Table of Contents

Simple DDoS analysis

Challenge

Initial conditions: 10Gbit channel, periodic powerful DDoS attack to one of the network ip-address takes place which leads to service degradation. The DDoS attack diagram is shown below, it shows that the DDoS attack power with the current traffic power in total exceeds the channel capacity.

Solution

Since it's impossible to quickly expand the channel capacity and increase the DPI power, pursue the following sequence of actions: 1. get the list of IP addresses subjected to DDoS 2. place the IP in the null route (to a blackhole)

Configuration

1. create /home/ddos_check directory
2. install the ipfixreceiver on the server, where the full netflow data will be gathered. Use the following configuration - ipfixreceiverflow2.conf file. The guaranteed delivery isn't required so we use UDP transport

```
[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
```

save file in /home/ddos_check

3. set permission for the 1599 UDP port in iptables, launch the ipfixreceiver.

```
ipfixreceiver2 --daemon --umask=000 --
pidfile=/var/run/ipfixreceiver.1599.pid -f
/home/ddos_check/ipfixreceiverflow2.ini
# check out that the process is listening
netstat -anpl | grep 1599
udp   124968      0 0.0.0.0:1599                0.0.0.0:*
21820/ipfixreceiver
```

4. create flow processing file: /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
```


Don't forget chmod a+x /home/ddos_check/rcflowprocess

5. create the script to calculate the number of sessions 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$$$ tmp2$$$
```

Don't forget chmod a+x /home/ddos_check/topcnt

6. install bc

```
 yum -y install bc
```

7. create the script /home/ddos_check/cntsums

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

Don't forget chmod a+x /home/ddos_check/cntsums

8. create the script top-20 /home/ddos_check/topsize on the basis of maximum traffic value

```
#!/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()
```

 Don't forget chmod a+x /home/ddos_check/topsize

9. add the line to the /var/spool/cron/root in order the flow data to be deleted within 24 hours

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

10. configure DPI settings

```
netflow=8
netflow_full_collector_type=1
netflow_dev=eth2
netflow_timeout=10
#!!!here the source ip should be specified
netflow_full_collector=127.0.0.1:1599
netflow_passive_timeout=20
netflow_active_timeout=60
```

 this setting requires the restart

Working with scripts

If you've configured fasdpi properly the ipfixreceiver will start to receive data in the /home/ddos_check/flow directory ater fastdpi restart, after the occurring *.gz files (once in a second -

timing of rotation is being set in the source) check the scripts.

```
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 )
```

observe that one ip info from the top is quite different, it means the address is to be subjected the DDoS attack and is overflowed from the outside. Check the top by size all over the src and dst 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 observe that at the moment this IP received and send 5.4 GB of traffic, the next only 226 MB. Thus, the assumption about DDoS attack on IP = 77.XXX.XX.64 is confirmed. We move this address to the blackhole, the channel capacity is not enough, so it can be blocked only by the superior provider using the null route.

httpd configuration

If you have http on your server where the IPFIX is reveived then it can be specified in the /etc/httpd/conf/httpd.conf configuration file:

```
<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/"
```

It is desirable to add an access restriction, but in this example it is not considered.
Accordingly, after httpd restart you are able to get the page (shown below) following the link

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

So what does that leave us?

It is interesting to understand if there is a possibility simply to block an ip from the outside? Or there present a couple of dozen?
Create a script for calculating the attack power /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$$$ top20$$$
```

 Don't forget chmod a+x /home/ddos_check/topip launch it, put the decimal IP representation from () within the TOP-20 sessions as the second option:

```
./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
...
```

As shown in the table above DDoS attack involves 58 thousands of addresses.

Related links

- DDOS protection using BGP