360 INTERNET SECURITY CENTER

The Elknot DDoS Botnets We Watched

Ya Liu, Hui Wang

Network Security Research Lab, Qihoo 360

Agenda

360 INTERNET SECURITY CENTER

- Sample analysis
- The C2 protocol
- Infection vector
- Statistics on the tracked attacks
- A real DDoS attack event against DNS root name servers

About Elknot

- An infamous DDoS bot family being around for years
  - written in C++ , mainly targeting x86 platforms, supporting versatile DDoS attack methods
- 2 versions have been discovered so far
  - The first version is usually named Elknot or Mayday, while the second is known as BillGates
- Our work is about the second version
  - We call it Elknot/BillGates here

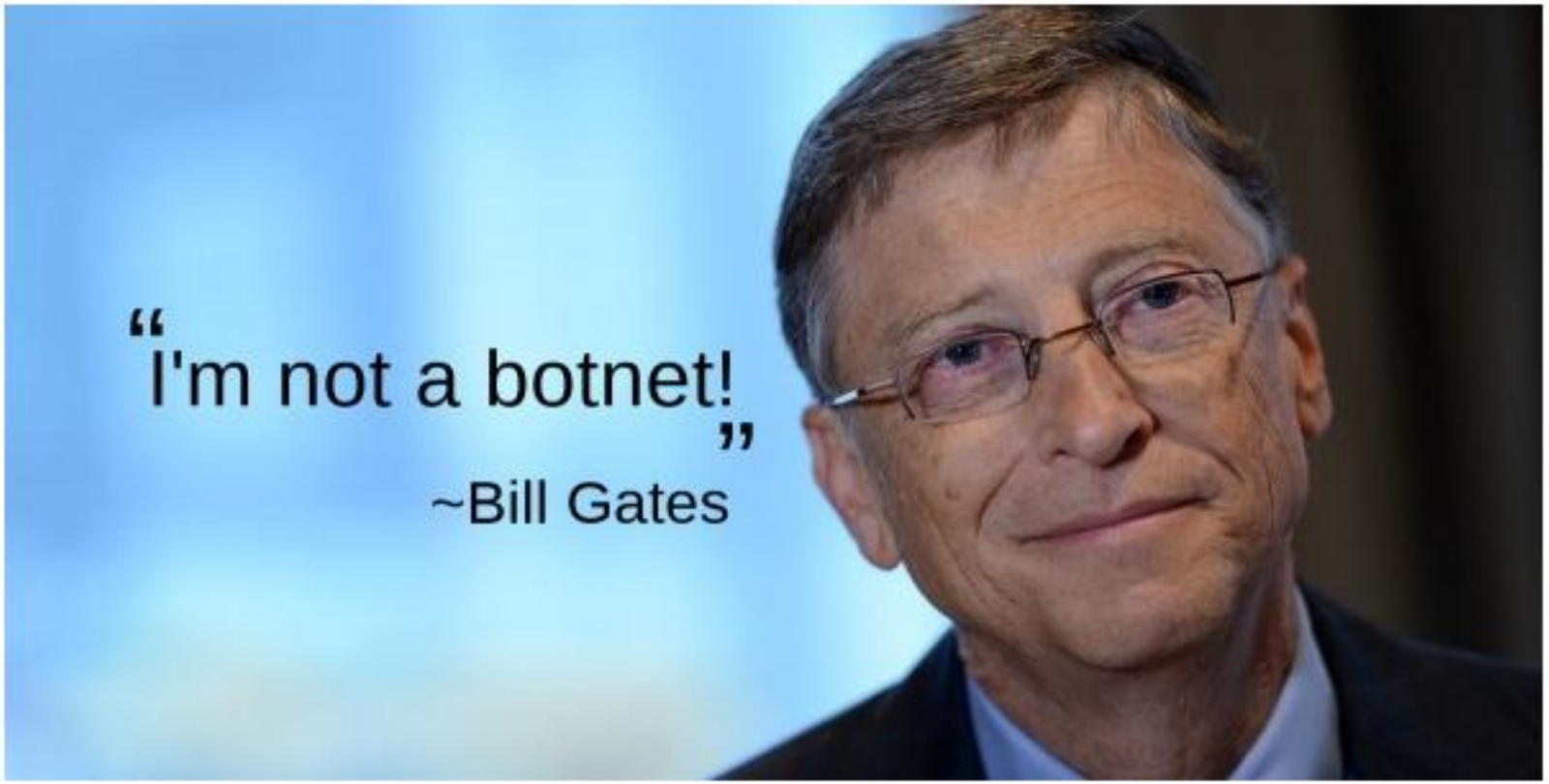
Bill & Gates

360 INTERNET SECURITY CENTER

February 26, 2014 at 17:51

Development → investigate Linux Botnet «BillGates»

Reverse-engineering * Information security *



https://habrahabr.ru/post/213973/

Sample evolution

360 INTERNET SECURITY CENTER

- >= 4 variants exist
  - e.g., in a simplified version of Elknot/BillGates Bill and Gates modules are merged into one single module
- PE.BillGates: "*When ELF.BillGates met Windows*"
  - https://thisissecurity.net/2015/09/30/when-elf-billgatesmet-windows/
- Although the binary code changes a lot, the C2 protocol remains unchanged

The C2 configuration

- The plain configuration is composed of one or more lines
  - MD5: 8285f35183f0341b8dfe425b7348411d
  - line1: ‘abu2.jack52088.com:36665:1:1:buyaocaowo:1’
  - line2: ‘lzj.passwd1.com:30000:1:1:buyaocaowo:1’

- 2 configuration encryption schemes have been found
  - RSA encryption
  - XOR like encryption

RSA encryption

360 INTERNET SECURITY CENTER

- A custom implementation of text book style RSA algorithm is used
  - Encryption: $c \equiv m^e \pmod{n}$
  - Decryption: $c^d \equiv (m^e)^d \equiv m \pmod{n}$

- 2 RSA class names: ***CBigInt*** and ***ClHNs5r***

Class name	Sample count	Example sample
CBigInt	1,710	MD5 hash: 603170ad361f6e098c8681ed264155eb
ClHNs5r	2,115	MD5 hash: 8285f35183f0341b8dfe425b7348411d

RSA big integers

- Big integers of RsaC/RsaD/RsaN are stored in HEX format strings in sample

```
std::string::string((std::string *)&v24);
std::allocator<char>::allocator(&v25);
std::string::string(
  &v23,
  "5F1E29B3C6D0F0DCB909E91C1639F1FBDE3C70159B49386B81397386F9E3117996B2368D72E4C0204F9E56A58DE2A8EA87B76146746F2BE571CB"
  "36CD850431458C75BC15B85EF998C10EF3DB4511FBD1C2C74430147B9F7535420DCD8E60E820566798FCD39290FB7722E078AC0E3B76B6B1C696"
  "B617DA48AEC02EC57E49CF5",
  &v25);
std::allocator<char>::~allocator(&v25);
std::allocator<char>::allocator(&v26);
std::string::string(
  &v22,
  "A9EA3EA8E500AEBAA810A4681FC2C6283E682906B6F00AEAEC8A168CFBBE83442814EF068C0C19788794CBA2B39C581EB80E5C3CE3CCE30274E8DF84B9CA447B",
  &v26);
std::allocator<char>::~allocator(&v26);
std::allocator<char>::allocator(&v27);
std::string::string(
  &v21,
  "B82B4CC4791409B3A7A71D9293700136DE2CD2A61C42DA4D5C7E7EEF75868782C049D7D3CDD52334C99DF52EC576483424061 48A52F3A3BDE03B"
  "2BFAA8821B4E00F3DD81C7E0E765E7599B70D5385BB33040E66CC06237A003919B2849FA45B1F04F8A0F1DA256953E1340157F7FB22E16935EF9"
  "4C3C18014F3D9A8008F52A5",
  &v27);
```

MD5=603170ad361f6e098c8681ed264155eb

- RSA strings can be extracted from samples with tools like “strings” and “egrep”

Example RSA integers

360 INTERNET SECURITY CENTER

strings 603170ad361f6e098c8681ed264155eb |egrep -e "[A-F0-9]{126,}"

```
1 ~$ strings 603170ad361f6e098c8681ed264155eb.sample |egrep -e "[A-Fa-f0-9]{126,}"
2 5F1E29B3C6D0F0DCB909E91C1639F1FBDE3C70159B49386B81397386F9E3117996B2368D72E4C0204
  F9E56A58DE2A8EA87B76146746F2BE571CB36CD850431458C75BC15B85EF998C10EF3DB4511FBD1C2
  C74430147B9F7535420DCD8E60E820566798FCD39290FB7722E078AC0E3B76B6B1C696B617DA48AEC
  02EC57E49CF5
3 A9EA3EA8E500AEBAA810A4681FC2C6283E682906B6F00AEAEC8A168CFBBE83442814EF068C0C19788
  794CBA2B39C581EB80E5C3CE3CCE30274E8DF84B9CA447B
4 B82B4CC4791409B3A7A71D9293700136DE2CD2A61C42DA4D5C7E7EEF75868782C049D7D3CDD52334C
  99DF52EC57648342406148A52F3A3BDE03B2BFAA8821B4E00F3DD81C7E0E765E7599B70D5385BB330
  40E66CC06237A003919B2849FA45B1F04F8A0F1DA256953E1340157F7FB22E16935EF94C3C18014F3
  D9A8008F52A5
```

Decrypt C2 from RSA strings

360 INTERNET SECURITY CENTER

- Run RSA decryptor on the combinations of the extracted big integers

```
6 /opt/elknot/rsa_decryptor   -d A9EA3EA8E500AEBAA810A4681FC2C6283E682906B6F00AEAEC
  8A168CFBBE83442814EF068C0C19788794CBA2B39C581EB80E5C3CE3CCE30274E8DF84B9CA447B -n
   5F1E29B3C6D0F0DCB909E91C1639F1FBDE3C70159B49386B81397386F9E3117996B2368D72E4C020
  4F9E56A58DE2A8EA87B76146746F2BE571CB36CD850431458C75BC15B85EF998C10EF3DB4511FBD1C
  2C74430147B9F7535420DCD8E60E820566798FCD39290FB7722E078AC0E3B76B6B1C696B617DA48AE
  C02EC57E49CF5 -c B82B4CC4791409B3A7A71D9293700136DE2CD2A61C42DA4D5C7E7EEF75868782
  C049D7D3CDD52334C99DF52EC57648342406148A52F3A3BDE03B2BFAA8821B4E00F3DD81C7E0E765E
  7599B70D5385BB33040E66CC06237A003919B2849FA45B1F04F8A0F1DA256953E1340157F7FB22E16
  935EF94C3C18014F3D9A8008F52A5
7 /opt/elknot/rsa_decryptor   -d A9EA3EA8E500AEBAA810A4681FC2C6283E682906B6F00AEAEC
  8A168CFBBE83442814EF068C0C19788794CBA2B39C581EB80E5C3CE3CCE30274E8DF84B9CA447B -n
   B82B4CC4791409B3A7A71D9293700136DE2CD2A61C42DA4D5C7E7EEF75868782C049D7D3CDD52334
  C99DF52EC57648342406148A52F3A3BDE03B2BFAA8821B4E00F3DD81C7E0E765E7599B70D5385BB33
  040E66CC06237A003919B2849FA45B1F04F8A0F1DA256953E1340157F7FB22E16935EF94C3C18014F
  3D9A8008F52A5 -c 5F1E29B3C6D0F0DCB909E91C1639F1FBDE3C70159B49386B81397386F9E31179
  96B2368D72E4C0204F9E56A58DE2A8EA87B76146746F2BE571CB36CD850431458C75BC15B85EF998C
  10EF3DB4511FBD1C2C74430147B9F7535420DCD8E60E820566798FCD39290FB7722E078AC0E3B76B6
  B1C696B617DA48AEC02EC57E49CF5
```

- Save all valid results

```
decryptor result:  116.10.189.246:30000:1:1:h:578856:579372:579888
```

XOR like encryption

360 INTERNET SECURITY CENTER

"New Elknot/Billgates Variant with XOR like C2 Configuration Encryption Scheme"

http://blog.netlab.360.com/new-elknot-billgates-variant-with-xor-like-c2-configuration-encryption-scheme/

```
08077A8B  lea     eax, ss:[ebp+var_18]
08077A8E  push    eax                          // std::string *
08077A8F  lea     eax, ss:[ebp+var_24]
08077A92  push    eax                          // std::string *
08077A93  call    _Z5Mid89RSsS_S_S_
08077A98  add     esp, b1 0x10
08077A9B  sub     esp, b1 0xC
08077A9E  lea     eax, ss:[ebp+var_24]
08077AA1  push    eax                          // this
08077AA2  call    _ZNKSs5c_strEv
08077AA7  add     esp, b1 0x10
08077AAA  lea     edx, ss:[ebp+strsVector]
08077AAD  sub     esp, b1 4
08077AB0  push    b1 0x3A
08077AB2  push    eax
08077AB3  push    edx
08077AB4  call    DecryptC2Cfg
08077AB9  add     esp, b1 0xC
08077ABC  sub     esp, b1 0xC
08077ABF  lea     eax, ss:[ebp+strsVector]
08077AC2  push    eax
08077AC3  call    _ZNKSt6vectorISsSaISsEE4sizeEv
08077AC8  add     esp, b1 0x10
08077ACB  cmp     eax, b1 6
08077ACE  setnz   b1 al
08077AD1  test    b1 al, b1 al
08077AD3  jz      0x8077AE0
```

XOR decryption code

2579aa65a28c32778790ec1c673abc49(MD5)

```
08077A8B  lea     eax, ss:[ebp+cipher]
08077A8E  push    eax
08077A8F  lea     eax, ss:[ebp+plain]
08077A92  push    eax
08077A93  call    RsaDecrypt
08077A98  add     esp, b1 0x10
08077A9B  sub     esp, b1 0xC
08077A9E  lea     eax, ss:[ebp+plain]
08077AA1  push    eax
08077AA2  call    _ZNKSs5c_strEv
08077AA7  add     esp, b1 0x10
08077AAA  lea     edx, ss:[ebp+strs]
08077AAD  sub     esp, b1 4
08077AB0  push    b1 0x3A
08077AB2  push    eax
08077AB3  push    edx
08077AB4  call    _ZN8CUtility5SplitEPKcc
08077AB9  add     esp, b1 0xC
08077ABC  sub     esp, b1 0xC
08077ABF  lea     eax, ss:[ebp+strs]
08077AC2  push    eax
08077AC3  call    _ZNKSt6vectorISsSaISsEE4sizeEv
08077AC8  add     esp, b1 0x10
08077ACB  cmp     eax, b1 6
08077ACE  setnz   b1 al
08077AD1  test    b1 al, b1 al
08077AD3  jz      0x8077AE0
```

RSA decryption code

8285f35183f0341b8dfe425b7348411d (MD5)

Example configuration lines

360 INTERNET SECURITY CENTER

- MD5: 8285f35183f0341b8dfe425b7348411d
  - C&C line1: ‘abu2.jack52088.com:36665:1:1:buyaocaowo:1’
  - C&C line2: ‘lzj.passwd1.com:30000:1:1:buyaocaowo:1’
- MD5: f71a34d018f804dc607ce170b9869f89
  - C&C line: ‘199.101.117.24:25000:1:1::1:698412:697896:697380’
- MD5: 4a56386b7d6061cdf70f64e366a5f62c
  - C&C line: ‘162.221.12.191:36000:1:1:h:hy:0:623424:622908:622392’
- MD5: 8d60793576180ec70032ada57d98ce00
  - C&C line: ‘204.152.199.46:36000:1:1:h:ms:598896:599412:599928’

Configuration parameters

360 INTERNET SECURITY CENTER

- There are as much as 10 parameter items

Name	Value	Position	Description
C&C server	FQDN domain or IP address	1	Always existing
C&C port	TCP port	2	Always existing
IsListener	0 or 1	3	See [2]
IsService	0 or 1	4	See [2]
CampaignName	String	Unfixed	Always existing
EnableBackdoor	0 or 1	Unfixed	Optional, see [2]
BillTail	'h'	Unfixed	Optional
RsaCrypt/RsaD/RsaN	Integer	Unfixed	Three offsets pointing to strings of RsaCrypt/RsaD/RsaN, always existing together

[2] THE ELASTIC BOTNET REPORT, https://www.novetta.com/wp-content/uploads/2015/06/NTRG_ElasticBotnetReport_06102015.pdf

- Only 5 parameter combinations were found
  - each combination can be related to one variant

Our classification scheme

360 INTERNET SECURITY CENTER

- The classification details include:
  - Parameter count
  - Whether RSA parameters are present
  - Whether the BillTail parameter exists

- Classification standards and results on 3,334 samples

Class #	Line format	Sample count
1	param_count=6, no_rsa_offsets, no_bill_tail	2,403
2	param_count=9, with_rsa_offsets, no_bill_tail	1,626
3	param_count=10, with_rsa_offsets, with_bill_tail	157
4	param_count=9, with_rsa_offsets, with_bill_tail	28
5	param_count=8, with_rsa_offsets, with_bill_tail	1

Attack methods

360 INTERNET SECURITY CENTER

Session based?	Attack type	Description
Yes	CAttackCc	HTTP flood attack
	CAttackIe	Not implemented
	CAttackTns	To attack TCP-based DNS
No	CAttackCompress	TCP packet attack
	CAttackDns	DNS flood
	CAttackAmp	DNS amplification attack
	CAttackIcmp	ICMP flood
	CAttackSyn	TCP syn flood
	CAttackUdp	UDP flood
	CAttackPrx	Similar as CAttackDns

CAttackCompress and CAttackPrx

- CAttackCompress is a kind of TCP packet attack, where different TCP flags can be instructed in the attacking packets

- CAttackPrx actually shares the same code with CAttackDns, except that it gets some parameters from a different global variable

DNS RSD attack in Elknot/BillGates

360 INTERNET SECURITY CENTER

- *"A DNS cache-busting technique for DDOS-style attacks against Authoritative Name Servers"*
  - https://blog.cloudmark.com/2014/10/07/a-dns-cache-busting-technique-for-ddos-style-attacks-against-authoritative-name-servers/

- Elknot/BillGates has its own RSD implementation

- The subdomains have the following patterns:
  - The length varies from 1 to 16
  - Each subdomain only includes characters of 'a' ~ 'z'
  - The subdomains are all initiated with characters of 'a'

Example DNS RSD attack domains

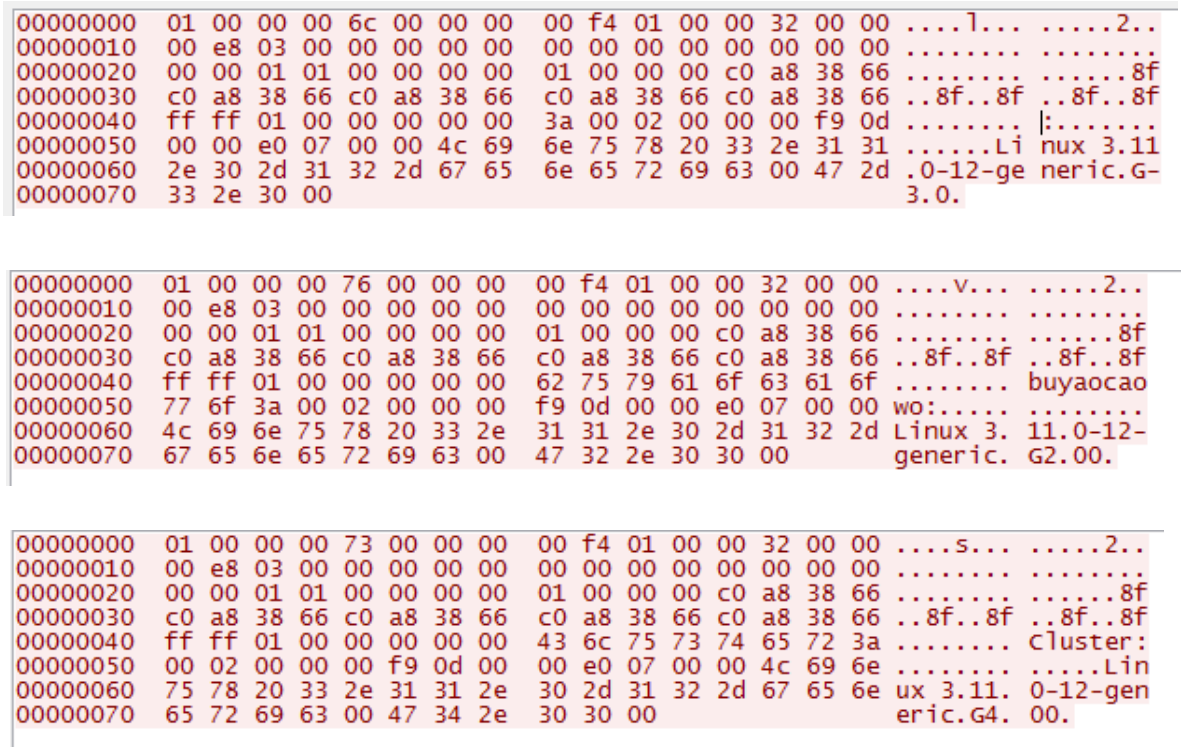
360 INTERNET SECURITY CENTER

```
ya.wap.hnpho.com
xaa.wap.hnpho.com
aaya.wap.hnpho.com
azaaa.wap.hnpho.com
kaaaaa.wap.hnpho.com
aaaapaa.wap.hnpho.com
aaaauaaa.wap.hnpho.com
abaaaaaaa.wap.hnpho.com
kaaaaaaaaa.wap.hnpho.com
aaaaaadaaaa.wap.hnpho.com
aaaaaaaaaasa.wap.hnpho.com
aoaaaaaaaaaaa.wap.hnpho.com
ahaaaaaaaaaaaa.wap.hnpho.com
alaaaaaaaaaaaaa.wap.hnpho.com
aaaaaaaaaaaaaaia.wap.hnpho.com
```

```
v.wap.hnpho.com
kx.wap.hnpho.com
lcp.wap.hnpho.com
ydah.wap.hnpho.com
lqdag.wap.hnpho.com
svaaqn.wap.hnpho.com
hnaejca.wap.hnpho.com
ulolmakx.wap.hnpho.com
abzqazgba.wap.hnpho.com
avazkaabaa.wap.hnpho.com
aaaakakyyan.wap.hnpho.com
exuamayaaaqa.wap.hnpho.com
aoaarathawaam.wap.hnpho.com
eralatoaaamaaa.wap.hnpho.com
ueaaaaaakaaamev.wap.hnpho.com
gvaaaaaaaaajazsa.wap.hnpho.com
```

```
j.wap.hnpho.com
in.wap.hnpho.com
rzu.wap.hnpho.com
qpyt.wap.hnpho.com
wewmq.wap.hnpho.com
edozwn.wap.hnpho.com
wombyib.wap.hnpho.com
sxihgzin.wap.hnpho.com
pqevbskdv.wap.hnpho.com
shidwlazed.wap.hnpho.com
tepurrlcauc.wap.hnpho.com
obkhwdonshsn.wap.hnpho.com
nopqrsthvwklz.wap.hnpho.com
ojkjktmdgfkbip.wap.hnpho.com
kqxfkxtlnskvkhv.wap.hnpho.com
ivkbulwtupohelgh.wap.hnpho.com
```

The C2 protocol: REGISTER



```
00000000  01 00 00 00 6c 00 00 00  00 f4 01 00 00 32 00 00 ....l... .....2..
00000010  00 e8 03 00 00 00 00 00  00 00 00 00 00 00 00 00 ........ ........
00000020  00 00 01 01 00 00 00 00  01 00 00 00 c0 a8 38 66 ........ ......8f
00000030  c0 a8 38 66 c0 a8 38 66  c0 a8 38 66 c0 a8 38 66 ..8f..8f ..8f..8f
00000040  ff ff 01 00 00 00 00 00  3a 00 02 00 00 00 f9 0d ........ :.......
00000050  00 00 e0 07 00 00 4c 69  6e 75 78 20 33 2e 31 31 ......Li nux 3.11
00000060  2e 30 2d 31 32 2d 67 65  6e 65 72 69 63 00 47 2d .0-12-ge neric.G-
00000070  33 2e 30 00                                      3.0.
```

```
00000000  01 00 00 00 76 00 00 00  00 f4 01 00 00 32 00 00 ....v... .....2..
00000010  00 e8 03 00 00 00 00 00  00 00 00 00 00 00 00 00 ........ ........
00000020  00 00 01 01 00 00 00 00  01 00 00 00 c0 a8 38 66 ........ ......8f
00000030  c0 a8 38 66 c0 a8 38 66  c0 a8 38 66 c0 a8 38 66 ..8f..8f ..8f..8f
00000040  ff ff 01 00 00 00 00 00  62 75 79 61 6f 63 61 6f ........ buyaocao
00000050  77 6f 3a 00 02 00 00 00  f9 0d 00 00 e0 07 00 00 wo:..... ........
00000060  4c 69 6e 75 78 20 33 2e  31 31 2e 30 2d 31 32 2d Linux 3. 11.0-12-
00000070  67 65 6e 65 72 69 63 00  47 32 2e 30 30 00       generic. G2.00.
```

```
00000000  01 00 00 00 73 00 00 00  00 f4 01 00 00 32 00 00 ....s... .....2..
00000010  00 e8 03 00 00 00 00 00  00 00 00 00 00 00 00 00 ........ ........
00000020  00 00 01 01 00 00 00 00  01 00 00 00 c0 a8 38 66 ........ ......8f
00000030  c0 a8 38 66 c0 a8 38 66  c0 a8 38 66 c0 a8 38 66 ..8f..8f ..8f..8f
00000040  ff ff 01 00 00 00 00 00  43 6c 75 73 74 65 72 3a ........ Cluster:
00000050  00 02 00 00 00 f9 0d 00  00 e0 07 00 00 4c 69 6e ........ .....Lin
00000060  75 78 20 33 2e 31 31 2e  30 2d 31 32 2d 67 65 6e ux 3.11. 0-12-gen
00000070  65 72 69 63 00 47 34 2e  30 30 00                eric.G4. 00.
```

```
struct REGISTER {
    msg_hdr hdr;
    u8 conf[0x40];
    std::string campaign;
    u32 cpu_num;
    u32 cpu_spd;
    u32 mem_size;
    std::string os;
    std::string magic;
};
```

Example attack command


```
0000001C  01 00 00 00 83 00 00 00  00 f4 01 00 00 32 00 00   ........ .....2..
0000002C  00 e8 03 00 00 3a 24 00  00 00 00 00 00 01 00 00   .....:$. ........
0000003C  00 01 00 00 00 21 02 00  d0 07 00 00 00 00 01 00   .....!.. ........
0000004C  00 00 20 00 00 36 00 00  00 36 00 00 00 04 00 00   .. ..6.. .6......
0000005C  00 2c 01 00 00 68 6b 2e  64 76 2e 6e 65 78 74 6d   .,...hk. dv.nextm
0000006C  65 64 69 61 2e 63 6f 6d  00 00 03 00 00 00 31 39   edia.com ......19
0000007C  38 2e 34 31 2e 32 32 32  2e 35 00 35 00 31 39 38   8.41.222 .5.5.198
0000008C  2e 34 31 2e 32 32 32 2e  36 00 35 00 31 39 38 2e   .41.222. 6.5.198.
0000009C  34 31 2e 32 32 33 2e 36  00 35 00                  41.223.6 .5.
```

target1=*198.41.222.5_53*

target2=*198.41.222.6_53*

target3=*198.41.223.5_53*

attack_type=*dns_flood*

domain=*hd.dv.nextmedia.com*

Command code

360 INTERNET SECURITY CENTER

Code	Description
1	StartAttack
2	StopAttack
3	Configure
5	UpdateModule
9	ExecuteShellCommand

Infection vector

360 INTERNET SECURITY CENTER

Vector	Unique URLs	Unique samples
ssh (22)	1484	1333
MySQL (3306)	98	94
Elasticsearch (9200)	73	64

**The statistics is done on our honeypot data from Jan. to Sep. 2016*

Our command tracking system

360 INTERNET SECURITY CENTER

- DDoS bots are classified based on their C2 protocols
  - ~40 common DDoS families are being tracked
    - Elknot/BillGates, XOR.DDoS, Mr.Black, Gafgyt, Nitol, etc.
- C2's are extracted from samples
- Received commands are parsed and saved into databases for later analysis
  - ~600M commands have been received

A summary of tracking data

360 INTERNET SECURITY CENTER

- 4,200 collected samples
- 1,885 extracted C2 controllers with 858 used to be active
- 40,590,314 attack commands were received from 498 C2 controllers
- 57,102 unique victims were checked

** The statistics is done on our track data till May 31, 2016*

Stats on the 1,885 C2's

360 INTERNET SECURITY CENTER

IP vs FQDN

Format	Sample count
IP	1085
FQDN	800

Top countries of active C2 IPs

Country/Region	Total
China	500
USA	120
Hong Kong	39
Canada	11
Korea	7
Taiwan	3
India	2
Japan	1
Thailand	1

Stats on attack types

360 INTERNET SECURITY CENTER

Attack type	Count
CAttackCompress	32,545,578
CAttackDns	4,384,967
CAttackTns	79,820
CAttackPrx	4,409
CAttackUdp	841
CAttackAmp	32
CAttackIcmp	28
CAttackTcp	13
CAttackIe	8
CAttackCc	1

- CAttackCompress accounts for ~80% of the received commands
  - Over 90% of them belonged to the Tsunami Attack [1]
- DNS flood was another favorite attack method by Elknot/BillGates attackers
  - CAttackDns/CAttackTns

[1] *Researchers observe new type of SYN flood DDoS attack*, http://www.scmagazine.com/researchers-observe-new-type-of-syn-fl ood-ddosattack/article/376576/

Top countries of 57,102 victims

360 INTERNET SECURITY CENTER

Country/Region	Count
China	40545
USA	11451
Hong Kong	2200
Taiwan	635
Japan	544
Korea,	509
Canada	427
Singapore	285
France	185
Netherlands	125

Top ASNs of 57,102 victims

360 INTERNET SECURITY CENTER

ASN	ASN Name	Country	Count
AS37963	Hangzhou Alibaba	CN	11,844
AS4134	Chinanet	CN	8,466
AS58543	Guangdong	CN	5,055
AS4837	CNCGROUP China169 Backbone	CN	4,974
AS13335	CloudFlare, Inc.	US	2,365
AS26484	HOSTSPACE NETWORKS LLC	US	1,971
AS23650	CHINANET jiangsu backbone	CN	1,799
AS133774	Fuzhou	CN	1,542
AS133775	Xiamen	CN	1,182
AS17816	China Unicom IP network China169	CN	952

Other interesting findings

360 INTERNET SECURITY CENTER

- The same command was usually repeatedly distributed
  - ***30s*** was the most commonly seen interval value, which makes it possible to detect Elknot/BillGates C2 communication from Netflow data
- It's common that multiple C2 controllers jointly attacked the same victim(s)
  - even together with other DDoS botnet families

Attacks to root name servers

360 INTERNET SECURITY CENTER

- DNS root name servers were attacked on 30 Nov and 1 Dec, 2015
  - http://www.root-servers.org/news/events-of-20151130.txt
- Details of the attack on November 30:
  - 12 root name servers were attacked by 5 Elknot/BillGates C2 controllers
  - The attack lasted ~2.5 hours
- Details of the attack on December 1:
  - 12 root name servers were attacked were 3 C2 controllers
  - The attack lasted ~1 hour

Speculations on the motives

360 INTERNET SECURITY CENTER

- The real target was *916yy.com*, because:
  - On November 20, 2015, a China DNS service provider was DNS flooded by the same Elknot/BillGates C2's
  - On December 2, 2015, the same attack was again observed
  - From December 2015 to January 2016 similar attacks were observed multiple times
  - *916yy.com* was repeatedly used in the above attacks

- We think the Elknot/BillGates attackers were just to have a test to see whether better effects could be obtained by attacking the root name servers

A mysterious DNS attack tool

360 INTERNET SECURITY CENTER

- It also supports DNS RSD attack and shares the same subdomain patterns with Elknot/BillGates
  - but differs in packet fingerprint
- It's observed many times being used together with some Elknot/BillGates botnets in the past years
  - Including the attacks to root name servers
- Since we have not seen its sample, we have no idea whether it's a botnet family or packet generation tool
  - We monitor its activity with honeypots

Q&A