Обработка статистики NetFlow

Во второй лабораторной работе был рассмотрен процесс настройки мониторинга сетевого трафика с использованием протокола NetFlow на базе операционной системы Debian GNU/Linux версии 7.6.0. Теперь каждую минуту в директории /var/cache/nfdump на вашем лабораторном компьютере создаётся новый бинарный файл NetFlow с записями о сетевых потоках за прошлую минуту. При ротации файла запускается скрипт /usr/local/sbin/nfdump.pl, который должен производить следующие операции:

1. Преобразовывать бинарные файлы в текстовые таблицы, пригодные для статистической обработки.
2. Выводить статистику по количеству завершившихся потоков.
3. Обнаруживать атаку на веб-сервер по количеству запросов с одного IP-адреса.
4. Заносить IP-адрес атакующего в файл журнала.
5. Блокировать IP-адрес атакующего компьютера.
6. Разблокировать IP-адрес через определённое время.

Так как текстовые файлы nfdump представляют собой таблицы большого объёма, то для их разбора лучше всего подойдёт специально для этих целей созданный скриптовой язык Perl. Именно на нём будет выполнен обрабатывающий скрипт. Интерпретатор этого языка уже предустановлен в базовой версии Debian сервера.

Для разблокировки IP-адреса при блокировании он должен быть куда-то записан, а так же сохранено время блокировки. Удобней всего проводить запись в базу данных. Кроме того в БД можно сохранять ежеминутную статистику по количеству потоков. Для этого установим, если этого не было сделано раньше, базу данных MySQL:

```
apt-get install mysql-server
```

Помните, что большинство команд администрирования доступны только привилегированным пользователям! Для получения привилегий воспользуйтесь командами su или sudo.

В процессе установки будет запрошен пароль для учётной записи root базы данных, как это показано на рисунке 1. Затем на следующем экране нужно повторить этот пароль, чтобы не было ошибки. Не путайте пароль администратора MySQL с паролем операционной системы!

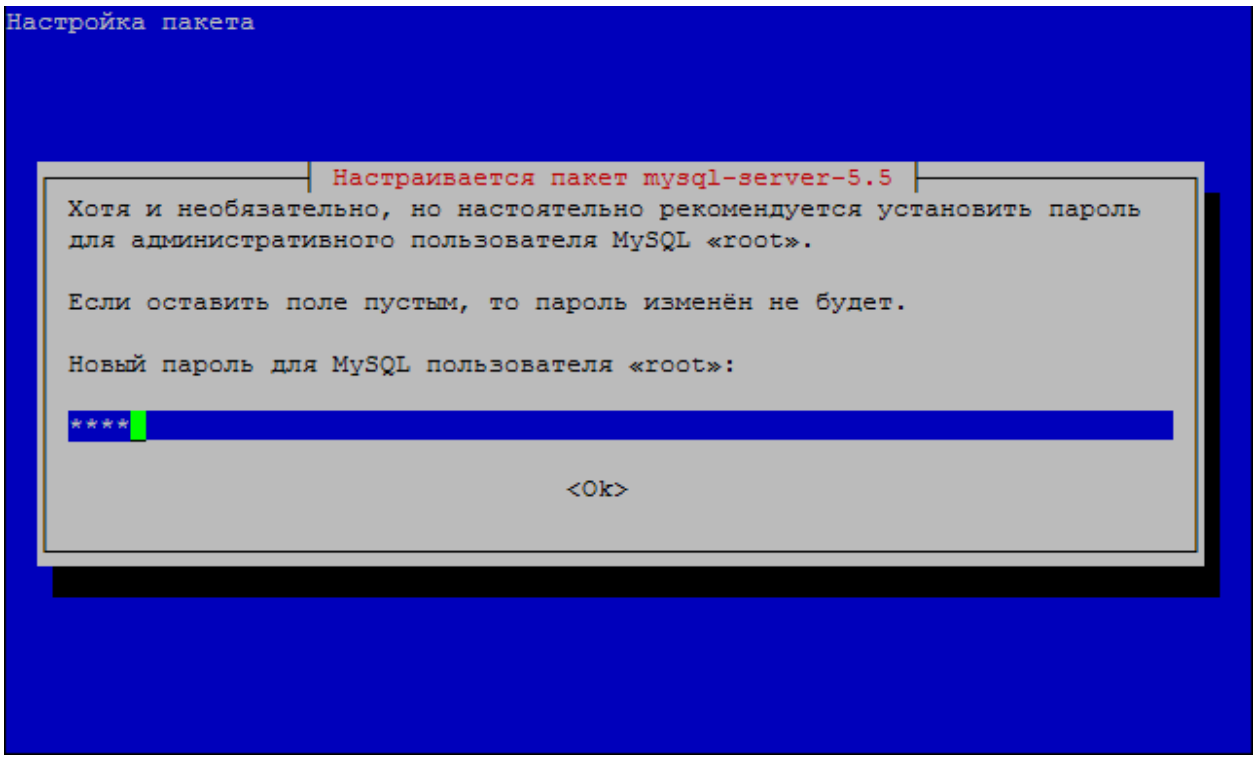

Рисунок 1 – Установка MySQL: запрос root пароля

Обратите внимание, что если MySQL не работает должным образом её можно полностью переустановить, выполнив следующие команды:

```
apt-get remove --purge ^mysql-server-* mysql-common
apt-get install mysql-server
```

Теперь пришло время создать базу данных, таблицы и пользователя для скрипта. Это можно сделать, подключившись к БД следующей командой:

```
mysql –user=root -p
```

Либо установить на веб-сервер phpMyAdmin [pma]. Затем выполните следующие SQL команды:

```
# Создание БД с именем nfdump_pl
CREATE DATABASE `nfdump_pl` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;

# Создание пользователя nfdump_pl, который имеет только локальный доступ на выборку, вставку, обновление и удаление записей в базе данных nfdump_pl с паролем password
GRANT SELECT, INSERT, UPDATE, DELETE PRIVILEGES ON `nfdump_pl` TO nfdump_pl@localhost IDENTIFIED BY 'password';

# Переход в контекст БД nfdump_pl
USE `nfdump_pl`;

# Таблица для хранения забаненных IP-адресов
# Создание таблицы banlist с полями id (автоматический идентификатор записи), ip (забаненный IP-адрес), time (автоматически проставляемое время создания записи)
CREATE TABLE IF NOT EXISTS `banlist` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `ip` int(10) unsigned NOT NULL,
```

```
  `time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`),
  UNIQUE KEY `ip` (`ip`),
  KEY `time` (`time`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;

# Таблица для сохранения количества потоков в определённое время
# Создание таблицы flow_cnt с полями ts (автоматически проставляемое время создания записи), flows (количество потоков)
CREATE TABLE IF NOT EXISTS `flow_cnt` (
  `ts` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `flows` int(10) unsigned NOT NULL,
  PRIMARY KEY (`ts`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

Для занесения IP-адреса в таблицу необходимо выполнить по крайней мере два запроса к БД:

1. Проверить, не заблокирован ли IP-адрес уже.

2. Если заблокирован, то обновить время блокировки, в противном случае добавить IP-адрес в таблицу.

Для уменьшения количества запросов к БД ниже приведено создание специальной функция ban. Она будет использоваться в скрипте. Код необходимо выполнить в SQL окне:

```
# Создание функции ban, принимающей в качестве аргумента IP адрес, а возвращающей состояние: 0 – IP-адрес уже существует в таблице, 1 – IP-адрес внесён в таблицу.
CREATE FUNCTION `ban`(parip VARCHAR(15)) RETURNS tinyint(1)
BEGIN
  # Переменная для идентификатора записи
  DECLARE vid INT(10) UNSIGNED DEFAULT 0;
  # Переменная для преобразованного в число IP адреса
  DECLARE iip INT(10) UNSIGNED DEFAULT INET_ATON(parip);
  # Поиск IP-адреса в таблице banlist
  SELECT id INTO vid FROM banlist WHERE ip=iip LIMIT 1;
  # Если IP-адрес не найден
  IF vid = 0 THEN
    # Вставляется новая запись
    INSERT INTO banlist (ip) VALUES (iip);
  ELSE # Если адрес найден
    # Обновляется время в существующей записи
    UPDATE banlist SET time = CURRENT_TIMESTAMP WHERE id = vid;
  END IF;
  # Возвращается 0 – IP-адрес уже существует в таблице, 1 – IP-адрес внесён в таблицу
  RETURN vid=0;
END
```

Далее приведён код скрипта на языке Perl с комментариями:

```
#!/usr/bin/perl
use 5.8.8; use strict; use warnings; use DBI; # подключение Perl библиотек

# данные для подключения к БД
my $hostname = "localhost";
my $database = "nfdump_pl";
my $user = "nfdump_pl";
my $password = "password";
```

```
my $dbh = 0; # указатель на подключение к БД
my $sth = 0; # запрос к БД

my $path_nfcapd = "/var/cache/nfdump/";      # путь к бинарным файлам nfcapd
my $path_nfdump = "/var/nfdump/";      # путь к текстовым файлам nfdump

# поиск самого нового файла файла в архиве
chdir $path;                           # задаёт текущей директорию $path
my (@file_list) = glob "nfcapd.20*";   # сохраняем список файлов по маске в массив
@file_list
my $flowfile = $file_list[$#file_list]; # в переменную $flowfile сохраняем значение
последнего элемента массива @file_list
undef @file_list;                      # удаляем из памяти массив @file_list
##############################

# преобразование бинарного формата nfdump в текстовый в той же директории
system "nfdump -r ".$path_nfcapd.$flowfile." > ".$path_nfcapd.$flowfile.".txt";
# удаление бинарного файла (если конечно мы не хотим сохранить его в архиве)
system "rm -f ".$path_nfcapd.$flowfile;

open(FileData,$path_nfdump.$flowfile.".txt"); # открытие текстового файла nfdump
my @lines = <FileData>; # считываем содержимое файла в массив строк
close(FileData); # закрываем файл

# копирование текстового файла nfdump в архивную директорию (если нам нужен архив)
system "cp ".$path_nfcapd.$flowfile.".txt ".$path_nfdump.$flowfile.".txt";
# удаление текстового файла nfdump из директории с бинарными файлами
system "rm -f ".$path_capd.$flowfile.".txt";

my $fcnt = 0; # общее количество потоков на весь файл (за минуту)

my @IPs = ();             # массив [IP-адрес, Дата] для потоков

for ($i = 1; $i < @lines-4; $i++) { # заполнение массивов исходных данных
  if    (    $lines[$i]    =~    m/^(\d{4}-\d{2}-\d{2}    \d{2}:\d{2}:\d{2}\.\d{3})
{1,5}\d{1,5}\.\d{3}    \w{1,6}    +(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}):[\d    ]{6}->
+(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}):\d{1,5} +\d+ +\d+ +(\d+)$/ ) {
# каждая строка файла nfdump обрабатывается регулярным выражением и на каждой
итерации переменным присваиваются следующие значения:
# $1 – Дата и время фиксации потока в файле
# $2 – IP-адрес источника
# $3 – IP-адрес назначения
# $4 – зафиксированное количество потоков
    if ($3 eq "IP нашего сервера") { # для всех потоков идущих на наш сервер
      if (($2 ne "IP DNS 1") && ($2 ne "IP DNS 2")) { # исключаем ДНС сервера
        $fcnt  +=  $4;  # прибавляем к переменной $fcnt количество зафиксированных
потоков
        push(@IPs , [$2,$1]); # добавляем в массив @IPs запись о потоке
      }
    }
  }
}

# сортировка массива @IPs по IP-адресу источника
@IPs = sort { ( $a->[0] cmp $b->[0] ) } @IPs;

open ( LogFile, ">> /var/log/nfdump.pl.log" ); # открываем файл журнала для записи
$dbh  =  DBI->connect("DBI:mysql:$database:$hostname",  $user,  $password)  ||  print
LogFile "Got error". $dbh->errstr ."\n"; # получение доступа к БД, в случае неудачи
ошибка сохраняется в журнал
```

```
# заносим общее число потоков в таблицу flow_cnt БД
$sth = $dbh->prepare("INSERT INTO flow_cnt (flows) VALUES (".$fcnt.")"); # подготовка
запроса
$sth->execute; # выполнение запроса

my $ref = 0; # служебная переменная для результатов запросов к БД

my $count = 1; # счетчик числа потоков
my $i = 0; # служебная переменная для цикла
for ($i = 1; $i < $#IPs; $i++) { # проход по всем элементам массива IPs
  if ($IPs[$i-1][0] eq $IPs[$i][0]) { # если текущий IP адрес в элементе массива IPs
равен предыдущему, то число потоков на этот IP увеличивается на единицу
    ++($count);
  } else { # в противном случае
    if ($count > 300) { # проверяется не превысило ли количество потоков 300 и если
превысило, то
      my $ipT = $IPs[$i-1][0]; # в переменную $IPt сохраняется IP-адрес атакующего
      print LogFile $IPs[$i-1][1]," ",$ipT," ",$count,"\n"; # IP-адрес атакующего
заносится в журнал

#>>>>> Следующие строки используются только в данной лабораторной работе для
непосредственной блокировки IP-адреса атакующего и в дальнейшем в шестой лабораторной
работе будет предложен универсальный способ блокировки при помощи fail2ban, который
работает с файлами журналов.

      # IP-адрес атакующего заносится в таблицу забаненных в БД
      $sth = $dbh->prepare("select ban('".$ipT."');"); # подготовка запроса к БД
      $sth->execute; # выполнение запроса к БД
      if (($ref = $sth->fetchrow_arrayref) && ($$ref[0]==1)) { # проверяется
выполнился ли запрос к БД и какой статус вернула функция ban (0 уже в таблице
забаненных, 1 успешно помещён в список)
        system "iptables -I fail2ban-galcev -s ".$ipT." -j DROP"; # блокировка IP-
адреса на сервере с помощью iptables
      }
#<<<<<
    }
    $count = 1; # сброс счётчика количества потоков
  }
}

#>>>>> В шестой лабораторной работе будет предложен универсальный способ блокировки
при помощи ipset, который работает значительно быстрее.

# функция разбана IP-адресов по прошествии 10 минут
# из таблицы banlist выбираются все IP-адреса, занесённые более 10 минут назад
$sth = $dbh->prepare("SELECT id, INET_NTOA(ip) FROM banlist WHERE
time<CURRENT_TIMESTAMP - INTERVAL 10 MINUTE;"); # подготовка запроса к БД
$sth->execute; # выполнение запроса к БД
while ($ref = $sth->fetchrow_arrayref) { # цикл по всем выбранным IP-адресам
  system "iptables -D fail2ban-galcev -s ".$$ref[1]." -j DROP"; # разблокировка IP-
адреса на сервере с помощью iptables
  # удаление IP-адреса из БД
  my $sth2 = $dbh->prepare("DELETE FROM banlist WHERE id='".$$ref[0]."';"); #
подготовка запроса
  $sth2->execute; # выполнение запроса
  $sth2->finish; # очистка памяти
}
#<<<<<

$sth->finish; # очистка памяти
$dbh->disconnect; # закрытие подключения к БД
close(LogFile); # закрытие файла журнала
```

```
exit( 0 ); # успешное завершение скрипта
```

Не забудьте создать директорию /var/nfdump/ для архива файлов nfdump в текстовом виде. Обратите внимание, что эта директория будет быстро заполняться новыми файлами и вам придётся периодически её очищать. Если вы не хотите содержать полный архив – закомментируйте копирование файлов в эту директорию в начале скрипта.

Проверить работу скрипта можно выполнив следующие команды:

```
# Подключение к MySQL
mysql –user=nfdump_pl –p

# Вход в контекст БД nfdump_pl
USE `nfdump_pl`

# Запрос 10 последних записей из таблицы flow_cnt
SELECT * FROM `flow_cnt` ORDER BY `ts` DESC LIMIT 10;
```

Перед вами должна появиться таблица следующего вида:

```
+---------------------+-------+
| ts                  | flows |
+---------------------+-------+
| 2014-06-04 23:49:01 |   143 |
| 2014-06-04 23:48:01 |   164 |
| 2014-06-04 23:47:00 |   138 |
| 2014-06-04 23:46:00 |   198 |
| 2014-06-04 23:45:00 |   163 |
| 2014-06-04 23:44:00 |   256 |
| 2014-06-04 23:43:00 |   183 |
| 2014-06-04 23:42:00 |   175 |
| 2014-06-04 23:41:01 |   232 |
| 2014-06-04 23:40:01 |   203 |
+---------------------+-------+
10 rows in set (0.00 sec)
```

В ней отображены последние 10 записей, которые скрипт сделал в БД. Первая запись должна быть сделана меньше минуты назад.

Для того чтобы иметь возможность отлаживать скрипт и видеть ошибки, происходящие в нём рекомендуется вышеуказанный скрипт назвать /usr/local/sbin/nfdump2.pl, а в качестве файла /usr/local/sbin/nfdump.pl использовать нижеследующий:

```
#!/usr/bin/perl
use 5.8.8; use strict; use warnings; use DBI; # подключение Perl библиотек

# Выполнение скрипта nfdump2.pl и добавление стандартного вывода и вывода ошибок в
файл журнала.
system "/usr/local/sbin/nfdump2.pl >> /var/log/nfdump.pl.errors.log 2>&1";

exit( 0 ); # успешное завершение скрипта
```

Этот скрипт избавит нас от проблем с программой nfdump при поломке главного скрипта, а так сохранит информацию для отладки в файл журнала.

В связи с тем, что в директории /var/cache/nfdump/ постоянно создаются и удаляются файлы (она участвует в конвертации из бинарного формата nfdump в текстовый) создаётся большая нагрузка на систему хранения данных компьютера. Все эти файлы временные, поэтому в этой директории правильно будет использовать файловую систему tmpfs. Её смысл в том, чтобы не записывать временные файлы на диск, а отвести часть оперативной памяти для их хранения. Таким образом, производительность доступа к этим файлам значительно повышается, а износ системы хранения данных уменьшается. Для этого необходимо добавить в файл /etc/fstab следующую информацию:

```
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
nfdump        /var/cache/nfdump  tmpfs   size=24M        0       0
```

Что означает, что в директории /var/cache/nfdump/ будет использоваться файловая система tmpfs размеров в 24Мб. Далее приведён список команд для внесения изменений:

```
# Добавляет необходимую строку в файл fstab
echo "nfdump /var/cache/nfdump tmpfs size=24M 0 0" >> /etc/fstab

# Монтирование новой файловой системы
mount nfdump

# Проверка
df -h /var/cache/nfdump

# Размонтирование файловой системы при необходимости
umount nfdump
```

Когда вы убедитесь, что скрипт работает, и БД наполняется можно перейти к следующему этапу – создание страницы для отображения общей статистики по потокам. Ниже приведёт PHP скрипт flows.php, который делает выборку из базы данных и выводит на страницу среднее значение потоков в минуту за прошлые 1, 5, 10, 30 и 60 минут. Страница автоматически обновляется каждые 10 секунд.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Refresh" content="10" />
  <title>Flows Statistics</title>
</head>
<body>
<?

// Параметры подключения к БД
```

```
$hostname = "localhost";
$database = "nfdump_pl";
$user = "nfdump_pl";
$password = "password";

// Подключение к MySQL
$link = mysql_connect($hostname, $user, $password);
if (!$link) {
    die('Ошибка соединения: ' . mysql_error());
}

// Переход в контекст БД
$db_selected = mysql_select_db($database, $link);
if (!$db_selected) {
    die ('Не удалось выбрать базу foo: ' . mysql_error());
}

// Выборка последних 60 записей из таблицы flow_cnt
$result = mysql_query("SELECT flows FROM flow_cnt ORDER BY ts DESC LIMIT 60");
if (!$result) {
    die ('Ошибка запроса: ' . mysql_error());
}

$cnt=0; // количество элементов (минут в данном случае) в выборке
$sum=0; // сумма потоков во всех полученных записях

while($row=mysql_fetch_array($result)) { // цикл по всем полученным записям
    $sum+=$row[0]; // прибавляем к сумме количество потоков в текущей записи
    $cnt++; // количество элементов (минут) в выборке +1
    if($cnt==1)echo $sum." flow/min (last minute)<br />\n";
    // если было просуммировано пять элементов (минут) выборки, то делим сумму на 5 и
выводим округлённый результат на страницу
    else if($cnt==5)echo round($sum/5)." flow/min (last 5 minute)<br />\n";
    else if($cnt==10)echo round($sum/10)." flow/min (last 10 minute)<br />\n";
    else if($cnt==30)echo round($sum/30)." flow/min (last 30 minute)<br />\n";
    else if($cnt==60)echo round($sum/60)." flow/min (last hour)<br />\n";
}

mysql_close($link); // закрываем подключение к MySQL

?>
</body>
</html>
```

Указанный скрипт необходимо разместить на веб-сервере, в директории, в который исполняются PHP скрипты. Чтобы увидеть результат обратитесь к скрипту при помощи браузера, например http://localhost/flows.php.

[pma] www.phpmyadmin.net