Botnet

Emptiness: A New Evolving Botnet

Hui Wang, Alex.Turing

Aug 9, 2019 • 5 min read

Background

Our honeypot system captured a new DDoS botnet sample on 2019-06-23. We named it Emptiness which comes from the running process name as well as its C2 domain. Emptiness is written by Golang and supports both Windows and Linux. Our further analysis reveal its iterative evolution: the early version Emptiness was only armed with DDoS function and was propagated through Mirai.shiina variant botnet that had been run by the same author for a long time. The latter version Emptiness was added with ssh scan to enable self-propagation. Besides, C2 communication protocols of both Emptiness and Mirai.shiina kept changing constantly, which increases the difficulty of being tracked by security researchers.

The Timeline of Emptiness

- 2019-06-09: we first observed Emptiness download URL,but was not able to download the bot sample at that time.

Time	sip	dport	uri	ptype
June 9th 2019, 12:51:28.059	185.13.38.224	56,160	http://blogentry.hopto.org/emptiness	udp
June 9th 2019, 12:51:28.290	185.13.38.224	56,160	http://blogentry.hopto.org/emptiness	udp
June 9th 2019, 12:51:34.677	185.13.38.224	59,131	http://blogentry.hopto.org/emptiness	udp
June 9th 2019, 12:51:34.858	185.13.38.224	59,131	http://blogentry.hopto.org/emptiness	udp
June 10th 2019, 12:38:39.887	185.13.38.224	45,836	http://blogentry.hopto.org/emptiness	udp
June 10th 2019, 12:38:40.247	185.13.38.224	49,390	http://blogentry.hopto.org/emptiness	udp

- 2019-06-23: we captured bot sample Emptiness.V1

- 2019-06-26: we captured bot sample Emptiness.V1.1
- 2019-07-03: we exposed DDoS attacks launched by Emptiness.v1.1
- 2019-07-06: we captured bot sample Emptiness.v2, which said hello to us (360Netlab) "politely".

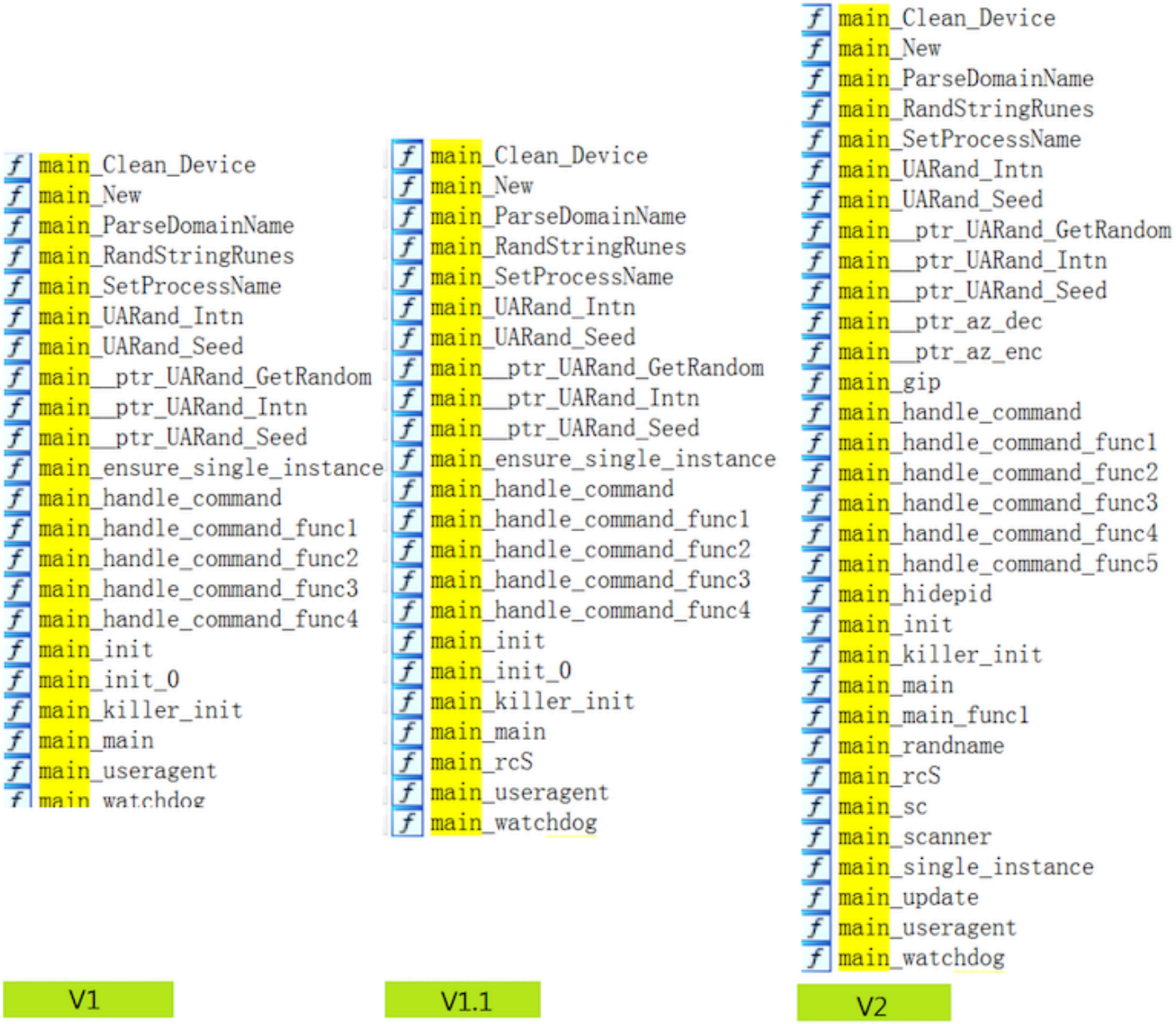
```
mov     rcx, cs:main_fuck_u_netlab360
```

Emptiness Sample Analysis

The following table summarizes the main differences between three versions of Emptiness.

Version	Sample hash	CC	Encryption/ Encode	Command
v1	f41464471a0ac9c165e4aeb55283934e	bruhitsnot.cf:8080	No	.cc .dns .udp .http .stop
v1.1	420bb6147ca091a22f8f5bbbb49d51f3	35.229.244.105:8080	Base64	.cc .dns .udp .http .stop
v2	7b1943ff6c563ce1043963e2f017ad8d	ggwp.emptiness.tk:23336 version2.ilove26.cf:23336 luckyhere.mashiro.tk:23336 imtesting.shiina.ga:23336	Xor/Base64	.cc .dns .udp .http .stop .exec .show .kill .update

All these three versions are written by Golang and are protected by standard UPX. As shown in the following figure, their main function logics are mainly implemented in main package and can be categorized into three aspects: host behavior, data encoding/encryption and C2 protocol. Next we will introduce Emptiness from these three aspects.



Host Behavior

Most host behaviors performed by Emptiness are common operations, such as ensuring singleton, modifying process name, turn off watchdog, clear history logs, delete system command, stop system service and so on. The only thing worth to note is that Emptiness will kill process using specific ports through its main_killer_init function. In this way, it can remove bot competitors and close vulnerable ports and thus occupy victim machine exclusively. The following figures show its methods of killing process and the ports it intends to kill.

- Methods of killing process
  - V1 & V1.1

    ```
    fuser -k -n tcp
    ```

  - V2

```
1:      fuser -k /tcp
2:      lsof -i tcp:%s | grep LISTEN | awk '{print $2}' | xargs kill
```

- Ports it intends to kill:

```
"22"
"23"
"80"
"420"
"1337"
"1991"
"5332"
"6553"
"6554"
"6666"
"6667"
"6697"
"18904"
"37215"
"42026"
"48101"
"52869"
"57643"
```

Data Encoding and Encryption

We demonstrate how Emptiness encodes or encrypts data in different versions using its register message.

- Emptiness.V1 communicate through plaintext and no encoding or encryption is involved.
- Emptiness.V1.1 encodes the message with Base64
- Emptiness.V2 first encodes the message with Base64, then encrypts it with XOR transformation and the XOR key is “B2BB01039307BAA2"
  -

C2 Protocol

Register Message

- Emptiness.v1

```
00000000  69 6c 6f 76 65 32 36                              |ilove26|
```

- Emptiness.v1.1

```
00000000  61 57 78 76 64 6d 55 79  4e 67 3d 3d              |aWxvdmUyNg==|
Base64 encode, plaintext is: ilove26
```

- Emptiness.v2

```
00000000  64 33 65 63 37 39 37 35  66 37 36 61 65 66 64 62  |d3ec7975f76aefd
00000010  66 63 64 63 33 63 33 65                           |fcdc3c3e|
Base64 encode and then XOR transformation, plaintext is: ilove26
```

Supported Command

- Emptiness.V1 and Emptiness.V1.1

```
.cc
.udp
.dns
.http
.stop
```

- Emptiness.V2

```
.cc
.dns
.udp
.http
.stop
.exec
.slow
.kill
.update
```

Examples of DDoS Attack Command

- UDP attack

```
LnVkcCAxMi4xNjIuMjIwLjEwNiA4MCAyNTYgNTUgNzIw
Base64 decode
00000000  2e 75 64 70 20 31 32 2e  31 36 32 2e 32 32 30 2e  |.udp 12.162.220.|
00000010  31 30 36 20 38 30 20 32  35 36 20 35 35 20 37 32  |106 80 256 55 72|
00000020  30                                                |0|
```

- HTTP attack

```
Lmh0dHAgd3d3LmxscnJ5LmNuIDgwIC8gNjAgNjA=
Base64 decode
00000000  2e 68 74 74 70 20 77 77  77 2e 6c 6c 72 72 79 2e  |.http www.llrry.|
00000010  63 6e 20 38 30 20 2f 20  36 30 20 36 30           |cn 80 / 60 60|
```

The Relation between Emptiness and Mirai.shiina

Two of the four Emptiness.V2 C2 servers(`luckyhere.mashiro.tk` and `imtesting.shiina.ga`) were used by mirai variant. Mirai.shiina and the corresponding sample is `f6e9f3567684a0a7402ad97209b8525b` . We name it Mirai.shiina based on its infection command `/bin/busybox SHIINA` and its sample file name `shine.{arch}` . Sharing the same C2 server strongly indicates that Emptiness and Mirai.shiina belong to the same author.

More Info about Mirai.shiina

Our monitor system shows that Mirai.shiina botnet was running at least from 2019-03 to 2019-07 and during this period it was very active and deliver attack command frequently. The figure below shows some example of its attack commands.

time	botname	cc_server	cc_ip	cc_port	type	atk_type	target_host	target_port
2019-03-18 18:56:55+08:00	mirai	34.80.131.135	34.80.131.135	1337	ddos	atk_10	3.95. [redacted] 9	80
2019-03-19 17:06:28+08:00	mirai	34.80.131.135	34.80.131.135	443	ddos	atk_10	103.1 [redacted] .48	80
2019-04-11 18:41:11+08:00	mirai	35.235.102.123	35.235.102.123	1337	ddos	atk_5	35.19 [redacted] .167	80
2019-04-26 22:25:09+08:00	mirai	ililililililllii.sytes.net	35.235.102.123	1337	ddos	atk_9	138.6 [redacted] 157	80
2019-04-28 16:50:16+08:00	mirai	35.235.102.123	35.235.102.123	1337	ddos	atk_9	190.1 [redacted] .202	80
2019-04-28 16:50:23+08:00	mirai	ililililililllii.sytes.net	35.235.102.123	1337	ddos	atk_9	190.1 [redacted] .202	80
2019-05-02 17:59:47+08:00	mirai	fuckcia.hopto.org	35.201.141.13	1337	ddos	atk_0	115.2 [redacted] 4.22	80
2019-05-05 23:23:36+08:00	mirai	35.235.102.123	35.235.102.123	1337	ddos	atk_7	64.32 [redacted] 6	NULL
2019-05-13 19:05:08+08:00	mirai	ilililililililil.hopto.org	35.235.102.123	1337	ddos	atk_3	221.1 [redacted] 7.133	443
2019-05-28 13:16:45+08:00	mirai	asdfghjklzxcvbnm.zapto.org	35.224.155.10	8080	ddos	atk_3	103.1 [redacted] .48	NULL
2019-06-14 20:12:05+08:00	mirai	dfghjkllkjhgf.ml	35.226.164.220	6666	ddos	atk_6	74.91 [redacted] 103	80

Besides, we can divide Mirai.shiina into four versions according to the difference of its C2 protocol, see the following table and we infer that the author kept

modifying C2 protocol continuously to prevent from being tracked.

Version	Sample hash	CC	Register message	Sample captured time
v0	77e7dd8982e7bb21d536264f0635d5cb	34.80.131.135:1337	\x00\x00\x00\x01	2019-03-18 17:00:18+08:00
v1	209a78969d88c667c32e550ce47b8ff9	shiina.mashiro.tk:17	\x01\x03\x03\x07	2019-06-20 23:46:21+08:00
v2	0cf288e07e888cd7748b30fa4a67ca84	shiina.mashiro.tk:19	\x04\x02\x00\x01	2019-06-22 00:15:43+08:00
v3	f6e9f3567684a0a7402ad97209b8525b	luckyhere.mashiro.tk:13372 imtesting.shiina.ga:13372	\x01\x03\x03\x07\x04\x02\x00\x06	2019-07-25 23:56:15+08:00

Question: Why the author wants to build the new Emptiness Botnet?

According to the above analysis, we know that the author of Emptiness has also run Mirai.shiina botnet for a long time, which is active to launch DDoS attacks and seems to own quite a few bots. So why the author still wants to build the new Emptiness botnet? We guess the reasons behind are:

- To support cross platform. Emptiness is written by Golang, which can be easily migrated across different platform. Golang currently supports more than 40 platforms, including `android/arm`, `Darwin/arm`, `linux/arm`, `windows/arm`, etc.
- To evade detection. Today Mirai is commonly known and can be detected by most of anti-virus engines. Developing a completely new botnet can greatly decrease the risk of being detected. As an example, we submit one Emptiness sample to VT and found that only one anti-virus vendor identified it as malware.

Contact us

Readers are always welcomed to reach us on twitter, WeChat 360Netlab or email to netlab at 360 dot cn.

IoC

Emptiness CC

```
emp.web2tor.cf
bruhitsnot.tk
bruhitsnot.cf
emptiness.web2tor.cf
version2.ilove26.cf
luckyhere.mashiro.tk
imtesting.shiina.ga
ggwp.emptiness.tk
```

Emptiness Sample MD5

```
f41464471a0ac9c165e4aeb55283934e
420bb6147ca091a22f8f5bbbb49d51f3
7b1943ff6c563ce1043963e2f017ad8d
53bb43411ecbad39b18b0662b53c07a0
1899667e48c64b113c0de54cf3bb63d5
```

Mirai.shiina CC

```
34.80.131.135
shiina.mashiro.tk
luckyhere.mashiro.tk
imtesting.shiina.ga
```

Mirai.shiina Sample MD5

```
77e7dd8982e7bb21d536264f0635d5cb
209a78969d88c667c32e550ce47b8ff9
```

```
0cf288e07e888cd7748b30fa4a67ca84
f6e9f3567684a0a7402ad97209b8525b
```


— 360 Netlab Blog - Network Security Research Lab at 360 —

Botnet

僵尸网络911 S5的数字遗产

Heads up! Xdr33, A Variant Of CIA's HIVE Attack Kit Emerges

DDoS

那些和185.244.25.0/24网段有关的Botnet

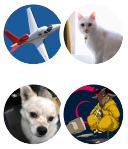
Botnet

那些总是想要和别人强行发生关系的僵尸网络之Emptiness

警惕：魔改后的CIA攻击套件Hive进入黑灰产领域

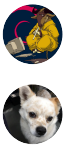
See all 114 posts →

根据我们的观察，过去几年185.244.25.0/24这个网段出现了超多的Botnet，包括但不仅限于mirai、gafgyt、tsunami、fbot、moobot、handymanny等，他们属于同一个组织或共享了相关代码。下表是过去一年我们关于该网段的一些统计数据。可以看出该网段有很多的CC和攻击行为。 Count of CC (host:port) Count of attack target host Count of...



Sep 27, 2019 • 9 min read

背景 2019年06月23日我们捕获了一个全新的DDoS僵尸网络样本，因其启动时设置的进程名以及C2中有emptiness字样，所以我们将其命名为Emptiness。Emptiness由golang编写，当前发现的样本包括Windows和Linux两种平台版本。在溯源过程中，我们发现其作者长期维护着一个mirai变种僵尸网络，早期的Emptiness自身没有传播能力...



Aug 9, 2019 • 7 min read