay**.

```
sit@sit:~$ sudo vgdisplay
  --- Volume group ---
  VG Name               storage
  System ID
  Format                lvm2
  Metadata Areas        5
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                5
  Act PV                5
  VG Size               39.98 GiB
  PE Size               4.00 MiB
  Total PE              10235
  Alloc PE / Size       0 / 0
  Free  PE / Size       10235 / 39.98 GiB
  VG UUID               Nf04a2-sQ5O-zRfO-V3jc-wpTj-KjYx-aKpeCK
```

Удаление группы томов.

Для удаления группы томов необходимо убедиться, что целевая группа томов не содержит логических томов. Далее необходимо деактивировать группу томов

```
sudo vgchange -an storage
```

После чего удалить группу томов командой

```
sudo vgremove storage
```

Примечание

Для того, чтобы добавить ранее инициализированный физический том в существующую группу томов используется команда **vgextend**

sudo vgextend storage /dev/sd*

Для того, чтобы удалить физический том из группы томов необходимо воспользоваться командой **vgreduce**

sudo vgreduce storage /dev/sd*

Создание логического тома.

Для того, чтобы например создать логический том “sit”, размером 1800Мб, необходимо выполнить команду

sudo lvcreate -L1800 -n sit storage

Примечание

Без указания суффикса размеру раздела, по умолчанию используется множитель M «мегабайт» (в системе СИ равный 10^6 байт), что показано в примере выше. Суффиксы в верхнем регистре - KMGTPE соответствуют единицам в системе СИ с основанием 10. Например, G — гигабайт равен 10^9 байт, а суффиксы в нижнем регистре - kmgtpe соответствуют единицам в системе IEC (с основанием 2), например g — гибибайт равен 2^{30}байт.

Для того, чтобы создать логический том размером 100 логических экстентов с записью по двум физическим томам и размером блока данных в 4 KB

sudo lvcreate -i2 -I4 -l100 -n sit storage

Если необходимо создать логический том, который будет полностью занимать группу томов, то сперва используйте команду vgdisplay, чтобы узнать полный размер группы томов, а после этого выполните команду **lvcreate**.

sudo vgdisplay storage | grep "Total PE"

Total PE 10230

sudo lvcreate -l 10230 storage -n sit

Эти команды создают логический том sit, полностью заполняющий группу томов. Тоже самое можно реализовать командой

lvcreate -l100%FREE storage -n sit

Удаление логических томов.

Перед удалением логический том должен быть размонтирован

sudo umount /dev/storage/sit

sudo lvremove /dev/storage/sit

lvremove -- do you really want to remove "/dev/storage/sit"? [y/n]: y

lvremove -- doing automatic backup of volume group "storage"

lvremove -- logical volume "/dev/storage/sit" successfully removed

Увеличение логических томов.

Для того, чтобы увеличить логический том, необходимо указать команде lvextend размер, до которого будет увеличен том (в экстентах или в размере)

sudo lvextend -L15G /dev/storage/sit

lvextend -- extending logical volume "/dev/storage/sit" to 15 GB

lvextend -- doing automatic backup of volume group "storage"

lvextend -- logical volume "/dev/storage/sit" successfully extended

В результате /dev/storage/sit увеличится до 15Гбайт.

Примечание

Для изменения размера файловых систем ext2, ext3 и ext4 используйте **resize2fs**.

Создание снап-шотов LVM

Для того, чтобы создать снапшот необходимо использовать **lvcreate -s**

sudo lvcreate -s -L10GB -n backup /dev/storage/sit

Таким образом мы создадим снапшот в 10 GB с именем backup для хранения изменений.

Задания к лабораторной работе
Часть 1

- Добавить пять виртуальных жестких дисков.
- Запустить Linux.
- Установить mdadm.
- Ознакомится с утилитой mdadm, ее возможностями и параметрами.
- В отдельном терминале следить за состоянием файла /proc/mdstat
- Собрать RAID 1 с помощью mdadm.
- Создать на созданном RAID файловую систему ext4.
- Смонтировать созданную файловую систему.
- Записать туда файл raid.txt с произвольным содержимым.
- Разрушить один из дисков RAID и проследить за происходящим в файле /proc/mdstat
- Проверить целостность файла raid.txt
- Остановить RAID 1.
- Очистить информацию дисков о принадлежности к программному RAID.
- Собрать RAID 0 с помощью mdadm.
- Создать на созданном RAID файловую систему ext3.
- Смонтировать созданную файловую систему.
- Записать туда файл raid.txt с произвольным содержимым.
- Разрушить один из дисков RAID и проследить за происходящим в файле /proc/mdstat
- Проверить целостность файла raid.txt
- Остановить RAID 0.
- Очистить информацию дисков о принадлежности к программному RAID.
- Собрать RAID 5 с диском горячей замены с помощью mdadm.
- Создать на созданном RAID файловую систему ext4.
- Смонтировать созданную файловую систему.
- Записать туда файл raid.txt с произвольным содержимым.
- Разрушить три диска RAID и проследить за происходящим в файле /proc/mdstat
- Проверить целостность файла raid.txt
- Остановить RAID 5.

- Очистить информацию дисков о принадлежности к программному RAID.
- Собрать RAID 10 с диском горячей замены с помощью mdadm.
- Создать на созданном RAID файловую систему ext2.
- Смонтировать созданную файловую систему.
- Записать туда файл raid.txt с произвольным содержимым.
- Разрушить два диска RAID и проследить за происходящим в файле /proc/mdstat
- Проверить целостность файла raid.txt
- Остановить RAID 10.
- Очистить информацию дисков о принадлежности к программному RAID.

Часть 2

- Инициализировать физические диски, поверх которых будет создан LVM.
- Создать группу томов на основе четырех виртуальных жестких дисков.
- Создать логический том.
- На созданном логическом томе создать файловую систему.
- Смонтировать систему и создать файл LVM.txt .
- Добавить в группу томов еще один виртуальный жесткий диск.
- Определить количество добавленных экстентов.
- Расширить созданный логический том на размер добавленных экстентов.
- Увеличить размер файловой системы.
- Сделать снапшот логического тома.
- Удалить группу томов и снапшот.

Вопросы к лабораторной работе

1. В чем достоинства и недостатки основных уровней RAID?
2. Какие основные команды использовались в лабораторной работе и каковы их назначения?
3. В чем смысл каждого из основных уровней RAID?
4. Сколько минимально необходимо дисков для каждого из основных уровней RAID?
5. Сколько максимально может выйти из строя дисков в основных уровнях RAID-массивов без потери данных?
6. Порядок действий для создания логического тома.

7. Что такое Snapshot? Как его создать, и какое его функциональное назначение в LVM.

8. Что такое экстенты в LVM, какое их функциональное назначение.

9. В чем отличие логического тома от физического?

Составьте отчет о выполнении лабораторной работы. Включите в него копии экрана и ответы на вопросы лабораторной работы.

Контроль выполнения задания производится по окончании занятия и на консультациях в форме защиты выполненной работы, предоставленной в электронном и в бумажном виде в форме «Отчет по лабораторной работе».

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-2	1-3	1-3

Оценочные средства: собеседование, отчет.

4.Лабораторная работа №4. Программное восстановление данных.

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

4.1. Теоретическая часть

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**.

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**.

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>**

Восстановление данных Foremost.

Foremost - консольная программа, позволяющая искать файлы на дисках или их образах по hex-данным, характерным заголовкам и окончаниям. Программа проверяет файлы на предмет совпадения заранее определённых hex-кодов (сигнатур), соответствующих наиболее распространённым форматам файлов. После чего экстрагирует их из диска/образа и складывает в каталог, вместе с подробным отчётом о том, чего, сколько и откуда было восстановлено. Типы файлов, которые foremost может сразу восстановить: jpg, gif, png, bmp, avi, exe, mpg, wav, riff, wmv, mov, pdf, ole, doc, zip, rar, htm, cpp. Есть возможность добавлять свои форматы (в конфигурационном файле /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**: **<sudo foremost -t jpg,gif,png,bmp -i /dev/sdb -o ~/out_dir>**
-

Задания к лабораторной работе

- 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.
- 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.

Вопросы к лабораторной работе

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?

Составьте отчет о выполнении лабораторной работы. Включите в него копии экрана и ответы на вопросы лабораторной работы.

Контроль выполнения задания производится по окончании занятия и на консультациях в форме защиты выполненной работы, предоставленной в электронном и в бумажном виде в форме «Отчет по лабораторной работе».

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-2	1-3	1-3

Оценочные средства: собеседование, отчет.

2.5.Лабораторная работа № 5. Обнаружение и предотвращение вторжений.

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

5.1. 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.

Задания к лабораторной работе

• 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.

• 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/
```

Вопросы к лабораторной работе

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?
10. Как в SNORT создавать логи?

Составьте отчет о выполнении лабораторной работы. Включите в него копии экрана и ответы на вопросы лабораторной работы.

Контроль выполнения задания производится по окончании занятия и на консультациях в форме защиты выполненной работы, предоставленной в электронном и в бумажном виде в форме «Отчет по практической работе».

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2	1-2	1-3	1-3

Оценочные средства: отчет.

6.Лабораторная работа № 6. Электронная цифровая подпись.

Цель работы:

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

6.1.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:

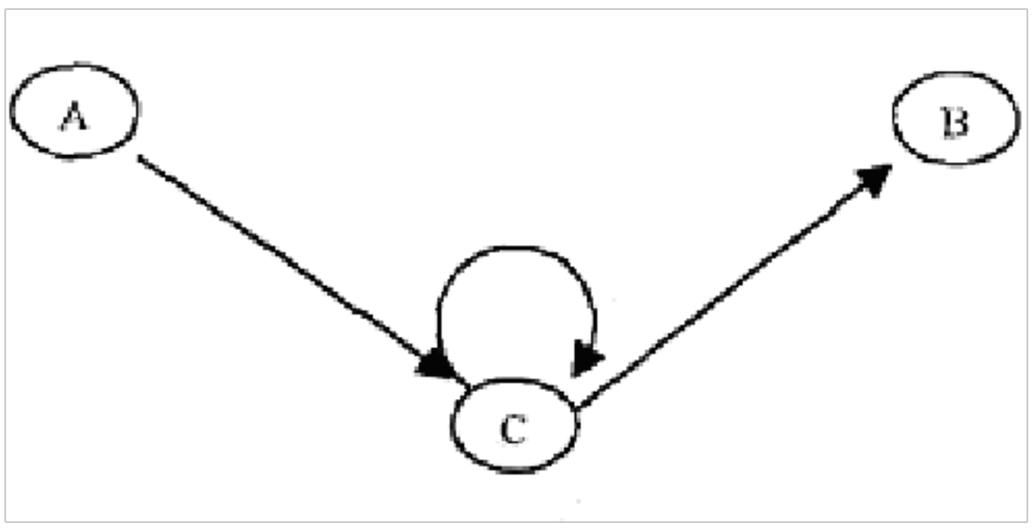

Рис. 5.1. Основные методы построения схем ЭЦП. Схема 1.

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. 5.2. Основные методы построения схем ЭЦП. Схема 2.

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.5.3. Основные методы построения схем ЭЦП. Схема 3.

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 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) (см. лабораторную работу №11 (4)).

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) $y=5 \bmod 23=17$

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.

Задание

Реализовать приложение, позволяющие решить задачи в соответствии с вариантом.

Задачи

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>

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

Контроль выполнения задания производится по окончании занятия и на консультациях в форме защиты выполненной работы, предоставленной в электронном и в бумажном виде в форме «Отчет по практической работе».

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..

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-2 Акустоэлектромагнитный активный канал утечки речевой информации	1-2	1-3	1-3

Оценочные средства: отчет.

7.Лабораторная работа № 7. Программно-аппаратное шифрование данных при их хранении.

Цель работы:

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

7.1.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.

Рис. 7.1. В свойствах папки устанавливаем шифрование

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. 7.1).

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. 7.2).

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.

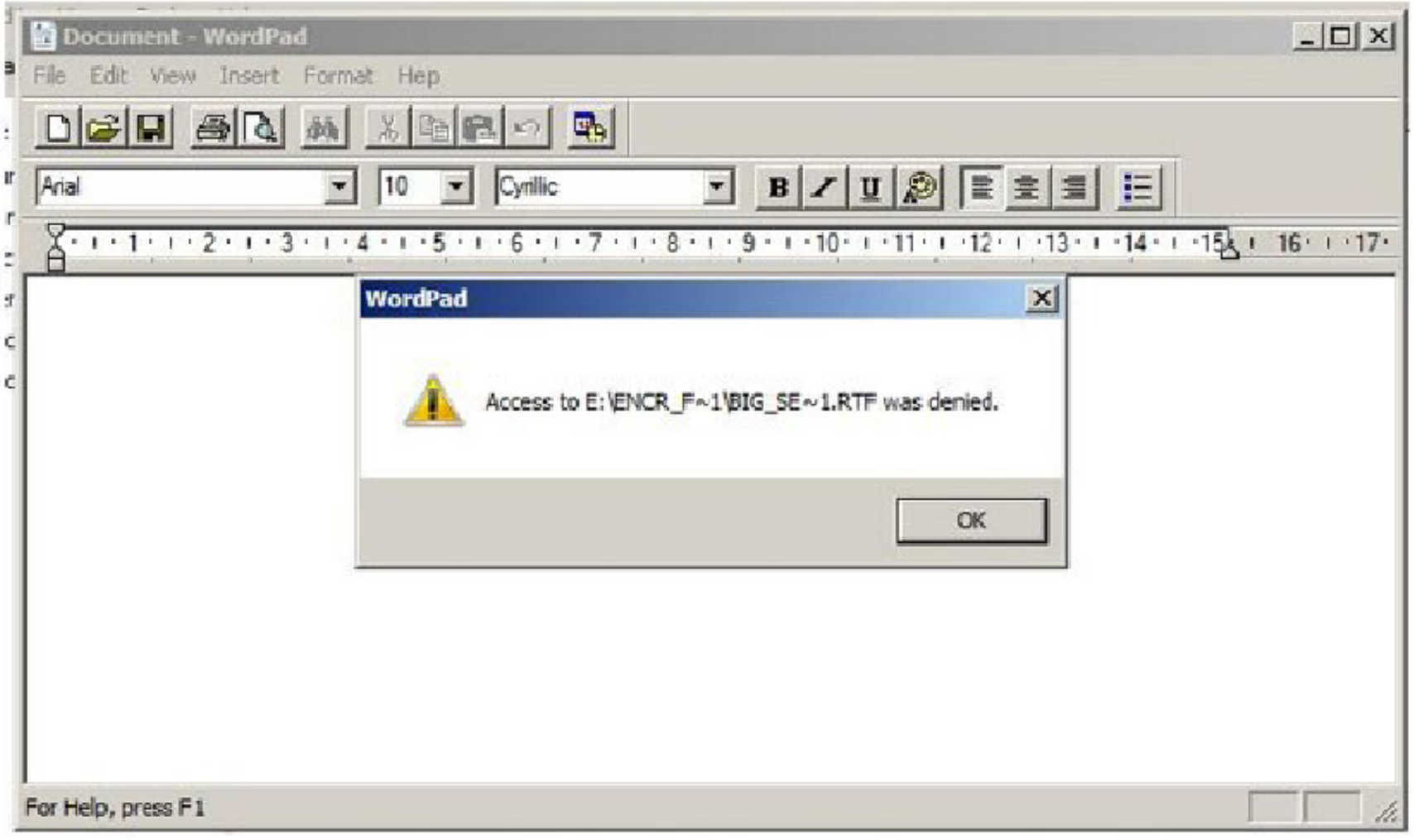

Рис. 7.2. Другой пользователь прочитать файл не сможет

Практическая часть

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