Botnet

那些总是想要和别人强行发生关系的僵尸网络之Emptiness

Hui Wang, Alex.Turing

Aug 9, 2019 • 7 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自身没有传播能力只有DDoS功能，是由mirai loader来完成样本植入的，后期的版本增加了ssh扫描功能，可独立完成Emptiness自身样本的传播。我们还注意到其不断修改mirai变种和Emptiness的CC协议，也许是为了对抗安全研究人员跟踪其僵尸网络攻击行为。

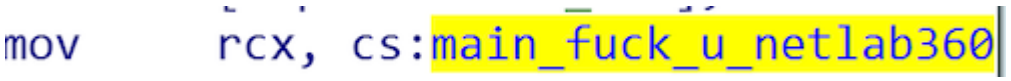
Emptiness的那些事

- 我们猜测其最早出现时间应该是2019年06月09日，遗憾的是当时我们并没有成功下载到 `http://blogentry.hopto.org/emptiness` 相关的样本。下图是我们捕获这个URL的时间，

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日我们首次捕获到该僵尸网络样本v1版本
- 2019-06-26日我们首次捕获到该僵尸网络样本v1.1版本
- 2019-07-03日我们曝光了其DDoS攻击行为

- 2019-07-06作者更新v2版本样本，并在样本中留下了一匹"羊驼"。表示想要和我们强行发生关系。

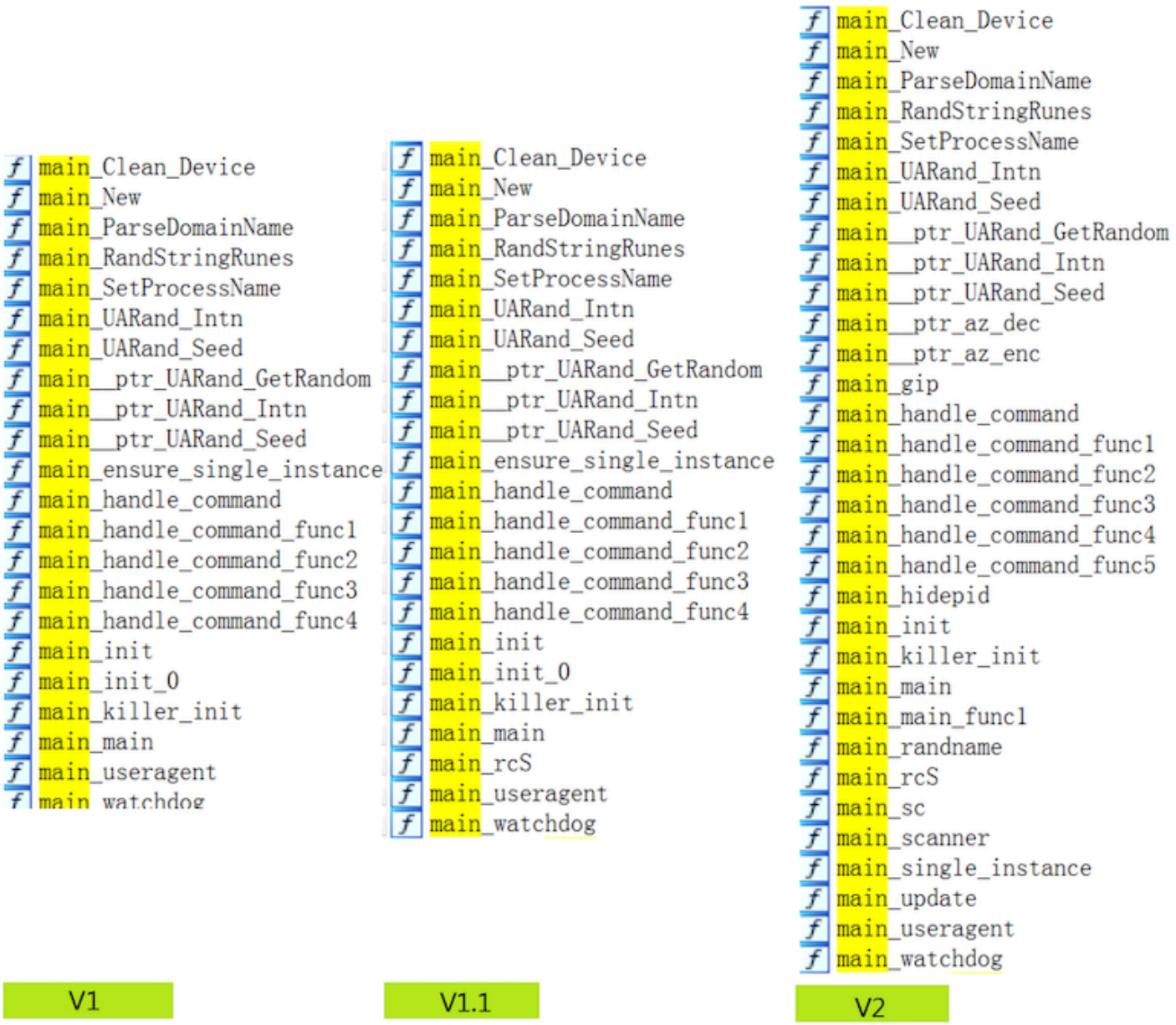


Emptiness样本分析

通过上线包，支持的指令，资源加密等特征的变化，可以得知其作者正在积极的进行迭代开发，我们将捕获到的样本分成 3 个版本，以下是它们的详细对比图:

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

3 个版本都使用了标准UPX壳，使用upx -d 即可脱壳。脱壳后知道这个家族是用Go语言的编写的，主要逻辑在main包里实现，函数对比如下图所示，根据函数的功能，下文将从**主机行为**，**数据编码&加密**，**CC流量**这 3 个维度来剖析Emptiness。



各版本函数对比图

主机行为

Emptiness在主机行为层面，并无太多特色，都是些常见操作，比如确保单一实例，修改进程名，关闭watchdog，清理history记录，删除系统命令，关闭系统服务之类。唯一值得一提的是会通过**main_killer_init**函数结束占用特定端口的进程。

- 目的：杀死竞争对手bot，关闭常见可感染端口，达到独占受害者机器的目的
- 杀死进程实现方法
  - V1 & V1.1

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

- V2

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

- 杀死占用以下端口的进程：

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

数据编码&加密

我们将以上线包为例来说明Emptiness不同版本所使用的数据编码&加密方法。

数据编码&加密代码对比

- V1版本无特殊编码/加密资源
- V1.1版本Base64编码通信包
- V2版本使用XOR加密，Base64编码通信包，CC等敏感资源。
  -
  - XOR加密算法（golang code）

```
func (empt *Emptiness) Xor(data []byte) []byte {

    key := "B2BB01039307BAA2"
    xorKey, _ := hex.DecodeString(key)
    keyLen := len(xorKey)

    for idx, item := range data {
        data[idx] = item ^ xorKey[idx%keyLen]
    }

    return data
}
```

CC流量

指令

- V1&V1.1版本支持的指令

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

- V2版本支持的指令

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

上线包

- V1版本

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

- V1.1版本

```
00000000  61 57 78 76 64 6d 55 79  4e 67 3d 3d              |aWxvdmUyNg==|
Base64编码，解码后为ilove26
```

- V2版本

```
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|
XOR加密，解密后为aWxvdmUyNg==
Base64编码，解码后为ilove26
```

DDoS攻击包示例

- udp攻击

```
LnVkcCAxMi4xNjIuMjIwLjEwNiA4MCAyNTYgNTUgNzIw
Base64编码，解码后为
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攻击

```
Lmh0dHAgd3d3LmxscnJ5LmNuIDgwIC8gNjAgNjA=
Base64编码，解码后为
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|
```

Emptiness与mirai.shiina的关系

我们观察到Emptiness v2使用的4个CC其中2个（`luckyhere.mashiro.tk`和`imtesting.shiina.ga`）曾经被另外一个mirai变种作为CC被使用过，相关mirai样本`f6e9f3567684a0a7402ad97209b8525b`。根据其可感染环境验证命令`/bin/busybox SHIINA`以及常用样本文件名称`shiina.{arch}`等我们将该mirai变种命名为mirai.shiina。通过共用相同CC这一点，直接证明Emptiness和mirai.shiina属于同一个作者。

关于mirai.shiina

从我们的数据看该僵尸网络的作者至少从2019年03月到2019年07月一直运营着这个僵尸网络。仅从CC协议差异的角度，大致可将mirai.shiina分为4个版本，各版本部分样本信息如下：

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

可以看出作者一直在持续更新其样本，后期不断修改CC协议，也许是为了对抗安全研究人员跟踪其僵尸网络？另外从我们跟踪结果看，该僵尸网络的攻击指令从未间断，一直处于活跃状态。下图是一些攻击指令示例：

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	ilililililililii.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	ilililililililii.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	ilililililililili.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

mirai.shiina DDoS 攻击指令

我们好奇作者为什么要新建一个emptiness僵尸网络

从前面的分析可以看出，作者长期在维护这个mirai.shiina僵尸网络，从不断有攻击指令看估计也控制了不少肉鸡。既然已经有一个发起可以攻击僵尸网络，为什么还要建一个新的呢？真实原因不得而知。我们猜测也许是为了：

- 跨平台？Emptiness由golang编写。golang天生强大的可移植性，用golang编写代码可轻松实现跨平台。当前golang已支持的平台包括 `android/arm`，`darwin/arm`，`linux/arm`，`windows/arm`等40多种。
- 逃避杀毒引擎检测？我们都知道mirai烂大街，大部分杀毒引擎都能检测出来。作者构建一个全新的出来了也许是为了降低杀毒引擎检出率。我们将其中一个样本投递到VT，只有一家杀毒引擎检测为恶意软件，其他厂家均未检出。

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
shiina.mashiro.tk
```

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

Mirai.shiina Sample MD5

```
77e7dd8982e7bb21d536264f0635d5cb
209a78969d88c667c32e550ce47b8ff9
0cf288e07e888cd7748b30fa4a67ca84
f6e9f3567684a0a7402ad97209b8525b
```


Botnet

Emptiness: A New Evolving Botnet

Botnet

Some Fiberhome routers are being utilized as SSH

僵尸网络911 S5的数字遗产

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

警惕：魔改后的CIA攻击套件Hive进入黑灰产领域

See all 114 posts →

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

Aug 9, 2019 · 5 min read

tunneling proxy nodes

Background introduction On July 24, 2019, our Unknown Threat Detection System highlighted a suspicious ELF file with 0 VirusTotal detection. When we further looked into it, we realized it is a component of an IoT botnet targeting Fiberhome router. But it does not do the regular stuff such as DDos,

Aug 2, 2019 · 5 min read