### INetSim

[INetSim](http://www.inetsim.org/index.html)是一个非常方便和强大的实用程序，允许你在一台机器上模拟一堆标准的Internet服务。默认情况下，它将模拟可以轻松调整的DNS，HTTP和SMTP。由于我们后续会将受害者机器配置为无Internet访问，因此我们需要使用INetSim进行模拟。

安装INetSim的[方法](http://www.inetsim.org/downloads.html)有多种，最简单的方法就是运行以下命令来安装（在分析机中）。

$ sudo su  
$ echo "deb http://www.inetsim.org/debian/ binary/" > /etc/apt/sources.list.d/inetsim.list  
$ wget -O - http://www.inetsim.org/inetsim-archive-signing-key.asc | apt-key add -  
$ apt update  
$ apt install inetsim

) 使用虚拟机和inetsim搭建恶意代码分析环境，见参考资料；(10分)

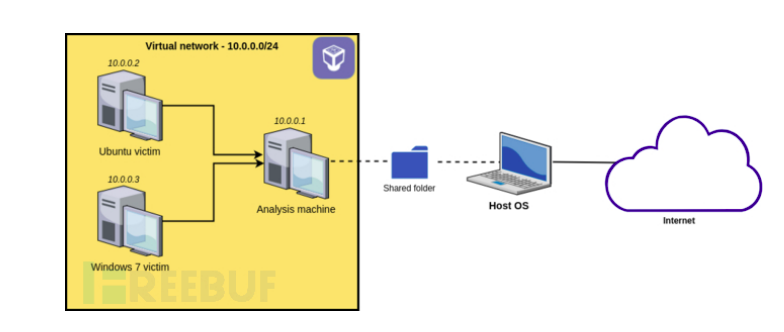
1）在Linux搭建cuckoo沙箱；(10分)

2）使用python或手动上传第1题分析得的恶意软件，阅读分析报告，分析其在进程、注册表、网络等方面的行为，解释分析结果。(10分)

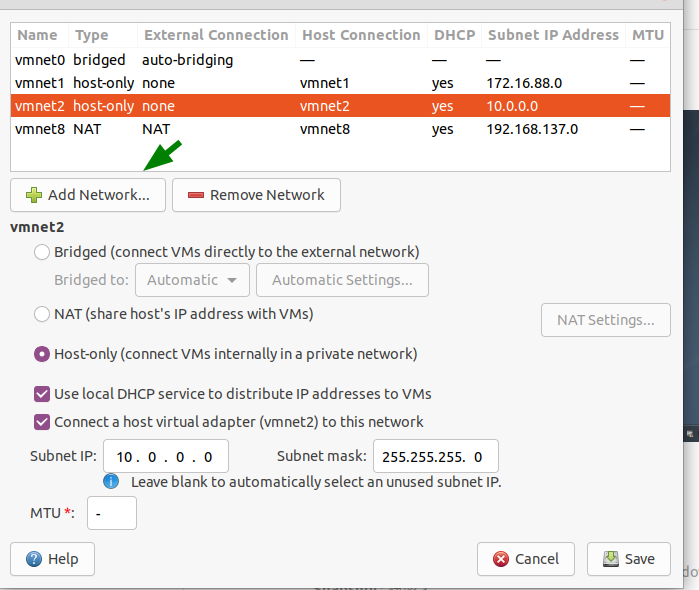
***\*4. Suricata使用(20分)\***\*

1) 在Linux使用***\*源码\***\*安装流量分析系统Suricata；(10分)

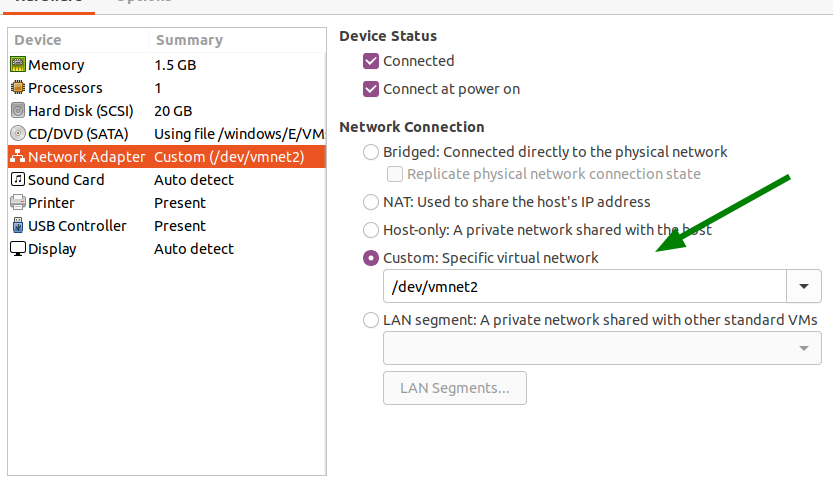
网络拓扑图



添加一个host only网络环境



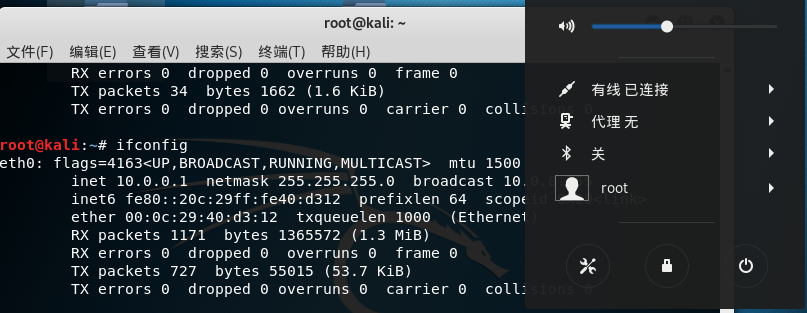
为你的分析机器与受害机选择网络

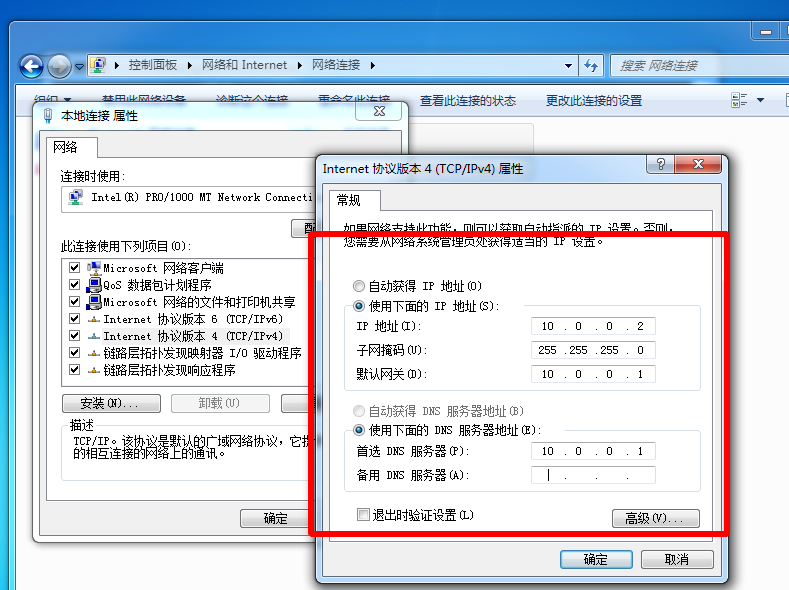


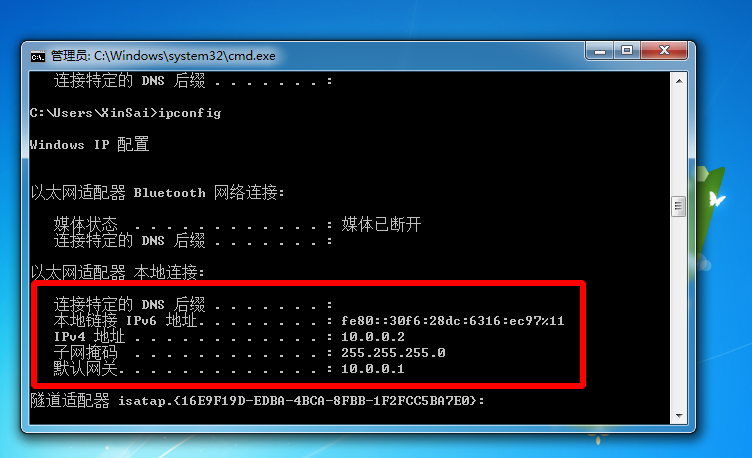
修改网卡，有简单方式用不是很好



重启一下网卡  
ifconfig eth0 down   
 ifconfig eth0 up



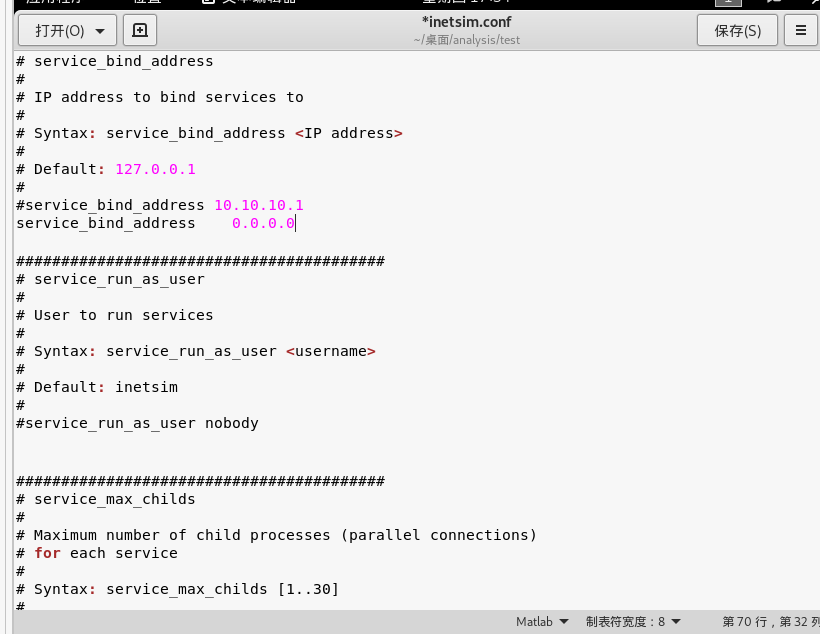




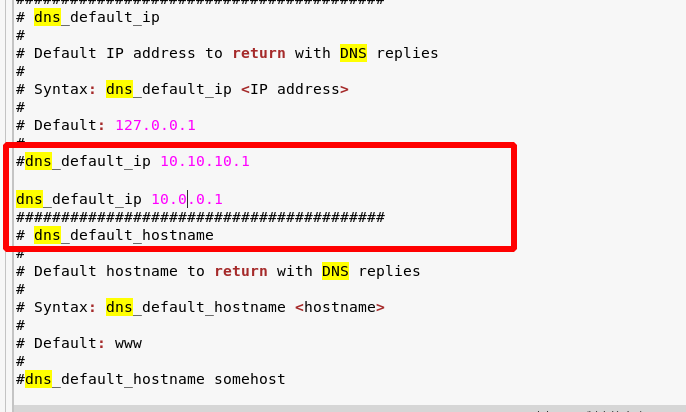


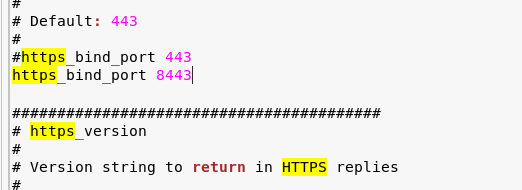
4.创建和还原快照

$ mkdir analysis/test-analysis  
$ cp /etc/inetsim/inetsim.conf analysis/test-analysis  
$ sudo cp -r /var/lib/inetsim analysis/test-analysis/data  
$ sudo chmod -R 777 data  
$ cd analysis/test-analysis



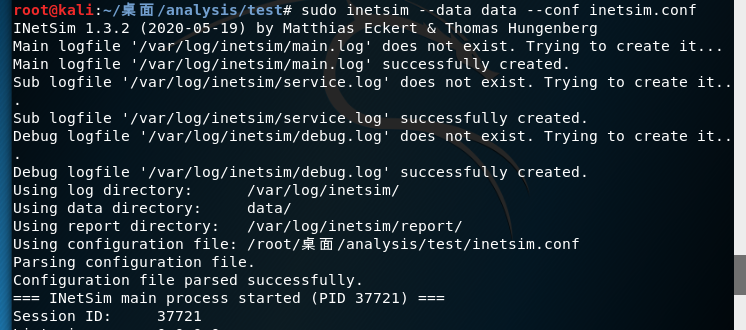
$ sudo systemctl disable systemd-resolved.service  
$ sudo service systemd-resolved stop

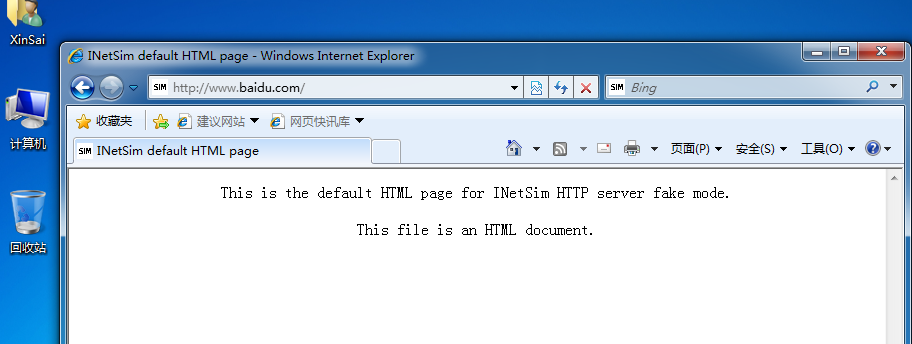




启动

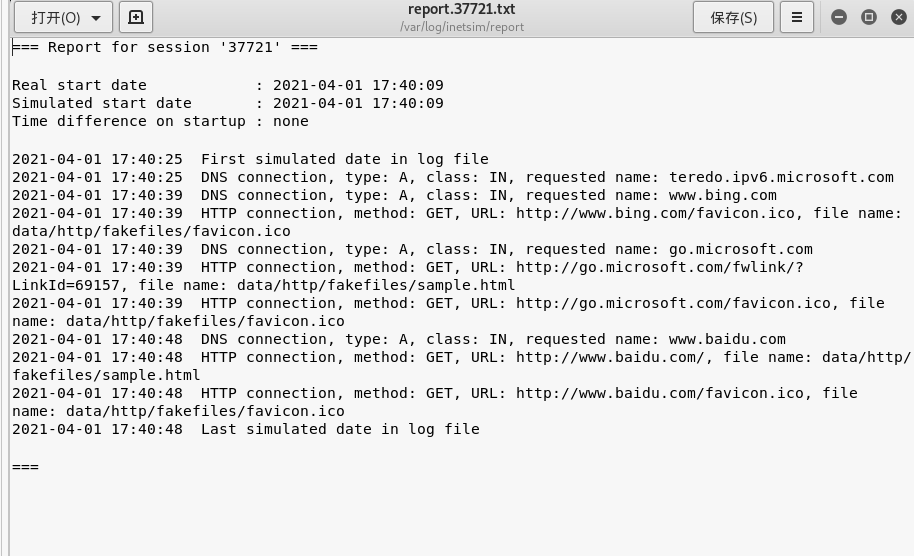
sudo inetsim --data data --conf inetsim.conf



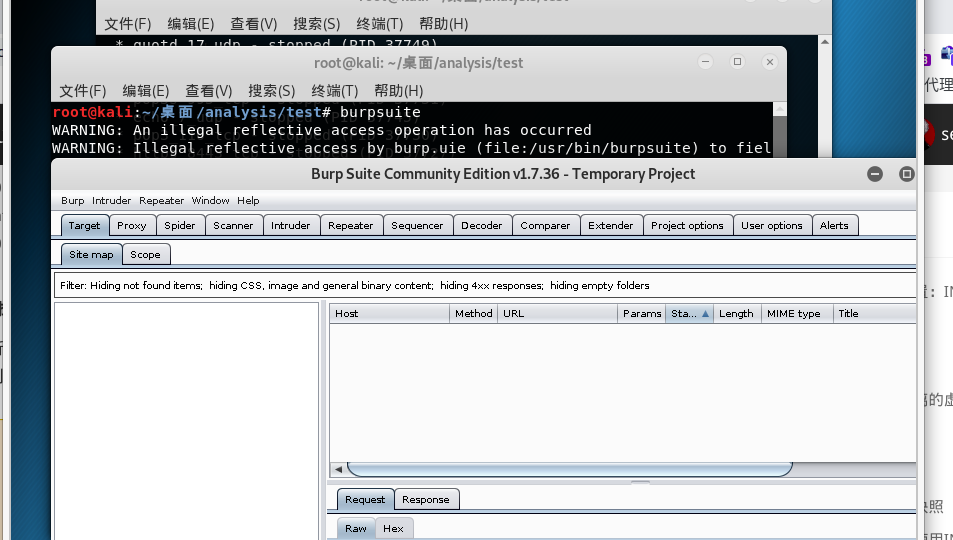


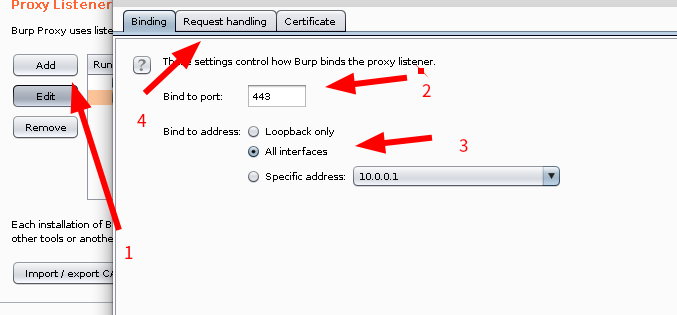
ctrl+c结束并查看摘要报告

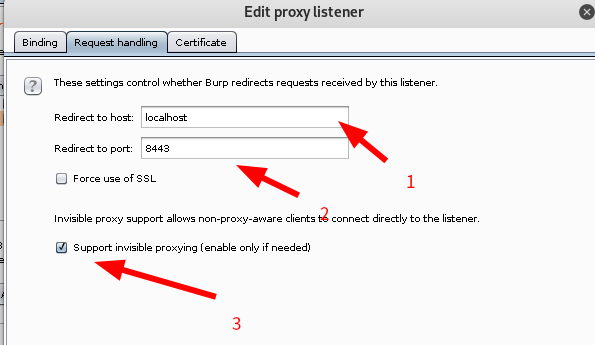
gedit /var/log/inetsim/report/report.37721.txt



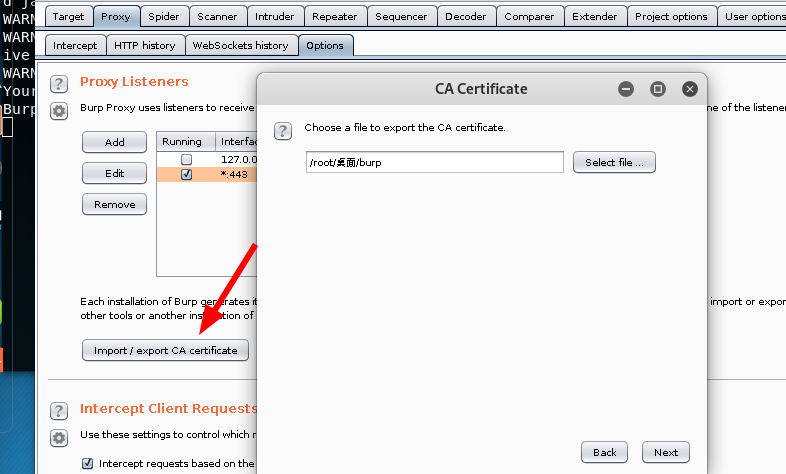
打开burpsuite

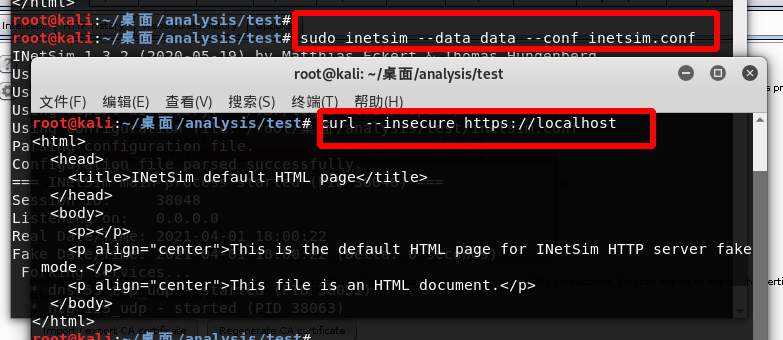




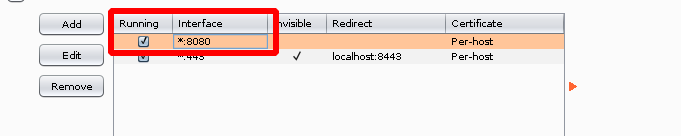


导出证书

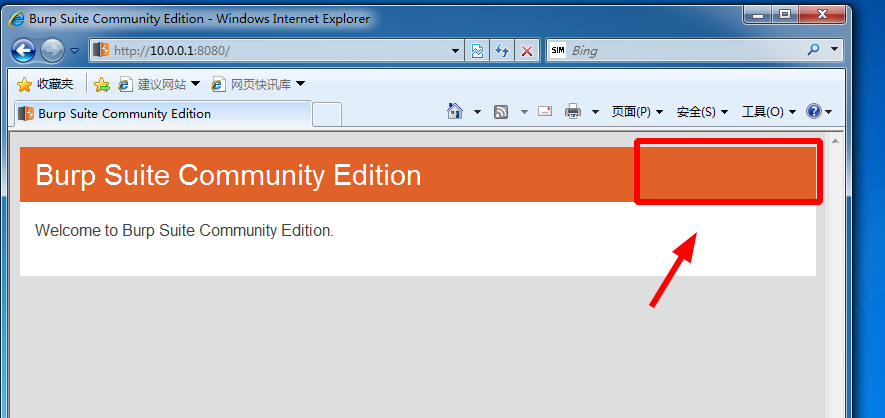


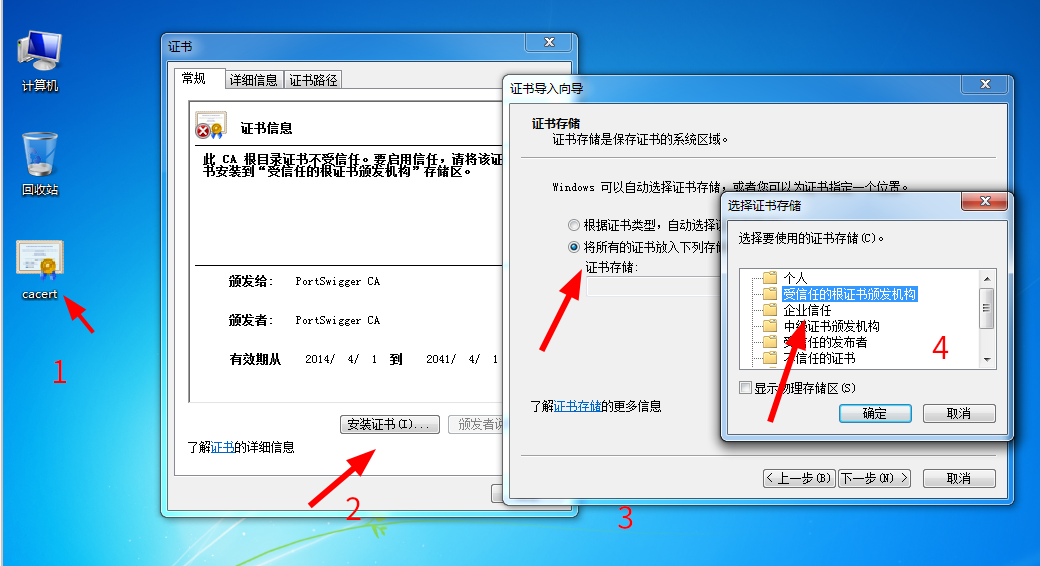


添加监听



访问并点击右上角下载证书





给win7打快照

装了Burp CA证书的Clean状态

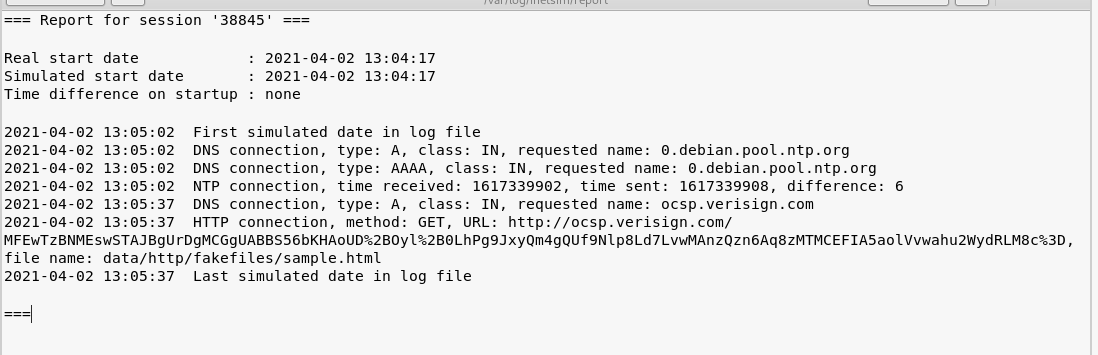
所在状态

点击运行



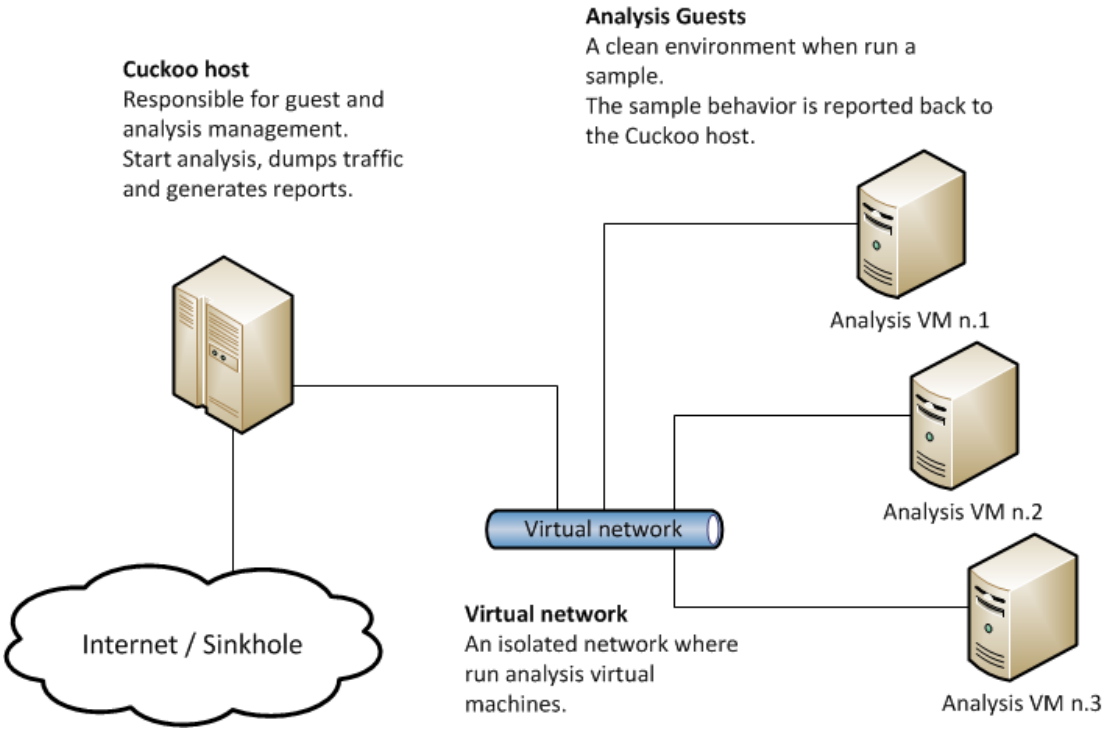


查看生成的报告



cuckoo 安装

cuckoo架构



#换源  
sudo add-apt-repository universe  
sudo apt update

# 解决办法：

1.

sudo apt-get update  
apt-get -f install

2.使用aptitude

apt-get install aptitude

aptitude install

sudo apt-get install python python-pip python-dev libffi-dev libssl-dev  
sudo apt-get install python-virtualenv python-setuptools  
sudo apt-get install libjpeg-dev zlib1g-dev swig

什么玩意，搞了半天这个依赖都没弄对

emm，之前都没这些问题

直接装了个anaconda试一试，反正是python环境

为了使用基于Django的Web界面，需要MongoDB：

$ sudo apt-get install mongodb

为了将PostgreSQL用作数据库（我们的建议），还必须安装PostgreSQL：

$ sudo apt-get install postgresql libpq-dev

$ sudo pip install -U pip setuptools  
$ sudo pip install -U cuckoo

算了，恶心玩意不弄了，睡觉

s

sudo add-apt-repository ppa:oisf/suricata-stable

**Suricata**是一个免费、开源、成熟、快速、健壮的网络威胁检测引擎。Suricata引擎能够进行实时入侵检测(IDS)、内联入侵预防(IPS)、网络安全监控(NSM)和离线pcap处理。Suricata使用强大而广泛的规则和签名语言来检查网络流量，并提供强大的Lua脚本支持来检测复杂的威胁。使用标准的输入和输出格式(如YAML和JSON)，使用现有的SIEMs、Splunk、Logstash/Elasticsearch、Kibana和其他数据库等工具进行集成将变得非常简单。Suricata项目和代码由开放信息安全基金会(OISF)拥有和支持，OISF是一个非盈利基金会，致力于确保Suricata作为一个开源项目的开发和持续成功。

[Suricata](https://suricata-ids.org/) 是一款高性能的网络IDS、IPS和网络安全监控引擎。最著名的 IDS/IPS 是 [Snort](https://www.snort.org/)，那为什么我们还要用 Suricata 呢？最主要的原因是 Snort 只支持单线程，而 Suricata 支持多线程，另外 Suricata 内置支持 IPv6，并且支持现有的 Snort 规则和签名。

## 2. Suricata 规则

一条 Suricata 规则由三个部分组成：

* 操作(Action)
* 头部(Header)
* 规则(Rule options)



#### action

Action操作  
pass:如果匹配到了规则，则 Suricata 停止扫描数据包并跳过当前所有规则（指的是当前规则包内的所有规则）  
drop:丢弃数据包 IPS/inline下使用  
REject:  
Alter:  
 当匹配到规则时，Suricata 不会对数据包进行任何操作，会像对正常数据包一样进行放行，除了会记录一条只有管理员能够看到的警报。  
  
默认的优先级为：Pass > Drop > Reject > Alert。

#### header

头部中包含如下几项：  
  
协议(Protocol)  
源/目的地址(Source and destination)  
端口号(Ports)  
流向(Direction)

协议

其取值可以为：tcp，udp，icmp，ip，http，ftp，tls(包含ssl)，smb，dns等等。

原地质与目的地址

源/目的地址可以设置为 IP 地址或者在配置文件(Suricata.yaml)里定义的变量，如：$HOME\_NET和$EXTERNAL\_NET。具体可以设置  
  
! 1.1.1.1 (除了 1.1.1.1 以外的所有 IP)  
![1.1.1.1, 1.1.1.2] (除了 1.1.1.1 和 1.1.1.2 以外的所有 IP)   
$HOME\_NET (配置文件中设置的 HOME\_NET)  
[$EXTERNAL\_NET, !$HOME\_NET] (包含 EXTERNAL\_NET 但不含 HOME\_NET)  
[10.0.0.0/24, !10.0.0.5] (10.0.0.0/24 除了 10.0.0.5)   
[…..,[....]]  
[…. ,![.....]]

#### 端口

! 排除  
: 范围  
[] 包含  
, 分隔符

[80, 81, 82] (端口号 80, 81 和 82)  
[80: 82] (范围 80 到 82)  
[1024: ] (从 1024 起到最大端口号)  
!80 (除了 80 以外的所有端口号)  
[80:100,!99] (从 80 到 100 除了 99 的端口)  
[1:80,![2,4]]  
[….[.....]]

#### **流向（Direction**

source -> destination   
source <> destination (双向匹配)

例子：

**alert** **tcp** 1.2.3.4 1024 **-** > 5.6.7.8 80

### 规则

规则的每一条都遵循固定的格式：

name: settings;

#### msg

msg 关键字就是当本条规则匹配中时，显示在日志中的提示内容，格式如下：  
  
msg: “..........”;   
msg 关键字必须为每条规则的第一个关键字，也就是说必须出现在规则的最前面。  
msg:"ATTACK-RESPONSES 403 Forbidden";   
msg:"ET EXPLOIT SMB-DS DCERPC PnP bind attempt";

**Sid (signature id)**

表示sid 关键字识每条规则的 id，必须位数字且唯一。示例如下：

sid:2008124;

**Rev (Revision)**

rev 关键字标识修订版本号，每次修改后加一。

**Gid (group id)**

gid 关键字表示每条规则分组的 id，功能类似与 sid，不过因为是分组的 id，所以值可以重复，另外就是如果不指定的话，默认所有规则的 gid 值都为 1。其实这个值也只有在日志中才可能出现，如下所示：

**Classtype**

classtype 关键字表示对当前规则的分类的情况，这个关键字一般是在 classification.conf 文件中指定由名称，描述和优先级组成，它可以用来显示当前规则只是一条信息或是遭受了黑客的攻击。classtype 定义示例如下：

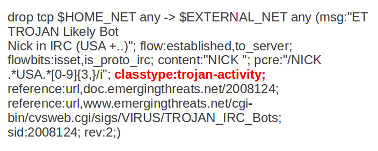
**Reference**

reference 字段表示当前规则相关信息所在 url，可以同时指定多条 url，一般规则如下：

reference: url, www.info.nl

**Priority**

priority 关键字用于指定当前规则的优先级，取值范围从 1 到 255，取值越小优先级越大，但常用值为 1-4。



### content

content关键字在suricata规则中非常重要，大部分规则都要使用这个关键字来匹配数据包中的内容，其格式如下

content:".......";

content:"|4d5a|"

规则所在目录

default-rule-path: /etc/suricata/rules   
rule-files:   
 - botcc.rules  
 - ciarmy.rules  
 - compromised.rules  
 - drop.rules  
# - dshield.rules  
# - emerging-activex.rules  
...

flow流匹配：

flow是特定时间内具有相同数据的数据包（5元组信息）同属于一个流，suricata会将这些流量保存在内存中。

flowbits set , name 设置条件

flowbits isset, name 选择条件

to*client/from*server 服务器到客户端

to*server/from*client 客户端到服务器

established 匹配已经建立连接的（tcp则是经过3次握手之后，udp则是有双向流量）

no\_established 匹配不属于建立连接的

only\_stream 匹配由流引擎重新组装的数据包

no\_stream 不匹配流引擎重新组装的数据包

阀值 threshold：

threshold: type <threshold|limit|both>, track <by*src|by*dst>, count , seconds

threshold 最小阀值 也就是说只有匹配到至少多少次数才进行告警

limit 限制告警次数，例如最少5分钟内告警一次

调节阀值主要是通过2种方法，一种是通过规则内的threshold关键字来调节，下图中类型是limit也就是限制告警次数，track by\_src代表来源ip，seconds 60 每个60秒告警一次count 1

MALWARE Windows executable sent when host clamis to send an image