# 网络扫描

**http://sectools.org/tag/web-scanners/**

网络安全扫描top 5



## wireshark

查看网络数据包

sudo apt-get install wireshark

sudo wget <https://www.wireshark.org/download/src/wireshark-1.12.4.tar.bz2>

sudo bzip2 -d wireshark-1.12.4.tar.bz2

sudo tar xf wireshark-1.12.4.tar

cd wireshark-1.12.4/

sudo chown -R dwt:dwt wireshark-1.12.4

sudo ./configure

sudo make

sudo make install

启动wireshark &

**问题：configure: error: I couldn't find yacc (or bison or ...);**

sudo apt-get install flex bison

**问题：configure: error: Qt is not available**

sudo apt-get install libqt4-\*

sudo apt-get install qt5-default

sudo apt-get install qtcreator

http://www.cnblogs.com/tankxiao/archive/2012/10/10/2711777.html

## nmap

NMap（Network Mapper，网络映射器），用来扫描开放的网络连接端口。确定哪些服务运行在哪些端口，并且推断计算机运行哪个操作系统。

其基本功能有三个，一是**探测主机状态**，二是**扫描端口和服务**，三是**推断主机的操作系统**。

wget <https://nmap.org/dist/nmap-6.47.tar.bz2>

bzip2 -cd nmap-6.47.tar.bz2 | tar xvf -

cd nmap-6.47

./configure

make

make install

nmap -A -T4 [www.baidu.com](http://www.baidu.com)

<https://nmap.org/man/zh/>

-p port-ranges：指定扫描端口的范围；

-sV：启动版本检测，Nmap试图识别正在运行的服务及其版本；

-O：启用远程操作系统检测；

-A：启用版本检测和操作系统检测；

-T[0-5]：设置时间选项，较大的数字意味两次探测时间间隔小。

-oN/-oX/-G file：指定文件输出格式，正常/XML/grepable。

# 漏洞扫描

vulnerability scanner

http://resources.infosecinstitute.com/14-popular-web-application-vulnerability-scanners/

网站安全扫描top 5



## Nessus\*\*

漏洞扫描工具的标准，已闭源

## OpenVAS\*\*

OpenVAS是类似Nessus的综合型漏洞扫描器，可以用来识别远程主机、Web应用存在的各种漏洞。Nessus曾经是业内开源漏洞扫描工具的标准，在Nessus商业化不再开放源代码后，在它的原始项目中分支出OpenVAS开源项目。经过多年的发展，OpenVAS已成为当前最好用的开源漏洞扫描工具，功能非常强大，甚至可以与一些商业的漏洞扫描工具媲美。OpenVAS使用NVT（Network Vulnerabilty Test，网络漏洞测试）脚本对多种远程系统（包括Windows、Linux、UNIX以及Web应用程序等）的安全问题进行检测。

OpenVAS开发组维护了一套免费的NVT库，并定期对其进行更新，以保证可以检测出最新的系统漏洞。

### CentOS安装(不成功)

CentOS 7

systemctl stop firewalld.service

systemctl disable firewalld.service

vi /etc/sysconfig/selinux

SELINUX=disabled

yum install -y wget bzip2

wget -q -O - http://www.atomicorp.com/installers/atomic |sh

yum install -y openvas

vi /etc/redis.conf

# create a unix domain socket to listen on

unixsocket /tmp/redis.sock

# set permissions for the socket

unixsocketperm 755

systemctl restart redis.service

systemctl enable redis.service

ll /tmp/redis.sock

openvas-nvt-sync

openvas-scapdata-sync

openvas-certdata-sync

openvas-mkcert-client -n -i

openvasmd --rebuild

openvas-setup

检查安装是否完成，根据提示修改错误

vi openvas-check-setup

<https://svn.wald.intevation.org/svn/openvas/trunk/tools/openvas-check-setup>

chmod +x openvas-check-setup

./openvas-check-setup

wget https://localhost:9392测试一下是否启动，用户名admin密码123456

**问题：扫描任务停止在1%**

openvas-mkcert-client -n om -i

openvasmd --progress --rebuild -v

tail /var/log/openvas/openvasmd.log

tail /var/log/openvas/openvassd.log

**问题：redis-server is not running or not listening on socket:/tmp/redis.sock**

vi /etc/redis.conf

# create a unix domain socket to listen on

unixsocket /tmp/redis.sock

# set permissions for the socket

unixsocketperm 755

重启redis

systemctl enable redis.service

systemctl restart redis.service

ll /tmp/redis.sock 看是否已存在该文件

**问题：ERROR: No OpenVAS SCAP database found. (Tried: /var/lib/openvas/scap-data/scap.db)**

openvas-scapdata-sync

**重启OpenVAS**

systemctl restart openvas-manager

systemctl restart openvas-scanner

systemctl restart gsad

**redis源码安装**

yum install -y tcl gcc

wget http://download.redis.io/redis-stable.tar.gz

tar xvzf redis-stable.tar.gz

cd redis-stable

make MALLOC=libc

make test

make install

redis-server

redis-cli ping

### 命令

openvas-mkcert：创建证书

openvas-mkcert-client -n om -i：创建OpenVAS客户端SSL证书，-n非交互创建。-i安装证书

openvas-scapdata-sync：更新scapdata

openvas-certdata-sync：更新certdata

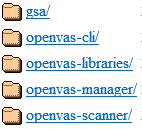
openvas-nvt-sync：更新NVT数据库

openvassd：启动scanner

gsad：启动gsa

### 使用

http://www.openvas.org/src-doc/



openvas-scanner(扫描器)： openvassd，负责调用各种漏洞检测插件，完成实际的扫描操作。

openvas-manager(管理器)：openvasmd，负责分配扫描任务，并根据扫描结果生产评估报告。

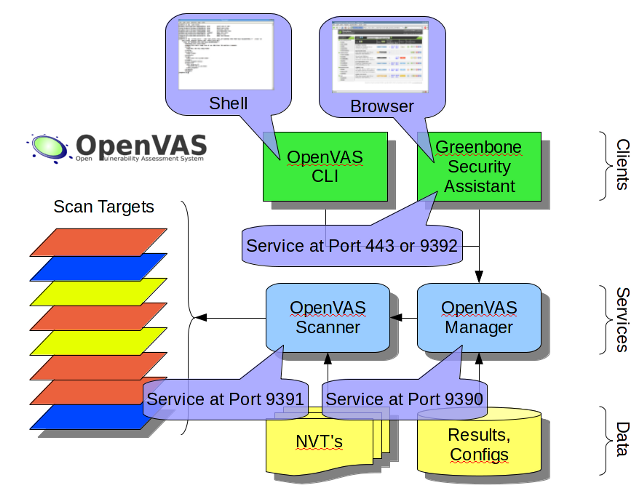
openvas-administrator(管理者)：openvasad，负责管理配置信息，用户授权等相关工作。

openvas-cli(命令行接口)：omp，提供从命令行访问OpenVAS服务层程序。

greenbone-security-assistant(安装助手)：gsad，提供访问OpenVAS Manager的web接口，便于通过浏览器来执行扫描任务，是使用最简便的客户层组件。

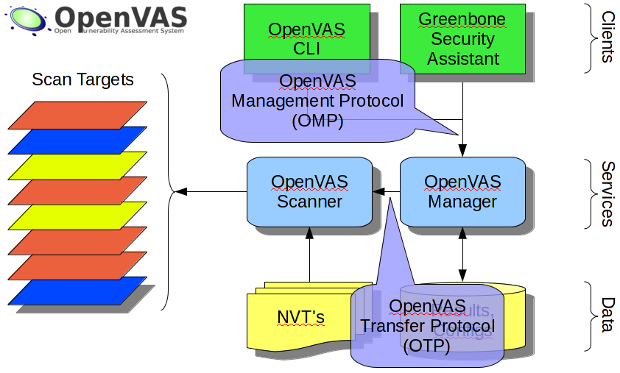
Greenbone-Desktop-Suite (桌面套件)：提供访问OpenVAS Manager的图形程序界面。

NVT特征库

[](http://files.dotblogs.com.tw/ghoseliang/1408/201485135236669.jpg)

Scanner执行NVT（Network Vulnerability Tests）扫描域名，OpenVAS NVT Feed每天都会更新。

Manager通过OTP协议（XML格式）控制Scanner扫描，并向上提供无状态的OMP协议供客户端调用。将配置和扫描结果保存在SQLite数据库中。利用访问控制用户、组、角色。



#### openvasmd

OpenVAS Manager是OpenVAS Scanner和各种客户端（GSA，OMP，GSD）之间的连接层。它保存扫描结果，回收Scanner资源。Manager通过认证授权连接Scanner。

openvasmd：启动manager

openvas-nvt-sync：升级NVT特征库

openvasmd --rebuild：初始化数据库，需要首先启动openvassd，Manager从Scanner中获取所有NVT特征。每次更新完NVT后，必须重建NVT缓存

* --rebuild：锁住数据库
* --update：更新时，客户端仍能访问Manager

openvasmd --migrate：迁移数据库至当前数据模型

openvasmd --create-user=myuser：创建管理员

openvasmd --optimize=<option>：优化数据库

* + vacuum：释放未使用空间以减少文件大小
  + analyze：更新内部数据，优化查询
  + cleanup-config-prefs：删除重复配置
  + remove-open-port-results：删除老版本OpenVAS的检查结果
  + cleanup-port-names：清除不符合老版本中<port>/<protocol>格式的数据

#### omp

除了gsad提供的网站，还可通过omp（openvas-cli）命令行启动扫描任务

<http://www.openvas.org/omp-2-0.html#command_details>

OpenVAS Management Protocol

omp -h localhost -p 9390 -u admin -w 123456 -g 显示配置信息

1. **通过xml启动扫描，查看xml如何配置**

omp -h localhost -p 9390 -u admin -w 123456 -v --xml='<help/>'

1. **查看所有配置信息**

omp -h localhost -p 9390 -u admin -w 123456 -v --xml='<get\_configs/>'

**格式化XML，找到Full and fast的id号**

<config id="**708f25c4-7489-11df-8094-002264764cea**">

<owner>

<name></name>

</owner>

<name>Full and very deep</name>

<comment>Most NVT's; don't trust previously collected information; slow.</comment>

<creation\_time>2015-10-23T08:29:37Z</creation\_time>

<modification\_time>2015-10-23T08:29:37Z</modification\_time>

<writable>0</writable>

<in\_use>1</in\_use>

<permissions></permissions>

<user\_tags>

<count>0</count>

</user\_tags>

<family\_count>59

<growing>1</growing></family\_count>

<nvt\_count>43935

<growing>1</growing></nvt\_count>

<type>0</type>

</config>

1. **创建扫描目标**

omp --host=127.0.0.1 --username=admin --password=123456 -v --xml='<create\_target><name>default</name><hosts>192.168.1.3</hosts></create\_target>'

**记下返回的扫描目标ID**

<create\_target\_response id="**45f4be72-1be4-490c-bc1f-caaccc0193e6**" status\_text="OK, resource created" status="201"></create\_target\_response>

**查看所有扫描目标**

omp --host=127.0.0.1 --username=admin --password=123456 -v --xml='<get\_targets/>'

1. **创建扫描任务**

omp -h localhost -p 9390 -u admin -w 123456 -v --xml="<create\_task><config id='**708f25c4-7489-11df-8094-002264764cea**'/><target id='**45f4be72-1be4-490c-bc1f-caaccc0193e6**'/><name>default</name><comment></comment></create\_task>"

**记下返回的扫描任务ID**

<create\_task\_response id="**c1692d6d-3e20-48c1-a4f2-79390d269939**" status\_text="OK, resource created" status="201"></create\_task\_response>

**查看所有扫描任务**

omp --host=127.0.0.1 --username=admin --password=123456 -v --xml='<get\_tasks/>'

1. **启动扫描任务**

omp --host=127.0.0.1 --username=admin --password=123456 -v -S $task\_id

omp --host=127.0.0.1 --username=admin --password=123456 -v --xml="<start\_task task\_id='**c1692d6d-3e20-48c1-a4f2-79390d269939**'/>"

**记下返回的输出ID**

<start\_task\_response status\_text="OK, request submitted" status="202"><report\_id>**b4b4c7ba-d650-4ab7-86a7-4875508409ed**</report\_id></start\_task\_response>

1. **返回所有输出格式**

omp --host=127.0.0.1 --username=admin --password=123456 -v --get-report-formats

omp --host=127.0.0.1 --username=admin --password=123456 --get-report **b4b4c7ba-d650-4ab7-86a7-4875508409ed** --format=**a3810a62-1f62-11e1-9219-406186ea4fc5** > results.txt

# omp --help

Usage:

omp [OPTION...] - OpenVAS OMP Command Line Interface

Help Options:

-?, --help Show help options

Application Options:

-h, --host=<host> Connect to manager on host <host>

-p, --port=<number> Use port number <number>

-V, --version Print version.

-v, --verbose Verbose messages (WARNING: may reveal passwords).

--use-certs Use client certificates to authenticate.

--client-cert=<cert-file> Client certificate. Default: /var/lib/openvas/CA/clientcert.pem

--client-key=<key-file> Client key. Default: /var/lib/openvas/private/CA/clientkey.pem

--client-ca-cert=<cert-file> Client CA certificate. Default: /var/lib/openvas/CA/cacert.pem

-u, --username=<username> OMP username

-w, --password=<password> OMP password

--config-file=<config-file> Configuration file for connection parameters.

-P, --prompt Prompt to exit.

-O, --get-omp-version Print OMP version.

-n, --name=<name> Name for create-task.

-C, --create-task Create a task.

-m, --comment=<name> Comment for create-task.

-c, --config=<config> Config for create-task.

-t, --target=<target> Target for create-task.

-E, --delete-report Delete one or more reports.

-D, --delete-task Delete one or more tasks.

-R, --get-report Get report of one task.

-F, --get-report-formats Get report formats. (OMP 2.0 only)

-f, --format=<format> Format for get-report.

--filter=<string> Filter string for get-report

-G, --get-tasks Get status of one, many or all tasks.

-g, --get-configs Get configs.

-T, --get-targets Get targets.

-i, --pretty-print In combination with -X, pretty print the response.

-S, --start-task Start one or more tasks.

-M, --modify-task Modify a task.

--ping Ping OMP server

--timeout=<number> Wait <number> seconds for OMP ping response

--file Add text in stdin as file on task.

-X, --xml=<command> XML command (e.g. "<help/>"). "-" to read from stdin.

--send-file=<file> Replace SENDFILE in xml with base64 of file.

--details Enable detailed view.

## Nexpose\*

商业

## Nikto

Web Server测试工具，检查漏洞脚本，配置错误等6700多个潜在安全问题，不提供XSS和SQL注入检查。

Ubuntu 14.10

sudo apt-get install perl openssl libnet-ssleay-perl

wget <http://www.cirt.net/nikto/nikto-2.1.5.tar.gz>

tar -xvf nikto-2.1.5.tar.gz

cd nikto-2.1.5

perl nikto.pl -host 192.168.1.2 -port 8080

结果只能输出到文件，没有数据库

更新：

perl nikto.pl -update

## w3af

w3af是一个Web应用程序攻击和检查框架.该项目已超过130个插件,其中包括检查网站爬虫,SQL注入(SQL Injection),跨站(XSS),本地文件包含(LFI),远程文件包含(RFI)等.该项目的目标是要建立一个框架,以寻找和开发Web应用安全漏洞,所以很容易使用和扩展。

<http://docs.w3af.org/en/latest/index.html>

Python

### 安装

Ubuntu 14.10

sudo apt-get install git

python2.7自带

sudo apt-get install python-pip python-dev libffi-dev openssl

sudo pip install --upgrade pip

git clone https://github.com/andresriancho/w3af.git

cd w3af/

./w3af\_console或者./w3af\_gui

. /tmp/w3af\_dependency\_install.sh

### 使用

<http://blog.idf.cn/wp-content/uploads/2013/04/IDF%E4%BD%9C%E5%93%81%EF%BC%9Aw3af%E7%94%A8%E6%88%B7%E6%89%8B%E5%86%8C.pdf>

./w3af\_console

w3af>>> help

w3af>>> help target

Configure the target URL

w3af>>> plugins

w3af/plugins>>> help

w3af/plugins>>> list crawl

w3af/plugins>>> crawl web\_spider

w3af/plugins>>> list audit

w3af/plugins>>> audit all

w3af/plugins>>> audit

w3af/plugins>>> audit desc xss

w3af/plugins>>> audit config xss

w3af/plugins/audit/config:xss>>> view

w3af/plugins>>> audit config xss

w3af/plugins/audit/config:xss>>> view

w3af/plugins>>> attack all

w3af>>> target

w3af/config:target>>> set target http://192.168.1.2:8080

w3af/config:target>>> back

w3af>>>start

plugins：

* crawl：漏洞挖掘，寻找URL和注入点，如web spider，输入为URL，输出为多个注入点。
* audit：漏洞分析，根据crawl plugins找到的注入点，发送特殊数据定位漏洞，如SQL注入发送a’b’’c到所有注入点。
* attack：漏洞攻击，利用audit plugins找到的漏洞，如返回远程服务器的shell或从SQL注入的漏洞dump数据表。
* infrastructure：系统认证信息
* grep：漏洞匹配，分析HTTP请求响应体中的漏洞。如在HTML请求体中找到password，生成漏洞。
* output：输出文本，xml或html。
* managle：基于正则表达式修改请求响应。
* bruteforce：暴力破解，在crawl阶段暴力登陆。
* evasion：通过修改HTTP数据包绕过入侵检测规则。

推荐配置：

* crawl: web\_spider
* audit: Enable all
* grep: Enable all

重复执行可使用脚本，脚本实例在scripts目录下

vi scripts/exploit\_fast.w3af

修改target配置

./w3af\_console -s scripts/exploit\_fast.w3af

## SQLMap

SQLMap是检测利用SQL注入漏洞的免费开源工具

六种SQL注入技术：

* boolean-based blind
* time-based blind
* error-based
* UNION query-based：UNION ALL SELECT
* stacked queries
* out-of-band

### 安装

sudo apt-get install git

git clone <https://github.com/sqlmapproject/sqlmap.git>

cd sqlmap

python sqlmap.py --version

python sqlmap.py -u http://flc.jlu.edu.cn/index.php?action=dxwy

列出数据库名

sqlmap -u http://flc.jlu.edu.cn/index.php?action=dxwy --dbs

-D指定数据库名，查看数据库表

sqlmap -u http://flc.jlu.edu.cn/index.php?action=dxwy -D vhost48330 --tables

-T指定数据库表，列出所有字段

sqlmap -u http://flc.jlu.edu.cn/index.php?action=dxwy -D vhost48330 -T admin --columns

-C指定数据库字段，--dump备份数据库

sqlmap -u http://flc.jlu.edu.cn/index.php?action=dxwy -D vhost48330 -T admin -C ac,id,password --dump

--cookie=COOKIE                 在需要登录的地方，需要登录后的cookie

--proxy="[http://127.0.0.1:8087](http://127.0.0.1:8087/)" 使用HTTP代理隐藏自己的身份，比如使用goagent等

--sql-query=QUERY               执行一个sql语句，不一定支持

## Wapiti

## Grabber

Python，无GUI无PDF输出，自用，无法商用

检查如下漏洞：

* Cross site scripting
* SQL injection
* Ajax testing
* File inclusion
* JS source code analyzer
* Backup file check

## Vega

https://subgraph.com/vega

Java，有GUI，OS X, Linux and Windows

检查如下漏洞：

* SQL injection
* header injection
* directory listing
* shell injection
* cross site scripting
* file inclusion

## Arachni

## WebScarab

Java

## Zed Attack Proxy

Windows, Unix/Linux and Macintosh

## Ratproxy

## Skipfish

C，google出品，效率高1秒处理2000请求

## Wfuzz

暴力破解web应用工具，支持对网站目录、登录信息、应用资源文件等的暴力猜解， 还可以进行GET及POST参数的猜解，sql注入、xss漏洞的测试等。该工具所有功能都依赖于字典。

git clone <https://github.com/xmendez/wfuzz.git>

./wfuzz.py -h 查看命令

./wfuzz.py -c -z file,wordlist/general/common.txt --sc 200 <http://10.10.4.132:8080/FUZZ>

在wordlist下面提供了很多破解文件，--sc只接受返回码为200的结果

**参数中FUZZ 相当一个变量，用字典中的字段来替换它来完成猜解。**

./wfuzz.py -c -z file,wordlist/general/common.txt --hc 404 -o html http://10.10.4.132:8080/FUZZ > results.html

暴力破解10.10.4.132:8080上的资源，隐藏返回码为404的结果，使用common.txt中的字段代替FUZZ，输出格式为html，输出文件名为results.html

./wfuzz.py -c -z range,1-100 --hc 404 http:// 10.10.4.132:8080/list.asp?id=FUZZ

从1至100作为参数代替FUZZ

## Watcher

只支持window

Watcher是一个实时的，被动分析基于HTTP协议的Web应用程序的安全工具(Fiddler插件)。所谓被动分析就是指它不会对现有系统产生任何破坏。Watcher可以被安全的用于分布式网络，共享式网络和专用宿主网络。Watcher能够自动检查这些问题：cookie设置、SSL配置、信息泄露等。另外，Watcher是一个非常好的HTTP协议调试工具，不过，它的HTTP调试功能是以Fiddler插件方式工作的。总之，Watcher是一个很好的渗透测试工具，有助于渗透测试人员的渗透测试工作。

## Netsparker

## Grendel-Scan

# 入侵检测

* IDS：入侵检测系统，Intrusion Dection System，对网络、系统的运行状态进行监视，必须挂接在所有流量必经的链路上。
* IPS：入侵防御系统，Intrusion Prevention System，必须部署在数据转发路径上，根据预先设定的安全策略，对流经报文进行深度检测，一旦发现网络攻击，则根据攻击的威胁级别采取抵御措施命入告警、丢弃报文、切断会话、切断TCP连接等。

IDS分为Host IDS和Network IDS：

* HIDS：检测和分析主机内部异常，如文件系统、网络设备进出包、安装软件等。
* NIDS：在流经所有设备流量的位置部署（如防火墙处），将流量与已知攻击比对，当发现异常行为时向管理员告警。著名实现有OPNET和NetSim。

https://www.alienvault.com/blogs/security-essentials/open-source-intrusion-detection-tools-a-quick-overview

入侵检测分为特征检测和异常检测（Signature detection和Anomaly Detection）。

* 特征检测IDS通过检查恶意流量的规则和模式满足特征后发出警报。
* 异常检测IDS通过检测偏离平均值的异常活动，如服务器发出比以往多的HTTP请求或者进入DMZ的新主机。

|  |  |  |
| --- | --- | --- |
|  | Pros | Cons |
| Snort | 易于安装，用户广泛，资源丰富 | 没有GUI，包处理慢 |
| Suricata | 使用Snort规则，多线程GPU加速 | 容易误报，系统和网络资源消耗大 |
| Bro IDS | 平台适用于各种网络安全情况 | 需要编程经验，Bro DSL获取不够成熟 |
| OpenWIPS-ng | 模块化插拔，可DIY配置软硬件 | 为无线安全解决方案 |
| Security Onion | 由多种领先开源解决方案组成的安全栈，提供简易的安装配置工具 | 由多种技术组成的平台，也继承了各技术的缺点 |

## Snort\*\*

已成为IDS标准。已由IDS发展为IPS，由于缺少GUI和管理员console，需要open source tools辅助，如Snortby，Base或Squil。

### 安装

#### rpm安装

操作系统：CentOS 7

<http://www.packet6.com/installing-snort-on-centos/>

<http://wiki.aanval.com/wiki/Community:Snort_2.9.4.X_Installation_Guide_for_CentOS_6.3>

虚拟机必须采用桥接方式，设置好静态IP，NETMASK，DNS

yum install -y wget gcc flex bison zlib zlib-devel libpcap libpcap-devel pcre pcre-devel libdnet libdnet-devel tcpdump mlocate

yum install https://www.snort.org/downloads/snort/daq-2.0.6-1.centos7.x86\_64.rpm

yum install https://www.snort.org/downloads/snort/snort-**2.9.7.6**-1.centos7.x86\_64.rpm

wget <https://www.snort.org/downloads/community/community-rules.tar.gz>

tar -xvf community-rules.tar.gz -C /etc/snort

snortrule分为community-rules（所有人可下载），snortrules-snapshot（注册用户可下载，比subscription晚一个月），subscription（付费后可下载）

注册后可获得oinkcode号，下载snortrules-snapshot的后缀必须和snort版本号一致

wget https://www.snort.org/rules/snortrules-snapshot-**2976**.tar.gz?oinkcode=9a212bf809503114f4f33ec591b323b3ad149a1f

mv snortrules-snapshot-2976.tar.gz?oinkcode=9a212bf809503114f4f33ec591b323b3ad149a1f snortrules-snapshot-2976.tar.gz

tar zxvf snortrules-snapshot-2976.tar.gz -C /etc/snort

解压后的文件里包含最新的防护规则，可用该规则覆盖原有配置以生效

ls

etc preproc\_rules rules so\_rules

mv /etc/snort/etc/\* /etc/snort/

touch /etc/snort/rules/white\_list.rules /etc/snort/rules/black\_list.rules

chown -R snort:snort /etc/snort

vi /etc/snort/snort.conf

# such as: c:\snort\rules

var RULE\_PATH ./rules

var SO\_RULE\_PATH ../so\_rules

var PREPROC\_RULE\_PATH ./preproc\_rules

# If you are using reputation preprocessor set these

var WHITE\_LIST\_PATH ./rules

var BLACK\_LIST\_PATH ./rules

# path to dynamic preprocessor libraries

dynamicpreprocessor directory /usr/lib64/snort-2.9.7.6\_dynamicpreprocessor

# path to base preprocessor engine

dynamicengine /usr/lib64/snort-2.9.7.6\_dynamicengine/libsf\_engine.so

mkdir -p /usr/local/lib/snort\_dynamicrules

chown -R snort:snort /usr/local/lib/snort\_dynamicrules

chmod -R 700 /usr/local/lib/snort\_dynamicrules

测试是否配置成功：

snort -T -i eno16777736 -u snort -g snort -c /etc/snort/snort.conf

添加规则，任何访问本机的ICMP和TCP数据包都会触发警告。

vi /etc/snort/rules/local.rules

alert icmp any any -> $HOME\_NET any (msg:"ICMP Test NOW!!!"; classtype:not-suspicious; sid:1000001; rev:1;)

alert tcp any any -> $HOME\_NET 80 (msg:"HTTP Test NOW!!!"; classtype:not-suspicious; sid:1000002; rev:1;)

snort -T -c /etc/snort/snort.conf

snort -d -h 192.168.1.0/24 -l . -c /etc/snort/snort.conf -i eno16777736

vi log/alert

**问题：ERROR: /etc/snort/snort.conf(253) Could not stat dynamic module path "/usr/local/lib/snort\_dynamicrules": No such file or directory.**

mkdir -p /usr/local/lib/snort\_dynamicrules

chown -R snort:snort /usr/local/lib/snort\_dynamicrules

chmod -R 700 /usr/local/lib/snort\_dynamicrules

将动态规则拷贝到该目录下

**问题：ERROR: /etc/snort/snort.conf(243) Could not stat dynamic module path "/usr/local/lib/snort\_dynamicpreprocessor/": No such file or directory.**

找不到动态连接库，

方法1：

ln -s /usr/local/lib/snort\_dynamicpreprocessor /usr/lib64/snort-2.9.7.3\_dynamicpreprocessor

ln -s /usr/local/lib/snort\_dynamicengine /usr/lib64/snort-2.9.7.3\_dynamicengine/libsf\_engine.so

方法2：修改/etc/snort/snort.conf文件中的指向

**问题：ERROR: Cannot decode data link type 239**

需要设置端口

snort -v -i eno16777736

#### 导入MySQL

从2.9.3.0版本之后不再支持直接写入MySQL，而是通过Barnyard2读取unified2文件到MySQL中。

将snortrules下的sid-msg.map拷贝到/etc/snort/下

vi /etc/snort/snort.conf

# Setup the network addresses you are protecting

ipvar HOME\_NET any

# Set up the external network addresses. Leave as "any" in most situations

ipvar EXTERNAL\_NET any

输出文件

output unified2: filename snort.log, limit 128, mpls\_event\_types, vlan\_event\_types

**配置数据库**

wget <http://dev.mysql.com/get/mysql-community-release-el7-5.noarch.rpm>

rpm -ivh mysql-community-release-el7-5.noarch.rpm

yum install -y mysql-community-server mysql-devel

systemctl start mysql

echo "CREATE DATABASE snort;" | mysql -u root -p

mysql -D snort -u root -p < create\_mysql create\_mysql在github的schemas目录下

<https://github.com/eldondev/Snort/blob/master/schemas/create_mysql>

GRANT ALL PRIVILEGES on snort.\* to snort@'%' IDENTIFIED BY '123456';

GRANT ALL PRIVILEGES on snort.\* to snort@'localhost' IDENTIFIED BY '123456';

**安装Barnyard2**

yum install -y git autoconf libtool

git clone <https://github.com/firnsy/barnyard2.git>

cd barnyard2

./autogen.sh

./configure --with-mysql --with-mysql-libraries=/usr/lib64/mysql

make

make install

cp etc/barnyard2.conf /etc/

vi /etc/barnyard2.conf

config hostname: localhost

config interface: eno16777736

config logdir: /var/log/barnyard2/

config waldo\_file: /var/log/barnyard2/barnyard2.waldo

output database: log, mysql, dbname=snort user=snort password=123456 host=localhost

mkdir /var/log/barnyard2/

**测试**

snort -d -h 192.168.1.0/24 -l /var/log/snort -c /etc/snort/snort.conf -i eno16777736

barnyard2 -c /etc/barnyard2.conf -d /var/log/snort -f snort.log -w /var/log/barnyard2/barnyard2.waldo

查看snort数据库是否新增记录

select count(\*) from event;

**开机启动(不成功)**

vi barnyard2

#!/bin/sh

#

# barnyard2

#

# read file into mysql

#

# chkconfig: 2345 20 80

# description: Runs the IBM Power RAID adapter initialization daemon

# processname: barnyard2

# pidfile: /var/run/barnyard2.pid

#

case $1 in

start)

echo "starting $0..."

barnyard2 -c /etc/barnyard2.conf -d /var/log/snort -f snort.log -w /var/log/barnyard2/barnyard2.waldo

echo -e 'done.'

;;

stop)

echo "stopping $0..."

killall barnyard2

echo -e 'done.'

;;

restart)

$0 stop

$0 start

;;

\*)

echo "usage: $0 (start|stop|restart)"

;;

esac

chmod 700 barnyard2

cp barnyard2 /etc/init.d/

#### 源码安装

https://www.andrew.cmu.edu/user/rdanyliw/snort/snortdb/snortdb\_install.html

yum install -y gcc flex bison libpcap libpcap-devel pcre-devel libdnet-devel zlib-devel

tar zxvf daq-2.0.6.tar.gz

cd daq-2.0.6

./configure

make

make install

tar zxvf snort-2.9.7.6.tar.gz

cd snort-2.9.7.6

./configure

make

make install

wget <https://www.snort.org/downloads/community/community-rules.tar.gz>

tar -xvf community-rules.tar.gz -C /etc/snort

mkdir -p /etc/snort

wget https://www.snort.org/rules/snortrules-snapshot-**2976**.tar.gz?oinkcode=9a212bf809503114f4f33ec591b323b3ad149a1f

tar zxvf ~/snortrules-snapshot-2976.tar.gz -C /etc/snort

cp /etc/snort/etc/\* /etc/snort/

mkdir /usr/local/lib/snort\_dynamicrules

touch /etc/snort/rules/white\_list.rules /etc/snort/rules/black\_list.rules

groupadd -g 40000 snort

useradd snort -u 40000 -d /var/log/snort -s /sbin/nologin -c SNORT\_IDS -g snort

cd /etc/snort/

chown -R snort:snort \*

chown -R snort:snort /var/log/snort

chown -R snort:snort /usr/local/lib/snort\_dynamicrules

chmod -R 700 /usr/local/lib/snort\_dynamicrules

vi /etc/snort/snort.conf

# such as: c:\snort\rules

var RULE\_PATH ./rules

var SO\_RULE\_PATH ../so\_rules

var PREPROC\_RULE\_PATH ./preproc\_rules

# If you are using reputation preprocessor set these

var WHITE\_LIST\_PATH ./rules

var BLACK\_LIST\_PATH ./rules

# path to dynamic preprocessor libraries

dynamicpreprocessor directory /usr/local/lib/snort\_dynamicpreprocessor/

# path to base preprocessor engine

dynamicengine /usr/local/lib/snort\_dynamicengine/libsf\_engine.so

snort -d -h 192.168.1.0/24 -l . -c /etc/snort/snort.conf -i eno16777736

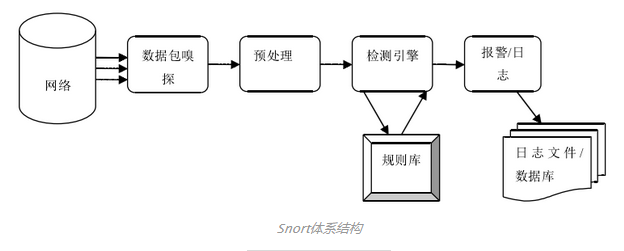
### 使用

<http://drops.wooyun.org/%E8%BF%90%E7%BB%B4%E5%AE%89%E5%85%A8/9232>

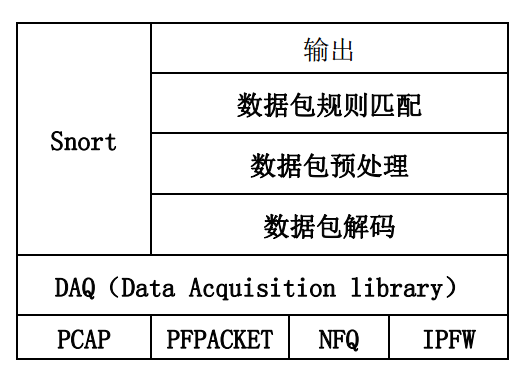
#### 工作模式

Snort有三种工作模式：

* + Sniffer mode嗅探器：仅仅是从网络上读取数据包并作为连续不断的流显示在终端上。通过-v启动。
  + Packet Logger mode数据包记录器模式：把数据包记录到硬盘上。
  + NIDS网路入侵检测模式：是最复杂的，而且是可配置的。我们可以让snort分析网络数据流以匹配用户定义的一些规则，并根据检测结果采取一定的动作。通过-c 加配置文件启动。



#### 模块



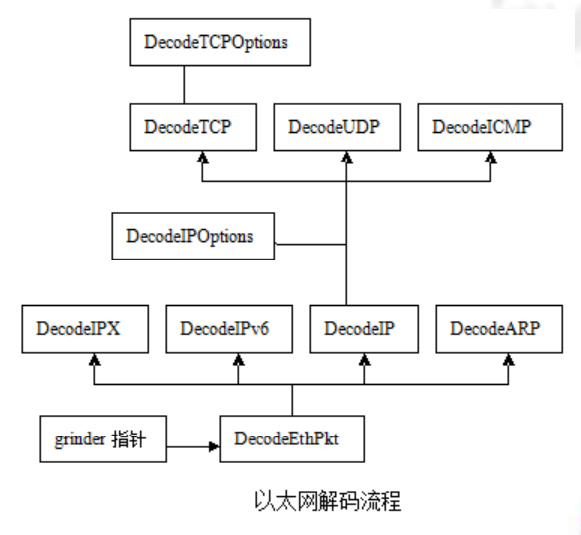
1. **数据包捕获模块**

通过DAQ调用底层函数库，捕获网络数据包；

1. **解码模块**

对捕获数据包进行协议解码；

将捕获的数据包通过下图的流程解码成Packet结构体。通过数据包的协议字段，选择IPX、IPv6、IP、ARP之一协议解码，IP数据包再根据TCP，UDP，ICMP解码。



1. **预处理模块**

以插件形式存在，数据包需要经过预处理才能被主探测引擎进行规则匹配，对IP分片进行重组，防止ARP欺骗等；

Snort主要包含以下预处理器：

• 包重组预处理器：为了防止攻击信息被拆分到多个包中从而逃避了Snort的检测；

Frag3：IP分片重组和攻击监测。

Stream5：维持TCP流状态，进行会话重组。

* • 协议规范化预处理器：把数据包协议解码成一个统一的格式，再传送给检测模块；

HttpInspect：规范HTTP流，如将Unicode或hex翻译成snort可以识别的 字符集。

RpcDecode：规范RPC调用。

• 异常检测预处理器：

ARPspoof：检测ARP欺骗。

SfPortscan：检测端口扫描。

1. **规则匹配模块**

根据设置规则对数据包进行匹配；

规则头由RuleTreeNode(RTN)结构表示，规则选项由 OptTreeNode(OTN)表示。

* RTN结构中主要包括：

struct RuleTreeNode{

IpAddrSet IpAddrSet \*sip，\*dip：源IP地址，目的IP地址。

PortObject \*src\_portobject，\*dst\_portobject：源端口，目的端口。

OptTreeNode \*down：指向相关的OTN。

RuleFpList \*rule\_func：规则头匹配函数。

}

* OTN结构中主要包括：

struct OptTreeNode{

OptFpList \*opt\_func：规则选项匹配函数。

void \*ds\_list[PLUGIN\_MAX]：插件需要的数据结构指针。

RuleTreeNode \*\*proto nodes：指向其关联的RTN。

}

1. **输出模块**

以插件形式存在，当匹配成功时记录或报警。

#### 命令

snort

-A：6种报警机制：full，fast，socket，syslog，smb，none

-d：输出应用层包

-e：输出数据链路层包

-v：verbose，降低执行速度可能导致丢包

-i：网卡端口

-l：日志输出目录，**如果指定目录，则输出日志，目录必须存在**

-h：监听网段，**如果指定，则对进入该网段的所有数据包检测，如果未指定，则记录本机数据包**

-b：以二进制格式记录日志，速度快

-r：从日志中读取数据包

-T：测试并报告当前配置文件是否正确

#### 规则（重要）

snort的规则定义方式已成为所有IDS的规则标准。

http://man.chinaunix.net/network/snort/Snortman.htm

规则分为规则头(rule header)和规则选项(rule options)两部分：

alert tcp any any -> 192.168.1.0/24 143 (content: “|90C8 C0FF FFFF|/bin/sh”;

msg: “IMAP buffer overflow!”;)

动作 协议 源IP地址 源端口 方向 目标地址 目标端口 ( 规则选项)



1. **Rule Header**
2. **action**

alert告警并记录，log记录，pass忽略，activate告警并跳至另一动态规则，dynamic被activate rule激活前保持空闲，drop丢弃并记录，reject阻塞并记录，sdrop阻塞不记录

activate和dynamic实例，第二条规则被第一条规则激活

\begin{figure}\begin{verbatim}activate tcp !$HOME_NET any -> $HOME_NET 143 (f...
...ny -> $HOME_NET 143 (activated_by:1; count:50;)\end{verbatim}
\par\end{figure}

1. **protocols**

TCP，UDP，ICMP，IP，ARP，IGRP，GRE，OSPF，RIP，IPX

1. **IP或网段**

\begin{figure}\begin{verbatim}alert tcp !192.168.1.0/24 any -> 192.168.1.0/24...
... 86 a5\vert''; msg:''external mountd access'';)\end{verbatim}
\par\end{figure}

1. **Port**

\begin{figure}\begin{verbatim}log udp any any -> 192.168.1.0/24 1:1024\end{ve...
... or equal to 1024 going to ports
greater than or equal to 500
\par\end{figure}

1. **direction operator**

只有->和<>两类方向操作符，没有<-

\begin{figure}\begin{verbatim}log tcp !192.168.1.0/24 any <> 192.168.1.0/24 23\end{verbatim}
\par\end{figure}

1. **Rule Options**
2. **msg**

输出日志，msg:”<message>”

1. **reference**

引用外部攻击说明系统，在output中调用指定系统返回攻击说明，用以输出攻击详细信息

reference:<id system>, <id>; [reference:<id system>, <id>;]

引用时用<id>配合url前缀，调用

1. **gid**

当一条规则被激活时，说明哪个部分激活该规则，也可以理解为对decoder，preprocessor，rule进行分组。1代表Rule subsystem，100以上代表preprocessors和decoder，具体查看vi /usr/share/doc/snort-2.9.7.6/generators，与sid同用。gid:<generator id>;gid是可选的，如果在规则中没有指定gid则默认为1，即普通规则。为避免冲突，自定义分组请从1000000开始。

1. **sid**

规则唯一标志，输出时便于查找。必填，小于100保留未来使用，100-999999Snort使用，>=1000000local rule使用，具体查看sig-msg.map（消息与规则ID的映射），与rev同用。

1. **rev**

修订版本号，与sid同用可以唯一确定是哪条规则的第几个版本。

1. **classtype**

对规则进行分类，可选值在classification.config中。

1. **priority**

严重程度，覆盖classtype的默认优先级

1. **metadata**

元数据，metadata: key value, key value;

1. **content**

包匹配内容，如果匹配则触发响应，||代表十六进制内容，不支持正则表达式。

1. **pcre**

包匹配内容，支持perl的正则表达式，前面有content可以快速过滤掉不符合要求的packet

#### 举例

alert tcp $EXTERNAL\_NET $HTTP\_PORTS -> $HOME\_NET any (msg:"BROWSER-CHROME Google Chrome net-internals uri fragment identifier XSS attempt"; flow:established,to\_client; file\_data; content:"chrome://net-internals/view-cache/"; nocase; pcre:"/^\x23([\x22\x27\x3c\x3e\x28\x29]|script|onload|src)/iR"; metadata:service http; reference:bugtraq,39667; reference:cve,2010-1503; reference:url,code.google.com/p/chromium/issues/detail?id=40137; classtype:attempted-user; sid:23471; rev:5;)

flow：established建立起TCP连接，to\_client服务器响应

file\_data：在指定缓冲区中搜索content内容

nocase：忽略大小写

pcre：匹配正则表达式，前面有content可以快速过滤掉不符合要求的packet

metadata：service http目标服务的标示符，这里表明是http服务

reference：引用外部攻击说明系统，在output中调用指定系统返回攻击说明，用以输出攻击详细信息

classtype：对攻击类型分类，分类可选值在classification.config中

### 目录



#### snort.conf

核心配置文件，用于启动NIDS。

1) Set the network variables. 设置各类网络地址，规则中易于使用

2) Configure the decoder 设置解码器

3) Configure the base detection engine 设置基础检测引擎

4) Configure dynamic loaded libraries 设置动态链接库

5) Configure preprocessors 设置预处理器

6) Configure output plugins 设置输出插件

7) Customize your rule set 设置自定义规则

8) Customize preprocessor and decoder rule set设置预处理、解码器规则

9) Customize shared object rule set 设置共享对象规则集

#### classification.config

**规则分类配置文件**

# config classification:shortname,short description,priority

**#格式为名称,描述,优先级**

config classification: attempted-admin,Attempted Administrator Privilege Gain,1

config classification: attempted-recon,Attempted Information Leak,2

可在规则中通过classtype设置规则所属的classtype，该classtype必须已在classification.config中定义。如果规则中不设置priority则使用classification.config中的默认优先级，如果设置则覆盖默认优先级。

alert TCP any any -> any 80 (msg: "EXPLOIT ntpdx overflow"; \

dsize: > 128; classtype:attempted-admin; priority:10;

priority设为10，覆盖了默认的1

alert TCP any any -> any 25 (msg:"SMTP expn root"; flags:A+; \

content:"expn root"; nocase; classtype:attempted-recon;)

没有设置priority，采用默认的2

#### reference.config

**外部参考配置文件**

# config reference: system URL

**# 格式为代号 url前缀**

config reference: bugtraq http://www.securityfocus.com/bid/

config reference: cve http://cve.mitre.org/cgi-bin/cvename.cgi?name=

config reference: arachNIDS http://www.whitehats.com/info/IDS

config reference: osvdb http://osvdb.org/show/osvdb/

# Note, this one needs a suffix as well.... lets add that in a bit.

config reference: McAfee http://vil.nai.com/vil/content/v\_

config reference: nessus http://cgi.nessus.org/plugins/dump.php3?id=

config reference: url http://

config reference: msb <http://technet.microsoft.com/en-us/security/bulletin/>

例如：

alert tcp any any -> any 7070 (msg:"IDS411/dos-realaudio"; \

flags:AP; content:"|fff4 fffd 06|"; reference:bugtraq,1387; )

当该规则被激活时，output plugin会根据reference的配置在reference.config文件中找到bugtraq对应的前缀<http://www.securityfocus.com/bid/>，在末尾附上1387，最终调用url为<http://www.securityfocus.com/bid/1387>，获得返回内容后，解析bugtraq格式的返回值，获取并输出攻击描述详情。

#### sid-msg.map

**sid与sid含义描述的映射关系表**

**以sid-msg.map中的最后一条记录为例，格式为sid || sid含义描述。**

34368 || **MALWARE-CNC Win.Trojan.Banload variant outbound connection** || url,www.virustotal.com/en/file/fc2cc624c2357bad23eaff951c4eac3a1f1c1c3ec5133665c7e101f4f4e3bbba/analysis/1430145774/

**其对应的规则在rules/malware-cnc.rules中**

alert tcp $HOME\_NET any -> $EXTERNAL\_NET $HTTP\_PORTS (msg:"**MALWARE-CNC Win.Trojan.Banload variant outbound connection**"; flow:to\_server,established; urilen:19; content:"/arquivo/cookie.txt"; fast\_pattern:only; http\_uri; content:"Accept: text/html, \*/\*|0D 0A|Accept-Encoding: identity|0D 0A|"; http\_header; metadata:impact\_flag red, policy balanced-ips drop, policy security-ips drop, ruleset community, service http; reference:url,www.virustotal.com/en/file/fc2cc624c2357bad23eaff951c4eac3a1f1c1c3ec5133665c7e101f4f4e3bbba/analysis/1430145774/; classtype:trojan-activity; sid:34368; rev:1;)

所以新增规则时除了在rules/local.rules中新增，还要在sid-msg.map里新增相应记录。

**为何要重复写两次msg呢？**

因为Snort输出的unified文件不包含msg字段只有sid，所以当使用barnyard2将unified文件中的数据导入MySQL时，要通过sid-msg.map的sid找到相应的msg，写入数据库。

如果sid-msg.map中没有找到相应的msg，则数据库中只有gid:sid，而不是gid-msg:sid-msg。gid-msg.map也是这个原因而存在。

Snort提供perl脚本将rules规则转换为sid-msg.map。

create-sidmap.pl /etc/snort/rules > /etc/snort/sid-msg.map

#### gen-msg.map

**gid与gid含义描述的映射关系表**

# Format: generatorid || alertid || MSG

**# 格式为gid, alertid, MSG（用于描述gid含义）**

1 || 1 || snort general alert

2 || 1 || tag: Tagged Packet

3 || 1 || snort dynamic alert

105 || 1 || spp\_bo: Back Orifice Traffic Detected

106 || 1 || spp\_rpc\_decode: Fragmented RPC Records

112 || 1 || spp\_arpspoof: Directed ARP Request

116 || 1 || snort\_decoder: WARNING: Not IPv4 datagram

119 || 1 || http\_inspect: ASCII ENCODING

gid是对decoder，preprocessor和rule进行分组。

preproc\_rules /decoder.rules中的规则通过gid基本都是116。

preproc\_rules /preprocessor.rules中的规则通过gid分为若干组。

BO的gid为105

RPC的gid为106

ARPSPOOF的gid为112

HI\_CLIENT的gid为119

rules/\*.rule由于规则中没有设置gid，gid默认为1，属于snort general alert。

#### unicode.map

预处理器http\_inspect编码翻译文件

#### threshold.conf

事件过滤配置文件

#### preproc\_rules

预处理器及解码器规则集

#### so\_rules

Share Object规则集

#### rules

规则集

### Snortby(不成功)

入侵检测数据展示

安装好Snort、Suricata或Sagan

**Git,Ruby:**

yum install git ruby patch

**Rails:**

gem source -a <http://rubygems.org>

gem sources -a <http://ruby.taobao.org/>

gem update

gem update --system

gem install rails -V

**rake:**

gem install rake -V -v '0.9.2'

**ImageMagick:**

yum install -y ImageMagick ImageMagick-devel

**Wkhtmltopdf:**

wget <http://jaist.dl.sourceforge.net/project/wkhtmltopdf/0.12.2.1/wkhtmltox-0.12.2.1_linux-centos7-amd64.rpm>

yum localinstall -y wkhtmltox-0.12.2.1\_linux-centos7-amd64.rpm

**mariadb:**

yum install mariadb-server

systemctl start mariadb.service

systemctl enable mariadb.service

mysql -u root -p

**Snorby:**

cd /var/www/

git clone <http://github.com/Snorby/snorby.git>

cd snorby/

bundle install

bundle install --path vendor/cache

vi /etc/snort/snort.conf

517行添加：

output unified2: filename snort.log, limit 128, mpls\_event\_types, vlan\_event\_types

问题：gem update

**ERROR: While executing gem ... (Gem::SpecificGemNotFoundException)**

Could not find a valid gem 'json' (= 1.8.3) locally or in a repository

gem source -a <http://rubygems.org>

问题：gem install rails

**ERROR: While executing gem ... (Gem::RemoteFetcher::FetchError)**

Errno::ECONNRESET: Connection reset by peer - SSL\_connect (<https://api.rubygems.org/quick/Marshal.4.8/rails-4.2.2.gemspec.rz>)

gem sources --remove <https://rubygems.org/>

gem sources -a <http://ruby.taobao.org/>

gem sources -l

**问题：Failed to complete patch task**

yum install -y patch

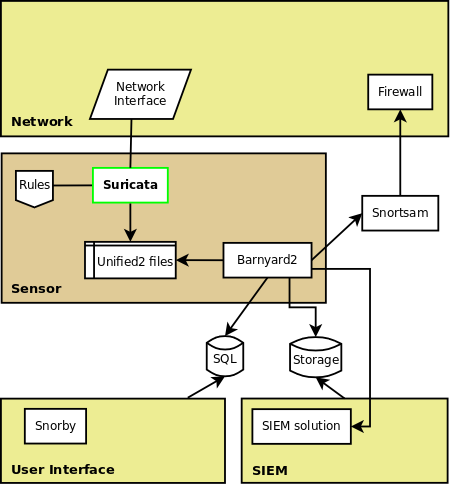
问题：git clone https:// fail

git config --global http.sslverify true

git config --global http.sslCAPath /etc/pki/tls/certs

## Suricata\*\*

与Snort相比，优势为多线程，可硬件加速，文件提取（获得黑客下载的挂马文件）



Barnyard2 读取二进制文件存入MySQL数据库中，Snorby显示数据界面

### 安装

https://redmine.openinfosecfoundation.org/projects/suricata/wiki/CentOS\_7\_Installation

sudo yum -y install wget gcc libpcap-devel pcre-devel libyaml-devel file-devel zlib-devel jansson-devel nss-devel libcap-ng-devel libnet-devel

sudo yum install -y epel-release

sudo yum install -y libnetfilter\_queue-devel

wget <http://www.openinfosecfoundation.org/download/suricata-2.0.8.tar.gz>

tar -xvzf suricata-2.0.8.tar.gz

cd suricata-2.0.8

./configure --enable-nfqueue --prefix=/usr --sysconfdir=/etc --localstatedir=/var

make

sudo make install

sudo ldconfig

make install-full

vi /etc/suricata/suricata.yaml 可根据需要修改配置

规则下载并更新

sudo suricata -c /etc/suricata/suricata.yaml -i eno16777736 --init-errors-fatal

如果失败，查看日志

cd /var/log/suricata

tail http.log

tail -n 50 stats.log

主配置文件suricata.yaml：

https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Suricatayaml

## OSSEC\*

<http://easysb.cn/?/article/1>

主机入侵检测系统，日志分析、文件完整性检查，策略监控，rootkit检测，实时报警。

### 安装

**Server端：**

/bin/systemctl stop firewalld.service

/bin/systemctl disable firewalld.service

yum install -y wget gcc make

wget -U ossec http://www.ossec.net/files/ossec-hids-2.8.1.tar.gz

tar zxvf ossec-hids-2.8.1.tar.gz

cd ossec-hids-2.8.1

安装Server

./install.sh

输入server

/var/ossec/bin/ossec-control start

ps aux | grep ossec

/var/ossec/bin/manage\_agents

填入Agent端IP，导出密钥

/var/ossec/bin/ossec-control stop

/var/ossec/bin/ossec-control start

**Agent端：**

**两种agents，installable agents安装在机器上，通过加密消息协议将报告返回server；agentless agents不需要在远程机器上安装，在OSSEC manager上通过RPC方法获取远程机器的信息。**

跟安装Server一样

./install.sh时输入agent，和Server IP

/var/ossec/bin/manage\_agents

导入Server端提取的密钥

/var/ossec/bin/ossec-control stop

/var/ossec/bin/ossec-control start

## Security Onion\*

网络安全监控

## OpenWIPS-NG\*

无线IDS

## Bro IDS\*

## Kismet

无线IDS标准

## Samhain

## OpenDLP

# 挂马检测

## nsdecoder

## iScanner

<https://sitecheck.sucuri.net//>

<http://urlquery.net>

<http://quttera.com/website-malware-scanner>

<http://www.google.com/safebrowsing/diagnostic?site=www.example.com>

<http://code.google.com/apis/safebrowsing/key_signup.html>

# 渗透测试

## Metasploit\*

渗透测试框架

### 安装

关闭防火墙和杀毒软件

systemctl stop firewalld.service

systemctl disable firewalld.service

yum install -y wget

wget <http://downloads.metasploit.com/data/releases/metasploit-latest-linux-x64-installer.run>

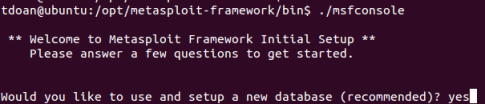
chmod +x metasploit-latest-linux-x64-installer.run

sudo ./metasploit-latest-linux-x64-installer.run

/opt/metasploit/createuser 设置一个初始化用户，记住用户名密码

<https://localhost:3790>

填入license key，非美国和加拿大用户收不到license key



Creating database at /Users/joesmith/.msf4/db

Starting Postgresql

Creating database users

Creating initial database schema

\*\* Metasploit Framework Initial Setup Complete \*\*

[\*] Starting the Metasploit Framework console...-[\*] The initial module cache will be built in the background, this can take 2-5 minutes...

/

Metasploit Park, System Security Interface

Version 4.0.5, Alpha E

Ready...

> access security

access: PERMISSION DENIED.

> access main security grid

access: PERMISSION DENIED....and...

YOU DIDN'T SAY THE MAGIC WORD!

YOU DIDN'T SAY THE MAGIC WORD!

=[ metasploit v4.11.0-dev [core:4.11.0.pre.dev api:1.0.0]]

+ -- --=[ 1454 exploits - 827 auxiliary - 229 post ]

+ -- --=[ 376 payloads - 37 encoders - 8 nops ]

+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >

检查数据库状态

$ db\_status

[\*] postgresql connected to msf

bundle install

./msfconsole

模块路径：

/path/to/metasploit/apps/pro/msf3/modules: binaries安装

/path/to/metasploit-framework-repo/modules: github安装

模块根据目的和操作分为几类：

* Exploit：针对目标主机执行一系列命令以找寻漏洞，利用漏洞进入主机，包括buffer overflow，code injection，web漏洞。
* Auxiliary：辅助模块不执行payload，包括scanner，fuzzers和拒绝服务攻击。
* Post-Exploitation：收集被攻击主机的信息，以便进一步行动，包括hash dumps，应用服务仿真。
* Payload：用于在攻陷主机后执行的脚本代码，开启Meterpreter或命令shell。Meterpreter运行你执行DLL文件。
* NOP generator：产生一系列随机字节以躲过IDS或IPS的特征分析。

msfdb init: 初始化数据库

msfdb reinit: 重新初始化数据库

msfdb delete: 删除数据库

msfdb start: 启动数据库

msfdb stop: 停止数据库

msfdb status: 显示数据库状态

# 蜜罐

## 分类

**高交互式蜜罐VS低交互式蜜罐**

低交互式蜜罐只是模拟出了真正操作系统的一部分，例如模拟一个FTP服务。虽然低交互式蜜罐容易建立和维护，但模拟可能不足以吸引攻击者，还可能导致攻击者绕过系统发起攻击，从而使蜜罐在这种情况下失效，如honeyd。

高交互式蜜罐是一部装有真正操作系统，并可完全被攻破的系统。与攻击者进行交互的是一部包含了完整服务的真实系统。用于网络安全的高交互式蜜罐提供了真实操作系统的服务和应用程序，使其可以获得关于攻击者更可靠的信息。但是部署和维护起来十分困难，而且被攻破的系统可能会被用来攻击互联网上其他的系统，这必须承担很高的风险，如Sebak。

中等交互蜜罐：与低交互类似，但会与木马交互下载可执行文件用于分析，如Nepenthes，Dionaea。

所有著名蜜罐列表：

**物理蜜罐VS虚拟蜜罐**

物理蜜罐是网络上一台拥有IP地址的真实机器，虚拟蜜罐是由一台机器模拟的，响应发送给虚拟蜜罐的网络流量。

<https://www.honeynet.org/project>

虚拟蜜罐：从僵尸网络追踪到入侵检测

## 蜜罐发展史

1. **欺骗工具包DTK**

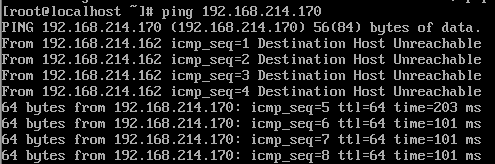
最早的低交互蜜罐，绑定在系统的未使用端口上,对任何想探测这些端口的攻击源提供欺骗性网络服务。

1. **LaBrea**

接受网络上所有空闲IP 地址的TCP 连接，并通过TCP 协议中的窗口调节与持久捕捉等技巧实现一种Tarpit 服务，能够尽可能地拖长无效连接的持续时间,从而减缓网络扫描探测与蠕虫传播的速度。

sudo apt-get install labrea

sudo labrea -v -i eth0 -sz -d -n 192.168.214.128/25



修改配置文件

sudo vi /etc/labrea/labrea.conf

192.168.214.188 exclude 排除188IP被捕获

192.168.214.192/26 exclude 排除26网段被捕获

0-20 portignore 排除0-20端口被捕获

1. **Tiny Honeypot**
2. 针对任何端口上的任何连接，提供一个登陆标识和一个超级用户外壳程序（root shell）。
3. 收集得到的所有数据。

thp使用xinetd绑定到一个端口，然后使用内置防火墙将到所有端口的流量重定向到该端口。

已经没有源码下载。

1. **Honeyd**

著名安全专家Provos开发的一款"虚拟蜜罐框架性开源软件"

1) 引入了在网络协议栈层次上模拟各种类型蜜罐系统的方法

2) Honeyd 支持在"协议栈指纹特征"上伪装成指定的操作系统版本,对攻击者利用nmap 等工具实施主动指纹识别进行欺骗

3) 支持模拟构建虚拟网络拓扑结构,并以插件方式提供对各种应用层网络服务的模拟响应

4) 利用Honeyd软件, 安全研究人员可以很容易地按照需求定制出一个包含指定操作系统类型与应用服务的蜜罐系统,用于蠕虫检测与应对、垃圾邮件监测等多种用途

5) 由于Honeyd 最早引入了网络协议栈层次上的蜜罐系统模拟机制,以及采用了可集成各种应用层服务蜜罐的灵活框架性结构,使其在蜜罐工具软件发展过程中具有举足轻重的重要地位

1. **Nepenthes蜜罐软件**

The Honeynet Project Giraffe Chapter 开发

1) 它继承了Honeyd 的网络协议栈模拟机制与框架性结构, 针对互联网上主动传播恶意代码的监测需求, 实现了可供大规模部署的恶意代码样本采集工具

2) 与之前蜜罐系统尝试模拟整个网络服务交互过程不同, Nepenthes 的基本设计原则是只模拟网络服务中存在安全漏洞的部分, 使用"Shellcode 启发式识别"与"仿真执行技术"来发现针

对网络服务安全漏洞的渗透攻击, 从中提取到主动传播恶意代码的下载链接,并进一步捕获样本

3) 这种机制使其较其他已有蜜罐工具对自动化传播恶意代码捕获更为高效

1. **Dionaea**

Nepenthes已被新一代恶意代码样本捕获蜜罐软件Dionaea所替代

1) Dionaea采用内嵌Python 脚本代码实现对漏洞服务的模拟

2) 同样采用Libemu来检测Shellcode

3) 并支持IPv6 与TLS 协议

4) Dionaea蜜罐软件是目前技术最为先进、体系结构最优化的虚拟蜜罐工具

1. **GHH(Google hack honeypot)**

1) 针对Web应用攻击威胁研究并开发的Web应用服务蜜罐

2) GHH针对搜索存有安全漏洞Web应用程序的Google Hacking技术来诱骗Web应用程序攻击并进行日志记录,可以发现命令注入、Web 垃圾邮件、博客垃圾评论注入、网页篡改、植入僵尸程序、搭建钓鱼站点等各种攻击事件

1. **HIHAT(high interaction honeypot analysistoolkit)**

1) 可将任意的PHP应用程序自动地转换为提供充分交互环境的Web蜜罐工具

2) 并通过"透明链接方式"获取恶意Web访问请求, 从而对现有PHP应用程序所面临的威胁进行监测分析

1. **Kojoney、Kippo蜜罐**

1) 模拟为SSH网络服务进程, 记录每次SSH口令暴力破解所尝试使用的用户名与口令

2) 并在口令猜测成功之后为攻击者提供模拟的shell 执行环境

3) 对攻击源IP 地址、使用的SSH 客户端类型、输入的控制命令以及下载的攻击工具文件进行捕获与记录.

1. **客户端蜜罐工具软件Capture-HPC**

1) 近年来,由于防火墙、入侵防御系统等网络边界防御机制的广泛应用,针对传统网络服务的渗透攻击变得越来越难以成功实现,以浏览器与插件为主要目标的客户端渗透攻击逐渐成为互联网上的主流安全威胁

2) 而蜜罐技术也随着安全威胁热点的这一变化,演化出客户端蜜罐工具软件.Capture-HPC

3) 这是一个高交互式的客户端蜜罐框架,支持在Windows 虚拟机环境中运行IE,Firefox 等浏览器,并通过内核中的系统状态变化监控机制来检测浏览器当前访问的网页中是否包含客户端渗透攻击代码

1. **PhoneyC**

1) 采用"浏览器仿真"与"Javascript动态分析技术"来对抗恶意网页脚本的混淆机制

2) 并通过模拟各种已知浏览器与插件安全漏洞来检测出恶意网页中包含的渗透攻击类型

3) 通过对Javascript引擎进行opcode指令动态插装,实现了对恶意网页中的heapspray 堆散射攻击的检测能力.

## kippo

**中等交互SSH蜜罐**，模拟SSH服务，记录口令猜测记录和攻击源IP，shell命令交互，Curl/wget真实下载。包括Kippo-Graph和Kippo2MySQL。

语言：Python

* 虚拟文件系统，可增删文件
* 包含虚拟文件，如/etc/passwd
* 以UML 兼容格式记录shell会话日志，可真实重现会话过程
* wget保存下载文件供分析
* 其他，如ssh假连接

### 安装

**CentOS 7：**

yum install python-devel python-zope-interface.

wget <http://pypi.python.org/packages/source/T/Twisted/Twisted-12.0.0.tar.bz2>

tar jxvf Twisted-12.0.0.tar.bz2

cd Twisted-12.0.0/

python setup.py build

python setup.py install

useradd kippo

su - kippo

git clone <https://github.com/desaster/kippo.git>

cd kippo

cp kippo.cfg.dist kippo.cfg

./start.sh

端口启动在2222，尝试ssh登录2222端口root/123456，执行操作

vi log/kippo.log 里面记录了登陆用户的所有操作

**Ubuntu 14：**

sudo apt-get update && sudo apt-get upgrade

sudo vi /etc/ssh/sshd\_config

# What ports, IPs and protocols we listen for

Port 2222

sudo reboot

sudo apt-get install python-dev openssl python-openssl python-pyasn1 python-twisted

sudo apt-get install git-core

sudo apt-get install authbind

sudo adduser kippo

sudo visudo

kippo ALL=(ALL:ALL) ALL

sudo touch /etc/authbind/byport/22

sudo chown kippo:kippo /etc/authbind/byport/22

sudo chmod 777 /etc/authbind/byport/22

su - kippo

git clone <https://github.com/desaster/kippo.git>

cd kippo/

mv kippo.cfg.dist kippo.cfg

vi kippo.cfg

ssh\_port = 22

vi start.sh

echo "Starting kippo in the background..."

authbind --deep twistd -y kippo.tac -l log/kippo.log --pidfile kippo.pid

./start.sh

sudo netstat -antp

**安装Kippo-Graph：**

sudo apt-get install -y libapache2-mod-php5 php5-mysql php5-gd php5-curl

sudo apt-get install -y python-mysqldb mysql-server libapache2-mod-php5

mysql -u root -p

CREATE DATABASE kippo;

GRANT ALL ON kippo.\* TO 'kippo'@'localhost' IDENTIFIED BY 'Kippo-DB-pass';

exit

mysql -u kippo -p

USE kippo;

source /home/kippo/kippo/doc/sql/mysql.sql;

exit

sudo vi kippo.cfg

[database\_mysql]

host = localhost

database = kippo

username = kippo

password = Kippo-DB-pass

port = 3306

ps aux | grep kippo

kill -9 xxxx

sudo /etc/init.d/apache2 restart

sudo update-rc.d -f apache2 remove

sudo wget <http://bruteforce.gr/wp-content/uploads/kippo-graph-1.4.2.tar.gz>

sudo mv kippo-graph-1.4.2.tar.gz /var/www

cd /var/www

sudo tar zxvf kippo-graph-1.4.2.tar.gz

sudo mv /var/www/html /var/www/html\_bak

sudo mv kippo-graph-1.4.2 html

cd kippo-graph/

sudo chmod 777 generated-graphs/

sudo cp config.php.dist config.php

* dl/ - 攻击者通过wget下载的文件在该目录中
* log/kippo.log - 调试日志，记录攻击者操作
* log/tty/ - 会话日志
* utils/playlog.py - 重现会话日志的工具
* utils/createfs.py - 创建fs.pickle文件
* fs.pickle - 假文件系统目录列表
* honeyfs/ - 假文件系统包含的文件，可覆盖为真实文件系统

## honeyd

官网：<http://www.honeyd.org/>

语言：C

Honeyd是一款非常优秀的开源虚拟蜜罐软件，由Google公司软件工程师Niels Provos于2003年开始研发。 **可在安装机器上虚拟数千台虚拟蜜罐，2007年后不再更新。**

它允许你在一个物理机器上建立上百个虚拟蜜罐。该虚拟蜜罐可以被配置为模仿许多不同的操作系统和服务的系统，允许你模拟任意的网络结构。

### 安装

yum remove libevent libevent-devel libdnet libdnet-devel

yum install -y gcc gcc-c++ flex yacc readline readline-devel zlib-devel

wget <http://monkey.org/~provos/libevent-1.4.14b-stable.tar.gz>

wget <http://sourceforge.net/projects/libdnet/files/libdnet/libdnet-1.11/libdnet-1.11.tar.gz>

wget <http://www.tcpdump.org/release/libpcap-1.1.1.tar.gz>

分别解压安装上面三个包

wget http://www.citi.umich.edu/u/provos/honeyd/arpd-0.2.tar.gz

tar zxvf arpd-0.2.tar.gz

cd arpd

vi arpd.c

添加#define \_\_FUNCTION\_\_ ""

vi configure

将所有/usr/lib/libc.so改为/usr/lib64/libc.so，/lib/libc.so改为/lib64/libc.so

ln -s /usr/local/lib/libevent-1.4.so.2.2.0 /usr/lib64/libevent-1.4.so.2

arpd -i eno1 192.168.0.0/16

**arpd对所有在192.168.x.x网段中未使用IP的ARP请求响应本机的MAC地址**

tar zxvf honeyd-1.5c.tar.gz

cd honeyd-1.5c

./configure

make

make install

mkdir /var/log/honeyd

chmod -R 777 /var/log/honeyd

Windows

192.168.1.3

CentOS

192.168.1.4

Honeyd

192.168.1.5

vi honeyd.conf

create default

set default default tcp action block

set default default udp action block

set default default icmp action block

create windows

set windows personality "Microsoft Windows XP Professional SP1"

set windows default tcp action reset

set windows default udp action reset

add windows tcp port 135 open

add windows tcp port 139 open

add windows tcp port 445 open

set windows ethernet "00:00:24:ab:8c:12"

bind 192.168.1.5 windows

honeyd -d -f honeyd.conf -l /var/log/honeyd/honeyd.log -s /var/log/honeyd/service.log -i eno1

在另一台机器上执行：

ping 192.168.1.5

nmap -A -T4 192.168.1.5

路由重定向，在路由器上执行

route -n add -net 192.168.1.128/25 192.168.1.4将该网段的数据包都发至192.168.1.4主机

arp -s 192.168.1.135 40:2c:f4:eb:b0:e5 将发到135的数据包转到指定主机上

### 命令

honeyd [-dP] [-l logfile] [-s servicelog] [-p fingerprints] [-0 p0f-file] [-x xprobe]  
                    [-a assoc] [-f file] [-i interface] [-u uid] [-g gid]  
                    [--webserver-address address] [--webserver-port port]  
                    [--webserver-root path] [--rrdtool-path path]  
                    [--disable-webserver] [--disable-update] [--verify-config]  
                    [--fix-webserver-permissions] [-V|--version] [-h|--help] [--include-dir]  
                    [--data-dir] [net ...]

-d：以deamon形式运行

-f configfile：后接配置文件

-l logfile：后接日志文件

-I interface：端口名称

-s servicefile：后接服务日志

net：指定IP或网段，如果没有指定，则监视全部IP

<http://blog.csdn.net/figo1986/article/details/7356978>

<http://travisaltman.com/honeypot-honeyd-tutorial-part-1-getting-started/>

### 配置文件

<http://www.honeyd.org/configuration.php>

<http://hynesim-org.diasite.fr/files/652/simulatingNetworksWithHoneyd_RoshenChandran.pdf>

#### default template

**如果没有为某个IP指定模版，则采用default模版，因此需要设置一个名为default的模板。**

### Default Template

create default

# Set default behavior，**如果没有设置某端口的tcp请求方法，则采用default tcp action reset**。

set default personality "Windows NT4 / Win95 / Win98"

set default default tcp action reset

set default default udp action reset

set default default icmp action open

# Add specific services

add default tcp port 139 open

add default tcp port 137 open

add default udp port 137 open

add default udp port 135 open

**端口行为：**

TCP（默认为Open）

* Open：建立连接，返回Syn/Ack响应；
* Block：丢包或者不响应；
* Reset：返回RST；
* Tarpit：持续连接

UDP（默认为Closed）

* Open：不响应；
* Block：丢包或不响应；
* Reset：返回ICMP端口错误消息；

ICMP（默认为Open）

* Open：响应ICMP包；
* Block：丢包或不响应。

#### dynamic template

根据条件采用不同模版响应请求。honeyd采用被动指纹发现攻击者的OS，从而返回不同操作系统。必须在定义完所有模板后定义动态模板。能被发现的OS在pf.os文件中。

### Dynamic honeypot

dynamic magichost

add magichost use sticky if source os = "windows" 根据操作系统

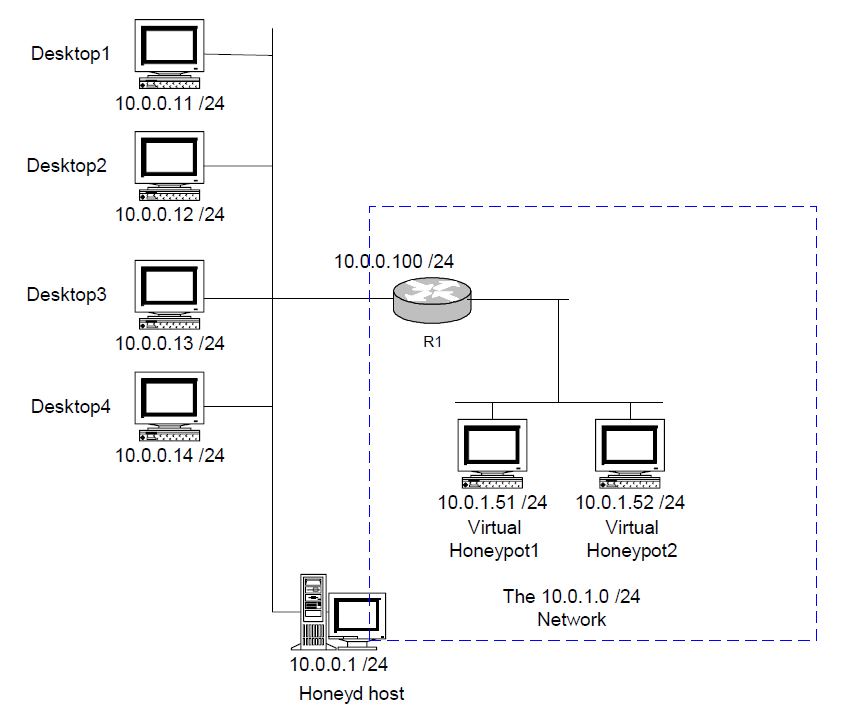
add magichost use suse70 if source ip = 192.168.1.0/28 根据源IP

add magichost use router if time between 12:00am - 5:00am 根据时间

add magichost otherwise use default

bind 192.168.1.100 magichost

#### network template

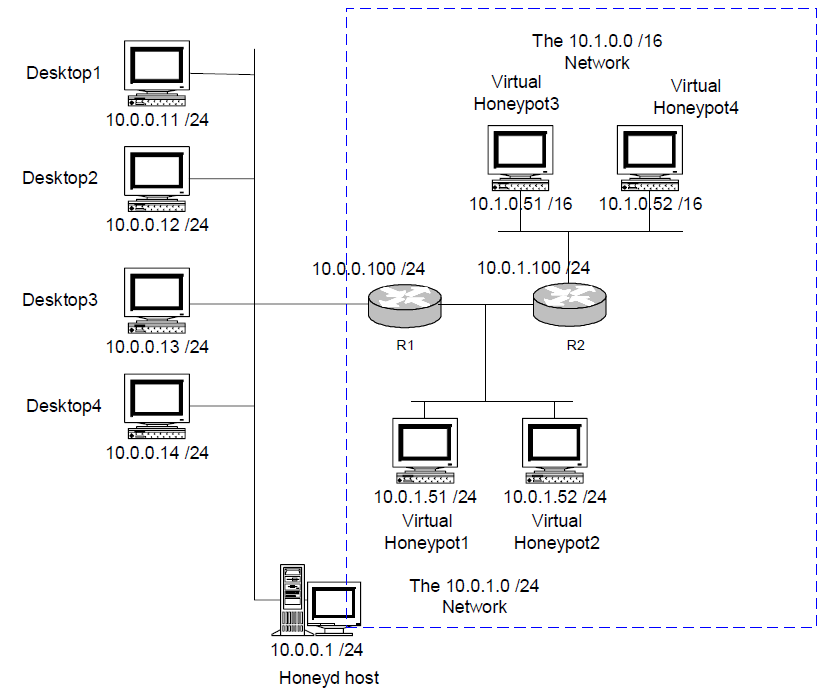


通过route entry创建接入路由器，通过IP为10.0.0.100的路由器，**访问**网段10.0.0.0/16

route entry 10.0.0.100 network 10.0.0.0/16

通过10.0.0.100路由器可**直接访问**网段10.0.1.0/24，**route link是直接访问网段，不需要下一跳**

route 10.0.0.100 link 10.0.1.0/24



增加连接到10.0.0.100路由器的新路由器10.0.1.100，通过新路由器可访问网段10.1.0.0/16

route 10.0.0.100 add net 10.1.0.0/16 10.0.1.100 latency 50ms loss 0.1 bandwidth 1Mbps

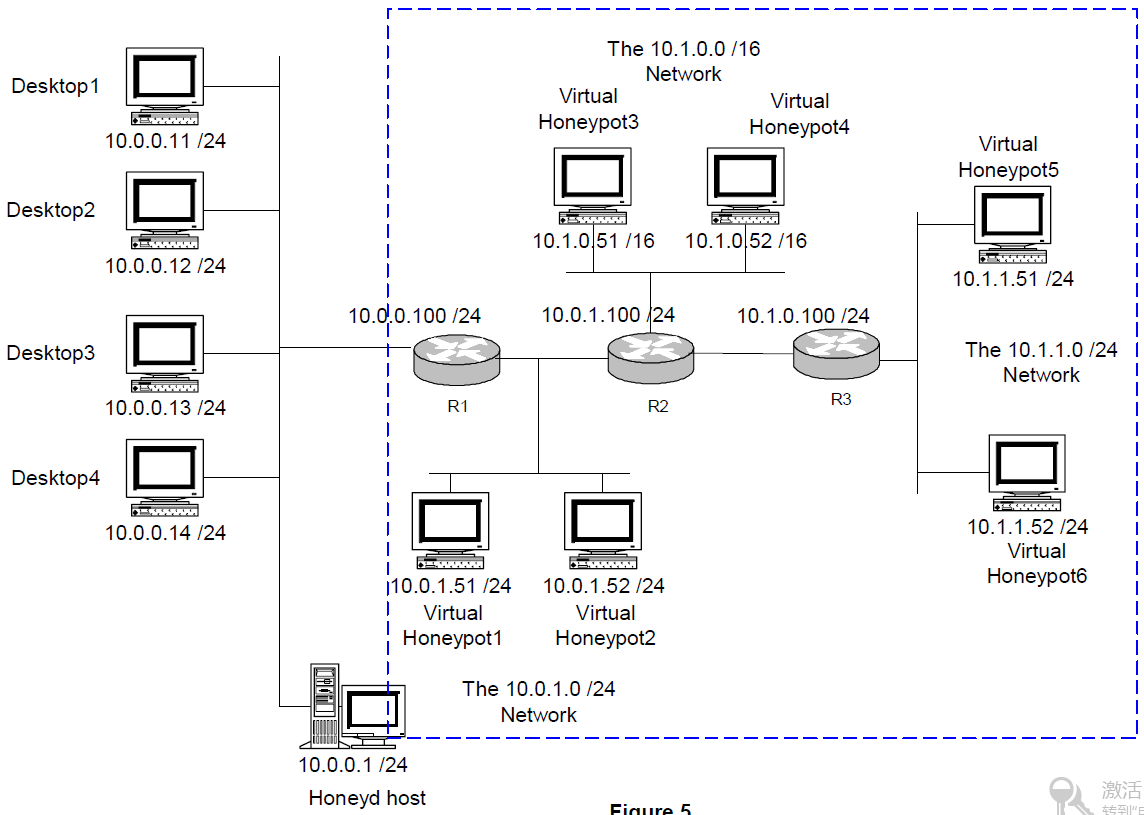
通过新路由器10.0.1.100可直接访问网段10.1.0.0/16

route 10.0.1.100 link 10.1.0.0/16

latency：每一跳延时，milliseconds

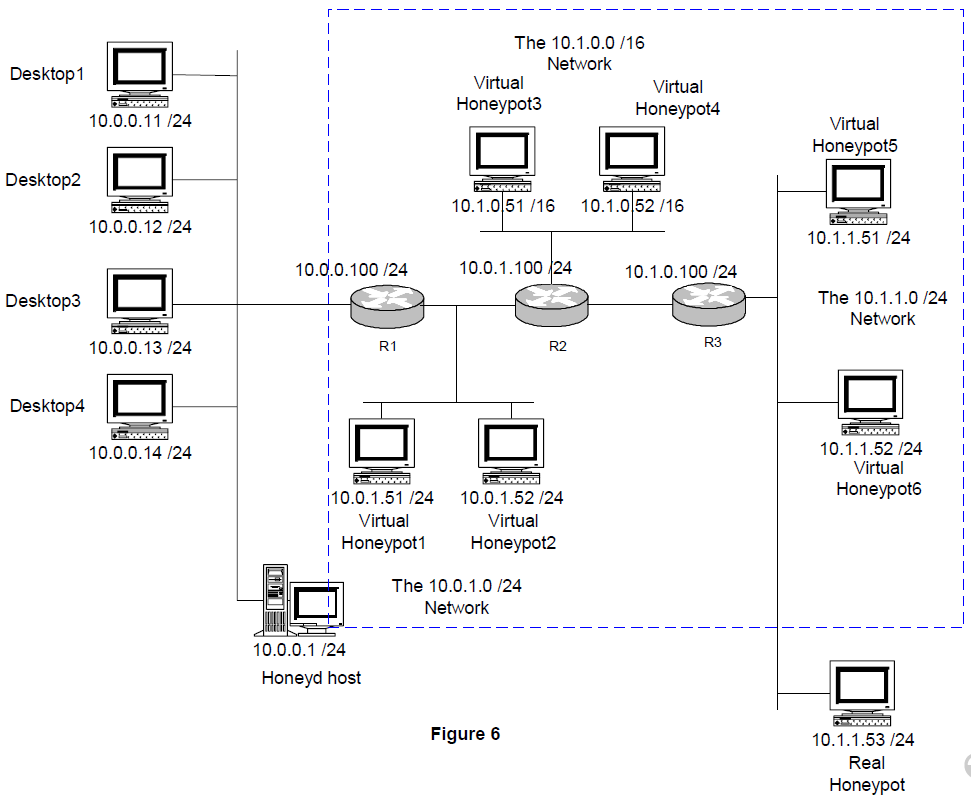
loss：丢包率，%。

bandwidth：排队时长，Kbps或Mbps或Gbps



route 10.0.1.100 add net 10.1.1.0/24 10.1.0.100

route 10.1.0.100 link 10.1.1.0/24



通过eth0可访问10.1.1.53实体蜜罐，根据网络拓扑其将被帮顶到Route 3上。

bind 10.1.1.53 to eth0

c:\> tracert 10.1.1.53 查看其跳数

#### GRE tunnel

从虚拟网络到远程网络的隧道传输数据。

route entry 172.20.254.1 network 10.3.2.0/24

route 172.20.254.1 link 10.3.2.0/24

route 172.20.254.1 add net 10.3.1.0/24 tunnel 172.20.254.1 172.30.254.1

### 其他

**所有可用OS在nmap.prints和xprobe2.conf**

**仅支持简单拓扑，对多个路由器架设的复杂网络，会导致主机无法访问，原因未知**

**脚本需增加读执行权限 chmod r+x scripts/xx.pl**

**服务脚本从网站下载，放到scripts目录下，配置在honeyd.conf中**

## Dionaea

官网：<http://dionaea.carnivore.it/>

语言：python

系统：Ubuntu 特定版本

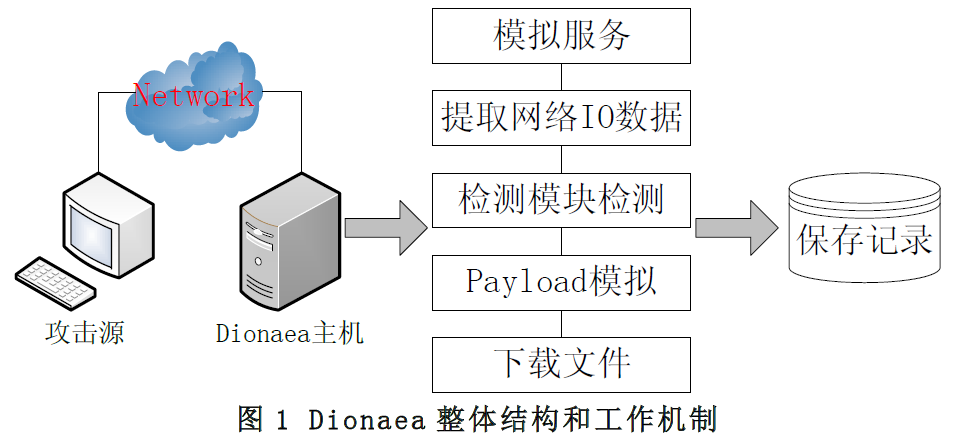
低交互蜜罐，收集漏洞和挂马，在本机虚拟蜜罐，**2013年后不再更新**

<http://www.freebuf.com/articles/system/12696.html>

<http://netsec.ccert.edu.cn/zhugejw/files/2011/09/Dionaea低交互式蜜罐介绍.pdf>

Dionaea是运行于Linux上的一个应用程序，将程序运行于网络环境下，它开放Internet上常见服务的默认端口，当有外来连接时，模拟正常服务给予反馈，同时记录下出入网络数据流。网络数据流经由检测模块检测后按类别进行处理，如果有 shellcode 则进行仿真执行；程序会自动下载 shellcode 中指定下载或后续攻击命令指定下载的恶意文件。从捕获数据到下载恶意文件，整个流程的信息都被保存到数据库中，留待分析或提交到第三方分析机构。

Dionaea 整体结构和工作机制如图：



### 源码安装

<http://wiki.khairulazam.net/index.php?title=Installing_Dionaea_honeypot_on_Ubuntu_Server_14.04>

**支持Ubuntu Server 或Desktop 14.04**

sudo apt-get install libudns-dev libglib2.0-dev libssl-dev libcurl4-openssl-dev libreadline-dev libsqlite3-dev python-dev libtool automake autoconf build-essential subversion

sudo mkdir /opt/dionaea

git clone git://git.carnivore.it/liblcfg.git liblcfg

cd liblcfg/code

sudo autoreconf -vi

sudo ./configure --prefix=/opt/dionaea

sudo make install

cd ..

cd ..

sudo apt-get install libemu-dev

sudo apt-get install libnl-3-dev libnl-genl-3-dev libnl-nf-3-dev libnl-route-3-dev

wget <http://dist.schmorp.de/libev/libev-4.19.tar.gz>

tar xfz libev-4.19.tar.gz

cd libev-4.19

sudo ./configure --prefix=/opt/dionaea

sudo make install

sudo ldconfig

cd ..

wget <http://www.python.org/ftp/python/3.2.2/Python-3.2.2.tgz>

tar xfz Python-3.2.2.tgz

cd Python-3.2.2/

sudo ./configure --enable-shared --prefix=/opt/dionaea --with-computed-gotos --enable-ipv6 LDFLAGS="-Wl,-rpath=/opt/dionaea/lib/ -L/usr/lib/x86\_64-linux-gnu/"

sudo make

sudo make install

sudo ldconfig

cd ..

wget <http://cython.org/release/Cython-0.21rc1.tar.gz>

tar xfz Cython-0.21rc1.tar.gz

cd Cython-0.21rc1

sudo /opt/dionaea/bin/python3 setup.py install

sudo ldconfig

cd ..

wget <http://www.tcpdump.org/release/libpcap-1.6.2.tar.gz>

tar xfz libpcap-1.6.2.tar.gz

cd libpcap-1.6.2

sudo ./configure --prefix=/opt/dionaea

sudo make

sudo make install

sudo ldconfig

cd ..

sudo apt-get install p0f -y

cd /

sudo mkdir nonexistent

sudo chown -R nobody:nogroup nonexistent

sudo mkdir /var/p0f

sudo p0f -i eth0:0 -u nobody -Q /tmp/p0f.sock -q -l -d -o /var/p0f/p0f.log

sudo chown nobody:nogroup /tmp/p0f.sock

sudo ps -ef | grep p0f

nobody 460 1 0 May17 ? 00:00:01 /usr/sbin/p0f -u nobody -i venet0:0 -Q /tmp/p0f.sock -q -l -d -o /var/p0f/p0f.log

git clone git://git.carnivore.it/dionaea.git dionaea

cd dionaea

sudo autoreconf -vi

vi configure 跳到13785行

LIB\_SSL\_LIBS="-L$ssl\_lib -lssl"

改为LIB\_SSL\_LIBS="-L$ssl\_lib -lssl -lcrypto"

sudo ./configure --with-lcfg-include=/opt/dionaea/include/ \

--with-lcfg-lib=/opt/dionaea/lib/ \

--with-python=/opt/dionaea/bin/python3.2 \

--with-cython-dir=/opt/dionaea/bin \

--with-udns-include=/opt/dionaea/include/ \

--with-udns-lib=/opt/dionaea/lib/ \

--with-emu-include=/opt/dionaea/include/ \

--with-emu-lib=/opt/dionaea/lib/ \

--with-gc-include=/usr/include/gc \

--with-ev-include=/opt/dionaea/include \

--with-ev-lib=/opt/dionaea/lib \

--with-nl-include=/opt/dionaea/include \

--with-nl-lib=/opt/dionaea/lib/ \

--with-curl-config=/usr/bin/ \

--with-pcap-include=/opt/dionaea/include \

--with-pcap-lib=/opt/dionaea/lib/ \

--with-ssl-lib=/usr/lib/x86\_64-linux-gnu/ \

--disable-werror

sudo make

sudo make install

sudo ldconfig

sudo chown -R nobody:nogroup /opt/dionaea/var/dionaea

sudo chown -R nobody:nogroup /opt/dionaea/var/log

sudo /opt/dionaea/bin/dionaea -u nobody -g nogroup -c /opt/dionaea/etc/dionaea/dionaea.conf -w /opt/dionaea -p /opt/dionaea/var/dionaea.pid -D

sudo ps -ef | grep dionaea

sudo netstat -tnlp | grep dionaea

ls /opt/dionaea/var/log

dionaea-errors.log

dionaea.log

ls /opt/dionaea/var/dionaea

binaries

bistreams

logsql.sqlite

sipaccounts.sqlite

vtcache.sqlite

wwwroot

**问题：libemu 安装报错：typedef 'PPEB\_LDR\_DATA' locally defined but not used**

gcc版本问题，从4.8切换回4.7

sudo apt-get install gcc-4.7 g++-4.7

sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.7 40

sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-4.7 40

gcc -v、g++ -v 命令分别查看gcc、g++当前使用的版本

**问题：Libtool library used but 'LIBTOOL' is undefined**

sudo apt-get install libtool

**问题：‘g\_thread\_init’ is deprecated**

2.32.0之后废弃了一些方法，需要从2.34.0降到2.30.0

sudo apt-get remove glib2.0-dev

sudo apt-get install libffi-dev

sudo wget https://launchpad.net/ubuntu/+archive/primary/+files/glib2.0\_2.30.0.orig.tar.bz2

sudo bzip2 -d glib2.0\_2.30.0.orig.tar.bz2

sudo tar xvf glib2.0\_2.30.0.orig.tar

cd glib-2.30.0/

sudo ./configure

sudo make

sudo make install

**问题：ssl --- configure:error**

sudo apt-get install libcurl4-openssl-dev libssl-dev openssl

ac\_cv\_lib\_ssl\_SSL\_connect"

### 编译安装(推荐)

**注意：Ubuntu版本为14.04**

<https://www.dinotools.de/en/2014/08/22/install-dionaea-on-ubuntu-14-04/>

sudo apt-get update

sudo apt-get dist-upgrade

sudo apt-get install software-properties-common

sudo add-apt-repository ppa:honeynet/nightly

sudo apt-get update

sudo apt-get install dionaea-phibo

sudo mv /etc/dionaea/dionaea.conf.dist /etc/dionaea/dionaea.conf

sudo service dionaea-phibo start

ps aux | grep dionaea

dionaea 28835 34.9 1.3 50892 18592 ? Ssl 10:14 0:12 /usr/bin/dionaea -u dionaea -g dionaea -c /etc/dionaea/dionaea.conf -w /var/lib/dionaea -p /var/run/dionaea.pid -D

nmap -A -T4 192.168.1.5

/etc/dionaea/dionaea.conf：核心配置文件

/var/log/dionaea：日志文件

/var/lib/dionaea：其他输出

安装DionaeaFR用于前端web展示Dionaea的数据。

<http://bruteforce.gr/visualizing-dionaeas-results-with-dionaeafr.html>

唯一需要注意的是NodeJS下载node-v0.10.33.tar.gz版本

**问题：dionaea is not running,** **Import failed dionaea.services**

通过输出详细日志，知道找不到数据库文件sipaccounts.sqlite

sudo dionaea -l all,-debug -L '\*' -u dionaea -g dionaea -c /etc/dionaea/dionaea.conf

如果源码编译安装后，在/opt/dionaea/var/dionaea目录下有该文件，拷贝至/var/lib/dionaea

sudo cp /opt/dionaea/var/dionaea/\* /var/lib/dionaea

sudo vi /etc/dionaea/dionaea.conf 修改路径采用绝对路径

users = "/opt/dionaea/var/dionaea/sipaccounts.sqlite"

**问题：/var/run/dionaeafr/dionaeafr.pid not found**

没有生成目录/var/run/dionaeafr/dionaeafr，导致pid文件无法写入

mkdir /var/run/dionaeafr

问题：找不到文件logsql.sqlite，导致DionaeaFR无法读到

vi DionaeaFR/setting.py

修改数据库文件指向/opt/dionaea/var/run/dionaea/logsql.sqlite

sudo /opt/dionaea/bin/dionaea -u nobody -g nogroup -c /opt/dionaea/etc/dionaea/dionaea.conf -w /opt/dionaea -D

更换python方法：

sudo update-alternatives --install /usr/bin/python python /usr/bin/python3.1 1

### 命令

dionaea [-c, --config=FILE] [-D, --daemonize] [-g, --group=GROUP]

            [-G, --garbage=[collect|debug]] [-h, --help] [-H, --large-help]  
            [-l, --log-levels=WHAT] [-L, --log-domains=WHAT] [-u, --user=USER]  
            [-p, --pid-file=FILE] [-r, --chroot=DIR] [-V, --version] [-w, --workingdir=DIR]

选项的意义分别是：

-c,--config=File：配置文件，默认为/opt/dionaea/etc/dionaea.conf。

-D,--daemonize：以deamon形式运行。

-g,--group=GROUP：启动后切换到某个用户组，默认为保持当前组。

-G , --garbage=[collect|debug]：收集垃圾数据，用于调试内存泄露。不能用于 valgrind 软件。

-h：帮助信息。

-H：帮助信息，包括默认值信息。

-l：选择事件记录级别，可选 all, debug, info, message, warning, critical, error，多选使用“,”做分隔，排除使用“-”。

-L：选择域，支持通配符“\*”和“?”，多选使用“,”，排除使用“-”。

-u：启动后切换到某个用户，默认保持当前用户。

-p：记录 pid 到指定文件。

-r：指定启动后切换根目录到指定目录，默认下不切换。

-V：显示版本信息。

-w：设定进程工作目录，默认下为/opt/dionaea。

例子:

cd /opt/dionaea/bin

# ./dionaea -l all,-debug -L '\*'

# ./dionaea -l all,-debug -L 'con\*,py\*'

# dionaea -u dionaea -g dionaea -r /opt/dionaea/ -w /opt/dionaea -p /opt/dionaea/var/dionaea.pid

### 配置文件

dionaea.conf主配置文件，包含以下几部分：

* logging：控制日志，如轮转周期，日志等级。
* processors：受攻击时的双向流操作，默认运行emu processor检查shellcode
* downloads：保存下载挂马的位置。
* bistreams：保存数据流位置。
* submit：通过http和ftp发送文件的位置。
* listen：指定监听的IP和接口，默认all。一共有三种模式：
  + getifaddrs：对所有的ip上都开启相关端口的监听,127.0.0.1也会监听,默认的选项
  + manua：只在指定ip上开启端口监听
  + nl：指定监听的接口
* modules：配置各种模块的工作参数.部分”必须模块”: curl、libemu、pcap模拟的服务等信息
  + pcap:用来拒绝连接的请求.
  + curl:用来传输文件,下载文件或者向第三方站点提交样本
  + emu:用来检测或者模拟shellcode执行
  + python:被dionaea调用，来模拟服务

## Glastopf

**低交互蜜罐**，**模拟包含各种漏洞的Web应用**，根据攻击者发送的请求，检查是否存在诸如=http://或CAST(0x字符串，下载并分析文件，返回尽量符合攻击者预期的结果，以诱导攻击者进一步行动。

语言：Python

系统：支持多种OS，这里采用Ubuntu 14.xx

https://honeynet.org/files/KYT-Glastopf-Final\_v1.pdf

### 安装

sudo apt-get update

sudo apt-get install python2.7 python-openssl python-gevent libevent-dev python2.7-dev build-essential make

sudo apt-get install python-chardet python-requests python-sqlalchemy python-lxml

sudo apt-get install python-beautifulsoup mongodb python-pip python-dev python-setuptools

sudo apt-get install g++ git php5 php5-dev liblapack-dev gfortran libmysqlclient-dev

sudo apt-get install libxml2-dev libxslt-dev

sudo pip install --upgrade distribute

cd /opt

sudo git clone git://github.com/glastopf/BFR.git

cd BFR

sudo phpize

sudo ./configure --enable-bfr

sudo make

sudo make install

ls /usr/lib/php5/20121212/bfr.so

sudo echo "zend\_extension = /usr/lib/php5/20121212/bfr.so" >> /etc/php5/cli/php.ini

cd /opt

sudo git clone https://github.com/glastopf/glastopf.git

cd glastopf

sudo python setup.py install

cd /opt

sudo mkdir myhoneypot

cd myhoneypot

sudo glastopf-runner

访问<http://ip>

**安装MySQL代替sqlite：**

sudo apt-get install mysql-server python-mysqldb

mysql -u root -p

mysql> create database glaspot;

mysql> create user 'glaspot'@'localhost' identified by 'glaspot';

mysql> grant all privileges on glaspot.\* to 'glaspot'@'localhost';

mysql> flush privileges;

mysql> exit

sudo vi /opt/myhoneypot/glastopf.cfg

[main-database]

connection\_string = mysql://glaspot:glaspot@localhost/glaspot

**问题：NameError: name 'sys\_platform' is not defined**

sudo rm -rf /usr/local/lib/python2.7/dist-packages/distribute-0.7.3-py2.7.egg-info/

sudo rm -rf /usr/local/lib/python2.7/dist-packages/setuptools\*

sudo apt-get remove python-pip

sudo wget <https://pypi.python.org/packages/source/d/distribute/distribute-0.6.35.tar.gz>

sudo tar zxvf distribute-0.6.35.tar.gz

cd distribute-0.6.35/

sudo python setup.py install

ls /usr/local/lib/python2.7/dist-packages/

**问题：Could not find any downloads that satisfy the requirement hpfeeds**

不要用sudo pip install glastopf安装，用源码安装

**问题：FeedException while connecting: Could not connect to broker**

端口20000被屏蔽了nc hpfriends.honeycloud.net 20000

## Wordpot

**模拟wordpress蜜罐，功能太简单，日志没有保存到数据库中**

语言：python

### 安装

cd /opt

sudo git clone <https://github.com/gbrindisi/wordpot.git>

cd wordpot/

sudo pip install flask

sudo python wordpot.py

sudo vi wordpot.conf

可修改Host，port，theme等参数

或者通过命令行修改，如python wordpot --theme=THEMENAME

**修改theme**

theme内容保存在static/wp-content/themes/以themename命名的目录下

ls /opt/wordpot/wordpot/static/wp-content/themes/twentyeleven

theme模版保存在templates/目录下：

ls /opt/wordpot/wordpot/templates/twentyeleven.html

python wordpot --theme= twentyeleven

## GHH

http://sourceforge.net/projects/ghh/?source=directory

## PhoneyC

虚拟蜜罐客户端，

语言：python

## Sebak

安装在高交互蜜罐上的内核模块，用于攻击数据采集，例如系统按键活动。

系统：Win32和Linux

## HoneyDrive

预装了蜜罐的Linux镜像，采用Xubuntu Desktop 12.04.4 LTS版本，10个预装的蜜罐软件，如Kippo，Dionaea，Amun，Honeyd，Glastopf，Wordpot，Conpot，Thug和PhoneyC。包含预设脚本分析查看获取到的数据，如Kippo-Graph，Honeyd-Viz，DionaeaFR，ELK stack。

**只支持VirutalBox镜像，版本较老，不建议采用**

# 安全可视化

数据可视化是科学、艺术和设计的结合，当枯燥隐晦的数据被数据科学家们以优雅、简明、直观的视觉方式呈现时，带给人们的不仅仅是一种全新的观察世界的方法，而且往往具备艺术作品般的强大冲击力和说服力。

在信息安全领域，随着越来越多的CISO们开始依靠数据分析来从海量数据中发现新的安全威胁，并且越来越多的企业IT部门开始利用安全分析技术，数据可视化已经成为网络安全技术的一个重要趋势。

<http://secviz.org/>

## 安全可视化公司

### Pixcloud

[Pixlcloud](http://pixlcloud.com/)下一代可视化分析平台，CEO Raffael Marty

## 数据源

数据源主要包括：

* Perimeter threat use-cases
* Network flow data
* Firewall data
* IDS/IPS data
* Proxy data
* User activity
* Host-based data analysis

获取数据的方式包括：

* tcpdump and wireshark to analyze packet captures
* argus, nfdump, nfsen, and silk to process traffic flows
* snort, bro, suricata as intrusion detection systems
* p0f, npad for passive network analysis
* iptables, pf, pix as examples of firewalls

### 抓包

- [Source Destination Confusion](http://raffy.ch/blog/2013/10/22/cleaning-up-network-traffic-logs-vast-2013-challenge/)

- [Packet Sniffing Cheat Sheet](http://www.scribd.com/doc/79439036/Packet-Sniff-Craft-Cheatsheet#scribd)

- [PCAPNG](http://netresec.com/?b=12CB2A2)

- [Wireshark](http://www.wireshark.org/)

- [Network Miner](http://www.netresec.com/?page=NetworkMiner)

- [Scapy](http://www.secdev.org/projects/scapy/)

- [Scapy Tutorial](http://media.packetlife.net/media/library/36/scapy.pdf)

- [CloudShark](http://www.cloudshark.com)

argus, nfdump, nfsen, silk

### 流量

- [Argus](http://www.qosient.com/argus/)

- [Argus cool stuff](http://nsmwiki.org/Argus)

- [NFDump](http://nfdump.sourceforge.net/)

- [NFSen](http://nfsen.sourceforge.net/)

- [Nfsight](http://sourceforge.net/p/nfsight/home/Nfsight/)

- [SiLK](http://tools.netsa.cert.org/silk/docs.html)

- [Rayon](http://tools.netsa.cert.org/rayon/index.html)

- [Rayon Presentation](http://secviz.org/content/www.cert.org/flocon/2011/presentations/Groce_Rayon.pdf)

- [FlowViewer](https://sourceforge.net/projects/flowviewer/)

### 威胁智能感知系统

- [STIX](https://stix.mitre.org)

- [TAXII](http://taxii.mitre.org/)

- [Cybox](https://cybox.mitre.org)

- [OpenIOC](http://openioc.org)

- [Emerging Threats](http://rules.emergingthreats.net/)

- [ENISA Report on Threat Intelligence](http://bit.ly/enisareport)

- [Collective Intelligence Framework (CIF)](https://code.google.com/p/collective-intelligence-framework/)

- [ThreatIntelligence](https://github.com/syphon1c/Threatelligence)

### 入侵检测

- [Fragroute](http://monkey.org/~dugsong/fragroute/)

- Snort、[Snort tips](http://doc.emergingthreats.net/bin/view/Main/SnortSigs101)、[More Snort tips](http://doc.emergingthreats.net/bin/view/Main/WhatEverySnortUserShouldDo)、[Snorby](http://snorby.org)

- [Packet PIG](https://github.com/packetloop/packetpig)

- [Suricata](https://suricata-ids.org)

- [Bro](http://www.bro.org)、[Bro intro presentation](https://github.com/LiamRandall/bro-scripts/blob/master/talks-and-training/Bro_IDS-and-the-Bro-Network-Programming-Language-Liam_Randall-Hectaman-Shmoocon-2013.pdf)、[Bro intro video](http://www.youtube.com/watch?v=7DCPuHdCbpw)、[Bro APT1](https://github.com/sethhall/bro-apt1)

### 被动监听

- [P0f](http://lcamtuf.coredump.cx/p0f.shtml%20)

- [P0f 3](http://lcamtuf.coredump.cx/p0f3)

- [Passive DNS](https://github.com/gamelinux/passivedns)

- [PRADS](http://gamelinux.github.io/prads/)

- [prads2snort](https://github.com/gamelinux/prads/blob/master/tools/prads2snort)

防火墙：iptables, pf, pix

### 主机数据

- [CollectD](http://collectd.org)

- [HekaD](http://blog.mozilla.org/services/2013/04/30/introducing-hek)

- [OSSEC](http://www.ossec.net/)

- [Graphite](http://graphite.wikidot.com/)

- [StatsD](https://github.com/etsy/statsd/)

- [Munin](http://munin-monitoring.org/)

- [Ganglia](http://ganglia.sourceforge.net/)

- [RRD](http://oss.oetiker.ch/rrdtool/)

- [Logster](https://github.com/etsy/logster)

## 数据处理

- [CSVKit](http://csvkit.readthedocs.org/en/latest/) ：对CVS文件进行SQL查询

- [LogParser](http://www.microsoft.com/en-us/download/details.aspx?id=24659)

- [CommandlineFu](http://commandlinefu.com)

- [Regex Lib](http://regexlib.com)

- [Regular Expressio Information](http://www.regular-expressions.info)

- [Regex One](http://regexone.com)

- [RegExr](http://gskinner.com/RegExr)

- [Geo Lookup On The Commandline](http://raffy.ch/blog/2007/02/24/geo-lookup-on-the-command-line/)

- [Log Analysis Scripts](https://github.com/zrlram/loganalysis)

- [LogParser Studio](http://gallery.technet.microsoft.com/office/Log-Parser-Studio-cd458765)

- [httpry](http://dumpsterventures.com/jason/httpry/)

- [dnstop](http://dns.measurement-factory.com/tools/dnstop/)

- [Emerging Threats](http://rules.emergingthreats.net/open)

- [HoneySnap](https://projects.honeynet.org/honeysnap/)

## 日志管理

* rsyslog, syslog-ng, nxlog
* logstash, graylog
* commercial log management and SIEM solutions

- [Snare](http://www.intersectalliance.com/projects/SnareWindows/index.html)

- [NXLog](http://nxlog-ce.sourceforge.net)

- [rsyslog](http://rsyslog.com)

- [syslog-ng](http://www.balabit.com/network-security/syslog-ng)

- [rsyslog configuration wizzard](http://www.rsyslog.com/rsyslog-configuration-builder/)

- [swatch](http://sourceforge.net/projects/swatch/)

- [tenshi](http://www.inversepath.com/tenshi.html)

- [ossec](http://www.ossec.net/)

- [ossim](http://ossim.org)

- [lasso](http://open.loglogic.com/lasso)

- [logsurfer](http://www.crypt.gen.nz/logsurfer/)

- [loghound](http://ristov.users.sourceforge.net/loghound/)

- [Simple LogFile Clustering Tool](http://ristov.users.sourceforge.net/slct/)

- [logzilla](http://sourceforge.net/projects/php-syslog-ng/)

- [MS LogParser](http://technet.microsoft.com/en-us/scriptcenter/dd919274.aspx)

- [Octopussy](http://8pussy.org/)

- [Sagan](http://sagan.quadrantsec.com/)

- [ELSA](https://code.google.com/p/enterprise-log-search-and-archive/)

- [Security Onion](http://securityonion.blogspot.com/)

**-** [**LogStash**](http://logstash.net)

- [LogStash book](http://www.logstashbook.com)

- [Graylog2](http://www.graylog2.org)

- [Sguil](http://sguil.sourceforge.net/)

- [Squert](http://www.squertproject.org/)

**-** [**Moloch**](https://github.com/aol/moloch )

- [SEC](http://simple-evcorr.sourceforge.net/)

- [SEC Article](http://ristov.users.sourceforge.net/publications/sec-issa2012.pdf)

## 大数据

* hadoop (HDFS, map-reduce, HBase, Hive, Impala, Zookeper)
* search engines like: elastic search, Solr
* key-value stores like MongoDB, Cassandra, etc.
* OLAP and OLTP

- [Big Data Landscape](http://www.bigdatalandscape.com/)

- [Geneology of Databases](http://www.hpi.uni-potsdam.de/fileadmin/hpi/FG_Naumann/projekte/RDBMSGenealogy/RDBMS_Genealogy_V3_print.pdf)

- [Database Landscape 451 Group](https://blogs.the451group.com/information_management/files/2012/11/DBlandscape.jpg)

- [Apache Sentry](https://sentry.incubator.apache.org/)

**-** [**ElasticSearch**](http://elasticsearch.org)

- [BigDesk - ElasticSearch Monitoring](http://bigdesk.org/)

- [Berkeley Data Analysis Stack (BDAS)](https://amplab.cs.berkeley.edu/software)

**-** [**Spark**](http://spark.apache.org)

- [Spark MLLib](https://spark.apache.org/mllib/)

- [Twitter Storm](https://storm.incubator.apache.org)

- [Moloch](https://github.com/aol/moloch)

- [R Project](http://www.r-project.org)

- [R Reference Card](http://cran.r-project.org/doc/contrib/Short-refcard.pd)

- [R Big Data](http://r-forge.r-project.org/projects/rhadoop)

- [R Studio](http://rstudio.co)

- [Log Analysis with R](http://www.r-bloggers.com/log-file-analysis-with-r)

- [ggplot2](http://had.co.nz/ggplot2)

- [Pandas](http://pandas.pydata.org/)

- [sci-kit learn](http://scikit-learn.org/stable/)

- [NumPy](http://www.numpy.org/)

Lucene

## 数据可视化工具

* graphviz, tulip, cytoscape, and gephi
* afterglow
* treemap
* mondrian, ggobi

**-** [**AfterGlow**](http://afterglow.sf.net)

- [AfterGlow Cloud](http://afterglow.secviz.org)

- [GraphViz Workshop](http://graphviz-dev.appspot.com/)

- [Open Refine](http://openrefine.org/)

**-** [**Treemap**](http://www.cs.umd.edu/hcil/treemap)

- [**PicViz**](http://www.wallinfire.net/picviz/)

- [Logster](https://www.clarifiednetworks.com/logster)

**-** [**Gephi**](http://gephi.org)

- [GraphInsight](http://www.graphinsight.com/)

- [Cytoscape](http://www.cytoscape.org/)

**-** [**Mondrian**](http://www.theusrus.de/Mondrian/)

- [GGobi](http://www.ggobi.org/)

- [TimeFlow](https://github.com/FlowingMedia/TimeFlow/wiki)

- [R Cheat Sheet](http://cran.r-project.org/doc/contrib/Short-refcard.pdf)

- [ggplot2](http://had.co.nz/ggplot2/)

**-** [**Packetpig**](https://github.com/bigsnarfdude/packetpig)

Graphiti

tulip

### DAVIX

<http://www.secviz.org/node/89>

DAVIX用于数据分析可视化的CD，部署了最主流的可视化工具。

DAVIX 2014提供基于Ubuntu Server 13.10的镜像，可直接在VMPlayer(5.0版本以上)或VMWare Workstation（9.0版本以上）运行。或者利用[脚本](https://github.com/secviz/davix/)在全新的Ubuntu13.10上安装。

其包含的可视化工具列表如下：

flowtag

[Gephi](http://gephi.org)

Google Earth

ELK Stack

PicViz

Update to AfterGlow (1.6.5)

dns-browse

iPython

netsed

nsm-console

PRADS

R Studio

rsyslog

tcpstat

TeleTrafficTapper

TreeMap

Mondrian

Parvis

LGL

R Project

InteVis

EtherApe

GGobi

Tulip

Cytoscape

GUESS

RT3DG

Walrus

Dottty and lneato

GraphViz

glTail

Shoki

TimeSearcher

TNV

NVisionIP

Rumint

MRTG/RRD

GnuPlot

Ploticus

## 数据可视化库

- [Processing](http://processingjs.org/)

- [Javascript Tag Clouds](https://github.com/stef/timecloud/wiki)

- [Javascript Sparklines](http://omnipotent.net/jquery.sparkline/)

- [Protovis](http://vis.stanford.edu/protovis/)

- [D3](http://mbostock.github.com/d3/)

- [OpenGraphiti](http://www.opengraphiti.com/)

- [TheJit](http://thejit.org/)

- [HighCharts](http://www.highcharts.com/)

- [Geckoboard](http://geckoboard.com)

- [Smoothie Charts](http://smoothiecharts.org/)

- [Dotspotting](http://dotspotting.org)

- [Google Fusion Tables](http://www.google.com/fusiontables)

- [Polymaps](http://polymaps.org/)

- [MatPlotLib](http://matplotlib.org/)

- [PhiloGL](http://www.senchalabs.org/philogl/)

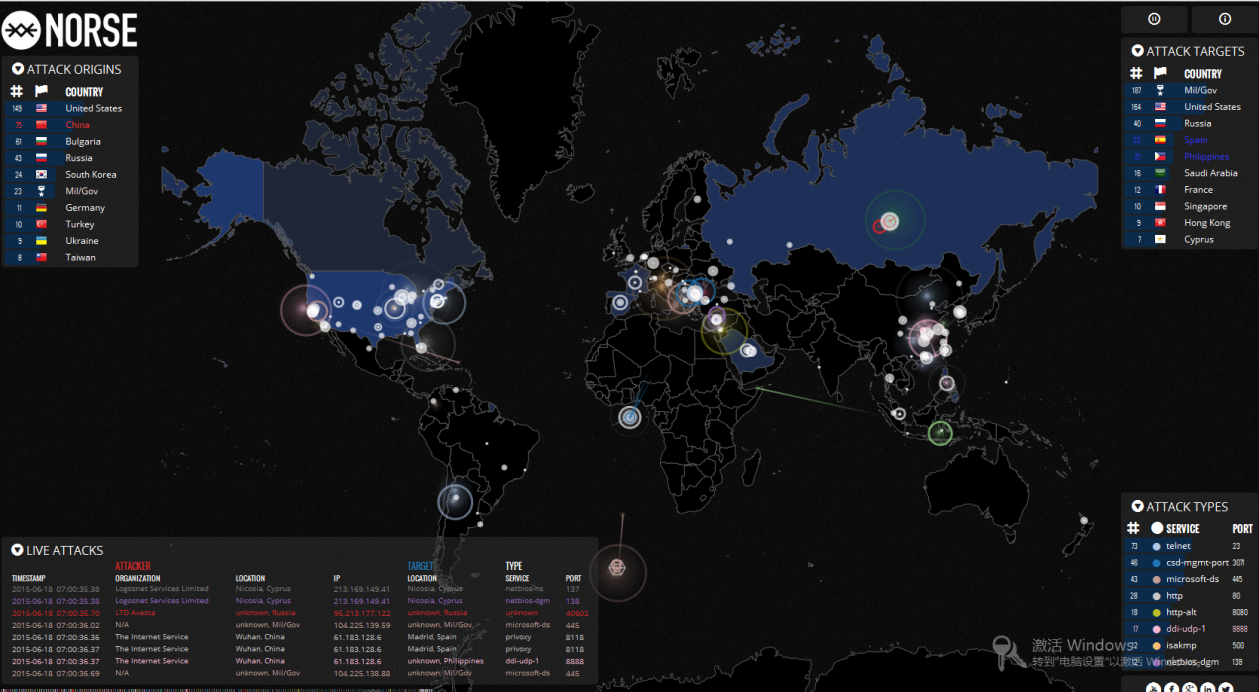
**KeyLines：网络可视化技术**

**WebGL：Web 3D技术，可用于地图显示**

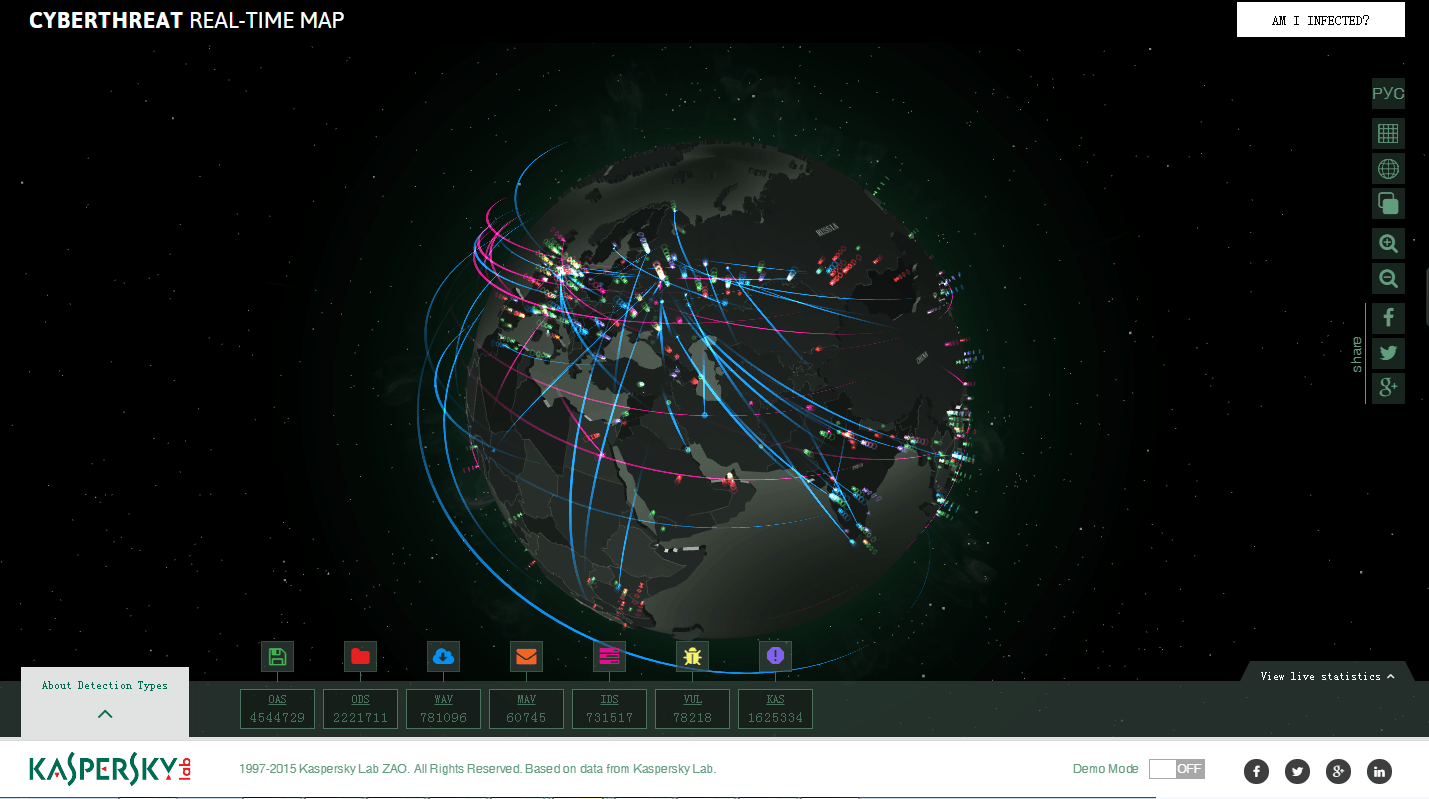
## 数据可视化方法

常用的八种数据可视化方法：<http://www.aqniu.com/neotech/1293.html>

### 地理信息可视化



[Arbor网络的DDoS攻击地图](http://www.arbornetworks.com/asert/map/)，通过与Google Ideas的合作，以可视化互动的方式展示DDoS攻击的历史和趋势（数据来自Arbor网络的ATLAS全球威胁监控系统）。该地图的本质是一套反映全球范围内黑客攻击频率的快照。它利用Norse 的“蜜罐”攻击陷阱显示出所有实时渗透攻击活动。

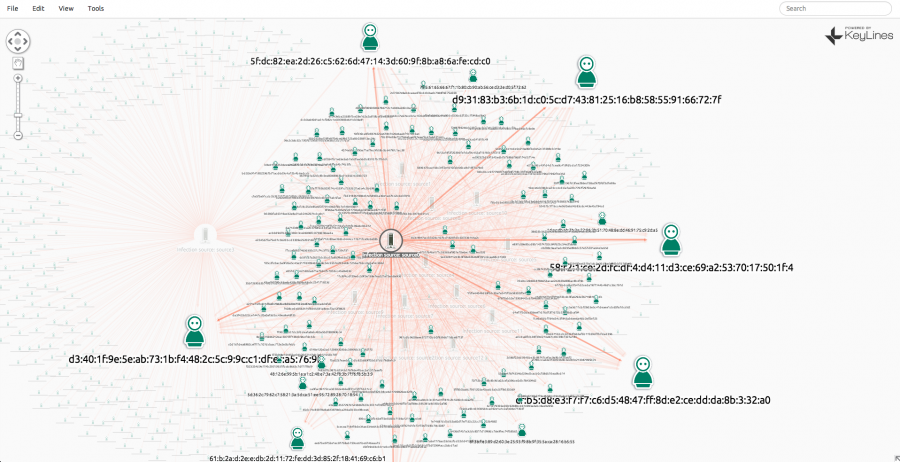


卡巴斯基实验室发布了一款可以实时呈现出全球网络安全事件的[交互式网络威胁地图](http://cybermap.kaspersky.com/)。这些网络威胁包括经实时扫描和手动扫描检测出的恶意对象、电子邮件和网页病毒、经漏洞和入侵检测组件发现的对象。

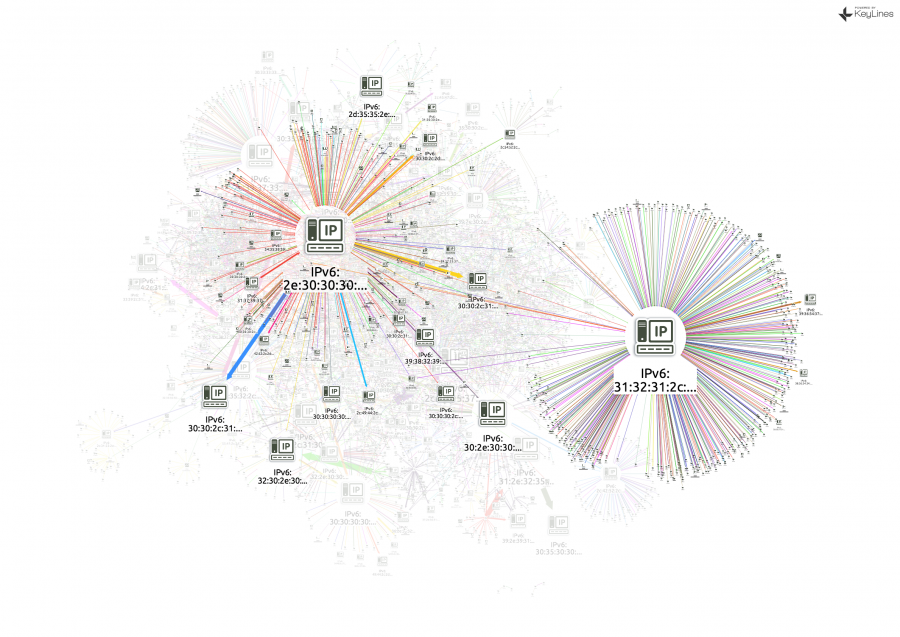
<http://www.aqniu.com/tools/security-tools/2511.html>

### 层级树状图（Hierarchical Tree Map）

### 关系图 （Link Charts）

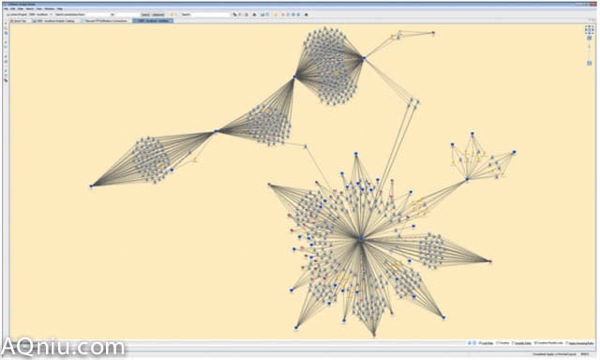


该关系图显示网络中的木马传播途径，7800多台机器的网络中，首先12台机器感染，黄色代表正常流量，红色代表只少有一个受感染的包。



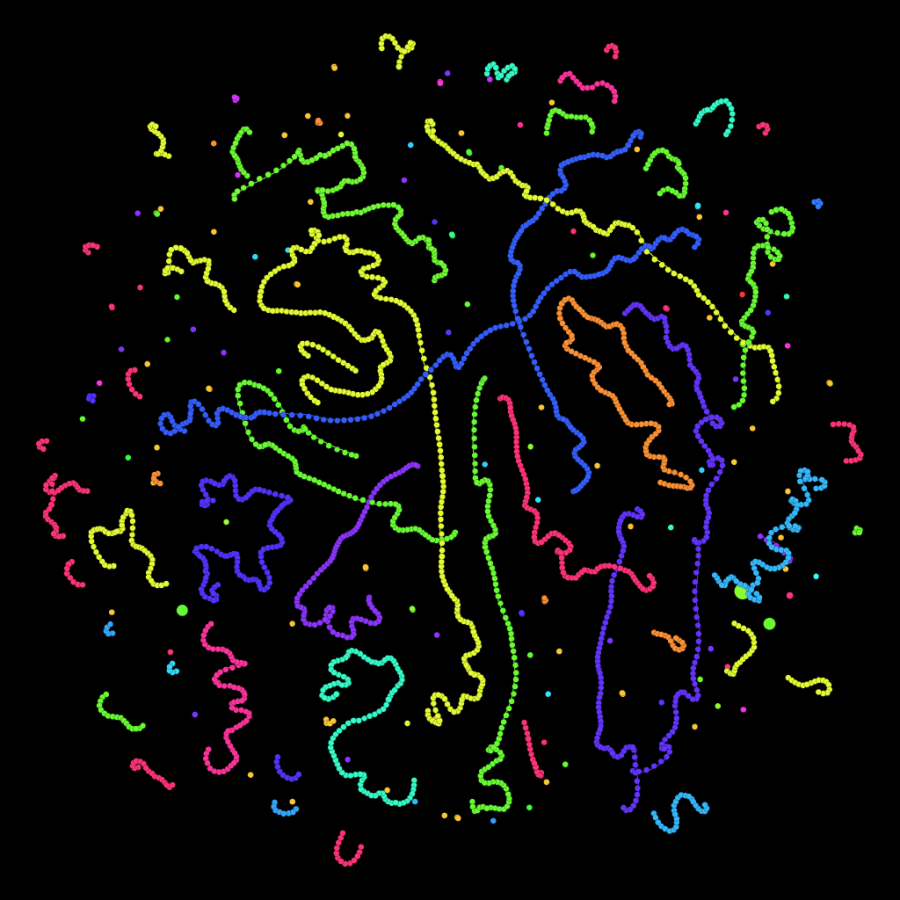
该关系图显示San Diego大学中5000台计算机的僵尸网络，连线的不同颜色代表包的不同协议，粗细代表包大小，节点代表计算机。

### 图形匹配（Graph Pattern Matching）



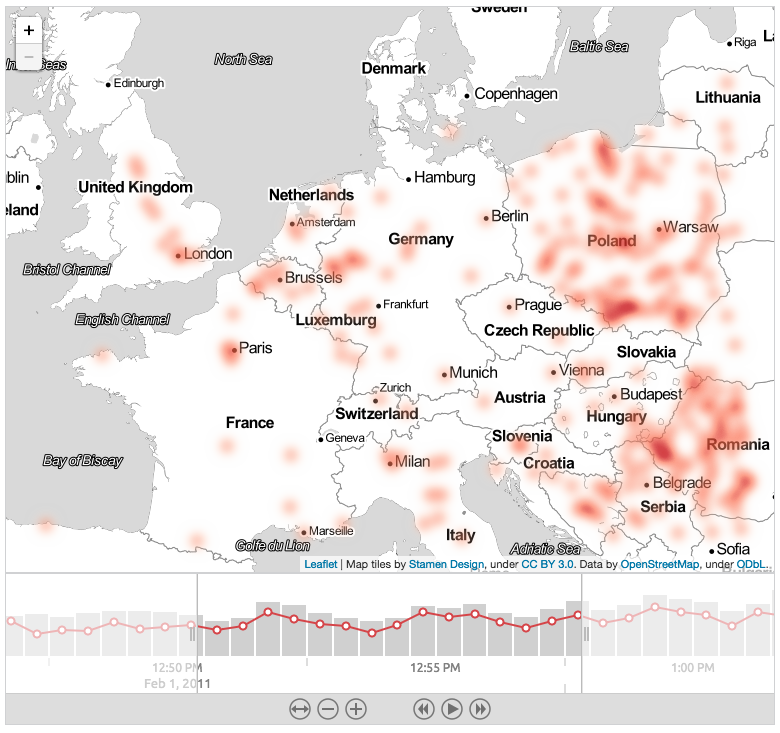
图形匹配能快速发现行为趋势。例如在21CT的这张可视图表中，恶意数据抽取行为一目了然，外部机器（标记为旗帜）试图访问的内部机器（蓝色终端）闪电图形将触发警报。

### 3D可视化



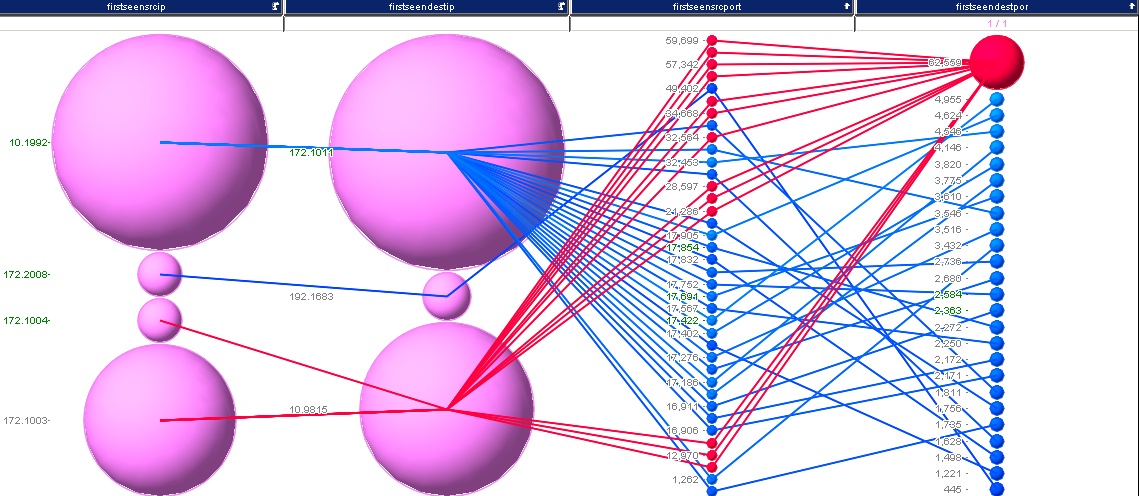
通过logstash和elasticsearch从Apache logfile中提取数据，在Gephi中显示

### 时间线（TimeLine）



显示僵尸网络活动区域，下面的时间轴显示过滤流量的变化。

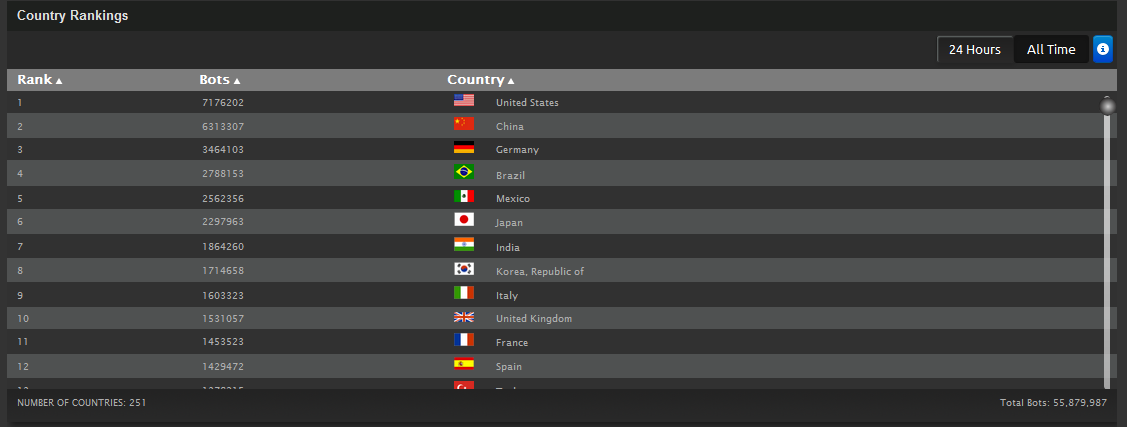
### 平行坐标标记（Parallel Coordinate Plots）



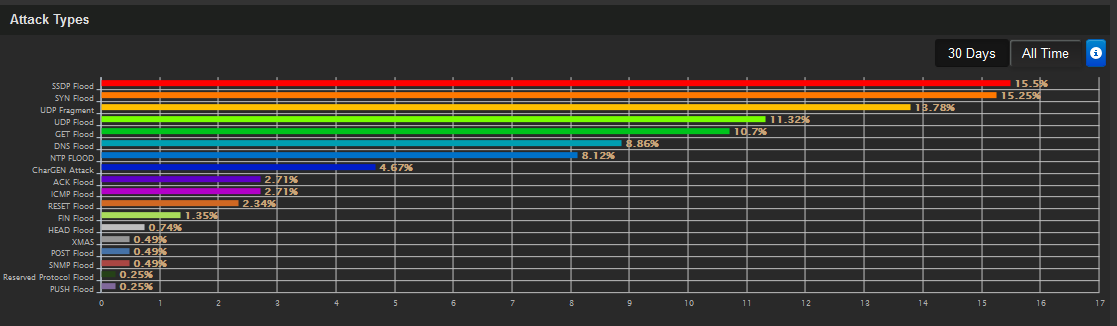
平行坐标标记能有效处理网络分析产生的大数据集，是非常强大的网络数据的可视化方法。上图是安全可视化专家Raffeal Marty绘制的[网络日志数据图](http://raffy.ch/blog/2013/10/22/cleaning-up-network-traffic-logs-vast-2013-challenge/)。

### 标准图表

相对可视化技术，传统标准图表被人熟知，因此易于理解，直观清晰。



表格显示了Akamai客户端24小时内受到DDoS攻击的来源国家排名。



柱状图显示了Akamai客户端统计的30天内DDoS攻击类型的排名。

<http://www.stateoftheinternet.com/trends-visualizations-security-real-time-global-ddos-attack-sources-types-and-targets.html>

# WebGL

Web上的3D技术，开源库包括threejs，unity，babylon js，famous，d3，scenejs，x3dom，kineticjs。

## threejs

http://hewebgl.com/article/articledir/1