前言

  红蓝对抗的思想最早可追溯到我国现存最早的一部兵书《孙子兵法》，孙子·谋攻篇：“知彼知己，百战不殆；”，意为如果对敌我双方的情况都能了解透彻，打多少次仗都不会失败。在信息安全领域目前大家都有一个共识：“未知攻，焉知防”，攻防对抗本身是一个持续的过程，在具体的对抗中，对对手了解越多就会占据主导地位。

**红蓝对抗的主要目的在于，提高公司安全成熟度及其检测和响应攻击的能力。**

**Red Teams attack, and Blue Teams defend, but the primary goal is shared between them: improve the security posture of the organization.**

0. ) 准备工作

1 ) 组织结构图   
2 ) 全网拓扑图   
3 ) 各系统逻辑结构图   
4 ) 各系统之间的调用关系   
5 ) 数据流关系   
6 ) 核心资产清单   
7 ) 应急响应计划   
8 ) 业务连续性计划   
9 ) 灾难恢复计划

1. ) 简单安全评估

**1.1 ) 端口扫描和漏洞检测**

**主机发现（Ping探测）：**   
# nmap -sn -PE IP地址或地址段

**端口扫描：**   
# nmap –open IP地址或地址段

**服务版本检测：**   
# nmap -sV IP地址或地址段

**扫描多个端口：**   
# nmap -p 80,443 IP地址或地址段

**UDP端口扫描：**   
# nmap -sU -p 53 IP地址或地址段

**TCP/UDP端口扫描（-Pn 跳过主机发现）：**   
# nmap -v -Pn -SU -ST -p U:53,111,137,T:21-   
25,80,139,8080 IP地址或地址段

**Nessus漏洞检测：**   
# nessus -q -x -T html 服务器IP 服务器端口 管理员帐号 密码 目标.txt 输出报告.html

**OPENVAS漏洞检测：**   
# apt -y install pcregrep   
# wget <https://goo.gl/TYbLwE>   
# chmod +x openvas-automate.sh && ./openvas-automate.sh 目标IP

**1.2 ) WINDOWS系统篇**

1.2.1 网络发现

**基本网络发现**   
# C:> net view /all   
# C:> net view \\主机名

**Ping探测**   
# C:> for /L %I in (1,1,254) do ping -w 30 -n 1 192.168.1.%I | find “回复” >> 输出.txt

1.2.2 DHCP

**启用DHCP服务器日志功能：**   
# C:> reg add HKLM\System\CurrentControlSet\Services\DhcpServer\Parameters /v ActivityLogFlag /t REG\_DWORD /d 1   
默认日志文件路径：   
C:> %windir%\System32\Dhcp

1.2.3 DNS

**启用DNS服务器日志功能：**   
# C:> DNSCmd DNS服务器名 /config /logLevel 0x8100F331   
# 配置日志文件路径:   
C:> DNSCmd DNS服务器名 /config /LogFilePath C:\dns.log   
# 配置日志文件大小:   
C:> DNSCmd DNS服务器名 /config /logfilemaxsize 0xffffffff

1.2.4 哈希值

**文件校验和完整性验证（FCIV）：**   
Ref：<http://support2.microsoft.com/kb/841290>   
# 计算单个文件hash值:   
C:> fciv.exe 文件名   
# 计算C盘所有文件的哈市值并把结果保存到文件中:   
C:> fciv.exe c:\ -r -sha1 -xml 结果.xml   
# 列出所有hash值:   
C:> fciv.exe -list -sha1 -xml 结果.xml   
# **certutil & PowerShell方法**   
# certutil -hashfile 文件名 SHA1   
# PS C:> Get-FileHash 文件名 | Format-List   
# PS C:> Get-FileHash -algorithm md5 文件名

1.2.5 NETBIOS

**nbtstat 扫描**   
# C:> nbtstat -A 目标IP地址   
**NetBIOS缓存**   
# C:> nbtstat -c   
**批量扫描**   
# C:> for /L %I in (1,1,254) do nbtstat -An 192.168.1.%I

1.2.6 微软基线安全分析器(MBSA)

**扫描单个IP**   
# C:> mbsacli.exe /target IP地址 /n os+iis+sql+password   
**扫描IP地址段**   
# C:> mbsacli.exe /r IP地址段 /n os+iis+sql+password

**1.3 ) LINUX系统篇**

1.3.1 网络发现

**查看开放的SMB共享**   
# smbclient -L 目标主机名

**Ping探测**   
  
\# for ip in $(seq 1 254); do ping -c1 -w2 192.168.1.$ip>/dev/null; [ $? -eq 0 ] && echo "192.168.1.$ip UP" || : ; done

1.3.2 DHCP

**DHCP日志查询**   
RHEL/CentOS   
# cat /var/lib/dhcpd/dhcpd. leases   
Debian/Ubuntu   
# grep -Ei ‘dhcp’ /var/log/syslog.1

1.3.3 DNS

**DNS日志查询**   
# rndc querylog && tail -f /var/log/messages | grep named

1.3.4 哈希值

**计算/sbin目录下所有可执行文件的HASH值**   
# find /sbin -type f -exec md5sum {} >> md5sums.txt \;   
# md5deep -rs /sbin > md5sums.txt

1.3.5 NETBIOS

**nbtstat 扫描**   
# nbtscan 目标IP地址或IP地址段   
举例：nbtscan 192.168.1.2-100

2. ) 安全加固

**2.1 ) WINDOWS系统篇**

**2.1.1 禁用/停止服务**   
# C:> sc query   
# C:> sc config “服务名” start= disabled   
# C:> sc stop “服务名”   
# C:> wmic service where name=”服务名” call ChangeStartmode Disabled

**2.1.2 防火墙管理**   
# 列出所有规则:   
# C:> netsh advfirewall firewall show rule name=all   
# 启用或禁用防火墙:   
C:> netsh advfirewall set currentprofile state on   
C:> netsh advfirewall set currentprofile firewallpolicy blockinboundalways,allowoutbound   
C:> netsh advfirewall set publicprofile state on   
C:> netsh advfirewall set privateprofile state on   
C:> netsh advfirewall set domainprofile state on   
C:> netsh advfirewall set allprofile state on   
C:> netsh advfirewall set allprof ile state off   
# 配置举例：   
  
netsh advfirewall firewall add rule name="开放TCP:80端口" dir=in action=allow protocol=TCP localport=80   
netsh advfirewall firewall add rule name="开放TCP:443端口" dir=in action=allow protocol=TCP localport=443   
netsh advfirewall firewall add rule name="屏蔽TCP:445端口" dir=in action=block protocol=TCP localport=445   
netsh advfirewall firewall add rule name="允许MyApp" dir=in action=allow program="C:\MyApp\MyApp.exe" enable=yes

**2.1.3 清除DNS缓存和Netios缓存**   
# C:> ipconfig /flushdns   
# C:> nbtstat -R

**2.1.4 应用控制**   
# AppLocker配置   
  
\# 导入Applocker模块   
PS C:\> import-module Applocker   
\# 查看system32目录下所有exe文件的Applocker信息   
PS C:\> Get-ApplockerFileinformation -Directory C:\Windows\System32\ -Recurse -FileType Exe   
\# 增加一条针对system32目录下所有的exe文件的允许规则   
PS C:\> Get-Childitem C:\Windows\System32\\*,exe | Get-ApplockerFileinformation | New-ApplockerPolicy -RuleType Publisher, Hash -User Everyone -RuleNamePrefix System32

**2.1.5 IPSEC**   
# 1.）使用预共享密钥的方式新建一条IPSEC本地安全策略，应用到所有连接和协议   
  
C:\> netsh ipsec static add filter filterlist=MyIPsecFilter srcaddr=Any dstaddr=Any protocol=ANY   
C:\> netsh ipsec static add filteraction name=MyIPsecAction action=negotiate   
C:\> netsh ipsec static add policy name=MyIPsecPolicy assign=yes   
C:\> netsh ipsec static add rule name=MyIPsecRule policy=MyIPsecPolicy filterlist=MyIPsecFilter filteraction=MyIPsecAction conntype=all activate=yes psk=密码  
  
# 2.）新建一条允许访问外网TCP 80和443端口的IPSEC策略   
  
C:\> netsh ipsec static add filteraction name=Allow action=permit   
C:\> netsh ipsec static add filter filterlist=WebFilter srcaddr=Any dstaddr=Any protocol=TCP dstport=80   
C:\> netsh ipsec static add filter filterlist=WebFilter srcaddr=Any dstaddr=Any protocol=TCP dstport=443   
C:\> netsh ipsec static add rule name=WebAllow policy=MyIPsecPolicy filterlist=WebFilter filteraction=Allow conntype=all activate=yes psk=密码   
  
# 3.）查看和禁用某条IPSEC本地安全策略   
  
C:\> netsh ipsec static show policy name=MyIPsecPolicy   
C:\> netsh ipsec static set policy name=MyIPsecPolicy assign=no   
  
# 新建一条IPSEC对应的防火墙规则，源地址和目的地址为any   
  
C:\> netsh advfirewall consec add rule name="IPSEC" endpointl=any endpoint2=any action=requireinrequireout qmsecmethods=default   
  
# 新建一条IPSEC对应的防火墙规则，所有出站请求必须提供预共享密钥   
  
C:\> netsh advfirewall firewall add rule name="IPSEC\_Out" dir=out action=allow enable=yes profile=any localip=any remoteip=any protocol=any interfacetype=any security=authenticate

**2.1.6 其他安全策略**   
# 禁用远程桌面连接   
  
C:\> reg add "HKLM\SYSTEM\CurrentControlSet\Control\TerminalServer" /f /v fDenyTSConnections /t REG\_DWORD /d 1   
  
# 只发送NTLMv2响应（防止“永恒之蓝”漏洞攻击）   
  
C:\> reg add HKLM\SYSTEM\CurrentControlSet\Control\Lsa\ /v lmcompatibilitylevel /t REG\_DWORD /d 5 /f   
  
# 禁用IPV6   
  
C:\> reg add HKLM\SYSTEM\CurrentControlSet\services\TCPIP6\Parameters /v DisabledComponents /t REG\_DWORD /d 255 /f   
  
# 禁用sticky键   
  
C:\> reg add "HKCU\ControlPanel\Accessibility\StickyKeys" /v Flags /t REG\_SZ /d 506 /f   
  
# 禁用管理共享（Servers/Workstations）   
  
C:\> reg add HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters /f /v AutoShareServer /t REG\_DWORD /d 0   
C:\> reg add HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters /f /v AutoShareWks /t REG\_DWORD /d 0   
  
# 禁用注册表编辑器和CMD命令提示符   
  
C:\> reg add HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System /v DisableRegistryTools /t REG\_DWORD /d 1 /f   
C:\> reg add HKCU\Software\Policies\Microsoft\Windows\System /v DisableCMD /t REG\_DWORD /d 1 /f   
  
# 启用UAC   
  
C:\> reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System /v EnableLUA /t REG\_DWORD /d 1 /f   
  
# 启用防火墙日志功能   
  
C:\> netsh firewall set logging droppedpackets = enable   
C:\> netsh firewall set logging connections = enable

**2.2 ) LINUX系统篇**

**2.2.1 服务管理**   
# 查看服务状态   
  
service --status-all   
ps -ef OR ps -aux   
initctl list   
systemctl list-unit-files   
  
# 启动，停止和禁用服务   
  
\# For Upstart services:   
/etc/init.d/apache2 start | stop | status   
service apache2 start | stop | status   
update-rc.d apache2 disable   
\# For Systemd services:   
systemctl start | stop | status ntp.service   
systemctl disable sshd.service   
  
**2.2.2 防火墙管理**   
# iptables 常用操作：   
  
iptables-save > filewall\_rules.bak \# 导出当前规则   
iptables -vnL --line \# 列出所有规则   
iptables -S \# 同上   
iptables -P INPUT DROP \# 默认策略，禁止所有连接   
iptables -A INPUT -s 10.10.10.10 -j DROP \# 禁止单个IP   
iptables -A INPUT -s 10,10.10.0/24 -j DROP \# 禁止一个网段   
iptables -A INPUT -p tcp --dport ssh -s 10.10.10.10 -j DROP \# 禁止某IP访问本机SSH服务   
iptables -A INPUT -p tcp --dport ssh -j DROP \# 禁止访问本机SSH服务   
iptables -I INPUT 5 -m limit --limit 5/min -j LOG --log-prefix "iptables denied: " --log-level 7 \# 启用日志记录   
iptables -F \# 清除当前已加载的规则   
  
**2.2.3 DNS缓存**   
# Unix/Linux系统没有系统级别DNS缓存

**2.2.4 配置IPSEC**   
# 在两台服务器之间建立IPSEC通道   
  
\# 1.）添加防火墙规则允许IPSEC协议   
iptables -A INPUT -p esp -j ACCEPT   
iptables -A INPUT -p ah -j ACCEPT   
iptables -A INPUT -p udp --dport 500 -j ACCEPT   
iptables -A INPUT -p udp --dport 4500 -j ACCEPT   
\# 安装Racoon   
apt -y install racoon   
\# 2.）编辑配置文件：/etc/ipsec-tools.conf   
flush;   
spdflush;   
spdadd 主机A的IP地址 主机B的IP地址 any -P out ipsec   
esp/transport//require;   
spdadd 主机B的IP地址 主机A的IP地址 any -P in ipsec   
esp/transport//require;   
\# 3.）编辑配置文件：/etc/racoon/racoon.conf   
log notify;   
path pre\_shared\_key "/etc/racoon/psk.txt";   
path certificate "/etc/racoon/certs";   
remote anonymous {   
exchange\_mode main,aggressive;   
proposal {   
encryption\_algorithm aes\_256;   
hash\_algorithm sha256;   
authentication\_method   
pre\_shared\_key;   
dh\_group modp1024;   
}   
generate\_policy off;   
}   
sainfo anonymous{   
pfs\_group 2;   
encryption\_algorithm aes\_256;   
authentication\_algorithm hmac\_sha256;   
compression\_algorithm deflate;   
}   
\# 4.）添加预共享密钥   
主机A：echo 主机B 123 >> /etc/racoon/psk.txt   
主机B：echo 主机A 123 >> /etc/racoon/psk.txt   
\# 5.）重启服务，检查协商及配置策略   
service setkey restart   
setkey -D   
setkey -DP

3. ) 检测（Visibility）

**3.1 ) 网络安全监控**

**3.1.1 数据包捕捉与分析**   
  
1.）TCPDUMP   
tcpdump -tttt -n -vv \# 打印时戳，禁用名称解析并以verbose方式显示   
tcpdump -nn -c 1000 | awk '{print $3}' | cut -d. -f1-4 | sort -n | uniq -c | sort -nr \# 捕捉1000个数据包，找出Top talkers   
tcpdump -w target.pcap -i any dst targetIP and port 80 \# 在所有接口上捕捉目标IP为：targetIP且端口为80的数据包并写入target.pcap文件   
tcpdump host 10.0.0.1 && host 10.0.0.2 \# 捕捉两个主机之间的数据包   
tcpdump not net 10.10 && not host 192.168.1.2 \#捕捉非10.10网段及非192.168.1.2主机的数据包  
tcpdump host 10.10.10.10 && \(10.10.10.20 or 10.10.10.30\) \# 捕捉主机A与主机B或C的数据包  
tcpdump -n -s0 -C 100 -w 001.pcap \# 轮询，文件大小超过100M后自动创建新文件   
tcpdump -w - | ssh ServerIP -p 50005 "cat - > /tmp/remotecapture.pcap" \# 通过SSH保存数据包到远程服务器上的/tmp/remotecapture.pcap文件   
tcpdump -n -A -s0 port http or port ftp or port smtp or port imap or port pop3 | egrep -i 'pass=|pwd=|log=|login=|user=|username=|pw=|passw=|P   
asswd=|password=|pass:|user:|username:|password:|login:|pass|user' --color=auto --line-buffered -B20 \# 抓取明文密码   
tcpdump -s 1500 -A '(tcp[((tcp[12:1] & 0xf0) >> 2)+5:1] = 0x01) and (tcp[((tcp[12:1] & 0xf0) >> 2):1] = 0x16)' \#查找自签名证书   
2.）TSHARK   
tshark -nr 001.pcap -Y "ssl.handshake.ciphersuites" -Vx | grep "ServerName:" | sort | uniq -c | sort -r \# 提取证书Server Name字段   
tshark -D \# 列出所有可用接口   
tshark -i eth0 -i eth1 \# 监听多个接口   
tshark -nn -w 001.pcap \# 禁用名称解析并保存到文件   
tshark arp or icmp \# 捕捉arp或者icmp   
tshark "host 主机A && host 主机B" \# 捕捉两个主机之间的数据包   
tshark -r 001.pcap \# 读取数据包   
tshark -n -e ip.src -e ip.dst -T fields -E separator=, -2 -R ip -r 001.pcap \# 提取源/目的IP地址   
tshark -n -e ip.src -e dns,qry.name -E separator=';' -T fields port 53 \# 提取DNS查询的源IP及DNS查询的域名   
tshark -2 -R http.request -T fields -E separator=';' -e http.host -e http.request.uri -r 001.pcap \# 提取HTTP请求中的host参数和请求uri   
tshark -n -c 150 I awk '{print $4}' I sort -n | uniq -c | sort -nr \# 提取top talkers   
tshark -q -z io,phs -r 001.pcap \# 协议统计   
tshark -n -c 100 -e ip.src -Y "dns.flags.response eq 1" -T fields port 53 \# 提取DNS响应数据包的DNS服务器地址   
tshark -n -e http.request.uri -Y http.request -T fields | grep exe \# 提取通过http请求方式下载exe可执行文件的数据包   
3.）SNORT   
snort -T -c /etc/snort/snort.conf \# 配置文件测试   
snort -dv -r 001.log \# 读取数据包   
snort -dvr 001.log icmp \# 提取icmp数据包   
snort -K ascii -l 001 \# 以ASCII格式显示   
snort -q -A console -i eth0 -c /etc/snort/snort.conf \# 在终端上显示snort events   
echo 'log tcp 192.168.1.0/24 any -> 192.168.1.95 22 ( msg: "ssh access" ; sid:1618008; )' > 001.rule && snort -T -c 001.rule \# 规则测试   
mkdir logs && snort -vd -c 001.rule -r 001.pcap -A console -l logs \# 执行规则   
4.）Bro NSM   
安装及下载相关软件包和数据包   
apt -y install bro bro-aux   
pip install bro-pkg   
bro-pkg install bro/hosom/file-extraction   
wget https://www.malware-traffic-analysis.net/2018/01/12/2018-01-12-NanoCore-RAT-traffic.pcap.zip   
wget https://www.bro.org/static/exchange-2013/faf-exercise.pcap   
bro -r 2018-01-12-NanoCore-RAT-traffic.pcap \# 从pcap文件中读取并创建相关日志文件   
bro -r faf-exercise.pcap /root/.bro-pkg/scratch/file-extraction/scripts/plugins/extract-pe.bro && ls -lhct ./extract\_files/ \# 提取出exe文件  
bro -r faf-exercise.pcap /usr/share/bro/policy/frameworks/files/extract-all-files.bro \# 提取多个类型的文件   
bro -C -r faf-exercise.pcap && cat ssl.log | bro-cut server\_name , subject , issuer \# 提取证书中的server\_name,issuer和subjects字段   
cat conn.log | bro-cut id.orig\_h , id.orig\_p , id.resp\_h , id.resp\_p , proto , conn\_state \# 提取源IP，源端口，目的IP，目的端口，协议类型，tcp标记   
cat dns.log | bro-cut query | sort -u \# 提取DNS查询name   
cat http.log | bro-cut id.orig\_h , id.orig\_p , id.resp\_h , id.resp\_p , host , uri , referrer \# 提取源IP，源端口，目的IP，目的端口，host，uri，referrer字段   
cat http.log | bro-cut user\_agent | sort -u \# 提取user\_agent字段   
5.）EDITCAP   
editcap -F pcap -c 1000 orignal.pcap out\_split.pcap \# 以1000为单位进行分割   
editcap -F pcap -t+3600 orignal.pcap out\_split.pcap \# 以1小时为单位进行分割   
6.）MERGECAP   
mergecap -w merged\_cap.pcap capl.pcap cap2.pcap cap3.pcap \# 合并多个文件   
7.）PacketTotal   
https://www.packettotal.com/app/analysis?id=c8c11b792272ac19a49299a3687466be&name=files   
8.）NetworkMiner   
http://netres.ec/?b=173588E   
  
**3.2 ) 蜜罐技术**   
**3.2.1 WINDOWS系统篇**   
  
1.） 端口蜜罐   
\# 原理：监听端口，当客户端成功建立TCP（3次握手）连接后，记录访问日志，然后添加防火墙规则封禁此IP   
PS C:\> certutil.exe -urlcache -split -f https://raw.githubusercontent.com/Pwdrkeg/honeyport/master/honeyport.ps1   
PS C:\> .\honeyport.ps1 -Ports 4444,22,21,23 -WhiteList 192.168.10.1,192.168.10.2 -Block $true -Verbose   
PS C:\> Get-EventLog HoneyPort \# 查看日志记录信息   
PS C:\> stop-job -name HoneyPort \# 停止任务   
PS C:\> remove-job -name HoneyPort \# 移除任务

**3.2.2 LINUX系统篇**   
  
1.） 端口蜜罐   
\# 原理同上   
wget https://raw.githubusercontent.com/gchetrick/honeyports/master/honeyports-0.5.py   
python honeyports-0.5.py -p 1234 -h 192.168.1.100 -D   
  
**3.3 ) (PASSIVE)监控DNS解析**   
  
apt -y install dnstop   
dnstop -l 3 eth0   
dnstop -l 3 001.pcap | out.txt   
  
**3.4 ) 日志审计**   
  
1.）WINDOWS   
\# 增加日志文件大小进行日志审计   
C:\> reg add HKLM\Software\Policies\Microsoft\Windows\Eventlog\Application /v MaxSize /t REG\_DWORD /d 0x19000   
C:\> reg add HKLM\Software\Policies\Microsoft\Windows\Eventlog\Security /v MaxSize /t REG\_DWORD /d 0x64000   
C:\> reg add HKLM\Software\Policies\Microsoft\Windows\EventLog\System /v MaxSize /t REG\_DWORD /d 0x19000   
\# 查看Windows事件日志-安全日志的配置   
C:\> wevtutil gl Security   
\# 检查审核策略   
auditpol /get /category:\*   
\# 对所有项启用成功和失败的审核策略   
C:\> auditpol /set /category:\* /success:enable /failure:enable   
\# 查看已配置的事件日志的概要信息   
PS C:\> Get-Eventlog -list   
\# 取最近5条应用程序日志   
PS C:\> Get-Eventlog -newest 5 -logname application | Format-List   
\# 取Eent ID：4672的所有日志   
PS C:\> Get-Eventlog Security | ? { $\_.Eventid -eq 4672}   
\# 登录与注销事件   
PS C:\> Get-Eventlog Security 4625,4634,4647,4624,4625,4648,4675,6272,6273,6274,6275,6276,6277,6278,6279,6280,4649,4778,4779,4800,4801,4802,4803,5378,5632,5633,4964 -after ((get-date).addDays(-1))   
\# DPAPI行为，进程终止，RPC事件   
PS C:\> Get-EventLog Security 4692,4693,4694,4695,4689,5712 -after ((get-date).addDays(-1))   
\# 文件共享，文件系统，SAM，注册表，证书时间   
PS C:\ Get-EventLog Security 4671,4691,4698,4699,4700,4701,4702,5148,5149,5888,5889,5890,4657,5039,4659,4660,4661,4663,4656,4658,4690,4874,4875,4880,4881,4882,4884,4885,4888,4890,4891,4892,4895,4896,4898,5145,5140,5142,5143,5144,5168,5140,5142,5143,5144,5168,5140,5142,5143,5144,5168,4664,4985,5152,5153,5031,5140,5150,5151,5154,5155,5156,5157,5158,5159 -after ((get-date).addDays(-1))   
\# 查看Eent ID：4672的详细信息   
Get-Eventlog Security | ? { $\_.Eventid -eq 4672} | Format-List   
2.）LINUX   
\# 认证日志   
tail /var/log/auth. log   
grep -i "fail" /var/log/auth. log   
tail /var/log/secure   
grep -i "fail" /var/log/secure   
\# samba，cron,sudo相关日志   
grep -i samba /var/log/syslog   
grep -i samba /var/log/messages   
grep -i cron /var/log/syslog   
grep -i sudo /var/log/auth. log   
grep -i sudo /var/log/secure   
\# Apache 404错误日志   
grep 404 apache.log | grep -v -E "favicon.ico|robots.txt"   
\# 监控新文件，5分钟刷新一次   
watch -n 300 -d ls -lR /web\_root

4. ) 响应（取证）

**4.1 ) LIVE TRIAGE（收集运行时系统信息）**

**4.1.1 WINDOWS系统篇**   
  
1.）系统信息   
C:\> echo %DATE% %TIME%   
C:\> hostname   
C:\> systeminfo   
C:\> systeminfo | findstr /B /C:"OS Name" /C:"OS Version"   
C:\> wmic csproduct get name   
C:\> wmic bios get serialnumber   
C:\> wmic computersystem list brief   
C:\> psinfo -accepteula -s -h -d   
2.）用户信息   
C:\> whoami   
C:\> net users   
C:\> net localgroup administrators   
C:\> net group administrators   
C:\> wmic rdtoggle list   
C:\> wmic useraccount list   
C:\> wmic group list   
C:\> wmic netlogin get name,lastlogon,badpasswordcount   
C:\> wmic netclient list brief   
C:\> doskey /history > history.txt   
3.）网络信息   
C:\> netstat -e   
C:\> netstat -naob   
C:\> netstat -nr   
C:\> netstat -vb   
C:\> nbtstat -s   
C:\> route print   
C:\> arp -a   
C:\> ipconfig /displaydns   
C:\> netsh winhttp show proxy   
C:\> ipconfig /allcompartments /all   
C:\> netsh wlan show interfaces   
C:\> netsh wlan show all   
C:\> reg query "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings\Connections\WinHttpSettings"   
C:\> type %SYSTEMROOT%\system32\drivers\etc\hosts   
C:\> wmic nicconfig get descriptions,IPaddress,MACaddress   
C:\> wmic netuse get name,username,connectiontype, localname   
4.）服务信息   
C:\> at   
C:\> tasklist   
C:\> tasklist /svc   
C:\> tasklist /SVC /fi "imagename eq svchost.exe"   
C:\> tasklist /SVC /fi "imagename eq svchost.exe"   
C:\> schtasks   
C:\> net start   
C:\> sc query   
C:\> wmic service list brief | findstr "Running"   
C:\> wmic service list conf ig   
C:\> wmic process list brief   
C:\> wmic process list status   
C:\> wmic process list memory   
C:\> wmic job list brief   
PS C:\> Get-Service | Where-Object { $\_.Status -eq "running" }   
5.）策略、补丁、环境变量信息   
C:\> set   
C:\> gpresult /r   
C:\> gpresult /z > output.txt   
C:\> gpresult /H report.html /F   
C:\> wmic qfe   
6.）自启动信息   
C:\> wmic startup list full   
C:\> wmic ntdomain list brief   
6.1）检查自启动文件目录   
C:\> dir "%SystemDrive%\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"   
C:\> dir "%SystemDrive%\Documents and Settings\All Users\Start Menu\Programs\Startup"  
C:\> dir %userprofile%\Start Menu\Programs\Startup   
C:\> %ProgramFiles%\Startup\   
C:\> dir C:\Windows\Start Menu\Programs\startup   
C:\> dir "C:\Users\%username%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup"   
C:\> dir "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"   
C:\> dir "%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup"   
C:\> dir "%ALLUSERSPROFILE%\Microsoft\Windows\Start Menu\Programs\Startup"   
C:\> dir "%ALLUSERSPROFILE%\Start Menu\Programs\Startup"   
C:\> type C:\Windows\winstart.bat   
C:\> type %windir%\wininit.ini   
C:\> type %windir%\win.ini   
C:\> type C:\Autoexec.bat"   
6.2）使用autoruns   
C:\> autorunsc -accepteula -m   
6.3）自启动注册表位置   
HKEY\_CLASSES\_ROOT:   
C:\> reg query HKCR\Comfile\Shell\Open\Command   
C:\> reg query HKCR\Batfile\Shell\Open\Command   
C:\> reg query HKCR\htafile\Shell\Open\Command   
C:\> reg query HKCR\Exefile\Shell\Open\Command   
C:\> reg query HKCR\Exefiles\Shell\Open\Command   
C:\> reg query HKCR\piffile\shell\open\command   
HKEY\_CURRENT\_USERS:   
C:\> reg query "HKCU\Control Panel\Desktop"   
C:\> reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run"  
C:\> reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Run"   
C:\> reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Runonce"   
C:\> reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnceEx"   
C:\> reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\RunServices"   
C:\> reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce"   
C:\> reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Windows\Run"   
C:\> reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Windows\Load"   
C:\> reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Windows\Scripts"   
C:\> reg query "HKCU\Software\Microsoft\WindowsNT\CurrentVersion\Windows" /f run   
C:\> reg query "HKCU\Software\Microsoft\WindowsNT\CurrentVersion\Windows" /f load   
C:\> reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run"  
C:\> reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs"   
C:\> reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedMRU"   
C:\> reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComD1g32\OpenSaveMRU"   
C:\> reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMRU"  
C:\> reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComD1g32\OpenSavePidlMRU" /s  
C:\> reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU"   
C:\> reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders"   
C:\> reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders"   
C:\> reg query "HKCU\Software\Microsoft\Windows\CurrentVersion\Applets\RegEdit" /v LastKey   
C:\> reg query "HKCU\Software\Microsoft\InternetExplorer\" TypedURLs   
C:\> reg query "HKCU\Software\Policies\Microsoft\Windows\ControlPanel\Desktop"   
HKEY\_LOCAL\_MACHINE:   
C:\> reg query "HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components" /s   
C:\> reg query "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\explorer\User Shell Folders"   
C:\> reg query "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\explorer\Shell Folders"   
C:\> reg query "HKLM\Software\Microsoft\Windows\CurrentVersion\explorer\ShellExecuteHooks"   
C:\> reg query "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Browser Helper Objects" /s   
C:\> reg query "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run"  
C:\> reg query "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"   
C:\> reg query "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Runonce"   
C:\> reg query "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx"   
C:\> reg query "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices"   
C:\> reg query "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServicesOnce"   
C:\> reg query "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Winlogon\Userinit"   
C:\> reg query "HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\shellServiceObjectDelayLoad"   
C:\> reg query "HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Schedule\TaskCache\Tasks" /s   
C:\> reg query "HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Windows"   
C:\> reg query "HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Windows" /f Appinit\_DLLs   
C:\> reg query "HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Winlogon" /f Shell   
C:\> reg query "HKLM\SOFTWARE\Mic rosoft\WindowsNT\CurrentVersion\Winlogon" /f Userinit   
C:\> reg query "HKLM\SOFTWARE\Policies\Microsoft\Windows\Systern\Scripts"   
C:\> reg query "HKLM\SOFTWARE\Classes\batfile\shell\open\cornrnand"   
C:\> reg query "HKLM\SOFTWARE\Classes\cornfile\shell\open\cornrnand"   
C:\> reg query "HKLM\SOFTWARE\Classes\exefile\shell\open\command"   
C:\> reg query "HKLM\SOFTWARE\Classes\htafile\Shell\Open\Command"   
C:\> reg query "HKLM\SOFTWARE\Classes\piffile\shell\open\command"   
C:\> reg query "HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Explorer\Browser Helper Objects" /s   
C:\> reg query "HKLM\SYSTEM\CurrentControlSet\Control\SessionManager"   
C:\> reg query "HKLM\SYSTEM\CurrentControlSet\Control\SessionManager\KnownDLLs"   
C:\> reg query "HKLM\SYSTEM\ControlSet001\Control\SessionManager\KnownDLLs"   
7.）取日志文件   
C:\> wevtutil epl Security C:\bak\Security-logs.evtx   
C:\> wevtutil epl System C:\bak\System-logs.evtx   
C:\> wevtutil epl Application C:\bak\Application-logs.evtx   
8.）文件、目录、共享信息   
C:\> net use \\目标IP   
C:\> net share   
C:\> net session   
C:\> wmic volume list brief   
C:\> wmic logicaldisk get description,filesystem,name,size   
C:\> wmic share get name,path   
\# 查找多个类型的文件或某个文件   
C:\> dir /A /S /T:A \*.exe \*.dll \*.bat \*.PS1 \*.zip   
C:\> dir /A /S /T:A evil.exe   
\# 查找2017/1/1之后创建的文件   
C:\> forfiles /p C:\ /M \*.exe /S /D +2017/1/1 /C "cmd /c echo @fdate @ftime @path"   
C:\> for %G in (.exe, .dll, .bat, .ps) do forfiles -p "C:" -m \*%G -s -d +2017/1/1 -c "cmd /c echo @fdate @ftime @path"   
\# 查找文件大小>20MB的文件   
forfiles /S /M \* /C "cmd /c if @fsize GEQ 2097152 echo @path @fsize"   
\# 在Alternate Data Streams中查找文件   
C:\> streams -s 文件或目录   
\# 检查数字签名，vt扫描   
C:\> sigcheck -e -u -vr -s C:\   
C:\> listdlls.exe -u   
\# 扫描病毒   
C:\> "C:\Program Files\Windows Defender\MpCmdRun.exe" -SignatureUpdate   
C:\> "C:\Program Files\Windows Defender\MpCmdRun.exe" -Scan

**4.2 ) LIVE TRIAGE（收集运行时系统信息）**

**4.2.1 LINUX篇**   
  
1.）系统信息   
uname -a   
uptime   
timedatectl   
mount   
2.）用户信息   
w   
lastlog   
last   
faillog -a   
cat /etc/passwd   
cat /etc/shadow   
cat /etc/group   
cat /etc/sudoers   
\# 查找UID为0的用户   
awk -F: '($3 == "0") {print}' /etc/passwd   
egrep ':0+' /etc/passwd   
cat /root/.ssh/authorized\_keys   
lsof -u root   
cat /root/.bash\_history   
3.）网络信息   
\# 查看网络接口   
ifconfig OR ip a l   
\# 查看监听端口   
netstat -tupnl   
\# 查看网络连接   
netstat -tupnla   
netstat -tupnlax   
\# 路由信息   
route OR netstat -r OR ip r l   
\# ARP表   
arp -ne   
\# 监听端口的进程   
lsof -i   
4.）服务信息   
\# 列出所有进程   
ps aux OR ps -ef   
\# 已加载内核模块   
lsmod   
\# 打开的文件   
lsof   
lsof -c sshd   
lsof -p PID   
lsof -nPi | cut -f1 -d" " | uniq | tail -n +2   
\# 监控日志   
less +F /var/log/messages   
tail -F /var/log/messages   
journalctl -u ssh.service -f   
\# 列出所有服务   
chkconfig --list   
systemctl list-units   
5.）策略、补丁、环境变量信息   
\# 检查pam.d目录相关文件   
cat /etc/pam.d/common\*   
\# 自启动信息 - 计划任务   
crontab -l   
crontab -u root -l   
cat /etc/crontab   
ls /etc/cron,\*   
6.）命令历史   
cat /root/.\*history   
7.）文件、目录、共享信息   
df -ah   
ls -lhcta /etc/init.d/   
stat -x filename   
file filename   
\# 特殊属性文件   
lsattr -R / | grep "\-i-"   
\# 全局可写文件   
find / -xdev -type d \( -perm -0002 -a ! -perm -1000 \) -print   
\# 某时间点之后新建的文件   
find / -newermt 2018-01-22q   
\# 打印文件的所有属性信息   
find /labs -printf "%m;%Ax;%AT;%Tx;%TT;%Cx;%CT;%U;%G;%s;%p\n"   
\# 查看文件的元数据   
stat 文件名   
8.) 简单基线检查   
wget https://raw.githubusercontent.com/pentestmonkey/unix-privesc-check/1\_x/unix-privesc-check && ./unix-privesc-check > output.txt   
9.) 检测rootkit   
chkrootkit   
rkhunter --update && rkhunter -check   
tiger && less /var/log/tiger/security.report.\*   
lynis && lynis audit system && more /var/logs/lynis. log   
10.) Fastir Collector Linux，收集artefacts，包括：内核版本、内核模块、网卡、系统版本、主机名、登录、网络连接、SSH know\_host、日志文件、进程数据、自启动等信息   
wget https://raw.githubusercontent.com/SekoiaLab/Fastir\_Collector\_Linux/master/fastIR\_collector\_linux.py  
python fastIR\_collector\_linux.py --debug --output\_dir output   
11.) Sysdig and Sysdig Falco 行为监控   
\# 观察root用户查看过的目录   
sysdig -p"%evt.arg.path" "evt.type=chdir and user.name=root"   
\# 观察SSHD行为   
sysdig -A -c echo\_fds fd.name=/dev/ptmx and proc.name=sshd   
\# id为5459的登录shell执行过的所有命令   
sysdig -r trace.scap.gz -c spy\_users proc.loginshellid=5459   
\# 安装，启动falco   
curl -s https://s3.amazonaws.com/download.draios.com/DRAIOS-GPG-KEY.public | apt-key add -   
curl -s -o /etc/apt/sources.list.d/draios.list http://download.draios.com/stable/deb/draios.list   
sudo apt update   
apt -y install falco   
modprobe sysdig-probe   
service falco start   
falco

**4.3 ) 病毒样本分析**   
  
\# 静态分析   
\# 挂载Sysinternals工具集   
\\live.sysinternals.com\tools   
\# 检查数字签名   
C:\> sigcheck.exe -u -e C:\malware   
C:\> sigcheck.exe -vt malware.exe   
\# 16机制和ASCII方式查看PE文件   
hexdump -C -n 500 malware.exe   
od -x mailware.exe   
xxd malware.exe   
strings -a malware.exe | more   
\# 内存镜像分析   
python vol.py -f malware\_memory\_dump.raw -profile=Win7SPFix64 malfind -D /output   
python vol.py -f malware\_memory\_dump.raw -profile=Win7SPFix64 malfind -p PID -D /output   
python vol.py -f malware\_memory\_dump.raw -profile=Win7SPFix64 pslist   
python vol.py -f malware\_memory\_dump.raw -profile=Win7SPFix64 pstree   
python vol.py -f malware\_memory\_dump.raw -profile=Win7SPFix64 dlllist   
python vol.py -f malware\_memory\_dump.raw -profile=Win7SPFix64 dlldump -D /output   
\# HASH分析   
curl -v --request POST --url https://www.virustotal.com/vtapi/v2/file/report' -d apikey=VT API KEY -d 'resource=样本文件hash'   
curl -v -F 'file=malware.exe' -F apikey=VT API KEY>https://www.virustotal.com/vtapi/v2/file/scan   
whois -h hash,cymru.com 样本文件hash   
\# 获取磁盘和内存镜像   
\# WINDOWS   
C:\> psexec.exe \\IP -u <DOMAIN>\administrator -p 123 -c mdd\_l.3.exe --o C:\memory.dmp   
C:\> dc3dd.exe if=\\.\c: of=d:\diskiamge.dd hash=md5 log=d:\output.log   
\# LINUX   
dd if=/dev/fmem of=/tmp/mem\_dump.dd   
\# 使用LiME   
get https://github.com/504ensicslabs/LiME/archive/master.zip   
unzip master.zip   
cd LiME-master/src   
make   
cp lime-\*.ko /media/USB/   
insmod lime-3.13.0-79-generic.ko "path=/media/USB/mem\_dump.lime format= raw"   
\# 从内存中拷贝PE文件   
cp /proc/进程ID/exe /output   
\# 创建进程core dump   
gcore 进程ID   
strings -a gcore.\* | more   
dd if=/dev/sda of=/root/sda.dd   
dd if=/dev/sda | ssh root@RemoteIP "dd of=/root/sda.dd"   
\# 通过netcat传送接收镜像文件   
bzip2 -c /dev/sda | nc 8.8.8.8 53   
nc -p 53 -l | bzip2 -d | dd of=/root/sda.dd

5. ) 常用技巧（TIPS & TRICKS）

**5.1 ) 技巧**

**5.1.1 WINDOWS系统篇**   
  
\# 将命令结果通过管道输出到粘帖板，然后将粘帖板的内容重定向到文件   
C:\> some\_command.exe | clip   
PS C:\> Get-Clipboard > clip.txt   
\# 检查注册表某路径是否存在   
PS C:\> Test-Path "HKCU:\Software\Microsoft\123"   
\# 可靠文件复制   
robocopy c:\src \\目标计算机\dst /E   
\# 检查某目录是否存在ps1,vbs扩展的文件   
PS C:\> Test-Path C:\Scripts\Archive\\* -include \*.ps1, \*.vbs   
\# 合并多个文件   
C:\> type 1.txt 2.txt > output.txt   
\# 多个桌面窗口（Desktops）   
C:\>"%ProgramFiles%\Internet Explorer\iexplore.exe" https://live.sysinternals.com/desktops.exe   
\# 在远程计算机执行命令   
C:\> psexec.exe \\远程计算机 -u admin -p 123 /c c:\123.exe   
PS C:\> Invoke-Command -远程计算机 { ls }   
\# 比较两个文件的差异   
PS C:\> Compare-Object (-Content 1.log) -DifferenceObject (Get-Content 2.log)   
\# 进制转换与编码   
C:\> set /a 0xff   
PS C:\> 0xff   
C:\> certutil -decode BASE64编码文件 output.file   
\# 解码XOR，搜索关键字：http   
C:\> xorsearch.exe -i -s input.file http

**5.1.2 LINUX系统篇**   
  
\# 通过ssh在远程服务器上抓包   
ssh root@8.8.8.8 tcpdump -i any -U -s 0 -w - 'not port 22'

**5.2 ) SNORT**   
# SNORT规则检测Meterpreter   
  
\# Snort rules by Didier Stevens (http://DidierStevens.com)   
alert tcp $HOME\_NET any -> $EXTERNAL\_NET $HTTP\_PORTS (msg:"Metasploit Meterpreter"; flow:to\_server,established; content:"RECV"; http\_client\_body; depth:4; fast\_pattern; isdataat:!0,relative; urilen:23<>24,norm; content:"POST"; pcre:"/^\/[a-z0-9]{4,5}\_[a-z0-9]{16}\/$/Ui"; classtype:trojan-activity; reference:url,blog.didierstevens.com/2015/05/11/detecting-network-traffic-from-metasploits-meterpreter-reverse-http-module/; sid:1618008; rev:1;)   
https://didierstevens.com/files/software/snort-rules-V0\_0\_1.zip   
\# SNORT规则检测PSEXEC   
alert tcp $HOME\_NET any -> $HOME\_NET [139,445] (msg:"POLICY-OTHER use of psexec remote administration tool"; flow:to\_server,established; content:"|FF|SMB|A2|"; depth:5; offset:4; content:"|5C 00|p|00|s|00|e|00|x|00|e|00|c|00|s|00|v|00|c"; nocase; metadata:service netbios-ssn; reference:url,technet.microsoft.com/en-us/sysinternals/bb897553.aspx; classtype:policy-violation; sid:24008; rev:1;)   
alert tcp $HOME\_NET any -> $HOME\_NET [139,445] (msg:"POLICY-OTHER use of psexec remote administration tool SMBv2"; flow:to\_server,established; content:"|FE|SMB"; depth:8; nocase; content:"|05 00|"; within:2; distance:8; content:"P|00|S||E|00|X|00|E|00|S|00|V|00|C|00|"; fast\_pattern:only; metadata:service netbios-ssn; reference:url,technet.microsoft.com/en-us/sysinternals/bb897553.aspx; classtype:policy-violation; sid:30281; rev:1;)   
  
**5.3 ) Bro NSM**   
# 检测横向渗透   
  
wget https://raw.githubusercontent.com/richiercyrus/Bro-Scripts/master/detect-mal-smb-files.bro   
bro -r faf-exercise.pcap detect-mal-smb-files.bro   
less notice.log   
\# 检测勒索软件   
wget https://raw.githubusercontent.com/fox-it/bro-scripts/master/smb-ransomware/smb-ransomware.bro   
bro -r faf-exercise.pcap smb-ransomware.bro

**5.4 ) 检测DOS/DDOS**   
# 检测攻击类型SYN Flood，ICMP Flood，UDP Flood   
  
tshark -r 001.pcap -q -z io,phs   
tshark -c 1000 - -z io,phs   
tcpdump -tnr $ | awk -F '.' '{print $1"."$2"."$3"."$4}' | sort | uniq -c | sort -n | tail   
tcpdump -qnn "tcp[tcpflags] & (tcp-syn) != 0"   
netstat -s   
tcpdump -nn not arp and not icmp and not udp   
netstat -n | awk '{print $6}' | sort | uniq -c | sort -nr | head   
\# 应用层   
tshark -c 10000 -T fields -e http.host | sort | uniq -c | sort -r | head -n 10   
tshark -r capture6 -T fields -e http.request.full\\_uri | sort | uniq -c | sort -r | head -n 10c   
tcpdump -n 'tcp[32:4] = 0x47455420' | cut -f 7- -d":"   
\# 查找http请求中包含：GIF,ZIP,JPEG,PDF,PNG扩展的数据包   
tshark -Y "http contains "ff:d8"" || "http contains "GIF89a"" || "http contains "\x50\x4B\x03\x04"" || "http contains "\xff\xd8"" || "http contains "%PDF"" || "http contains "\x89\x50\x4E\x47""   
取'user-agent'和refer字段   
tcpdump -c 1000 -Ann I grep -Ei 'user-agent' | sort | uniq -c | sort -nr | head -10   
tcpdump -i en0 -A -s 500 | grep -i refer   
\# 第二层攻击   
tcpdump 'arp or icmp'   
tcpdump -tnr 001.pcap ARP | awk -F '.' '{print $1"."$2"."$3"."$4}' | sort | uniq -c | sort -n | tail   
tshark -r 001.pcap -q -z io,phs | grep arp.duplicate-address-detected

**5.5 ) 蓝队兵器谱**   
  
6.4.1） KALI 渗透测试发行版   
https://www.kali.org   
6.4.2）SIFT SANS 取证工具箱   
http://sift.readthedocs.org/   
6.4.3）REMNUX 软件逆向和病毒分析发行版   
https://remnux.org   
6.4.4）OPENVAS   
http://www.openvas.org   
6.4.5）Security Onion 入侵检测、网络安全监控、日志分析发行版   
https://securityonion.net   
6.4.6）OSSEC 开源主机入侵检测系统   
http://ossec.github.io