https://www.youtube.com/watch?v=O1SI_ELNoZg
https://github.com/ksemaev/manuals
https://github.com/ksemaev/manuals/blob/master/docx/Proxy%20and%20firewall%20on%20Ubuntu.docx

Мануал по созданию простейшего прокси+firewall на Ubuntu

(поэтапно, как на уроках)

1) Ставим нужные пакеты:
sudo apt-get install bind9 squid3

2) Разрешаем проброс пакетов между сетевыми интерфейсами указанием в файле **/etc/sysctl.conf** ключа *net.ipv4.ip_forward=1*

3) Для упрощения отключаем родной firewall (позже по желанию включим):

sudo ufw disable

4) Проверяем что по умолчанию в iptables все разрешено и нет никаких правил:

sudo iptables -L

5) Включаем политику блокировки по умолчанию всех входящих и проходящих пакетов (исходящие можем оставить в разрешенных):

sudo iptables -P INPUT DROP

sudo iptables -P FORWARD DROP

6) Можем разрешить входящий пинг и проверить его из локалки:

sudo iptables -A INPUT -p icmp -j ACCEPT
sudo iptables -A INPUT -p tcp -m tcp --dport 22 -j ACCEPT

7) Можем создать промежуточную резервную копию правил iptables:

sudo iptables-save > iptables.backup

8) Разрешаем входящие и проходящие пакеты в рамках уже существующих соединений:

sudo iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

sudo iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT

9) Разрешаем пинг наружу, через хост firewall:

sudo iptables -A FORWARD -p icmp -j ACCEPT

10) Для того чтобы заработали пакеты, проходящие из локалки (у меня это 192.168.0.0/24) наружу нужно включить SNAT или, что проще в данном случае, правило маскарада (замена адреса источника пакетов из локальной сети на внешний адрес хоста firewall):

sudo iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -j MASQUERADE

11) Можем попробовать сохранить правила iptables:

sudo apt install iptables-persistent

sudo service iptables-persistent save

12) Можно, для того чтобы можно было разрешать DNS –запросы, разрешить проходящий DNS трафик (он работает по TCP и UDP). Если хочется использовать свой DNS сервер на этой же машине, то вместо это шага выполняется шаг № 17.

sudo iptables –A FORWARD –p tcp –m multiport --ports 53 –s 192.168.0.0/24 –j ACCEPT

sudo iptables –A FORWARD –p udp –m multiport --ports 53 –s 192.168.0.0/24 –j ACCEPT

13) Но чтобы два раза не вставать, лучше сделать сразу правило для всех необходимых протоколов. Поэтому я удаляю правило с DNS для tcp и создаю правило сразу для многих нужных протоколов (номера портов нужных протоколов можно легко нагуглить):

sudo iptables –D FORWARD –p tcp –m multiport --ports 53 –s 192.168.0.0/24 –j ACCEPT

sudo iptables –A FORWARD –p tcp –m multiport --ports 80,8080,110,5190,25,21,443 –s 192.168.0.0/24 –j ACCEPT

14) В качестве возможного варианта рассматриваем проброс портов вовнутрь сети, а именно: если мы хотим из интернета (то есть стучась во внешнюю сетевую карту firewall, в моем случае это enp0s3) попадать на какой-то веб-сервер, который находится в локальной сети по адресу 192.168.0.101 и прослушивает запросы по порту 80, то нам нужно создать правило DNAT (проброс порта) и разрешить проходящие пакеты по порту 80:

sudo iptables –t nat -A PREROUTING -i enp0s3 -p tcp -m tcp --dport 80 -j DNAT --to-destination 192.168.0.101

sudo iptables -A FORWARD -i enp0s3 -p tcp --dport 80 -j ACCEPT

15) Если у нас есть еще один веб-сервер в локалке (у меня 192.168.0.102), который использует тот же 80 порт, то нам нужно на firewall открыть какой-нибудь другой порт (у меня

в примере 81) и соединения к нему пробрасывать на нужный веб сервер по порту 80:

sudo iptables –t nat -A PREROUTING -i enp0s3 -p tcp -m tcp --dport 81 -j DNAT --to-destination 192.168.0.102:80

16) Можем попробовать сохранить правила командой:

sudo dpkg-reconfigure iptables-persistent

17) Для того чтобы самим разрешать DNS запросы на нашем сервере, нужно ему разрешить эти запросы прослушивать в iptables:

sudo iptables -A INPUT -p tcp -m multiport --ports 53 -s 192.168.0.0/24 -j ACCEPT
sudo iptables -A INPUT -p tcp -m multiport --ports 53 -s 192.168.0.0/24 -j ACCEPT

18) Также необходимо разрешить весь петлевой траффик, так как есть программы его использующие (т.е. трафик не выходящий за пределы нашего сервера, но проходящий через сетевые интерфейсы):

sudo iptables -A INPUT -i lo -j ACCEPT

19) Сервер bind в данной конфигурации будет тупо перенаправлять все DNS-запросы через себя на указанный ему DNS-сервер. Для этого в его конфиге нужно указать куда передавать трафик, по какому порту и сетевой карте слеушать запросы, и (если мы не хотим чтоб он сам пытался разрешить запрос) указать ему сразу делать перенаправление запроса. Все это осуществляется редактированием файла **/etc/bind/named.conf.options**:

forwarders { dns_server_ip; };
forward first;
Listen-on port 53 { lan_ipl; };
//dnssec-validation auto;

20) В текущей версии Ubuntu 16.04 управлять демоном bind пытается rndc, а ей для этого нужны ключи. Их можно создать командой:

sudo rndc-confgen -r /dev/urandom

Незакомментированную часть вывода следует отправить в файл **/etc/bind/rndc.conf**, а закоментированную в /**etc/bind/named.conf**

22) Теперь можно перезагрузить демон bind и проверить как сам сервер (и машины у которых он указан в качестве DNS) разрешают DNS запросы:

sudo service bind9 restart
nslookup ya.ru

---

23) Проверяю версию squid командой /***usr/sbin/squid -v*** и делаю резервную копию файла конфигурации:

sudo cp /etc/squid/squid.conf /etc/squid/squid.conf.backup

24) Файл конфигурации **/etc/squid/squid.conf** сильно отличается от версии к версии, но параметры в нем неизменны. В частности нам нужно указать:

- локальную сеть:

acl localnet src 192.168.0.0/24

- разрешить из нее доступ:

http_access allow localnet
http_access allow localhost

- указать адрес локальной сетевой карты, порт и опцию прозрачности (в старых версиях она называется не intercept a transparent):

http_port 192.168.0.10:3128 intercept

25) Для применения настроек squid следует перезапустить:
sudo service squid restart

26) Теперь следует убрать из правил iptables страрое правило проброса http и https портов, например:
sudo iptables -D FORWARD -s 192.168.0.0/24 -p tcp -m multiport --ports 80,8080,443,110,25,21 -j ACCEPT
sudo iptables -A FORWARD -s 192.168.0.0/24 -p tcp -m multiport --ports 110,25,21 -j ACCEPT

26) Затем создать правило приема трафика по порту прослушки squid и правила переадресации входящих http запросов на этот порт:
sudo iptables -A INPUT -s 192.168.0.0/24 -p tcp -m multiport --ports 3128 -j ACCEPT
sudo iptables -t nat -A PREROUTING -s 192.168.0.0/24 -p tcp -m multiport --dports 80,8080 -j REDIRECT --to-ports 3128

Для работы с https squid придется пересобрать: https://imbicile.pp.ru/ubuntu-16-04-prozrachnyj-squid-https/

27) Включаем в репозитории те, который начинаются с src: **/etc/apt/sources.list**

28) Обновляем информацию о репозиториях и ставим нужные для сборки пакеты:
apt-get update
apt-get install openssl devscripts build-essential dpkg-dev libssl-dev

29) Создаем директорию для сборки, переходим в нее (mkdir, cd) и качаем все что нужно:
apt-get build-dep squid3
chmod 777 squid3_3.5.12-1ubuntu7.2.dsc
apt-get source squid3

30) Настраиваем параметры сборки:
cd squid3-3.5.12
vim debian/rules
Дописываем опции:
--enable-ssl \
--enable-ssl-crtd \
--with-openssl \

31) Собираем пакет, ставим его устанавливая неразрешенные зависимости:
dpkg-buildpackage -d
cd ../
dpkg -i *.deb
apt-get install –f
dpkg -i *.deb
squid –v

32) Создаем необходимые сертификаты:
cd /etc/squid/
openssl req -new -newkey rsa:1024 -days 365 -nodes -x509 -keyout squidCA.pem -out squidCA.pem

33) Создаем конфиг:

cp squid.conf squid.conf.bak

cat squid.conf.bak | grep -v "^#" | grep -v "^$" > squid.conf

Вписываем:

acl localnet src 192.168.0.0/24

http_access allow localnet

http_port 192.168.20.1:3128 intercept

https_port 192.168.20.1:3129 intercept ssl-bump cert=/etc/squid/squidCA.pem

ssl_bump peek all

ssl_bump splice all

sslcrtd_program /usr/lib/squid/ssl_crtd -s /var/lib/ssl_db -M 4MB

34) Перезапускаем сквид:

squid -k parse

squid -k reconfigure

35) Перехватываем пакеты и разрешаем их:

sudo iptables -A INPUT -s 192.168.0.0/24 -p tcp -m multiport --ports 3129 -j ACCEPT

sudo iptables -t nat -A PREROUTING -s 192.168.0.0/24 -p tcp -m multiport --dports 443 -j REDIRECT --to-ports 3128