Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)

Факультет «Информатика и системы управления»
Кафедра «Компьютерные системы и сети»

В.Ю. Мельников, Е.К. Пугачев

Исследование методов защиты операционных систем и данных

Электронное учебное издание

Методические указания по выполнению лабораторных работ

по дисциплине "Операционные системы"

2017

Введение

Степень защищенности компьютера во многом зависит от совершенства установленной на нем операционной системы, которая должна обладать развитыми средствами шифрования данных и иметь инструменты настройки безопасности, как для одиночного компьютера, так и компьютера работающего в сети.

Цель работы - исследование методов защиты операционных систем и информации в операционных системах Linux, Windows и др.

Продолжительность работы - 4 часа.

Многопользовательская среда Linux

В отличие от Windows, Linux изначально был многопользовательским. Каждый пользователь имеет свой уникальный идентификатор UID (User ID), который представляет собой целое число в пределах от 0 до 65535. Пользователь с UID 0 – это суперпользователь (superuser, root), который имеет доступ ко всем ресурсам независимо от настроек доступа к ним (именно поэтому не рекомендуется постоянно работать с учётной записи суперпользователя – в случае взлома сессии злоумышленник получит доступ ко всем ресурсам).

Программы и демоны часто запускаются от имени специальных пользователей. Обычно, эти пользователи имеют то же имя, что и программа и создаются автоматически при установке программы. Под именем этого пользователя нельзя войти, он не имеет домашнего каталога. В случае взлома программы, злоумышленник получит доступ только к файлам этой программы и общедоступным ресурсам.

Каждый пользователь может принадлежать к нескольким группам У каждой группы есть свой идентификатор GID (Group ID), представляющий собой 16-битное целое число. Создание групп и внесение пользователя в

группы выполняется системным администратором (обычно при создании нового пользователя).

Каждому файлу и каталогу назначаются различные права доступа для различных пользователей и групп.

Кроме того, права пользователей настраиваются в конфигурационных файлах многих программ согласно документации на эти программы.

Создание пользователей

Добавим пользователя «user1» и зададим ему пароль:

useradd -m `user1`

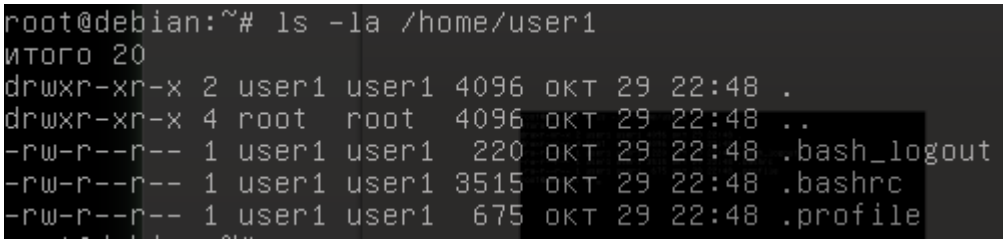
passwd `user1`

<пароль>

<повтор пароля>

Не забудьте опцию -m. Она создаёт домашний каталог, записывает в него шаблоны сценариев инициализации. По умолчанию, создаётся домашний каталог /home/ИМЯ_ПОЛЬЗОВАТЕЛЯ. Посмотрим, его содержимое:

`ls -la /home/user1`



Команда автоматически создала следующие сценарии:

~/.bash_profile — выполняется при входе пользователя в систему;

~/.bashrc — выполняется при каждом запуске дочернего интерпретатора команд;

~/.bash_logout — выполняется при выходе из системы.

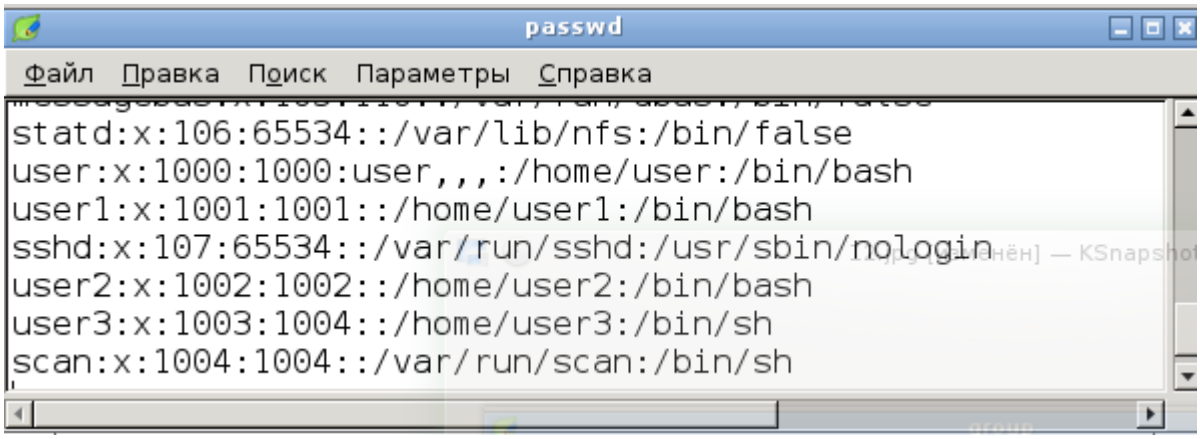
В файле «.bashrc» пользователь может добавить:

- Алиасы команд с часто употребляемыми опциями. Например, «`alias ll='ls -l'`» Этой командой мы сегодня часто будем пользоваться. Есть смысл раскомментировать соответствующую строку.

- Вид строки приглашения командной строки. Например, «PS1='\t\u\w\$'» выводит в строку приглашения: время (\t), имя пользователя (\u) и текущий каталог (\w).
- команда umask определяет права на вновь создаваемые файлы. Мы рассмотрим её позже

Чтобы изменения вступили в силу запустите новый экземпляр интерпретатора команд командой «bash»

Учётные всех пользователей данные хранятся в файле /etc/passwd



Поля файла passwd имеют следующую структуру:

`login:password:UID:GID:GECOS:home:shell`

Где:

`login` – регистрационное имя пользователя;

`password` – хэш пароля;

`UID` – идентификатор пользователя;

`GID` – идентификатор группы по умолчанию;

`GECOS` – информация о пользователе (например, полное имя, почта, телефон и т.д., данное поле не имеет чёткого синтаксиса, может использоваться для автоматизации некоторых процессов, например, отправки пользователю почтового сообщения при авторизации его логина в системе);

`home` – домашняя папка пользователя;

`shell` – Оболочка - интерпретатор команд. Именно он запускается при входе в Linux и выполняет команды, которые вводите в консоли. Он же вызывается для выполнения командных скриптов.

Изменить информацию о пользователе можно прямо в этом файле (командой vipw), или с помощью команды «**usermod**»

Для удаления пользователей используется команда:

userdel ПОЛЬЗОВАТЕЛЬ

Для смены пароля используется команда:

passwd ПОЛЬЗОВАТЕЛЬ

В отличии от прочих команд этой темы эта команда доступна всем пользователям. Так что пароль пользователя может сменить как root, так и сам пользователь.

Группы пользователей

Если несколько пользователей должны иметь доступ к файлу или каталогу следует создать группу и предоставить права этой группе.

При создании пользователя командой «useradd -m» автоматически создаётся так же группа с именем пользователя. Этой группой можно воспользоваться, чтобы дать права на свой файл только одному пользователю.

Команда «**groupadd** ГРУППА» создаёт новую группу

Команда «**groupdel** ГРУППА» удаляет группу.

Команда «**usermod -g** ГРУППА ПОЛЬЗОВАТЕЛЬ» - изменяет первичную группу пользователя. Файлы и каталоги, создаваемые пользователем будут принадлежать этой первичной группе. При создании пользователя в debian, пользователю автоматически назначается группа с тем же именем, что и имя пользователя.

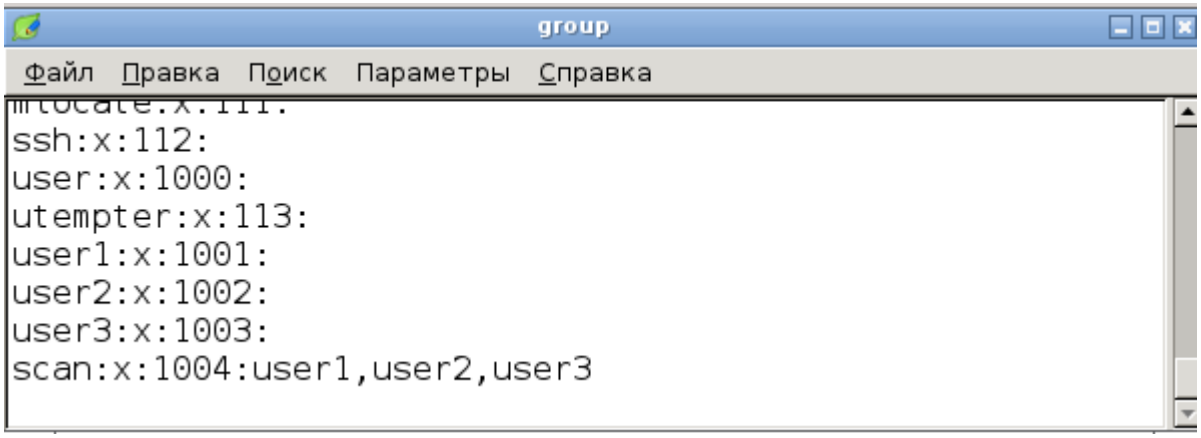
Команда «**usermod -G** ГРУППА1,ГРУППА2,... ПОЛЬЗОВАТЕЛЬ» - задаёт список дополнительных групп пользователя. При этом старый список дополнительных групп теряется, поэтому обычно используют другую форму:

Команда «**usermod -a -G** ГРУППА ПОЛЬЗОВАТЕЛЬ» - добавляет к списку групп заданного пользователя заданную дополнительную группу.

Для просмотра, в какие группы входит пользователь, используется команда «**groups** ПОЛЬЗОВАТЕЛЬ».

```
root@debian:/home# groupadd grp1
root@debian:/home# usermod -a -G grp1 user1
root@debian:/home# groups user1
user1 : user1 grp1
root@debian:/home# usermod -a -G grp1 user
root@debian:/home# groups user
user : user cdrom floppy audio dip video plugdev netdev grp1
root@debian:/home# _
```

Полный список групп с атрибутами содержится в файле /etc/group.



Файл group содержит записи следующего вида:

`group_name:password:GID:users`

Где:

`group_name` – имя группы;

`password` – пароль (в зашифрованном виде);

`GID` – идентификатор группы;

`users` – список пользователей, входящих в эту группу (через запятую). В этом списке не перечисляются пользователи, для которых эта группа является первичной.

Этот файл можно редактировать (командой «vigr»).

Управление правами доступа к файлам и каталогам

В Linux имеется 2 системы управления правами доступа:

- Улучшенная система прав доступа ACL (будет рассмотрена позже).
- Классическая система управления правами доступа. Информацию именно этой системы отображают все файловые менеджеры. Несмотря на простоту, возможностей этой системы достаточно в большинстве случаев, поэтому сначала рассмотрим её.

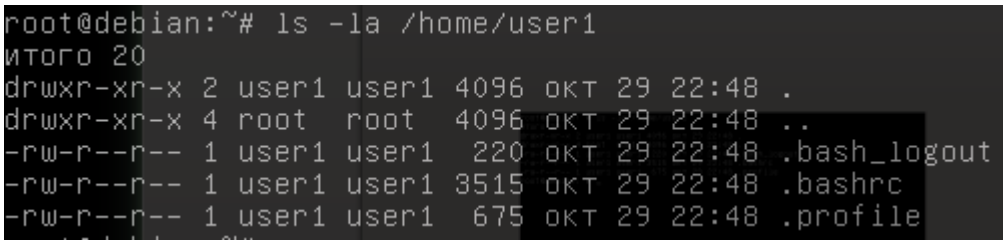
Пользователь, который создаёт файл или каталог становится его владельцем. И имеет на него все права. Чтобы дать доступ к этому файлу другим пользователям следует:

- Создать группу (groupadd)
- Включить в неё этих пользователей. (usermod)
- Сменить у файла группу — владельца (chown)
- При необходимости, дать права на запись (chmod)

Если надо дать права только одному пользователю, можно воспользоваться группой, которая автоматически создаётся вместе с пользователем и имеет имя пользователя.

Вернёмся к выводу команды

```
ls -la /home/user1
```



Для каждого файла и каталога linux хранит:

- идентификатор пользователя — владельца,
- идентификатор группы
- 16 битовое целое число, где каждый бит определяет права на чтение(r), запись(w), и выполнение(x) для владельца(u), группы(g) и прочих пользователей(o). Тут же хранится бит признака каталога (d) и

специальные права suid,sgid(s) и sticky bit(t). Рассмотрим эти права подробнее:

Категория пользователей	**Символ**	**Для файла**	**Для каталога**
	d	-	Признак каталога
Права пользователя владельца	r	Можно читать файл	Можно просматривать содержимое каталога
	w	Можно изменять файл	Можно создавать, удалять и переименовывать файлы
	x	Можно запускать на выполнение	Можно читать и выполнять файлы из каталога, даже если нет прав на чтение каталога
	s	Любой пользователь может запустить файл с правами владельца Применяется для системных утилит.	Не применяется
Права группы	r	Можно читать файл	Можно просматривать содержимое каталога
	w	Можно изменять файл	Можно создавать, удалять и переименовывать файлы
	x	Можно запускать на выполнение	Можно читать и выполнять файлы из каталога, даже если нет прав на чтение каталога
	s	Любой пользователь может запустить файл с правами группы Применяется для системных утилит.	Все файлы, создаваемые в каталоге принадлежат группе, владеющей каталогом Применяется для общих каталогов группы

Права прочих	r	Можно читать файл	Можно просматривать содержимое каталога
	w	Можно изменять файл	Можно создавать, удалять и переименовывать файлы
	x	Можно запускать на выполнение	Можно читать и выполнять файлы из каталога, даже если нет прав на чтение каталога
	t	Не используется	Удалять и переименовывать файлы в этом каталоге может только владелец файла или каталога, даже если есть права на запись в этот каталог Применяется для каталогов временных файлов

Разберём права на домашние каталоги пользователей по умолчанию:

```
root@debian:/home# ls -l /home/
итого 8
drwxr-xr-x 2 user  user  4096 ноя 18 02:15 user
drwxr-xr-x 7 user1 user1 4096 ноя 25 01:51 user1
```

У каталога «/home/user1»

Пользователь-владелец – «user1», группа-владелец – его собственная группа «user1». «d» признак каталога. Права пользователя-владельца (первая тройка) - «`rwx`» (естественно все права).

А вот права группы-владельца и прочих пользователей, пожалуй, избыточны: (вторая и третья тройки) «`r-x`» – права на чтение каталога (r) и возможность обращаться к подкаталогам и файлам этого каталога (x). С этими же правами пользователь будет создавать все новые каталоги и файлы. Значит, если ничего не предпринять, любой пользователь сможет просмотреть любой каталог и файл пользователя.

Безопаснее было бы установить права «`d rwx --- ---`»

С другой стороны, если пользователь хочет создать в домашней папке каталог и открыть к нему доступ для некоторой группы, придётся дать хотя бы права «d rwx --x ---», иначе пользователи этой группы не дойдут до этого каталога. Но лучше создавать общедоступные каталоги в каталоге /usr/share

Права на файл и каталог можно изменить командой:

chmod [**-R**] РЕЖИМ ФАЙЛ

Если задана опция -R, команда меняет права не только заданного каталога, но и входящих в него файлов и каталогов (рекурсивно). Обратите внимание, «R» надо задавать именно в верхнем регистре.

Режим задаётся в форме [ugoa][+-=][rwxst]

Первая группа символов определяет категорию пользователей: владелец (u), группа (g), прочие пользователи (o), все категории (a)— определяет

Вторая группа определяет воздействие: разрешить (+), запретить (-) установить (=)

Третья группа символов определяет изменяемые права (смотри приведённую выше таблицу)

Рассмотрим некоторые примеры:

chmod o-rw /home/user1 — отнимает у «прочих» пользователей права просматривать и изменять домашний каталог пользователя user1

```
root@debian:/home/user1# ls -l /home/
итого 8
drwxr-xr-x 2 user  user  4096 ноя 18 02:15 user
drwxr-xr-x 7 user1 user1 4096 ноя 25 01:51 user1
root@debian:/home/user1# chmod o-rw /home/user1
root@debian:/home/user1# ls -l /home/
итого 8
drwxr-xr-x 2 user  user  4096 ноя 18 02:15 user
drwxr-x--x 7 user1 user1 4096 ноя 25 01:51 user1
```

chmod o+x /home/user1 — даёт «прочим» пользователям возможность добраться до некоторых файлов и подкаталогов домашнего каталога пользователя user1, даже когда у них нет прав на чтение самого каталога.

chmod g=r /home/user1 — даёт права только на чтение пользователям группы, которой принадлежит файл.

chmod u=rwx,go-rwx /home/user1/testdir даёт все права владельцу файла и отнимает их у группы и прочих

chmod a-rwx /home/user1/testfile отнимает все права у всех, включая владельца. Теперь доступ к testfile имеет только пользователь root. Пользователь root вообще имеет права на всё что угодно. Постарайтесь не дать команду «rm -rf /» или от имени rot. Она удаляет все файлы и каталоги в файловой системе. Поэтому, подключайтесь пользователем root только когда вы собираетесь администрировать систему

Опытные пользователи иногда задают в команде «chmod» права в виде восьмеричного числа.

число составляется из следующих битов

Владелец	Группа	Прочие
400(r) 200(w) 100(x)	040(r) 020(w) 010(x)	004(r) 002(w) 001(x)

Например,

chmod 700 /home/user1/testdir — даёт все права (7=4+2+1) пользователю владельцу и отнимает все права у группы и прочих пользователей.

chmod 644 file — даёт права на чтение и запись (6=4+2) пользователю владельцу и права на чтение (4) у группы и прочих.

chmod 777 /home/user1/testdir — даёт все права для всех.

Смена владельца

Для того, чтобы изменить группу, которой принадлежит файл или каталог следует воспользоваться командой:

chgrp [**-R**] ГРУППА ФАЙЛ

При необходимости, пользователя - владельца файла можно сменить командой:

chown [**-R**] ВЛАДЕЛЕЦ ФАЙЛ

Если задана опция -R, команда меняет права не только заданного каталога, но и входящих в него файлов и каталогов (рекурсивно).

Эту команду часто приходится давать, если файл или каталог был создан пользователем root (в том числе, программой, запущенной от имени root).

Можно одной командой сменить и пользователя — владельца и группу:

chown [**-R**] ВЛАДЕЛЕЦ**:**ГРУППА ФАЙЛ

```
root@debian:/home# mkdir /home/user1/work
root@debian:/home# ls -l /home/user1/
итого 4
drwxr-xr-x 2 root root 4096 фев 11 00:30 work
root@debian:/home# chown user1:user1 /home/user1/work/
root@debian:/home# ls -l /home/user1/
итого 4
drwxr-xr-x 2 user1 user1 4096 фев 11 00:30 work
```

Иногда, этой же командой изменяют только группу

chown [**-R**] **:**ГРУППА ФАЙЛ

Права доступа на новые файлы и каталоги

При создании новых файлов и каталогов, в том числе путём копирования, новый файл получает права, заданные командой

umask РЕЖИМ

Чтобы узнать текущий режим дайте эту команду без параметров. Выводится восьмеричное число, которое указывает какие права доступа **запрещены**:

Число составляется из следующих битов:

Владелец	Группа	Прочие
400(r) 200(w) 100(x)	040(r) 020(w) 010(x)	004(r) 002(w) 001(x)

Данная команда не задаёт права на выполнение файлов.

По умолчанию установлено umask=022 — запрещено изменение файлов и каталогов группе и прочим пользователям. Остаются права на чтение файлов и каталогов.

Режим umask=027 запрещает изменение файлов и каталогов группе и запрещает все операции прочим пользователям.

Смена пользователя (switch user)

Когда требуется сменить пользователя используйте команду

su ПОЛЬЗОВАТЕЛЬ

Но помните, что при такой смене пользователя не выполняются сценарии инициализации, и вы остаётесь в том же каталоге.

Для того, чтобы получить полное окружение заданного пользователя используйте команду

su -l ПОЛЬЗОВАТЕЛЬ

Для смены пользователя на root удобнее дать команду «**su»** без параметров (в некоторых версиях linux «**su -**»)

Для защиты утилита su использует подключаемые модули аутентификации (Pluggable Authentication Modules, PAM), их настройки могут быть найдены в /etc/pam.conf и в папке /etc/pam.d/ (проверьте содержимое этой папки с помощью консоли).

Чтобы вернуться к работе с прежним пользователем, выполните команду «**exit**»

Воспользуйтесь командой «pstree» и определите, как работает команда «su» и что делает «exit».

Выполнение команд от имени другого пользователя

Помните, что суперпользователь root имеет права, на абсолютно все операции и работать под этим именем надо как можно меньше.

Для того, чтобы выполнить одну команду с правами другого пользователя безопаснее использовать одну из следующих команд:

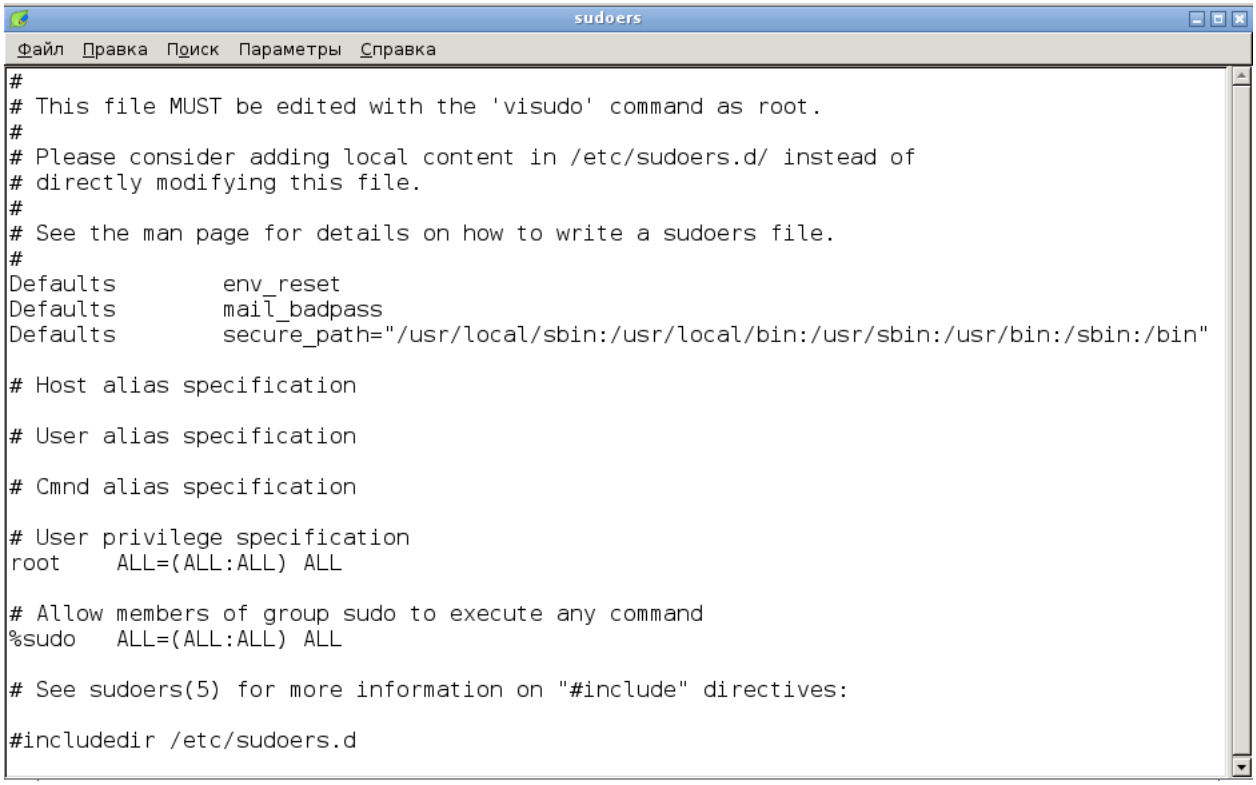
sudo -u ПОЛЬЗОВАТЕЛЬ КОММАНДА

sudo КОММАНДА (для выполнения команды от имени root)

sudo (обычно расшифровывается как «superuser do») – запуск команд от имени суперпользователя. Во многих дистрибутивах эта утилита является предустановленной, но в используемых в ЛР облегчённых дистрибутивах её нет. Установить её можно следующей командой:

```
apt-get install sudo
```

После установки требуется добавить пользователя в группу sudo. Для этого необходимо отредактировать файл /etc/sudoers.



Найдите строку с привилегиями пользователей (user privilege specification) и добавьте после строки пользователя root следующую запись:

```
user ALL=(ALL) ALL
```

Убедитесь, что теперь можно устанавливать недостающие пакеты без смены пользователя:

```
sudo apt-get update
```

По умолчанию при использовании команды sudo используется пароль **вызывающего пользователя, а не суперпользователя**, т.е. нужно вводить пароль «user», а не «root». Также у sudo есть таймаут работы (в минутах), в течение которых sudo не будет просить повторно ввести пароль. Добавление следующей строки к группе строк «Defaults» сделает таймаут нулевым (т.е. sudo будет спрашивать пароль при каждом запуске) и заставит утилиту запрашивать пароль не вызывающего пользователя, а пароль root:

```
Defaults timestamp_timeout=0,rootpw
```

Обратите внимание на то, как изменился запрос на ввод пароля при выполнении sudo.

Для предоставления пользователю прав суперпользователя мы редактировали файл, добавляя конкретного пользователя. Внимательно изучите содержимое файла. Как ещё можно предоставлять пользователям права суперпользователя?

Прочие утилиты для работы с учётными данными

`vipw` – запускает стандартный (т.е. указанной в переменной среды EDITOR) редактор и открывает в нём копию файла /etc/passwd. Это делается для безопасности: два пользователя не могут одновременно выполнить редактирование. После сохранения результатов и закрытия редактора замещает файл passwd изменённой копией;

`chfn` – изменение поля GECOS;

`chsh` – установить другую оболочку (изменение интерпретатора);

`id` – выдаёт информацию о пользователе (UID, GID, группы);

`vigr` - запускает стандартный редактор и открывает в нём копию файла /etc/group. После сохранения результатов и закрытия редактора замещает файл group изменённой копией;

groups – выдаёт список групп текущего пользователя;

w, who и users – Утилиты для просмотра работающих в системе пользователей. Утилита w выводит наиболее подробную информацию. Посмотрите результаты работы этих утилит;

whoami - Просмотр информации о своём пользователе (своего логина);

stat - подробный отчёт о файле

```
root@debian:~# stat /etc/passwd
  Файл: «/etc/passwd»
  Размер: 1124          Блоков: 8          Блок В/В: 4096   обычный файл
Устройство: 801h/2049d  Inode: 6273        Ссылки: 1
Доступ: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Доступ: 2013-12-07 16:11:17.232056770 +0400
Модифицирован: 2013-11-23 19:11:57.621251726 +0400
Изменён: 2013-11-23 19:11:57.653251323 +0400
 Создан: -
```

Политика паролей

Существует ещё один файл, похожий на /etc/passwd, он называется /etc/shadow. Содержимое этого файла показано на рисунке 2.

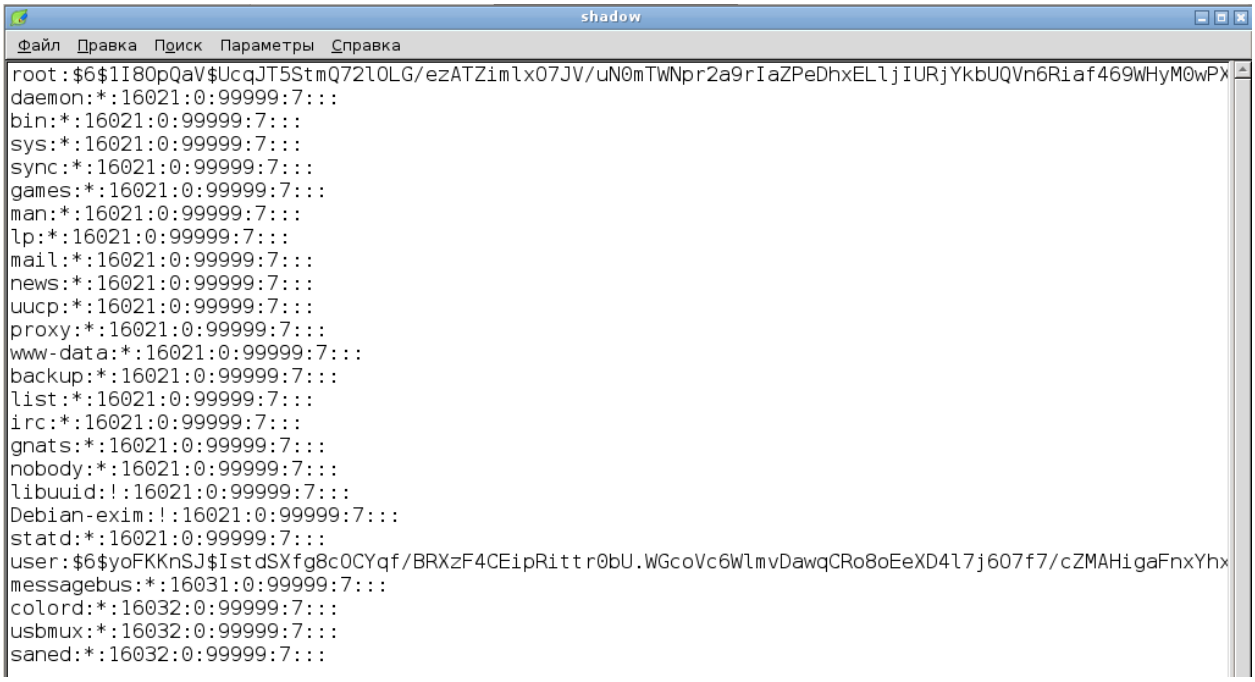


Рисунок 2 – Файл /etc/shadow

Файл shadow содержит в себе записи следующего вида:

`login:password:lastchanged:min:max:warn:inactive:expire`

Где:

`login` – регистрационное имя пользователя;

`password` – пароль в зашифрованном виде;

`lastchanged` – количество дней с 1 января 1970 года до даты, когда пароль был изменён;

`min` – минимальное число дней действия пароля;

`max` – максимальное число дней действия пароля;

`warn` – число дней до истечения срока действия пароля, т.е. до предупреждения пользователя об этом;

`inactive` – число дней, после которого с момента истечения срока действия пароля учетная запись будет заблокирована;

`expire` – число дней (с 1 января 1970), которое учётная запись заблокирована.

Этот файл помогает в настройке политики паролей, позднее он будет описан подробнее.

Настройка права доступа с помощью ACL

В большинстве случаев, достаточно классической системы прав доступа, в сложных случаях в Linux имеется улучшенная система ACL (Access Control List — список контроля доступа). В частности, с помощью неё можно задать на заданный файл, или каталог разные права для каждого из нескольких пользователей и/или групп.

getfacl - (get file access control list) выводит полную информацию о правах доступа к файлу или каталогу.

Для установки прав доступа используется помощью утилита setfacl. Примеры её использования:

`setfacl –m “u:user2:rwx” file // Установить пользователю user2 права на чтение, запись и выполнения файла file`

`setfacl –m “g:sudo:r-x” file // Установить группе sudo права на чтение и выполнение файла file`

Обратите внимание на различные права пользователей user1(владелец) и user2

Системы мандатного контроля доступа

Описанная выше модель безопасности представляет собой избирательное (дискреционное) управление доступом (DAC), в такой системе должны быть заданы чёткие однозначные соответствия субъекта с правами доступа к объекту. В модели мандатного контроля (MAC) пользователь, обладающий мандатом некоторого уровня, имеет доступ к ресурсам более низкого уровня доступа (т.е. менее защищённым), но не имеет доступа к более защищённым. Пользователь не может полностью управлять правами доступа к создаваемым им ресурсам во избежание проникновения злоумышленников.

Эти системы дают возможность настройки прав процессов без создания специальных пользователей для каждого процесса.

Имеются следующие системы MAC: AppArmor и SELinux.

SELinux — имеет большие возможности, но и настройка его более сложная. К сожалению, разработчики debian 8 исключили набор политик безопасности из репозитория, так что настраивать их придётся самостоятельно. Таким образом, для начинающего пользователя debian лучше использовать AppArmor.

AppArmor

Доступ к ресурсам определяется на основе профилей (profiles), которые привязаны к пути файла или каталога. Профиль разрабатывается индивидуально под каждое приложение и каждую версию linux (если приложение использует ресурсы linux).

Для установки AppArmor надо дать команду:

```
sudo apt-get install apparmor apparmor-profiles apparmor-utils apparmor-profiles-extra
```

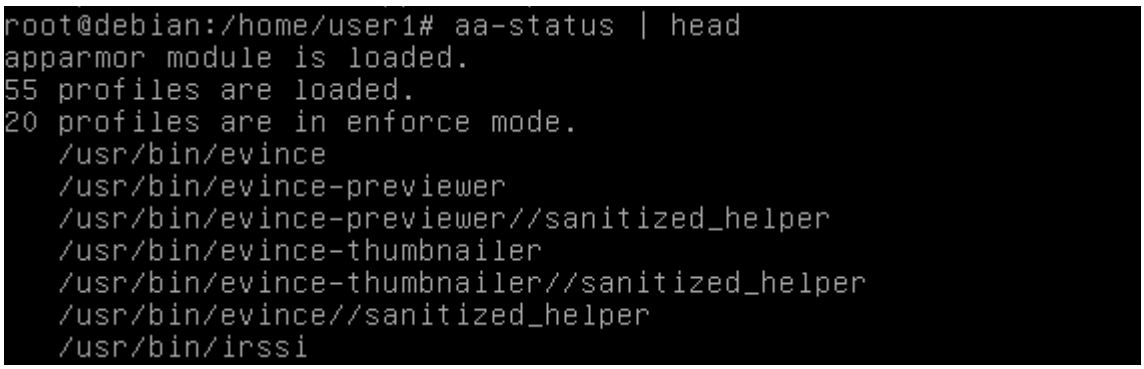
Затем, надо отредактировать файл «/etc/default/grub» - дописать в строку «GRUB_CMDLINE_LINUX=» текст " apparmor=1 security=apparmor"

и дать команды:

```
update-grub
```

```
reboot
```

Команда «`aa-status`» выводит информацию о загруженных профилях



Из первой строки видно, что AppArmor загружен.

Далее перечислены команды, которые будут запускаться с блокировкой нарушений (enforce mode) согласно загруженным профилям.

Далее перечислены команды, которые будут запущены в режиме обучения (complain mode) - программа будет только регистрировать нарушения ничего не блокируя.

В репозитории имеются профили для всех популярных приложений, кроме того, имеется утилита, с помощью которой легко создать профиль для нового приложения

Создадим для примера новый профиль для команды «free»:

- Запускаем утилиту «aa-genprof» командой

```
aa-genprof free
```

```
root@debian:/home/user1# sudo aa-genprof free
Writing updated profile for /usr/bin/free.
Setting /usr/bin/free to complain mode.

Before you begin, you may wish to check if a
profile already exists for the application you
wish to confine. See the following wiki page for
more information:
http://wiki.apparmor.net/index.php/Profiles

Please start the application to be profiled in
another window and exercise its functionality now.

Once completed, select the "Scan" option below in
order to scan the system logs for AppArmor events.

For each AppArmor event, you will be given the
opportunity to choose whether the access should be
allowed or denied.

Profiling: /usr/bin/free

[(S)can system log for AppArmor events] / (F)inish
```

- Запускаем программу в новом окне, работаем в ней, стараясь задействовать все функции.

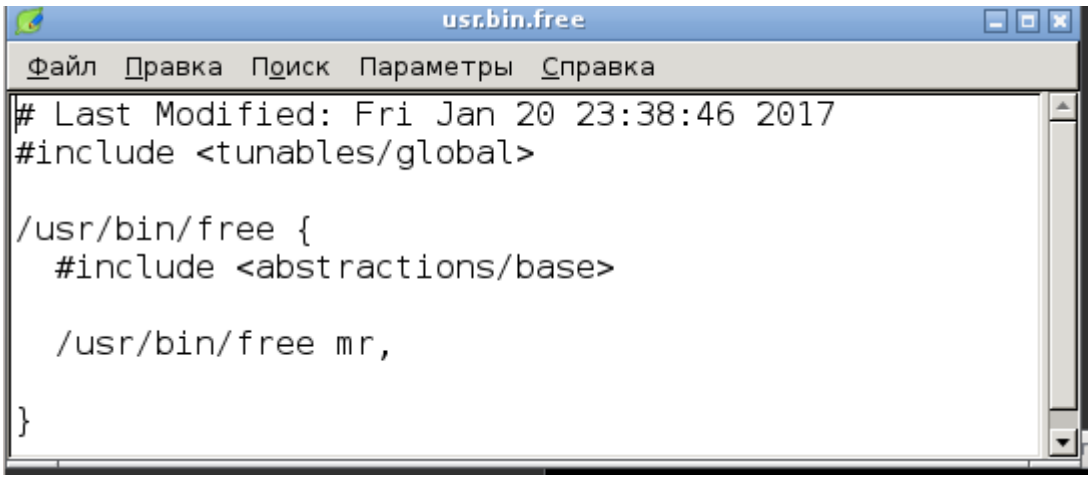
```
uxterm
root@debian:~# free -h
             total       used       free     shared    buffers     cached
Mem:          627M       269M       357M       4,4M        16M       183M
-/+ buffers/cache:        70M       557M
Swap:         382M         0B       382M
root@debian:~#
```

- Переключаемся в окно, в котором запустили «aa-genprof» и нажимаем «F». Утилита создаёт профиль «usr.bin.free», записывает его в каталог «/etc/apparmor.d/», загружает его, переводит в режим блокировки нарушений и завершает работу. Имя профиля соответствует пути к исполняемому файлу команды (в нашем случае это /usr/bin/free)

```
[(S)can system log for AppArmor events] / (F)inish
Setting /usr/bin/free to enforce mode.

Reloaded AppArmor profiles in enforce mode.

Please consider contributing your new profile!
See the following wiki page for more information:
http://wiki.apparmor.net/index.php/Profiles

Finished generating profile for /usr/bin/free.
```

- Редактируем профиль, выкидываем (комментируем) подозрительные файлы и каталоги.

  leafpad /etc/apparmor.d/usr.bin.free



Перед «{» записана команда «/usr/bin/free»
В фигурных скобках перечислены ресурсы, к которым может обращаться наша команда и права на эти ресурсы:
r - разрешить чтение
w - разрешить запись
a - разрешить запись в конец файла
px - разрешить запуск новых процессов если для них есть профиль
Px - разрешить запуск новых процессов, если для них есть профиль и стереть переменные окружения
ix - разрешить запуск нового процесса под профилем текущего
m - разрешить загружать исполняемые файлы в память и запускать
l - разрешить создавать символические ссылки на исполняемые файлы
k - разрешить блокировать файлы
ux - не контролировать новые процессы
Ux - не контролировать новые процессы и очистить переменные окружения.

- В некоторых дистрибутивах linux приходится дополнительно активировать защиту командой

```
sudo aa-enforce skype
```

Нарушения записываются в журнал «`/var/log/kern.log`»

SELinux

SELinux (Security-Enchansed Linux) представляет собой реализацию системы мандатного контроля доступа (MAC). SELinux был разработан АНБ, включён в ядро Linux в 2003 году (версия 2.6). По умолчанию в Linux существует папка /etc/selinux.

Основные понятия SELinux:

Сущность – термин, близкий к понятию «пользователь», однако сущность не меняется при исполнении команды su. Сущностью, например, можно назвать физического пользователя.

Домен – список действий, которые должен производить отдельный процесс. Домен должен включать в себя минимально все действия, необходимые процессу для выполнения его задачи.

Роль – совокупность нескольких доменов, которые могут быть использованы.

Тип – набор действий, которые могут быть выполнены применительно к объекту. Домен относится к процессам, тип же – к файлам, сокетам, каналам и т.д.

Контекст безопасности – набор всех атрибутов, связанных с объектами и субъектами. Для субъекта (процесса): сущность, роль, домен (могут быть добавлены и другие параметры).

Переход – смена контекста безопасности. Виды переходов:

1. Переход домена процесса – смена контекста безопасности процесса. SELinux меняет контекст запускаемого процесса на основе метки безопасности.
2. Переход типа файла – например, в определённом подкаталоге создаётся файл. Он будет невидим некоторой программе (например, WEB-серверу), т.к. эта программа не имеет доступа к контексту безопасности пользователя, требуется сменить контекст безопасности.

Политика – набор правил, контролирующих взаимодействие ролей, доменов и т.д.

Установка SELinux

SELinux включён в состав ядра Linux (начиная с версии 2.6)

Если версия вашего Linux ниже установите базовые модули SELinux , введите следующую команду (в одну строчку):

```
apt-get install selinux-basics selinux-policy-default auditd
```

`auditd` – демон аудита системы (в том числе и системных вызовов). Его настройки находятся в файле etc/audit/auditd.conf (рисунок 12).

Рисунок 12 – Файл auditd.conf

Подробную информацию можно получить с помощью man auditd.conf.

Правила аудита находятся в файле /etc/audit/audit.rules.

После завершения установки наберите в консоли:

```
selinux-activate
```

Перезагрузите Linux:

```
shutdown –r now
```

При загрузке можно заметить новые строки (рисунок 13).

Рисунок 13 – Загрузка Linux после установки SELinux

После первой перезарузки автоматически последует вторая.

После перезагрузки проверьте корректность установки:

```
check-selinux-installation
```

Для Debian Wheezy: не обращать внимания на предупреждение
o /etc/pam.d/login.

В дальнейшем проверку работы SELinux можно осуществлять с помощью команды `sestatus`.

Настройка

Для начала будет настроен демон аудита. Вывод существующих правил:

```
auditctl –l
```

На данном этапе не должно существовать никаких правил. Возможно добавление аудита только по определённым видам активности (добавляются после ключа –p):

r – аудит событий чтения;

w – аудит событий записи;

x – аудит событий исполнения/поиска;

a – аудит событий смены атрибута.

Установка аудита всех событий для папки /usr/bin/:

```
auditctl –w /usr/bin/ -p rwxa
```

Запустите leafpad, закройте его и посмотрите результат:

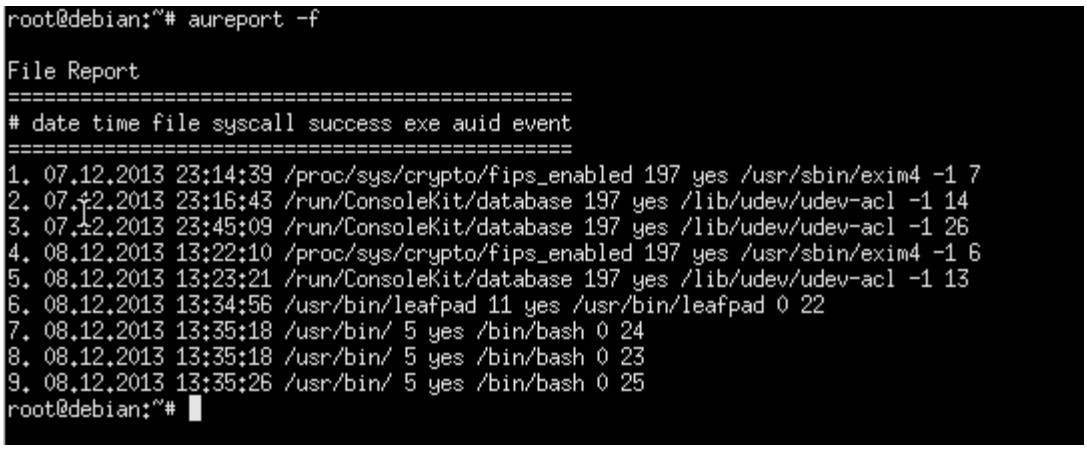


Рисунок 14 – Отчёт auditd

Самостоятельно расшифруйте вывод команды. Откройте man aureport и внесите в отчёт некоторые ключи, которые могут быть полезны при аудите системы. Приложите скриншоты работы aureport с этими ключами.

SELinux умеет работать в нескольких режимах:

- Принудительный (enforcing) – политики SELinux применяются принудительно, доступ может быть запрещён.
- Разрешающий (permissive) – политики SELinux не применяются, но действия журналируются.
- Отключён (disabled) – SELinux отключён, применяются только стандартные политики DAC.

Получить режим можно с помощью утилиты `getenforce`. Для перехода между принудительным и разрешающим режимами используется команда `setenforce`:

- `setenforce 0` – переход в разрешающий режим.
- `setenforce 1` – переход в принудительный режим.

Команда setenforce устанавливает режим только для текущей сессии, после перезагрузки результат не сохраняется.

Установите режим в enforcing. Попробуйте открыть leafpad через консоль. Затем просмотрите журналы для MAC и файлов с помощью aureport. Закройте Openbox и попытайтесь снова запустить командой startx. Что произошло? Переставьте SELinux обратно в разрешающий режим и запустите Openbox.

Основной файл конфигурации SELinux: /etc/selinux/config (рисунок 15).

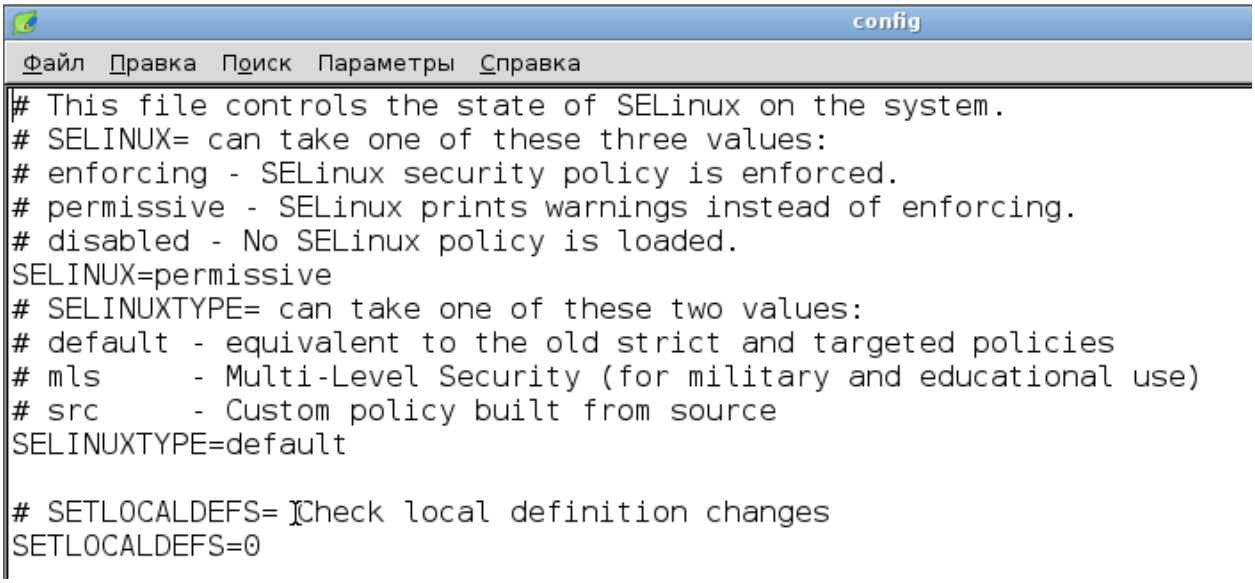


Рисунок 15 – Файл настроек SELinux

Как видно из рисунка, SELinux находится в разрешающем режиме (что подтверждалось выводами sestatus и getenforce). Для отключения SELinux значение должно быть равно «disabled».

В SELinux существуют булевы переключатели, позволяющие вносить изменения в политики SELinux без перезагрузки. Для просмотра переключателей необходимо выполнить:

```
semanage boolean -l
```

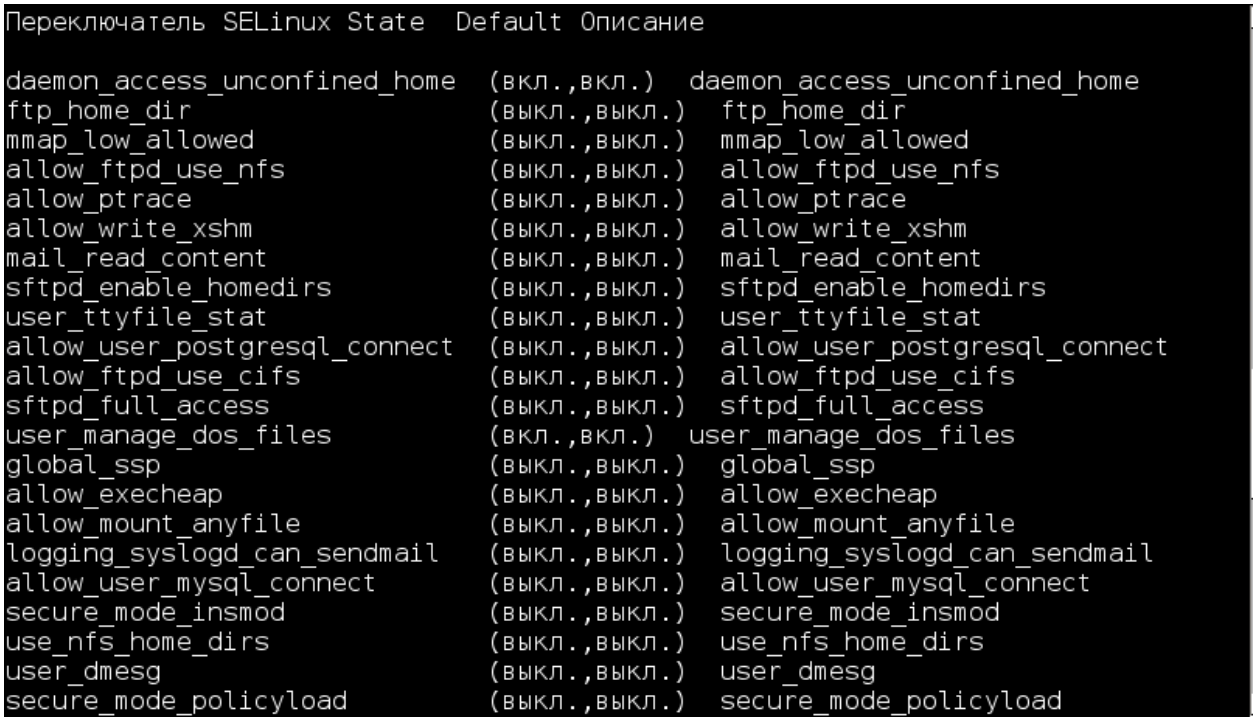


Рисунок 16 – Булевы переключатели

Получить значение переключателей (без описания) можно использовать команду `getsebool –a`. Для получения значения одного переключателя используется `getsebool` *`boolean-name`*.

Для установки переключателя используется команда `setsebool`. Установите переключатель mail_read_content:

```
setsebool mail_read_content on
```

Проверьте с помощью команды getsebool.

Установленное значение не сохранится после перезагрузки, оно действует только в течение одной сессии. Если бы было нужно установить переключатель mail_read_content для постоянного использования, команда выглядела бы так:

```
setsebool -P mail_read_content on
```

В SELinux существуют ограниченные и неограниченные пользователи. В SELinux существуют собственные пользователи, обычные пользователи соотносятся с ними, наследуя ограничения пользователей SELinux. Для просмотра пользователей выполните следующую команду:

```
semanage login –l
```

Все пользователи Linux (кроме root) по умолчанию соотносятся с логином __default__.

Предположите, чем ограниченные пользователи отличаются от неограниченных (unconfined). Также предположите, чем ограниченные процессы отличаются от неограниченных.

Важно: если пользователь относится к неограниченному домену, но выполняет приложение, которое выполняет переход в свой собственный домен, являющийся ограниченным, то пользователь будет ограничен.

Обратите внимание на то, как изменился вывод утилиты `id` после установки SELinux.

Можно сопоставлять пользователей Linux и SELinux сразу при создании пользователя, например, так:

```
useradd –Z seuser_u linuxuser
```

В данном примере создаётся пользователь Linux linuxuser, который сопоставляется пользователю SELinux seuser_u. Выполнение этой команды на использующейся в ЛР конфигурации не приведёт ни к чему, т.к. пользователя selinux_u не существует в текущей конфигурации. Попробуйте ассоциировать linuxuser с system_u. Посмотрите, как изменился вывод semanage login. Удалите пользователя linuxuser с помощью userdel. Ещё раз посмотрите вывод semanage login. Выполните `semanage login -d linuxuser`.

Для управления SELinux с помощью GUI существуют различные пакеты, в Debian можно использовать SEAdmin. В Fedora с DE часто используется sealert, отображающий предупреждения в графическом виде.

Шифрование

В Debian для шифрования томов установлена утилита `cryptsetup`. Эта утилита использует алгоритмы Cryptographic API, интегрированные ядро. Утилита доступна для Linux начиная с версии 2.6.4.

Пример использования утилиты cryptsetup:

```
cryptsetup –c aes -h sha512 -s 256 -y create sec
/dev/hdXY
```

Где:

-c aes - алгоритм шифрования;

-h sha512 - алгоритм хэширования;

-s 256 - длина ключа;

-y - запрашивать подтверждение ключевой фразы;

sec - имя устройства device mapper ;

/dev/hdXY - зашифрованный раздел или устройство.

Виды возможных угроз

В Linux обычно не употребляется слово «вирус». Существуют следующие виды угроз:

- Руткит (rootkit) – вредоносная программа, захватывающая управление учётной записью суперпользователя, тем самым может либо полностью заблокировать систему, либо шифровать присутствие злоумышленника в системе. Для поиска руткитов можно использовать утилиту chkrootkit. Этот пакет есть в стандартном репозитории Debian. Список доступного в репозитории ПО для поиска руткитов можно вывести следующей командой:
  ```
  apt-cache search rootkit
  ```
- Эксплоит (exploit) – вредоносный код, внедрённый в систему для поиска уязвимостей.
- Также существуют и другие виды уязвимостей, существующие также для Windows и OS X: «черви», ботнеты, перехватчики ввода и т.д.

Стоит отметить, что, несмотря на распространённость Linux на серверных платформах, вредоносных программ для него меньше, чем для Windows. Ряд уязвимостей, присущих Linux, являются кроссплатформенными, такими, например, как уязвимости Java или продуктов Adobe. Одно из достоинств Linux – более чёткое разграничение прав, что вытекает из его модульной архитектуры.

В Приложении 1представлен фрагмент информационного бюллетеня Jet Info. Информация, приведённая в нём, относится к 2005 году, однако приведённые в нём разоблачения мифов актуальны.

Рекомендации по улучшению безопасности Linux

Ниже приведены некоторые советы, полезные пользователям и системным администраторам Linux, помогающие улучшить безопасность.

Не работать с учётной записи root

Здесь существуют 2 опасности: опасность взлома злоумышленниками сессии (и тогда они получат доступ ко всей системе) и опасность собственной ошибки. В некоторых системах запрещено работать с учётной записи суперпользователя, для проведения административных работ используется su. В данных ЛР почти всегда использовалась учетная запись суперпользователя для удобства, в реальных условиях лучше так не делать.

Шифрование

Этот совет актуален для системных администраторов. Создавайте зашифрованные разделы, используйте безопасные каналы связи: используйте SSH, VPN, сконфигурируйте SSL. Старайтесь не использовать FTP и Telnet. Для FTP лучше использовать SFTP или FTPS, которые добавляют к FTP либо SSL, либо шифрование TLS.

Настройка сети

Linux предоставляет гибкие инструменты по управлению сетевым трафиком. Создайте фильтры, «чёрный» и «белый» списки. Во многих дистрибутивах Linux существует брандмауэр.

Незнакомые сервисы

Данный совет касается в основном администраторов. Есть 2 заблуждения относительно незнакомых сервисов в системе: «если не знаешь, что это – не трогай» и «если не знаешь, что это – выключи». Правильный подход: «если не знаешь, что это, узнай, что это такое, а потом реши, что с этим делать». Ещё одним советом, относящимся к минимизации процессов и сервисов, является минимизация количества установленного ПО. На сервере стоит держать только необходимое ПО.

Обновления

Необходимо поддерживать ядро Linux в обновлённом состоянии. Следует устанавливать последние версии ПО, в которых исправляются ошибки безопасности. Внимание: в Linux существует несколько веток ПО (stable, test, canary…), обновления лучше брать из stable. Если приложение работает корректно, стабильно и не имеет проблем с безопасностью и выполняет все поставленные перед ним задачи, возможно, не следует его обновлять сразу же после выхода новой версии: возможно, в ней ещё есть мелкие ошибки или появились уязвимости, связанные с введением новых функций.

Резервные копии

От ошибки никто не застрахован. Любая система может быть взломана, любое оборудование может быть взломано. Стоит иметь резервную копию в безопасном месте.

Использование сильных паролей

Рекомендуется использовать сильные пароли, содержащие буквы латинского алфавита в различном регистре, цифры и символы. Существуют

различные утилиты, генерирующие пароли, но можно обойтись стандартными средствами, используя urandom. Можно использовать специальные утилиты для проверки пароля на стойкость к взлому.

Политика паролей

В данной ЛР был описан файл /etc/shadow. Существует 2 способа редактирования политики паролей:

1. Правка файла /etc/shadow.
2. Использование утилиты `chage.`

Второй способ является гораздо более предпочтительным. Утилита chage позволяет установить минимальное и максимальное время работы пароля для каждого пользователя, подробнее можно узнать в странице man.

Параметр remember модуля pam_unix может запретить пользователю менять пароль на тот, который он когда-то уже использовал (количество паролей для пользователя настраивается).

Аудит

Стоит пользоваться различными демонами аудита (например, auditd) и системами автоматической генерации отчётов на основании журналов (например, `logwatch` и `logchek`). Лучше всего настроить демона аудита самостоятельно, а не использовать стандартные настройки.

Можно просматривать неудачные попытки входа в систему с помощью `faillog`.

Использование систем защиты

На сервере рекомендуется использовать SELinux или его аналог, AppArmor от компании Novell.

В некоторых дистрибутивах (например, Ubuntu и Kubuntu) существуют приложения, сохраняющие пароли (в зашифрованном виде) и позволяющие ими управлять. В Kubuntu это приложение называется KDE Wallet (KWallet), оно доступно во всех дистрибутивах с KDE SC.

Файлы без владельца

Такие файлы являются потенциальной брешью в безопасности. При обнаружении файла, не имеющего владельца, нужно либо найти его владельца и присвоить файл ему, либо удалить это файл.

Использование различных разделов диска

Рекомендуется размещать файлы пользователей и системные файлы на разных дисковых разделах. Рекомендуется монтировать на различных разделах:

- /usr
- /home
- /var и /var/tmp
- /tmp

Разделы дисков можно помечать следующим образом (о том, как это сделать, будет рассказано в ЛР, посвящённой файловым системам):

- noexec – не разрешается исполнять бинарные файлы (предотвращается исполнение бинарных файлов, но разрешается исполнять скрипты).
- nodev - не разрешается указывать посимвольные и специальные устройства (предотвращается использование файлов устройств, таких как zero, sda и т.д).
- nosuid – не разрешается иметь доступ для SUID/SGID (предотвращается изменение битов идентификатора пользователя и идентификатора группы).

Настройка IPv6

Если IPv6 не используется, его лучше отключить. Брандмауэр Linux должен быть настроен особым образом в случае работы с IPv6.

Использование централизованного сервера идентификации

Для централизованного управления учётными данными рекомендуется использовать Kerberos.

Определение прослушивающихся сетевых портов

С помощью стандартных утилит (например, `netstat`) можно определить, какие порты открыты и прослушиваются. Лучше не оставлять порты открытыми, если в этом нет необходимости.

Использование системы обнаружения вторжений

Система обнаружения сетевых вторжений (Network Intrusion Detection System, NIDS) занимается мониторингом сетевого трафика, пытаясь

обнаружить подозрительную активность. Примером такого приложения является Snort.

Обеспечение физической безопасности

Компьютер (и тем более сервер) должен находиться в физической безопасности. То же самое касается паролей.

Выводы

Даже стандартные утилиты Linux предоставляют возможности гибкой конфигурации политик безопасности. По умолчанию используется избирательное управление доступом (DAC), но SELinux и его аналоги реализует мандатное управление доступом (MAC).

Приложение 1: Фрагменты статьи «Windows и Linux: что безопаснее?»

Миф 1. Безопасность — вопрос количества: чем меньше инсталляций, тем безопаснее

Пожалуй, чаще всего повторяемый миф при сравнении безопасности Windows и Linux — утверждение, что с ОС Windows случается больше инцидентов, связанных с вирусами, сетевыми червями, троянскими программами и другими проблемами, только из-за того, что злоумышленники предпочитают вмешиваться в работу программного обеспечения, которое имеет наибольшую инсталляционную базу. Это рассуждение приводят в защиту ОС Windows и Windows-приложений. Windows преобладает на настольных компьютерах; именно поэтому на Windows и Windows-приложения направлено большинство атак, и именно поэтому не наблюдаются вирусы, сетевые черви и троянские кони для Linux. Нельзя отрицать, что до некоторой степени это верно, однако отнюдь не бесспорен вывод, который отсюда делается, а именно: Linux и Linux-приложения не являются более безопасными, чем Windows и Windows-приложения, просто Linux — слишком незначительная цель для того, чтобы тратить усилия на организацию атаки.

Это рассуждение легко опровергнуть, если учесть, что самым популярным программным обеспечением для web-серверов Интернета является Apache. Как следует из выполненного компанией Netcraft обзора web-сайтов за сентябрь 2004 г., 68% web-сайтов используют web-сервер Apache и только 21% web-сайтов используют Microsoft IIS. Если проблемы с безопасностью возникают из-за того, что злоумышленники нацеливаются на самую обширную инсталляционную базу, то должно наблюдаться больше червей, вирусов и прочих вредоносных программ, нацеленных на Apache и те операционные системы, под управлением которых он функционирует, чем на Windows и IIS. Более того, должно регистрироваться больше атак против

Apache, чем против IIS, так как из приведенного выше рассуждения следует, что проблема заключается в количествах, а не в уязвимостях.

Однако в реальности наблюдается обратная картина. В течение долгого времени IIS был главной целью для сетевых червей и других атак, и эти атаки были весьма успешными. Сетевой червь Code Red, который использовал переполнение буфера в сервисе IIS для получения контроля над web-серверами, заразил около 300 тысяч серверов, и его распространение прекратилось лишь потому, что это было предусмотрено в программном коде данного червя. Воздействие еще одного сетевого червя IISWorm оказалось ограниченным только благодаря тому, что эта программа была плохо написана, а вовсе не из-за успешной защиты IIS.

Да, известны сетевые черви для Apache, например, Slapper. (В действительности Slapper использует известную уязвимость в протоколе OpenSSL, а не в Apache). Но сетевые черви для Apache редко становятся сенсацией, поскольку их воздействие очень ограничено и они легко искореняются. На атакуемых сайтах уже были приняты меры, исключающие возможность использования ошибки в OpenSSL. Кроме того, благодаря модульной структуре Linux и UNIX было очень просто очистить и восстановить зараженные сайты. Для этого потребовалось всего несколько команд и не было необходимости в перезагрузке.

Возможно, именно поэтому в рейтинге Netcraft среди 50 web-сайтов, лидирующих по длительности непрерывной работы (промежуток времени между перезагрузками), оказалось 47, использующих Apache. Ни на одном из этих 50 web-сайтов не использовались ни Windows, ни Microsoft IIS. Поэтому, если верно предположение, что злоумышленники атакуют наиболее распространенные программные платформы, то возникает вопрос: почему хакеры так успешно взламывают программное обеспечение и операционную систему, самые популярные среди настольных компьютеров, заражают 300

тысяч серверов IIS, но неспособны нанести подобный ущерб наиболее популярному web-серверу и его операционным системам?

Читатели, посетившие web-сайт Netcraft, не могут не заметить, что все 50 серверов из списка лучших по длительности непрерывной работы используют какую-либо разновидность BSD, в основном BSD/OS. Ни один из них не функционирует под управлением Windows и ни один — под управлением Linux. Самое продолжительное время непрерывной работы среди 50 лидеров составляет 1 768 дней подряд или почти 5 лет.

В результате складывается впечатление, что BSD по своей надежности превосходит все остальные операционные системы, однако информация Netcraft, хотя и непреднамеренно, вводит в заблуждение. Netcraft определяет продолжительность непрерывной работы операционных систем по данным, которые регистрируются самими операционными системами. Linux, Solaris, HP-UX и некоторые версии FreeBSD регистрируют не более 497 дней непрерывной работы, после чего счетчики обнуляются, и отсчет начинается заново. Поэтому и кажется, что все web-сайты, размещенные на компьютерах под управлением Linux, Solaris, HP-UX и, в некоторых случаях, FreeBSD, перезагружаются каждые 497 дней, даже если они функционируют годами. При составлении обзора Netcraft ни для одной из этих операционных систем не будет зарегистрировано время непрерывной работы, превышающее 497 дней, даже если они годами функционируют без перезагрузки, так что названные системы никогда не окажутся среди 50 лидеров.

Это объясняет, почему Linux, Solaris и HP-UX не могут продемонстрировать такой же впечатляющий результат по количеству последовательных дней непрерывной работы, как BSD, даже если на самом деле эти операционные системы годами функционируют без перезагрузки. Но это не объясняет, почему среди 50 лидеров нет серверов под управлением ОС Windows. Windows не обнуляет счетчик времени непрерывной работы.

Очевидно, просто не существует web-сайтов под управлением ОС Windows, которые функционировали бы без перезагрузки достаточно долго, чтобы попасть в список 50 лучших.

Принимая во внимание казус с 497-дневным циклом обнуления становится неясным, как сравнить продолжительность непрерывной работы для Linux и Windows, основываясь на опубликованных данных Netcraft. Два результата — не статистика, но и они кое о чем говорят, особенно если один из них относится к web-сайту компании Microsoft. На сентябрь 2004 года средняя продолжительность непрерывной работы web-серверов под управлением ОС Windows, поддерживающих собственный web-сайт компании Microsoft (www.microsoft.com), составила примерно 59 дней. Максимальная продолжительность непрерывной работы для Windows Server 2003 на том же сайте — 111 дней, минимальная — 5 дней. Сравним эти результаты с данными для сайта www.linux.com (типовой сайт под управлением ОС Linux), у которого и средняя, и максимальная продолжительность непрерывной работы составляет 348 дней. Точное совпадение средней и максимальной величины означает, что эти серверы либо обнулили счетчик времени 348 дней назад, проработав до этого 497 дней, либо 348 дней назад они были перезагружены или впервые включены.

Из всего вышесказанного следует вывод, что при оценке количества успешных атак на программное обеспечение решающий фактор — это качество, а не количество.

Миф 2. Открытый код опасен по определению

Впечатляющие данные о продолжительности непрерывной работы, полученные для Apache, позволяют усомниться еще в одном популярном мифе: открытый исходный код (в котором функциональная структура

приложений является общедоступной) более опасен, чем патентованный исходный код (в котором функциональная структура является засекреченной), поскольку хакеры могут воспользоваться этим кодом для обнаружения и использования брешей.

Однако факты говорят о другом. Количество предназначенных для ОС Windows эффективных вирусов, программ-шпионов, сетевых червей, троянских и других вредоносных программ огромно, а количество компьютеров, постоянно заражаемых различными комбинациями вышеперечисленного, так велико, что вряд ли поддается исчислению. Вредоносное программное обеспечение "свирепствует" настолько, что для компрометации свежеустановленной, "непропатченной" версии Windows XP после подключения компьютера к Интернету требуется в среднем 16 минут — меньше, чем время, необходимое данной ОС для загрузки и инсталляции программных коррекций, которые позволят защитить этот компьютер.

Еще один пример. Web-сервер Apache имеет открытый код. Microsoft IIS — патентованный. В этом случае факты опровергают и миф о наибольшей популярности, и миф об опасности открытого кода. Web-сервер Apache — наиболее популярный web-сервер. Если бы оба эти мифа были правдой, следовало бы ожидать, что Apache и операционные системы, под управлением которых он функционирует, будут чаще подвергаться атакам, чем Microsoft Windows и IIS. На деле же все происходит с точностью до наоборот. Apache занимает почти монопольное положение в списке серверов с наибольшей продолжительностью непрерывной работы. Ни Microsoft Windows, ни Microsoft IIS нет в списке 50 серверов с самым продолжительным временем непрерывной работы. Очевидно, тот факт, что злоумышленники имеют доступ к открытому коду Apache, не дает им никакого преимущества в организации более успешных атак против Apache, чем против IIS.

Миф 3. Выводы на основании единственной характеристики

Слабость других популярных мифов о большей безопасности Windows по сравнению с Linux в том, что они основаны на единственном показателе — каком-нибудь одном аспекте измерения безопасности. Это верно для любого источника информации, будь то данные компетентного исследования или малообоснованная информация на уровне слухов.

Широко известно утверждение: "Для Linux зарегистрировано больше предупреждений об уязвимостях, чем для Windows, поэтому можно говорить о меньшей безопасности Linux по сравнению с Windows". Не менее распространено и такое: "Время между обнаружением бреши и созданием соответствующей программной коррекции в случае Linux больше, чем в случае Windows, поэтому можно говорить о меньшей безопасности Linux по сравнению с Windows".

Второе утверждение совершенно непостижимо. Невозможно объяснить, почему считается, что по такому показателю, как среднее время от момента обнаружения бреши до выпуска соответствующей программной коррекции, Microsoft превосходит **какую-либо** из конкурирующих операционных систем, не говоря уже о Linux. Компании Microsoft потребовалось семь месяцев (!), чтобы устранить одну из самых серьезных уязвимостей в системе защиты (Microsoft Security Bulletin MS04-007 ASN.1 Vulnerability, компания eEye Digital Security сообщила об этой задержке в информационном бюллетене AD20040210). А ведь еще имеются бреши, о которых Microsoft открыто заявляет, что они **никогда** не будут ликвидированы. В бюллетене Microsoft Security Bulletin MS03-010 об уязвимости Denial Of Service (отказ в обслуживании) в Windows NT говорится, что эта уязвимость никогда не будет устранена. Позднее компания Microsoft заявила, что не будут устраняться уязвимости в Internet Explorer для всех версий операционных систем,

выпущенных ранее Windows XP. В статистическом смысле случай с семью месяцами между обнаружением и устранением ошибки не окажет значительного влияния на среднее время реагирования при условии, что найдется достаточно много примеров чрезвычайно быстрого реагирования, которые скомпенсируют подобные аномалии, если это действительно аномалии. Но стоит включить в выборку только один случай "никогда" — и о статистическом среднем можно забыть навсегда.

Архитектура Windows и архитектура Linux

Возможно, причиной мифа о том, что Windows намного чаще подвергается атакам, чем Linux, являются вирусы, троянские кони и сетевые черви, ориентированные на электронную почту и web-навигатор. Несомненно, на настольных компьютерах инсталляций Windows больше, чем инсталляций Linux. Весьма возможно, что программное обеспечение Windows **для настольных компьютеров** чаще подвергается атакам, потому что Windows преобладает на настольных компьютерах. Но остается без ответа один важный вопрос. Действительно ли атаки на Windows столь успешны исключительно из-за их многочисленности, или же этот факт объясняется наличием проектных недоработок и неудачных проектных решений Windows?

Значительная часть (если не большинство) вирусов, троянских коней, сетевых червей и других вредоносных программ, заражающих компьютеры с Windows, используют для этого уязвимости в Microsoft Outlook и Internet Explorer. Поэтому можно поставить вопрос по-другому, включив в рассмотрение тот же тип программного обеспечения для Linux (наиболее часто используемые web-навигаторы, программы электронной почты, текстовые процессоры и т. д.): имеет ли Linux столько же уязвимостей в системе безопасности, сколько их имеет Windows?

Архитектура Windows

Вирусы, троянские кони и другие вредоносные программы поражают настольные компьютеры с Windows по целому ряду причин, свойственных Windows и чуждых Linux:

1. Windows только недавно эволюционировала от однопользовательской модели к многопользовательской.
2. Windows по своей архитектуре является монолитной, а не модульной системой.
3. В Windows слишком широко используется RPC-механизм.
4. Windows фокусируется на знакомом графическом интерфейсе для настольных компьютеров.

Архитектура Linux

По данным обзора Summer 2004 Evans Data Linux Developers Survey, 93% разработчиков Linux сталкивались не более чем с двумя случаями компро-метации компьютера с ОС Linux. 87% встретился только один подобный инцидент, а в практике 78% не было ситуаций, когда компьютер под управле-нием Linux сколько-нибудь серьезно пострадал от действий злоумышленников. В тех редких случаях, когда им удалось добиться успеха, основной причиной были ненадлежащим образом сконфигурированные настройки безопасности.

Однако для данной статьи важнее тот приведенный в обзоре факт, что 92% респондентов никогда не сталкивались со случаями заражения ОС Linux вирусами, троянскими и другими вредоносными программами.

То обстоятельство, что вирусы, троянские и другие вредоносные программы редко могут (если вообще могут) заразить Linux-системы, частично можно объяснить следующими причинами:

1. Linux имеет долгую историю использования тщательно проработанной многопользовательской архитектуры.
2. По своей архитектуре Linux является в основном модульной системой.
3. Функционирование Linux не зависит от RPC-механизма, а сервисы обычно по умолчанию настроены не использовать RPC-механизм.
4. Серверы Linux идеально подходят для удаленного администрирования.

При дальнейшем чтении следует помнить о вариациях в используемых по умолчанию конфигурациях различных дистрибутивов ОС Linux, поэтому

то, что верно для Red Hat Linux, может оказаться неправильным для Debian, и еще больше отличий может быть в SuSE. По большей части в том, что касается конфигураций по умолчанию, все основные дистрибутивы Linux следуют одним и тем же разумным правилам.

Приложение 2: Протокол RDP и систма VNC

В советах выше предлагалось использовать SSH. SSH (Secure Shell) – протокол для организации удалённого управления ОС (может использоваться и для других целей). Как правило, с помощью SSH производится подключение к удалённой консоли. В случае, если требуется подключится к удалённому рабочему столу, обычно используются два других решния: протокол RDP и систему VNC.

Протокол RDP (Remote Desktop Protocol, протокол удалённого рабочего стола) поддерживается компанией Microsoft и широко используется в её продуктах. Среди основных возможностей протокола:

- DES-шифрование;
- Подключение устройств к удалённой машине;
- Воспроизведение звука (звук перенаправляется на локальную машину);
- Поддержка общего буфера обмена.

Система VNC (Virtual Network Computing, виртуальная сетевая система) использует протокол RFB (Remote Frame Buffer, удалённый кадровый буфер). Она транслирует команды управления с локального компьютера на удалённый и результаты (в виде кадров) с удалённой машины на локальную. По умолчанию VNC не использует шифрования, для обеспечения безопасности сессию можно установить через SSL, SSH или VPN. Часто используется UNIX-системах.

Для защиты информации используется целый комплекс средств, включающий в себя технические, программно-аппаратные средства и административные меры. По мере развития средств защиты компьютерных систем развиваются и средства нападения.

Для примера рассмотрим атаки защищенной системы посредством программных закладок. *Программная закладка* – это программа или фрагмент программы, скрытно внедряемый в защищенную систему и позволяющий злоумышленнику осуществлять в несанкционированный доступ к ресурсам защищенной системы.

Основная опасность программных закладок заключается в том, что, являясь частью защищенной системы, она способна принимать активные меры по маскировке своего присутствия в системе. При внедрении в систему закладки в защищенной системе создается скрытый канал информационного обмена, который, как правило, остается незамеченным для администраторов системы в течение длительного времени. Практически все известные программные закладки, применявшиеся в разное время различными злоумышленниками, были выявлены либо из-за ошибок, допущенных при программировании закладки, либо чисто случайно.

Наиболее распространенный класс программных закладок – закладки, перехватывающие пароли пользователей операционных систем (*перехватчики паролей*). Перехватчики паролей были разработаны в разное время для целого ряда операционных систем, включая многие версии Windows и UNIX. Перехватчик паролей, внедренный в ОС, получает доступ к паролям, вводимым пользователями при входе в систему. Перехватив очередной пароль, закладка записывает его в специальный файл или в любое другое место, доступное злоумышленнику, внедрившему закладку в систему.

Перехватчики паролей ***первого рода*** действуют по следующему алгоритму. Злоумышленник запускает программу, которая имитирует приглашение пользователю для входа в систему и ждет ввода. Когда пользователь вводит имя и пароль, закладка сохраняет их в доступном злоумышленнику месте, после чего завершает работу и осуществляет выход из системы пользователя-злоумышленника (в большинстве операционных систем выход пользователя из системы можно осуществить программно). По окончании работы закладки на экране появляется настоящее приглашение для входа пользователя в систему.

Пользователь, ставший жертвой закладки, видит, что он не вошел в систему, и что ему снова предлагается ввести имя и пароль. Пользователь предполагает, что при вводе пароля произошла ошибка, и вводит имя и пароль повторно. После этого пользователь входит в систему, и дальнейшая его работа протекает нормально. Некоторые закладки, функционирующие по данной схеме, перед завершением работы выдают на экран правдоподобное сообщение об ошибке, например: “Пароль введен неправильно. Попробуйте еще раз”.

Основным достоинством этого класса перехватчиков паролей является то, что написание подобной программной закладки не требует от злоумышленника никакой специальной квалификации. Любой пользователь, умеющий программировать хотя бы на языке BASIC, может написать такую программу за считанные часы.

Перехватчики паролей первого рода представляют наибольшую опасность для тех операционных систем, в которых приглашение пользователю на вход имеет очень простой вид. Например:

```
login: user

password:
```

Задача создания программы, подделывающей такое приглашение, тривиальна.

Усложнение внешнего вида приглашения на вход в систему несколько затрудняет решение задачи перехвата паролей, однако не создает для злоумышленника никаких принципиальных трудностей. Для того, чтобы существенно затруднить внедрение в систему перехватчиков паролей первого рода, необходимы более сложные меры защиты. Примером операционной системы, где такие меры реализованы, является Windows NT.

В Windows NT обычная работа пользователя и аутентификация пользователя при входе в систему осуществляются на разных *рабочих полях* (desktops). Рабочее поле Windows NT представляет собой совокупность окон, одновременно видимых на экране. Только процессы, окна которых расположены на одном рабочем поле, могут взаимодействовать между собой, используя средства Windows GUI. Понятие рабочего поля Windows NT близко к понятию терминала UNIX.

Процесс Winlogon, получающий от пользователя имя и пароль, выполняется на отдельном рабочем поле (*рабочем поле аутентификации*). Никакой другой процесс, в том числе и перехватчик паролей, не имеет доступа к этому рабочему полю. Поэтому приглашение пользователю на вход в систему, выводимое перехватчиком паролей первого рода, может располагаться только на рабочем поле прикладных программ, где выполняются все программ, запущенные пользователем.

Переключение экрана компьютера с одного рабочего поля на другое производится при нажатии комбинации клавиш Ctrl-Alt-Del. Win32-подсистема Windows NT обрабатывает эту комбинацию по-особому – сообщение о нажатии Ctrl-Alt-Del посылается только процессу Winlogon. Для всех других процессов, в частности, для всех прикладных программ, запущенных пользователем, нажатие этой комбинации клавиш совершенно незаметно.

При старте системы на экран компьютера вначале отображается рабочее поле аутентификации. Однако пользователь вводит имя и пароль не сразу, а только после нажатия Ctrl-Alt-Del. Когда пользователь завершает сеанс работы с системой, на экран также выводится рабочее поле аутентификации, и, так же как и в предыдущем случае, новый пользователь может ввести пароль для входа в систему только после нажатия Ctrl-Alt-Del.

Если в систему внедрен перехватчик паролей первого рода, то, для того, чтобы он смог перехватить пароль пользователя, он должен по крайней мере обработать нажатие пользователем Ctrl-Alt-Del. В противном случае при нажатии пользователем этой комбинации клавиш произойдет переключение на рабочее поле аутентификации, рабочее поле прикладных программ станет неактивным, и перехватчик паролей просто не сможет ничего перехватить – сообщения о нажатии пользователем клавиш будут приходить на другое рабочее поле. Однако для всех прикладных программ факт нажатия пользователем Ctrl-Alt-Del всегда остается незамеченным. Поэтому пароль будет воспринят не программной закладкой, а процессом Winlogon.

Для обеспечения надежной защиты от перехватчиков паролей первого рода необходимо два условия:

1. Программа, получающая от пользователя имя и пароль при входе в систему, выполняется на изолированном терминале (*терминале аутентификации*), недоступном прикладным программам.

2. Факт переключения пользовательской консоли на терминал аутентификации незаметен прикладным программам. Прикладные программы не могут запретить переключение консоли на терминал аутентификации.

Перехватчики паролей ***второго рода*** перехватывают все данные, вводимые пользователем с клавиатуры. Простейшие программные закладки данного типа просто сбрасывают все эти данные на жесткий диск компьютера или в любое другое место, доступное злоумышленнику. Более совершенные закладки анализируют перехваченные данные и отсеивают информацию, заведомо не имеющую отношения к паролям.

Такие закладки представляют собой резидентные программы, перехватывающие одно или несколько прерываний процессора, имеющих отношение к работе с клавиатурой.

Информация о нажатой клавише и введенном символе, возвращаемая этими прерываниями, используется закладками для своих целей.

В конце 1997 года на хакерских серверах Internet появились перехватчики паролей второго рода для Windows 3.x и Windows 95.

Программные интерфейсы Win16 и Win32 поддерживают специальный механизм фильтров (hooks), который может быть использован для перехвата паролей пользователей. С помощью этого механизма прикладные программы и сама операционная система решают целый ряд задач, в том числе и задачу поддержки национальных раскладок клавиатуры. Любой русификатор клавиатуры, работающий в среде Windows, перехватывает всю информацию, вводимую пользователем с клавиатуры, в том числе и пароли. Несложно написать русификатор так, чтобы он, помимо своих основных функций, выполнял бы и функции перехватчика паролей. Написание программы локализации клавиатуры является достаточно простой задачей.

Также Windows поддерживает цепочки фильтров, с помощью которых несколько программ могут одновременно получать доступ к информации, вводимой с клавиатуры, и обрабатывать ее так, как считают нужным, при необходимости передавая обработанную информацию дальше по цепочке. Можно встроить перехватчик паролей в цепочку фильтров перед русификатором или после него, так, что вся информация, вводимая пользователем с клавиатуры, проходит и через русификатор, и через перехватчик паролей.

В большинстве случаев верно следующее утверждение. *Если ОС допускает переключение раскладки клавиатуры при вводе пароля, то для этой ОС можно написать перехватчик паролей второго рода*. Действительно, если для операционной системы существует программа локализации раскладки клавиатуры, и если эта программа используется при вводе пароля, после незначительного изменения исходного текста эта программа превращается в перехватчик паролей второго рода.

Для организации защиты от перехватчиков паролей второго рода необходимо добиться выполнения в ОС следующих трех условий:

1. Переключение раскладки клавиатуры в процессе ввода пароля невозможно. В противном случае задача создания перехватчика паролей второго рода существенно упрощается.

2. Конфигурирование цепочки программных модулей, участвующих в получении операционной системой пароля пользователя, доступно только администраторам системы.

3. Доступ на запись к файлам программных модулей, участвующих в получении пароля пользователя, не предоставляется никому. Доступ на запись к атрибутам защиты этих файлов предоставляется только администраторам. Любые обращения с целью записи к этим файлам, а также к их атрибутам защиты, регистрируются в системном журнале аудита.

Для выполнения перечисленных условия выполнялись, необходимо, чтобы подсистема защиты операционной системы поддерживала разграничение доступа и аудит.

К перехватчикам паролей ***третьего рода*** относятся программные закладки, полностью или частично подменяющие собой подсистему аутентификации операционной системы. Поскольку задача создания такой программной закладки гораздо сложнее, чем задача создания перехватчика паролей первого или второго рода, этот класс программных закладок появился недавно.

Перехватчик паролей третьего рода может быть написан для любой многопользовательской операционной системы. Сложность создания такого перехватчика паролей зависит от сложности алгоритмов, реализуемых подсистемой аутентификации, сложности интерфейса между ее отдельными модулями, а также от степени документированности подсистемы аутентификации операционной системы. В целом задача создания перехватчика паролей третьего рода гораздо сложнее задачи создания перехватчика паролей первого или второго рода.

Поскольку перехватчики паролей третьего рода частично берут на себя функции подсистемы защиты операционной системы, перехватчик паролей третьего рода при внедрении в систему должен выполнить по крайней мере одно из следующих действий:

- подменить собой один или несколько системных файлов;

- внедриться в один или несколько системных файлов по одному из “вирусных” алгоритмов;

- использовать поддерживаемые операционной системой интерфейсные связи между программными модулями подсистемы защиты для встраивания себя в цепочку программных модулей, обрабатывающих введенный пользователем пароль;

- использовать для той же цели низкоуровневые интерфейсные связи операционной системы, используемые подсистемой защиты для решения своих задач.

Каждое из этих действий оставляет в операционной системе следы, которые могут быть выявлены с помощью следующих мер защиты:

1. Соблюдение адекватной политики безопасности. Подсистема аутентификации должна быть самым защищенным местом операционной системы. Меры, необходимые для поддержания адекватной политики безопасности, сильно различаются для разных операционных систем.

2. Контроль целостности исполняемых файлов операционной системы. Необходимо контролировать не только файлы, входящие в состав подсистемы защиты, но и библиотеки, содержащие низкоуровневые функции операционной системы.

3. Контроль целостности интерфейсных связей внутри подсистемы защиты, а также интерфейсных связей, используемых подсистемой защиты для решения низкоуровневых задач.

Построение абсолютно надежной защиты против перехватчиков паролей третьего рода представляется невозможным. Поскольку машинный код перехватчиков паролей третьего рода выполняется не в контексте пользователя, а в контексте операционной системы, перехватчик паролей третьего рода может принимать меры, затрудняющие его обнаружение администраторами системы, в частности:

- перехват системных вызовов, которые могут использоваться администраторами для выявления программной закладки в целях подмены возвращаемой информации;

- фильтрация регистрируемых сообщений аудита.

Для того чтобы защита от перехватчиков паролей была эффективной, политика безопасности, принятая в системе, должна удовлетворять следующим требованиям:

1. Конфигурирование цепочки программных модулей, участвующих в получении операционной системой пароля пользователя, доступно только администраторам системы.

2. Доступ на запись к файлам этих программных модулей предоставляется только администраторам системы.

3. Конфигурирование подсистемы аутентификации доступно только администраторам системы.

4. Доступ на запись к системным файлам предоставляется только администраторам системы.

Приложение 3. Общие вопросы безопасности Windows.

Операционная система Windows 2000 обладает развитыми средствами шифрования данных с *открытым ключом* (public key), представляющими собой дальнейшее развитие служб шифрования информации Windows NT, входящих в состав более ранних версий операционной системы.

Windows 2000 располагает интегрированным набором служб и инструментов администрирования, предназначенных для создания, реализации и управления приложениями, использующими алгоритмы шифрования с *открытым* ключом. Это позволит независимым разработчикам программного обеспечения применять в своих продуктах технологию *общего ключа* (shared key).

Применение алгоритмов шифрования с открытым ключом.

Криптография - это наука о защите данных. Алгоритмы криптографии математическими методами комбинируют входной открытый текст и ключ шифрования, генерируя, таким образом, зашифрованные данные. Применяя хороший алгоритм шифрования, можно сделать практически невозможным с точки зрения необходимых вычислительных и временных ресурсов, взлом защиты и получения открытого текста подбором ключа. Для быстрого выполнения подобного преобразования необходим *расшифровывающий ключ*. В традиционном шифровании с секретным ключом (симметричном шифровании) зашифровывающий и расшифровывающий ключи совпадают. Стороны, обменивающиеся зашифрованными данными, должны знать секретный ключ. Процесс обмена информацией о секретном ключе является узким местом в вопросах безопасности.

Фундаментальное отличие шифрования с открытым ключом заключается в том, что зашифровывающий и расшифровывающий ключи не совпадают. Шифрование информации является односторонним процессом:

открытые данные шифруются с помощью зашифровывающего ключа, однако с помощью того же ключа нельзя осуществить обратное преобразование и опять получить открытые данные. Для этого необходимо применить расшифровывающий ключ, который *связан* с зашифровывающим ключом, но *не совпадает* с ним. Подобная технология шифрования предполагает, что каждый пользователь имеет в своем распоряжении пару ключей - открытый ключ и личный ключ. Открыто распространяя открытый ключ, вы даете возможность другим пользователям посылать вам зашифрованные данные, которые могут быть расшифрованы с помощью известного только вам личного ключа. Аналогично с помощью личного ключа вы можете преобразовать данные так, чтобы другая сторона убедилась в том, что информация пришла именно от вас. Эта возможность применяется при работе с электронными подписями.

Появление пары "личный ключ/открытый ключ" привело к возникновению нескольких новых технологий. Наиболее важными из которых являются электронные подписи, распределенная аутентификация, соглашение о секретном ключе и шифрование больших объемов данных без предварительного соглашения о секретном ключе.

Существует несколько хорошо известных алгоритмов шифрования с открытым ключом. Некоторые из них, например RSA (Rivest-Shamir-Adelman) и шифрование с помощью эллиптической кривой (Elliptic Curve Criptography, ECC), являются алгоритмами общего употребления в том смысле, что они поддерживают все упомянутые выше операции. Другие алгоритмы поддерживают только некоторые операции. К ним относятся: алгоритм электронной подписи (Digital Signature Algorithm, DSA), используемый только для работы с электронными подписями, и алгоритм Diffie-Hellman (D-H), применяемый только для соглашения о секретных ключах.

Электронные подписи.

Наиболее ярким проявлением всех преимуществ шифрования с открытым ключом является технология *электронных подписей.* Она основана на математическом преобразовании, комбинирующим данные с секретным ключом таким образом, что:

- Только владелец секретного ключа может создать электронную подпись.
- Любой пользователь, обладающий соответствующим открытым ключом, может проверить истинность электронной подписи.
- Любая модификация подписанных данных (даже изменение одного бита) делает неверной электронную подпись.

Электронные подписи представляют собой данные, которые могут быть переданы вместе с подписанной информацией. Кроме того, цифровая подпись позволяет проверить, что данные не были изменены при передаче.

Шифрование с открытым ключом применяется для создания надежной службы *распределенной аутентификации*, гарантирующей, что данные пришли получателю от истинного корреспондента.

Шифрование с открытым ключом позволяет двум сторонам, используя открытый ключ в незащищенной сети, договориться о секретном ключе. Обе стороны посылают друг другу половины секретного ключа, зашифрованных соответствующими открытыми ключами. Каждая из сторон получает возможность расшифровать полученную половину секретного ключа и, соединив ее со своей половиной ключа, получить весь секретный ключ.

Поскольку алгоритмы шифрования с открытым ключом требуют значительно больших, по сравнению с алгоритмами секретного ключа, вычислительных ресурсов, они плохо подходят для шифрования больших объемов данных. Поэтому существует технология, комбинирующая оба алгоритма. В соответствии с ней весь объем данных шифруется с помощью секретного ключа, который в свою очередь шифруется с открытым ключом и посылается корреспонденту вместе с зашифрованными данными. Приемная сторона расшифровывает секретный ключ с помощью своего личного ключа, а затем расшифровывает данные с помощью полученного секретного ключа.

При шифровании с открытым ключом жизненно важным является абсолютно достоверная ассоциация открытого ключа и передавшей его стороны, поскольку в обратном случае возможна подмена открытого ключа и осуществление несанкционированного доступа к передаваемым зашифрованным данным. Необходим механизм, гарантирующий достоверность корреспондента. Одним из таких механизмов является применение сертификата, созданного авторизованным генератором сертификатов.

Сертификат - это средство, позволяющее гарантированно установить связь между переданным открытым ключом и передавшей его стороной, владеющей соответствующим личным ключом. Сам сертификат представляет собой набор данных, закрытых цифровой подписью. Информация сертификата подтверждает истинность открытого ключа и владельца соответствующего личного ключа.

Обычно сертификаты содержат дополнительную информацию, позволяющую идентифицировать владельца личного ключа, соответствующего данному открытому ключу. Сертификат должен быть подписан авторизованным генератором сертификатов.

Наиболее распространенным на данный момент стандартом сертификатов является ITU-T X.509. Это фундаментальная технология, используемая в Windows 2000. Однако это не единственная форма сертификатов.

Поставщик сертификатов, или центр авторизации (Certificate authority - CA) это организация или служба, создающая сертификаты. CA выступает в качестве гаранта истинности связи между открытым ключом субъекта и идентифицирующей этот субъект информацией, содержащейся в сертификате. Различные CA могут применять для проверки связи различные средства, поэтому перед выбором достойного доверия CA важно хорошо понять политику данного CA и применяемые им процедуры проверки.

После получения подписанного сообщения встает вопрос: насколько данной подписи можно доверять? Действительно ли подпись была поставлена тем, кого она представляет? Математическую верность подписи можно проверить после получении подписанного сообщения. Для этого применяется открытый ключ. Но при этом нет полной уверенности в том, что используемый открытый ключ действительно принадлежит корреспонденту, от которого получено подписанное сообщение. Возникает необходимость проверки принадлежности открытого ключа. Она может быть проведена с помощью сертификата, созданного центром авторизации, пользующимся доверием у стороны, получившей подписанное сообщение. В сертификате должна содержаться следующая информация:

- Криптографически верная подпись, идентифицирующая создателя сертификата;
- Подтверждение связи между стороной, приславшей подписанное сообщение, и ее открытым ключом;
- Сертификат должен быть создан CA, которому приемная сторона доверяет.
- Истинность полученного сертификата может быть проверена с открытым ключом, принадлежащим создавшему этот сертификат CA. Однако здесь возникает еще одна проблема. А действительно ли открытый ключ принадлежит данному CA? Для получения ответа на этот вопрос необходимо применить еще один сертификат. В результате возникает цепочка сертификатов, начинающаяся с сертификата открытого ключа стороны, передавшей подписанное сообщение, и заканчивающаяся сертификатом CA, которому приемная сторона безоговорочно доверяет. Такой сертификат называется *корневым сертификатом доверия* (trusted root certificate), поскольку он формирует *корень* (верхний узел) *иерархии открытых ключей и идентификаторов связи*, которую приемная сторона рассматривает как достойную доверия. Если приемная сторона определила CA, которому

она будет безоговорочно доверять, то полным доверием будут пользоваться и все дочерние CA, идентифицированные корневым сертификатом доверия.

Описанная выше модель сертификации истинности передающей стороны предполагает, что секретно приемная сторона должна получить только информацию о наборе корневых сертификатов доверия.

Компоненты Windows 2000 для шифрования с открытым ключом.

На рис. 1 показана взаимосвязь средств Windows 2000, позволяющих применять шифрование с открытым ключом.

Рисунок 1 – Схема взаимосвязи средств шифрования.

Изображенные средства могут находиться на одном компьютере и эффективно работать. Ключевым звеном всей схемы является *служба сертификатов Microsoftt* (Microsoft Certificate Services). Она позволяет создать один или несколько CA предприятия, поддерживающих создание и

отзыв сертификатов. Они интегрированы в Active Directory, где хранится информация о политике CA и их местоположении. Кроме того, с помощью Active Directory выполняется публикация информации о сертификатах и их отзыве.

Средства работы с открытым ключом не заменяют существующих механизмов доверительных отношений между доменами и аутентификации, основанных на контроллерах доменов и центра распространения ключей Kerberos (Key *Oistriubution Center*, KOC). Напротив, данные средства взаимодействуют с этими службами, что позволяет приложениям безопасно передавать конфиденциальную информацию по Internet и корпоративным глобальным каналам

Поддержка прикладных средств шифрования информации с открытым ключом включена в состав программного обеспечения операционных систем Windows 2000/NT/95/98. На рис. 2 показана структура служб, предназначенных для поддержки прикладных программ.

Рисунок 2 – Службы средств шифрования информации с открытым ключом, поддерживающие прикладные программы

Основанием архитектуры прикладных средств шифрования информации с открытым ключом является библиотека Crypto API. Она позволяет работать со всеми устанавливаемыми провайдерами услуг шифрования (Cryptographic Service Providers, CSP) через стандартный интерфейс. Услуги CSP могут быть реализованы на программном уровне или с помощью специального оборудования. Они поддерживают различные длины ключей и алгоритмы шифрования. Как видно на рис. 2, один из CSP поддерживает смарт-карты. Услугами служб шифрования пользуются службы управления сертификатами. Они соответствуют стандарту X.509 v3 и позволяют организовывать принудительное хранение, службу подсчета и поддержку расшифровывания. Кроме того, эти службы предназначены для работы с различными промышленными стандартами сообщений. В основном они поддерживают стандарты PKCS и разработанный IETF (Internet Ellgil1eering Task Force) еще сырой набор стандартов РЮХ (Public Key Infrastructure, X.509).

Остальные службы используют Crypto API для придания дополнительной функциональности прикладным программам. Защищенный канал (Secure Channel) поддерживает сетевую аутентификацию и шифрование в соответствии со стандартными протоколами TLS и SSL, обращение к которым может быть выполнено с помощью интерфейсов Microsoft WinInet и SSPI. Служба Autl1enticode предназначена для выполнения проверки и подписи объектов, что в основном используется при получении информации через Internet.

В состав программного обеспечения служб поддержки прикладных средств шифрования входит поддержка интерфейса, предназначенного для работы со смарт-картами общего назначения. Они используются при программно независимой интеграции криптографических смарт-карт для выполнения регистрации в компьютере и сети Window 2000.

Политика безопасности шифрования с открытым ключом.

Политики безопасности действуют в масштабах области (site) каталога Active Directory, домена или организационной единицы и влияют на все ассоциированные с этими объектами группы, компьютеры и пользователей. Безопасность шифрования с открытым ключом является одним из аспектов общей политики безопасности Windows 2000 и интегрирована в ее структуру. Это механизм, с помощью которого можно посредством объектов политики безопасности централизованно осуществлять настройку и управление глобальной политикой работы с открытым ключом.

С помощью политики открытого ключа можно определять следующие аспекты безопасности Windows 2000:

- Доверенные корни CA;
- Регистрация и обновление сертификатов;
- Регистрация в системе с помощью смарт-карты.

Групповые политики безопасности.

В предыдущих версиях Windows NT Server *политика безопасности домена* хранилась в базе данных Диспетчера безопасности учетных записей SAM (Security Account Manager). Политика состояла из *дескриптора безопасности*, предоставляющего доступ к выполнению операций (таких, например, как создание учетной записи и просмотр учетных записей) и *свойств*, описывающих политики в отношении паролей и блокировки учетных записей пользователей. *Локальная политика* хранилась в базе данных политик и состояла из информации о привилегиях пользователей и конфигурации аудита. Она реплицировалась между контроллерами домена, поэтому все контроллеры получали одинаковые настройки аудита и привилегий. Политика домена действовала в отношении всего домена, но не могла быть обшей для нескольких доменов.

Реализация политик безопасности в Windows 2000 предоставляет значительно более широкие возможности. В случае необходимости вы

можете устанавливать политики для всего дерева доменов. Различные контроллеры в пределах одного домена могут обладать индивидуальными политиками безопасности. Установив политику безопасности в одном месте, администраторы могут контролировать безопасность всех серверов и рабочих станций домена.

Групповая политика имеет следующие преимущества:

- Основываясь на службе Active Directory системы Windows 2000, позволяет выполнять как централизованное, так и децентрализованное управление параметров политики.
- Обладает гибкостью и масштабируемостью. Применяется в широком наборе конфигураций системы, предназначенных как для малого бизнеса, так и для больших корпораций.
- Дает возможность использовать простой интегрированный инструмент управления политикой. Редактор групповой политики представляет собой оснастку консоли управления Microsoft, расширяющую такие инструменты администрирования, как, например, Active Directory Manager и Cотpl1ter Managemel1t. Администраторы могут делегировать право управления объектами групповой политики.
- Редактор групповой политики обладает простым и хорошо понятным интерфейсом.
- Обладает высокой степенью надежности и безопасности.

Групповые политики расширяют и используют преимущества Active Directory. Их настройки находятся в объектах групповых политик, которые в свою очередь ассоциируются с такими контейнерами Active Directory, как области, домены и организационные единицы.

Политики безопасности Windows 2000 хранятся в двух типах объектов политик каталога: *объекты локальной "политики безопасности* и *объекты политики безопасности домена*.

В объекте политики безопасности, определяющем безопасность домена или организационной единицы, хранится следующая информация:

- **Политика пароля** - устанавливает политику паролей для учетных записей домена.
- **Политика блокировки учетных записей** - включает или отключает политику блокировки учетных записей.
- **Дескриптор безопасности для операций масштаба домена** – устанавливает безопасность для определенного объекта политики домена. Определяет, кто может изменять настройки свойств объекта политики безопасности домена.
- **Политика регистрации Kerberos** (обновление билета).
- **Политика восстановления данных EFS.**
- **Ссылка на назначенный объект политики безопасности компьютера**

В домене могут быть созданы свои собственные объекты политик безопасности или храниться реплика и ссылка на политику безопасности другого домена в дереве. Если политика реплицируется из другого домена, репликацией должен управлять сервер глобального каталога. Ссылка на объект политики безопасности домена хранится в объекте "домен".

В объекте локальной политики безопасности хранится следующая информация:

- **Политика аудита** - включает и отключает локальный аудит. Устанавливает, какие события должны быть подвергнуты аудиту.
- **Политика привилегий** - определяет, какие права и разрешения получают учетные записи.
- **Локальные роли** - определяет, какие роли будут играть такие пользователи и группы, как Administrators, Account Operators или Print Operators.
- **Разрешения и настройки аудита** для объекта политики безопасности компьютера. Устанавливают безопасность для объекта политики безопасности компьютера, выбираемой по умолчанию. Эти настройки

определяют, кто может изменять свойства объекта безопасности компьютера.

Отдельно стоящая машина или машина, работающая в составе рабочей группы, получает локальную политику безопасности. Компьютер, присоединенный к домену, получает сначала локальную политику, а затем политику, определенную в масштабах домена. Это значит, что политика безопасности масштаба домена имеет приоритет над локальной политикой безопасности. Каждая политика безопасности представляет собой *объект групповой политики* (Group Policy Object, GPO). В основном GPO используются для определения политики безопасности домена. Однако на каждом компьютере находится один GPO, необходимый для задания локальной политики безопасности.

Объекты групповой политики, используемые для формирования безопасности домена, объединяются в *скелет групповой политики* (Group Policy Framework, GPF). Сюда включаются следующие политики безопасности:

- **Политики учетных записей** *(Account* Policies). Позволяют настраивать политики безопасности учетных записей (как в масштабах домена, так и локальных учетных записей). Здесь определяются политика паролей, политика блокировки паролей и новая политика Kerberos, распространяющаяся на весь домен.
- **Локальные политики** (Local Policies). Позволяют настраивать политику аудита, назначать права пользователей и различные параметры безопасности, которые могут быть настроены в системе Windows 2000.
- **Ограниченное членство в группах** (Restricted *Groups).* Такие политики позволяют настраивать членство в специфических группах. Сюда обычно включают встроенные группы, такие как Administrators, Backup Operators и другие, имеющие по умолчанию права администратора. В эту категорию могут быть включены и другие группы, безопасность которых требует

особого внимания и членство в которых должно регулироваться на уровне политики.

- **Системные службы** (System Services). Позволяют настраивать безопасность и параметры загрузки для работающих на компьютере служб. В этом разделе могут быть использованы расширения, с помощью которых можно осуществлять настройку безопасности, специфическую для данной службы. Например, расширение File File Sharing Service позволяет настраивать политику безопасности для службы создания общего доступа к файлу (активизация подписи SMB, ограничение анонимного доступа к общим ресурсам, формирование безопасности различных сетевых общих ресурсов и т.д.).
- **Реестр** (Registry). Позволяет .настраивать безопасность различных ключей. реестра.
- **Файловая система** (File System). Позволяет настраивать безопасность определенных файлов.
- **Политики открытых ключей (**Public Key Policies). Позволяют настраивать политики безопасности в отношении шифрования информации с помощью EFS, авторизации корневого сертификата в масштабах домена, авторизации доверенного сертификата и т.д.
- **Политика безопасности IP** (IPSEC). Позволяет настраивать политику безопасности **IP** для компьютеров, находящихся в определенной области действия.
- **Политики безопасности, определяемые GPO из структуры групповой политики**. Действуют на компьютеры и частично на пользователей. Все политики, о которых говорилось выше, действуют только на компьютеры. Даже политики учетных записей распространяются на весь компьютер, хотя они влияют на учетные записи.

Поскольку политика безопасности Windows 2000 Server значительно отличается от политик предыдущих версий Windows NT, при переходе к Windows 2000 низкоуровневые политики безопасности не переносятся. Если

при переходе создается новое дерево доменов, одновременно создается и новая политика безопасности, назначаемая по умолчанию. Если при переходе домен присоединяется к уже существующему дереву, политика безопасности берется от родительского домена.

Отличие локальных учетных записей от доменных.

В каждом компьютере Windows 2000, являющемся клиентом или отдельно стоящей машиной, создаются локальные учетные записи и группы. Они отличаются от аналогичных сетевых объектов, поддерживаемых контроллерами доменов Windows NT. Отличие заключается в том, что учетная запись, созданная в сетевом домене, позволяет регистрироваться в сети с помощью любого компьютера, присоединенного к домену. С другой стороны, локальные учетные записи могут быть использованы для регистрации только на том компьютере, где они находятся. Большинство политик безопасности учетных записей (например, политика паролей) действует на все учетные записи, включая и локальные. Исключение составляют политики безопасности Kerberos. Они доступны только в контексте домена Windows NT.

Локальные политики всегда действуют на индивидуальные машины, независимо от того, является данная машина контроллером домена или нет. Они могут быть установлены независимо от политик самого домена.

Инструменты настройки безопасности.

Обеспечение эффективной безопасности системы имеет несколько аспектов. Во-первых, операционная система должна соответствовать базовым требованиям к безопасности. Например, требования к системе, соответствующей уровню безопасности C2, включают защиту памяти, дискреционное управление доступом и наличие средств аудита.

Во-вторых, для создания эффективной безопасности следует создать инструмент, позволяющий управлять всеми средствами обеспечения безопасности, заложенными в операционной системе. В Windows 2000 для

управления безопасностью системы при меняется Security Configuration Tool Set (Набор инструментов настройки безопасности). В него включены параметры безопасности операционной системы, собранные в единый блок управления, и ряд программных инструментов, позволяющих управлять этими параметрами.

Набор инструментов настройки безопасности состоит из следующих компонентов:

- *Security Configuration Service* (Служба настройки безопасности - реализована в виде оснастки MMC) - служба, являющаяся ядром Security Configuration Tool Set. Она работает на любой машине и отвечает за выполнение функций, связанных с настройкой безопасности и ее анализом. Эта служба является центральной для всей инфраструктуры безопасности системы.
- *Setup Security* (Безопасность установки) - первоначальная конфигурация безопасности, создаваемая при установке Windows 2000 с помощью заранее определенного шаблона, поставляемого вместе с системой. На каждом компьютере Windows 2000 формируется первоначальная база данных безопасности, называемая "Локальная политика компьютера" (Local Computer Policy).
- *Security Configuration Editor* (Редактор настройки безопасности) - этот инструмент позволяет определять конфигурации безопасности, не зависящие от машины, которые хранятся в виде текстовых файлов.
- *Security Configuration Manager* (Диспетчер настройки безопасности) - этот инструмент позволяет импортировать одну или несколько хранящихся конфигураций безопасности в базу данных безопасности (это может быть база данных Локальной политики компьютера или любая другая личная база). Импорт конфигураций создает специфическую для машины базу данных безопасности, которая хранит композитную настройку. Ее можно активизировать на компьютере и проанализировать состояние текущей

конфигурации безопасности по отношению к композитной настройке, хранящейся в базе данных.

- *Security Settings Extension to Group Policy Editor* (Расширение Редактора групповой политики) - эта программа расширяет Редактор групповой политики и позволяет определять конфигурацию безопасности как часть объекта групповой политики.

Приложение 4. Безопасность одиночного компьютера.

Управление локальными политиками безопасности.

В операционной системе Windows 2000 реализована новая концепция управления политиками безопасности компьютера. Каждый компьютер является элементом безопасности, обладающим правами доступа и локальной политикой безопасности. Одновременно с доменом создается и объект политики безопасности компьютера, который впоследствии может быть отредактирован. Кроме того, вы можете создать другой объект безопасности компьютера, имеющий необходимые свойства. Политика безопасности может быть создана для каждой функциональной категории компьютеров (например, для контроллеров домена, серверов приложений, серверов печати и др.). Свой объект политики могут иметь отдельные компьютеры домена. Каждый компьютер может иметь только *одну политику безопасности,* "ассоциированную с ним в данный момент времени. В Windows 2000 управление политикой безопасности компьютеров домена дает администраторам возможность лучше контролировать компьютеры, входящие в состав домена. Отдельно стоящие компьютеры и машины, находящиеся в составе рабочих групп, обладают локальным *объектом групповой политики (*Group Policy Object, GPO), определяющим их локальную политику. Он может быть отредактирован аналогично любому другому GPO каталога. Работа политик безопасности компьютера подчиняется следующим принципам:

- При включении компьютера в домен по умолчанию используется политика безопасности компьютера, определенная локально. Однако администратор домена имеет возможность изменить ее и заставить все компьютеры домена использовать политику безопасности компьютера, сконфигурированную в масштабе домена. Это позволяет получить больший контроль над политиками безопасности, используемыми различными машинами, входящих в состав домена.

- Если администратор хочет, чтобы компьютер, входящий в состав домена, использовал политику безопасности, сконфигурированную в масштабе домена, он должен создать в домене объект "компьютер".

Для того чтобы компьютер использовал политику безопасности компьютера, сконфигурированную в масштабе всего домена необходимо:

- Запустите оснастку Active Directory Manager.
- Установите указатель мыши на имени компьютера, который должен использовать политику масштаба домена, и нажмите правую кнопку. В появившемся меню выберите команду **Properties.**
- В окне свойств компьютера перейдите на вкладку **General** (Общие) и просмотрите информацию в поле **Computer security policy** (Политика безопасности компьютера). По умолчанию должно быть установлено значение **locally defined security policy** (Политика безопасности, определенная локально).
- Для выбора другой политики безопасности нажмите кнопку **Change** (Изменить).
- Появится окно диалога **Change Computer Security Policy** (Изменение политики безопасности компьютера). Возможен один из следующих вариантов:

- Использовать политику безопасности домена, установленную по умолчанию.

- Использовать другую политику безопасности, существующую в данном домене. Для выбора нужного объекта политики безопасности компьютера нажмите кнопку **Browse** (Обзор). Этот вариант следует применять, если вам необходимо, чтобы определенный набор компьютеров домена использовал одинаковые политики безопасности, которые отличаются от политики безопасности домена, установленной по умолчанию. Например, вы можете сконфигурировать все контроллеры домена с политикой

безопасности, которая будет отличаться от политики безопасности рабочих станций, входящих в состав домена.

- Использовать локальную политику безопасности. В этом случае выполняется эмуляция работы политики безопасности предыдущих версий Windows NT. Локальные администраторы компьютеров смогут самостоятельно определять и редактировать политики безопасности компьютеров. Этот вариант устанавливается в компьютерах по умолчанию.
- Нажмите кнопку OK**.**

В следующих разделах приведены некоторые примеры применения политик безопасности компьютеров.

Блокирование локальных учетных записей.

После нескольких неудачных попыток регистрации в системе учетная запись пользователя должна быть заблокирована. Вы можете установить допустимое максимальное количество неудачных попыток. Следует заметить, что разблокировать учетную запись может только администратор.

Для модификации локальной политики учетных записей:

1. Установите указатель мыши на значке My Computer и нажмите правую кнопку. В появившемся контекстном меню выберите команду Manage (Управление) . Запустится оснастка Computer Management.
2. В разделе System Tools (Системные инструменты) выберите Group Policy Editor (GPE).
3. В GPE выберите раздел Security Settings (Настройки безопасности), в нем выберите Account Policies (Политики учетных записей) и внутри этого раздела выберите Account Lockout Policy (Политика блокировки пароля). Появится окно, показанное на рис. 17.2, в котором будет отображен набор настроек политики блокировки.

4. Для запуска средства редактирования сделайте двойной щелчок на разделе, настройку которого вы хотите изменить, например, Account lockout count.
5. В появившемся окне введите с клавиатуры новое значение и снимите флажок Exclude this setting from configuration (Исключить эту настройку из конфигурации).
6. Нажмите кнопку OK. в результате будет установлено новое значение параметра.
7. Сделайте двойной щелчок на разделе Lockout account for (Блокировать учетную запись для). В появившемся окне установите длительность блокировки учетной записи (от 0 до бесконечности (Forever». Нажмите кнопку OK. (Как и в предыдущем случае следует снять флажок Exclude this setting from configuration.).
8. Для сохранения сделанных изменений завершите работу программы Computer Management.

Настройка безопасности для ключа реестра.

Ниже показано, как можно настроить безопасность для ключа реестра HKEY_LOCAL_MACHINE\Software\Microsoft\ Windows NT Администраторы получат полный контроль, а все остальные аутентифицированные пользователи будут иметь доступ только на чтение.

Для настройки безопасности для ключа реестра:

- Запустите оснастку Compiter Management и выберите Group Policy Editor.
- В GPE укажите мышью на раздел Security Settings, выберите папку Registry (Реестр). В правом подокне будут отображены ключи реестра, на которые распространяется политика безопасности, установленная по умолчанию.

- Установите указатель мыши на папке **Registry** и нажмите правую кнопку. В появившемся меню выберите команду **Add Key** (Добавить ключ). Запустится браузер реестра.
- Перейдите к **Machine\Software\Microsoft\ Windows NT** и выберите **Windows NT**. (В качестве альтернативы можно ввести полный путь к редактируемому полю.).
- Нажмите кнопку **OK.** Появится окно диалога, с помощью которого можно настроить безопасность выбранного ключа.
- Для изменения списка объектов, имеющих разрешение доступа к данному ключу, следует использовать кнопки Add (Добавить) и Remove (Удалить). Здесь вы сможете задать полный доступ только для администраторов и доступ только на чтение для остальных пользователей. После завершения корректировки данных в окне Configuration Security for (Конфигурация безопасности для) нажмите кнопку OK.
- При настройке безопасности ключа реестра можно задать три типа действия безопасности:
    - Наследовать (Inberit) - безопасность, установленная для данного ключа реестра, будет автоматически наследоваться ниже по дереву. Это значит, что все напрямую заданные права доступа к подключам будут сохранены. К ним будут добавлены унаследованные разрешения доступа.
    - Перезаписать (Overwrite) - безопасность, установленная для данного ключа реестра, будет также автоматически наследоваться ниже по дереву. Но в данном случае любые напрямую заданные права доступа к подключам будут перезаписаны новыми установками. Действовать будет только вновь созданная безопасность.
    - Игнорировать (Ignore) - это исключение из предыдущего случая. Например, для родительского ключа установлена

новая безопасность. Она наследуется всеми подключами и перезаписывает все остальные прямые установки, но один из подключей может быть установлен таким образом, что настройки безопасности родительского ключа будут игнорироваться. В этом случае на подключ и дерево под ним не будут распространяться настройки безопасности родительского ключа.

- Закройте программы настройки. Все сделанные вами изменения в настройках будут сохранены и начнут работать в локальной политике безопасности.

Настройка безопасности для диска.

С помощью политики безопасности вы сможете установить различные права доступа к устройству для разных пользователей (например, полный контроль для администраторов и права на чтение и создание подкаталогов, а также полный контроль над создаваемыми файлами).

Для настройки безопасности устройства:

1. Запустите оснастку Computer Management и выберите Group Policy Editor.
2. В GPE укажите мышью на раздел Security Settings, выберите. папку FileSystem (Файловая система).
3. Установите указатель мыши на папке File System, щелкните правой кнопкой и выберите команду Add Folder (Добавить папку). Запустится браузер файловой системы.
4. В окне диалога Browse for Folder (Обзор папки) выберите Local Disk (C:) и нажмите кнопку OK.
5. В появившемся окне диалога File and Registry Object Configuration (Конфигурация объекта файловой системы или реестра) выберите тип действия безопасности и

нажмите кнопку Edit Security (Редактировать безопасность).

6. В появившемся окне диалога Access Control Settings for (Настройка управления доступом для) с помощью кнопки Remove можно очистить список пользователей. С помощью кнопки Add можно добавить в поле Permissions (Разрешения) следующие объекты:
    - *Администраторы и система* получают полный доступ, который распространяется на все подкаталоги.
    - *Аутентифицированные пользователи* получают права Traverse Folder (Проходить сквозь папку), List Folder (Просматривать папку), Read Attributes (Читать атрибуты), Read Extended Attributes (Читать расширенные атрибуты), Read Permissions (Читать права доступа) и Create Files and Folders (Создавать файлы и папки). Они должны распространяться только на данную папку. Эта настройка не должна наследоваться.
    - *Создатель/владелец* получает полный контроль. Это позволит пользователям иметь полный доступ к тем объектам файловой системы, которые они создали, и ни к чему больше. В соответствии с предыдущей настройкой пользователи могут создавать подкаталоги, теперь они получают над ними полный контроль.

7. Окно диалога, с помощью которого можно настраивать безопасность папки, дает возможность задать следующие типы действия безопасности: наследовать, перезаписать и игнорировать.

Шифрующая файловая система EFS.

Для обеспечения безопасности данных на персональных компьютерах рекомендуется загружать операционную систему не с жесткого, а с гибкого диска. Это позволит избежать отказов в работе жесткого диска и разрушения загрузочных разделов. К сожалению, с помощью гибкого диска можно загрузить несколько различных операционных систем, поэтому любой пользователь, получивший физический доступ к компьютеру, может обойти встроенную систему управления доступом файловой системы Windows 2000 (NTFS) и с помощью определенных инструментов' прочесть информацию жесткого диска. Многие конфигурации оборудования позволяют применять пароли, регулирующие доступ при загрузке. Однако такие средства не имеют широкого распространения. Кроме того, в случае, если на компьютере работает несколько пользователей, подобный подход не дает хороших результатов, да и сама защита с помощью пароля недостаточно надежна. Ниже рассмотрены типичные примеры того, как может быть осуществлен несанкционированный доступ к данным:

- *Хищение компьютера типа Laptop.* Любой злоумышленник может незаметно и быстро похитить компьютер типа Laptop, а затем получить доступ к конфиденциальной информации, находящейся на его жестком диске.
- *Неограниченный доступ.* Компьютер оставлен в рабочем состоянии и за ним никто не наблюдает. Любой пользователь может подойти к такому компьютеру и получить доступ к конфиденциальной информации.

- Основной целью создания системы безопасности является зашита конфиденциальной информации, которая обычно находится в незащищенных файлах на жестком диске, от несанкционированного доступа. Доступ к данным можно ограничить с помощью средств NTFS. Такой подход обеспечивает хорошую степень защиты, если единственной загружаемой операционной системой является Windows 2000, жесткий диск не может быть физически удален из компьютера, и данные находятся в разделе NTFS. Если кто-либо захочет получить доступ к данным, он может осуществить свое желание, получив физический доступ к компьютеру или жесткому диску. Существуют *инструменты, позволяющие получить доступ к файлам, находящимся в разделе NTFS*, из операционных систем MS-DOS или UNIX в обход системы безопасности NTFS.

Из приведенных выше соображений следует вывод: единственно надежный способ зашиты информации - это шифрующая файловая система. На рынке программного обеспечения существует целый набор продуктов, обеспечивающих шифрование данных с помощью образованного от пароля ключа на уровне приложений. Однако такой подход имеет ряд ограничений:

- **Ручное шифрование и расшифровывание.** Службы шифрования большинства продуктов не прозрачны для пользователей. Пользователю приходится расшифровывать файл перед каждым его использованием, а затем опять зашифровывать. Если пользователь забывает зашифровать файл после окончания работы с ним, информация остается незащищенной. Поскольку каждый раз необходимо указывать, какой файл должен быть зашифрован (и расшифрован), применение такого метода защиты информации сильно затруднено.
- **Утечка информации из временных файлов и файлов подкачки.** Практически все приложения в процессе редактирования

документов создают временные файлы. Они остаются на диске незашифрованными несмотря на то, что оригинальный файл -зашифрован. Кроме того, шифрование информации на уровне приложений выполняется в режиме пользователя Windows 2000. Это значит, что ключ, применяющийся для такого типа шифрования, может храниться в файле подкачки. В результате, с помощью изучения данных файла подкачки можно получить ключ и расшифровать все документы пользователя.

- **Слабая криптостойкость ключей.** Ключи образуются от паролей или случайных фраз. Поэтому в случае, если пароль был легко запоминаемым, атаки с помощью словарей могут легко привести к взлому системы защиты.
- **Невозможность восстановления данных.** Большинство продуктов, позволяющих шифровать информацию, не предоставляют средств восстановления данных, что является дополнительным поводом для пользователей не применять средства шифрования. Это особенно касается тех работников, которые не хотят запоминать дополнительный пароль. С другой стороны, средство восстановления данных с помощью пароля, представляет собой еще одну брешь в системе защиты информации. Все, что необходимо злоумышленнику, - это пароль, предназначенный для запуска механизма восстановления данных, который позволит получить доступ к зашифрованным файлам.

Все перечисленные выше проблемы позволяет решить *Шифрующая файловая система* (Encrypting File System, EFS), реализованная в Windows 2000 и работающая только на NTFS 5.0. Следующие разделы содержат детальное описание технологии шифрования, места шифрования в операционной системе, взаимодействия с пользователями и способа восстановления данных.

Технология шифрования EFS.

Система EFS основана на шифровании с открытым ключом и использует все возможности архитектуры CryptoAPI в Windows 2000. Каждый файл шифруется с помощью случайно сгенерированного ключа, зависящего от пары открытого (public) и закрытого (secret) ключей пользователя. Подобный подход в значительной степени затрудняет осуществление большого набора атак, основанных на *криптоанализе.* При криптозащите файлов может быть использован любой алгоритм симметричного шифрования. Текущая версия EFS использует алгоритм DES. В дальнейшем, вероятно, появится возможность применения альтернативных алгоритмов. EFS позволяет осуществлять шифрование и дешифрование файлов, находящихся на удаленных файловых серверах. (в данном случае EFS может работать только с файлами, находящимися на диске. Шифрующая файловая система не осуществляет криптозащиту данных, передаваемых по сети. Для шифрования передаваемой информации в операционной системе Windows 2000 следует использовать специальные сетевые протоколы, например SSL/PCT.

Система EFS хорошо интегрирована в NTFS. При создании временных файлов они наследуют атрибуты оригинальных файлов (если файлы находятся в разделе NTFS). При шифровании файла шифруются также и его временные копии. EFS находится в ядре Windows 2000 и использует для хранения ключей специальный пул, невыгружаемый на жесткий диск. Поэтому ключи никогда не попадают в файл подкачки.

Взаимодействие с пользователем.

Конфигурация EFS, устанавливаемая по умолчанию, позволяет пользователю шифровать свои файлы без всякого вмешательства со стороны администратора. В этом случае EFS автоматически генерирует для пользователя пару открытого и закрытого ключей, применяемую для криптозащиты данных.

Шифрование и дешифрование файлов может быть выполнено как для определенных файлов, так и для целого каталога. Криптозащита каталога прозрачна для пользователя. Все файлы и подкаталоги, созданные в

зашифровываемом каталоге, шифруются автоматически. Каждый файл обладает уникальным ключом, позволяющим легко выполнять операцию переименования. Если вы переименовываете файл, находящийся в зашифрованном каталоге, и переносите его в незашифрованный каталог, сам файл остается зашифрованным. Службы шифрования и дешифрования доступны через Windows NT Explorer. Кроме того, можно полностью использовать все доступные средства криптозащиты данных с помощью набора утилит командной строки и интерфейсов администрирования.

Система EFS исключает необходимость предварительного расшифровывания данных при доступе к ним. Операции шифрования и дешифрования выполняются автоматически при записи или считывании информации. EFS автоматически распознает зашифрованный файл и найдет соответствующий ключ пользователя в системном хранилище ключей. Поскольку *механизм хранения ключей* основан на использовании CryptoAPI, пользователи получают возможность хранить ключи на защищенных устройствах, например смарткартах.

Текущая версия EFS не позволяет создавать общие ресурсы. Однако ее архитектура предполагает возможность создания общих ресурсов, к которым пользователи могут обращаться с помощью их общих ключей. Впоследствии каждый пользователь может независимо расшифровать файл с использованием его закрытого ключа. Пользователи могут быть легко добавлены (если они имеют сконфигурированную пару ключей) или удалены из группы, имеющей разрешение доступа к общему ресурсу.

Система EFS располагает встроенными средствами восстановления данных. Инфраструктура системы безопасности Windows 2000 позволяет создать необходимые для этого ключи. Шифрование файлов может быть использовано только в случае, если в системе создан один или несколько ключей восстановления информации. EFS позволяет *агентам восстановления данных* создавать открытые ключи, которые затем используются при восстановлении файлов. Используя ключ восстановления,

можно получить только сгенерированный случайным образом ключ, с помощью которого был зашифрован конкретный файл. Поэтому агенту восстановления не может случайно стать доступной другая конфиденциальная информация.

Средство восстановления данных предназначено для использования в разнообразных конфигурациях вычислительных сред. Оно применяется в случае, например, когда организации необходимо получить доступ к информации, зашифрованной уволенным пользователем, или когда ключ, с помощью которого данные были зашифрованы, утерян. Политика восстановления может быть определена на контроллере домена Windows 2000. В этом случае она распространяется на все компьютеры домена и находится под контролем администраторов домена, которые с помощью службы Active Directory могут делегировать право управления политикой восстановления файлов учетным записям администраторов безопасности. Это позволяет получить возможность следить за кругом лиц, имеющих право восстанавливать зашифрованные файлы, а также гибко предоставлять пользователям это право или лишать его. Кроме того, с помощью создания нескольких конфигураций ключей восстановления EFS позволяет одновременно применять несколько агентов, что делает процедуры восстановления значительно более устойчивыми и гибкими.

Система EFS может быть использована и в домашних условиях. Она автоматически генерирует ключи восстановления в отсутствие домена Windows 2000 и хранит их как ключи машины. Для восстановления данных в домашних условиях пользователи могут использовать утилиты командной строки и учетную запись администратора.

Конфигурация ключей пользователей.

Пользователи могут генерировать, экспортировать, импортировать открытые ключи, используемые в EFS для шифрования, а также управлять ими. Конфигурация интегрирована с другими настройками безопасности, устанавливаемыми пользователем. Этот пункт предназначен для опытных

пользователей, которые хотят иметь средство управления собственными ключами. Обычно пользователям не приходится самостоятельно выполнять конфигурацию, поскольку EFS автоматически генерирует ключи шифрования файлов.

Утилита cipher.

Эта утилита командной строки позволяет шифровать и расшифровывать файлы. Описание ее ключей приведено ниже:

/E - Шифрует указанные в качестве параметра файлы. Каталоги помечаются как зашифрованные, все файлы, которые будут помещены в них впоследствии, шифруются автоматически.

/D - Расшифровывает все указанные после ключа файлы. Каталоги помечаются как незашифрованные, все файлы, которые будут помещены в них впоследствии, шифроваться не будут.

/R - Расшифровывает файлы с помощью информации восстановления.

/S - Выполняет определенную ключом операцию над файлами и всеми подкаталогами, находящимися в данном каталоге.

/I - Продолжает выполнение указанной операции даже после возникновения ошибочной ситуации. По умолчанию при появлении ошибки программа cipher останавливается.

/F - Осуществляет принудительное шифрование всех файлов, указанных после ключа, даже если они уже зашифрованы. По умолчанию уже зашифрованные файлы не подвергаются вторичному шифрованию.

/Q - Выдает только краткую информацию.

/P - Устанавливает политику восстановления для данной машины или домена. Если существует файл ключа, cipher использует его содержимое для установки политики. Если файл ключа отсутствует, политика восстановления генерируется автоматически, а результат записывается в файл ключа/

/K - Сохраняет ключи EFS пользователя в заданном файле ключа.

/L - Загружает ключи, находящиеся в заданном файле ключа, в текущий контекст.

Параметр *filename* может быть маской, файлом или каталогом. При вводе команды cipher без параметров, она выдает информацию о том, зашифрован ли данный каталог или файлы, находящиеся в нем. Если параметр *filename* присутствует, то имен файлов может быть и несколько. Между собой параметры должны быть разделены пробелом.

Для того чтобы зашифровать каталог My *Documents* введите команду: c:\cipher /E "My Documents"

Для того чтобы зашифровать все файлы, имена которых содержат набор символов "cndfl", введите команду: c:\cipher /E /S *cndfl*

Команда *copy* Команда copy расширена. Она получила дополнительные ключи, позволяющие импортировать и экспортировать зашифрованные файлы в промежуточный формат. Синтаксис команды copy имеет следующий вид:

copy [/E 1/1] *sourcefile destinationfile*

Новые ключи команды Copy:

/E - Экспортирует зашифрованный файл *(sourcefile)* в виде шифрованного потока битов в файл *destinationfile.* Файл *destinationfile* может не быть томом NTFS, но может быть файлом FAT на гибком диске.

/I - Импортирует шифрованный поток битов из файла *sourcefile* в файл, зашифрованный с помощью EFS на томе NTFS. Файл *sourcefile* может ненаходиться на томе NTFS, но файл *destinationfile* должен находиться на томе NTFS.

(Перечисленные ключи были доступны в Windows 2000 Beta 1)

Шифрование файлов.

Чтобы зашифровать файлы необходимо выбрать их в Windows NT Explorer, а затем установить на выбранном файле указатель мыши и нажать правую кнопку. Появится контекстное меню, в котором нужно выбрать команду **Encrypt** (Шифровать) и EFS зашифрует выбранные файлы.

После завершения шифрования файлы сохраняются на диске. Теперь при выполнении операций чтения и записи файлы будут расшифровываться

и зашифровываться автоматически. Для того, чтобы узнать, зашифрован ли файл, пользователь может просмотреть свойства файла и выяснить установлен ли флаг Encrypted.

Поскольку шифрование выполняется автоматически, пользователь может работать с файлом так же, как и до установки его криптозащиты. редактировать его, как и прежде. Все остальные пользователи, которые попытаются получить доступ к зашифрованному файлу, получат сообщение об ошибке доступа, поскольку они не владеют необходимым секретным ключом, позволяющим им расшифровать файл. Заново данные шифруются только в случае, если файл копируется в зашифрованный каталог.

С помощью контекстного меню Explorer пользователи могут шифровать не только файлы, но и целые каталоги. Если сделать каталог зашифрованным, это значит, что все находящиеся в нем или появившиеся впоследствии файлы и подкаталоги, будут подвергаться шифрованию по умолчанию. Список файлов каталога не шифруется, поэтому при наличии соответствующих разрешений доступа он может быть просмотрен обычным способом.

Операция шифрования каталогов аналогична шифрованию файлов - просто выделите определенный каталог, а затем выберите в контекстном меню Windows NT Explorer команду **Encrypt**. В этом случае каталог помечается как зашифрованный, а все файлы и подкаталоги, находящиеся в нем шифруются. Создавая зашифрованные каталоги и копируя туда файлы, содержащие конфиденциальную информацию, пользователи могут осуществлять криптозащиту данных.

Дешифрование файлов и каталогов.

При выполнении обычных операций пользователю не нужно заботиться о дешифровании файлов, поскольку EFS автоматически шифрует и дешифрует данные при их записи и чтении. Однако при организации общего доступа к файлам возникает необходимость принудительного дешифрования данных.

Дешифровать файл или каталог можно с помощью контекстного меню Windows NT Explorer. Эта операция практически идентична операции шифрования, за исключением выбираемой команды - **Decrypt** (Дешифровать) вместо **Encrypt.** В результате выполнения этой операции EFS дешифрует все выбранные файлы и пометит их незашифрованными. В случае снятия криптозащиты с каталога контекстное меню позволяет рекурсивно дешифровать все находящиеся в каталоге файлы и подкаталоги.

Операция восстановления.

По умолчанию на всех компьютерах, принадлежащих к домену, установлена *локальная система безопасности.* В этом случае пользователи могут сразу же работать с EFS. Однако, если несколько компьютеров объединены в домен, на них устанавливается *система безопасности домена* (индивидуальная или выбираемая по умолчанию). В домене пользователи не смогут применять EFS до тех пор, пока не будет установлена политика восстановления, которая представляет собой элемент либо политики безопасности домена Windows 2000, либо политики безопасности локального компьютера для отдельно стоящего сервера или рабочей станции. В масштабе домена, политика восстановления распространяется на все компьютеры домена. Интерфейс пользователя политики EFS входит в состав общего интерфейса настройки политики домена или локальной политики. Он дает возможность пользователю при помощи средств управления ключами генерировать, экспортировать, импортировать и архивировать *ключи восстановления.* Интеграция политики восстановления в общую политику безопасности позволяет создать последовательную и хорошо понятную модель обеспечения защиты информации. Подсистема безопасности Windows 2000 обеспечивает настройку, репликацию и кэширование политики EFS. Поэтому пользователи имеют возможность использовать шифрование на компьютере, временно отключенном от общей вычислительной системы, например, на портативном компьютере, а также могут регистрироваться в

системе с помощью своей учетной записи, используя кэшированную информацию входа.

Восстановить файл, секретная часть ключа которого утеряна, можно с помощью утилиты командной строки cipher:

- Если файл находится на компьютере домена, то в профиль агента восстановления необходимо загрузить ключи восстановления. Для этого с клавиатуры введите следующую команду: cipher /1: <*.имя файла ПОЛИТИКИ восстановления, устанавливаемой* в *домене*>
- Если файл находится на отдельно стоящем компьютере, то для расшифрования файла с помощью ключей восстановления, загруженных в профиль агента восстановления, необходимо ввести команду: cipher /d <Путь к восстанавливаемому файлу>

Восстановление данных, зашифрованных с помощью неизвестного закрытого ключа.

Необходимо, чтобы политика восстановления была установлена на уровне домена (или локально, если машина не является членом домена) *до* использования EFS. Политику восстановления устанавливают администраторы домена или пользователи *(агенты восстановления,* которым делегированы соответствующие права), контролирующие ключи восстановления всех машин домена.

Если пользователь забыл закрытый ключ, файл, зашифрованный с помощью такого ключа, может быть восстановлен после его экспортирования и пересылки по электронной почте одному из агентов восстановления, который импортирует файл на защищенную машину, где находится закрытый ключ восстановления. С помощью полученного ключа и с использованием соответствующей утилиты командной строки файл может быть дешифрован и возвращен пользователю. В небольших вычислительных системах или домашних условиях, где домен отсутствует, восстановление может быть выполнено на самом отдельно стоящем компьютере.

Приложение 5. Безопасность компьютеров в сети.

Создание объектов политики безопасности в Active Directory.

Каждый домен должен обладать ассоциированной с ним *политикой безопасности.* Объект групповой политики безопасности (Group Policy Object, GPO) автоматически создается при создании домена. Впоследствии этот установленный по умолчанию объект может быть отредактирован. Используя политики безопасности, можно одновременно управлять поведением всех компьютеров в домене. Кроме того, можно легко проконтролировать, *кто* имеет право их администрировать. В каждом домене должна существовать *только одна* политика безопасности.

При формировании безопасности сети предприятия, очень важно *правильно организовать* политики безопасности, что позволит минимизировать избыточность параметров и оптимизировать управление сетью. К сожалению, эти цели находятся в некотором противоречии друг с другом. Для минимизации избыточности объекты GPO должны с максимальной точностью описывать каждую из политик, действующих на конкретные домены и организационные единицы (OU), что приводит к увеличению числа GPO. Для улучшения управления сетью следует, наоборот, создавать небольшое число GPO. Поэтому в каждом конкретном случае необходимо правильно сбалансировать количество объектов политик безопасности.

Для правильного планирования структуры политик безопасности сети следует сделать следующее:

- Разделить политики на логические группы. Например, политики учетных записей могут быть объединены в одну группу.
- Для каждой логической группы создать один или несколько объектов GPO, 4имеющих различные настройки политик безопасности.
- Распределить объекты-компьютеры по иерархическим древовидным структурам, состоящие из организационных единиц.

В качестве критерия при таком распределении следует выбрать функцию, которую несет данный компьютер.

В основном, каждая OU должна иметь определенную политику безопасности, действующую на все находящиеся в ней компьютеры. Зачастую это сделать непросто, поскольку OU могут определять географическое расположение организации, а также и иерархию управления сетью.

В случаях, когда политики безопасности должны распространяться на подмножество компьютеров организации, можно сделать следующее:

- Создать в различных OU организации внутренние OU, переместить туда нужные объекты-компьютеры и назначить внутренним OU собственную политику безопасности.
- Если вы не хотите создавать внутренние OU, можно использовать схему фильтрации GPO, основанную на разрешениях доступа. С ее помощью вы сможете задать соответствие компьютеров и действующих на них GPO.

Протокол аутентификации Kerberos.

Протокол Kerberos представляет собой набор методов идентификации и проверки истинности партнеров по обмену информацией (рабочих станций, пользователей или серверов) в открытой (незащищенной) сети. Процесс идентификации *не зависит* от аутентификации, выполняемой сетевой операционной системой, *не основывается* в принятии решений на адресах хостов и *не предполагает* обязательную организацию физической безопасности всех хостов сети. Кроме того, допускается, что пакеты информации, передаваемые посети, могут быть модифицированы, прочитаны и переданы в любой момент времени. Следует, однако, отметить, что большинство приложений использует функции протокола Kerberos только при создании сеансов передачи потоков информации. В этом случае предполагается, что последующее несанкционированное разрушение потока

данных невозможно. Поэтому применяется прямое доверие, основанное на адресе хоста. Kerberos выполняет аутентификацию как доверенная служба третьей стороны, используя криптографию с помощью общего секретного ключа (shared secret key).

Аутентификация выполняется следующим образом:

- Клиент посылает запрос серверу аутентификации (Authentication server , AS) на информацию, однозначно идентифицирующую определенный сервер.
- AS передает требуемую информацию, зашифрованную с помощью известного пользователю ключа. Переданная информация состоит из "билета" сервера и временного ключа, предназначенного для шифрования (часто называемого *ключом сеанса).*
- Клиент пересылает серверу билет, содержащий идентификатор клиента и ключ сеанса, зашифрованные с помощью ключа, известного серверу.
- Теперь ключ сеанса известен клиенту и серверу. Он может быть использован для аутентификации клиента, а также для аутентификации сервера. Он может быть применен для шифрования передаваемой в сеансе информации или для взаимного обмена ключами подсеанса, предназначенными для шифрования последующей передаваемой информации.

Работа Kerberos основана на одном или нескольких серверах аутентификации, работающих на физически защищенном хосте. Серверы аутентификации ведут базы данных партнеров по обмену информацией в сети (пользователей, серверов и т.д.) И их секретных ключей. Программный код, осуществляющий функционирование самого протокола и процесса шифрования данных, находится в специальных библиотеках. Для того чтобы выполнять аутентификацию Kerberos для своих транзакций, приложения должны сделать несколько обращений к библиотекам Kerberos. Процесс

аутентификации состоит из обмена необходимыми сообщениями с сервером аутентификации Kerberos.

Протокол Kerberos состоит из нескольких субпротоколов (или обменов сообщениями). Существует два метода, которыми клиент может запросить у сервера Kerberos информацию, идентифицирующую определенный сервер. Первый способ предполагает, что клиент посылает AS простой текстовый запрос билета для конкретного сервера, а в ответ получает данные, зашифрованные с помощью своего секретного ключа. Как правило, в данном случае клиент посылает запрос на *билет, позволяющий получить билет* (Ticket granting ticket, TGT), который в дальнейшем используется для работы с *выдающим билеты сервером* (Ticket granting Server, TGS). Второй способ предполагает, что клиент посылает TGT на TGS так же, как будто он обменивается информацией с другим сервером приложений, требующим аутентификации Kerberos.

Информация, идентифицирующая сервер, может быть использована для идентификации партнеров по транзакции, что позволит гарантировать целостность передаваемых между ними сообщений или сохранить в секрете передаваемую информацию.

Для идентификации партнеров по транзакции клиент посылает билет на сервер. Поскольку посылаемый билет "открыт" (некоторые его части зашифрованы, но они не помешают выполнить посылку копии) и может быть перехвачен и использован злоумышленником, ДЛЯ подтверждения истинности партнера, пославшего билет, передается дополнительная информация, называемая *а у т е н т и ф и к а т о р о м.* Она зашифрована с помощью ключа сеанса и содержит отсчет времени, подтверждающий, что сообщение было сгенерировано недавно и не является копией оригинальной посылки. Шифрование аутентификатора с помощью ключа сеанса доказывает, что информация была передана истинным партнером по обмену данными. Поскольку кроме запрашивающего партнера и сервера никто

больше не знает ключа сеанса (он никогда не посылается по сети в открытом виде), с его помощью можно полностью гарантировать истинность партнера.

Целостность сообщений, которыми обмениваются партнеры, гарантируется с помощью ключа сеанса (передается в билете и содержится в информации идентификации партнера). Этот подход позволяет обнаружить атаки типа посылки злоумышленником перехваченной копии запроса и модификации потока данных. Это достигается генерированием и пересылкой *контрольной суммы* (хэш-функции) сообщения клиента, зашифрованной с помощью ключа сеанса. Безопасность и целостность сообщений, которыми обмениваются партнеры, может быть обеспечена шифрованием передаваемых данных с помощью ключа сеанса, передаваемого в билете и содержащегося в информации идентификации партнера.

Описанная выше аутентификация требует доступа на чтение к базе данных Kerberos. Однако иногда записи базы данных могут быть модифицированы. Это происходит, например, при добавлении новых партнеров по обмену информацией или при изменении секретного ключа партнера. Изменения базы данных выполняются с помощью специального протокола обмена между клиентом и сервером Kerberos, применяющимся и при поддержке нескольких копий баз данных Kerberos.

Взаимодействие с удаленными владениями.

Протокол Kerberos может работать вне пределов одной компании. Клиент данной организации может быть аутентифицирован на сервере, находящемся в другой организации. Каждое предприятие, желающее при менять в своей сети Kerberos, должно установить границы своего "владения" (realm). Имя владения, в котором зарегистрирован данный пользователь, составляет часть его имени и может быть использовано конечной службой для принятия решения, должен ли данный запрос быть удовлетворен.

Установив общие ключи для владений, администраторы могут позволить клиентам различных владений выполнять *удаленную аутентификацию.* Конечно, обладая необходимыми разрешениями, клиент может выполнить

регистрацию партнера, имеющего не связанное с данным владением имя, и установить нормальный обмен сообщениями. Однако даже при незначительном количестве удаленных регистраций такой подход приводит к большим неудобствам, поэтому рекомендуется применять более автоматизированные методы. При обмене общими во владениях ключами (inter realm keys) (при передаче информации в каждом направлении может быть использован отдельный ключ) службы выдачи билетов регистрируются в противоположном владении в качестве партнера по обмену данными. После этого клиент имеет возможность получать от локальной службы выдачи билетов TGT для такой же службы, находящейся в удаленном владении. При использовании этого TGT для его дешифрования удаленная служба выдачи билетов применяет общий для владений ключ, который, как правило, отличается от ключа TGS. Это гарантирует, что данный TGT был передан собственным TGS клиента. Билеты, посланные удаленной службой выдачи билетов, укажут конечной службе, что аутентификация клиента была выполнена в удаленном владении.

Одно владение может обмениваться информацией с другим владением, если оба они обладают общим ключом, или если локальная служба выдачи билетов обладает общим ключом с промежуточным владением, в свою очередь обладающим общим ключом с целевым владением. Путь аутентификации представляет собой последовательность промежуточных владений, каждое из которых может обмениваться информацией со своими соседями.

Как правило, владения организованы в иерархическую структуру. Каждое владение обладает общим ключом со своим родителем и отдельным общим ключом с каждым дочерним владением. Если между двумя владениями не существует общего ключа, иерархическая структура позволяет легко установить путь аутентификации. Если иерархическая структура владений не используется, для обнаружения пути аутентификации следует применять базу данных.

Как правило, владения организованы в иерархическую структуру, поэтому аутентификацию можно осуществить, воспользовавшись альтернативными путями, минуя промежуточные владения. Это может позволить оптимизировать обмен данными между двумя владениями. Конечной службе важно знать, через какие владения проходит путь аутентификации, поскольку от этого зависит достоверность всего процесса аутентификации. Для облегчения принятия такого решения каждый билет содержит поле, где хранятся имена всех владений, которые составляют путь аутентификации.

Требования к рабочему окружению.

Протокол Kerberos налагает несколько требований на свое рабочее окружение, в котором он может эффективно работать:

- Kerberos не противодействует атакам типа "Отказ в обслуживании". Ряд особенностей протокола Kerberos позволяют злоумышленнику заставить приложение не принимать участие в процессе аутентификации. Обнаружение и борьба с таким типом атак (некоторые из которых могут проявляться В установлении необычных режимов работы программного обеспечения), как правило, лучше всего осуществляются администраторами систем.
- Партнеры по обмену данными должны хранить свои секретные ключи в надежном месте. Если злоумышленник каким-либо образом похитит секретный ключ, он сможет выдать себя за одного из партнеров или имперсонализировать сервер для законного клиента.
- Kerberos не противодействует атакам типа "Подбор пароля". Если пользователь устанавливает легко угадываемый пароль, злоумышленник, вполне вероятно, сможет его узнать подбором с применением словаря.

- Каждый хост сети должен иметь часы, которые приблизительно синхронизируются с часами других хостов. Синхронизация необходима, чтобы было легче обнаружить факт передачи копии заранее перехваченного сообщения. Степень приблизительности синхронизации может быть установлен индивидуально для каждого сервера. Сам протокол синхронизации серверов сети должен быть защищен от атак злоумышленников.
- Идентификаторы партнеров не могут быть повторно использованы через небольшой промежуток времени. Как правило, для управления доступом используются *списки управления доступом* (Access Control List, ASL), в которых хранятся разрешения доступа, предоставленные всем партнерам по обмену данными. Если в базе данных списков управления доступом остался список уничтоженного партнера по обмену данными и идентификатор этого партнера используется вторично, новый партнер унаследует все права доступа уничтоженного партнера. Избежать подобной опасности можно, только если запретить использование идентификаторов уничтоженных партнеров в течение продолжительного времени, а лучше вообще сделать идентификаторы уникальными.

Протоколы обмена сообщениями.

Протокол службы аутентификации - предназначен для выполнения обмена информацией между клиентом и AS Kerberos.

Протокол аутентификации клиента и сервера используется сетевыми приложениями для аутентификации клиента в сервере и наоборот. Для успешного выполнения, аутентификации клиент должен заранее получить с помощью службы выдачи билетов или TOS информацию идентификации сервера.

Протокол выдачи билетов предназначен для обмена информацией между клиентом и сервером Kerberos, выдающим билеты (TOS). Он инициируется клиентом при необходимости получить информацию идентификации для определенного сервера (который может быть зарегистрирован в удаленном владении).

Протокол KRB_SAFE используется клиентами при необходимости обнаружения несанкционированной модификации сообщений, которыми они обмениваются. Модификация сообщений обнаруживается с помощью подсчета контрольной суммы данных пользователя и дополнительной контрольной информации. Контрольная сумма шифруется с помощью специального ключа, который выбирается в результате переговоров, или с помощью ключа сеанса.

Протокол KRB_PRIV используется клиентами при необходимости передачи чрезвычайно конфиденциальных данных и обнаружения несанкционированной модификации сообщений. Это делается с помощью подсчета контрольной суммы данных пользователя и дополнительной контрольной информации.

Протокол KRB_CRED используется клиентами при необходимости *от*правки информации идентификации Kerberos от одного хоста другому хосту.

База данных Kerberos.

Сервер Kerberos должен иметь доступ к базе данных, где хранятся информация об идентификаторах партнеров по обмену данными и секретные ключи аутентифицируемых партнеров. Реализация сервера Kerberos не предполагает обязательное расположение сервера и баз данных на одной машине. Существует возможность хранения базы данных партнеров по обмену данными в пространстве имен сети, если записи этой базы защищены от несанкционированного доступа. Однако с точки зрения целостности и безопасности данных этого делать не рекомендуется.

Модель распределенной безопасности Windows 2000.

Модель распределенной безопасности Windows 2000 основана на трех основных концепциях:

- Каждая рабочая станция и сервер имеют прямой доверенный путь (trust path) к контроллеру того домена, членом которого является данная машина. Доверенный путь устанавливается службой NetLogon с помощью аутентифицированного соединения RPC с контроллером домена. Защищенный канал устанавливается и с другими доменами Windows NT с помощью междоменных доверительных отношений. Он используется для проверки информации безопасности, включая идентификаторы безопасности (Security Identifiers, SID) пользователей и групп.
- Перед выполнением запрошенных клиентом операций сетевые службы имперсонализируют контекст безопасности этого клиента. Имперсонализация основана на маркере адреса безопасности, созданном локальной системой безопасности (Local Security Authority, LSA). Он представляет собой авторизацию клиента на сервере. Поток, находящийся на сервере и соответствующий данному клиенту, имперсонализирует контекст безопасности клиента и выполняет операции в соответствии с авторизацией данного клиента, а не в соответствии с идентификатором безопасности сервера. Имперсонализация поддерживается всеми службами Windows 2000, включая, например, службу удаленного файлового сервера CIFS\SNB. Аутентифицированный RPC и DCOM поддерживают имперсонализацию для распределенных приложений. Такие службы BackOffice, как Exchange, SNA Server и Internet Information Server, также поддерживают имперсонализацию.

- Ядро Windows 2000 поддерживает объектно-ориентированное управление доступом, сравнивая SID в маркере доступа с правами доступа, определенными в списке управления доступом данного объекта. Каждый объект Windows 2000 (ключи реестра, файлы и каталоги NTFS, общие ресурсы, объекты ядра, очереди печати и т.д.) имеют собственные списки управления доступом. Ядро Windows 2000 проверяет разрешения доступа при каждой попытке доступа к данному объекту. Управление доступом и аудит осуществляются с помощью настройки свойств безопасности объекта, позволяющих предоставить доступ к объекту пользователю или группе. Управление авторизацией выполняется централизованно посредством включения пользователей в группы Windows 2000, которым предоставлены необходимые права доступа.

Безопасность lP.

Средства безопасности протокола IP позволяют управлять защитой всего IP трафика от источника информации до ее получателя. Возможности IP Security Management (Управление безопасностью IP) в системе Windows 2000 позволяют назначать и применять политику безопасности, которая гарантирует защищенный обмен информацией для всей сети. Механизм IP Security (Безопасность IP) реализован прозрачно для пользователя, администрирование безопасности централизовано и совмещает гарантии безопасного обмена информацией с легкостью использования.

Операционная система Microsoft Wil1dows 2000 Server упрощает развертывание и управление безопасностью сети при использовании средств Windows IP Security, которые являются гибкой и надежной реализацией Протокола безопасности IP (IP Security, IPSec).

Потребность в защите сетей, основанных на протоколе IP, уже достаточно велика и растет с каждым годом. В настоящее время в широко

взаимосвязанном деловом мире сетей internet, intranet, extranet, филиалов и удаленного доступа, по сетям передается важная информация, конфиденциальность которой нельзя нарушать. Одним из основных требований, предъявляемых к сети со стороны сетевых администраторов и прочих профессионалов, обслуживающих и использующих сети, является требование гарантии, что этот трафик будет защищен от:

- Модификации данных во время пути по сети.
- Перехвата, просмотра или копирования.
- Доступа субъектов, не имеющих на это прав.

Эти проблемы характеризуются такими показателями, как *целостность данных, конфиденциальность* и *аутентичность.* Кроме того, защита от *повторного использования* (replay protection) предотвращает принятие повторно посланного пакета.

Преимущества безопасности IP.

Сетевые атаки могут привести к неработоспособности системы, считыванию конфиденциальных данных и к другим дорогостоящим нарушениям. Требуются методы "сильного" шифрования и сертификации, основанные на криптографических алгоритмах, для безопасности обмена информации. Однако высокий уровень безопасности не должен ухудшать производительность труда пользователей или увеличивать затраты на администрирование.

Безопасность IP в Windows 2000 реализована прозрачно для пользователя. Для информационного взаимодействия под управлением средств безопасности IP пользователи не обязательно должны располагаться в одном домене, они могут находиться в любом доверенном (trusted) домене в пределах корпоративной сети. IP Security имеет централизованное администрирование. Политика безопасности' создается администратором домена для наиболее общих сценариев организации взаимодействия. Эта политика хранится в Active Directory и привязывается к доменной политике.

Когда компьютер регистрируется в домене, доменная политика автоматически подбирает политику безопасности, что помогает избежать необходимости конфигурировать каждый компьютер индивидуально.

IP Security в Windows 2000 обеспечивают следующие преимущества, которые помогают достичь высокого уровня безопасности взаимодействия при низких затратах:

- Централизованное администрирование политикой безопасности, что уменьшает затраты на административные издержки. Политика IPSec может быть создана и назначена на уровне домена, что устраняет необходимость индивидуального конфигурирования каждого компьютера. Однако если компьютер имеет уникальные требования, или это автономный компьютер, политика может быть назначена непосредственно.
- Прозрачность безопасности **IP** для пользователей и прикладных программ. Не нужно иметь отдельные программные средства безопасности для каждого протокола в стеке TCP/IP, поскольку приложения, использующие TCP/IP, передают данные уровню протокола IP**,** где они шифруются. Установленная и настроенная служба IPSec являются прозрачной для пользователя, не требуя обучения.
- Гибкость конфигурирования политики безопасности, которая помогает решать задачи в различных конфигурациях. Внутри каждой политики могут быть настроены службы безопасности, чтобы обеспечить потребности на всех уровнях, от индивидуального пользователя до сервера или до уровня предприятия. Политика может быть сконфигурирована так, чтобы соответствовать экспортным правилам и ограничениям.
- Конфиденциальные службы, которые предотвращают попытки несанкционированного доступа к чувствительным данным, в то

время как эти данные передаются между поддерживающими связь сторонами.

- Туннелирование. Данные могут быть посланы через безопасные туннели для обмена информацией в Internet и intranet.
- Усиленная служба аутентификации, которая предотвращает интерпретацию данных при помощи использования ложно предоставленных идентификаторов.
- Ключи с большой длиной и динамический повторный обмен ключами в течение текущих сеансов связи, что помогает защитить соединение против атак.
- Безопасная связь от начала до конца для частных пользователей сети внутри одного и того же домена или через любой доверенный домен внутри корпоративной сети.
- Безопасная связь от начала до конца между удаленными пользователями и пользователями в любом домене корпоративной сети, основанная на IP- адресах.
- Промышленный стандарт - IPSec, обеспечивает открытые возможности для частных технологий шифрования IP. Менеджеры сети извлекают выгоду из возникающей в результате использования открытых стандартов способности к взаимодействию с другими платформами и продуктами.
- Сертификаты с открытым ключом и поддержка ключей pre-shared. Это требуется для разрешения установления безопасной связи с компьютерами, которые не являются частью доверенного домена.
- IPSec работает в тандеме с другими механизмами защиты, сетевыми протоколами и базовыми механизмами защиты Windows 2000.
- Поддерживается шифрование сообщений RSVP для реализации QoS и ACS в Windows 2000. Не нужно отключать IPSec, чтобы

получить полное преимущество от приоритетного управления шириной полосы пропускания, обеспечиваемого этими службами.

Криптография.

Криптография - набор математических методов для шифрования и дешифрации данных. Ее применение обеспечивает надежную передачу данных и предотвращение их получения несанкционированной стороной. Две функции криптографии, используемые Windows 2000 IP Security: криптография с закрытым ключом и криптография с открытым ключом.

Алгоритмы криптографии.

Методика Diffie- Hellman (D-H) - алгоритм шифрования с открытым ключом (названный по имени изобретателей), который позволяет двум поддерживающим связь сущностям договариваться об общедоступном ключе без требования шифрования во время порождения ключа. Процесс начинается двумя сущностями, обменивающимися общедоступной информацией. Затем каждый объект объединяет общую информацию другой стороны со своей собственной секретной информацией, чтобы сгенерировать секретное общедоступное значение.

Код аутентификации хэшированного сообщения (HMAC, Hash Message Authentication Code HMAC) - алгоритм с закрытым ключом. Целостность, установление подлинности и предотвращение повторного использования сообщений обеспечивается этим алгоритмом. Установление подлинности, использующее функции хэширования (перемешивания), объединено с методом закрытого ключа. Хэшированное значение, известное также как *дайджест*(digest) сообщений, используется для создания и проверки цифровой подписи. Это уникальное значение намного меньшее, чем первоначальное сообщение, созданное из цифровой копии кадра данных. Если передаваемое сообщение изменилось по пути следования, то хэшированное значение будет отличаться от оригинала, а IP-пакет будет отвергнут.

HMAC-MD5 дайджест сообщений-5 (MD5, Message Digest) - функция хэширования, которая порождает 128-разрядное значение. Значение - подпись данного блока данных. Эта подпись используется для установления подлинности, целостности и предотвращения повторного использования.

HMAC-SHA - безопасный алгоритм хэширования (SHA, Secure Hash Algorithm) - это еще одна функция хэширования, которая порождает 160-разрядное значение подписи, используемое для установления подлинности, целостности и предотвращения повторного использования.

DES-CBC - стандарт шифрования данных (Data Encryption Standard, DES) - формирование цепочки шифрованных блоков (CBC, Cipher Block Chaining) алгоритм шифрования с закрытым ключом, используемый для конфиденциальности. Генерируется случайное число, которое используется совместно с закрытым ключом для шифрования данных.