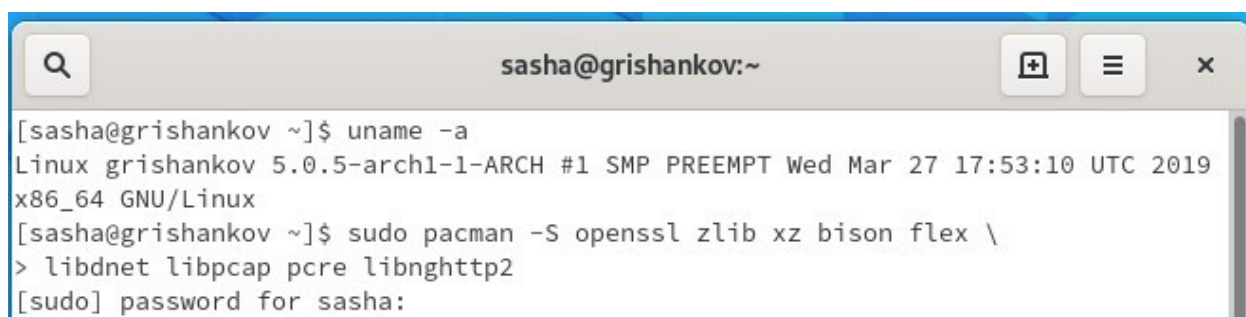


2. Установка системы IPS/IDS.

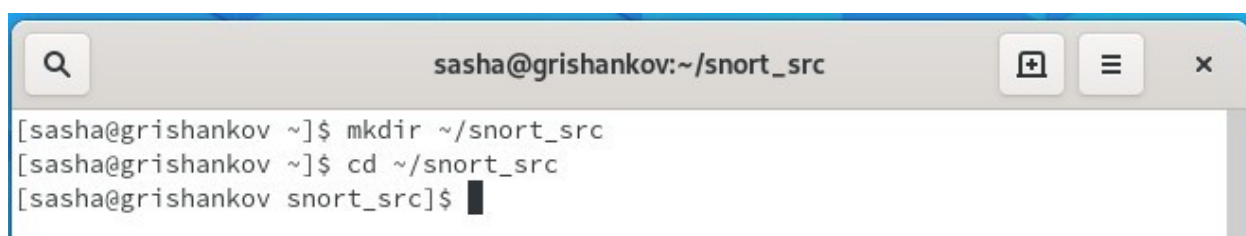
Была произведена установка дистрибутива Arch Linux. Команда `uname` отображает `hostname` – `grishankov`, версия ядра — `5.0.5`. Сперва следует произвести установку пакетов `openssl` (SSL & transport security), `zlib` (архивация), `xz` (библиотека для работы с архивами), `bison`(parser generator), `flex` (сканирование текста), `libdnet` (низкоуровневые сетевые операции), `libpcap` (захват пакетов), `pcre` (perl регулярные выражения), `libnghttp2`(http/2 уровня кадров) из репозитория Arch, с помощью команды `pacman -S [пакет1 пакет2 ...]`.



```
sasha@grishankov:~  
[sasha@grishankov ~]$ uname -a  
Linux grishankov 5.0.5-arch1-1-ARCH #1 SMP PREEMPT Wed Mar 27 17:53:10 UTC 2019  
x86_64 GNU/Linux  
[sasha@grishankov ~]$ sudo pacman -S openssl zlib xz bison flex \>  
> libdnet libpcap pcre libnghttp2  
[sudo] password for sasha:
```

Рисунок 1 - Установка пакетов

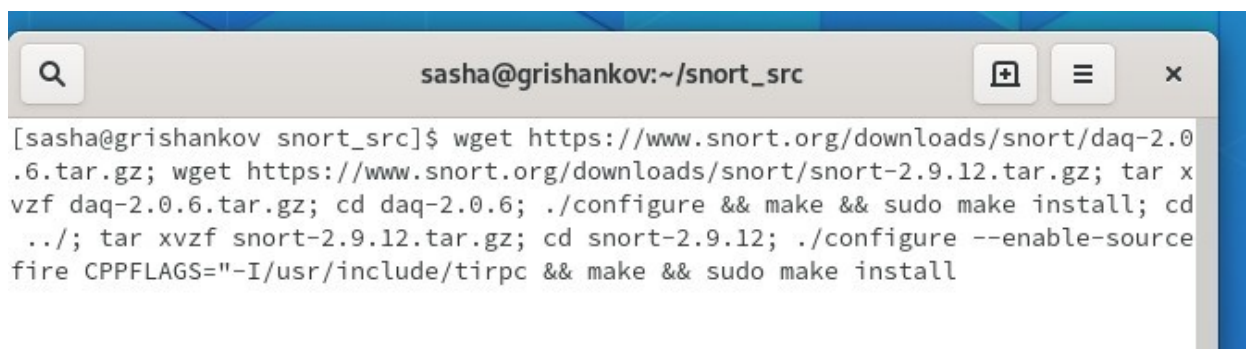
На скриншоте ниже был создан каталог для сохранения загруженных файлов tarball:



```
sasha@grishankov:~/snort_src  
[sasha@grishankov ~]$ mkdir ~/snort_src  
[sasha@grishankov ~]$ cd ~/snort_src  
[sasha@grishankov snort_src]$
```

Рисунок 2 - Создание каталога

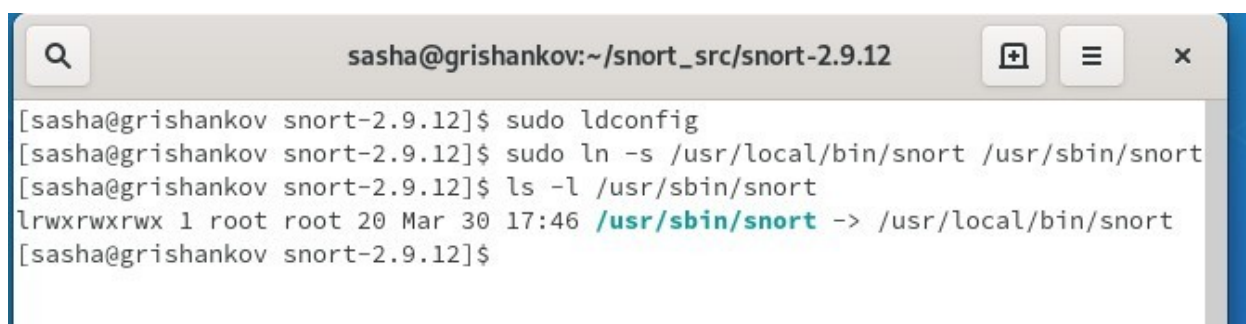
Следующей цепью команд производится загрузка DAQ и snort из источников, затем происходит разархивирование, переход в папку daq-2.0.6, конфигурация daq, сборка daq, установка daq. Аналогичные операции начиная с разархивирования происходят с загруженным источником snort.



```
[sasha@grishankov snort_src]$ wget https://www.snort.org/downloads/snort/daq-2.0.6.tar.gz; wget https://www.snort.org/downloads/snort/snort-2.9.12.tar.gz; tar xvfz daq-2.0.6.tar.gz; cd daq-2.0.6; ./configure && make && sudo make install; cd ../; tar xvfz snort-2.9.12.tar.gz; cd snort-2.9.12; ./configure --enable-source fire CPPFLAGS="-I/usr/include/tirpc && make && sudo make install
```

Рисунок 3 - Загрузка DAQ и Snort

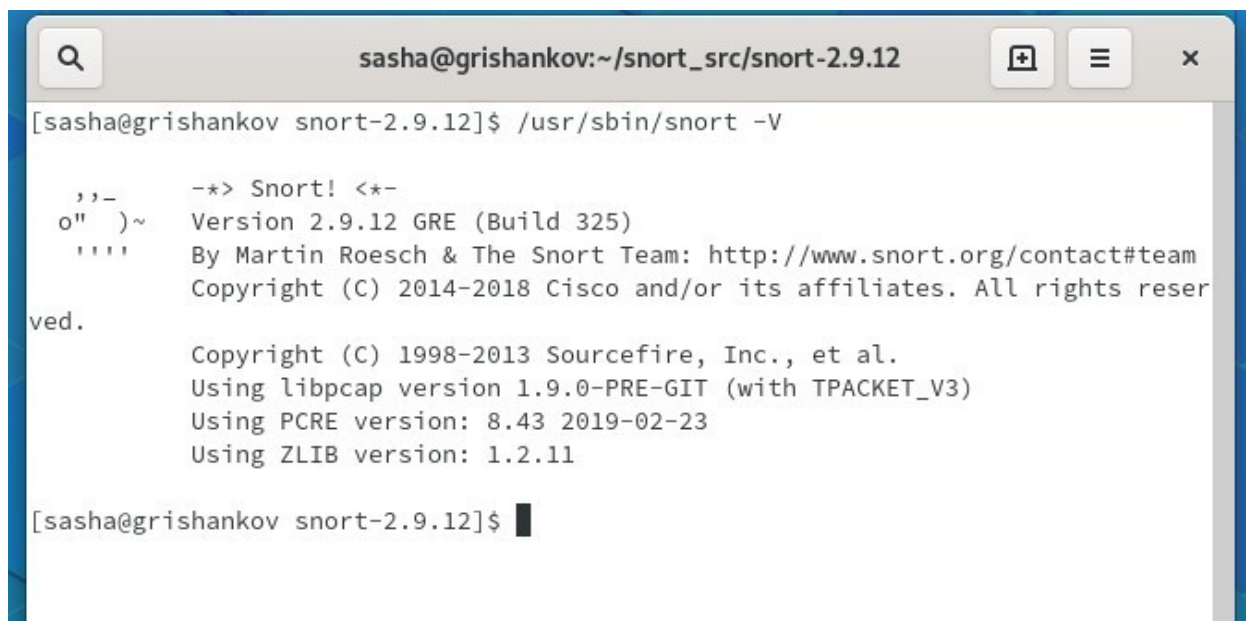
Поскольку при установке Snort двоичный файл Snort размещается в /usr/local/bin/ snort, рекомендуется создать символическую ссылку на /usr/sbin/snort:



```
[sasha@grishankov snort-2.9.12]$ sudo ldconfig
[sasha@grishankov snort-2.9.12]$ sudo ln -s /usr/local/bin/snort /usr/sbin/snort
[sasha@grishankov snort-2.9.12]$ ls -l /usr/sbin/snort
lrwxrwxrwx 1 root root 20 Mar 30 17:46 /usr/sbin/snort -> /usr/local/bin/snort
[sasha@grishankov snort-2.9.12]$
```

Рисунок 4 - Создание символической ссылки

Чтобы убедиться, что Snort Binary работает, был запущен Snort с флагом -V, в результате чего Snort отображает номер версии, а также сведения о версии libpcap, pcre и zlib.



```
sasha@grishankov:~/snort_src/snort-2.9.12
[sasha@grishankov snort-2.9.12]$ /usr/sbin/snort -V
,,-      -*> Snort! <*-
o" )~    Version 2.9.12 GRE (Build 325)
'''      By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
          Copyright (C) 2014-2018 Cisco and/or its affiliates. All rights reserved.

          Copyright (C) 1998-2013 Sourcefire, Inc., et al.
          Using libpcap version 1.9.0-PRE-GIT (with TPACKET_V3)
          Using PCRE version: 8.43 2019-02-23
          Using ZLIB version: 1.2.11

[sasha@grishankov snort-2.9.12]$
```

Рисунок 5 - Версия Snort

Необходимо убедиться, что сетевая карта не обрезает слишком большие пакеты. Некоторые сетевые карты имеют функции, называемые «Большая разгрузка приема» (lro) и «Общая разгрузка приема» (gro). Когда эти функции включены, сетевая карта выполняет сборку пакетов до того, как они будут обработаны ядром. По умолчанию Snort будет обрезать пакеты, размер которых превышает значение по умолчанию, составляющее 1518 байт. Кроме того, LRO и GRO могут вызвать проблемы с повторной сборкой на основе цели Stream5. Команда ниже отключает gro и lro.



```
sasha@grishankov:~
[sasha@grishankov ~]$ sudo pacman -S ethtool; ethtool -K ens33 gro off; \
> ethtool -K ens33 lro off
```

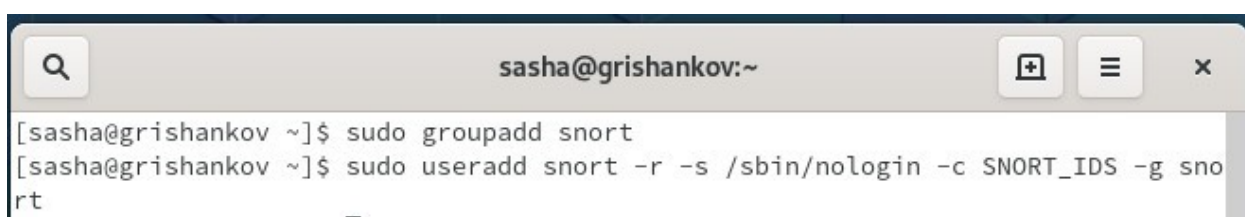
Рисунок 6 - Отключение gro и lro

НАСТРОЙКА СИСТЕМЫ IPS/IDS

3. Настройка параметров предотвращения вторжения

3.3 Базовая конфигурация

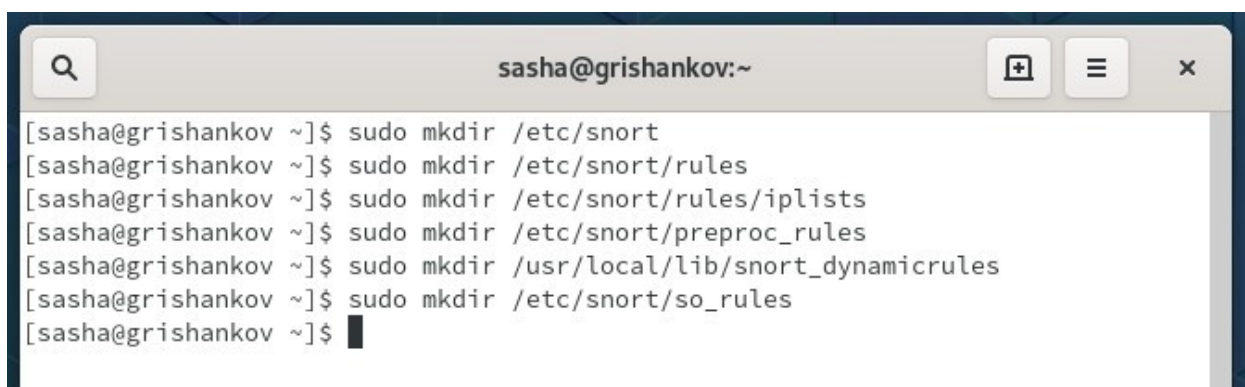
Во-первых, по соображениям безопасности Snort должен работать как непривилегированный пользователь. Для этого нужно создать пользователя и группу snort.



```
sasha@grishankov:~  
[sasha@grishankov ~]$ sudo groupadd snort  
[sasha@grishankov ~]$ sudo useradd snort -r -s /sbin/nologin -c SNORT_IDS -g snort
```

Рисунок 7 - Создание пользователя

Необходимо создать ряд файлов и папок, которые Snort ожидает при работе в режиме NIDS. затем изменить владельца этих файлов на пользователя snort.



```
sasha@grishankov:~  
[sasha@grishankov ~]$ sudo mkdir /etc/snort  
[sasha@grishankov ~]$ sudo mkdir /etc/snort/rules  
[sasha@grishankov ~]$ sudo mkdir /etc/snort/rules/iplists  
[sasha@grishankov ~]$ sudo mkdir /etc/snort/preproc_rules  
[sasha@grishankov ~]$ sudo mkdir /usr/local/lib/snort_dynamicrules  
[sasha@grishankov ~]$ sudo mkdir /etc/snort/so_rules  
[sasha@grishankov ~]$
```

Рисунок 8 - Создание файлов и директорий

Snort хранит файлы конфигурации в /etc/snort, правила в /etc/snort/rules, /usr/local/lib/snort_dynamicrules и сохраняет свои журналы в /var/log/snort:

- classification.config описывает типы классификаций атак;
- file_magic.conf описывает правила для определения типов файлов;
- reference.config содержит URL-адреса, которые предоставляют дополнительную информацию об оповещениях;
- snort.conf это файл конфигурации для Snort;
- threshold.conf позволяет контролировать количество событий, необходимых для создания оповещения;
- attribute table.dtd позволяет Snort использовать внешнюю информацию для определения протоколов и политик;
- gen-msg.map сообщает Snort, какой препроцессор используется по какому правилу;
- unicode.map обеспечивает отображение между языками Unicode и идентификатором..

```

sasha@grishankov:~$ sudo touch /etc/snort/rules/iplists/black_list.rules
sasha@grishankov:~$ sudo touch /etc/snort/rules/iplists/white_list.rules
sasha@grishankov:~$ sudo touch /etc/snort/rules/local.rules
sasha@grishankov:~$ sudo touch /etc/snort/sid-msg.map
sasha@grishankov:~$ sudo mkdir /var/log/snort
sasha@grishankov:~$ sudo mkdir /var/log/snort/archived_logs
sasha@grishankov:~$ sudo chmod -R 5775 /etc/snort
sasha@grishankov:~$ sudo chmod -R 5775 /var/log/snort
sasha@grishankov:~$ sudo chmod -R 5775 /var/log/snort/archived_logs
sasha@grishankov:~$ sudo chmod -R 5775 /etc/snort/so_rules
sasha@grishankov:~$ sudo chmod -R 5775 /usr/local/lib/snort_dynamicrules
sasha@grishankov:~$ sudo chown -R snort:snort /etc/snort
sasha@grishankov:~$ sudo chown -R snort:snort /var/log/snort
sasha@grishankov:~$ sudo chown -R snort:snort /usr/local/lib/snort_dynamicrules
sasha@grishankov:~$
  
```

Рисунок 9 - Изменение прав доступа

С помощью команд ниже необходимо переместить файлы которые идут с источниками snort в директорию /etc/snort.

```
sasha@grishankov:~/snort_src/snort-2.9.12/src/dynamic-prepro...
[sasha@grishankov ~]$ cd ~/snort_src/snort-2.9.12/etc/
[sasha@grishankov etc]$ sudo cp *.conf* /etc/snort
[sasha@grishankov etc]$ sudo cp *.map /etc/snort
[sasha@grishankov etc]$ sudo cp *.dtd /etc/snort
[sasha@grishankov etc]$ cd ~/snort_src/snort-2.9.12/src/dynamic-preprocessors/build/usr/local/lib/snort_dynamicpreprocessor/
[sasha@grishankov snort_dynamicpreprocessor]$ sudo cp * /usr/local/lib/snort_dynamicpreprocessor/
[sasha@grishankov snort_dynamicpreprocessor]$ tree /etc/snort
```

Рисунок 10 - Копирование необходимых файлов

Конфигурационная папка и структура файлов Snort теперь должны выглядеть следующим образом:

```
sasha@grishankov:~/snort_src/snort-2.9.12/src/dynamic-prepro...
[sasha@grishankov snort_dynamicpreprocessor]$ tree /etc/snort
/etc/snort
├── attribute_table.dtd
├── classification.config
├── file_magic.conf
├── gen-msg.map
├── preproc_rules
├── reference.config
├── rules
│   ├── iplist
│   │   ├── black_list.rules
│   │   └── white_list.rules
│   └── local.rules
├── sid-msg.map
├── snort.conf
├── so_rules
├── threshold.conf
└── unicode.map

4 directories, 12 files
[sasha@grishankov snort_dynamicpreprocessor]$
```

Рисунок 11 - Файловая структура каталога /etc/snort

Необходимо закомментировать строки, которые заставляют Snort импортировать набор файлов правил по умолчанию. Это делается так как PulledPork будет использоваться для управления наборами правил, который сохраняет все правила в одном файле. Самый простой способ закомментировать все эти строки - использовать sed для добавления символа «#» (хеш) к этим строкам. Это достигается с помощью следующей команды:



```
sasha@grishankov:~/snort_src/snort-2.9.12/src/dynamic-prepro...
[sasha@grishankov snort_dynamicpreprocessor]$ sudo sed -i 's/include \$RULE\_PATH/#include \$RULE\_PATH/' /etc/snort/snort.conf
[sasha@grishankov snort_dynamicpreprocessor]$
```

Рисунок 12 - Комментирование ненужных линий

Во-первых, нам нужно сообщить Snort о диапазоне вашей домашней сети и всех других внешних сетей. Мы делаем это, редактируя строки 45 и 48 в snort.conf, чтобы сообщить диапазон IP-адресов этих двух сетей.

```
# Setup the network addresses you are protecting
ipvar HOME_NET 192.168.0.0/16
```

Рисунок 13 - Настройка переменной HOME_NET

Далее нам нужно сообщить Snort о расположении всех папок, которые мы создали ранее. Эти настройки также являются частью файла snort.conf. Я включил номера строк после хеша.

```
var RULE_PATH /etc/snort/rules # line 104
var SO_RULE_PATH /etc/snort/so_rules # line 105
var PREPROC_RULE_PATH /etc/snort/preproc_rules # line 106

var WHITE_LIST_PATH /etc/snort/rules/iplists # line 113
var BLACK_LIST_PATH /etc/snort/rules/iplists # line 114
```

Рисунок 14 - Редактирование конфигурационного файла

Проверка файла конфигурации на ошибки (Флаг -T) реализуется командой ниже. Флаг -i – имя интерфейса.



```
sasha@grishankov:~/snort_src/snort-2.9.12/src/dynamic-prepro...  
[sasha@grishankov snort_dynamicpreprocessor]$ sudo snort -T -c /etc/snort/\> snort.conf -i ens33
```

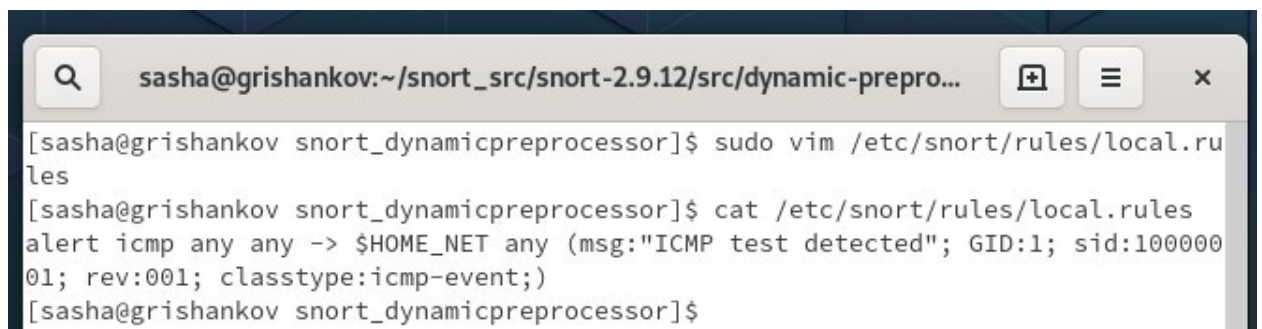
Рисунок 15 - Проверка на ошибки

Snort успешно проверил конфигурацию на наличие ошибок.

```
Snort successfully validated the configuration!  
Snort exiting  
[sasha@grishankov snort_dynamicpreprocessor]$
```

Рисунок 16 - Конфигурация прошла успешно

Было создано локальное правило. Это правило говорит о том, что для любых пакетов ICMP, которые snort видит из любой сети в нашу HOME_NET, генерируется предупреждение с текстом «ICMP test detected». Другая информация здесь (GID, REV, classtype) используется для группирования правил.



```
sasha@grishankov:~/snort_src/snort-2.9.12/src/dynamic-prepro...  
[sasha@grishankov snort_dynamicpreprocessor]$ sudo vim /etc/snort/rules/local.rules  
[sasha@grishankov snort_dynamicpreprocessor]$ cat /etc/snort/rules/local.rules  
alert icmp any any -> $HOME_NET any (msg:"ICMP test detected"; GID:1; sid:10000001; rev:001; classtype:icmp-event;)  
[sasha@grishankov snort_dynamicpreprocessor]$
```

Рисунок 17 - Создание правила

3.2 Конфигурация barnyard2

Barnyard2 - это интерпретатор с открытым исходным кодом для двоичных выходных файлов Snort unified2. Его основное использование позволяет Snort эффективно записывать на диск и оставляет задачу анализа двоичных данных в различных форматах отдельным процессом, который не заставит Snort пропускать сетевой трафик.

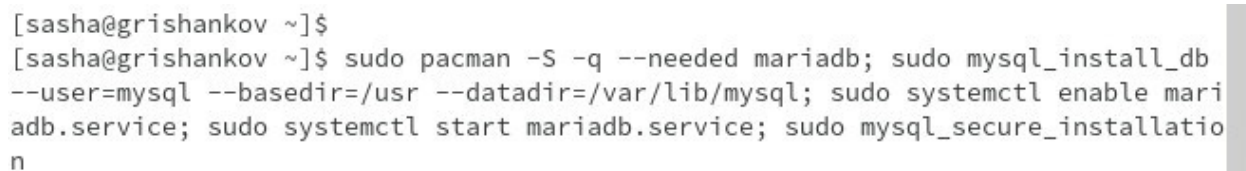
Чтобы barnyard2 знал о существовании локального правила, которое было создано ранее для обнаружения ICMP, следует заполнить файл sid-msg.map следующим образом:



```
sasha@grishankov:~/snort_src/snort-2.9.12/src/dynamic-prepro...
[sasha@grishankov snort_dynamicpreprocessor]$ sudo vim /etc/snort/sid-msg.map
[sasha@grishankov snort_dynamicpreprocessor]$ cat /etc/snort/sid-msg.map
#v2
1 || 10000001 || 001 || icmp-event || 0 || ICMP Test detected || url,tools.ietf.
org/html/rfc792
[sasha@grishankov snort_dynamicpreprocessor]$
```

Рисунок 18 - Редактирование sid-msg.map

Для того, чтобы barnyard2 мог полноценно работать, необходимо установить пакет mariadb. С помощью команд ниже был скачан пакет, произведена первичная конфигурация БД, указан директория бинарников и данных, включен сервис, запущен интерактивный режим безопасной конфигурации.



```
[sasha@grishankov ~]$
[sasha@grishankov ~]$ sudo pacman -S -q --needed mariadb; sudo mysql_install_db
--user=mysql --basedir=/usr --datadir=/var/lib/mysql; sudo systemctl enable mari
adb.service; sudo systemctl start mariadb.service; sudo mysql_secure_installatio
n
```

Рисунок 19 - Установка БД

В конфигурационном файле `snort.conf` следует добавить строку для вывода в формат `unified2`. Файлы вывода будут иметь формат `.u2`.

```
# unified2
# Recommended for most installs
output unified2: filename snort.u2, limit 128
# output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_types
```

Рисунок 20 - Изменение параметров конфигурации

Для скачивания источников `barnyard2` необходимо выполнить следующие команды:

```
cd ~/snort_src
wget https://github.com/firnsy/barnyard2/archive/master.tar.gz -O barnyard2-Master.tar.gz
tar zxvf barnyard2-Master.tar.gz
cd barnyard2-master
autoreconf -fvi -I ./m4
```

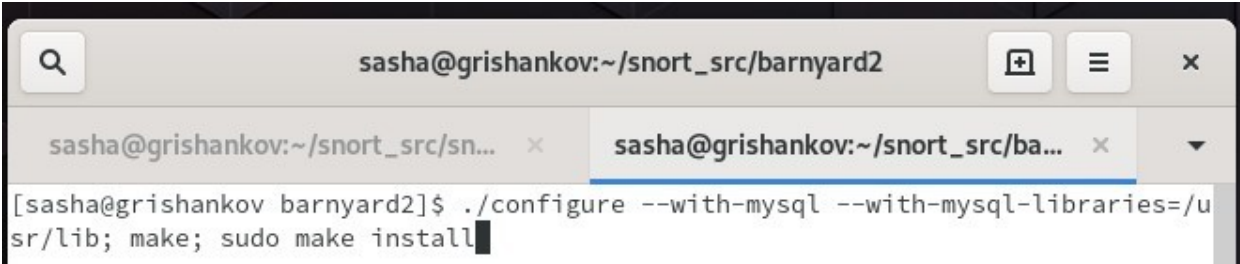
Рисунок 21 - Загрузка barnyard2

`Barnyard2` потребует наличие файла `dumbnet.h` который был создан как символическая ссылка на `dnet.h`. `ldconfig` — настроить динамические привязки компоновщика во время выполнения.

```
[sasha@grishankov barnyard2]$ sudo ln -s /usr/include/dnet.h /usr/include/dumbnet.h;
[sasha@grishankov barnyard2]$ sudo ldconfig
```

Рисунок 22 - Создание символической ссылки

Источники `barnyard2` необходимо сконфигурировать с указанием пути до библиотек `mysql`. Затем идут команды установки.



```
sasha@grishankov:~/snort_src/barnyard2
[sasha@grishankov barnyard2]$ ./configure --with-mysql --with-mysql-libraries=/usr/lib; make; sudo make install
```

Рисунок 23 - Компиляция barnyard2

Необходимо скопировать файл конфигурации barnyard2.conf в папку snort, а также создать директорию barnyard2 для хранения логов.

```
sudo cp ~/snort_src/barnyard2-master/etc/barnyard2.conf /etc/snort/

# the /var/log/barnyard2 folder is never used or referenced
# but barnyard2 will error without it existing
sudo mkdir /var/log/barnyard2
sudo chown snort.snort /var/log/barnyard2

sudo touch /var/log/snort/barnyard2.waldo
sudo chown snort.snort /var/log/snort/barnyard2.waldo
```

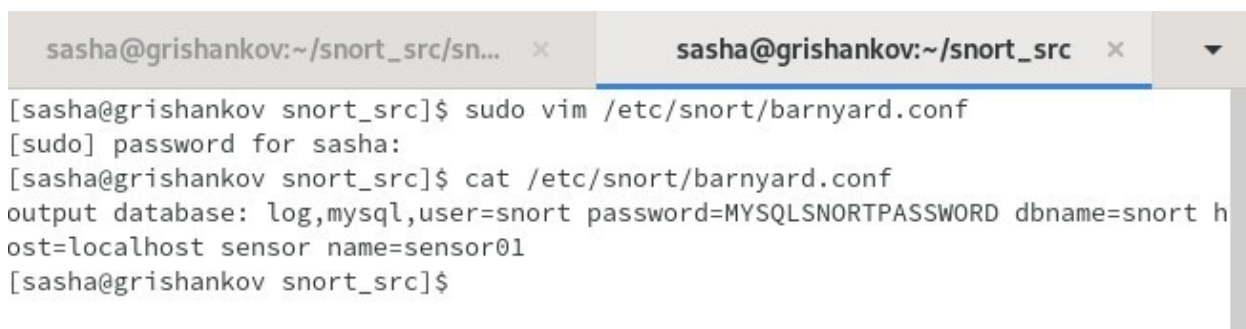
Рисунок 24 - Создание лог-файлов

С помощью указанных ниже команд была создана база данных snort с пользователем snort для хранения записей barnyard2.

```
$ mysql -u root -p
mysql> create database snort;
mysql> use snort;
mysql> source ~/snort_src/barnyard2-master/schemas/create_mysql
mysql> CREATE USER 'snort'@'localhost' IDENTIFIED BY 'MYSQLSNORTPASSWORD';
mysql> grant create, insert, select, delete, update on snort.* to 'snort'@'localhost';
mysql> exit
```

Рисунок 25 - Создание БД snort

В конфигурационном файле barnyard.conf были указаны параметры для вывода в созданную базу данных.



```
sasha@grishankov:~/snort_src/sn... x sasha@grishankov:~/snort_src x
[sasha@grishankov snort_src]$ sudo vim /etc/snort/barnyard.conf
[sudo] password for sasha:
[sasha@grishankov snort_src]$ cat /etc/snort/barnyard.conf
output database: log,mysql,user=snort password=MYSQLSNORTPASSWORD dbname=snort h
ost=localhost sensor name=sensor01
[sasha@grishankov snort_src]$
```

Рисунок 26 - Конфигурация barnyard2

3.3 Конфигурация pulledpork

Следующий шаг — установка pulledpork, чтобы правила обновлялись регулярно. Для начала, надо зарегистрировать на сайте snort и получить oinkcode (мой личный id для скачивания правил).

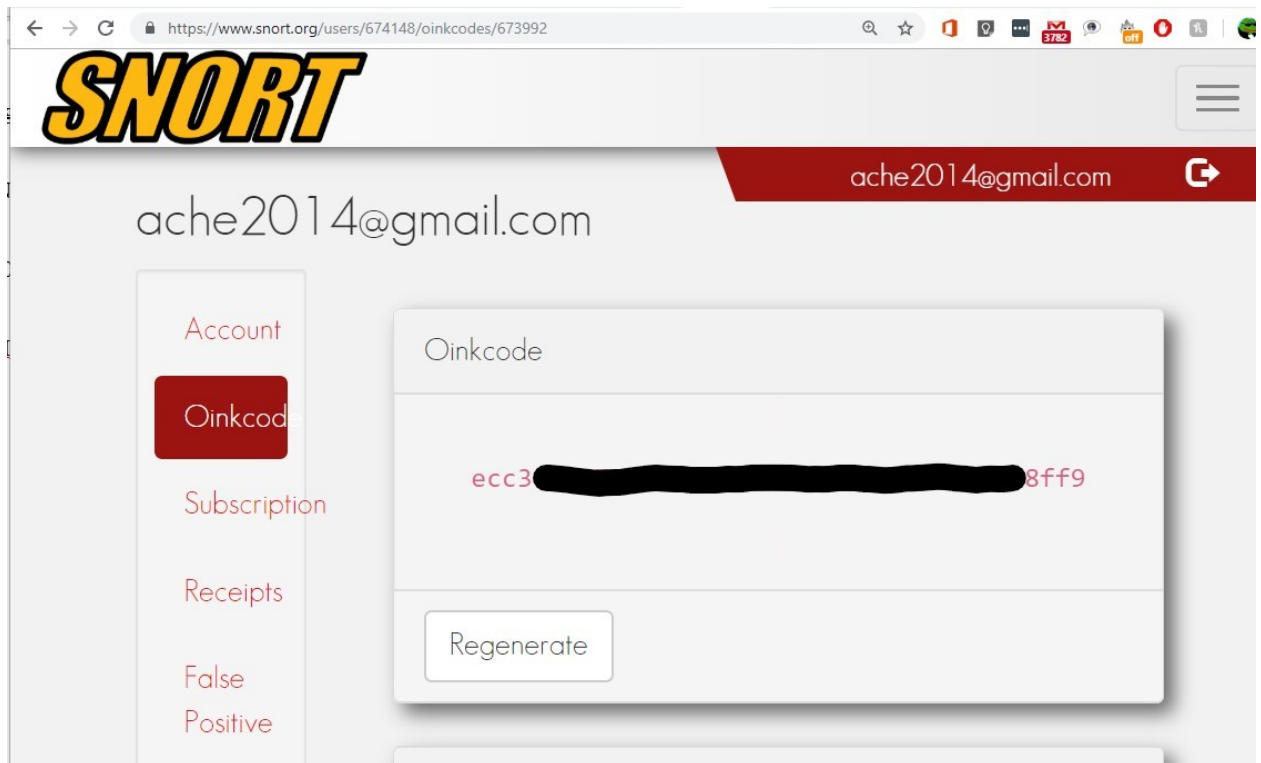


Рисунок 27 - Oinkcode

Для полноценной конфигурации pulledpork необходимо также установить пакеты cronie, perl-crypt-ssleay, perl-lwp-protocol-https, а также активировать планировщик cronie.

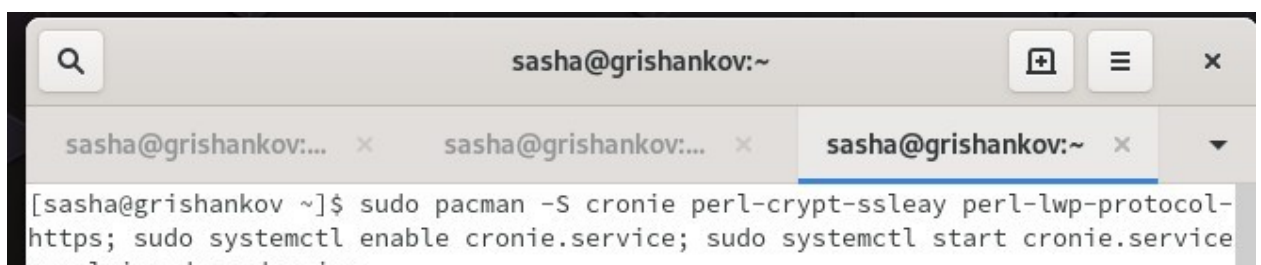



Рисунок 28 - Установка зависимостей

Был введен набор команд который означает: перейти в `snort_src`, скопировать ветвь `pulledpork` из `git`, перейти в `pulledpork`, скопировать скрипт `pulledpork.pl` в `/usr/local/bin`, присвоить права на исполнение скрипта, скопировать файлы конфигурации в директорию `snort`.



The image shows a terminal window with the title bar "sasha@grishankov:~/snort_src/pulledpork". The terminal content displays the following commands and their output:

```
[sasha@grishankov ~]$ cd snort_src/; git clone https://github.com/shirkdog/pulledpork.git; cd pulledpork; sudo cp pulledpork.pl /usr/local/bin; sudo chmod +x /usr/local/bin/pulledpork.pl; sudo cp etc/*.conf /etc/snort
```

The terminal output shows the successful execution of these commands, including the cloning of the repository, the copying of the pulledpork.pl script to /usr/local/bin, and the copying of configuration files to /etc/snort.

Рисунок 29 - Загрузка pulledpork

В конфигурационном файле `pulledpork.conf` был указан мой `oinkcode` (замазан в целях безопасности), пути для хранения правил и остальные атрибуты отображенные ниже начиная со строки `rule url=`.

```
sasha@grishankov:... x sasha@grishankov:... x sasha@grishankov:... x
[sasha@grishankov pulledpork]$ cat /etc/snort/pulledpork.conf |grep -v '^$\|^s\|^#'
rule_url=https://www.snort.org/reg-rules/|snortrules-snapshot.tar.gz|ecc3bcb
24bfff9
rule_url=https://snort.org/downloads/community/|community-rules.tar.gz|Community
rule_url=https://talosintelligence.com/documents/ip-blacklist|IPBLACKLIST|open
rule_url=https://snort.org/downloads/community/|opensource.gz|Opensource
ignore=deleted.rules,experimental.rules,local.rules
temp_path=/tmp
rule_path=/etc/snort/rules/snort.rules
local_rules=/etc/snort/rules/local.rules
sid_msg=/etc/snort/sid-msg.map
sid_msg_version=2
sid_changelog=/var/log/sid_changes.log
sorule_path=/usr/local/lib/snort_dynamicrules/
snort_path=/usr/local/bin/snort
config_path=/etc/snort/snort.conf
distro=FreeBSD-8-1
black_list=/etc/snort/rules/iplists/black_list.rules
IPRVersion=/etc/snort/rules/iplists
snort_control=/usr/local/bin/snort_control
version=0.7.4
[sasha@grishankov pulledpork]$
```

Рисунок 30 - Конфигурация pulledpork

Процесс загрузки правил был запущен. Скрипт проводит проверку сигнатур, затем загружает архив.

```
sasha@grishankov:~$ sudo /usr/local/bin/pulledpork.pl -c /etc/snort/pulledpork.conf -l
[sudo] password for sasha:

https://github.com/shirkdog/pulledpork

  -----
  \-----,\      )
  \---==\\ /      PulledPork v0.7.4 - Helping you protect your bitcoin wallet!
  \---==\\ /
  .-~~~~-.Y|\\_   Copyright (C) 2009-2017 JJ Cummings, Michael Shirk
@_/_      / 66\\_ and the PulledPork Team!
|      \   \   _(")
 \     /-| ||'---' Rules give me wings!
  \_   \_\\
~~~~~

Checking latest MD5 for snortrules-snapshot-29120.tar.gz....
Rules tarball download of snortrules-snapshot-29120.tar.gz....
```

Рисунок 31 - Запуск скрипта pulledpork

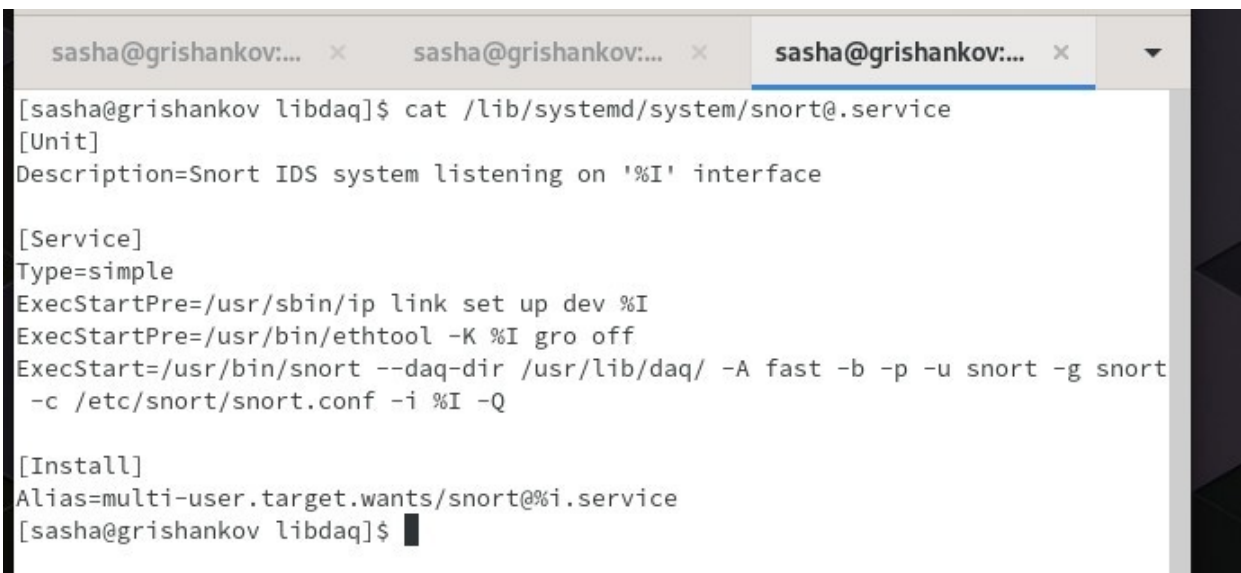
С помощью планировщика задач cronie, была создана задача на обновление правил snort каждый день в 3:23.

```
sasha@grishankov:~$ sudo crontab -l
# PulledPork checking rules existence every day at 3am
23 03 * * * /usr/local/bin/pulledpork.pl -c /etc/snort/pulledpork.conf -l
[sasha@grishankov ~]$
```

Рисунок 32 - Создание задачи обновления правил

3.4 Создание службы Snort@.service

Чтобы быстро запускать snort, а также легко контролировать его работу был создан сервис snort@. Ниже показано содержимое сервиса. Вкратце, при запуске сервера, отключается gro у указанного интерфейса (%I), а также запускается snort с некоторым необходимым набором атрибутов.



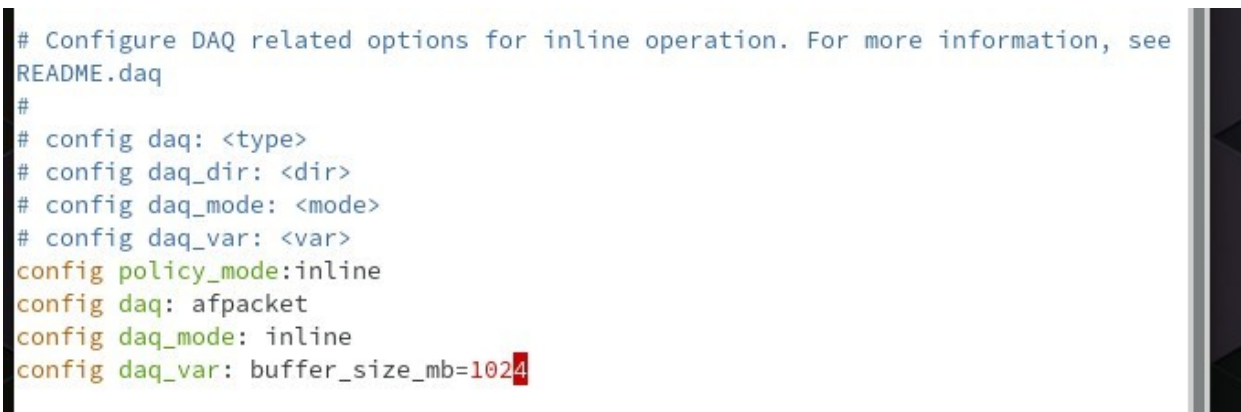
```
[sasha@grishankov libdaq]$ cat /lib/systemd/system/snort@.service
[Unit]
Description=Snort IDS system listening on '%I' interface

[Service]
Type=simple
ExecStartPre=/usr/sbin/ip link set up dev %I
ExecStartPre=/usr/bin/ethtool -K %I gro off
ExecStart=/usr/bin/snort --daq-dir /usr/lib/daq/ -A fast -b -p -u snort -g snort
-c /etc/snort/snort.conf -i %I -Q

[Install]
Alias=multi-user.target.wants/snort@%i.service
[sasha@grishankov libdaq]$
```

Рисунок 33 - Конфигурация сервиса

В файле snort.conf необходимо заполнить параметры показанные на скриншоте ниже. Это позволит работать снорту в режиме IPS, или по-другому inline режим.



```
# Configure DAQ related options for inline operation. For more information, see
README.daq
#
# config daq: <type>
# config daq_dir: <dir>
# config daq_mode: <mode>
# config daq_var: <var>
config policy_mode:inline
config daq: afpacket
config daq_mode: inline
config daq_var: buffer_size_mb=1024
```

Рисунок 34 - Конфигурация для работы в режиме IPS

Далее была установлена библиотека DAQ, которая позволяет работать в режиме IPS. DAQ или библиотека сбора данных, для ввода / вывода пакетов. DAQ заменяет прямые вызовы функций libpcap на уровень абстракции, который облегчает работу с различными аппаратными и программными интерфейсами, не требуя изменений в Snort. Можно выбрать тип и режим DAQ при вызове Snort для выполнения считывания с pcap или встроенной операции и т. Д.

```
[sasha@grishankov libdaq]$ git clone https://aur.archlinux.org/libdaq.git; cd libdaq; makepkg -sic
```

Рисунок 35 - Установка DAQ

Теперь можно проверить работоспособность сконфигурированного демона snort@ с помощью средств systemctl. Как видно ниже, snort был успешно запущен в режиме IPS – мост между интерфейсами ens33 и ens34.

```
[sasha@grishankov ~]$ sudo systemctl start snort@ens33:ens34
[sasha@grishankov ~]$ sudo systemctl status snort@ens33:ens34
● snort@ens33:ens34.service - Snort IDS system listening on 'ens33:ens34' interface
   Loaded: loaded (/usr/lib/systemd/system/snort@.service; disabled; vendor preset: enabled)
   Active: active (running) since Mon 2019-04-01 03:08:15 MSK; 8s ago
     Process: 1438 ExecStartPre=/usr/sbin/ip link set up dev ens33:ens34 (code=exited, status=0/SUCCESS)
     Process: 1439 ExecStartPre=/usr/bin/ethtool -K ens33:ens34 gro off (code=exited, status=0/SUCCESS)
    Main PID: 1440 (snort)
       Tasks: 2 (limit: 2646)
      Memory: 55.2M
    CGroup: /system.slice/system-snort.slice/snort@ens33:ens34.service
            └─1440 /usr/bin/snort --daq-dir /usr/lib/daq/ -A fast -b -p -u snort
```

Рисунок 36 - Запуск сервиса

4. Тестирования системы IPS/IDS

С помощью команды `ip addr` необходимо узнать адрес интерфейса `ens33`, на котором `snort` будет слушать трафик. Адрес видно на скриншоте ниже — `192.168.74.139`.

```
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:51:7b:28 brd ff:ff:ff:ff:ff:ff
    inet 192.168.74.139/24 brd 192.168.74.255 scope global dynamic noprefixroute ens33
        valid_lft 950sec preferred_lft 950sec
    inet6 fe80::910:d2d5:89fe:5835/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[sasha@grishankov ~]$
```

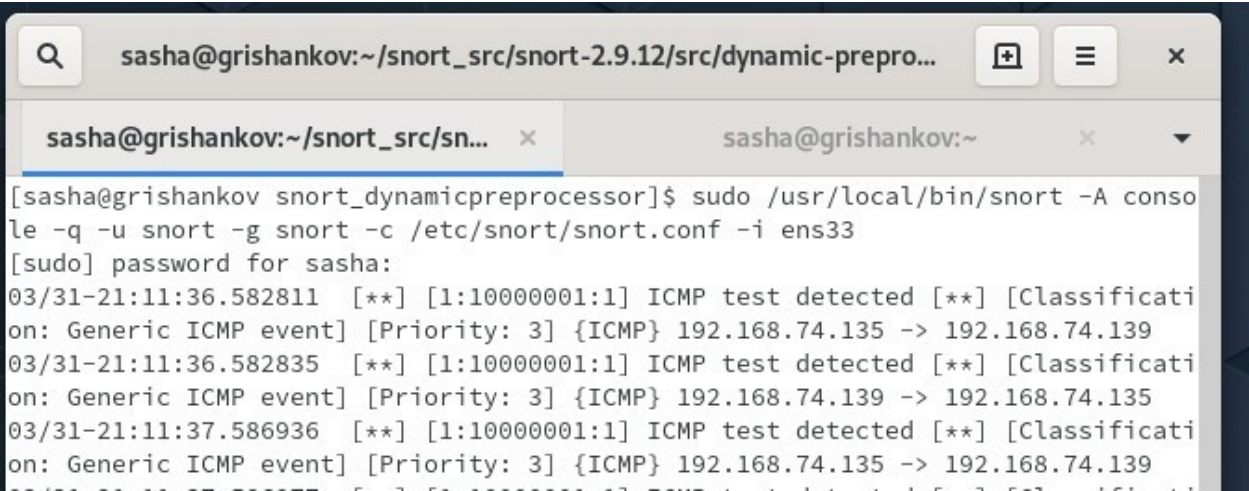
Рисунок 37 - IP адрес snort

Для того, чтобы проверить работоспособность тестового правила обнаружения ICMP трафика необходима вторая виртуальная машина. В данном примере был выбран дистрибутив Ubuntu 18.04. Следующие команды показывают адрес интерфейса `ens34`, а также запускают `ping` на машину `snort`.

```
sasha@sasha-virtual-machine: ~
File Edit View Search Terminal Help
sasha@sasha-virtual-machine:~$ ip addr show dev ens34; ping 192.168.74.139
3: ens34: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 1000
    link/ether 00:0c:29:b2:3b:68 brd ff:ff:ff:ff:ff:ff
    inet 192.168.74.135/24 brd 192.168.74.255 scope global ens34
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:feb2:3b68/64 scope link
        valid_lft forever preferred_lft forever
PING 192.168.74.139 (192.168.74.139) 56(84) bytes of data.
64 bytes from 192.168.74.139: icmp_seq=1 ttl=64 time=0.458 ms
64 bytes from 192.168.74.139: icmp_seq=2 ttl=64 time=0.977 ms
64 bytes from 192.168.74.139: icmp_seq=3 ttl=64 time=0.303 ms
64 bytes from 192.168.74.139: icmp_seq=4 ttl=64 time=0.405 ms
```

Рисунок 38 - ICMP ping

Был произведен запуск процесса snort. Вывод — на экран консоли, конфигурационный файл — snort.conf, интерфейс прослушивания — ens 33. После введение пароля уведомления о срабатывании тестовых правил ICMP начали появляться на экране консоли. Как видно, ICMP трафик приходит с адреса 192.168.74. 135 (виртуальная машина Ubuntu) на адрес машины snort.



```
sasha@grishankov:~/snort_src/snort-2.9.12/src/dynamic-preprocessor...
sasha@grishankov:~/snort_src/sn... x sasha@grishankov:~ x
[sasha@grishankov snort_dynamicpreprocessor]$ sudo /usr/local/bin/snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i ens33
[sudo] password for sasha:
03/31-21:11:36.582811  ** [1:10000001:1] ICMP test detected ** [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.74.135 -> 192.168.74.139
03/31-21:11:36.582835  ** [1:10000001:1] ICMP test detected ** [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.74.139 -> 192.168.74.135
03/31-21:11:37.586936  ** [1:10000001:1] ICMP test detected ** [Classification: Generic ICMP event] [Priority: 3] {ICMP} 192.168.74.135 -> 192.168.74.139
```

Рисунок 39 - Уведомления snort