2017 VFsec团队

应急响应，从懵逼到入门

JE2Se

应急响应的形式

远程应急响应服务

本地应急响应服务

网站入侵应急

互相伤害

主机入侵应急

网站篡改
网站瘫痪
页面挂马
…

木马病毒
后门攻击
异常登陆
…

事件分类

1.Web入侵：挂马、篡改、Webshell

2.系统入侵：系统异常、RDP爆破、SSH爆破、主机漏洞

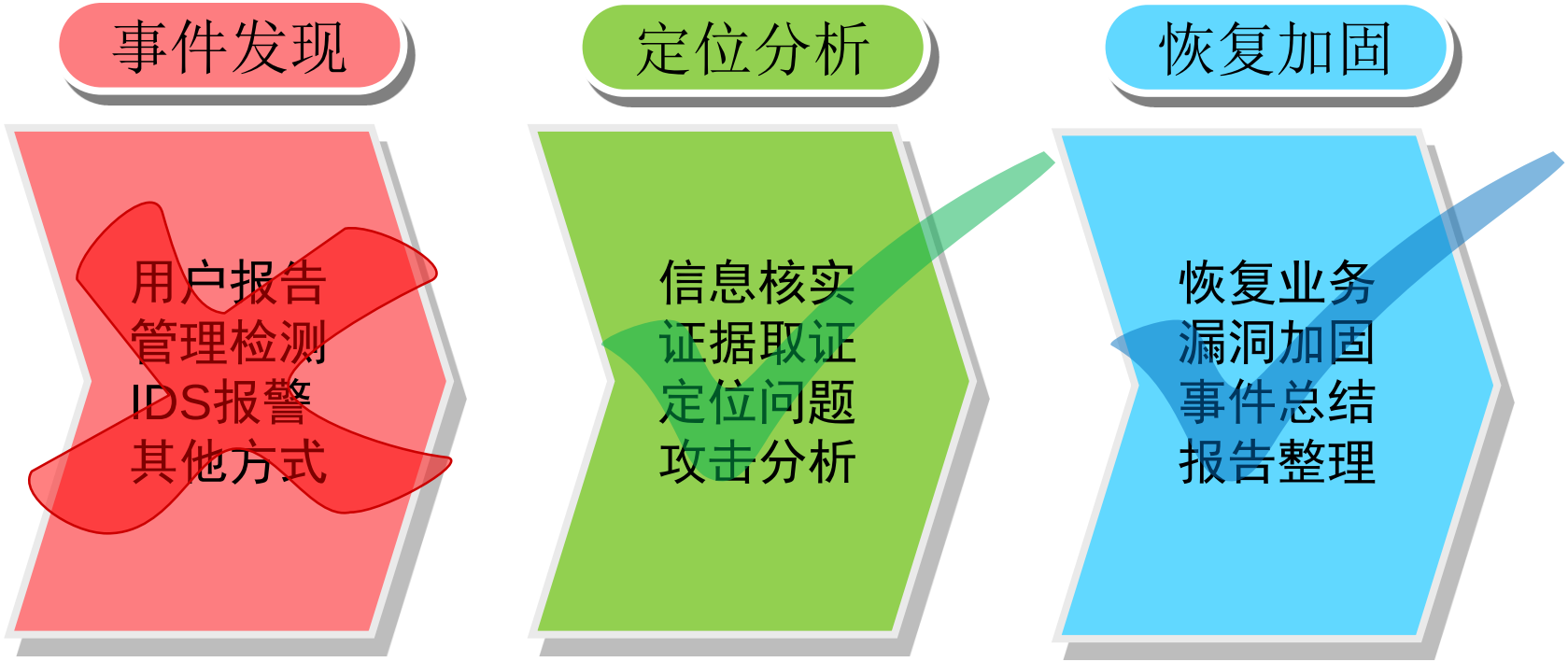
3.病毒木马：远控、后门、勒索软件

4.信息泄漏：刷库、数据库登录（弱口令）

5.网络流量：频繁发包、批量请求、DDOS攻击

VFsec

应急响应一般流程



如何做应急响应

- 确定攻击时间
- 查找攻击线索
- 梳理攻击流程
- 实施解决方案
- 定位攻击者

入侵信息核实

- 明确入侵网址/主机详情
- 跟踪事件发现人
- 了解事件发生特性
- 核实网络结构或系统框架
- 确定事件发生时间
- 知悉事件发生后处理办法
- 记录相关人员联系方法

排查思路

文件分析

文件日期、新增文件、可疑/异常文件、最近使用文件、浏览器下载文件

Webshell 排查与分析，核心应用关联目录文件分析

进程分析

当前活动进程 & 远程连接，启动进程&计划任务，服务

服务系统信息

环境变量/账号信息/History/系统配置文件

日志分析

操作系统日志

VFsec

数据收集

- 查看账户信息
- 检查补丁情况
- 查看系统日志
- 查看注册表/服务（Win）
- 查看用户连接状况
- 查看账户登录状况
- 搜索近期修改文件
- 查看网站日志
- 检查数据库修改情况
- 查看进程
- 检查防护设备日志

net user　　cat /etc/passwd

Systeminfo　　uname -a

运行-eventvwr　　/var/log/message 系统启动后的信息和错误日志

/var/log/secure　与安全相关的日志信息

/var/log/maillog　与邮件相关的日志信息

/var/log/cron　与定时任务相关的日志信息

运行-regedit / services.msc　　/var/log/spooler UUCP和news设备相关日志信息

/var/log/boot.log 进程启动和停止相关的日志消息

netstat　　netstat

quser　　who / last

.bash_history　命令行历史记录

用眼睛/工具（lchangedfiles）　　find / -ctime -1 -print

应用服务log目录　　应用服务log目录

数据库日志

任务管理器　　ps -aux

没错，就是查看它的日志

linux注意的文

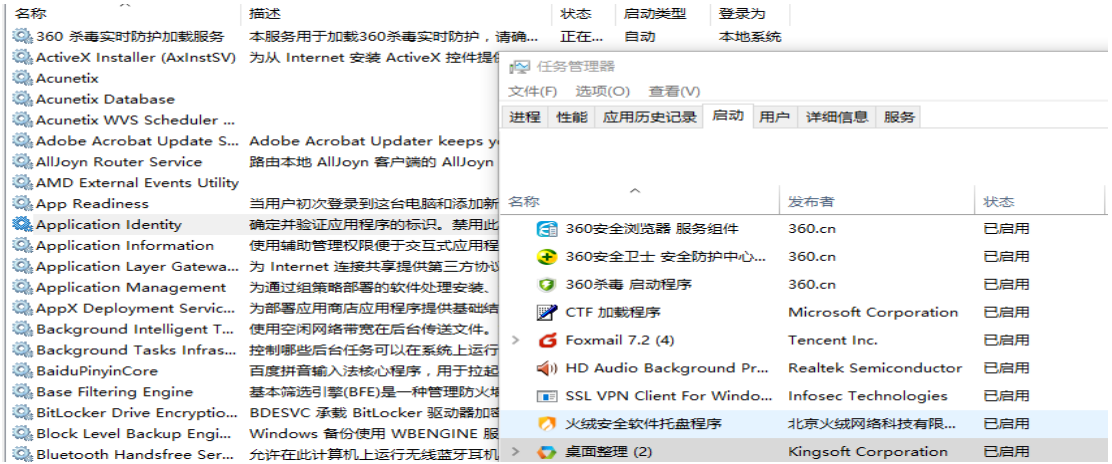
常用命令

信息收集 留意

1.在查询用户的过程中要留意隐藏账户的信息

HKEY_LOCAL_MACHINE\SAM\SAM\Domains\Account\Users\Names

2.判断是否为恶意进程或者服务部分可通过描述或者发布者进行判断



Apache、tomcat、Nginx、IIS的Web日志类

无论任何web服务器其实日志需要关注的东西是一致的，即access_log和error_log。

Mysql MSSQL数据库类

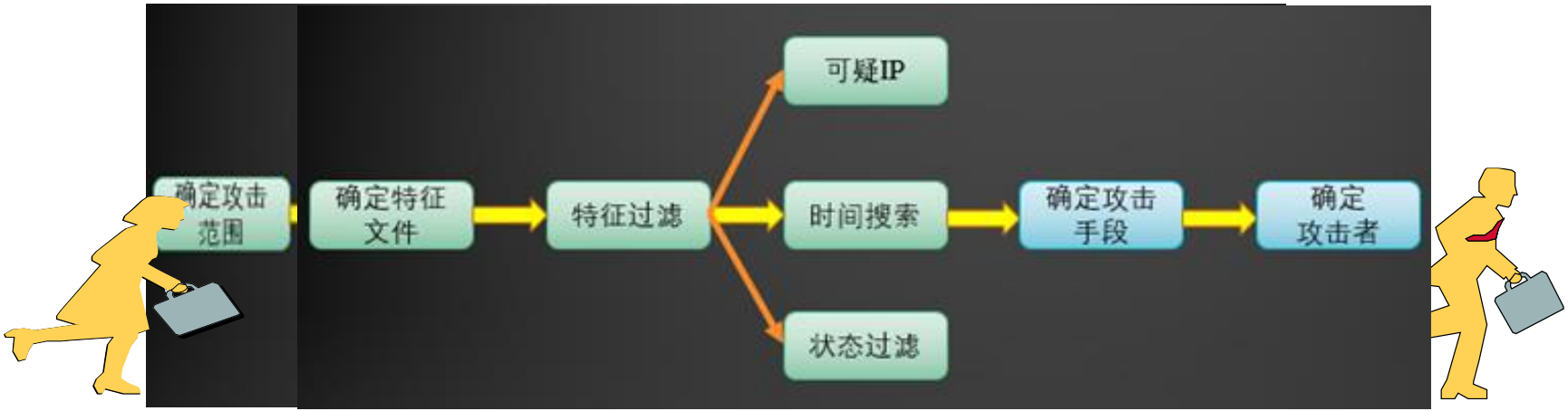
检查mysql\lib\plugin目录没有发现异常文件（参考UDF提权）

Mysql : select * from mysql.func

MSSQL，检查xp_cmdshell等存储过程正常与否

在对WEB日志进行安全分析时，按照下面两种思路展开逐步深入，还原整个攻击过程。

三.确定攻击范围



Windows下常用的工具

工具	主要功能
PCHunter/火绒剑	可查看进程、内核、服务等
Dos命令	查看信息
wireshark	分析数据流量
日志安全分析工具	日志安全分析工具能够对日志进行安全分析，可快速从日志中发现可疑的恶意攻击行为
D盾 /河马/杀毒软件	可以检查服务器指定目录中可能存在的**Webshell**文件以及恶意文件

指定漏洞的检测工具

VFsec

Linux下常用的工具

工具	主要功能
Dos命令	查看信息
Chkrootkit/rootkit hunter	查找检测**rootkit**后门的工具

web日志分析脚

Grep /find命令

- 查找一句话木马（<?php eval($_post[cmd]);?>

假设网站的目录为/app/website/，我们需要查看该目录下是否包含该形式的一句话木马文件：

方法1：

$ grep -i –r eval\(\$_post /app/website/*

其中-i表示不区分大小写，-r表示搜索指定目录及其子目录

方法2:

$find /app/website/ -type f|xargs grep eval\(\$_post

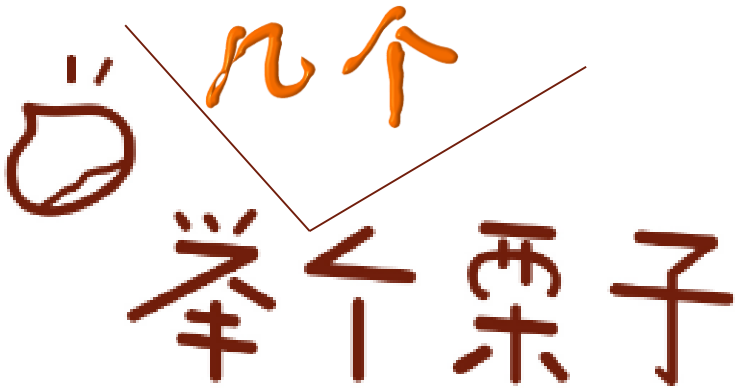
xargs 将find搜索出的文件名称变成 grep后面需要的参数

针对收集到的账户信息、文件修改情况、系统日志、网站日志等进行综合分析和归纳，确认是否存在以下情况：

- 系统含有非法账号
- 系统中含有异常服务程序
- 系统部分文件被篡改，或发现有新的文件
- 系统安全日志中有非正常登陆情况
- 网站日志中有非授权地址访问管理页面记录

数据分析

- 通过异常文件的创建和修改时间，一般可以判断攻击者对网站进行入侵的时间段；
- 对异常服务或进程的追踪，可以查找恶意文件，确认攻击后的后门，以及攻击时间；
- 网站目录下的异常文件，对判断攻击手段具有参考意义；
- 网站访问日志可以对攻击手段、时间和攻击源地址的追踪提供有力的证据。
- 系统安全日志中的登录信息同样可以用于判断攻击者来源。

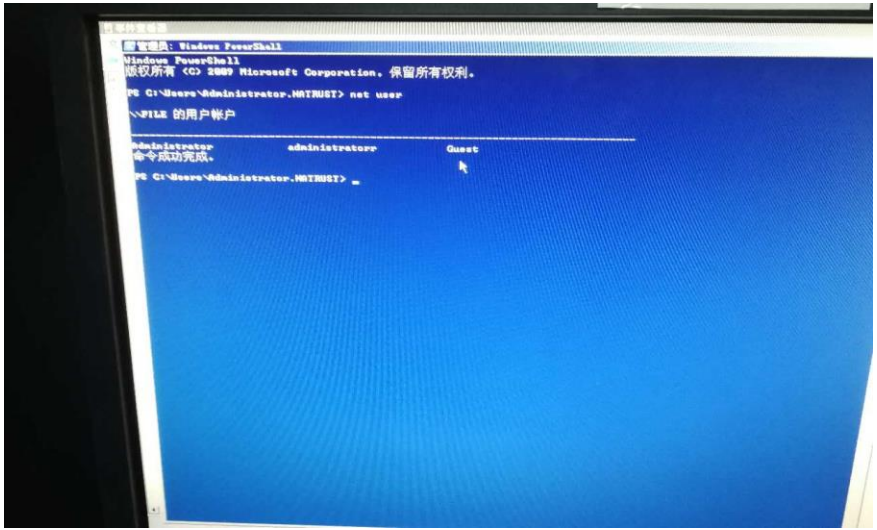

几个
举个栗子

第一个栗子

接到某客户通知，一台主机遭到入侵，并且安装恶意软件，该IP地址为192.168.10.237，不停的向国内某一IP进行DDOS攻击。并且在网站目录存放了攻击工具

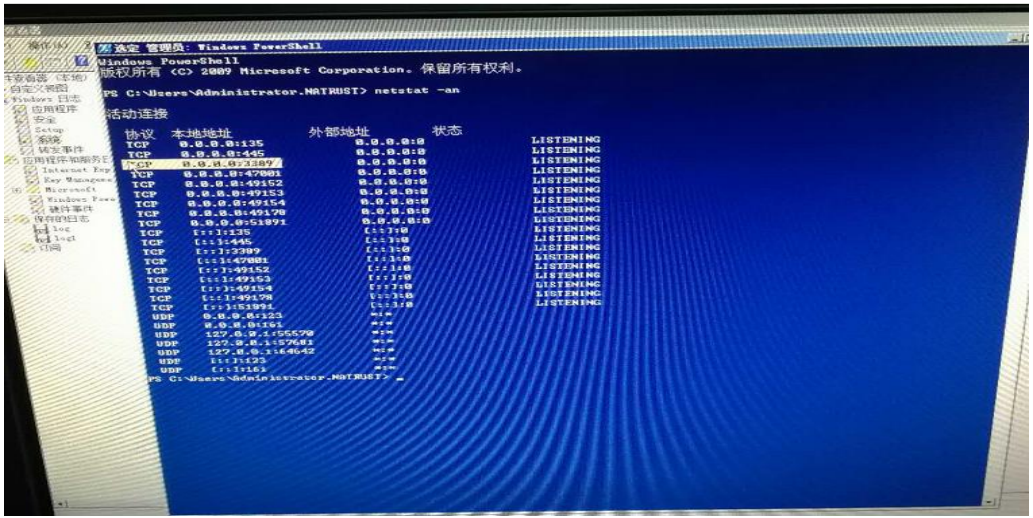
第一个栗子

首先检查了目标的是否存在恶意进程，服务，启动项，无影响，在用户信息发现情况，发现一个可疑用户administratorr用户。



第一个栗子

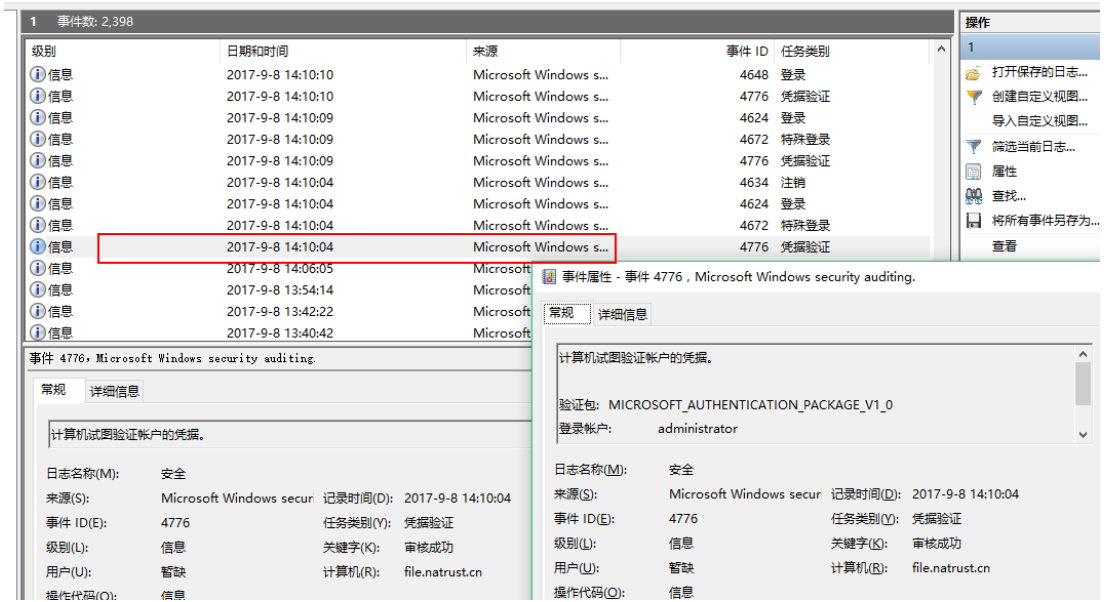
接下来检测目标机器开放端口的情况，发现该机器开放了3389端口。



VFsec

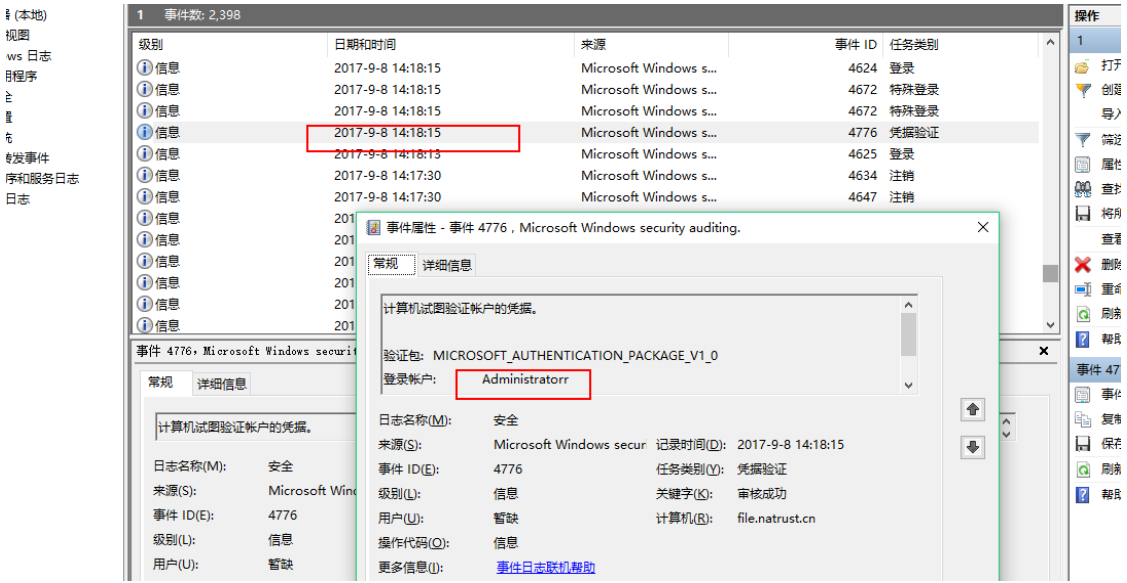
第一个栗子

检查windows安全日志，发现12点42分在登陆日志上发现存在暴力破解行为。于2点10分成功登陆到administrator用户。



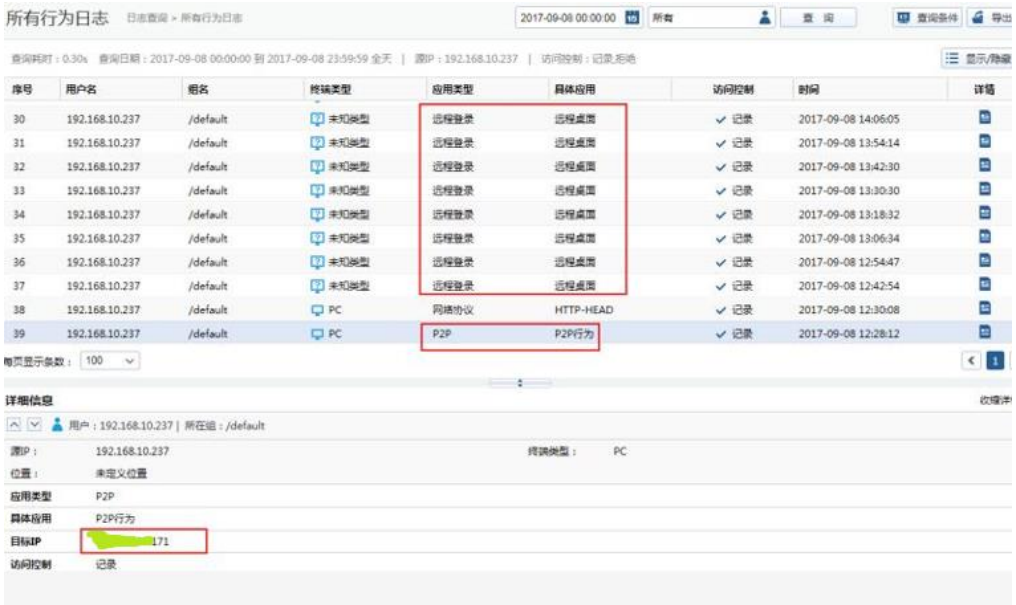
第一个栗子

同时进行了添加用户的操作，加入管理员组，并进行了新用户登录。



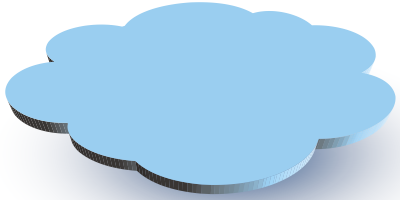
第一个栗子

在行为管理器的日志上查看，通过新创建的用户，P2P远程下载了攻击软件.

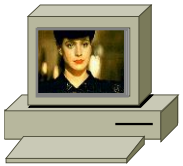


第二个栗子

接到某客户通知，他们遭到了入侵，并且被删除了一组数据，相关的数据库日志等等进行了清除处理。



IP：192.168.1.100
测试环境



IP：192.168.2.100
核心部署环境

SSH

隔离

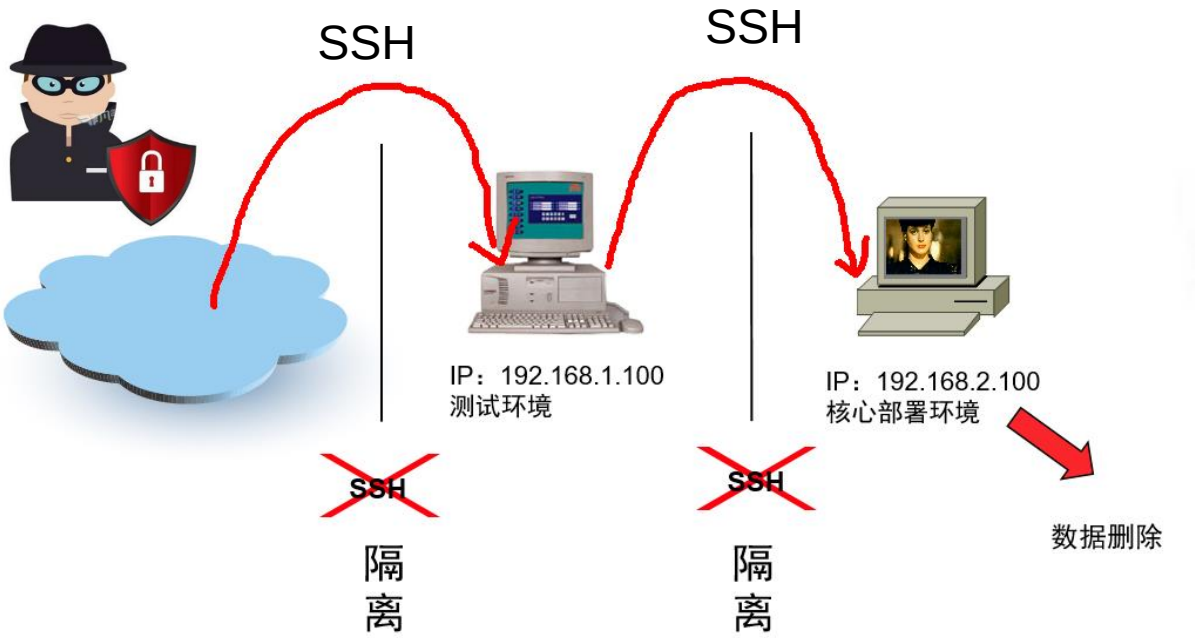
SSH

隔离

数据删除

第二个栗子

现有证据，存在secure.log，日志显示上，攻击为如下



VFsec

第二个栗子

我们有策略，外网无法ssh内网，1网段无法ssh2网段是不是外网入侵，内网渗透提权啥的

IP：
192.168.1.100
测试环境

SSH

隔离

IP：
192.168.2.100
核心部署环境

SSH

隔离

数据删除

第二个栗子

从外网通过SSH连接到测试服务器

在测试服务器上
的.history记录上发

```
rm -rf core.20362
ls
ls -lrt
tar -zcvf QY.tar.gz QY/
ls -lrt
sz QY.tar.gz
ssh sxit@10.0.102.174
ssh sxit@10.0.102.175
ssh sxit@10.0.102.176
ssh sxit@10.0.102.177
ps -fe | grep QY
ssh sxit@10.0.102.175
ssh sxit@10.0.102.174
ssh sxit@10.0.102.176
ssh sxit@10.0.102.177
ssh sxit@10.0.102.177
ssh sxit@10.0.102.176
ssh sxit@10.0.102.175
ssh sxit@10.0.102.174
last
history
 last
exit
ssh sxit@10.0.102.178
```

VFsec 第二个栗子

```
Aug 31 09:44:55 localhost passwd: gkr-pam: couldn't update the 'login' keyring password: no old password was entered
Aug 31 09:47:03 localhost sshd[12724]: Did not receive identification string from 10.0.104.124
Aug 31 10:02:03 localhost sshd[28520]: Did not receive identification string from 10.0.104.124
Aug 31 10:17:03 localhost sshd[10825]: Did not receive identification string from 10.0.104.124
Aug 31 10:32:03 localhost sshd[26320]: Did not receive identification string from 10.0.104.124
Aug 31 10:47:03 localhost sshd[9619]: Did not receive identification string from 10.0.104.124
Aug 31 11:02:03 localhost sshd[24996]: Did not receive identification string from 10.0.104.124
Aug 31 11:17:03 localhost sshd[8038]: Did not receive identification string from 10.0.104.124
Aug 31 11:28:21 localhost sshd[1990]: Accepted password for sxit from 10.0.104.43 port 39862 ssh2
Aug 31 11:28:21 localhost sshd[1990]: pam_unix(sshd:session): session opened for user sxit by (uid=0)
Aug 31 11:32:03 localhost sshd[23020]: Did not receive identification string from 10.0.104.124
Aug 31 11:47:03 localhost sshd[4017]: Did not receive identification string from 10.0.104.124
Aug 31 11:53:01 localhost sshd[4518]: pam_unix(sshd:session): session closed for user sxit
Aug 31 11:54:46 localhost sshd[14135]: Accepted password for sxit from 10.0.104.124 port 61722 ssh2
Aug 31 11:54:46 localhost sshd[14135]: pam_unix(sshd:session): session opened for user sxit by (uid=0)
Aug 31 12:02:01 localhost sshd[16930]: Did not receive identification string from 10.0.104.124
Aug 31 12:17:01 localhost sshd[30675]: Did not receive identification string from 10.0.104.124
Aug 31 12:32:01 localhost sshd[13072]: Did not receive identification string from 10.0.104.124
Aug 31 12:45:41 localhost sshd[5266]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.0.104.43  user=root
Aug 31 12:45:43 localhost sshd[5266]: Failed password for root from 10.0.104.43 port 54368 ssh2
Aug 31 12:45:50 localhost sshd[5277]: Connection closed by 10.0.104.43
Aug 31 12:46:01 localhost sshd[13458]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=10.0.104.43  user=sxit
Aug 31 12:46:03 localhost sshd[13458]: Failed password for sxit from 10.0.104.43 port 55351 ssh2
Aug 31 12:46:08 localhost sshd[13458]: Accepted password for sxit from 10.0.104.43 port 55351 ssh2
Aug 31 12:46:08 localhost sshd[13458]: pam_unix(sshd:session): session opened for user sxit by (uid=0)
Aug 31 12:47:01 localhost sshd[27817]: Did not receive identification string from 10.0.104.124
Aug 31 12:47:24 localhost sshd[1548]: Received disconnect from 10.0.104.43: 11: disconnected by user
Aug 31 12:47:24 localhost sshd[13458]: pam_unix(sshd:session): session closed for user sxit
Aug 31 12:47:30 localhost sshd[1276]: Accepted password for sxit from 10.0.104.43 port 57764 ssh2
Aug 31 12:47:30 localhost sshd[1276]: pam_unix(sshd:session): session opened for user sxit by (uid=0)
Aug 31 12:52:13 localhost sshd[2224]: Received disconnect from 10.0.104.43: 11: disconnected by user
Aug 31 12:52:13 localhost sshd[1276]: pam_unix(sshd:session): session closed for user sxit
Aug 31 12:52:24 localhost sshd[3944]: Accepted password for sxit from 10.0.104.43 port 37528 ssh2
Aug 31 12:52:24 localhost sshd[3944]: pam_unix(sshd:session): session opened for user sxit by (uid=0)
Aug 31 13:02:01 localhost sshd[9132]: Did not receive identification string from 10.0.104.124
Aug 31 13:17:01 localhost sshd[21726]: Did not receive identification string from 10.0.104.124
Aug 31 13:32:02 localhost sshd[2226]: Did not receive identification string from 10.0.104.124
Aug 31 13:43:30 localhost sshd[5419]: Received disconnect from 10.0.104.43: 11: disconnected by user
Aug 31 13:43:30 localhost sshd[3944]: pam_unix(sshd:session): session closed for user sxit
Aug 31 13:47:02 localhost sshd[15294]: Did not receive identification string from 10.0.104.124
```

其中的连接外网连接IP为172.168.1.1，时间发生在9点28分，成功接入内部。

通过secure的连接日志，发现登陆用户均为sxit用户，据了解。这是他们该应用系统的第三方外包开发使用的管理账号，而且是硬编码弱密码写在代码里面的。

第二个栗子

同时也检查了VPN的连接日志，发现172.168.1.1在8点53分曾经尝试通过VPN登陆。不过登录失败，之后9点28分通过ssh连接。Vpn登录失败的用户名正是外包开发人员。但是vpn账户在开发结束后就撤销了。

经济纠纷

顺藤摸瓜
足够沟通
证据断事

JE2Se

VFsec团队

VFsec

THANKS!

谢谢观看

Vfsec团队