UNIX (Linux/FreeBSD) взаимодействие с Windows

Оглавление

Цели курса

Современный подход к защите информации основывается на управлении идентификацией и авторизацией пользователей. В рамках курса рассматривается варианты решений этой задачи в сетях объединяющих UNIX сервера и рабочие станции Windows.

Уникальной особенностью курса является возможность выбора слушателем любимой операционной системы из двух наиболее популярных - Linux или FreeBSD.

Настройка стенда

Каждая рабочая станция слушателя имеет WinXP в качестве хост-системы на которой установлены 4 виртуальные машины:

- Gate (Linux или FreeBSD)
- Server (Linux или FreeBSD)
- Win2k3
- WinXP

Каждая из виртуальных машин должна быть настроена в соответствующей конфигурации:

Gate (Linux или FreeBSD)

RAM: 192-256MB

HDD: FreeBSD-Gate.vdi или Ubuntu-Gate.vdi

Network Card 1 (сетевой мост)

IP: 172.16.2.X/24

DNS/Gateway: 172.16.2.254

Network Card 1 (внутренняя сеть)

IP: 192.168.X.1/24

Не забудьте выключить поддержку USB

Предварительная настройка ОС

Закомментируйте эти строки в файле /etc/hosts

```
#192.168.X.1           gate.corpX.un gate
#192.168.X.10          server.corpX.un server
```

Настройка DNS

Во FreeBSD нужно выполнить следующее:

```
gate# cat /etc/rc.conf
...
named_enable="YES"
...

gate# cat /etc/namedb/named.conf
options {
        directory       "/etc/namedb";
```

```
        pid-file        "/var/run/named/pid";
};

zone "." {
        type hint;
        file "named.root";
};

zone "corpX.un" {
        type master;
        file "master/corpX.un";
};

zone "X.168.192.IN-ADDR.ARPA" {
        type master;
        file "master/corpX.rev";
};
```

```
gate# cd /etc/namedb/master/
```

```
gate# cat corpX.un
$TTL    3h
@                   SOA     gate root.gate  1 1d 12h 1w 3h
                    NS      gate
gate                A       192.168.X.1
server              A       192.168.X.10
win2k3              A       192.168.X.20
winxp               A       192.168.X.30
_kerberos._udp      SRV     01 00 88 gate
_kerberos._tcp      SRV     01 00 88 gate
_kpasswd._udp       SRV     01 00 464 gate
_kerberos-adm._tcp  SRV     01 00 749 gate
_kerberos           TXT     CORPX.UN
```

```
gate# cat corpX.rev
$TTL    3h
@       SOA     gate.corpX.un. root.gate.corpX.un.  1 1d 12h 1w 3h
        NS      gate.corpX.un.
1       PTR     gate.corpX.un.
10      PTR     server.corpX.un.
20      PTR     win2k3.corpX.un.
30      PTR     winxp.corpX.un.
```

```
gate# named-checkconf -z
```

```
gate# named-checkzone corpX.un corpX.un
```

```
gate# named-checkzone X.168.192.in-addr.arpa. corpX.rev
```

```
gate# /etc/rc.d/named start
```

```
gate# cat /etc/resolv.conf
domain corpX.un
nameserver 127.0.0.1
```

```
gate# host ya.ru
gate# host gate.corpX.un
gate# host server.corpX.un
gate# host 192.168.X.10
```

```
gate# host 192.168.X.1
gate# dig TXT _kerberos.corpX.un
gate# dig SRV _kerberos._tcp.corpX.un
```

В Ubuntu нужно выполнить следующее:

```
root@gate:~# apt-get install bind9

root@gate:~# cat /etc/bind/named.conf.local
zone "corpX.un" {
        type master;
        file "/var/cache/bind/corpX.un";
};

zone "X.168.192.in-addr.arpa" {
        type master;
        file "/var/cache/bind/corpX.rev";
};

root@gate:~# cd /var/cache/bind/

root@gate:~# cat corpX.un
$TTL    3h
@                   SOA     gate root.gate  1 1d 12h 1w 3h
                    NS      gate
gate                A       192.168.X.1
server              A       192.168.X.10
win2k3              A       192.168.X.20
winxp               A       192.168.X.30
_kerberos._udp      SRV     01 00 88 gate
_kerberos._tcp      SRV     01 00 88 gate
_kpasswd._udp       SRV     01 00 464 gate
_kerberos-adm._tcp  SRV     01 00 749 gate
_kerberos           TXT     CORPX.UN

root@gate:~# cat corpX.rev
$TTL    3h
@       SOA     gate.corpX.un. root.gate.corpX.un.  1 1d 12h 1w 3h
        NS      gate.corpX.un.
1       PTR     gate.corpX.un.
10      PTR     server.corpX.un.
20      PTR     win2k3.corpX.un.
30      PTR     winxp.corpX.un.


root@gate:~# named-checkconf -z

root@gate:~# named-checkzone corpX.un corpX.un

root@gate:~# named-checkzone X.168.192.in-addr.arpa. corpX.rev

root@gate:~# /etc/init.d/bind9 restart

root@gate:~# cat /etc/resolv.conf
domain corpX.un
nameserver 127.0.0.1

root@gate:~# host ya.ru
root@gate:~# host gate.corpX.un
root@gate:~# host server.corpX.un
```

```
root@gate:~# host 192.168.X.10
root@gate:~# host 192.168.X.1
root@gate:~# dig TXT _kerberos.corpX.un
root@gate:~# dig SRV _kerberos._tcp.corpX.un
```

Синхронизация времени

Во FreeBSD следует выполнить следующие шаги:

```
gate# cp /usr/share/zoneinfo/Europe/Moscow /etc/localtime

gate# ntpdate ntp.bmstu.ru
11 Sep 15:11:21 ntpdate[1769]: adjust time server 195.19.32.193 offset -0.010034
sec
```

В Ubuntu следует выполнить следующие шаги:

```
root@gate:~# cp /usr/share/zoneinfo/Europe/Moscow /etc/localtime

root@gate:~# ntpdate ntp.bmstu.ru
11 Sep 15:11:21 ntpdate[1769]: adjust time server 195.19.32.193 offset -0.010034
sec
```

Перезапуск служб NIS, NFS и RPCBIND во FreeBSD

```
gate# /etc/rc.d/nfsd stop
gate# /etc/rc.d/mountd stop
gate# /etc/rc.d/ypserv stop
gate# /etc/rc.d/rpcbind restart
gate# /etc/rc.d/ypserv start
gate# /etc/rc.d/mountd start
gate# /etc/rc.d/nfsd start
```

Server (Linux или FreeBSD)

RAM: 192-256MB

HDD: FreeBSD-Server.vdi или Ubuntu-Server.vdi

Network Card (внутренняя сеть)

IP: 192.168.X.10/24

DNS/Gateway: 192.168.X.1

Не забудьте выключить поддержку USB

Настройка DNS

Во FreeBSD

```
server# cat /etc/resolv.conf
domain corpX.un
nameserver 192.168.X.1
```

В Ubuntu

```
root@server:~# cat /etc/resolv.conf
domain corpX.un
nameserver 192.168.X.1
```

Синхронизация времени

Во FreeBSD следует выполнить следующие шаги:

```
gate# cp /usr/share/zoneinfo/Europe/Moscow /etc/localtime

gate# ntpdate ntp.bmstu.ru
11 Sep 15:11:21 ntpdate[1769]: adjust time server 195.19.32.193 offset -0.010034
sec
```

В Ubuntu следует выполнить следующие шаги:

```
root@gate:~# cp /usr/share/zoneinfo/Europe/Moscow /etc/localtime

root@gate:~# ntpdate ntp.bmstu.ru
11 Sep 15:11:21 ntpdate[1769]: adjust time server 195.19.32.193 offset -0.010034
sec
```

Win2k3 (Windows 2003)

RAM: 256MB

HDD: Win2k3_new.vdi

Network Card (внутренняя сеть)

IP: 192.168.X.20/24

DNS/Gateway: 192.168.X.1

Не забудьте выключить поддержку USB

WinXP (Windows XP)

RAM: 256MB

HDD: WinXP_new.vdi

Network Card (внутренняя сеть)

IP: 192.168.X.30/24

DNS/Gateway: 192.168.X.1

Не забудьте выключить поддержку USB

Управление идентификацией

Идентификация — объединяет два понятия «аутентификация» и «авторизация».

Аутентификация — проверка того факта, что пользователь действительно тот, за кого он себя выдает. Такая проверка может осуществляться на основе логина и пароля, электронного ключа (e-Token), сканирования отпечатков пальцев или сетчатки глаза.

Авторизация — проверка полномочий пользователя при доступе к ресурсу.

Аутентификация

В рамках курса рассматриваются следующие методы аутентификации:

- NIS
- RADIUS
- OPIE
- NTLM
- Kerberos
- SSPI/GSSAPI

NIS (Network Information Service)

Клиент-серверный протокол, созданный Sun Microsystems, который позволяет обеспечивать доступ к системной конфигурации по всей сети. Первоначально назывался Yellow Pages (Жёлтые страницы) по аналогии с бумажным справочником, в котором перечисляются телефонные номера, но из-за судебных преследований владельцев торговой марки был переименован в NIS.

Лабораторная работа №1

1. В систему gate добавляем пользователя user с uid=10001 gid=10001

2. Настройка сервера

Инсталяция, инициализация БД и запуск во FreeBSD

```
[gate:~] # cat /etc/rc.conf
...
rpcbind_enable="YES"
nisdomainname="corpX.un"
nis_server_enable="YES"
...

[gate:~] # /etc/rc.d/nisdomain start
Setting NIS domain: corpX.un.

[gate:~] # mkdir /var/yp/corpX.un

[gate:~] # /etc/rc.d/rpcbind start
...

[gate:~] # /etc/rc.d/ypserv start
Starting ypserv.

[gate:~] # cp /etc/master.passwd /var/yp/
[gate:~] # cd /var/yp
```

Оставьте в /var/yp/master.passwd только NIS пользователей

```
[gate:/var/yp] # cat master.passwd
user:$1$ACBSNtwF$jZN456YBe.28dQ.QQ4wet1:10001:10001::0:0:User &:/home/user:/bin/sh
```

```
[gate:/var/yp] # ypinit -m corpX.un
Server Type: MASTER Domain: corpX.un
…
Do you want this procedure to quit on non-fatal errors? [y/n: n]
…
Can we destroy the existing /var/yp/corpX.un and its contents? [y/n: n]  y
…
        master server   :  gate.corpX.un
        next host to add:  ^D
…
Is this correct?  [y/n: y]
…
[gate:/var/yp] #
```

Инсталяция, инициализация БД и запуск в Ubuntu

```
root@gate:~# apt-get install nis
...
Nis domain corpX.un
...
 * Starting NIS services
 * binding to YP server...
 * ....
Ждем, пока закончится...

root@gate:~# cat /etc/default/nis
...
NISSERVER=master
...
NISCLIENT=false
...

root@gate:~# /etc/init.d/nis start
<Ctrl-C>
(В случае проблем с запуском см. ниже)

root@gate:~# /usr/lib/yp/ypinit -m
…
        master server   :  gate.corpX.un
        next host to add:  ^D
…

root@gate:~# /etc/init.d/nis start
```

В случае проблем читаем: https://help.ubuntu.com/community/SettingUpNISHowTo

3. Настройка клиента

Запуск во FreeBSD

```
[server:~] # cat /etc/rc.conf
...
rpcbind_enable="YES"
nisdomainname="corpX.un"
nis_client_enable="YES"
...

[server:~] # /etc/rc.d/nisdomain start
Setting NIS domain: corpX.un.
```

```
[server:~] # /etc/rc.d/rpcbind start
Starting rpcbind.

[server:~] # /etc/rc.d/ypbind start
Starting ypbind.

[server:~] # ypcat passwd

[server:~] # vipw

[server:~] # cat /etc/master.passwd
...
+::::::::::

[server:~] # cat /etc/group
...
+:::
```

Установка и запуск в Ubuntu

```
root@server:~# apt-get install nis

root@server:~# ypcat passwd

root@server:~# cat /etc/passwd
...
+

root@server:~# cat /etc/shadow
...
+

root@server:~# cat /etc/group
...
+
```

4. Проверка

Выполните эту проверку на шлюзе и сервере

id user

5. Использование NFS для домашних каталогов

Во FreeBSD выполняются следующие инструкции

Для настройки NFS-сервера

```
[gate:~] # cat /etc/rc.conf
...
rpcbind_enable="YES"
mountd_enable="YES"
nfs_server_enable="YES"
...

[gate:~] # cat /etc/exports
/usr/home -maproot=0 server

[gate:~] # /etc/rc.d/mountd start
Starting mountd.
```

```
[gate:~] # /etc/rc.d/nfsd start
Starting nfsd.
```

Для настройки NFS-клиента
```
[server:~] # mount gate:/usr/home /usr/home/
```

В Ubuntu выполняются следующие инструкции

Для настройки NFS-сервера
```
root@gate:~# apt-get install nfs-kernel-server

root@gate:~# cat /etc/exports
/home server(rw,sync,no_subtree_check)

root@gate:~# /etc/init.d/nfs-kernel-server start
```

Для настройки NFS-клиента
```
root@server:~# apt-get install nfs-common

root@server:~# mount.nfs gate:/home /home
```

RADIUS (Remote Authentication in Dial-In User Service)

Это протокол AAA (Authentication, Authorization и Accounting), разработанный для передачи сведений между центральной платформой AAA и оборудованием Dial-Up доступа (NAS, Network Access Server) и системой биллинга (то есть, системой тарификации использованных ресурсов конкретным абонентом/пользователем).

- Authentication — процесс, позволяющий аутентифицировать (проверить подлинность) субъекта по его идентификационным данным, например, по логину (имя пользователя, номер телефона и т. д.) и паролю.
- Authorization — процесс, определяющий полномочия идентифицированного субъекта на доступ к определённым объектам или сервисам.
- Accounting — процесс, позволяющий вести сбор сведений (учётных данных) об использованных ресурсах. Первичными данными (то есть, традиционно передаваемых по протоколу RADIUS) являются величины входящего и исходящего трафиков: в байтах/октетах (с недавних пор в гигабайтах). Однако протокол предусматривает передачу данных любого типа, что реализуется посредством VSA (Vendor Specific Attributes).

RADIUS был разработан Livingston Enterprises (конкретно Карлом Ригни/Carl Rigney) для их серверов доступа (Network Access Server) серии PortMaster к сети internet, и позже, в 1997, был опубликован как RFC 2058 и RFC 2059 (текущие версии RFC 2865 и RFC 2866). На данный момент существует несколько коммерческих и свободно распространяемых (open-source) RADIUS-серверов. Они несколько отличаются друг от друга по своим возможностям, но большинство поддерживает списки пользователей в текстовых файлах, LDAP, различных базах данных. Учетные записи пользователей могут храниться в текстовых файлах, различных базах данных, или на внешних серверах. Часто для удаленного мониторинга используется SNMP. Существуют прокси-серверы (proxy/forwarding) для RADIUS, упрощающие централизованное администрирование и/или позволяющие реализовать

концепцию интернет-роуминга (internet roaming). Они могут изменять содержимое RADIUS-пакета на лету (в целях безопасности или для выполнения преобразования между диалектами). Популярность RADIUS-протокола, во многом объясняется: открытостью к наполнению новой функциональностью при сохранении работоспособности с устаревающим оборудованием, чрезвычайно высокой реактивностью при обработке запросов ввиду использования UDP в качестве транспорта пакетов, а также хорошо параллелизуемым алгоритмом обработки запросов; способностью функционировать в кластерных (Cluster) архитектурах (например OpenVMS) и мультипроцессорных (SMP) платформах (DEC Alpha, HP Integrity) — как с целью повышения производительности, так и для реализации отказоустойчивости.

Определён в

- RFC 2865 Remote Authentication Dial In User Service (RADIUS)
- RFC 2866 RADIUS Accounting

Также имеет отношение к

- RFC 2548 Microsoft Vendor-specific RADIUS Attributes
- RFC 2607 Proxy Chaining and Policy Implementation in Roaming
- RFC 2618 RADIUS Authentication Client MIB
- RFC 2619 RADIUS Authentication Server MIB
- RFC 2620 RADIUS Accounting Client MIB
- RFC 2621 RADIUS Accounting Server MIB
- RFC 2809 Implementation of L2TP Compulsory Tunneling via RADIUS
- RFC 2867 RADIUS Accounting Modifications for Tunnel Protocol Support
- RFC 2868 RADIUS Attributes for Tunnel Protocol Support
- RFC 2869 RADIUS Extensions
- RFC 2882 Network Access Servers Requirements: Extended RADIUS Practices
- RFC 3162 RADIUS and IPv6
- RFC 3575 IANA Considerations for RADIUS
- RFC 3576 Dynamic Authorization Extensions to RADIUS
- RFC 4672 RADIUS Dynamic Authorization Client MIB
- RFC 4673 RADIUS Dynamic Authorization Server MIB
- RFC 3579 RADIUS Support for EAP
- RFC 3580 IEEE 802.1X RADIUS Usage Guidelines
- RFC 4014 RADIUS Attributes Suboption for the DHCP Relay Agent Information Option

Лабораторная работа №2

1. Инсталяция сервера

Во FreeBSD выполняются следующие инструкции

```
[gate:~] # pkg_add -r freeradius

[gate:~] # cat /etc/rc.conf
...
radiusd_enable=yes
...

[gate:~] # cd /usr/local/etc/raddb/
```

В Ubuntu выполняются следующие инструкции
```
root@gate:~# apt-get install freeradius

root@gate:~# cd /etc/freeradius/
```

2. Настройка сервера
```
gate# cat sites-available/default
...
#     unix
...
#     unix
...

gate# cat users
user Cleartext-Password := "radius"

gate# cat clients.conf
...
client server.corpX.un {
        secret          = testing123
        shortname       = server
}
...
```

3. Запуск сервиса

Во FreeBSD
```
[gate:~] # /usr/local/etc/rc.d/radiusd start
```

В Ubuntu
```
root@gate:~# /etc/init.d/freeradius restart
```

4. Использование pam radius для сервиса login

Во FreeBSD выполняются следующие инструкции
```
[server:~] # cat /etc/radius.conf
auth gate testing123 3 2

[server:~] # cat /etc/pam.d/login
...
auth            sufficient      pam_radius.so           no_warn try_first_pass
auth            include         system
...
```

В Ubuntu выполняются следующие инструкции
```
root@server:~# apt-get install libpam-radius-auth

root@server:~# cat /etc/pam_radius_auth.conf
...
gate testing123 3
...

root@server:~# cat /etc/pam.d/login
...
auth       sufficient   pam_radius_auth.so
# Standard Un*x authentication.
```

5. Верните настройки /etc/pam.d/login в исходное состояние

OPIE (One-time Passwords In Everything)

Иногда возникает ситуация, когда нужно войти на сервер с чужого компьютера. Например, выехав к клиенту, вы выясняете, что он начисто забыл свой пароль. Не возвращаться же из-за этого в офис? А заходить на сервер с машины клиента достаточно опасно – кто знает, какие «шпионы» там установлены? Конечно, решить эту проблему можно несколькими способами. Например, используя ssh с аутентификацией ключевым файлом, который записан на дискету (к слову, дискету можно на радостях и в дисководе забыть, что отнюдь не поспособствует безопасности сервера). Или хорошим решением выглядит LiveCD – в этом случае вы гарантируете себе не только повышение безопасности, но и наличие всех необходимых программ.
Система предоставляет еще один удобный инструмент – одноразовые пароли, они реализуются системой OPIE (One-time Passwords In Everything), основанной на S/Key.
В общих чертах это выглядит следующим образом: при попытке войти на сервер после ввода имени пользователя вы получаете строку-оклик, содержащую номер итерации и некоторую последовательность символов – так называемое «зерно» (seed):

```
login as: sergio

otp-md5 496 ko5622 ext

Password:
```

Далее вам нужно ввести парольную фразу (отзыв), соответствующую указанному номеру и «зерну». Для генерации паролей используется утилита opiekey (поэтому одноразовые пароли на жаргоне иногда называют «опиками»). Ей нужно передать текущий номер последовательности и «зерно», затем будет запрошена секретная фраза (она задается во время активации учетной записи пользователя) и выведен список шести коротких слов, которые и являются одноразовым паролем-отзывом.
В отличие от системного пароля одноразовый нечувствителен к регистру символов, так что вводить его можно и маленькими буквами.
Если в ответ на первое приглашение «Password:» просто нажать <Enter>, то появится еще одно, с включенным эхо-отображением вводимых символов, что позволит вам видеть вводимый текст на экране.

При следующем входе номер последовательности уменьшится, и вводить нужно будет уже другой отзыв. Таким образом, вам не нужно заботиться о сохранности одноразового пароля после того, как он будет использован, – вы можете смело разложить листок с паролем на столе в присутствии пользователя, включить эхо-повтор вводимых символов и, ни от кого не прячась, спокойно набирать парольную фразу. Поскольку после нажатия <Enter> данный пароль теряет свою актуальность, его знание уже никому ничего не дает.
Собственно, основная идея использования одноразовых паролей как раз и заключается в том, что никакая секретная информация (в данном случае таковой является секретная фраза, на основе которой осуществляется генерация паролей) по сети не передается.
В уже все предусмотрено для использования одноразовых паролей. Достаточно создать для пользователей, которые будут работать по данной схеме, соответствующие записи в файле /etc/opiekeys.

Новая запись создается следующей командой:

root# opiepasswd -c -f

Ключ -c дает команду создать запись для пользователя, в данном случае это пользователь test. Если имя пользователя не указано, создается (или изменяется) запись для текущего пользователя. Обычный пользователь может управлять своей записью, root – записями всех пользователей.
Перед началом работы выводится предупреждение, что утилиту opiepasswd следует запускать только в защищенном режиме (с физической консоли или через ssh), чтобы исключить возможность перехвата секретной фразы. Знание данной фразы позволит любому человеку сгенерировать одноразовый пароль для входа под вашим именем.
После всех предупреждений запрашивается секретная фраза. Она понадобится вам в дальнейшем для генерации паролей и управления своей записью. Утилита требует, чтобы длина этой фразы была от 10 до 127 символов, иначе вы получите сообщение об ошибке.

В конце работы утилита выдает текущие номер последовательности (по умолчанию – 499) и слово-«зерно», в данном примере – ko6010. Запоминать эти значения не нужно – они будут выводиться на экран при каждой попытке войти в систему. Последней строкой выводится парольная фраза, которая к данному моменту уже бесполезна и служит только для проверки, – следующий номер последовательности устанавливается в 498.

Обратите внимание, что как только счетчик достигнет 0, вход в систему станет невозможным. Поэтому необходимо заблаговременно осуществлять повторную инициализацию записи в /etc/opiekeys, для чего используется opiepasswd без ключей. Также с помощью ключа -n вы можете задать начальное значение счетчика. Замечу, что переинициализация выполняется без запроса секретной фразы (она остается прежней). Используются только одноразовые пароли, так что повторная инициализация может быть выполнена и удаленно, с использованием небезопасного соединения.

Также утилита opiepasswd используется для дезактивации записи того или иного пользователя. Для этого используется ключ -d. Дезактивация не удаляет запись пользователя из файла opiekeys, а просто «забивает» секретную фразу звездочками. Теперь при входе пользователя в систему по-прежнему будет выдаваться строка-«оклик», но получить для нее правильную парольную фразу будет уже невозможно.

Чтобы полностью отменить запрос на ввод одноразового пароля и вернуться к прежней аутентификации системным паролем, достаточно удалить для соответствующего пользователя строку в файле /etc/opiekeys.

Следующая утилита, которая может быть полезна, – opieinfo. Она возвращает текущие значения номера последовательности и «зерна» для пользователя. Например, ее можно использовать, перед тем как генерировать пароли.

Утилита opiekey имеет несколько полезных ключей. Например, с помощью ключа -n можно указать, сколько паролей должно быть сгенерировано начиная с текущего номера итерации.

Теперь эти пароли можно аккуратно сохранить в надежное место и ехать с ними в командировку или к клиенту – средство для пяти безопасных входов в систему у вас есть.

Естественно, генерировать пароли можно и по мере необходимости, например, на своем домашнем компьютере или ноутбуке с помощью соответствующей утилиты. Но к такому инструменту предъявляются серьезные требования безопасности, поскольку во время генерации вы должны быть уверены, что никто не узнает вашу секретную фразу.

Вы можете настроить вход в систему как исключительно по одноразовым паролям, так и разрешив использование постоянных паролей, предоставив пользователю право выбирать в

зависимости от ситуации, как входить в систему. Например, вам может быть удобнее вводить системный пароль при работе из офиса, а «опики» использовать при необходимости зайти на сервер с «чужой» территории.

Для определения возможных способов доступа используется файл /etc/opieaccess. По умолчанию, как только для пользователя появляется запись в /etc/opiekeys, он уже не может пользоваться системным паролем.

Чтобы разрешить вход по обоим паролям, нужно в opieaccess создать запись типа permit для нужных адресов. Например, чтобы разрешить вход под системным паролем из локальной сети, а для внешних соединений оставить только «опики», можно использовать следующую строку:

```
permit 192.168.0.0 255.255.255.0
```

```
# permit 0.0.0.0 0.0.0.0
```

Закомментированная строка разрешит использование обоих типов паролей с любого адреса. Правило deny позволяет явно запретить использование системного пароля для указанной подсети.

Обратите внимание, что разрешение вводить системные пароли делает возможным использование уже скомпрометированного пароля, сводя на нет преимущества системы OPIE. Единственное, что при этом достигается,– это существенное снижение вероятности раскрыть свой пароль при работе в незащищенном режиме. Поэтому к вопросу раздачи прав следует относиться очень внимательно.

Более тонко процедуру входа в систему можно определить в настройках PAM (см. /etc/pam.d/system и прочие файлы). Нужно заметить, что не все приложения могут работать с одноразовыми паролями. Например, во FreeBSD утилиты login, su, ssh, telnetd поддерживают аутентификацию OPIE, popper – нет (но если вход с системным паролем разрешен, проблем с использованием этой программы не возникает). По умолчанию, если поддержка OPIE каким-то сервисом имеется, в соответствующем файле каталога /etc/pam.d будет присутствовать pam_opie.so.

Лабораторная работа №3

1. Установка OPIE-сервера

Во FreeBSD север OPIE присутствует в штатной поставке.

В Ubuntu необходимо выполнить эту команду:

```
root@gate:~# apt-get install opie-server
```

2. Инициализация opie для пользователя

```
gate# su - user
$ opiepasswd -c -f
...
Enter new secret pass phrase: opiepasswd
Again new secret pass phrase: opiepasswd
...
```

3. Проверка содержимого файла ключей OPIE

```
gate# cat /etc/opiekeys
user 0497 g23394           81a663d5347407bb  Oct 19,2006 09:48:53
```

4. Настройка pam opie для сервиса sshd

Во FreeBSD не требуется настройки.

В Ubuntu необходимо сделать следующее:

```
root@gate:~# cat /etc/ssh/sshd_config
...
ChallengeResponseAuthentication yes
...

root@gate:~# cat /etc/pam.d/sshd
...
auth    sufficient      pam_opie.so
# Standard Un*x authentication.
```

5. Настройка клиента OPIE

Генерация одного пароля

```
$ opiekey 498 g23394
...
Enter secret pass phrase: opiepasswd
PER AND BURT SEE DIVE GILD
```

Генерация нескольких паролей

```
$ opiekey -n 6 497 g23394
...
Enter secret pass phrase: opiepasswd
492: AURA BATH MAUL WASH OWN FIST
...
497: FLAG PEP VOID CODA TIC SKID
```

6. Верните настройки в исходное состояние

NTLM

NTLM (NT LAN Manager) — Является протоколом сетевой аутентификации, разработанной фирмой Microsoft для Windows NT

Когда, полтора десятка лет назад, компания Microsoft начала серьезную работу над созданием централизованных сетей масштабов предприятия при работе над операционной системой Windows NT, перед разработчиками была поставлена весьма сложная, и новая по тем временам задача - реализовать технологии single sign-on, и One user - one password.

One user - one password (один пользователь - один пароль) означает, что у пользователя должен быть только один пароль. Единый пароль используется для доступа ко всем ресурсам и протоколам сети. Single sign-on (единый вход) подразумевает, что этот пароль указывается всего один раз - при входе пользователя в сеть.

Можно много спорить о преимуществах и недостатках такого подхода, но бесспорно одно - этот подход удобен как для пользователей, так и для разработчиков приложений. Пользователь избавлен от необходимости помнить много паролей и вводить их, а

разработчику не надо задумываться над тем, как организовать аутентификацию пользователя.

Для этого необходимо было разработать такую схему аутентификации, которая позволила бы любому сетевому приложению передавать данные аутентификации независимо от сетевого протокола. Так родился NTLM и NTLMSSP (NTLM Security Service Provider) - подсистема позволяющая любому клиент-серверному приложению использовать NTLM ничего не зная о его внутренней структуре.

Нельзя сказать, чтобы Microsoft проигнорировал требования безопасности для протокола аутентификации. В общем-то, на тот момент протокол NTLM не был слабее многих уже использовавшихся протоколов, и в чем-то даже лучше. Но сейчас можно с уверенностью сказать, что вместе с протоколом NTLM появилось большое количество проблем связанных с его безопасностью. Часть проблем вызвана тем, что Microsoft должен был сохранить совместимость с существующими сетями LanManager для MS-DOS и Windows for Workgroups. Другие являются ошибками дизайна и объясняются новизной решаемой проблемы. Третьи являются исключительно криптографическими, т.к. тогда производители ПО редко имели в штате профессиональных криптоаналитиков.

Фактически, NTLM - это результат дальнейшего развития LANMAN

Никакой официальной информации о нём не поступало, но многое выяснила группа разработчиков Samba во время разработки своей программы, эта информация отражена в RFC 2433 для версии 1 и RFC 2759 для версии 2.

Для передачи на сервер аутентификации (PDC - главный контроллер домена) имени пользователя, хэша пароля и мандата домена в Windows 98 применяется протокол LANMAN, а в Windows NT - протокол NTLM. Windows 2000 и Windows XP по умолчанию делают попытку аутентификации Kerberos, в то же время они сохраняют обратную совместимость с аутентификацией NTLM.

Мы не будем углубляться в технические детали более, чем это необходимо для понимания проблемы, тем не менее, иногда от читателя потребуются некоторые минимальные представления о процессах аутентификации и авторизации, программировании и криптографии.

Что происходит после нажатия на Ctrl+Alt+Del? Появляется запрос локальной подсистемы безопасности (Local Security Authority, LSA) на ввод имени пользователя и пароля. После ввода пароль хэшируется (криптографический хэш - одностороннее преобразование усложняющее восстановление по нему оригинального пароля) и хэш помещается в хранилище LSA. В открытом виде он больше уже нигде не фигурирует (в старых версиях Windows пароль мог храниться в открытом виде или с обратимым шифрованием, т.к. старые версии LanManager использовали аутентификацию в открытом тексте, но не будем вспоминать эти времена). Кроме того, к хранилищу LSA нельзя обратиться напрямую стандартными методами. В хранилище хэши находятся до окончания сеанса работы.

Протокол NTLM относится к семейству challenge-response (запрос-ответ) протоколов. Это означает, что ни пароль ни его хэш никогда не передаются <как есть>, вместо этого они используются для генерации ответа (response) на случайный запрос (challenge). Аутентифицирующая сторона сравнивает полученный ответ с вычисленным локально. Генерация и проверка запроса и ответа осуществляется не приложениями, а провайдером NTLMSSP. Данные аутентификации, генерируемые NTLMSSP через специальные функции API (InitializeSecurityContext()/AcceptSecurityContext()) могут быть включены в любой протокол прикладного уровня, упаковка этих данных (называемых security blob - <начинка безопасности>) это все, что требуется от приложений с точки зрения NTLMSSP. После успешной проверки подсистема безопасности генерирует токен, который может быть

использован серверным приложением с правами локальной системы для имперсонирования пользователя, т.е. при подключении пользователя к серверному приложению серверное приложение может работать от его имени. В таком случае пользователь совершает вход на удаленную систему. Возникает вопрос - а может ли серверное приложение обратиться к другим сетевым ресурсам с использованием NTLM, не запрашивая дополнительной аутентификации? Если в хранилище LSA удаленного компьютера нет хэшей пароля пользователя - то это невозможно. Отсюда, например, невозможность <прозрачного> доступа к сетевому диску из telnet-сеанса или через Web-сервер если доступ через telnet или к Web происходит с NTLM аутентификацией.

Хэши NTLM

В семействе протоколов NTLM (как мы увидим далее, NTLM-подобных протоколов несколько) могут использоваться 2 типа хэшей: LM (LanManager) хэш, унаследованный от предыдущих реализаций LanManager и NT (New Technology) хэш, созданный для протокола NTLM. Соответственно, при входе пользователя в систему, как правило, от пароля берутся и хранятся оба этих хэша. Первая версия протокола NTLM для совместимости поддерживала оба ключа (NT или LM ключем обычно называют соответствующий хэш пароля). В более поздних реализациях используется только NT ключ, однако по-умолчанию LM хэш все равно создается при входе и помещается в хранилище LSA. Давайте рассмотрим оба алгоритма хэширования.

LM ключ получается из пароля в 8-битной OEM кодировке (cp866 для России) с помощью алгоритма DES.

Для справки: DES является симметричным блочным шифром, использующим 56 битный ключ для шифрования 64 битного блока текста. Реально, внутри алгоритма используется 64 битный ключ, однако длина ключа искусственно занижена по непонятным соображениям - 56 битный ключ <растягивается> за счет вставки дополнительного бита через каждые 7 бит ключа. Поскольку DES обладает относительной стойкостью к атакам известного открытого текста, он может быть использован в качестве криптографической хэш функции, если в качестве открытого текста использовании какой-либо известный текст, а в качестве ключа - хэшируемое слово. Известный текст может быть либо случайным (в таком случае он называется salt - соль, и хранится в месте с паролем), либо предопределенным, в таком случае он называется Magic Word - заклинание. В классической реализации crypt() в Unix использовался первый подход, в Windows используется магическое слово KGS!@#$% (посмотрите на клавиатуру:. Наверное, это был чей-то пароль). При использовании в качестве хэш функции DES генерирует 64 битный хэш по 56 битному тексту.

Поскольку DES позволяет получить хэш лишь от 7-символьного блока, то реально используется пароль из 14 символов (более короткий пароль дополняется нулями), который разбивается на два блока по 7 символов, от каждого из которых независимо вычисляется хэш. В итоге получается 128-битный хэш <склееный> из двух частей.

Недостатки алгоритма очевидны. Независимое вычисление двух блоков позволяет и их независимый взлом, т.е. реально каждый 64 бита хэша можно атаковать с целью восстановления пароля. Причем длинный пароль может быть легче восстановить чем более короткий. Например, для пароля из 12 символов, сначала за считанные секунды подбираются последние 5 символов, после чего делается предположение о структуре пароля и первые 7 символов пароля подбираются по более ограниченному алфавиту. В настоящее время известны очень быстрые реализации DES с использованием 64-битной арифметики, что делает его абсолютно непригодным для криптографии. В общем случае, восстановление

пароля по LM хэшу на современной технике вопрос не более чем нескольких дней. Кроме того, фиксированное магическое слово позволяет использование таблицы заранее посчитанных значений ключей, что делает возможным восстановление пароля по LM хэшу в реальном времени.

NT ключ вычисляется с помощью стандартного алгоритма хэширования MD4. Хэш MD4 берется от пароля записанного в 16-битной кодировке Unicode с последовательностью байт low endian (т.е. первым байтом идет номер символа в странице). Пароль вычисляется с учетом регистра. MD4 имеет несколько криптографических проблем, самой большой из них является маленькое время вычисления хэша, что позволяет перебирать достаточно большое количество комбинаций в единицу времени упрощая, например, атаку по словарю или подбор слабой комбинации символов.

Подсказка: существует большое количество программ для восстановления пароля из NT или LM ключа путем подбора по словарю или перебора - John-the-Ripper, LophtCrack, Cain & Abel. При наличии обоих ключей, обычно сначала восстанавливается пароль в верхнем регистре из LM-ключа, затем по NT-ключу восстанавливается регистр пароля. Такой подход, в частности, реализован в Cain & Abel (http://www.oxid.it), являющейся на сегодня наиболее мощным и универсальным инструментом для выполнения различных задач связанных с обнаружением слабых конфигураций, в т.ч. и многих проблем NTLM. Мы еще неоднократно будем возвращаться к возможностям этой утилиты. Самая быстрая реализация алгоритма DES ориентированная на взлом LM-ключей в Solar Designer'овском John-the-Ripper.

Совет: Можно запретить генерацию LM-ключей в системе путем установки в 1 значения NoLmHash в разделе реестра
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa

Собственно NTLM

На сегодняшний момент существует две основных версии NTLM. Первая версия NT LanManager 0.12, часто называемая NTLM v1 и NTLM v2. Кроме того, существует несколько основанных на NTLM диалектов, например протокол аутентификации MS-CHAPv2 (MS-CHAP является NTLM 0.12 в чистом виде). Мы подробней остановимся на NTLM 0.12, т.к. нас будут интересовать его, так часто обсуждаемые, криптографические уязвимости.

Итак, NTLMSSP генерирует <кусочки> данных безопасности (security blob) которыми обмениваются клиентское и серверное приложение. Обмен происходит в несколько этапов, и для NTLM 0.12 выглядит следующим образом:

1. Клиент посылает серверу запрос на аутентификацию.
2. Сервер отвечает пакетом, в котором указывается выбранная NTLM аутентификация и поле EncryptionKey которого содержит 64-битный случайный запрос (challenge).
3. Клиент посылает сообщение, содержащее поля AccountName (учетная запись), PrimaryDomain (домен учетной записи), CaseInsensitivePassword (пароль не чувствительный к регистру, фактически это LM-ответ) и CaseSensitivePassword (пароль чувствительный к регистру, фактически NT-ответ). Оба ответа являются 192-битными и вычисляются на основе NT и LM ключа по одному и тому же алгоритму. Если соответствующего ключа нет, то и соответствующий ответ будет нулевым.

Kerberos

Безопасность протокола в значительной мере основывается на том, что системные часы участников более-менее синхронны и на временных утверждениях подлинности, называемых *билетами Kerberos*.

Ниже приведено упрощенное описание протокола. Следующие аббревиатуры будут использованы:

- AS (Authentication Server) = Сервер аутентификации
- TGS (Ticket Granting Server) = Сервер предоставления билетов
- SS (Service Server) = Ресурс, предоставляющий некий сервис, к которому требуется получить доступ
- TGT (Ticket Granting Ticket) = Билет для получения билета

В двух словах клиент авторизируется на AS, используя свой долгосрочный секретный ключ, и получает билет от AS. Позже клиент может использовать этот билет для получения дополнительных билетов на доступ к ресурсам SS без необходимости прибегать к использованию своего секретного ключа.

Более детально:

Шаги входа пользователя в систему:

1. Пользователь вводит имя и пароль на клиентской машине.
2. Клиентская машина выполняет над паролем одностороннюю функцию (обычно хэш), и результат становится секретным ключом клиента/пользователя.

Шаги аутентификации клиента:

1. Клиент посылает простым текстом сообщение серверу AS, запрашивая сервисы от имени пользователя. Например так: «Пользователь АБВ хочет запросить сервисы». Обратите внимание, что ни секретный ключ, ни пароль не посылаются на AS.
2. AS проверяет, есть ли такой клиент в базе. Если есть, то назад AS отправляет следующие два сообщения:
    - Сообщение A: *Сессионный Ключ Client/TGS*, зашифрованный секретным ключом клиента/пользователя.
    - Сообщение B: TGT (который включает ID клиента, сетевой адрес клиента, период действия билета, и *Сессионный Ключ Client/TGS*), зашифрованный секретным ключом TGS.
3. Как только клиент получает сообщения A и B, он расшифровывает сообщение A, чтобы получить *Сессионный Ключ Client/TGS*. Этот сессионный ключ используется для дальнейшего обмена с сервером TGS. (Важно: Клиент не может расшифровать сообщение B, так как оно зашифровано секретным ключом TGS.) В этот момент у пользователя достаточно данных, чтобы авторизоваться на TGS.

Шаги авторизации клиента для получения сервиса:

1. При запросе сервисов клиент отправляет следующие два сообщения на TGS:
    - Сообщение C: Содержит TGT, полученный в сообщении B и ID требуемого сервиса.
    - Сообщение D: Аутентикатор (составленный из ID клиента и временного штампа), зашифрованный на *Сессионном Ключе Client/TGS*.
2. После получения сообщений C и D, TGS извлекает сообщение B из сообщения C и расшифровывает его используя секретный ключ TGS. Это дает ему *Сессионный Ключ*

Client/TGS. Используя его TGS расшифровывает сообщение D и посылает следующие два сообщения клиенту:

- Сообщение E: *Client-to-server ticket* (который содержит ID клиента, сетевой адрес клиента, время действия билета и *Сессионный Ключ Client/server*) зашифрованный секретным ключом сервиса.
- Сообщение F: *Сессионный ключ Client/server*, зашифрованный на *Сессионном Ключе Client/TGS*.

Шаги клиента при запросе сервиса:

1. При получении сообщений E и F от TGS, у клиента достаточно информации для авторизации на SS. Клиент соединяется с SS и посылает следующие два сообщения:
   - Сообщение E из предыдущего шага (*client-to-server ticket*, зашифрованный секретным ключом сервиса).
   - Сообщение G: новый аутентикатор, зашифрованный на *client/server session key*, и включающий ID клиента и временной штамп.
2. SS расшифровывает билет используя свой секретный ключ для получения *Сессионного Ключа Client/Server*. Используя сессионный ключ, SS расшифровывает аутентикатор и посылает клиенту следующее сообщение для подтверждения готовности обслужить клиента и показать, что сервер действительно является тем, за кого себя выдает:
   - Сообщение H: Временной штамп, указанный клиентом + 1, зашифрованный на *Сессионном Ключе Client/Server*.
3. Клиент расшифровывает подтверждение, используя *Сессионный Ключ Client/Server* и проверяет, действительно ли временной штамп корректно обновлен. Если это так, то клиент может доверять серверу и может начать посылать запросы на сервер.
4. Сервер предоставляет клиенту требуемый сервис.

Лабораторная работа №4

1. Настройка KDC сервера

Во FreeBSD следует выполнить следующие шаги

```
[gate:~] # cat /etc/rc.conf
...
kerberos5_server_enable="YES"
...

[gate:~] # cat /etc/krb5.conf
[libdefaults]
    default_realm = CORPX.UN

[gate:~] # kstash
Master key: 123

[gate:~] # kadmin -l
kadmin> init CORPX.UN
Realm max ticket life [unlimited]:
Realm max renewable ticket life [unlimited]:
kadmin> add user
...
```

```
user@CORPX.UN's Password: kpasswd
Verifying - user@CORPX.UN's Password: kpasswd
kadmin> list *
kadmin> quit


[gate:~] # /etc/rc.d/kerberos start

В Ubuntu выполняются следующие шаги

root@gate:~# apt-get install krb5-kdc krb5-admin-server

root@gate:~# krb5_newrealm
...
Enter KDC database master key:
Re-enter KDC database master key to verify:

root@gate:~# ls -l /var/lib/krb5kdc/

root@gate:~# cat /etc/krb5.conf
[libdefaults]
    default_realm = CORPX.UN

root@gate:~# kadmin.local

kadmin.local:  addprinc user
...
Enter password for principal "user@CORPX.UN": kpasswd
Re-enter password for principal "user@CORPX.UN": kpasswd

kadmin.local:  listprincs
...
user@CORPX.UN

root@gate:~# /etc/init.d/krb5-kdc restart

2. Настройка Kerberos клиента

Во FreeBSD следует выполнить следующий шаг

[server:~] # cat /etc/krb5.conf
[libdefaults]
    default_realm = CORPX.UN


В Ubuntu следует выполнить следующие шаги

root@server:~# apt-get install krb5-user

root@server:~# cat /etc/krb5.conf
[libdefaults]
    default_realm = CORPX.UN


3. Проверки
```

```
Во FreeBSD и Ubuntu следует выполнить эти шаги

gate# kinit user

gate# klist
gate# kdestroy

server# kinit user
server# klist
server# kdestroy
```

SSPI/GSSAPI

Security Support Provider Interface — универсальное API используемое в операционных системах Windows для решения задач аутентификации, причем в качестве провайдеров могут выступать такие протоколы как: NTLM, Kerberos, SSL и т. п.

SSPI является проприетарной реализацией стандартного GSSAPI (Generic Security Services Application Program Interface).

Лабораторная работа №5

1. Применение протокола GSSAPI для сервиса sshd

Во FreeBSD выполните следующие шаги

```
[gate:~] # kadmin -l
kadmin> add -r host/server.corpX.un
...
kadmin> ext -k /usr/home/student/server.keytab host/server.corpX.un

kadmin> quit

[gate:~] # chown student ~student/server.keytab

На сервере выполните следующее
[server:~] # scp student@gate:server.keytab .
[server:~] # ktutil copy server.keytab /etc/krb5.keytab
[server:~] # touch /etc/srvtab
[server:~] # ktutil list
...
[server:~] # cat /etc/ssh/sshd_config
...
GSSAPIAuthentication yes
…

На шлюзе выполните следующие клиентские настройки
[gate:~] # cat /etc/ssh/ssh_config
...
GSSAPIAuthentication yes
…
```

В Ubuntu выполните следующие шаги

```
root@gate:~# kadmin.local
kadmin.local:  addprinc -randkey host/server.corpX.un
...
kadmin.local:  ktadd -k /home/student/server.keytab host/server.corpX.un
kadmin.local:  quit
root@gate:~# chown student ~student/server.keytab
```

На сервере выполните следующее

```
root@server:~# scp student@gate:server.keytab .
root@server:~# ktutil
ktutil: rkt server.keytab
ktutil: list
ktutil: wkt /etc/krb5.keytab
ktutil: quit
root@server:~# cat /etc/ssh/sshd_config
...
GSSAPIAuthentication yes
…
```

На шлюзе выполните следующие клиентские настройки

```
root@gate:~# cat /etc/ssh/ssh_config
...
GSSAPIAuthentication yes
…
```

2. Настройка pam_kerberos для сервиса sshd

Во FreeBSD выполните следующий шаг

```
[gate:~] # cat /etc/pam.d/sshd
...
#auth    requisite       pam_opieaccess.so  no_warn allow_local
auth     sufficient      pam_krb5.so        no_warn try_first_pass
#auth    sufficient      pam_ssh.so         no_warn try_first_pass
```

В Ubuntu выполните следующие шаги

```
root@gate:~# apt-get install libpam-krb5
root@gate:~# cat /etc/pam.d/sshd
...
auth     sufficient      pam_krb5.so
# Standard Un*x authentication.
...
```

3. Проверка

Во FreeBSD выполните следующий шаг (под пользователем user)

```
user@gate$ ssh -vv server.corpX.un
```

В Ubuntu выполните следующий шаг (под пользователем user)

```
$ ssh -vv server.corpX.un
```

Авторизация

В современных операционных системах наиболее распространены следующие механизмы авторизации:

1. использование полномочий
2. использование списка контроля доступа (ACL)

LDAP

Lightweight Directory Access Protocol — «облегчённый протокол доступа к каталогам») — это сетевой протокол для доступа к службе каталогов X.500, разработанный IETF как облегчённый вариант разработанного ITU-T протокола DAP. LDAP — относительно простой протокол, использующий TCP/IP и позволяющий производить операции авторизации (bind), поиска (search) и сравнения (compare), а также операции добавления, изменения или удаления записей. Обычно LDAP-сервер принимает входящие соединения на порт 389 по протоколам TCP или UDP. Для LDAP-сеансов, инкапсулированных в SSL, обычно используется порт 636.

Всякая запись в каталоге LDAP состоит из одного или нескольких атрибутов и обладает уникальным именем (DN — англ. Distinguished Name). Уникальное имя может выглядеть, например, следующим образом: «cn=Иван Петров, ou=Сотрудники, dc=example, dc=com». Уникальное имя состоит из одного или нескольких относительных уникальных имен (RDN — англ. Relative Distinguished Name), разделённых запятой. Относительное уникальное имя имеет вид ИмяАтрибута=значение. На одном уровне каталога не может существовать двух записей с одинаковыми относительными уникальными именами. В силу такой структуры уникального имени записи в каталоге LDAP можно легко представить в виде дерева.

Запись может состоять только из тех атрибутов, которые определены в описании класса записи (object class), которые, в свою очередь, объединены в схемы (schema). В схеме определено, какие атрибуты являются для данного класса обязательными, а какие — необязательными. Также схема определяет тип и правила сравнения атрибутов. Каждый атрибут записи может хранить несколько значений.

В протоколе LDAP определены следующие операции для работы с Каталогом:

- Операции подключения/отключения
    - Подключение (*bind*) — позволяет ассоциировать клиента с определённым объектом Каталога (фактическим или виртуальным) для осуществления контроля доступа для всех прочих операций чтения/записи. Для того, чтобы работать с Каталогом, клиент обязан пройти аутентификацию как объект, отличительное имя (Distinguished Name) находится в пространстве имён, описываемом Каталогом. В запросе операции *bind* клиент может не указывать отличительное имя, в таком случае будет осуществлено подключение под специальным псевдонимом *anonymous* (обычно это что-то наподобие гостевой учетной записи с минимальными правами)
    - Отключение (*unbind*) — позволяет клиенту в рамках сеанса соединения с LDAP-сервером переключиться на аутентификацию с новым отличительным именем. Команда unbind возможна только после аутентификации на сервере с использованием bind, в противном случае вызов unbind возвращает ошибку
- Поиск (*search*) — чтение данных из Каталога. Операция сложная, на вход принимает

множество параметров, среди которых основными являются:

- База поиска (*baseDN*) — ветка DIT, от которой начинается поиск данных
- Глубина поиска (*scope*) — может иметь значения (в порядке увеличения охватываемой области): base, one, sub
    - base — поиск непосредственно в узле — базе поиска
    - one — поиск по всем узлам, являющимся прямыми потомками базового в иерархии, то есть лежащим на один уровень ниже него
    - sub — поиск по всей области, нижележащей относительно базы поиска (baseDN)
- Фильтр поиска (*searchFilter*) — это выражение, определяющее критерии отбора объектов каталога, попадающих в область поиска, задаваемую параметром scope. Выражение фильтра поиска записывается в обратной (префиксной) польской нотации, состоящей из логических (булевых) операторов и операндов, в свою очередь являющихся внутренними операторами сопоставления значений атрибутов LDAP (в левой части) с выражениями (в правой части) с использованием знака равенства.

Логические операторы представлены стандартным «набором»: **&** (логическое «И»), | (логическое «ИЛИ») и **!** (логическое «НЕ»).

Пример фильтра поиска:

```
(&(!(entryDN:dnSubtreeMatch:=dc=Piter,dc=Russia,ou=People,dc=example,dc=com))
(objectClass=sambaSamAccount)
(|(sn=Lazar*)(uid=Nakhims*)))
```

- Операции модификации — позволяют изменять данные в Каталоге, при этом в понятие модификации входит как добавление, удаление и перемещение записей целиком, так и редактирование записей на уровне их атрибутов. Подтипы модификации:
    - Добавление (*add*) — добавление новой записи
    - Удаление (*delete*) — удаление записи
    - Модификация RDN (*modrdn*) — перемещение/копирование записи
    - Модификация записи (*modify*) — позволяет редактировать запись на уровне её атрибутов,
        - добавляя новый атрибут или новое значение многозначного атрибута (add)
        - удаляя атрибут со всеми его значенями (delete)
        - заменяя одно значение атрибута на другое (replace)
        - а также увеличивая (уменьшая) значение атрибута в рамках атомарной операции (increment)
- Операция сравнения (*compare*) — позволяет для определённого отличительного имени сравнить выбранный атрибут с заданным значением

Операция запроса возможностей

В стандарте LDAP определена специальная операция, позволяющая клиентам получать информацию о поддерживаемых сервером версиях протокола и возможностях LDAP-сервера. Эта команда является надстройкой (расширением) для операции *search* и выполняется при следующем сочетании параметров последней:

- BIND анонимный

- База поиска *baseDN* указана как **""** (пустая строка)
- Глубина поиска *scope* указана как **base**
- Фильтр поиска: **(objectClass=*)**
- Перечень запрашиваемых атрибутов: либо явное перечисление, либо «+» (*ВНИМАНИЕ*! «*» не покажет значения служебных атрибутов, содержащих всю полезную информацию)

Например, при использовании LDAP-клиента из поставки OpenLDAP команда запроса возможностей может выглядеть как:

```
ldapsearch -x -H ldap://host:port -LLL -b "" -s base '(objectClass=*)'
suppotedControls supportedCapabilities
```

Операция запроса схемы

Для запроса информации о действующей схеме LDAP-каталога прежде необходимо выполнить Операцию запроса возможностей, получив значение атрибута *subschemaSubentry*.

```
ldapsearch -x -H ldap://host:port -LLL -s base -b ""
'(objectClass=*)' subschemaSubentry
```

Полученное значение используется в качестве *Отличительного имени* базы поиска (*baseDN*) в Операции запроса схемы, которую можно описать так:

- BIND анонимный, либо полный. Большинство серверов каталогов поддерживают запрос схемы без предварительного BIND, но, есть исключения (например, Active Directory);
- База поиска *baseDN* равна значению атрибута *subschemaSubentry*, возвращаемого Операцией запроса возможностей;
- Глубина поиска *scope* указана как **base**;
- Фильтр поиска: **(objectClass=*)**;
- Перечень запрашиваемых атрибутов: явное перечисление атрибутов (attributeTypes, objectClasses) возможно для всех серверов каталогов, в случае OpenLDAP и некоторых других (OpenDS,ApacheDS и т.д.) возможно указание «+»;

Например, при использовании LDAP-клиента из поставки OpenLDAP Операция запроса схемы может выглядеть так:

```
ldapsearch -x -H ldap://host:port -LLL -s base -b "cn=Subschema"
'(objectClass=*)' ldapSyntaxes matchingRules
```

Реализации LDAP

Серверная часть

LDAP является широко используемым стандартом доступа к службам каталогов. Из свободно распространяемых открытых реализаций наиболее известен сервер OpenLDAP, из проприетарных — поддержка протокола имеется в Active Directory — службе каталогов от компании Microsoft, предназначенной для централизации управления сетями Windows. Сервер IBM Lotus Domino в своем составе также имеет службу LDAP. Свои реализации служб каталогов, поддерживающие LDAP как протокол доступа, предлагают и другие крупные компании, например, Novell и Sun.

Клиентская часть

В качестве клиентов LDAP выступают как адресные книги почтовых клиентов, так и back-end'ы различных сетевых служб (серверы SMTP, Samba, UTS и т. д.).

Формат данных LDIF

LDAP Data Interchange Format (LDIF, Формат обмена данными LDAP) — формат представления записей службы каталогов или их изменений в текстовой форме. Записи каталога или их изменения представляются набором LDIF-записей, по одной на каждую запись каталога или изменение. LDIF-файл может содержать записи только одного типа, то есть только представление записей каталога или только представление изменений записей каталога.

LDIF был разработан в начале 90-х годов Тимом Хоузом (en:Tim Howes), Марком Смитом (Mark C Smith) и Гордоном Гудом (Gordon Good) в Мичиганском университете и был доработан и дополнен в конце 90-х для использования с LDAP версии 3. Эта, более поздняя версия формата, получила номер версии 1, была официально специфицирована IETF в RFC 2849, опубликована в июне 2000 года и в настоящее время имеет статус предложенного стандарта.

За прошедшие годы предложено множество расширений LDIF. Одно из них официально специфицировано IETF и опубликовано в RFC 4525. Также ожидаются публикации других расширений.

Формат записей каталога

Записи каталога представляются группами строк, разделенных пустой строкой, при этом каждая строка в группе представляет отдельное значение атрибута записи. Первая строка в группе должна представлять уникальное имя записи. Значение атрибута записывается в 7-битной кодировке ASCII и отделяется от его имени символом «:». Значения, не подходящие под эту кодировку, записываются в кодировке base64 и отделяются от имени атрибута символами «::». Также значение атрибута можно задать из внешнего ресурса, указав его единый указатель и отделяя от имени атрибута символами «:<». Схема file:// обязательна для всех реализаций и означает, что значение атрибута без изменений читается из указанного файла.

```
dn: <уникальное_имя>
<имя_атрибута>: <значение_атрибута>
<имя_атрибута>:: <base64_значение_атрибута>
<имя_атрибута>:< <url>

dn: <уникальное_имя>
<имя_атрибута>: <значение_атрибута>
<имя_атрибута>: <значение_атрибута>
```

Утилиты, использующие LDIF

- Пакет OpenLDAP включает утилиты, использующие LDIF для экспорта записей LDAP-сервера (ldapsearch), добавление записей на LDAP-сервер (ldapadd) и модификацию записей LDAP-сервера (ldapmodify).
- Netscape Communicator и Mozilla Application Suite используют LDIF в качестве одного

из форматов импорта и экспорта адресных книг. Следует отметить, что Yahoo! Mail, также использующий LDIF для экспорта адресной книги, неправильно кодирует некоторые символы. Например, амперсанд (&), который должен использоваться как есть, кодируется как специальный символ HTML (&). В результате, «John & Jane Doe» из адресной книги Yahoo! Mail становятся «John & Jane Doe» в адресной книге например Thunderbird'a.

- Microsoft Windows 2000 Server и Windows Server 2003 включают основанную на LDIF утилиту командной строки (LDIFDE), предназначенную для импорта и экспорта данных Active Directory.
- JXplorer — кроссплатформенное приложение с открытым исходным кодом, позволяющее просматривать LDIF-файлы и вносить простые изменения.

Ограничения LDIF

Значения многозначных атрибутов не могут быть непосредственно заменены. Необходимо сначала удалить значения атрибута, а затем использовать **«add:»** несколько раз чтобы вставить все необходимые значения.

Поля LDIF

`dn: уникальное имя`
Относится к имени, однозначно идентифицирующему запись каталога.

`dc: доменное имя`
Относится к каждому домену из полного имени. Например www.google.com должно быть записано как DC=www,DC=google,DC=com

o: organizationName

Относится к организации описанной в данной записи каталога.

`ou: organizational unit`
Относится к организационному подразделению (иногда — к группе пользователей), частью которого является пользователь. Если пользователь относится более чем к одной группе, то это можно записать в виде OU= Lawyer,OU= Judge.

`cn: common name`
Относится к имени объекта (имя человека; переговорной комнаты; рецепту; названию должности), о котором выполнен запрос.

Пример простой записи каталога с несколькими атрибутами:

```
dn: cn=The Postmaster,dc=example,dc=com
objectClass: organizationalRole
cn: The Postmaster
```

Лабораторная работа №6

Авторизация с использованием LDAP-сервера

1. Установка, настройка и запуск LDAP-сервера

Во FreeBSD следует выполнить следующие шаги

```
[gate:~] # pkg_add -r openldap24-server
```

```
[gate:~] # cat /usr/local/etc/openldap/slapd.conf
...
include         /usr/local/etc/openldap/schema/cosine.schema
include         /usr/local/etc/openldap/schema/nis.schema
...
suffix                "dc=corpX,dc=un"
rootdn                "cn=admin,dc=corpX,dc=un"
...

[gate:~] # cat /etc/rc.conf
...
slapd_enable="YES"
...

[gate:~] # /usr/local/etc/rc.d/slapd start

[gate:~] # rehash
```

В Ubuntu шаги будут следующими

```
root@gate:~# apt-get install slapd ldap-utils
root@gate:~# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/cosine.ldif
root@gate:~# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/nis.ldif
root@gate:~# cat config.ldif

# Load dynamic backend modules
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulepath: /usr/lib/ldap
olcModuleload: back_hdb

# Database settings
dn: olcDatabase=hdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcHdbConfig
olcDatabase: {1}hdb
olcSuffix: dc=corpX,dc=un
olcDbDirectory: /var/lib/ldap
olcRootDN: cn=admin,dc=corpX,dc=un
olcRootPW: secret
olcDbConfig: set_cachesize 0 2097152 0
olcDbConfig: set_lk_max_objects 1500
olcDbConfig: set_lk_max_locks 1500
olcDbConfig: set_lk_max_lockers 1500
olcDbIndex: objectClass eq
olcLastMod: TRUE
olcDbCheckpoint: 512 30
olcAccess: to attrs=userPassword by dn="cn=admin,dc=corpX,dc=un" write by
anonymous auth by self write by * none
olcAccess: to attrs=shadowLastChange by self write by * read
olcAccess: to dn.base="" by * read
olcAccess: to * by dn="cn=admin,dc=corpX,dc=un" write by * read

root@gate:~# ldapadd -Y EXTERNAL -H ldapi:/// -f config.ldif
```

2. Работа с LDAP-каталогом

В FreeBSD/Linux следует выполнить следующие шаги

```
gate# cat organization.ldif
dn: dc=corpX,dc=un
objectClass: dcObject
objectClass: organization
o: Students corporation X
dc: corpX
```

Из dcObject наследуется атрибут dc

Из organization наследуется атрибут o

```
gate# ldapadd -x -D "cn=admin,dc=corpX,dc=un" -w secret -f organization.ldif
```

3. Чтение каталога

В FreeBSD/Linux выполните следующий шаг

```
gate# ldapsearch -x -b "dc=corpX, dc=un"
```

4. Заполнение каталога информацией о пользователях

В FreeBSD/Linux выполните следующее

```
gate# cat users.ldif
dn: ou=users,dc=corpX,dc=un
objectClass: organizationalUnit
ou: users

dn: ou=groups,dc=corpX,dc=un
objectClass: organizationalUnit
ou: groups

dn: cn=user,ou=groups,dc=corpX,dc=un
objectClass: posixGroup
cn: user
gidnumber: 10001

dn: uid=user,ou=users,dc=corpX,dc=un
objectClass: account
objectClass: posixAccount
uid: user
cn: User user from LDAP
loginshell: /bin/sh
uidnumber: 10001
gidnumber: 10001
homedirectory: /home/user
gecos: User user from LDAP
userpassword: *
```

5. Импорт LDIF-файлов в каталог

В FreeBSD/Linux выполните следующее

```
gate# ldapadd -x -D "cn=admin,dc=corpX,dc=un" -w secret -f users.ldif
```

6. Поиск информации в LDAP-каталоге

В FreeBSD/Linux выполните следующее

```
gate# ldapsearch -x -b"dc=corpX,dc=un" "uid=user"
```

7. Удаление информации из ldap каталога

В FreeBSD/Linux выполните следующее

```
gate# ldapdelete -x -D "cn=admin,dc=corpX,dc=un" -w secret "uid=user,ou=users,dc=corpX,dc=un"
```

8. Тестирование доступности каталога с клиентов

Во FreeBSD выполните следующие шаги

```
[server:~] # pkg_add -r openldap24-client
[server:~] # pkg_add -r nss_ldap
[server:~] # rehash
[server:~] # cat /usr/local/etc/nss_ldap.conf
host gate
base dc=corpX,dc=un
nss_base_passwd               ou=users,dc=corpX,dc=un?one
nss_base_group                ou=groups,dc=corpX,dc=un?one

[server:~] # cat /etc/nsswitch.conf
...
passwd:         files ldap
group:          files ldap
...
[server:~] # ldapsearch -x -b"dc=corpX,dc=un" -h gate "uid=user
```

В Ubuntu выполните следующее

```
root@server:~# apt-get install ldap-utils
root@server:~# apt-get install libnss-ldap
...
LDAP server Uniform Resource Identifier: ldap://gate/
Distinguished name of the search base: dc=corpX,dc=un
LDAP version to use: 3
Make local root Database admin: No
Does the LDAP database require login? No
...
root@server:~# less /etc/ldap.conf
root@server:~# cat /etc/nsswitch.conf
...
passwd:         files ldap
group:          files ldap
shadow:         files ldap
...
root@server:~# ldapsearch -x -b"dc=corpX,dc=un" -h gate "uid=user
```

Настройка стенда для работы в составе Active Directory

Загрузите Windows 2003 и залогиньтесь под учетной записью Administrator с паролем test.

В первую очередь следует настроить сеть следующим образом:

Start→Control Panel→Network Connection→Local Area Connection→Properties

```
IP: 192.168.X.20/24
Gateway: 192.168.X.1
DNS: 192.168.X.1
```

Start→My Computer→Properties→Computer Name→Change

ad

И перезагрузите Windows 2003..

Для установки Active Directory выполните следующее:

Start→Run→dcpromo→Full DNS Name

```
corpX.un
```

Dns Registration Diagnostic→Install and configure server on this computer →…→Restore Password

```
password
```

→Install and configure DNS…

```
C:\Distrs\i386\...
```

→Finish→Restart Now

Настройка DNS сервера

В Windows 2003 в оснастке dnsmgmt добавьте следующие записи

```
gate   A  192.168.X.1
server  A  192.168.X.10
win2k3  A  192.168.X.20
winxp   A  192.168.X.30
```

На шлюзе в настройках DNS-сервера необходимо изменить настройки зоны corpX.un следующим образом:

```
zone "corpX.un" {
        type forward;
        forwarders {
                192.168.X.20;
        };
};
```

Не забудьте перезапустить DNS-сервера на шлюзе и в Windows 2003.

Для проверки настроек в командной строке Windows 2003 (cmd.exe) выполните следующие команды:

```
C:\>nslookup gate.corpX.un
...
Name:    gate.corpX.un
Address:  192.168.X.1

C:\>nslookup 192.168.X.1
...
Name:    gate.corpX.un
Address:  192.168.X.1

C:\>nslookup win2k3.corpX.un
...
Name:    win2k3.corpX.un
Address:  192.168.X.20

C:\>nslookup 192.168.X.20
...
Name:    win2k3.corpX.un
Address:  192.168.X.20
```

Включение Windows XP в домен CORP20

В первую очередь следует настроить сеть следующим образом:

Start→Control Panel→Network Connection→Local Area Connection→Properties

```
IP: 192.168.X.30/24
Gateway: 192.168.X.1
DNS: 192.168.X.1
```

Пуск→Мой компьютер→Свойства→Имя компьютера→Изменить→Имя компьютера
`winxp`

И перезагрузите Windows XP...

Для включения Windows XP в домен выполните следующее

Пуск→Мой компьютер→Свойства→Имя компьютера→Изменить→Является членом домена:

`corpX.un`

И перезагрузите Windows XP...

Не забудьте включить пользователя домена CORPX.UN/user в группу локальных администраторов:

- Входим локальным пользователем
- Пуск→Мой компьютер→Управление
- Добавляем доменного пользователя в группу локальных администраторов
- Регистрируемся доменным пользователем

KERBEROS аутентификация в Microsoft AD

В операционной системе Windows 2003 Центр распределения ключей (Key Distribution Center, KDC) реализован как служба домена. В качестве базы данных учетных записей он использует Active Directory. Кроме того, некоторые данные о пользователях поступают в него из глобального каталога (Global Catalog).

Как и в других реализациях протокола Kerberos, центр KDC Windows 2003 представляет собой единый процесс, объединяющий две службы:

- Служба аутентификации Authentication Service (AS). Эта служба выдает билеты на выдачу билетов (билеты TGT). Прежде, чем получить билет на обслуживание, сетевой клиент должен запросить первоначальный билет TGT, обратившись для этого к службе аутентификации того домена, где находится учетная запись пользователя.
- Служба выдачи билетов Ticket-Granting Service (TGS). Эта служба выдает билеты на доступ к другим службам своего домена или к службе выдачи билетов доверяемого домена. Чтобы обратиться в службу TGS, клиенту нужно сначала войти в контакт со службой выдачи билетов того домена, где находится учетная запись службы, представить свой билет TGT и запросить нужный билет. Если у клиента нет билета TGT, который открывает доступ к данной службе выдачи билетов, он может

воспользоваться процессом переадресации (referral process). Начальной точкой этого процесса является служба того домена, где находится учетная запись пользователя, а конечной – служба выдачи билетов домена, где находится учетная запись требуемой службы.

Центр KDC, как и служба каталогов Active Directory, имеется в каждом домене. Обе службы автоматически запускаются подсистемой LSA (Local Security Authority – распорядитель локальной безопасности), которая установлена на контроллере домена. Они работают в пространстве процессов этого распорядителя. Ни одну из этих служб остановить невозможно. Чтобы гарантировать постоянный доступ к KDC и Active Directory, в Windows 2003 предусмотрена возможность развертывания в каждом домене нескольких равноправных контроллеров домена. При этом запросы на аутентификацию и на выдачу билета, адресованные службе KDC данного домена, может принимать любой контроллер домена.

В доменах Windows 2003 служба KDC является абонентом безопасности. Как и предусмотрено документом RFC 1510, в этом качестве она выступает под именем krbtgt. Учетная запись абонента безопасности для нее создается автоматически при организации нового домена; эту запись нельзя ни изменить, ни переименовать. Пароль учетной записи KDC также присваивается автоматически, а затем регулярно меняется на плановой основе вместе с паролями доверенных учетных записей домена (domain trust account). Пароль учетной записи KDC используется при вычислении секретного ключа, необходимого для шифрования и расшифрования генерируемых этой службой билетов TGT. Пароль же доверенной учетной записи домена необходим для расчета междоменных (межобластных) ключей, которые используются для шифрования билетов переадресации.

Все экземпляры службы KDC одного домена используют единую учетную запись абонента безопасности с именем krbtgt. При обращении к центру распределения ключей домена клиент должен указать как имя абонента безопасности krbtgt, так и имя домена. Эти сведения приводятся и в билетах, где идентифицируют службу, выдавшую данный билет. Подробная информация о формах имен и адресных конвенциях приведена в документе RFC 1510.

База данных учетных записей

База данных, которая необходима службе KDC для получения информации относительно абонентов безопасности, хранится в каталоге Active Directory. Каждый абонент здесь представлен в виде учетной записи. Криптографические ключи, применяемые для связи с пользователем, компьютером или службой, хранятся в виде атрибутов объекта учетной записи конкретного абонента безопасности.

Серверами службы каталога Active Directory являются только контроллеры доменов. На каждом из них хранится копия каталога, в которую можно вносить изменения. Это позволяет создавать новые учетные записи, изменять пароли и корректировать состав групп, обратившись на любой контроллер домена. Изменения, внесенные в одну реплику каталога, автоматически переносятся на все другие его реплики. Правда, Windows 2003 не использует для этой цели протокол репликации Kerberos. Копирование и распространение информации, хранящейся в Active Directory, производится посредством собственного протокола децентрализованной репликации (multi-master replication protocol), разработанного корпорацией Microsoft , причем пересылка ее осуществляется по защищенным каналам между контроллерами доменов.

Физическим хранением информации об учетных записях управляет агент DSA (Directory System Agent – агент системы каталога). Этот защищенный процесс интегрирован с подсистемой LSA, работающей на контроллере домена. Клиенты службы каталога никогда не

получают прямого доступа к хранилищу с данными об учетных записях. Любой клиент, которому нужна информация из каталога, должен воспользоваться для подключения к DSA интерфейсом ADSI (Active Directory Service Interface – интерфейс служб Active Directory). Лишь после этого он может искать, читать и записывать объекты и их атрибуты.

Запросы на доступ к объектам или атрибутам каталога подлежат проверке в системе управления доступом Windows 2003. Подобно объектам файлов и папок в файловой системе NTFS, объекты Active Directory защищаются посредством ACL (Access Control List – список контроля доступа), где содержится информация о том, кто и каким способом имеет право обращаться к объектам. Правда, в объектах Active Directory, в отличие от файлов и папок, список контроля доступа имеется для каждого атрибута. Самым секретным элементом любой учетной записи, конечно же, является пароль. В объекте учетной записи атрибут пароля хранит не сам пароль, а криптографический ключ, полученный на его основе, однако этот ключ представляет для взломщика не меньшую ценность. По этой причине доступ к атрибуту пароля предоставляется исключительно владельцу учетной записи. Такого права не имеет никто другой, даже администратор. Прочесть информацию о пароле или изменить ее могут только процессы с привилегией Trusted Computer Base, которые работают в контексте безопасности LSA.

В Windows 2003 приняты меры и против возможного взлома учетной записи изнутри, то есть, злоумышленником с доступом к резервным копиям доменного контроллера. Чтобы помешать этому, атрибут пароля в объекте учетной записи подвергается второму шифрованию с использованием системного ключа. Этот криптографический ключ может храниться на сменном носителе, для которого нетрудно предусмотреть дополнительные меры защиты, либо на контроллере домена. Где хранить системный ключ, – выбирает администратор, одновременно определяя алгоритм шифрования атрибутов пароля.

Политика Kerberos

В среде Windows 2003 политика Kerberos определяется на уровне домена и реализуется службой KDC домена. Она сохраняется в каталоге Active Directory как подмножество атрибутов политики безопасности домена. По умолчанию вносить изменения в политику Kerberos имеют право только члены группы администраторов домена.

В политике Kerberos предусматриваются:

- Максимальный срок действия пользовательского билета (Maximum user ticket lifetime). Под «пользовательским билетом» здесь имеется в виду билет на выдачу билетов (билет TGT). Значение задается в часах и по умолчанию равно 10 час.
- Максимальное время, в течение которого допускается обновление пользовательского билета (Maximum lifetime that a user ticket can be renewed). Задается в сутках; по умолчанию составляет 7 суток.
- Максимальный срок действия служебного билета (Maximum service ticket lifetime). Под «служебным билетом» здесь имеется в виду сеансовый билет. Значение этого параметра должно быть более 10 минут, но менее значения Maximum user ticket lifetime. По умолчанию оно равно 10 час.
- Максимально допустимое отклонение в синхронизации компьютерных часов (Maximum tolerance for synchronization of computer clocks). Указывается в минутах; по умолчанию равно 5 мин.
- Проверка ограничений при входе пользователя в систему (Enforce user logon

restrictions). Если этот пункт помечен флажком, служба KDC анализирует каждый запрос на сеансовый билет и проверяет, имеет ли данный пользователь право на локальный вход в систему (привилегия Log on Locally) или на доступ к запрашиваемому компьютеру через сеть (привилегия Access this computer from network). Такая проверка занимает дополнительное время и может замедлить предоставление сетевых услуг, поэтому администратору предоставляется право ее отключения. По умолчанию она включена.

Делегирование аутентификации

Как уже отмечалось, политика Kerberos для домена может разрешать делегированную аутентификацию методом передаваемых билетов, однако такое разрешение вовсе не обязательно должно распространяться на всех пользователей или на все компьютеры. Воспользовавшись атрибутом учетной записи конкретного пользователя, администратор может запретить передачу его удостоверения с одного сервера на другой. Кроме того, можно запретить передачу удостоверения любого пользователя, установив соответствующий атрибут в учетной записи какого-либо компьютера. При необходимости можно также запретить делегирование аутентификации всем пользователям или всем компьютерам, входящим в организационное подразделение внутри домена.

Поставщик поддержки безопасности Kerberos

Протокол аутентификации Kerberos реализован в форме поставщика поддержки безопасности (security support provider, сокращенно SSP) – динамически подключаемой библиотеки, входящей в состав операционной системы. В Windows 2003 предусмотрен также поставщик SSP для протокола NTLM. По умолчанию оба эти компонента загружаются подсистемой LSA на компьютер Windows 2003 одновременно с загрузкой операционной системы. Каждый из этих поставщиков позволяет производить аутентификацию как входа в сеть, так и клиент-серверных подключений. Какой поставщик будет использован в том или ином конкретном случае, зависит от возможностей компьютера, с которым устанавливается связь, однако в первую очередь система всегда пытается воспользоваться поставщиком Kerberos SSP.

После того, как подсистема LSA создаст контекст безопасности для интерактивного пользователя, может потребоваться еще один экземпляр Kerberos SSP. Его загрузку производит процесс, запущенный в пользовательском контексте безопасности, чтобы обеспечить поддержку подписи и заверение сообщений.

Системные службы и приложения транспортного уровня обращаются к SSP через интерфейс SSPI (Security Support Provider Interface – интерфейс поставщика поддержки безопасности). Он представляет собой интерфейс Win32®, позволяющий просмотреть список доступных на системе поставщиков, выбрать один из них и использовать его для организации аутентифицированного подключения. Выполнение всех этих функций производится в SSPI стандартизированными методами, своего рода подпрограммами «черного ящика», которыми разработчик может пользоваться, даже не разбираясь в тонкостях самого протокола.

Кэш-память удостоверений

На компьютерах под управлением Windows 2003 билеты и ключи, полученные из службы KDC, сохраняются в кэш-памяти удостоверений – области оперативной памяти, защищенной подсистемой LSA. Содержимое кэш-памяти удостоверений не сбрасывается в файл подкачки за исключением случаев нехватки ресурсов системы. Выход абонента безопасности из

системы и отключение самой системы приводят к уничтожению всех хранящихся в памяти объектов.

Управление кэш-памятью удостоверений возложено на Kerberos SSP, который работает в контексте безопасности подсистемы LSA. Каждый раз, когда возникает необходимость в получении билета или ключа, либо в их обновлении, LSA обращается к Kerberos SSP, который и выполняет эту задачу.

В подсистеме LSA сохраняются также копии хешей паролей интерактивных пользователей. Если в ходе сеанса истекает срок действия пользовательского билета TGT, Kerberos SSP использует эту копию для получения нового билета. Это делается в фоновом режиме, без прекращения сеанса. Пароль здесь находится во временном хранении, его локальная копия уничтожается после прекращения сеанса работы пользователя.

По иному обстоит дело с хешами паролей для служб и компьютеров. Как и в прежних версиях Windows NT, они сохраняются в безопасной области системного реестра компьютера. Реестр используется также для хранения хешей паролей для учетных записей пользователей на локальных системах, однако локальные учетные записи пригодны исключительно для доступа к компьютерам, работающим в автономном режиме, но не через сеть.

Разрешение имен DNS

Согласно документу RFC 1510, для пересылки сообщений между клиентами и службой KDC должен использоваться протокол IP. Чтобы послать первоначальный запрос аутентификации, Kerberos SSP, установленному на клиентском компьютере, нужно найти адрес центра распределения ключей того домена, где находится учетная запись пользователя. Другими словами, необходимо знать имя сервера со службой KDC в доменной системе имен DNS. Если это имя может быть преобразовано в IP-адрес, Kerberos SSP направляет на него свое сообщение, если же нет, – выдается сигнал ошибки, указывающий на то, что домен не найден.

Служба KDC устанавливается на всех контроллерах домена Windows 2003. Кроме функций серверов KDC эти контроллеры выполняют еще и функции серверов LDAP (Lightweight Directory Access Protocol – облегченный протокол доступа к каталогам). Обе эти службы регистрируются в записях указателя служб системы DNS (записях ресурсов SRV). Чтобы найти контроллер домена, клиенту достаточно запросить на сервере DNS запись ресурса SRV с именем _ldap._tcp.dc._msdcs.ИмяДоменаDNS. А запросив запись ресурса SRV с именем _kerberos._udp.ИмяДоменаDNS, клиент получит в ответ адрес службы KDC. Компьютеры под управлением Windows 2003 могут обращаться и в те области Kerberos, которые не входят в домены Windows 2003. Здесь служба KDC размещается на доменных контроллерах, работающих под управлением других операционных систем, поэтому имена DNS для таких серверов KDC приходится сохранять в реестре клиентского компьютера. В этом случае Kerberos SSP сначала находит в реестре доменное имя DNS области пользователя, а затем запрашивает соответствующий сервер DNS и преобразует это имя в IP-адрес.

Лабораторная работа №7

Использование Kerberos при аутентификации по SSH в домене

Настройка Kerberos-клиента в FreeBSD/Linux

```
gate# cat /etc/krb5.conf
[libdefaults]
```

```
        default_realm = CORPX.UN
```

```
gate# kinit user
```

Настройка библиотеки pam kerberos для сервиса ssh

Во FreeBSD выполните следующий шаг

```
gate# cat /etc/pam.d/sshd
...
auth   sufficient      pam_krb5.so     no_warn try_first_pass
auth   required        pam_unix.so     no_warn try_first_pass
...
```

В Ubuntu выполните следующие шаги

```
root@gate:~# apt-get install libpam-heimdal

root@gate:~# cat /etc/pam.d/sshd
...
auth     sufficient      pam_krb5.so
# Standard Un*x authentication.
...
```

Затем войдите на gate по ssh с логином user и паролем Pa$$w0rd

NTLM аутентификация в Microsoft AD

Программы перебора паролей используют те же алгоритмы, что и ОС, хэшируя комбинацию и отправляя ее серверу. Теоретически, процесс полного перебора (брутфорса) может занимать довольно много времени, но если система будет использовать LM-хэш, задача становится тривиальной.

Некоторые клиентские программы могут без участия пользователя произвести NTLM аутентификацию при условии, что ее поддерживает сервер. Поэтому хэш можно получить даже при помощи telnet, просто подключившись к нужному порту. Программы, позволяющие подобрать пароли, имеются в свободном доступе - John the Ripper (www.openwall.com/john), LCP (www.lcpsoft.com/russian) и L0phtCrack LC5. Последнюю в открытом доступе найти нельзя; после приобретения Astake компанией Symantec она исчезла с сайта, но достаточно ввести в гугле «LC5 download», и .... Например, LCP может сама захватывать передаваемые по сети пакеты, или импортировать учетные записи с локального и удаленного компьютера, выполнять импорт файлов SAM, Sniff и созданных другими утилитами (LC, LCS и PwDump). Реализовано три типа атак для подбора паролей по хэшам: атака по словарю, гибридная атака по словарю и brute force.

Именно по этим причинам на смену протоколу сетевой аутентификации NTLMv1 пришел NTLMv2. Новая реализация во многом похожа на своего предшественника, но хэш образует более устойчивый к взлому алгоритм HMAC-MD5, а при запросе используется 128-разрядный ключ. Чтобы сделать невозможными некоторые атаки, где проигрываются ранее записанные учетные данные, в NTLMv2 введена метка времени. В доменной среде NTLMv2 применяется вместо Kerberos в ситуациях: аутентификация по IP-адресу, в рабочей группе, если клиент не принадлежит домену или текущему лесу (в том случае если не установлено доверительное отношение), и при невозможности использования Kerberos (например, блокировка firewall). Есть еще варианты, но о них чуть дальше.

Запретить хранение LM-хэшей в Windows 2k/XP/2k3 довольно просто: для этого необходимо

добавить в реестр параметр NoLMHash типа DWORD в раздел HKLM\SYSTEM\CurrentControlSet\Control\Lsa со значением 1 (подробнее о запрещении хранения LM-хэшей можно прочитать в статье KB299656). Кстати, если длина пароля более 15 символов, то сохраненный LM-хэш нельзя использовать для аутентификации, а значит, он непригоден и для взлома.

Параметр типа DWORD LMCompatibilityLevel в этом же разделе позволяет разрешить LM-аутентификацию только по запросу сервера или вообще запретить. Здесь указывается одно из 6 значений:

- 0 (по умолчанию) - использовать LM- и NT-ответы, NTLMv2 отключен;
- 1 - использовать при необходимости NTLMv2;
- 2 - только NT-ответ;
- 3 - только NTLMv2;
- 4 - отказывать контроллеру домена в LM-аутентификации;
- 5 - отказывать контроллеру домена в LM- и NT-аутентификации, только NTLMv2.

В доменной среде проще воспользоваться возможностями групповой политики (Group Policy Object), выбрав в редакторе GPO пункт «Параметры безопасности» по маршруту «Конфигурация компьютера → Конфигурация Windows → Параметры безопасности → Локальные политики» и активировав политику «Network security: Do not store LAN Manager hash value on next password change» (не хранить хэш-значения LAN Manager при следующей смене пароля). Начиная с Vista, она действует по умолчанию. Политика «Network Security: LAN Manager authentication level» определяет настройки NTLM; установив ее в «NTLM2 responses only», можно запретить использование LM и NTLMv1.

В Vista и выше LM-хэши и NTLMv1 также поддерживаются, параметр LmCompatibilityLevel установлен в 3, – то есть, по умолчанию для недоменной аутентификации используется NTLMv2, а политики запрещают их использование. Использование NTLM в доменной среде определяет политика «Network Security: Restrict NTLM: NTLM authentication for this domain». По умолчанию в Win2k8R2 она не установлена. Если в сети нет клиентов с устаревшими системами, ее можно переключить в «Deny all», полностью запретив использование этого протокола. Как вариант, при помощи этого параметра можно запретить NTLM при доступе к серверу домена или учетной записи.

Лабораторная работа №8

1. Настройка службы winbindd

Во FreeBSD выполняются следующие шаги

```
[gate:~] # pkg_add -r samba3
[gate:~] # cat /etc/rc.conf
…
nmbd_enable="NO"
smbd_enable="NO"
winbindd_enable="YES"
…
[gate:~] # cd /usr/local/etc/
[gate:~] # cat smb.conf
[global]
        workgroup = CORPX
        security = DOMAIN
```

```
        winbind use default domain = Yes
```

В Ubuntu эти шаги следующие

```
root@gate:~# apt-get install winbind
root@gate:~# cd /etc/samba
root@gate:~# cat smb.conf
[global]
        workgroup = CORPX
        security = DOMAIN
        winbind use default domain = Yes
```

2. Регистрация службы winbindd в домене

В FreeBSD/Ubuntu выполните следующее

```
gate# net rpc join -U Administrator
Administrators's password:
Joined domain CORPX
```

3. Запуск службы winbindd

Во FreeBSD выполняется следующий шаг

```
[gate:~] # /usr/local/etc/rc.d/samba start
```

В Ubuntu эти шаги следующие

```
root@gate:~# /etc/init.d/bind9 restart
root@gate:~# /etc/init.d/winbind restart
```

4. Проверка

Выполняется в FreeBSD/Ubuntu

```
gate# ntlm_auth --username=user
password:
NT_STATUS_OK: Success (0x0)
```

5. Использование NTLM аутентификации для proxy сервер squid

Во FreeBSD выполняются следующие шаги

```
[gate:~] # pkg_add -r squid
[gate:~] # chown root:squid /var/db/samba/winbindd_privileged/
[gate:~] # cat /etc/rc.conf
...
squid_enable=yes
[gate:~] # rehash
[gate:~] # squid -z
[gate:~] # cd /usr/local/etc/squid
[gate:~] # cat squid.conf
...
# for freebsd uncomment
# auth_param ntlm program /usr/local/bin/ntlm_auth --helper-protocol=squid-2.5-ntlmssp
...
acl inetuser proxy_auth REQUIRED
http_access allow inetuser
# http_access allow localnet
[gate:~] # /usr/local/etc/rc.d/squid start
```

В Ubuntu эти шаги следующие

```
root@gate:~# apt-get install squid
root@gate:~# cd /etc/squid
root@gate:~# cat squid.conf
...
# for linux uncomment
# auth_param ntlm program /usr/bin/ntlm_auth --helper-protocol=squid-2.5-ntlmssp
...
acl inetuser proxy_auth REQUIRED
http_access allow inetuser
# http_access allow localnet
root@gate:~# /etc/init.d/squid restart
```

5. В Windows XP в настройках браузера укажите работу через прокси 192.168.X.1 порт 3128 и проверьте доступность сайтов

6. Настройка библиотеки pam на использование winbind

Во FreeBSD выполняется следующий шаг

```
[gate:~] # cat /etc/pam.d/sshd
...
auth       sufficient      /usr/local/lib/pam_winbind.so
auth       required        pam_unix.so             no_warn try_first_pass
```

В Ubuntu эти шаги следующие

```
root@gate:~# apt-get install libpam-modules
root@gate:~# more /etc/pam.d/sshd
...
auth       sufficient  pam_winbind.so
# Standard Un*x authentication.
```

7. Залогиньтесь на gate по ssh под учетной записью user с паролем Pa$$w0rd

LDAP авторизация в Microsoft AD

Для поддержки некоторых атрибутов UNIX необходимо расширить схему Active Directory, что в ОС Windows 2003 Server делается легко. Откройте панель управления на вашем контроллере домена и откройте оснастку Add or Remove Programs > Add/Remove Windows Components. Найдите компонент Active Directory Services и в его составе выберите подкомпонент Identity Management for UNIX (если вы используете более раннюю версию Windows, этот компонент иногда называется Server for NIS). Установите это программное обеспечение, и схема LDAP будет расширена; в диалоговых окнах пользователей появится вкладка UNIX Attributes, которую мы скоро будем использовать.
В оснастке Active Directory Users and Computers отредактируйте свойства группы безопасности Domain Users. Обратите внимание на вкладку UNIX Attributes. Назначьте группу и NIS-домен группе Domain Users. Это сделает группу видимой для UNIX-систем. Данная группа будет являться для пользователя домена основной.

Лабораторная работа №9

LDAP авторизация в Microsoft AD

1. Получение информации о пользователе в AD по протоколу LDAP

В FreeBSD/Linux выполните следующее

```
gate# ldapsearch -x -h ad -b "dc=corpX,dc=un" -D  \
"cn=Administrator,cn=Users,dc=corpX,dc=un" -W "sAMAccountName=user"
```

2. Модификация схемы AD

В Windows 2003 устанавливаем NIS server из пакета SFU 3.5

Опции инсталяции:

```
Custom:
        Server for NIS
```

Добавляем группу “guser”

Устанавливаем ее UNIX свойство

```
gid: 10001
```

Добавляем UNIX атрибуты пользователю “user”

```
uid: 10001
группа по умолчанию: guser
home dir: /home/user
```

3. Удаляем учетную запись пользователя user из системы и все его файлы

Во FreeBSD выполните

```
[gate:~] # rmuser user && rm -rf /usr/home/user
```

В Ubuntu выполните

```
root@gate:~# userdel user && rm -rf /home/user
```

```
4. Проверка того, что пользователь не опознается
В FreeBSD/Linux выполните
gate# id user
В результате вы получите следующее
id: user: No such user
```

5. Настройка gate на использование AD

Во FreeBSD выполните следующее

```
[gate:~] # pkg_add -r nss_ldap
```

```
[gate:~] # cat /usr/local/etc/nss_ldap.conf
host 192.168.X.20  # for restart local dns
base dc=corpX,dc=un
binddn cn=Administrator,cn=Users,dc=corpX,dc=un
bindpw password
scope sub
nss_base_passwd         cn=Users,dc=corpX,dc=un?one
nss_base_group          cn=Users,dc=corpX,dc=un?one
```

```
nss_map_objectClass posixAccount User
nss_map_attribute uid msSFU30Name
nss_map_attribute uniqueMember msSFU30PosixMember
nss_map_attribute homeDirectory msSFU30HomeDirectory
nss_map_objectClass posixGroup Group
nss_map_attribute gidNumber msSFU30GidNumber
nss_map_attribute uidNumber msSFU30UidNumber
nss_map_attribute loginShell msSFU30LoginShell

[gate:~] # cat /etc/nsswitch.conf
...
group:  files ldap
passwd: files ldap

[gate:~] # pkg_add -r pam_mkhomedir

[gate:~] # cat /etc/pam.d/sshd
...
# session
session         required       /usr/local/lib/pam_mkhomedir.so
...
```

В Ubuntu выполните следующее

```
root@gate:~# apt-get install libnss-ldap
Ответы по умолчанию

root@gate:~# cat /etc/ldap.conf
host 192.168.X.20  # for restart local dns
base dc=corpX,dc=un
binddn cn=Administrator,cn=Users,dc=corpX,dc=un
bindpw password
scope sub
nss_base_passwd         cn=Users,dc=corpX,dc=un?one
nss_base_group          cn=Users,dc=corpX,dc=un?one
nss_map_objectClass posixAccount User
nss_map_attribute uid msSFU30Name
nss_map_attribute uniqueMember msSFU30PosixMember
nss_map_attribute homeDirectory msSFU30HomeDirectory
nss_map_objectClass posixGroup Group
nss_map_attribute gidNumber msSFU30GidNumber
nss_map_attribute uidNumber msSFU30UidNumber
nss_map_attribute loginShell msSFU30LoginShell

root@gate:~# cat /etc/nsswitch.conf
...
group:  files ldap
passwd: files ldap
shadow: files ldap


root@gate:~# apt-get install libpam-modules

root@gate:~# cat /etc/pam.d/sshd
...
session    required     pam_mkhomedir.so
# Standard Un*x session setup and teardown.
...
```

6. Попробуйте зайти на gate по ssh под учетной записью user с паролем Pa$$w0rd

7. Проверка учетной записи user

```
В FreeBSD/Linux выполните
gate# id user
```

В результате вы получите следующее

```
uid=10001(user) gid=10001(guser) groups=10001(guser)
```