Содержание

Простой DDoS анализ через CLI

Проблема

Исходные данные: канал 10Гбит, периодическая мощная DDoS атака на один из ip в сети приводит к деградации сервиса. График DDoS атаки далее, видно, что мощность DDoS атаки в сумме с текущим трафиком превышает емкость канала.

Решение

Так как быстро расширить канал и увеличить мощность DPI не представлялось возможным, выбрали следующий путь: 1. вычисляем список IP которые подверглись DDoS 2. переводим IP в null route (в blackhole)

Настройка

1. создаем директорию /home/ddos_check
2. устанавливаем ipfixreceiver2 на сервер, где будем собирать данные полный netflow используем следующую конфигурацию файл ipfixreceiverflow2.ini, не требуется гарантированная доставка поэтому используем UDP транспорт

```
[connect]
#protocol=tcp
protocol=udp
host=0.0.0.0
port=1599

[dump]
rotate_minutes=1
processcmd=/home/ddos_check/rcflowprocess %%s
dumpfiledir=/home/ddos_check/flow/

[InfoModel]
XMLElements = /etc/rcollector/xml/raw_flow.xml

[Template]
Elements = octetDeltaCount, packetDeltaCount, protocolIdentifier,
ipClassOfService, sourceTransportPort, sourceIPv4Address,
sourceIPv6Address, destinationTransportPort, destinationIPv4Address,
destinationIPv6Address, bgpSourceAsNumber, bgpDestinationAsNumber,
flowStartMilliseconds, flowEndMilliseconds, ingressInterface,
egressInterface, ipVersion, session_id, host_cn, DPI_PROTOCOL, login,
postNATSourceIPv4Address, postNAPTSourceTransportPort,
frgmt_delta_packs, repeat_delta_pack, packet_deliver_time
```

```
[ExportModel]
Elements = session_id, octetDeltaCount, protocolIdentifier,
DPI_PROTOCOL, sourceTransportPort, sourceIPv4Address : decode_unsigned,
destinationTransportPort, destinationIPv4Address : decode_unsigned,
bgpSourceAsNumber, bgpDestinationAsNumber, flowStartMilliseconds :
decode_unsigned, flowEndMilliseconds : decode_unsigned, login,
postNATSourceIPv4Address : decode_unsigned,
postNAPTSourceTransportPort, packetDeltaCount, sourceIPv6Address,
destinationIPv6Address


[logging]
loggers.root.level = information
loggers.root.channel = fileChannel
channels.fileChannel.class = FileChannel
channels.fileChannel.path = /var/log/ipfixreceiverflow2.log
channels.fileChannel.rotation = 1 M
channels.fileChannel.archive = timestamp
channels.fileChannel.purgeCount = 5
channels.fileChannel.formatter.class = PatternFormatter
channels.fileChannel.formatter.pattern = %Y-%m-%d %H:%M:%S.%i [%P] %p
%s - %t
channels.fileChannel.formatter.times = local
```

сохраняем файл в /home/ddos_check

3. вносим в iptables разрешение для порта 1599 - UDP, запускаем ipfixreceiver.

```
ipfixreceiver2 --daemon --umask=000 --
pidfile=/var/run/ipfixreceiver.1599.pid -f
/home/ddos_check/ipfixreceiverflow2.ini
#проверяем, что порт процесс слушает
netstat -anpl | grep 1599
udp   124968      0 0.0.0.0:1599                0.0.0.0:*
21820/ipfixreceiver
```

4. создаем файл обработки потока /home/ddos_check/rcflowprocess

```
#!/bin/bash

export
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin:/bin:/
home/ddos_check
export LD_LIBRARY_PATH=/usr/local/lib:/usr/local/lib:/usr/local/lib

gzip $1
echo "DDoS statistics" > /home/ddos_check/lastminute.txt
date >> /home/ddos_check/lastminute.txt

echo -e "\nSessions TOP20\t ip" >> /home/ddos_check/lastminute.txt
/home/ddos_check/topcnt $1.gz >> /home/ddos_check/lastminute.txt
```

```
echo -e "\n\nSummary bytes Only Destination TOP20\nsize(Mb) \t ip" >>
/home/ddos_check/lastminute.txt
/home/ddos_check/topsize $1.gz 2 | sort -n -r| head -20 | awk '{print
$1/1024/1024 " " $2'} >> /home/ddos_checklastminute.txt

echo -e "\n\nSummary bytes Only src IP TOP20\nsize(Mb) \t ip" >>
/home/ddos_check/lastminute.txt
/home/ddos_check/topsize $1.gz 1 | sort -n -r| head -20 | awk '{print
$1/1024/1024 " " $2'} >> /home/ddos_check/lastminute.txt

echo -e "\n\nSummary bytes by src+dsc TOP20\nsize(Mb)\tip" >>
/home/ddos_check/lastminute.txt
/home/ddos_check/topsize $1.gz 0 | sort -n -r| head -20 | awk '{print
$1/1024/1024 " " $2'} >> /home/ddos_check/lastminute.txt


mv -f /home/ddos_check/lastminute.txt
/var/www/html/ddos_check/lastminute.txt
chown apache /var/www/html/ddos_check/lastminute.txt
chmod a+w  /var/www/html/ddos_check/lastminute.txt
chcon -v --type=httpd_sys_content_t
/var/www/html/ddos_check/lastminute.txt
```

 не забываем сделать chmod a+x /home/ddos_check/rcflowprocess

5. создаем скрипт расчета к-ва сессий top-20 /home/ddos_check/topcnt

```
function getipv4() {
  s=`echo "obase=16; " $1 | bc | sed 's/../0x& /g'`
  ip=`printf '%d.%d.%d.%d' $s`
  echo -n $ip
}

zcat $1 | awk -F '\t' '{print $6 "\n" $8}' | sort | uniq -c -d | sort -
n -r > tmp$$$

head -20 tmp$$$ > tmp2$$$

echo -e "ip\t\thits"
while read p; do
 cnt=`echo $p | awk '{print $1}'`
 ipd=`echo $p | awk '{print $2}'`
 size=`/home/volja/cntsums $1 $ipd | awk '{print $2/1024/1024}'`
 getipv4 $ipd; echo -n -e "\t" $cnt; echo -e "\t s=" $size "(Mb)\t("
$ipd ")"
done < tmp2$$$


rm -f tmp$$$ tmp2$$$
```

 не забываем сделать chmod a+x /home/ddos_check/topcnt

6. устанавливаем bc

```
 yum -y install bc
```

7. создаем скрипт /home/ddos_check/cntsums

```
zcat $1 | grep $2 | awk '{sum += $2; sum2 += $3} END {print "octets= "
sum "\tpackets= " sum2}'
```

 не забываем сделать chmod a+x /home/ddos_check/cntsums

8. создаем скрипт ТОП-20 по максимальному объему трафика /home/ddos_check/topsize

```
#!/usr/bin/python
# usage:
# by source ip
#./topsize arch/attak1/url_05102017_152100.dump.gz 1 | sort -n -r| head
-20 | awk '{print $1/1024/1024 " " $2'}
# by destination ip
#./topsize arch/attak1/url_05102017_152100.dump.gz 2 | sort -n -r| head
-20 | awk '{print $1/1024/1024 " " $2'}
#

import sys, os, logging, ConfigParser, gzip

def main():
    delim='\t'
    accumulater={}
    for line in openinfile(sys.argv[1]):
        acc=long(0)
        fields = line.rstrip('\n').split(delim)
        if(sys.argv[2]=="0" or sys.argv[2]=="1"):
           # by src
           try:
              acc=accumulater[fields[5]]
              accumulater[fields[5]]=acc+long(fields[1])
           except KeyError:
              accumulater[fields[5]]=long(fields[1])
        if(sys.argv[2]=="0" or sys.argv[2]=="2"):
           #by dsc
           try:
              acc=accumulater[fields[7]]
              accumulater[fields[7]]=acc+long(fields[1])
           except KeyError:
              accumulater[fields[7]]=long(fields[1])

    for key, value in accumulater.iteritems():
```

```
        print str(value)+' '+ipv4str(long(key))


# open input file
def openinfile(filename):
    if filename is None:
        inf = sys.stdin
        logging.debug("input file: stdin")
    else:
        if ".gz" in filename:
            inf = gzip.open(filename, "rb")
        else:
            inf = open(filename, "rb")
        logging.debug("input file: " + filename)
    return inf

def ipv4str(ipv4):
    return str((ipv4 >> 24) & 0xFF) + '.' + str((ipv4 >> 16) & 0xFF) +
'.' + str((ipv4 >> 8) & 0xFF) + '.' + str((ipv4 & 0xFF))

if __name__ == "__main__":
    main()
```

 не забываем сделать chmod a+x /home/ddos_check/topsize

9. добавляем строку в /var/spool/cron/root что бы через сутки данные flow удалялись

```
15 4 * * * /bin/find /home/ddos_check/flow/ -name url_\*.dump.gz -cmin
+1440 -delete > /dev/null 2>&1
```

10. устанавливаем параметры на DPI

```
netflow=8
netflow_full_collector_type=1
netflow_dev=eth2
netflow_timeout=10
#!!!здесь укажите ip адрес вашего приемника
netflow_full_collector=127.0.0.1:1599
netflow_passive_timeout=20
netflow_active_timeout=60
```

 данная настройка требует рестарта

Работа со скриптами

После перезагрузки fastdpi если все сконфигурировали правильно ipfixreceiver начнет принимать данные в директорию /home/ddos_check/flow, после появления *.gz файлов (1 раз в

минуту интервал ротации был установлен в приемнике) проводим проверку скриптов.

```
cd /home/ddos_check
./topcnt flow/url_05102017_151800.dump.gz
ip              hits
77.XXX.XX.64     144889  s= 5379.99 (Mb)        ( 1299787840 )
77.88.8.8        14051   s= 2.26185 (Mb)        ( 1297614856 )
128.128.128.8    1642    s= 0.401568 (Mb)       ( 134744072 )
77.88.8.1        1578    s= 0.359544 (Mb)       ( 1297614849 )
...
77.XXX.XX.208    468     s= 1.45243 (Mb)        ( 1299785168 )
```

видим 1 шт. ip в топе с очень большим отрывом, этот адрес ddos'ят из вне забивая канал.
Проверяем top по размеру по всем src и dsc IP: ./topsize arch/attak1/url_05102017_151800.dump.gz 0 | sort -n -r| head -20 | awk '{print $1/1024/1024 " " $2'} 5380.01 77.XXX.XX.64 165.881 81.XXX.XXX.79 ... 27.2184 74.XXX.XXX.27 видим что в текущую минуту данный IP получил и отдал 5.4 Гбайт трафика, следующий только 165 Мбайт. Таким образом подтвердилось предположение о DDoS атаке на IP=77.XXX.XX.64.
Данный адрес отправляем в blackhole, так как мощности канала не хватает, т.е. блокировать можно только у вышестоящего провайдера с помощью null route.

Настройка httpd

Для оперативного доступа при наличии http на сервере где происходит прием IPFX можно прописать в конфигурации /etc/httpd/conf/httpd.conf:

```
<Directory "/var/www/html/ddos_check">
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

Alias /ddos_check/ "/var/www/html/ddos_check/"
```

Желательно еще добавить ограничение доступа, в данном примере не рассматривается.
Соответственно после рестарта httpd можно получить страницу по ссылке

```
http://<your_ip_http>/ddos_check/lastminute.txt
```

Что в остатке?

Интересно было понять м.б. можно заблокировать только ip извне? или их там пара десятков? создаем скрипт для подсчета мощности атаки /home/ddos_check/topip:

```
#!/bin/bash
```

```
function getipv4() {
  s=`echo "obase=16; " $1 | bc | sed 's/../0x& /g'`
  ip=`printf '%d.%d.%d.%d' $s`
  echo -n $ip
}

zcat $1 | grep $2 | awk -F '\t' '{print $8 "\n" $6}' | grep -v $2 | sort |
uniq -c -d | sort -n -r > tmp$$$

echo -e "ip\t\thits"
echo -n "unique ip="; wc -l tmp$$$ | awk '{print $1}'
head -20 tmp$$$ > top20$$$
while read p; do
 cnt=`echo $p | awk '{print $1}'`
 ipd=`echo $p | awk '{print $2}'`
 getipv4 $ipd; echo -e "\t" $cnt
done <top20$$$

rm -f tmp$$$ top20$$$
```

 не забывает сделать chmod a+x /home/ddos_check/topip запускаем, берем в качестве 2-го параметра десятичное представление IP из () в TOP-20 сессий:

```
./topip 'arch/attak1/url_05102017_15*' 1299787840
ip              hits
unique ip=58261
202.92.200.6     2626
110.76.131.6     2546
119.235.28.59    2150
38.70.202.194    1874
177.38.144.14    1830
138.204.18.18    1542
212.119.180.222  1452
88.220.134.2     1432
200.186.13.86    1389
...
```

как видно из таблицы в атаке участвуют 58 тысяч адресов.

Ссылки по теме

- Защита от DDOS атак средствами BGP