Организация интернет шлюза для локальной сети, с использованием кеширующего прокси сервера Squid, анализатора логов Sarg и кеширующего Dns сервера на базе Ubuntu server.

Подготовка дистрибутива.

Для этих целей мы выберем дистрибутив Ubuntu server ввиду его простоты и надежности.
Идем на http://www.ubuntu.com/getubuntu/download выбираем:
-Ubuntu 8.10 Desktop (the latest version) это последняя версия дистрибутива на момент написания статьи
-Choose a download location near you указываем страну с сервера которой будим скачивать.
-Computer Architecture выбираем архитектуру компьютера, по умолчанию выбираем 32bit version
И наконец Begin download!
После закачки дистрибутива записываем его на диск и приступим к установке.

Установка системы.

После загрузки компьютера с CD, появится меню состоящее из нескольких пунктов. Нажимаем F2 и выбираем **Русский** язык. Далее система большую часть сообщений выводит на русском языке. Выбираем страну,дальше можно выбрать клавиатуру. Система дает нам возможность с помощью списка или путем обнаружения из серии символов, которые необходимо ввести. Мы говорим, нет. Выбираем клавиатуру Russia, далее вариант переключения раскладки «по вкусу». Система начинает загружать драйвера и дополнительные компоненты для установки. Следующим шагом система пытается обнаружить в сети dhcp сервер и настроить сеть. Если нас не устраивают данные выдаваемые им или он попросту отсутствует, то далее мы можем выбрать пункт ***Настроить сеть вручную.*** Отвечаем на вопросы системы. Выберем часовой пояс для корректной работы.
И тут наступает ответственный момент разбивки дисков. Будим подразумевать, что в нашем распоряжении чистый винчестер. Для большинства случаев выбираем пункт ***Авто — использовать весь диск.*** Инсталлятор системы разбивает диска на два раздела / - корень системы, swap - файл подкачки. Дальше система копирует и распаковывает на винт базовые пакеты системы. После распаковки система задает вопрос о создании нового системного пользователя. Придумываем и вписываем имя. У нас спросили, хотим ли мы создать зашифрованный частным каталог,отвечаем нет. Следующий экран позволяет настроить доступ к прокси-сервера для доступа к менеджеру пакетов. Мы оставим его пустым, так как у нас есть прямой доступ к Интернету. В следующим окне отказываемся от автоматического обновления выбирая пункт ***No automatic updates.*** И так, мы дошли до выбора ПО. Для наших целей требуется DNS server и OpenSSH server. Вот и все, остальное система сделает за нас. Первый шаг к исполнении задуманного завершен.

Настройка системы.

После успешной загрузки системы, мы видим приглашение системы пройти авторизацию. Вводим логин созданного нами пользователя и его пароль. Мы в системе, но под непривилегированным пользователем, все команды связанные с настройкой сервера будит выполнять от имени root, вводим sudo bash и еще раз пароль. Теперь нам необходимо настроить систему «под себя».
Первым делом русифицируем ее.
Вводим в консоли:

```
apt-get install console-cyrillic          #инсталлируем шрифты для консоли
dpkg-reconfigure console-cyrillic
/etc/init.d/console-cyrillic start        #стартуем шрифты
```

Еще нам требуется обновить систему, но сначала сконфигурируем сеть. Будим считать что имеем 2 сетевых карты eth0 и eth1, смотрящих в локальную сеть и интернет соответственно. Адрес сервера в локальной сети 192.168.1.1. А вот тут рассмотрим два варианта. Провайдер выдал нам ip 195.195.195.195, основной шлюз 195.195.195.1.
Исправим файл:

vi /etc/network/interfaces

Приведем его к подобному виду:

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
        address 192.168.1.1
        netmask 255.255.255.0
auto eth1
iface eth1 inet static
        address 195.195.195.195
        netmask 255.255.255.0
        gateway 195.195.195.1
```

Или же если нужно использовать dhcp провайдера для поднятия eth1 пишем:

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
        address 192.168.1.1
        netmask 255.255.255.0
auto eth1
iface eth1 inet dhcp
```

Временно пропишем DNS провайдера в файле resolv.conf

vi /etc/resolv.conf

Указываем:
nameserver АДРЕС_ДНС_ПРОВАЙДЕРА
nameserver АДРЕС_ДНС_ПРОВАЙДЕРА

Далее следует перезагрузиться командой ***reboot***. После перезагрузи снова заходим в консоль с правами рута, с начало авторизировавшись под созданным пользователем, а потом набрав команду:

sudo bash

Проверим интернет командой:

ping mail.ru

Если нет ответа то смотрим пункты выше и перепроверяем введенные данные, при

благоприятном исходе выполняем следующее:

apt-get update #обновляем дерево пакетов
apt-get upgrade #обновляем пакеты
reboot #перезагрузка

Настраиваем DNS.

Установим bind9:

apt-get install bind9

Правим настройки в файле /etc/bind/named.conf.options

vi /etc/bind/named.conf.options

Приводим к виду:

```
options {
        directory "/var/cache/bind";
        listen-on port 53 {
                192.168.1.1;
        };

        forwarders {

                ДНС_СЕРВЕР_ПРОВАЙДЕРА_1;        #вписываем адреса DNS серверов
                ДНС_СЕРВЕР_ПРОВАЙДЕРА_2;        #одного сервера достаточно
                };
        };
logging {
category lame-servers {null; };
category edns-disabled { null; };
};
```

Запускаем DNS сервер.

/etc/init.d/bind9 start

Правим /etc/resolv.conf таким образом чтобы все DNS запросы система пропускала через свой же DNS сервер:

vi /etc/resolv.conf

Удаляем все записи и вписываем свой DNS сервер:

nameserver 192.168.1.1

Поднимаем прокси сервер.

Инсталлируем пакет:

apt-get install squid

Редактируем файл настроек squid:

vi /etc/squid/squid.conf

Ищем параметр http_port , и выставляем ему следующее:

http_port 192.168.1.1:3128 transparent

Этой строкой мы осуществляем прозрачное проксирование трафика идущего на 80 порт внешней сети.

icp_port 0

Отключаем возможность использования соседних прокси.

cache_mem 64 MB

Максимально число оперативной памяти занимаемое процессом.

maximum_object_size 4096 KB

Максимальный размер кешируемых объектов

minimum_object_size 0 KB

Соответственно минимальный размер

cache_dir ufs /var/cache/squid 100 16 256

Параметр указывает тип файловой системы для работы с кешем (ufs), расположение папки (/var/cache/squid), но нас интересует именно параметр со значением 100. Данный параметр задает максимальный обьем кеша на нашем диске. Можно повысить до разумных пределов если есть свободное место, я выставил себе 2048.
Следующий важный тег называется acl, который определяет листы(списки) доступа к серверу.
тип записи его следующий: acl aclname acltype string
где aclname - имя доступа, acltype - тип доступа, string - строка определяющая параметры доступа
подробнее про типы доступа можно узнать непосредственно в самом конфиге,я приведу лишь простой пример:

acl all src 192.168.1.0/24

описывает тип доступа под именем all для сети 192.168.1.0/24

visible_hostname localproxy

localproxy можете заменить на любое слово или словосочетание. Если не заполните данный параметр - squid будет очень сильно ругаться.
Перезапускаем сервис squid:

/etc/init.d/squid restart

Установка Sarg (Анализатор логов прокси сервера).

Инсталлируем Sarg и Apache2:

apt-get install sarg apache2

В Ubuntu настройки Apache можно не трогать, все будет работать с параметрами по умолчанию. В различных дистрибутивах и операционных системах файлы могут быть помещены в разные каталоги. В Ubuntu место дислокации - */etc/squid*, конфигурационный файл называется **sarg.conf**.

vi /etc/squid/sarg.conf

Указываем язык, возможные значения: Russian-koi8,
Russian_UTF-8, Russian-windows1251

language Russian_UTF-8
charset Cyrillic

Файл со статистикой, обрати внимание, что в некоторых
дистрах каталоги для Squid 3.0 называются squid3

access_log /var/log/squid3/access.log

Включаем построение графиков

graphs yes
graph_days_bytes_bar_color green

Каталог, в который помещаются отчеты

output_dir /var/www/squid-reports

Сортировка юзеров в выводе по USER CONNECT BYTES TIME
topuser_sort_field BYTES reverse
user_sort_field BYTES reverse

Тип отчета, включаем все
report_type topusers topsites sites_users users_sites date_time denied auth_failures site_user_time_date downloads

Создание отчетов производится при помощи скрипта **/usr/sbin/sarg-reports**, который запускается при помощи cron:

Запускаем ***crontab -e*** и вписываем следующее:

*00 08-18/1 * * * /usr/sbin/sarg-reports today*

*00 00 * * * /usr/sbin/sarg-reports daily*
*00 01 * * 1 /usr/sbin/sarg-reports weekly*
*30 02 1 * * /usr/sbin/sarg-reports monthly*

В итоге имеем статистику на Http://192.168.1.1/*squid-reports/*

Правила iptables.

Создаем файл в директории /root, например rules.sh и редактируем его:

vi /root/rules.sh

С таким наполнением:

```
INET_IFACE="eth1" # здесь имя инетовского интерфейса

LAN_IP="192.168.1.1"

LAN_IP_RANGE="192.168.1.0/24"

LAN_IFACE="eth0"

LO_IFACE="lo"

LO_IP="127.0.0.1"

IPTABLES="/sbin/iptables"

/sbin/depmod -a

/sbin/modprobe ip_tables

/sbin/modprobe ip_conntrack

/sbin/modprobe iptable_filter

/sbin/modprobe iptable_mangle

/sbin/modprobe iptable_nat

/sbin/modprobe ipt_LOG

/sbin/modprobe ipt_limit

/sbin/modprobe ipt_state

echo "1" > /proc/sys/net/ipv4/ip_forward

$IPTABLES -P INPUT ACCEPT

$IPTABLES -P OUTPUT ACCEPT

$IPTABLES -P FORWARD ACCEPT
```

$IPTABLES -t nat -F

$IPTABLES -F

$IPTABLES -X

$IPTABLES -A INPUT -s $LAN_IP_RANGE -d $LAN_IP -p tcp --destination-port 3128 -j ACCEPT

$IPTABLES -t nat -A PREROUTING -s $LAN_IP_RANGE -d $LAN_IP -p tcp --destination-port 80 -j ACCEPT

$IPTABLES -t nat -A PREROUTING -s $LAN_IP_RANGE -p tcp --destination-port 80 -j REDIRECT --to-port 3128

$IPTABLES -t nat -A POSTROUTING -o $INET_IFACE -j MASQUERADE # это включается при динамеческом ip, предыдущая строка выключается

$IPTABLES -A FORWARD -i $LAN_IFACE -o $INET_IFACE -m state --state NEW,ESTABLISHED -j ACCEPT

Разрешаем доступ к SSH серверу

$IPTABLES -A INPUT --protocol tcp --dport 22 -s $LAN_IP_RANGE -j ACCEPT

Разрешаем доступ к HTTP серверу на роутере

$IPTABLES -A INPUT --protocol tcp --dport 80 -s $LAN_IP_RANGE -j ACCEPT

Даем права на исполнение.

chmod +x /root/rules.sh

Запускаем скрипт:

/root/rules.sh

И сохраняем правила iptables:

iptables-save > /etc/iptables.rules

Дописываем сохраненные правила в /etc/network/interfaces после строк
address 192.168.1.1
netmask 255.255.255.0

Должно получиться:

auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
address 192.168.1.1

netmask 255.255.255.0
pre-up iptables-restore < /etc/iptables.rules
auto eth1
iface eth1 inet static
address 195.195.195.195
netmask 255.255.255.0
gateway 195.195.195.1

Чтобы форвардинг автоматически включался при запуске системы

Открываем файл:

vi /etc/sysctl.conf

и добавляем в него строчку:

net.ipv4.ip_forward = 1

Брокируем доступ в интрнет по mac адресу.

iptables -I FORWARD 1 -o eth1 -m mac --mac-source 32:43:25:25:25:42 -j DROP

Вот и все, шлюз с кеширующей проксей и днс готов, каждый день генерирующий отчеты по трафику.