INTERNET SECURITY CENTER

# Improve DDoS Botnet Tracking with Honeypots


Ya Liu

Network Security Research Lab, Qihoo 360


botconf2016
The botnet fighting conference
30 NOVEMBER - 2 DECEMBER 2016
LYON - FRANCE
4th edition

# Agenda

- About DDoS botnet tracking
- DDoS botnet families and their PGAs (Packet Generation Algorithm)
- Backscatter collection and analysis
- PGA analysis
- Experiments

# DDoS botnet tracking

360 INTERNET SECURITY CENTER

- It’s aimed to learn botnet assisted DDoS attacks
  - ***4w***: ***w***ho is being attacked by ***w***hat botnet families under ***w***hich C2 controllers with ***w***hat set of attacking parameters (e.g., attack type)

```
2016/11/23 15:15:07     mirai   securityupdates.us      5.188.232.103   23      ddos    tcp_ack_flood, target=109.163.224.34, mask_bits=32, atk_time=60, payload_size=1
2016/11/23 15:15:08     mirai   timeserver.host 188.209.49.106  23      ddos    tcp_ack_flood, target=109.163.224.34, mask_bits=32, atk_time=60, payload_size=1
2016/11/23 15:50:27     mirai   cnc.routersinthis.com   93.158.212.81   23      ddos    udp_flood, target=217.68.245.94, mask_bits=32, port=80, atk_time=100, port=80
2016/11/23 15:50:27     mirai   ftp.xenonbooter.xyz     93.158.212.81   23      ddos    udp_flood, target=217.68.245.94, mask_bits=32, port=80, atk_time=100, port=80
2016/11/23 17:18:10     mirai   cnc.routersinthis.com   93.158.212.81   23      ddos    udp_flood, target=82.144.163.26, mask_bits=32, port=80, atk_time=100, port=80
2016/11/23 17:18:10     mirai   ftp.xenonbooter.xyz     93.158.212.81   23      ddos    udp_flood, target=82.144.163.26, mask_bits=32, port=80, atk_time=100, port=80
2016/11/23 17:26:37     mirai   cnc.routersinthis.com   93.158.212.81   23      ddos    udp_flood, target=94.14.175.22, mask_bits=32, port=80, atk_time=100, port=80
2016/11/23 17:26:37     mirai   ftp.xenonbooter.xyz     93.158.212.81   23      ddos    udp_flood, target=94.14.175.22, mask_bits=32, port=80, atk_time=100, port=80
2016/11/23 17:51:30     mirai   cnc.routersinthis.com   93.158.212.81   23      ddos    udp_flood, target=90.221.219.57, mask_bits=32, port=80, atk_time=100, port=80
2016/11/23 17:51:30     mirai   ftp.xenonbooter.xyz     93.158.212.81   23      ddos    udp_flood, target=90.221.219.57, mask_bits=32, port=80, atk_time=100, port=80
```

time | family | C2 domain | C2 IP/port | command type | attack target & parameters

# Stats on our tracking

360 INTERNET SECURITY CENTER

- Our tracking started in 2014
- 30+ botnet families
- 6,000+ successfully tracked botnets
- 800+ million received attack commands
- 250K+ checked attack targets
- Our data has been shared many times with colleagues outside of our company

# How to evaluate it?

- For evaluation purpose, we need to know:
  - what ***family-unknown*** botnets are active in the wild?
  - what ***family-known*** C2 controllers are outside of our tracking list?

- Therefore we need information about the real attacks, and a method to connect them to the used botnet families

# DDoS backscatter

- It’s generated due to the use of spoofed source IPs in attacking packets
  - e.g., TCP SYN-ACKs acknowledged to spoofed SYNs
- It’s known as a cause of dark space traffic in parallel with scanning and network misconfigurations
- Solutions to detect & monitor DDoS attacks based on backscatters have been proposed in the past years

# Darknet? Or honeypot?

360 INTERNET SECURITY CENTER

| | Pros. | Cons. |
|---|---|---|
| Darknet | ☐ Collect a large number of packets destined to a block of unused addresses | ☐ Non-trivial deployment |
| Honeypot | ☐ Cost effective<br>☐ Easy to deploy | ☐ Less packets collected |

# Our scheme

360 INTERNET SECURITY CENTER

- Full traffic captures are taken on our dozens of low-interaction honeypots

- A special mechanism is designed to separate the ***wanted*** traffic from the ***unwanted***
  - *wanted*: traffic generated by honeypot applications
  - *unwanted*: scans, backscatters, etc.

# PGA: Packet Generation Algorithm

- In modern botnets, attacking packets are usually generated by the bots according some specific algorithm which we call PGA

- PGA attributes:
  - It's attack type specific
  - It's usually family specific
  - Fixed patterns usually exist in the generated packets

- Botnet families can by identified by PGA signatures

# MIRAI’s PGA for *stomp* attack

360 INTERNET SECURITY CENTER

*[*] copied from attack_tcp_stomp() of attack_tcp.c*

```
            iph->check = 0;
            iph->check = checksum_generic((uint16_t *)iph, sizeof (struct iphdr));

            tcph->seq = htons(stomp_data[i].seq++);
            tcph->ack_seq = htons(stomp_data[i].ack_seq);
            tcph->check = 0;
```

fixed “***0x0000***” can be found in ***tcph->seq*** and ***tcph->ack_seq***

# MIRAI's PGA for *gre_eth* attack

INTERNET SECURITY CENTER

*[*]copied from attack_gre_eth () in attack_gre.c*

```
if (ip_ident == 0xffff)
{
    iph->id = rand_next() & 0xffff;
    greiph->id = ~(iph->id - 1000);
}
if (sport == 0xffff)
    udph->source = rand_next() &      ffff;
if (dport == 0xffff)
    udph->dest = rand_ne

if (!gcip)
    greiph->daddr = rand_
else
    greiph->daddr = iph->daddr;
```

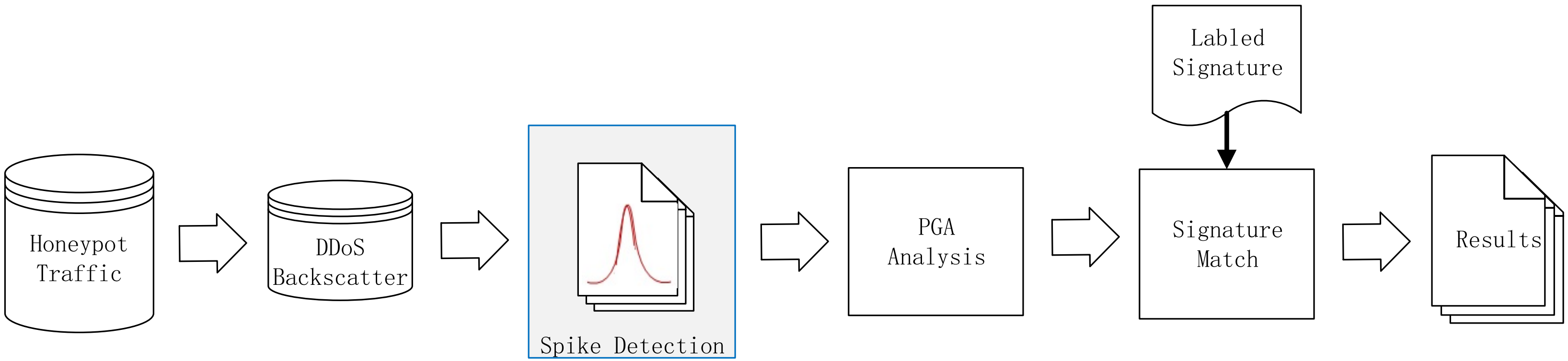
***greiph->id*** is bound to ***iph->id***

***greiph->daddr*** is bound to ***iph->addr***

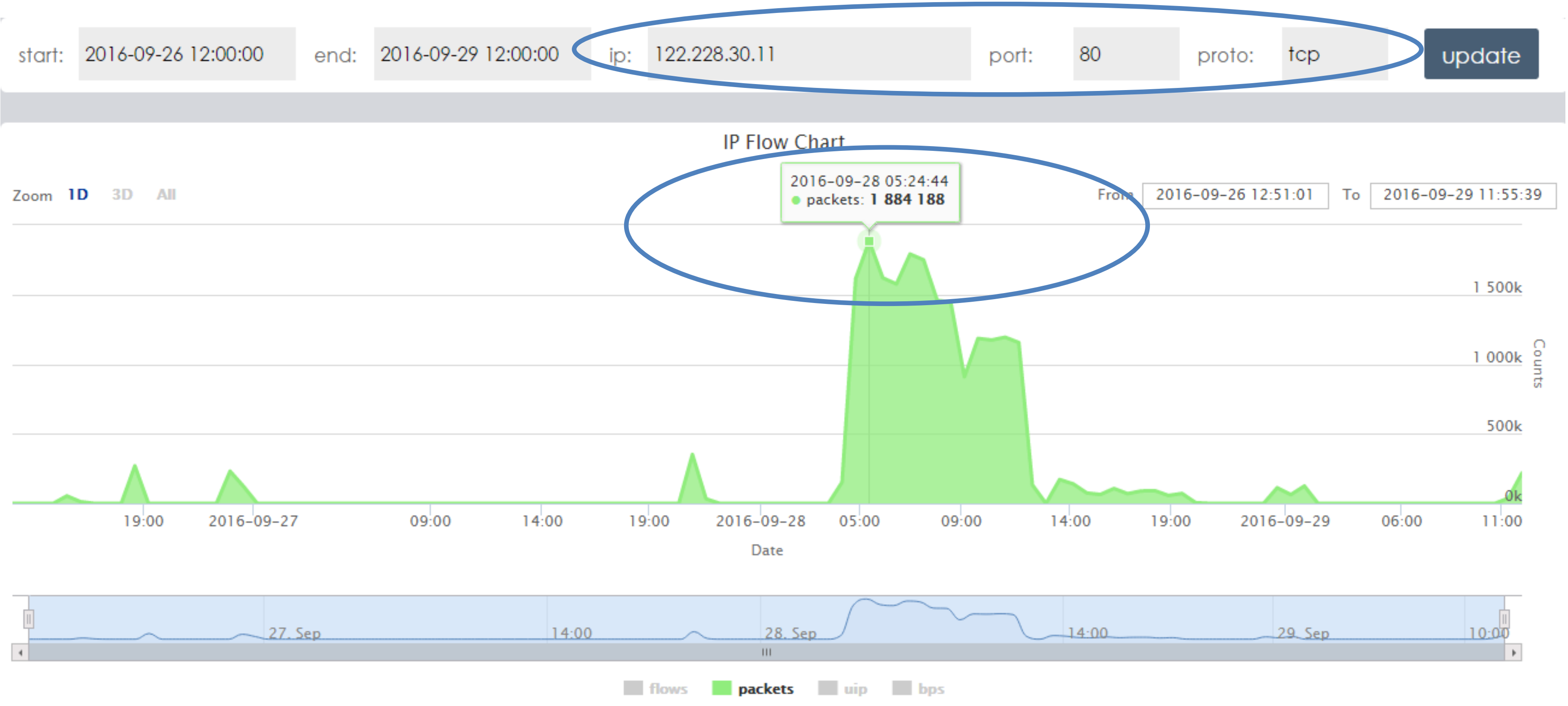
# From packets to bot families

- Buggy implementation and design flaws lead to PGA signatures which can be characterized by their packets
- PGA signatures can also be concluded by reverse engineering the bot sample
- It’s possible to correlate an DDoS attack to the used botnet families by PGA signature matching

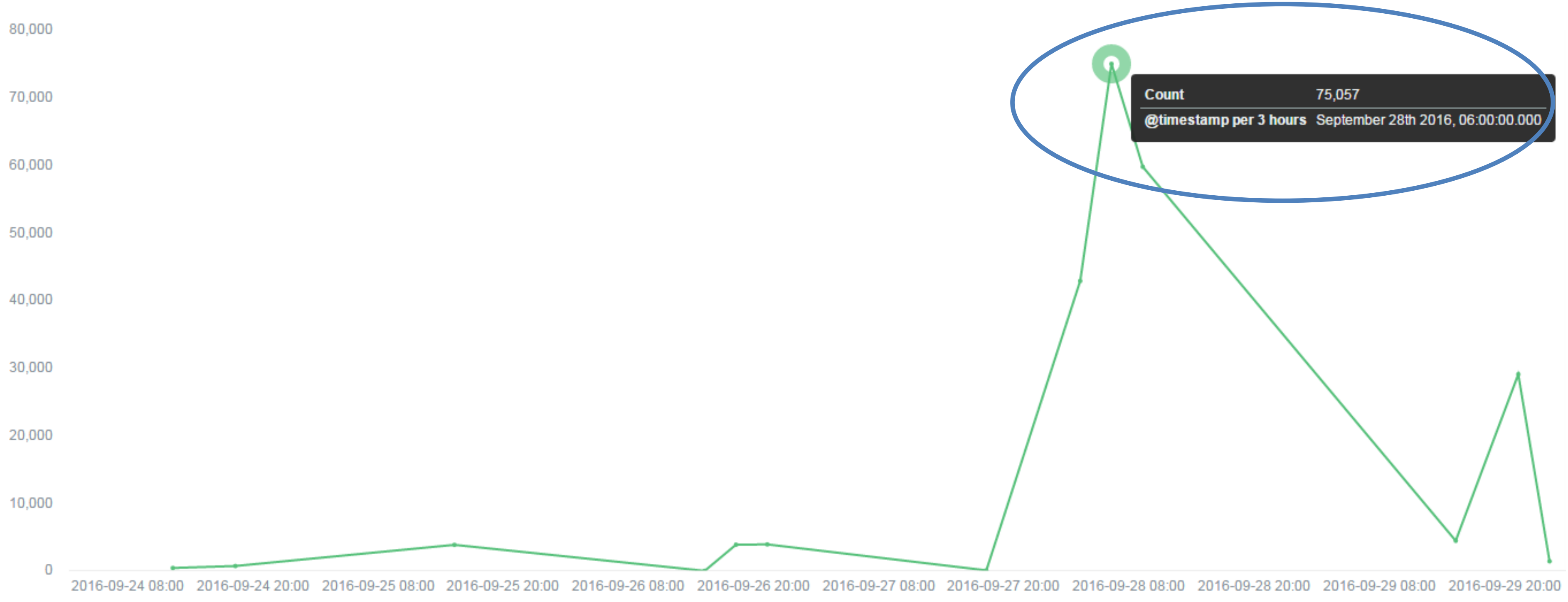
# The architecture overview



# A TCP-SYN spike example



# The reflected spike



# Our spike detection scheme

- Backscatters are sub-grouped based on:
  - {packet type, source IP, source port}
  - or {packet type, queried domain} in case DNS responses

| Policy name | Description |
|---|---|
| PACKET_TIME_INTERVAL | If 2 packets' interval is less than this value, they are grouped to the same spike. |
| LEAST_NUMBER_OF_PACKETS | The least number of packets a valid spike MUST have. |
| LEAST_NUMBER_OF_HONEYPOTS | The least number of honeypots a valid spike MUST hit. |

# 3 supported backscatters



- TCP SYN-ACK packets for detecting SYN flood
- DNS response packets for detecting query flood
- ICMP unreachable messages
  - ICMP type=3, code=3
  - The original attacking packets could be restored

# Restoring attacking pkt fields

INTERNET SECURITY CENTER

| Backscatter type | Restored attacking packet fields |
| --- | --- |
| TCP SYN-ACK | ☐ source/destination IP/port<br>☐ initial sequence number |
| DNS response | ☐ source/destination IP<br>☐ source port<br>☐ transaction ID (tid for short)<br>☐ queried domain |
| ICMP unreachable message | ☐ sip/dip/sport/dport/ISN for SYN<br>☐ sip/dip/sport/tid for DNS query |

# Packet feature vector & matrix

360 INTERNET SECURITY CENTER

- They are defined to find the fixed patterns in attacking packets
  - And to calculate the spike feature vector
- The vector is constructed by restored field bytes
  - A 16-dimension vector for SYN packet
    - {sip, dip, ISN, sport, dport}
  - A 12-dimension vector for DNS query packet
    - {sip, dip, sport, transaction-id}

# A real SYN packet feature matrix

fixed patterns

21 packets

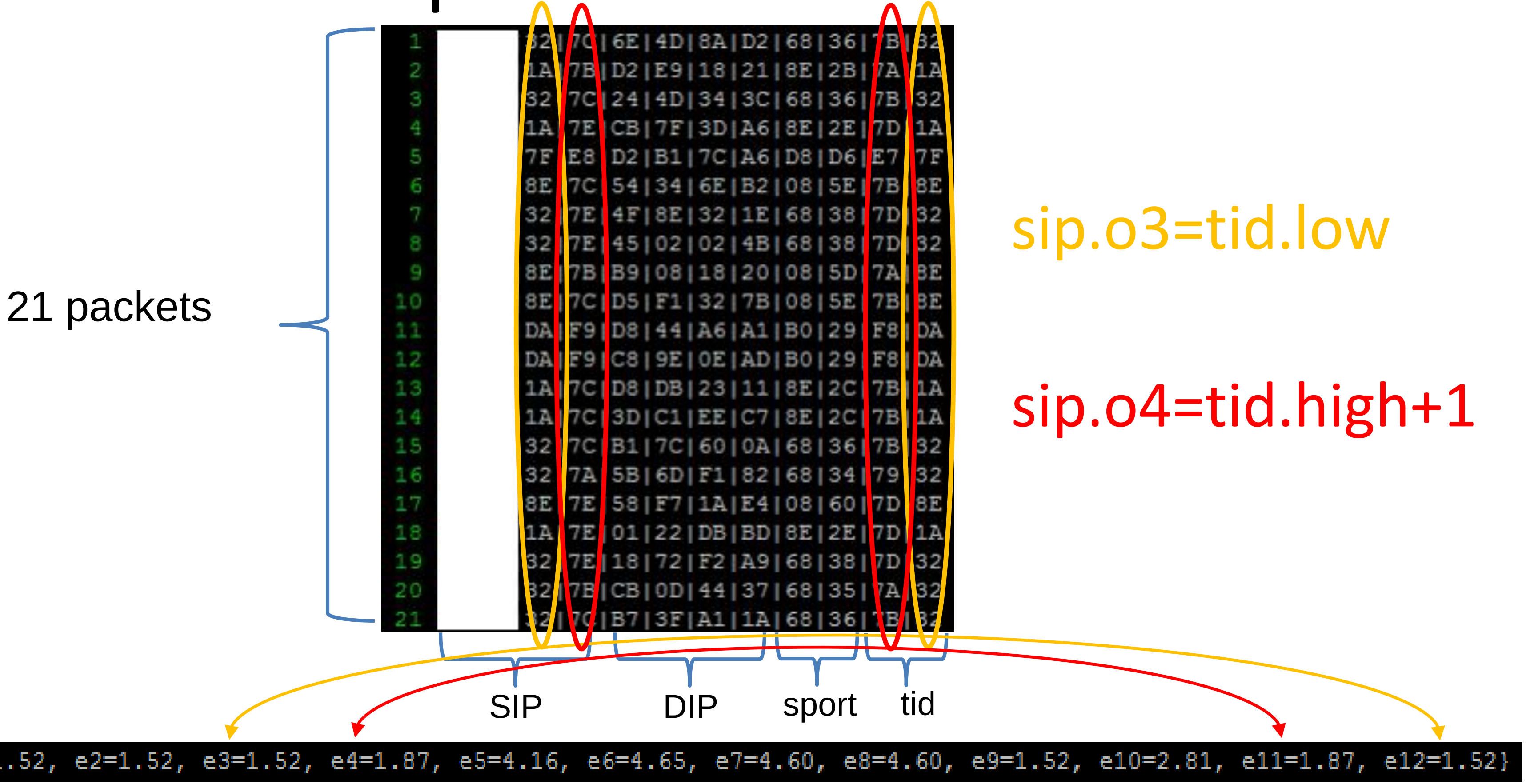
| # | SIP | DIP | ISN | sport | dport |
|---|---|---|---|---|---|
| 1 | [redacted] 7A | 05 2D 4C 5F | E9 F4 00 01 | 51 42 | 00 50 |
| 2 | [redacted] BE | 05 2D 4C 5F | FE E7 00 01 | 7D C7 | 00 50 |
| 3 | [redacted] A0 | 05 2D 4C 5F | 5F F2 00 01 | 1E 5D | 00 50 |
| 4 | [redacted] FD | 05 2D 4C 5F | AB B8 00 01 | C5 A0 | 00 50 |
| 5 | [redacted] 7B | 05 2D 4C 5F | 83 E7 00 01 | 4A DC | 00 50 |
| 6 | [redacted] 9F | 05 2D 4C 5F | F4 8E 00 01 | 5C E3 | 00 50 |
| 7 | [redacted] BD | 05 2D 4C 5F | 2B C5 00 01 | A1 97 | 00 50 |
| 8 | [redacted] B2 | 05 2D 4C 5F | 9D E2 00 01 | 85 BA | 00 50 |
| 9 | [redacted] FA | 05 2D 4C 5F | EC 63 00 01 | D0 46 | 00 50 |
| 10 | [redacted] 7D | 05 2D 4C 5F | B4 7C 00 01 | B0 F3 | 00 50 |
| 11 | [redacted] 7B | 05 2D 4C 5F | E5 94 00 01 | EF 56 | 00 50 |
| 12 | [redacted] 7A | 05 2D 4C 5F | 33 21 00 01 | 53 74 | 00 50 |
| 13 | [redacted] 7D | 05 2D 4C 5F | D8 EE 00 01 | 06 D7 | 00 50 |
| 14 | [redacted] 7A | 05 2D 4C 5F | 16 C4 00 01 | 84 BF | 00 50 |
| 15 | [redacted] 7C | 05 2D 4C 5F | E6 EE 00 01 | 71 A1 | 00 50 |
| 16 | [redacted] FC | 05 2D 4C 5F | 39 53 00 01 | EF 22 | 00 50 |
| 17 | [redacted] F9 | 05 2D 4C 5F | 7B 89 00 01 | 6B D5 | 00 50 |
| 18 | [redacted] F1 | 05 2D 4C 5F | 38 28 00 01 | EE D5 | 00 50 |
| 19 | [redacted] FE | 05 2D 4C 5F | 0F 74 00 01 | 90 4C | 00 50 |
| 20 | [redacted] 7E | 05 2D 4C 5F | 67 C1 00 01 | 02 2A | 00 50 |
| 21 | [redacted] 7B | 05 2D 4C 5F | 15 B3 00 01 | 6D D3 | 00 50 |

# Spike feature vector

- It’s used to find the bound relations among attacking packet fields, and to do spike clustering

- It’s obtained by calculating the *Shannon Entropy* of each column of packet feature matrix

# A real DNS spike feature vector



# Spike clustering & PGA profiling

360 INTERNET SECURITY CENTER

- Spike clustering is to find the similar spikes that are probably generated by the same family

- PGA is profiled in 2 ways:
  - fixed patterns are detected by checking element of 0.0
  - bounds are detected by checking the same elements

- The profiling result is used for signature matching

# About the bound relation

- One field is derived from another one:

  $$field\_n = f(field\_m)$$

- While the simplest bound relation is simple byte sharing, the real situation is complicated

- Our approach only detects the bound relations, with the exact relations left for manual analysis

# About the evaluation

- If a detected spike is not successfully correlated, it means there are family unknown botnets in the wild

- If a spike is successfully correlated, we just check our tracking list to see whether the attack has been tracked or not

# Experiments



- 2,333 SYN-ACK spikes and 1,835 DNS spikes are checked
  - from August, 2015 to October, 2016
- 4 large PGA clusters are found

| Cluster | PGA signatures | Spikes | Botnet family |
|---|---|---|---|
| sa_cls1 | $\{(p_9=p_{13}), (p_{10}=p_{14})\}$ | 1318 | XOR.DDoS |
| sa_cls2 | $\{(p_{13}=0x00), (p_{14}=0x01)\}$ | 131 | *unknown* |
| dns_cls1 | $\{(p_4=p_{11}+1), (p_3=p_{12})\}$ | 626 | *unknown* |
| dns_cls2 | $\{(p_3=p_9), (p_4=p_{10})\}$ | 21 | *unknown* |

# About dns_cls1

- It can be connected to a family-unknown attack tool which supports DNS random subdomain attack

```
2015-11-20 13:04:04 resolver=125.132.239.21, sport=36395, tid=31258, qname=olslix.quanshuwu.com
2015-11-20 13:20:10 resolver=85.46.222.186, sport=2144, tid=32142, qname=irozuz.quanshuwu.com
2015-11-20 13:32:41 resolver=118.125.92.160, sport=2141, tid=31374, qname=wxctwb.quanshuwu.com
2015-11-20 13:36:47 resolver=82.101.215.69, sport=2142, tid=31630, qname=ahwdozqhqbujyx.quanshuwu.com
2015-11-20 13:36:51 resolver=190.0.1.171, sport=30766, tid=32026, qname=ibwnexgnwxurcd.quanshuwu.com
2015-11-20 13:38:42 resolver=31.165.247.192, sport=26680, tid=32050, qname=fydrgrw.quanshuwu.com
2015-11-20 13:47:12 resolver=94.232.149.33, sport=2144, tid=32142, qname=mlmfyzav.quanshuwu.com
2015-11-20 13:51:27 resolver=182.255.72.235, sport=26678, tid=31538, qname=qzgngvcncfkbczwn.quanshuwu.com
```

- It shares the same subdomain pattern with Elknot/BillGates, but has different PGA signature
- It's still active, and mainly used to attack China online-game domains

# Conclusions

360 INTERNET SECURITY CENTER

- A backscatter collection scheme with honeypots
- A spike based attack detection scheme from DDoS backscatters
- A PGA analysis approach based on recovered attacking packet fields

# Q&A

liuya@360.cn