SQUID и Kerberos-авторизация через Active Directory

По мотивам http://www.k-max.name/windows/active-directory-as-kdc-nfsv4/#krb5

Почему именно Kerberos?

- Kerberos-авторизация в данный момент является самой надежной
- Нет необходимости в установке и настройке Samba (а эта штука периодически любит мозг выносить)

Исходные данные

- Сервер PROXY с установленным Ubuntu Server (в описанном примере это 14.04.1)
  - Сеть настроена верно, прописаны DNS/Gateway, все сервера сети разыменовываются верно (прямая и обратная зона настроены правильно)
  - 192.168.1.1 - IP нашего PROXY
  - 192.168.1.3 - IP DNS-сервера нашей сети
- Домен Active Directory
  - MYDOMAIN.RU - имя домена
  - ntp.mydomain.ru - сервер времени домена
  - dc.mydomain.ru - контроллер домена

Требования

- Авторизация через Active Directory
- Разным пользователям - разные уровни доступа к сайтам, включая ограничение по скорости
- Черный и белый списки сайтов
- Отчет по использованию интернета с указанием ФИО пользователей

Установка

Первое что нужно сделать - это присоединить наш сервер к домену Active Directory. Для этого в обязательном порядке необходимо синхронизировать время на всех серверах.

Установка службы времени и синхронизация

```
# apt-get install ntp
```

Редактируем /etc/ntp.conf - отключаем все сервера по умолчанию и добавляем свой:

```
server ntp.mydomain.ru
```

И перезапускаем сервис:

```
# service ntpd restart
```

Kerberos и домен AD

Установка Kerberos

Для подключения нам нужно установить Kerberos

Устанавливаем Kerberos:

```
# apt-get install krb5-admin-server krb5-config krb5-kdc krb5-user
```



ОБЯЗАТЕЛЬНО устанавливаем пакет libsasl2-modules-gssapi-mit для правильной работы с Active Directory

```
# apt-get install libsasl2-modules-gssapi-mit
```

Настройка Kerberos

Удаляем или переименовываем оригинальный файл /etc/krb5.conf и создаем новый со следующими настройками:



Регистр написания - очень важен!

```
[libdefaults]
    default_realm = MYDOMAIN.RU

[realms]
    MYDOMAIN.RU = {
        kdc = dc.mydomain.ru
    }
```

Получаем билет от контроллера домена:

```
# kinit domain_admin
```

где domain_admin - логин с правами Domain Administrator, а пароль будет запрошен в приглашении.

Смотрим получен ли билет:

```
# klist
```

Создаем keytab-файл на контроллере домена

keytab-файл это «связка ключей», файл содержащий в себе одну или несколько записей - ключей, которые используются вместо логина/пароля при запросе доступа у сервера KDC к какому-либо ресурсу. Другими словами, машина предоставляет данный файл серверу KDC как подтверждение своей достоверности, когда запрашивает у контроллера домена (KDC) доступ к какому-либо ресурсу (например доступ к службе или сетевому каталогу). В большинстве случаев, при настройке какой-либо службы в связке с Kerberos проблемы возникают именно на данном этапе, т.к. именно этот файл является связующим звеном между Windows 2008 и Linux (в нашем случае - Debian). Проблемы обычно связаны с различными типами шифрования, которые поддерживает Windows, но не поддерживает Linux и наоборот.

На контроллере домена даем команду (запустить из-под администратора):

```
C:\Windows\system32\ktpass.exe /princ HTTP/proxy.mydomain.ru@MYDOMAIN.RU
/mapuser proxy@MYDOMAIN.RU /crypto ALL /ptype KRB5_NT_PRINCIPAL /pass
+rndpass /out C:\tmp\proxy.keytab
```

Где proxy.mydomain.ru - FQDN нового прокси-сервера, а proxy@MYDOMAIN.RU - логин пользователя, от имени которого будут делаться запросы в AD

Теперь этот ключ Kerberos необходимо БЕЗОПАСНО скопировать на наш прокси (например чрез WinSCP) и приступить к следующему шагу.

Установка Squid

Устанавливаем:

```
# apt-get install squid
```

И приводим конфигурационный файл /etc/squid3/squid.conf к такому виду (описание групп доступа и их настройки см. ниже):

```
# Аутентификация в Active Directory
auth_param negotiate program /usr/lib/squid3/negotiate_kerberos_auth -s
HTTP/proxy.mydomain.ru@MYDOMAIN.RU
auth_param negotiate children 10
auth_param negotiate keep_alive on
external_acl_type inet_medium ttl=300 negative_ttl=60 %LOGIN
/usr/lib/squid3/ext_kerberos_ldap_group_acl -g Internet@MYDOMAIN.RU
external_acl_type inet_full ttl=300 negative_ttl=60 %LOGIN
/usr/lib/squid3/ext_kerberos_ldap_group_acl -g Internet_Full@MYDOMAIN.RU
external_acl_type inet_low ttl=300 negative_ttl=60 %LOGIN
/usr/lib/squid3/ext_kerberos_ldap_group_acl -g Internet_Low@MYDOMAIN.RU
```

```
# Стандартные порты
acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl CONNECT method CONNECT

# Белый и черный список
acl white_list url_regex -i "/etc/squid3/white_list"
acl black_list url_regex -i "/etc/squid3/black_list"

# Определяем группы доступа
acl my_full external inet_full
acl my_medium external inet_medium
acl my_low external inet_low

# Перечень сетей
acl all src all
acl our_networks src 192.168.102.0/24

# Авторизация требуется ОБЯЗАТЕЛЬНО, без нее никого не пускать
acl nt_group proxy_auth REQUIRED

# Стандартные разрешения
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localhost manager
http_access deny manager

# Права доступа для наших групп пользователей
http_access allow my_low white_list
http_access deny my_low
http_access deny my_medium black_list
http_access allow my_medium
http_access allow my_full

# Разрешаем локалхост
http_access allow localhost

# Запрещаем все остальное
http_access deny all

# Ограничение пропускной способности интернет-канала
```

```
delay_pools 3
delay_class 1 1
delay_class 2 1
delay_class 3 1
delay_parameters 1 -1/-1 -1/-1
delay_parameters 2 384000/384000
delay_parameters 3 32000/32000
delay_access 1 allow Full
delay_access 2 allow Medium
delay_access 3 allow Low

# Порты прокси-сервера
http_port 192.168.1.1:3128
http_port 192.168.1.1:3127 transparent

# Выделяем 3,5 Гб памяти для прокси
cache_mem 3584 MB

# Выделяем место на жестком диске для хранения файлов кэша
cache_dir ufs /var/spool/squid3 100 16 256

# Куда и в каком объеме будем писать логи
access_log /var/log/squid3/access.log
logfile_rotate 100
coredump_dir /var/spool/squid3

# Настройки кэширования
refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern ^gopher:        1440    0%      1440
refresh_pattern -i (/cgi-bin/|\?) 0     0%      0
refresh_pattern (Release|Packages(.gz)*)$      0       20%     2880
refresh_pattern -i \.(gif|png|jpg|jpeg|ico)$    3600    90%     43200
refresh_pattern .               0       20%     4320

# Запрещаем отображение версии прокси-сервера и имени
httpd_suppress_version_string on
visible_hostname PROXYSERVER

# Включаем русский язык для сообщений сервера
error_directory /usr/share/squid3/errors/Russian-1251
error_default_language ru

# Принудительно задаем желаемый DNS-сервер
dns_nameservers 192.168.1.3
dns_v4_first on
```

Проверяем конфиг на ошибки:

```
# squid3 -k parse
```

Если парсер не обнаружил никаких ошибок, можно применить новую конфигурацию Squid:

```
# squid3 -k reconfigure
```

или

```
# service squid3 restart
```

Squid + Kerberos

Чтобы Squid и соответствующие хелперы умели работать с Kerberos надо добавить в скрипт запуска Squid (или создать файл /etc/default/squid3) следующие строки:

```
KRB5_KTNAME=/etc/squid3/proxy.keytab
export KRB5_KTNAME
```

Squid, Kerberos и доменные группы

По умолчанию в Squid 3.3.8-1ubuntu6.1 не включен хелпер ext_kerberos_ldap_group_acl - а это значит что нельзя будет узнавать в какие группы включен пользователь. Для исправления это недостатка необходимо врукопашную собрать этот хелпер. Для этого необходимо установить пакет build-essential и все что ему нужно:

```
# apt-get install build-essential
```

Качаем с сайта http://www.squid-cache.org/ нужную версию Squid, в моем случае это http://www.squid-cache.org/Versions/v3/3.3/squid-3.3.8.tar.bz2 и разворачиваем куда-нибудь, например /usr/src/squid/squid-3.3.8

В этой папке делаем следующее:

```
# ./configure --enable-external-acl-helpers="kerberos_ldap_group" --
libexecdir=/usr/lib/squid3 --sysconfdir=/etc/squid3
# make
```

Теперь все готово для сборки самого хелпера:

```
# cd /usr/src/squid/squid-3.3.8/helpers/external_acl/kerberos_ldap_group
# make
```

Если все прошло без ошибок, в папке /usr/src/squid/squid-3.3.8/helpers/external_acl/kerberos_ldap_group можно найти готовый собранный хелпер ext_kerberos_ldap_group_acl и скопировать его в /usr/lib/squid3/ не забыв сделать исполняемым:

```
# chmod +x ext_kerberos_ldap_group_acl
# cp ext_kerberos_ldap_group_acl /usr/lib/squid3/
```

Немного подробнее о группах доступа

По условиям у нас в Active Directory есть 3 группы доступа:

- Internet_Full - пользователи, входящие в нее имеют доступ на любые сайты, скорость доступа не лимитирована
- Internet - пользователи имеют доступ на любые сайты, кроме запрещенных в файле /etc/squid3/black-list со скоростью 4 Мбит/сек (48000 Кбайт/сек * 8 = 4 Мбит/сек) на всю группу
- Internet_Low - пользователи имеют доступ только на сайты, перечисленные в файле /etc/squid3/white_list со скоростью 256 Кбит/сек (32000 Кбайт/сек * 8 = 256 Кбит/сек) на всю группу

Сопоставление групп идет в следующем кусочке:

```
# Определяем группы доступа
acl my_full external inet_full
acl my_medium external inet_medium
acl my_low external inet_low
```

Правила для iptables



Примечание: при работе в прозрачном (transparent) режиме аутентификация пользователей невозможна! Подробнее: http://wiki.squid-cache.org/Features/Authentication#Authentication_in_interception_and_transparent_modes

Завернуть всех пользователей (если PROXY является еще и шлюзом):

```
iptables -t nat -A PREROUTING -s 192.168.1.0/24 -p tcp –dport 80 -j REDIRECT –to-port 3127
```

Завернуть если PROXY - отдельный сервер:

```
iptables -t nat -A PREROUTING -s 192.168.1.0/24 -p tcp –dport 80 -j DNAT –to 192.168.1.1:3127
```

где 192.168.1.1 - IP нашего PROXY

Так же можно средствами Group Policy прописать прокси-сервер требуемым объектам GPO

Статистика пользователей

Остался последний этап - отображать красивую статистику посещений сайтов пользователями. Самый простой и красивый инструмент для этого - LightSquid.

Устанавливаем

```
# apt-get install apache2 lightsquid
```

Настраиваем Apache

Приводим файл /etc/apache2/conf-available/lightsquid.conf в такой вид (доступ на просмотр даем всем):

```
Alias   /lightsquid/    /usr/lib/cgi-bin/lightsquid/

<Location "/lightsquid/">
        Options +ExecCGI
        Require local
        Require ip 192.168.1.0/24
</Location>
```

Раскомментируем строку 219 в файле /etc/apache2/mods-enabled/mime.conf

```
AddHandler cgi-script .cgi .pl
```

Запускаем все это дело:

```
# a2enmod cgi
# a2enconf lightsquid
# service apache2 restart
```

Настраиваем LightSquid

Необходимо поставить пакет libnet-ldap-perl:

```
# apt-get install libnet-ldap-perl
```

Очень хочется чтобы в отчете фигурировало полное ФИО пользователя, а не доменное имя или IP-адрес. Для этого нужно заменить оригинальный файл /usr/share/lightsquid/ip2name/ip2name.squidauth следующим:

```
#contributor: esl
#specialy for squid with turned on user authentication
#simple version

use strict;
use warnings;
use Net::LDAP;
use Encode;
```

```
my $ldap;
my $message;
my %hDisplayName;

sub StartIp2Name() {
        my $server = "ldap://dc.mydomain.ru";
        $ldap = Net::LDAP->new( $server );
        return if(!defined $ldap);
        $message = $ldap->bind(q(mydomain.ru\lightsquid), password =>
"my_password");
}

sub Ip2Name($$$) {
  # $Lhost,$user,$Ltimestamp
  my $Lhost=shift;
  my $user =shift;
  $user    =URLDecode($user); #decode user name
  $user = substr($user, 0, index($user, "\@mydomain.ru"));
  return $Lhost if ($user eq "-");
  return $user if (!defined $ldap);
  return $user if ($message->code());

  if (!defined $hDisplayName{$user})
  {

    my $result = $ldap->search(
    base        => "dc=mydomain,dc=ru",
    filter      =>
"(&(objectCategory=person)(objectClass=user)(sAMAccountName=" . $user .
"))",
    );

my $first_entry = $result->entry(0);
if (!defined $first_entry)
  {
    return $Lhost;
  }

my $pure_displayName = $first_entry->get_value("displayName");
$pure_displayName =~ s/ /_/g;
Encode::from_to($pure_displayName, 'utf-8', 'windows-1251');

  $hDisplayName{$user}=$pure_displayName;
  }

  return $hDisplayName{$user};
}


sub StopIp2Name() {
        return if (!defined $ldap);
```

```
        $message = $ldap->unbind;
}

#warning !!!
1;
```

В этом файле исправляем нижеуказанные строки на свои:

```
...
        my $server = "ldap://dc.mydomain.ru";
...
        $message = $ldap->bind(q(MYDOMAIN.RU\LightSquid), password =>
"PASSWORD");
...
    base        => "dc=MYDOMAIN,dc=RU",
...
```

Теперь в /etc/lightsquid/lightsquid.cfg включаем преобразование логина в ФИО:

```
$ip2name="squidauth"
```

Запускаем /usr/share/lightsquid/check-setup.pl и если все хорошо, можно запускать /usr/share/lightsquid/lightparser.pl

В папке /var/lib/lightsquid/report должны появиться отчеты, которые можно поглядеть по адресу: http://192.168.1.1/lightsquid/

Траблшутинг

Ошибка "kerberos_ldap_group: ERROR: Error while binding to ldap server with SASL/GSSAPI: Local error"

Необходимо установить библиотеку Cyrus-SASL, в убунте она называется libsasl2-modules-gssapi-mit

squid, AC, active directory, kerberos, krb, прокси, сквид, авторизация, домен, lightsquid, статистика

From:
https://wiki.rtzra.ru/ - **RTzRa's hive**

Permanent link:
https://wiki.rtzra.ru/software/squid/squid-active-directory-kerberos

Last update: **2017/05/09 18:34**