# 0x01 MHN的安装

因为ThreatStream有部署脚本，所以安装MHN很简单，安装步骤如下：

yum update

yum install –y git sqlite-devel

cd /opt

sudo git clone https://github.com/threatstream/mhn

安装之前的处理事项：安装脚本的内容要根据机器环境以及要安装的组件文档做更新。

主要是 install\_mongo.sh要改动一下，根据mongo 官方文档安装最新稳定版。

Yum update 的时候排除了bash的更新，因为机器用的定制bash。

安装脚本install.sh 就是调用一系列的安装脚本实现安装。

cd mhn

./install.sh

# 0x02 MHN的排错

碰到的一些报错问题及解决方案：

报错

File

You must not use 8-bit bytestrings unless you use a text\_factory that can interpret 8-bit bytestrings (like text\_factory = str).

It is highly recommended that you instead just switch your application to Unicode strings.

[SQL: u'INSERT INTO rule\_references (text, rule\_id) VALUES (?, ?)']

[parameters: ('url,blogs.forcepoint.com/security-labs/trojanized-adobe-installer-used-install-dragonok\xe2\x80\x99s-new-custom-backdoor', 6673)]

解决

编辑 /mhn/server/mhn/api/models.py 文件，在开头加入:

import sys

reload(sys)

sys.setdefaultencoding('utf-8')

将所有的 ref.text = r 替换成 ref.text = r.decode('utf-8')

保存文件。删除产生的 mhn/server/mhn.db文件并重新安装。

重新安装要注释掉安装脚本中成功的部分代码再运行脚本，可以避免冲突，节省时间。

报错

src/github.com/jt6211/sockjs-go/sockjs/rawwebsocket.go:4:2: cannot find package "golang.org/x/net/websocket" in any of:

/usr/lib/golang/src/golang.org/x/net/websocket (from $GOROOT)

/opt/honeymap/server/src/golang.org/x/net/websocket (from $GOPATH)

解决

cd /opt/honeymap/server   
export GOPATH=/opt/honeymap/server   
go get github.com/golang/net

mkdir -p golang.org/x   
cp -rf src/github.com/golang/net/ ./golang.org/x/   
cp -rf golang.org/ /usr/local/go/src/   
go build   
sudo supervisorctl restart all

报错

提示python sqlite 找不到依赖，或者存库时有编码问题

解决

找到系统python版本，下载其源码包，重新编译。将编译得到的\_sqlite3.so文件复制到系统python包lib目录下。

cp \_sqlite3.so /usr/local/lib/python2.7/sqlite3/

cp \_sqlite3.so /usr/local/lib/python2.7/lib-dynload/

相关组件在安装包中都应该有，安装正确的话会自动安装到相关 lib 目录。

安装完成后执行sudo supervisorctl status看到服务运行状态并排错。

如果发现服务不正常，你可以使用一些命令和排查一些日志来来判断问题出在哪里，第一个命令就是supervisorctl，可以看到那些进程出问题了，那些在正常的运行

#!bash

supervisorctl status

#列出所有进程的状态

supervisorctl　restart [process|all]

#重启单个或者全部进程

supervisorctl start [process|all]

#开始单个进程或者全部进程

supervisorctl stop [process|all]

#停止单个进程或者全部进程

如果你发现一个进程的状态为ERROR或者FATAL，就需在 /etc/supervisor/conf.d/找到对应进程的配置文件，查看日志进行分析

更新配置文件

supervisorctl update

# 0x03 MHN的配置事项

机器位于公网，就可以跳过这一步，如果只是放在内网里面，则需要配置mnemosyne的配置文件

sudo vim /opt/mnemosyne/mnemosyne.cfg

ignore\_rfc1918 = False

允许节点使用私有地址和服务器进行通信

重启服务

sudo supervisorctl restart Mnemosyne

如果想修改smtp服务的配置，可以编辑config.py，

绝对路径

#!bash

/opt/mhn/server/config.py

cd /opt/mhn/server

sudo vim config.py

sudo supervisorctl restart mhn-uwsgi

更改用户的密码

#!bash

sudo su

cd /opt/mhn/

source env/bin/activate

cd server

python manual\_password\_reset.py

deactivate

exit

例如要非匿名的将收集到得攻击数据回传到ThreatStream，要做以下操作

#!bash

cd /opt/mhn/scripts

sudo ./disable\_collector.sh

执行enable\_collector.sh可以开启

Map 页面就是显示的地图，这里一开始点击会显示说 404，需要在配置文件把页面地址更改一下 > vim /opt/mhn/server/config.py

把 > HONEYMAP\_URL = ‘:3000’

改成

HONEYMAP\_URL = ‘[http://ip:3000'](http://ip:3000'/)

ip 即为该服务器 ip

# 0x04 MHN蜜罐安装

一旦安装了基础的服务，就能够部署蜜罐节点了，通过web来展示相关数据等等，可以根据具体的境况做一些简单的调整。

Deploy 页面即保存部署蜜罐的脚本，可以下拉选择不同的蜜罐，也可以自定义脚本。它提供的脚本不一定完全适用，有些情况需要自己更改一下。复制上面的 Deploy Command 到另外的机器上执行，完成后一般就会在 Sensors 显示新安装的蜜罐节点。

安装蜜罐节点很容易，所有的节点都基于同样的平台，MHN上对每个蜜罐都有对应的安装脚本，所以安装起来是非常容易的，只需要一个服务器去安装。

定义的安装脚本在MHN服务器的web界面中，在"Deploy"这个选项下有所有蜜罐的安装脚本。

http://192.168.5.11/ui/manage-deploy/?script\_id=11

注意根据实际情况更新安装脚本，点击update按钮保存。

使用页面中command 给的安装命令，注意端口是否正确。

支持的蜜罐及介绍

• Sort: https://www.snort.org/

• Suricata: http://suricata-ids.org/

• Dionaea: http://dionaea.carnivore.it/, 它是一个低交互式的蜜罐，能够模拟 MSSQL, SIP, HTTP, FTP, TFTP 等服务 drops 中有一篇介绍： http://drops.wooyun.org/papers/4584

• Conpot: http://conpot.org/

• Kippo: https://github.com/desaster/kippo，它是一个中等交互的蜜罐，能够下载任意文件。 drops 中有一篇介绍： http://drops.wooyun.org/papers/4578

• Amun: http://amunhoney.sourceforge.net/，它是一个低交互式蜜罐，但是已经从 2012 年之后不在维护了。

• Glastopf： http://glastopf.org/

• Wordpot： https://github.com/gbrindisi/wordpot

• ShockPot： https://github.com/threatstream/shockpot，模拟的 CVE-2014-6271，即破壳漏洞

• p0f： https://github.com/p0f/p0f

其他web蜜罐

https://github.com/mushorg/glastopf 8月前

https://github.com/dustyfresh/HoneyPress 7个月

https://github.com/mushorg/tanner 最近

https://shadowd.zecure.org/overview/introduction/

推荐蜜罐组合

Dionaea + kippo + snort + p0f

Amun + kippo + snort + p0f

Elastichoney + shockpot + snort + p0f

Conpot + snort + p0f

Elastichoney + Glastopf + snort + p0f

p0f和snort(suricata)结合部署在每天机器上可以提供实用的威胁情报，包括攻击者的操作系统，网络连接类型，payload类型等。

Deploy 页面即保存部署蜜罐的脚本，可以下拉选择不同的蜜罐，也可以自定义脚本。它提供的脚本不一定完全适用，有些情况需要自己更改一下。复制上面的 Deploy Command 到另外的机器上执行，完成后一般就会在 Sensors 显示新安装的蜜罐节点。

Attack 页面记录攻击源，攻击端口，受攻击节点，攻击事件，攻击协议。

Plyloads 页面显示一些攻击行为的 payload，包括 sql 注入，恶意扫描等，它会记录所有 url 访问。只针对http扫描有记录

Rules 页面可以自定义一些触发规则，一般保持默认就好

Sensors 是各个蜜罐节点的情况，成功运行的蜜罐会在这里显示

Charts 页面会显示 ssh 蜜罐收集到的用户名密码等，这里显示的是 cowrie 捕捉到的一些 ssh 尝试登录，以及 top 用户名和密码