远离应急，实现安全响应自动化运营

张嵩　　华泰证券信息安全总监

江旺　　华泰证券SRC负责人

ISC 互联网安全大会 中国 · 北京

Internet Security Conference 2018　Beijing · China

（原“中国互联网安全大会”）

目录

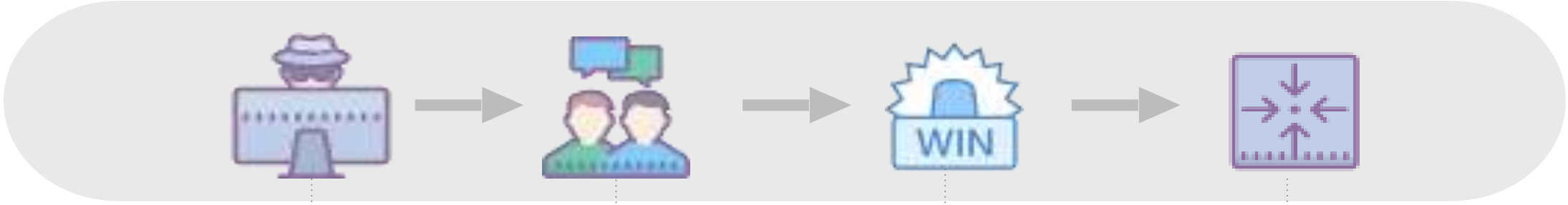
1、安全响应的现状和挑战

某次内网红蓝对抗暴露的问题

单一的EPP解决方案存在检测盲点

1、安全响应的现状和挑战（续）

红蓝对抗引发的思考



没有完美的防护
入侵事件始终存在

多维度的检测
纵深的分析研判

完善丰富的响应流程
不断降低的MTTR

持续循环迭代
形成闭环

目录

2、安全响应需要的数据和威胁情报的作用

安全响应依赖的基础数据与Kill-Chain各阶段对应

我被盯上了？	如何被搞的？				是不是被搞了？	谁搞的我？我被搞的多惨？
Reconnaissance	Weaponization	Delivery	Exploitation	Installation	Command & Control	Action & Objective
• 流量 • IPS、IDS告警 • 应用认证日志	• 邮件网关（如特定扩展名命中） • Sandbox告警 • PC的终端安全（含HIPS等）告警 • 互联网访问网关的威胁告警	• NGFW告警，WAF告警 • 墙后的南北向流量监控 • HIPS/HIDS告警 • 漏洞信息 • 主机OS日志和Sysmon/Osquery等实现的增强日志 • HTTP访问，应用（如登录、访问），邮件，AD日志等 • 入站FW日志或者入站流量监控日志 • 东西向流量	• 服务器上尽可能多的监控信息，如进程hash和行为 • HTTP访问日志、完整请求payload和响应payload的前150字节		• DNS日志 • 出站流量中的域名或IP访问，证书 • 出站FW日志	• 服务器上尽可能多的监控信息，如进程hash和行为

2、安全响应需要的数据和威胁情报的作用

Kill-Chain各阶段常用到的威胁情报数据—应急响应与安全分析的催化剂

我被盯上了？ | 如何被搞的？ | 是不是被搞了？ | 谁搞的我？我被搞的多惨？

Reconnaissance → Weaponization → Delivery → Exploitation → Installation → Command & Control → Action & Objective

- 收件人邮件地址
- 目标国家
- 目标行业
- 目标个体
- 扫描特征

- 算法
- 密钥
- 特定互斥量
- 执行流程
- 加解密方式
- 特定功能模块
- 对抗分析措施
- 源码工程路径
- 特定数据字串
- 语言编译环境
- 特定数字签名
- 组件组织架构
- 特别的错误

- 邮件名特征
- 邮件正文特征
- 目标邮件和地址
- 恶意代码进入方式
  - 鱼叉邮件
  - 水坑攻击
  - U盘
  - 主动渗透

- 特定特定事件的CVE、0-day
- 通用payload特征

- 主机特征：Mutex、写入的注册表项、文件名或路径等
- Yara rule
- 特定主机监控程序，如Sysmon/Osquery规则集
- Webshell特征
- 初始启动路径
- 持续启动方式
- 伪装正常模式

- 域名、URL、历史解析IP、WHOIS
- SSL证书
- 域名注册信息
- 域名使用偏好
- 域名命名偏好
- IP、IP反查域名、历史域名解析、RDNS
- IP所在ASN、地址位置
- 域名或IP信誉评级、标签、关联事件和关联通信样本
- 通信协议
- 后门工具
- 工具类型
- 工具配置
- 认证凭据

- 目标数据
- 打包方法
- 传输方法
- 破坏功能

目录

3、安全响应的支撑平台建设

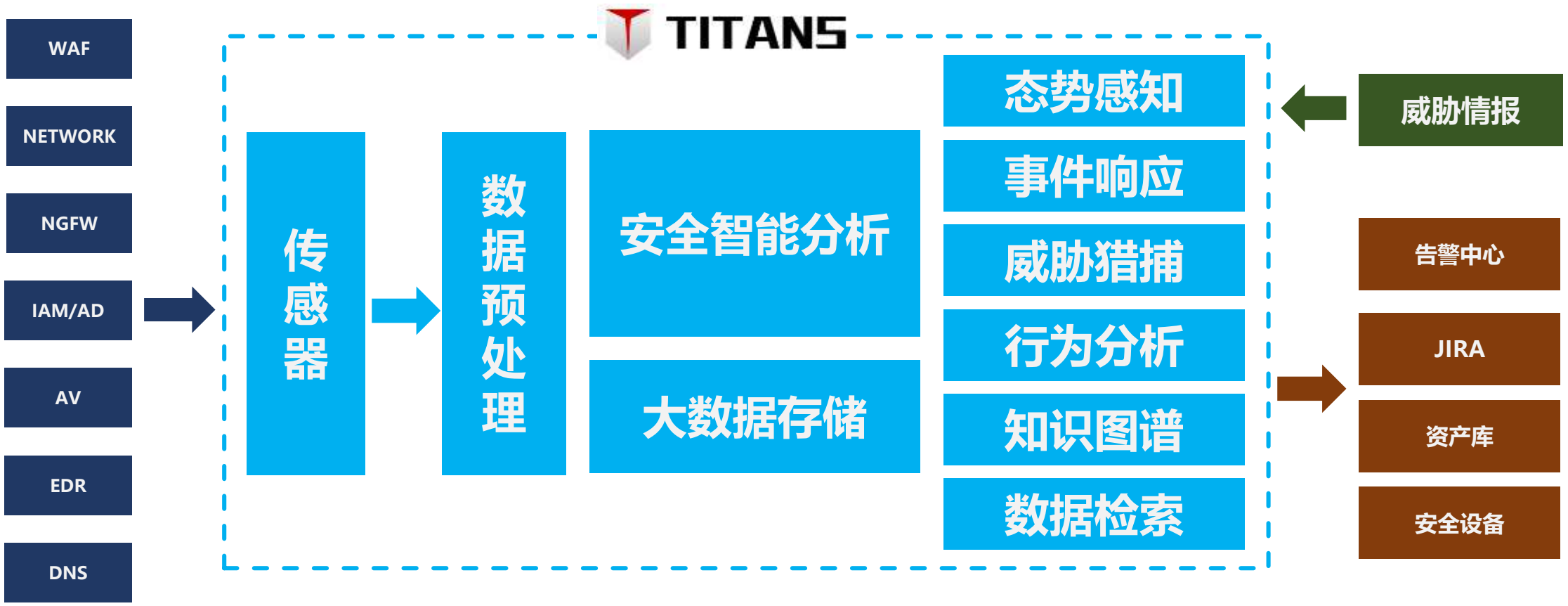
泰坦人工智能安全态势感知平台

TITANS 基于**大数据技术**，对企业全面的安全信息进行集中采集、存储和分析，利用流式计算、**智能分析引擎**、和**可视化**等手段，结合丰富的**威胁情报**，对企业面临的外部攻击、内部违规行为进行检测，为企业建立快速有效的威胁检测、分析、处置能力和全网**安全态势感知**能力，使得企业的信息安全可知、可见、可控。

大数据　　智能分析引擎　　威胁情报　　可视化

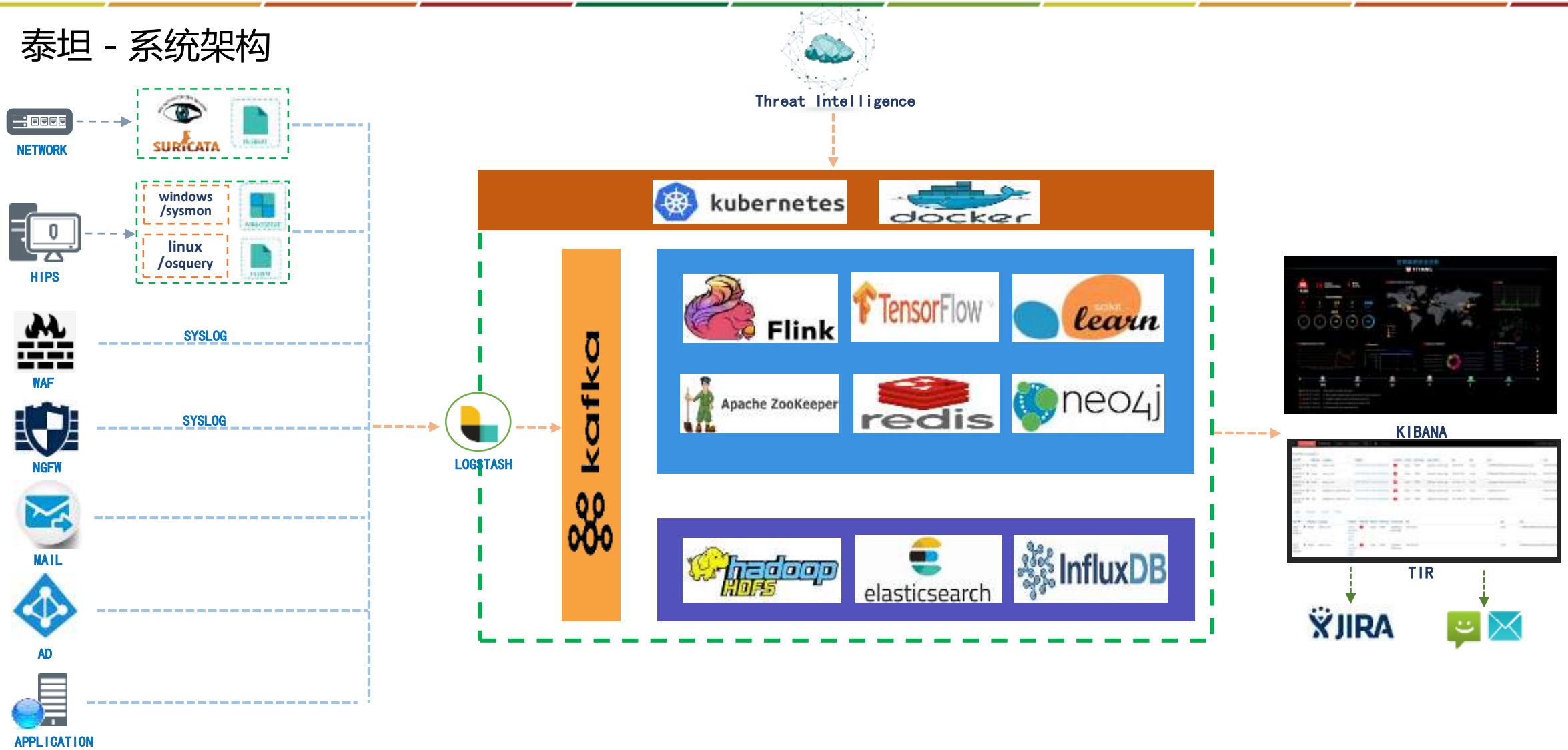
3、安全响应的支撑平台建设(续)

泰坦 - 系统架构



3、安全响应的支撑平台建设(续)

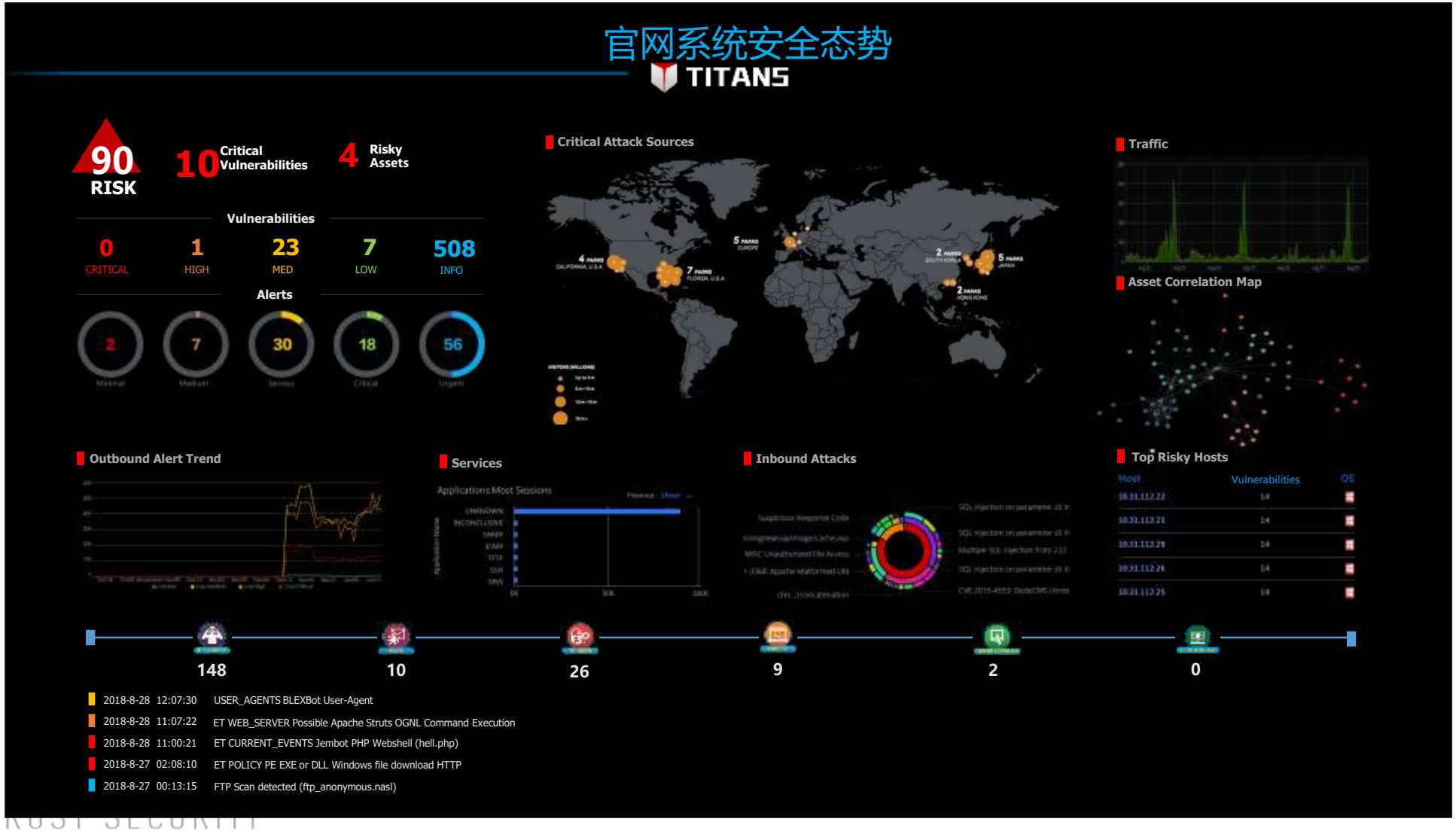
泰坦 - 系统架构



ZERO TRUST SECURITY

3、安全响应的支撑平台建设(续)

资产角度的安全可视化



目录

4、安全响应的“必修课”

（1） 例行漏洞与补丁运营与预警（难易度-低）

（2） 重大漏洞的应急响应（难易度-低）

（3） 安全监控与告警（难易度-中）

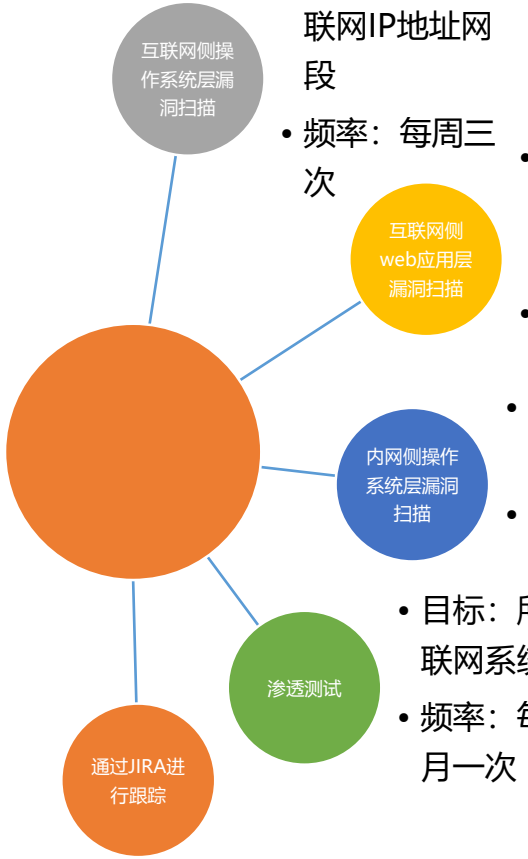
（4） 出站灰流量分析（难易度-中）

（5） 自动化关联分析（难易度-高）

（6） 红蓝对抗演习（难易度-高）

4、安全响应的“必修课”（续）

（1）例行漏洞与补丁运营与预警（难易度-低）



- 目标：所有互联网IP地址网段
- 频率：每周三次

- 目标：所有互联网web类应用
- 频率：每周1次

- 目标：所有内网服务器
- 频率：每周一次

- 目标：所有互联网系统
- 频率：每两个月一次

漏洞管理

微软补丁日通告机制

- 桌面终端补丁：自动化推送与安装+每月定时重启。
- 服务器补丁：互联网侧一个月内完成更新，内网侧一个季度内完成更新。
- 新上线服务器：物理机在交付时补丁即拉到最新，虚拟机每月对镜像做更新。

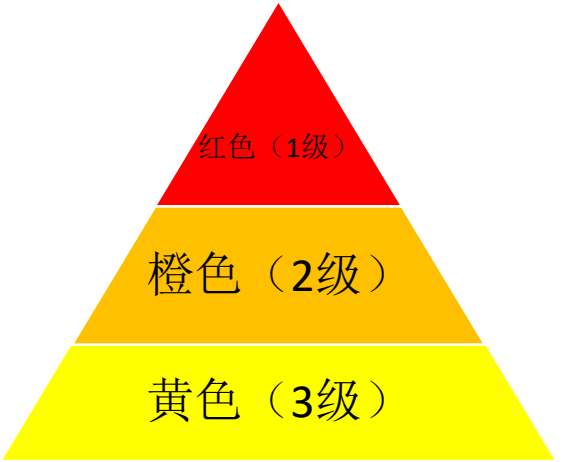
关注高危组件补丁级别

- 针对高危组件（如Weblogic、Jboss、websphere）的资产使用情况，制定专属仪表板；
- 根据官方通告，采取先互联网侧系统后内网侧的顺序安装补丁。

其他中间件

- 标准化：制定操作系统、中间件与组件的标准化版本并持续更新，与架构师团队共同推动执行。
- 新上线系统：一律与标准化版本保持一致。
- 已上线系统：每3个月拉齐一次。

补丁运营



黄色（3级）：公司某个别系统高可信度可以遭受攻击（但仍未）或正在遭受攻击，可导致个别系统攻陷。威胁或事件性质为孤立，预期影响范围为孤立系统。
橙色（2级）：公司大面积系统和信息资产可遭受攻击（但仍未）或正在遭受的针对个别系统的攻击正在朝着大面积蔓延的趋势发展，或多名员工遭受或感知攻击。威胁或事件性质为群体，预期影响范围为大面积系统或资产。
红色（1级）：公司正在遭受大面积攻击，可导致业务连续性影响或监管影响。该级别预警与信息技术部整体应急预案接轨，应适时启动公司级应急预案。

漏洞预警

4、安全响应的“必修课”

（2）重大漏洞的应急响应（难易度-低）

从厂商公告、安全情报厂商、安全社区和朋友圈等渠道获取漏洞情报。

根据漏洞影响的资产类型、公司资产数据库，确定受影响的服务器以及应该参与响应的系统管理员。

依据安全响应中心漏洞响应应急预案中的设定，确定本次漏洞响应级别为3级“黄色”。

通过邮件方式对全IT发布“橙色”预警。邮件正文直接给出已知受影响的系统名称、IP地址、人和团队等信息，同时要求其他人员开展自查，防止遗漏。

1、系统管理员根据官方给出的修复方式组织修复，
2、安全运营人员联系网络层IPS厂商获取IPS特征签名，并下发全网。

利用漏洞扫描器、POC脚本等方式对漏洞进行修复验证。
使用漏洞扫描器就该漏洞进行全网扫描。

通过邮件方式解除预警。

漏洞应急响应能力建设的几个关注点：

1、漏洞情报的获取。

2、精细化的资产管理，特别是高危组件的使用情况很重要。

3、应急响应流程的制定。

4、漏洞快速检测能力。

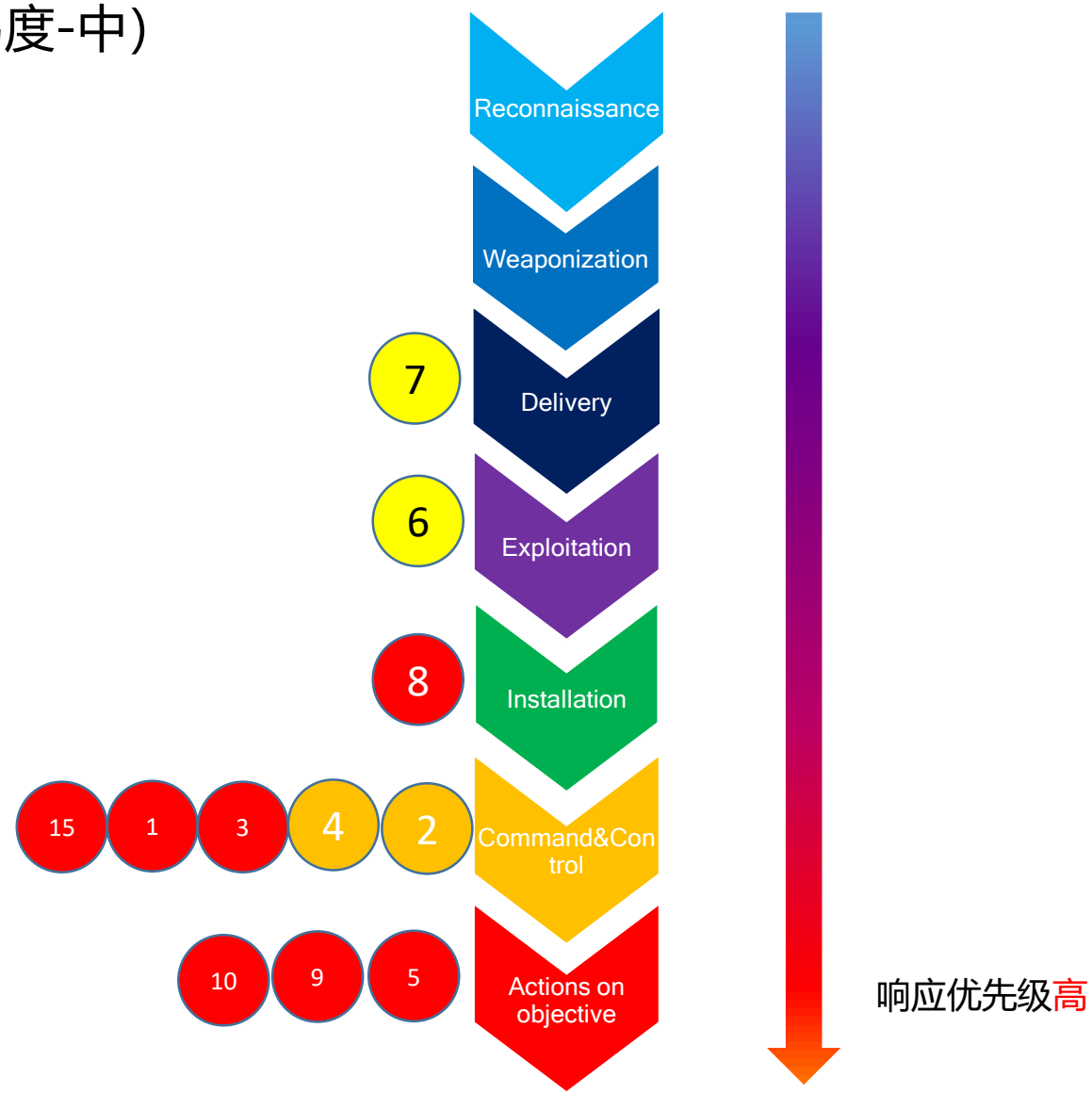
4、安全响应的“必修课”

（3）安全分级监控、告警与基于优先级响应（难易度-中）

安全告警类型、等级与Kill-Chain的关系

- **Intelligence类**
  - 1、出站恶意域名访问
  - 2、出站恶意IP访问
  - 3、办公网恶意DNS解析
  - 4、服务器区恶意DNS解析
  - 5、本地恶意文件hash
- **AV类**
  - 6、SEP终端IPS
  - 7、SEP终端病毒查杀（办公网）
  - 8、SEP终端病毒查杀（核心网）
- **EDR类**
  - 9、OSquery检测规则
  - 10、Sysmon检测规则
- **NIPS类**
  - 11、Suricata-ET PRO规则告警-高危
  - 12、Suricata-ET PRO规则告警-中危
  - 13、Suricata-ET PRO规则告警-低危
  - 14、Suricata-ET PRO规则告警-部分攻陷类相关-严重
- **AI类**
  - 15、DGA检测模型

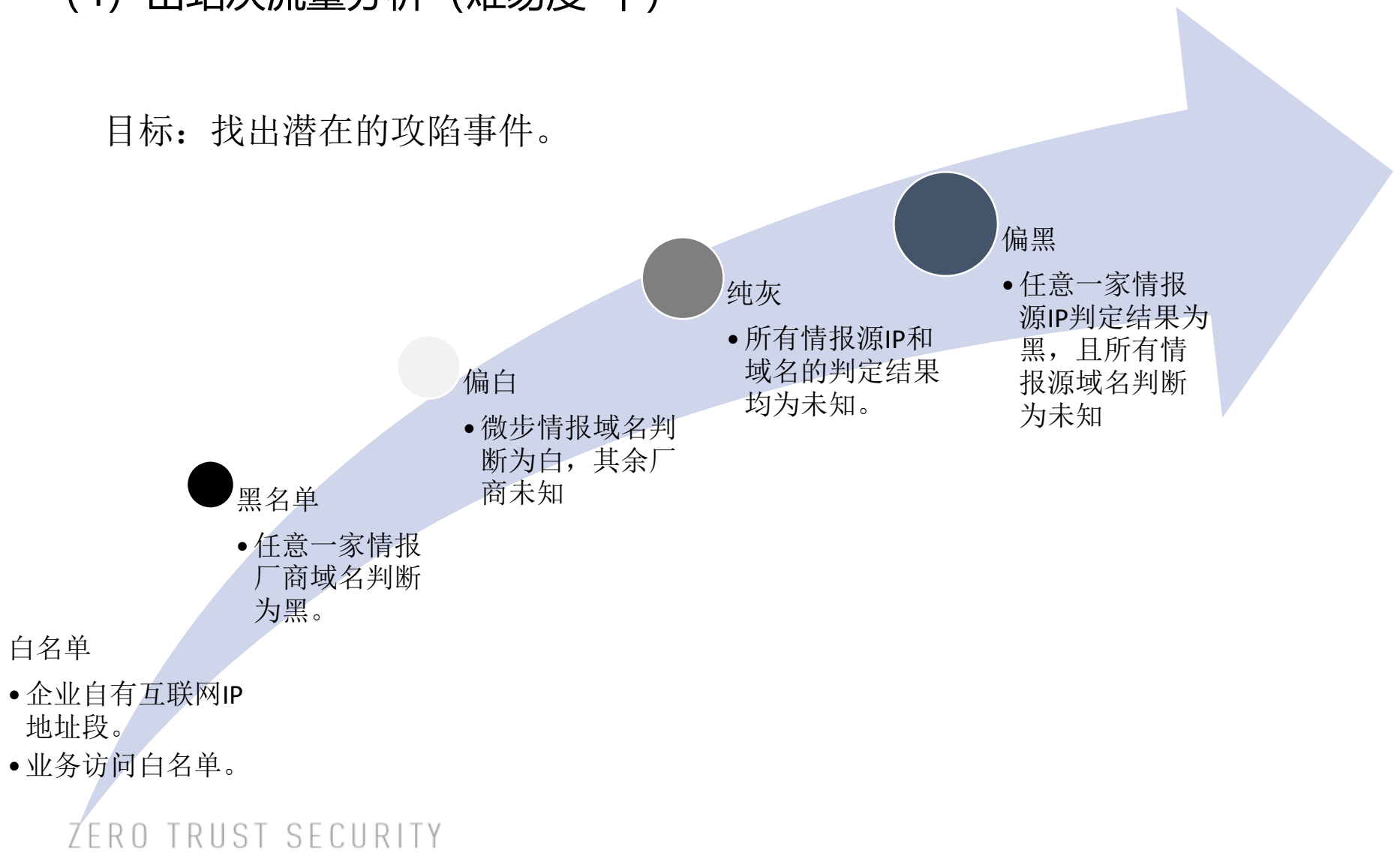
告警等级：严重 高危 中危 低危



4、安全响应的“必修课”

（4）出站灰流量分析（难易度-中）

目标：找出潜在的攻陷事件。



1、每月对公司威胁情报库中的纯灰连接进行人工判断，具体结合证书、whois、服务器管理员问询等进行综合判断，降低情报库中纯灰连接数量。

2、每天对本周新增的偏黑及黑的连接进行响应，查明事件原因，排除事件隐患。

4、安全响应的“必修课”

（5）自动化关联分析（难易度-高）

场景一：出站访问IP、域名、URL、DNS解析命中单一情报厂商IOC产生告警				
序号	关联元素	关联项目	关联项类型	目的
1	DIP、域名、URL、URI、DNS	该DIP、域名、URL、DNS解析在其他情报厂商上的信息。	情报	丰富情报信息。
2	DIP	该DIP在防火墙上的原始日志。	原始日志	判断出站连接是否成功。
3	DIP	该DIP的流量信息。	流量	查看疑似恶意流量的详情。
4	DIP	该DIP在EDR日志中的进程信息。	原始日志	查询该出站访问的进程名称。
5	HASH	动作4查询到进程的HASH多引擎威胁情报及沙箱检测结果。	情报	获取该进程的情报及沙箱分析结果。
6	SIP	该SIP的EPP告警。	规则类告警	判断该SIP服务器是否感染恶意程序。
7	SIP	该SIP的EDR告警。	规则类告警	判断该SIP服务器是否命中EDR规则。
8	SIP	该SIP的HIPS告警。	规则类告警	判断该SIP服务器是否有恶意网络行为。
9	SIP	该SIP的南北向NIDS告警。	规则类告警	判断该SIP服务器是否命中IDS规则。
10	SIP	该SIP的东西向NIDS告警。	规则类告警	判断该SIP服务器是否有横向移动行为。
11	SIP、域名	该SIP的DGA域名检测告警信息	AI类告警	判断该SIP是否发生过DGA域名解析+判断该域名是否为DGA域名。
12	SIP、DIP、域名、URL、DNS、HASH	该SIP和DIP的历史事件信息。	事件	查看历史事件库中是否有与该DIP、域名、URL、DNS、HASH解析和SIP相关的事件。

4、安全响应的“必修课” （续）

（5）自动化关联分析（难易度-高）（续）

场景二：针对高危资产的入站HTTP payload告警，例如weblogic反序列化漏洞				
序号	关联元素	关联项目	关联项类型	目的
1	SIP	该SIP在威胁情报厂商上的信息。	情报	丰富情报信息。
2	DIP	DIP服务器上的资产、版本及其漏洞信息	内部资产+漏洞信息	判断DIP上的资产及其版本是否与本次疑似攻击所利用的资产一致，以及是否存在本次疑似攻击所利用的漏洞。
3	SIP	该SIP的HTTP请求及其响应数据包	流量	查看该SIP所有的HTTP请求及其响应数据包的详细信息。UA、COOKIE、URI、POST DATA等。
4	DIP	DIP服务器上的EDR反弹shell或命令执行的告警	规则类告警	查看DIP机器上是否有反弹shell以及命令执行
5	SIP	SIP的NIDS Webshell上传告警信息	规则类告警	查看NIDS上是否有该SIP的webshell上传告警。
6	SIP	SIP的NIDS 命令执行告警信息	规则类告警	查看NIDS是否有该SIP的命令执行告警。
7	DIP	DIP服务器上的HIPS告警。	规则类告警	查看DIP上是否有恶意网络行为。
8	SIP、DIP	该SIP和DIP的历史事件信息。	事件	查看历史事件库中是否有与该DIP和SIP相关的事件。

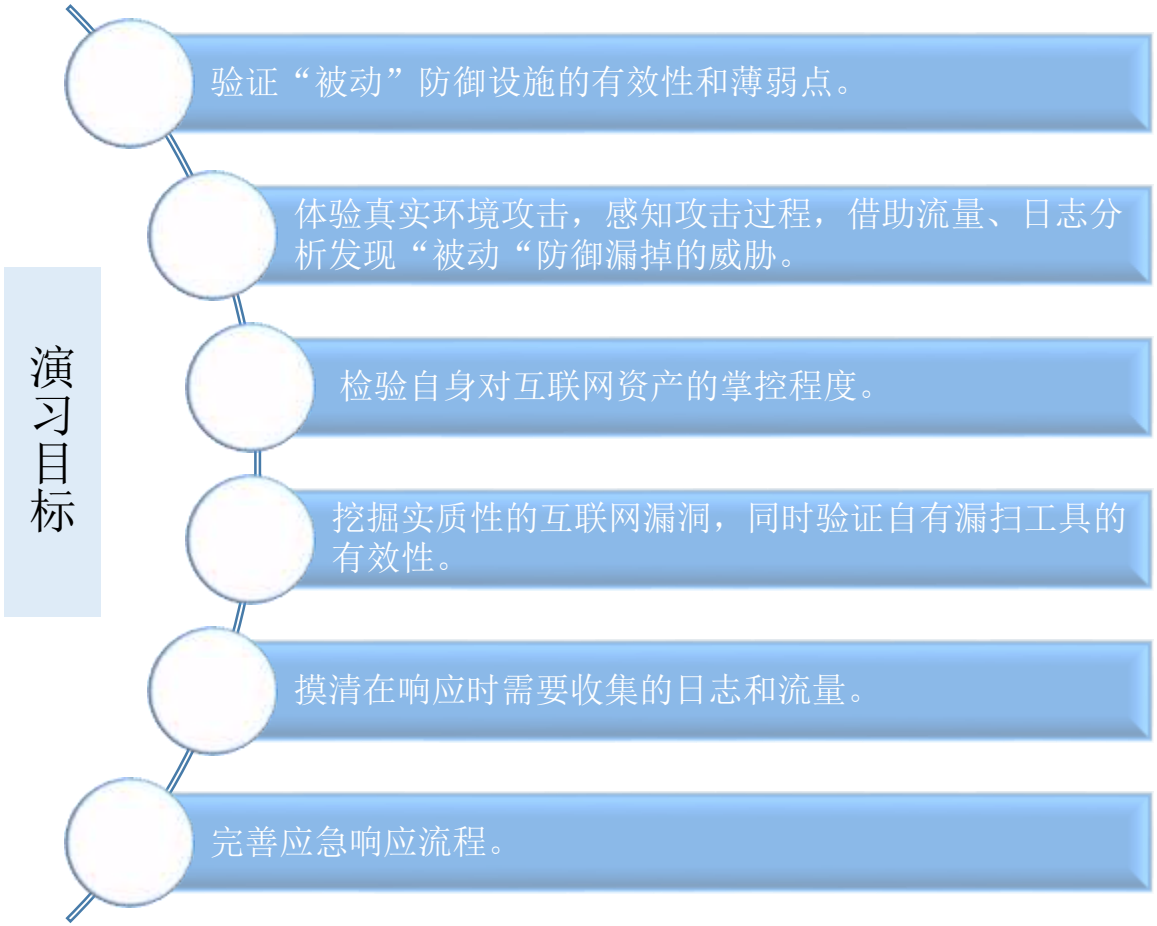
4、安全响应的“必修课”（续）

（5）自动化关联分析（难易度-高）（续）

场景三：EPP告警				
序号	关联元素	关联项目	关联项类型	目的
1	恶意文件HASH	恶意文件HASH的多引擎扫描结果	情报	丰富情报信息。
2	IP	该IP服务器上所有进程的nessus多引擎扫描结果。	Malware检测结果。	
3	IP、进程名	该IP服务器上恶意文件对应的相关进程的行为、网络连接、注册表等	原始日志	分析相关进程的行为。
4	IP	该IP服务器的EDR告警	规则类告警	查看该服务器上是否有其他恶意进程告警
5	IP	该IP服务器的出站的情报类告警	情报	查看该服务器是否有恶意出站访问。
6	IP	该IP服务器的HIPS告警	规则类告警	查看该服务器是否有横向的恶意访问。
7	IP	人工：使用THRECON对该服务器取证	规则类告警	获取该服务器的用户和组、进程、服务、注册表、网络行为、任务计划、日志以及补丁等必要信息，以便进行人工分析和判定。
8	IP、恶意文件HASH	该IP和恶意文件HASH的历史事件信息。	事件	查看历史事件库中是否有与该IP和恶意文件HASH相关的事件。

安全响应的“必修课”

（6）蓝对抗演习（难易度-高）



红军：
人员组成：SRC+安全服务公司技术专家
任务：1、策划与编排演习。
2、演习期间监测被动防御设备上的告警信息。
3、根据演习结果，优化防御性设备上的安全策略。

蓝军：
人员组成：众测平台的精英白帽子20+
任务：1、提供真实的高质量攻击流量。
2、找出具有实质性威胁的中高危漏洞。
3、演习结束后协助红军对攻击特征进行总结，形成IOC。

目录

终端检测响应/EDR主要工具集

SYSMON介绍（windows）

简介:

Sysmon是由Windows Sysinternals，当前最新版本为V8.0

下载链接:

https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon

主要功能:

监视和记录系统活动，包括进程创建、文件创建、网络连接等，并记录到Windows事件日志中。

安装:

sysmon –accepteula –i c:\windows\config.xml

卸载:

sysmon –u

更新配置文件:

sysmon –c c:\windows\config.xml

日志查看方法:

事件查看器-应用程序和服务日志-Microsoft-Windows-sysmon

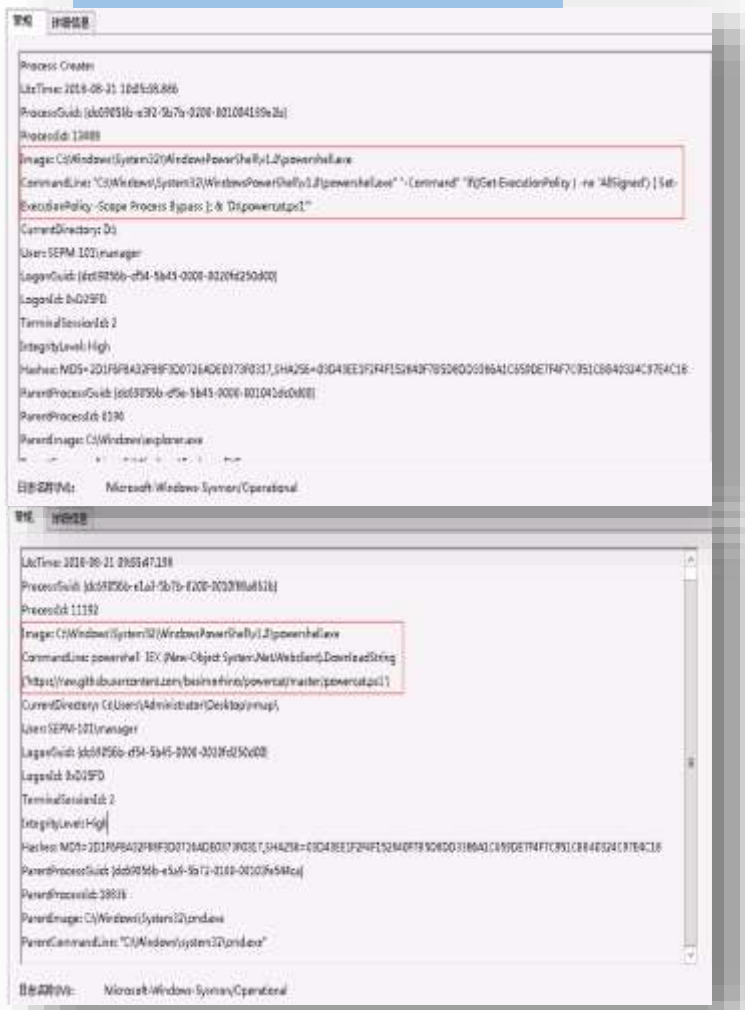
SYSMON的日志类别

Category	Event ID
Sysmon Service Status Changed	0
Process Create	1
File Creation Time Changed	2
Network Connection	3
Sysmon Service State Change	4
Process Terminated	5
Driver Loaded	6
Image Loaded	7
CreateRemoteThread	8
RawAccessRead	9
Process Access	10
File Create	11
Registry Object CreateDelete	12
Registry Value Create	13
Registry Object Rename	14
File Create Stream Hash	15
Sysmon Configuration Changed	16
Pipe Created	17
Pipe Connected	18
Error	255

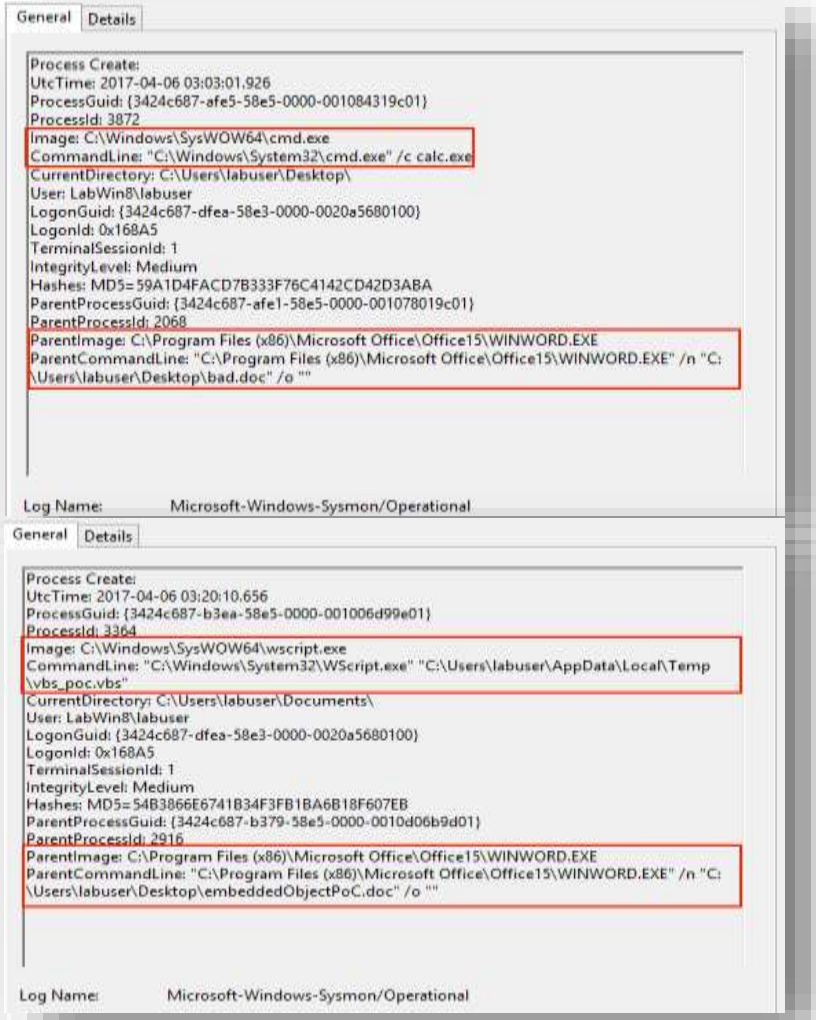
终端检测响应/EDR主要工具集（续）

SYSMON一些典型的应用场景

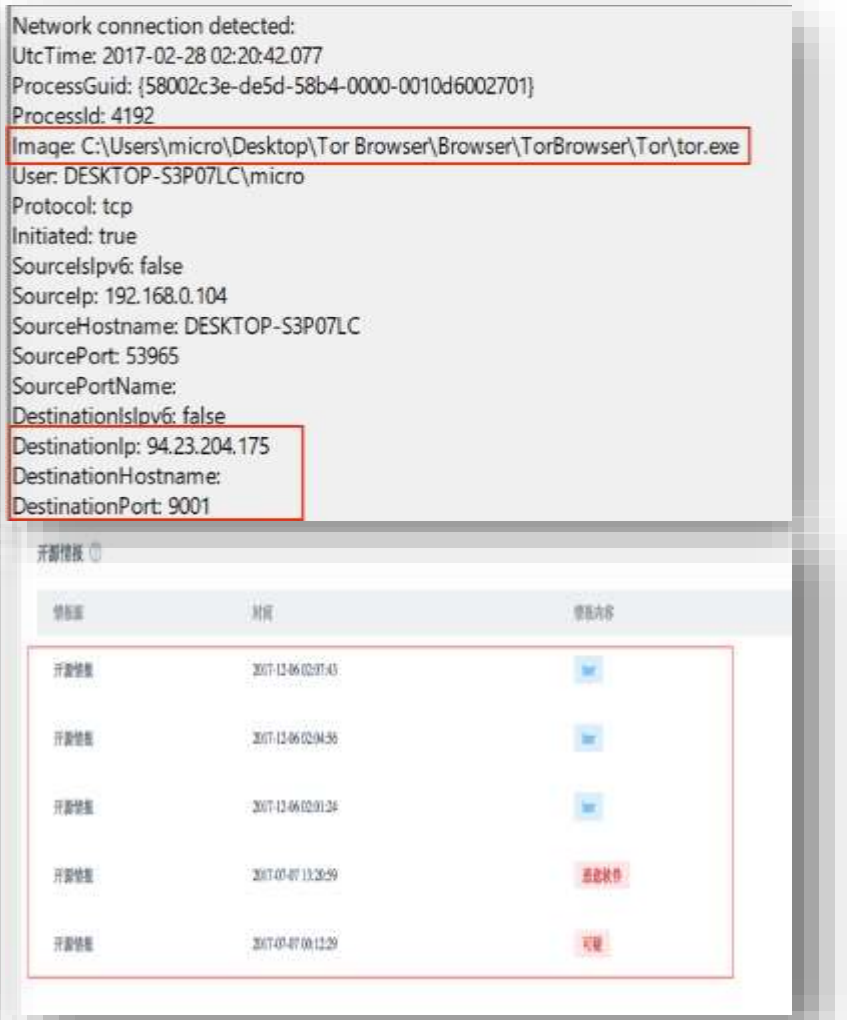
检测Powershell恶意执行



检测恶意word文档执行



检测恶意网络连接



终端检测响应/EDR主要工具集（续）

SYSMON实践建议

1|、推荐使用SwiftOnSecurity的公版配置文件进行mod（ https://github.com/SwiftOnSecurity/sysmon-config ）
2、使用开源规则集（ https://github.com/Neo23x0/sigma ）及自研规则集匹配日志内容进行告警。
3、实时采集，需要对各类事件做好进程过滤，以避免噪音过多。
4、文件hash日志与多源威胁情报进行关联告警
5、sysmon进程的保护和监控，日志文件及时上传集中存储。
6、日志采集检测规则已知，需要不断优化检测机制。
7、网络连接日志与流量采集、IDS、出入站情报等进行关联告警

终端检测响应/EDR主要工具集

THRecon介绍（windows）

简介：

THRecon是Github上一个开源的Threat Hunting**响应**工具，能够收集端点上的信息用于事件响应、 threat Hunting和现场取证。

链接： https://github.com/TonyPhipps/THRecon

安装：git clone https://github.com/TonyPhipps/THReconC:\Users\$env:UserName\Documents\WindowsPowerShell\Modules\THRecon

使用方法：

获取所有信息： Invoke-THR -Quick -Output c:\temp\ （收集的信息默认存储在c:\temp目录下）

获取主机的特定信息： Invoke-THR -Modules [Module1, Module2, etc.] -Output c:\temp\

例如：获取注册表地址信息， Invoke-THR -Modules registry

注意事项：

1、powershell需要以管理员身份运行。

2、使用前需要执行set-executionpolicy remotesigned.

3、扫描端需要powershell5.0或更高版本，被扫描端需要ps3.0或更高版本。

Host Info	Processes*	Services	Autoruns	Drivers
ARP	DLLs*	EnvVars	Hosts File	ADS
DNS	Strings*	Users & Groups	Ports	Select Registry
Hotfixes	Handles*	Sofware	Hardware	Event Logs
Net Adapters	Net Routes	Sessions	Shares	Certificates
Scheduled Tasks	TPM	Bitlocker	Recycle Bin	User Files

THRecon收集的信息类型

```
PS C:\tmp> Invoke-THR -Quick -all
Started Invoke-THR at 2018-08-29 23:53:14Z
Started Get-THR_ARP at 2018-08-29 23:53:14Z
Started Get-THR_Autoruns at 2018-08-29 23:53:14Z
Started Get-THR_BitLocker at 2018-08-29 23:53:14Z
Started Get-THR_Certificates at 2018-08-29 23:53:15Z
Started Get-THR_Computer at 2018-08-29 23:53:15Z
Started Get-THR_DNS at 2018-08-29 23:53:16Z
Started Get-THR_EnvVars at 2018-08-29 23:53:16Z
Started Get-THR_GroupMembers at 2018-08-29 23:53:16Z
Started Get-THR_Hardware at 2018-08-29 23:53:17Z
Started Get-THR_Hosts at 2018-08-29 23:53:17Z
Started Get-THR_Hotfixes at 2018-08-29 23:53:17Z
Started Get-THR_NetAdapters at 2018-08-29 23:53:17Z
Started Get-THR_NetRoute at 2018-08-29 23:53:18Z
Started Get-THR_TCPConnections at 2018-08-29 23:53:18Z
Started Get-THR_Registry at 2018-08-29T23:53:18.7092909+08:00
```

THRecon运行截图

终端检测响应/EDR主要工具集（续）

OSQUEYRY介绍（linux）

简介：

Facebook 著名开源操作系统检测和监控项目， 100%使用系统API实现，没有使用fork execve ， 也支持windows操作系统。

链接：

Github：https://github.com/facebook/osquery

官网：https://osquery.io

安装：

$ sudo rpm -ivh https://osquery-packages.s3.amazonaws.com/centos7/noarch/osquery-s3-centos7-repo-1-0.0.noarch.rpm
$ sudo yum install osquery
$ sudo service osqueryd start

更新配置文件：
覆盖文件（/etc/osquery/osquery.conf），并重启服务。

查询：
实时查询接口：输入osqueryi后执行SQL语句进行检索
历史日志存储在：/var/log/osquery/osqueryd.results.log目录下。

查询方法示例：

（1）查询系统用户

命令： select * from users;

```
Using a virtual database. Need help, type '.help'
osquery> select * from users;
+-------+-------+-----------+-----------+--------------+--------------+---------------
---------------+----------------+------+
| uid   | gid   | uid_signed | gid_signed | username     | description
ry             | shell          | uuid |
+-------+-------+-----------+-----------+--------------+--------------+---------------
---------------+----------------+------+
| 0     | 0     | 0          | 0          | root         | root
               | /bin/bash      |      |
| 1     | 1     | 1          | 1          | bin          | bin
               | /sbin/nologin  |      |
| 2     | 2     | 2          | 2          | daemon       | daemon
               | /sbin/nologin  |      |
| 3     | 4     | 3          | 4          | adm          | adm
m              | /sbin/nologin  |      |
| 4     | 7     | 4          | 7          | lp           | lp
ool/lpd        | /sbin/nologin  |      |
| 5     | 0     | 5          | 0          | sync         | sync
               | /bin/sync      |      |
```

（2）查询网络连接情况

命令： select processes.pid,processes.name,listening_ports.port,lisselect processes.pid,processes.name,listening_ports.port,listening_ports.protocol,listening_ports.address from listening_ports JOIN processes USING (pid) limit 20;

```
osquery> select processes.pid,processes.name,listening_ports
.port,lisselect processes.pid,processes.name,listening_ports
.port,listening_ports.protocol,listening_ports.address  from
 listening_ports JOIN processes USING (pid) limit 20;
+-------+-----------+-------+----------+-----------------+
| pid   | name      | port  | protocol | address         |
+-------+-----------+-------+----------+-----------------+
| 2021  | rpcbind   | 111   | 6        | 0.0.0.0         |
| 2286  | sshd      | 22    | 6        | 0.0.0.0         |
| 2081  | rpc.statd | 44662 | 6        | 0.0.0.0         |
| 2119  | cupsd     | 631   | 6        | 127.0.0.1       |
| 2501  | master    | 25    | 6        | 127.0.0.1       |
| 2081  | rpc.statd | 38850 | 6        | ::              |
| 2021  | rpcbind   | 111   | 6        | ::              |
| 2286  | sshd      | 22    | 6        | ::              |
| 2119  | cupsd     | 631   | 6        | ::1             |
| 2501  | master    | 25    | 6        | ::1             |
| 2021  | rpcbind   | 111   | 17       | 0.0.0.0         |
| 2119  | cupsd     | 631   | 17       | 0.0.0.0         |
| 2414  | ntpd      | 123   | 17       | 172.16.165.252  |
| 2414  | ntpd      | 123   | 17       | 127.0.0.1       |
| 2414  | ntpd      | 123   | 17       | 0.0.0.0         |
| 2081  | rpc.statd | 36862 | 17       | 0.0.0.0         |
| 2021  | rpcbind   | 924   | 17       | 0.0.0.0         |
| 34671 | dhclient  | 68    | 17       | 0.0.0.0         |
| 2081  | rpc.statd | 985   | 17       | 127.0.0.1       |
| 2021  | rpcbind   | 111   | 17       | ::              |
+-------+-----------+-------+----------+-----------------+
```

终端检测响应/EDR主要工具集（续）

OSQUERY日志类别

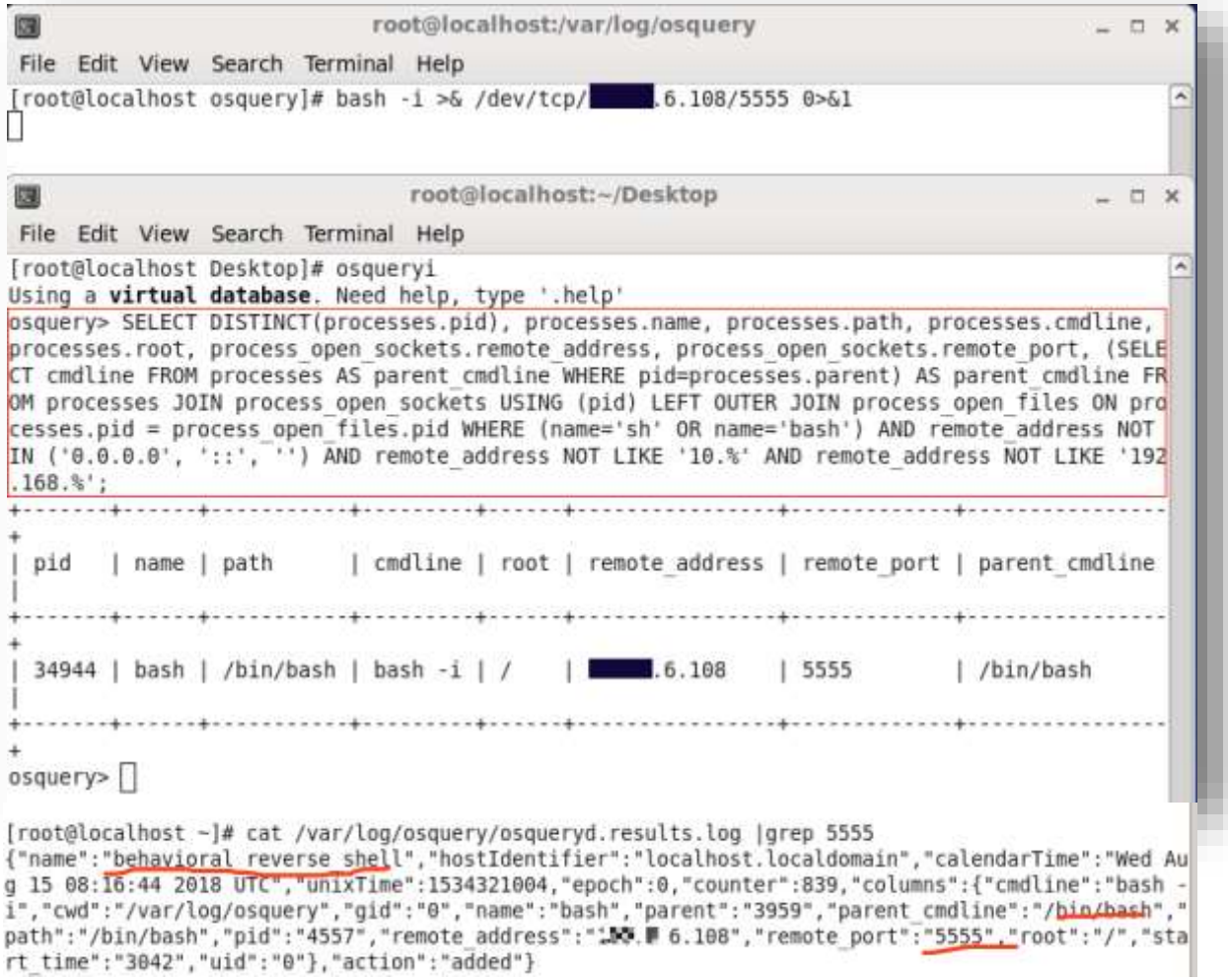
```
osquery> SELECT * FROM...
```

account_policy_data
acpi_tables
ad_config
alf
alf_exceptions
alf_explicit_auths
alf_services
app_schemes
apps
apt_sources
arp_cache
asl
augeas
authorization_mechanisms
authorization
authorized_keys
block_devices
browser_plugins
carbon_black_info
carves
certificates
chrome_extensions
cpu_time
cpuid
crashes
crontab
cups_destinations
cups_jobs
curl
curl_certificate
device_file
device_firmware
device_hash
device_partitions
disk_encryption
disk_events
dns_resolvers
docker_container_labels
docker_container_mounts
docker_container_networks
docker_container_ports
docker_container_processes
docker_container_stats
docker_containers
docker_image_labels
docker_images
docker_info
docker_network_labels
docker_networks
docker_version
docker_volume_labels
docker_volumes
etc_hosts
etc_protocols
etc_services
event_taps
extended_attributes
fan_speed_sensors
file
file_events
firefox_addons
gatekeeper
gatekeeper_approved_apps
groups
hardware_events
hash
homebrew_packages
intel_me_info
interface_addresses
interface_details
iokit_devicetree
iokit_registry
kernel_extensions
kernel_info
kernel_panics
keychain_acls
keychain_items
known_hosts
last
launchd
launchd_overrides
listening_ports
load_average
logged_in_users
magic
managed_policies
mdfind
memory_devices
mounts
nfs_shares
nvram
opera_extensions
os_version
osquery_events
osquery_extensions
osquery_flags
osquery_info
osquery_packs
osquery_registry
osquery_schedule
package_bom
package_install_history
package_receipts
pci_devices
platform_info
plist
power_sensors
preferences
process_envs
process_events
process_memory_map
process_open_files
process_open_sockets
processes
prometheus_metrics
python_packages
quicklook_cache
routes
safari_extensions
sandboxes
shared_folders
sharing_preferences
shell_history
signature
sip_config
smbios_tables
smc_keys
startup_items
sudoers
suid_bin
system_controls
system_info
temperature_sensors
time
time_machine_backups
time_machine_destinations
uptime
usb_devices
user_events
user_groups
user_interaction_events
user_ssh_keys
users
virtual_memory_info
wifi_networks
wifi_status
wifi_survey
xprotect_entries
xprotect_meta
xprotect_reports
yara
yara_events

终端检测响应/EDR主要工具集（续）

OSQUEYRY典型应用场景

检测反弹shell




```
osquery> SELECT DISTINCT(processes.pid),
processes.name, processes.path, processes.cmdline,
processes.root, process_open_sockets.remote_address,
process_open_sockets.remote_port, (SELECT cmdline FROM
processes AS parent_cmdline WHERE pid=processes.parent)
AS parent_cmdline FROM processes JOIN
process_open_sockets USING (pid) LEFT OUTER JOIN
process_open_files ON processes.pid =
process_open_files.pid WHERE (name='sh' OR
name='bash') AND remote_address NOT IN ('0.0.0.0', '::', '')
AND remote_address NOT LIKE '10.%' AND
remote_address NOT LIKE '192.168.%';
```

终端检测响应/EDR主要工具集（续）

OSQUEYRY实践建议

1、推荐使用Palantir的公版配置文件进行mod（ https://github.com/palantir/osquery-configuration ）

2、定时采集，存在瞬时状态漏采的情况，需要合理设置配置文件的interval。

3、对osquery服务的保护和监控，日志文件及时上传集中存储。

4、日志采集检测规则已知，需要不断优化检测机制。

目录

6、安全响应的playbook

Search

Bloomberg

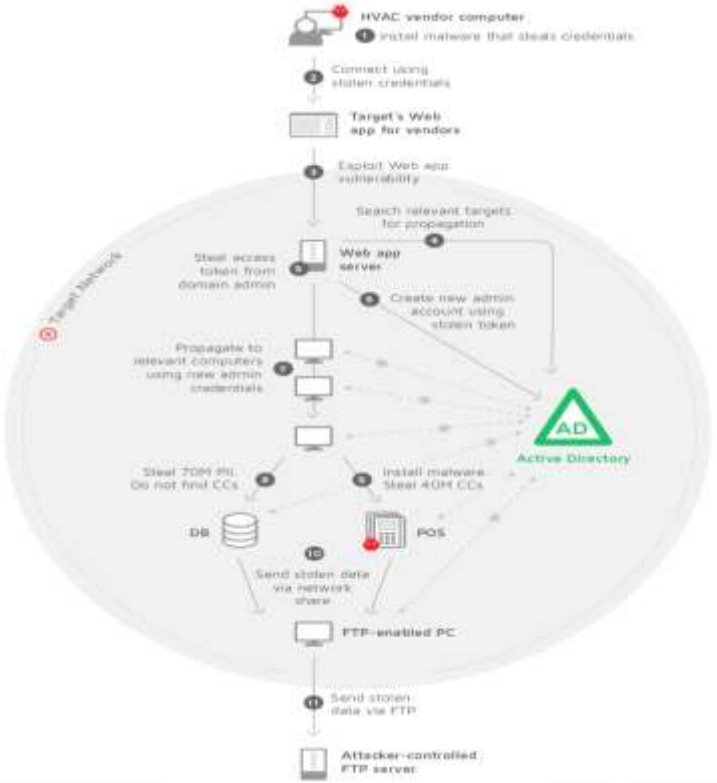
Sign I

Business

Missed Alarms and 40 Million Stolen Credit Card Numbers: How Target Blew It

Target ignored its own alarms—and turned its customers into victims

By Michael Riley, Ben Elgin, Dune Lawrence, and Carol Matlack
2014年3月17日 GMT+8 下午10:31



6、安全响应的playbook（续）

安全事件响应的目标

安全事件响应的两大目标：

- 提高信噪比

增加高保真的告警，击败告警疲劳，专注于真正的危险和问题

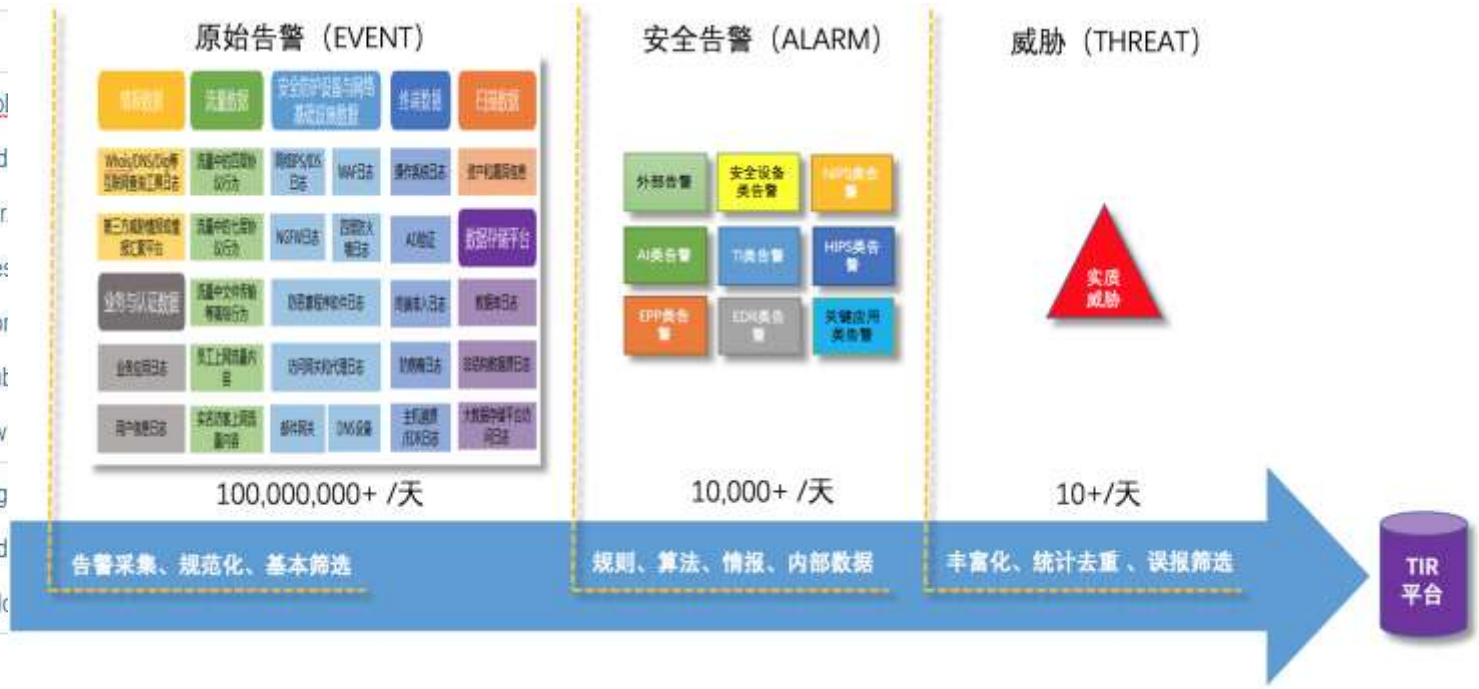
- 降低MTTR

固化流程，丰富场景，并持续运营，使响应时间不断降低

6、安全响应的playbook（续）

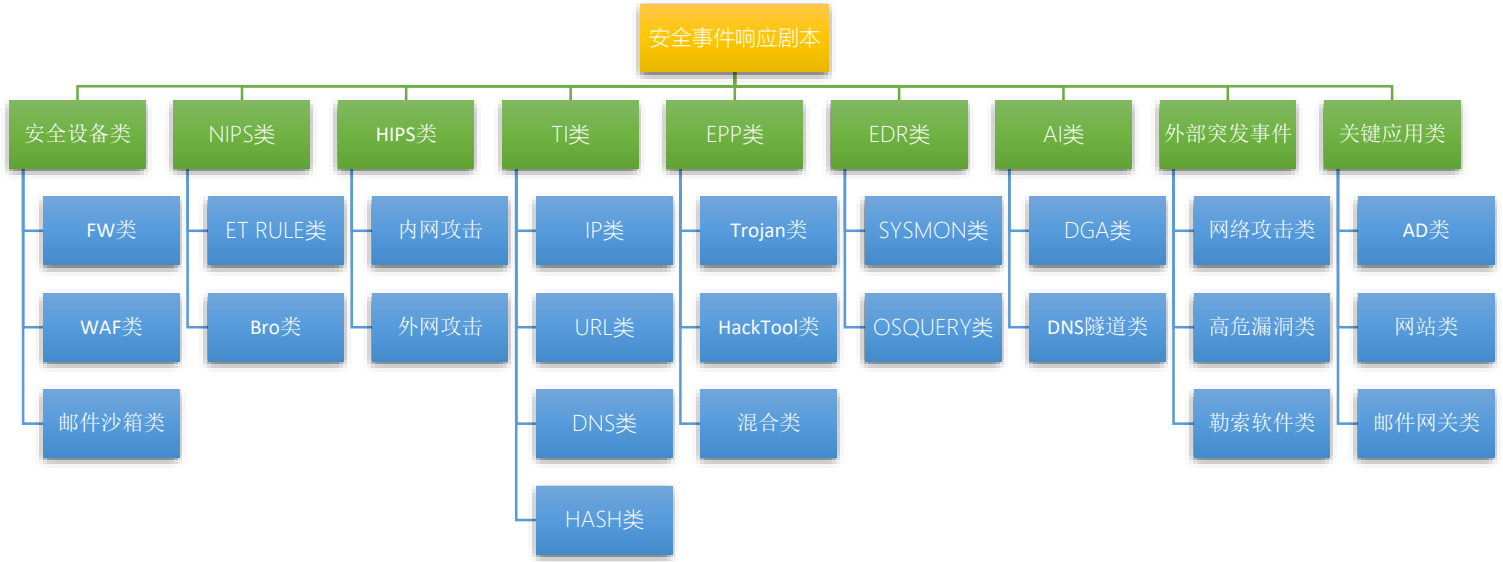
第一步：告警的分类定级和误报筛选

告警分类	告警描述	killchain	告警等级	误报筛选
alarm_intelligence_outbound_domain	出站恶意域名访问	C&C	Critical	url:"cdp1.publ url:"download url:"redirector url:"s3-us-wes url:"github.cor url:"raw.github url:"www.dow
alarm_intelligence_outbound_ip	出站恶意IP访问	C&C	High	url:"www.leag url:"download url:"au.downlo



6、安全响应的playbook（续）

第二步：固化基本流程并丰富分场景响应剧本（playbook）



6、安全响应的playbook（续）

第三步：按照事件优先级进行持续运营

DAY

- 出站Ti告警
- 严重及高危NIPS告警
- 严重及高危HIPS告警
- 严重及高危EDR告警

.....

WEEK

- 入站Ti告警趋势性分析
- 中低危HIPS告警
- 中低危EDR告警
- 中低危EPP告警

.....

MONTH

- 出站灰流量分析
- 企业情报库更新
- 威胁捕猎
- 当月情况review

.....

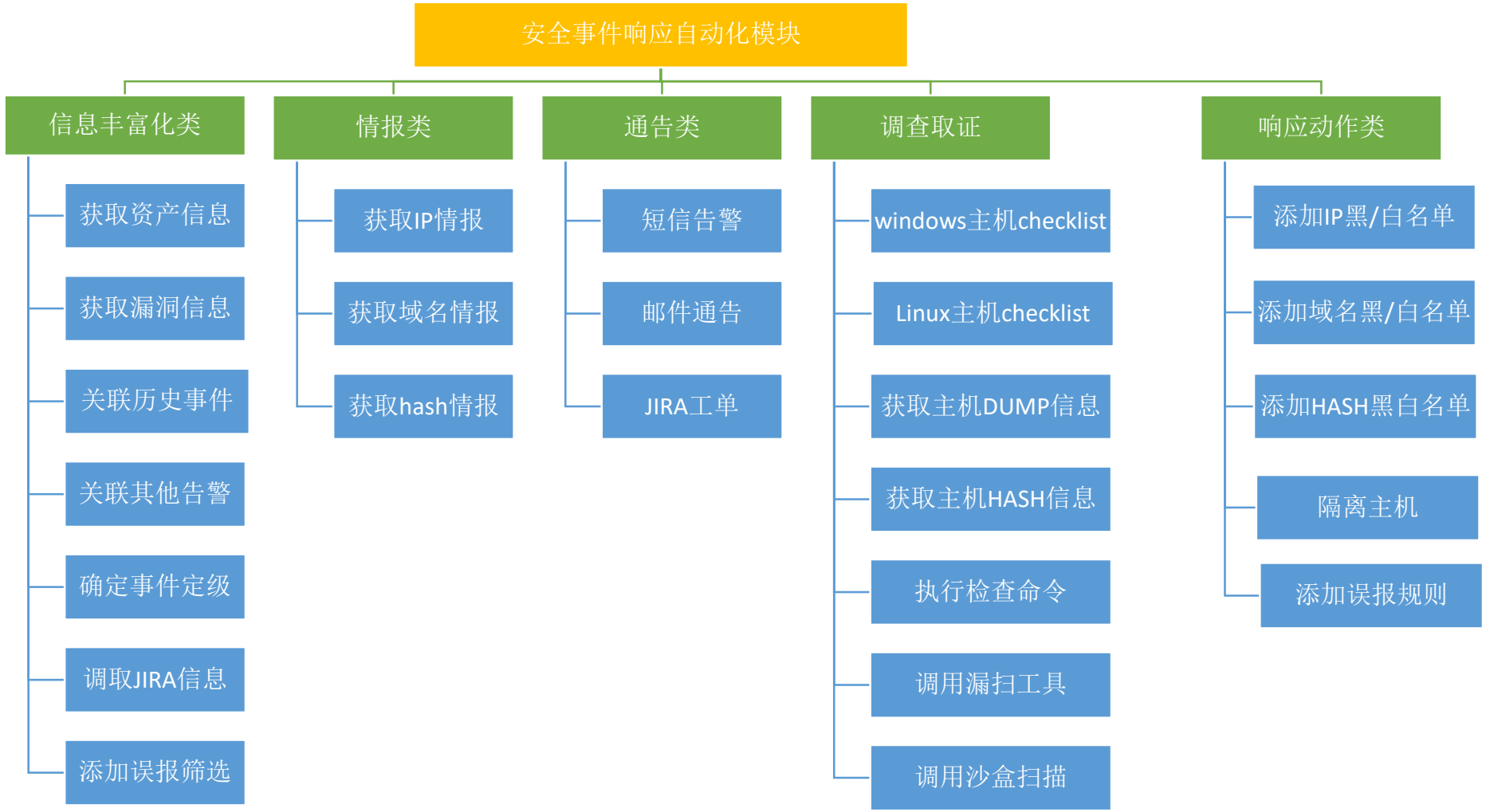
ANY TIME

- 突发事件的应急响应
- 红蓝对抗演习
- 监管部门安全通告
- 行业性演练

.....

6、安全响应的playbook（续）

第四步：运营过程中的自动化模块构建



6、安全响应的playbook（续）

第四步：运营过程中的自动化模块构建（续）

不建议自动化的任务：

- 人工复查点（事件的确认定级、事件的误报筛选等）
- 关键性系统（业务连续性要求较高的系统、包含大量敏感数据的系统等）
- 关键性决策（执行断网隔离动作、执行文件删除动作等）
- 高级人工分析任务（威胁捕猎、溯源分析、改进建议等）

……

6、安全响应的playbook

CASE STUDY： HIPS_MS17-010类响应剧本编排

背景情况：

某日，发现了一起攻击源头位于内网的HIPS高危告警，且攻击特征为MS17-010漏洞。当即进入“HIPS_MS17-010”响应流程。

	sip	dip	CIDS_sig_str
2018, 16:20:59.720	192.168.3.166	192.168.14.18	Attack : SMB Double Pulsar Ping
2018, 16:20:59.688	192.168.3.166	192.168.14.18	Attack : SMB Double Pulsar Ping
2018, 16:20:59.682	192.168.3.166	192.168.14.18	Attack : SMB Double Pulsar Ping
2018, 16:20:59.673	192.168.3.166	192.168.14.18	Attack : SMB Double Pulsar Ping

6、安全响应的playbook(续)

CASE STUDY： HIPS_MS17-010类响应剧本编排

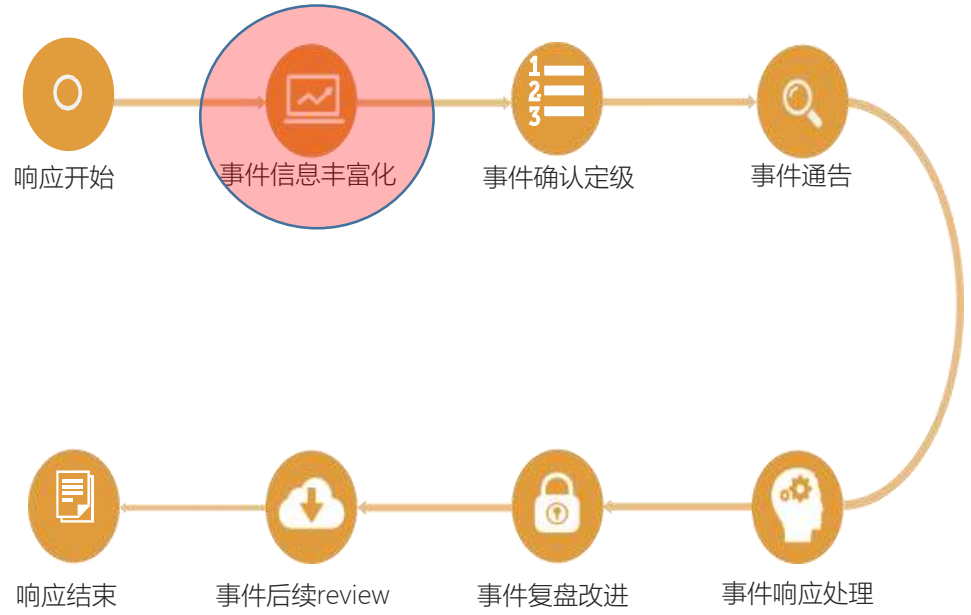
◆流程主要阶段：

◆自动化模块：

◆人工任务：

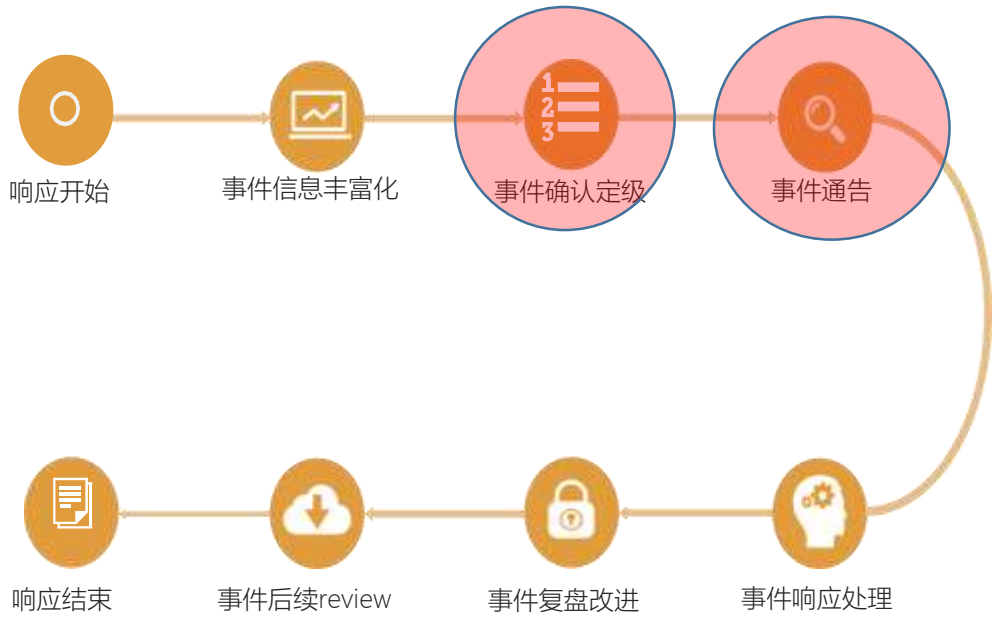
◆人工判断：

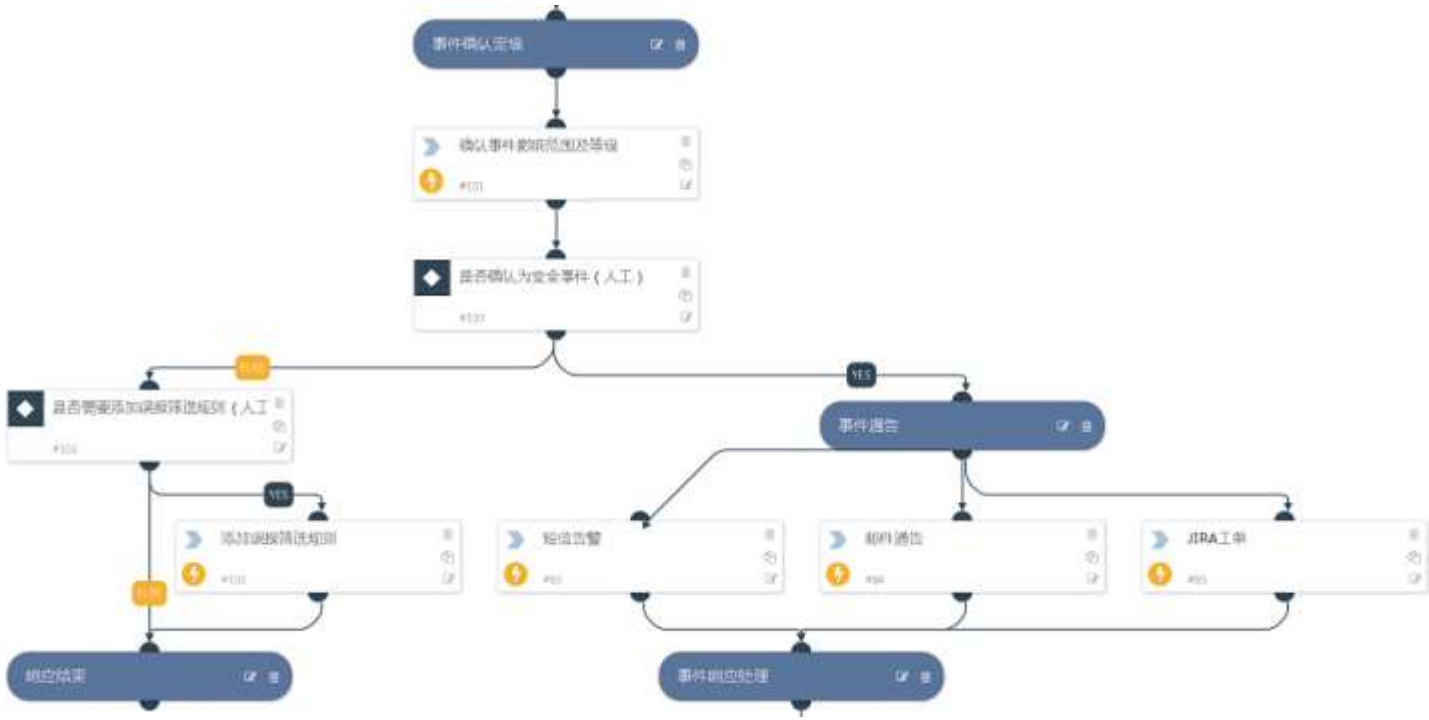
◆自动化判断：



6、安全响应的playbook(续)

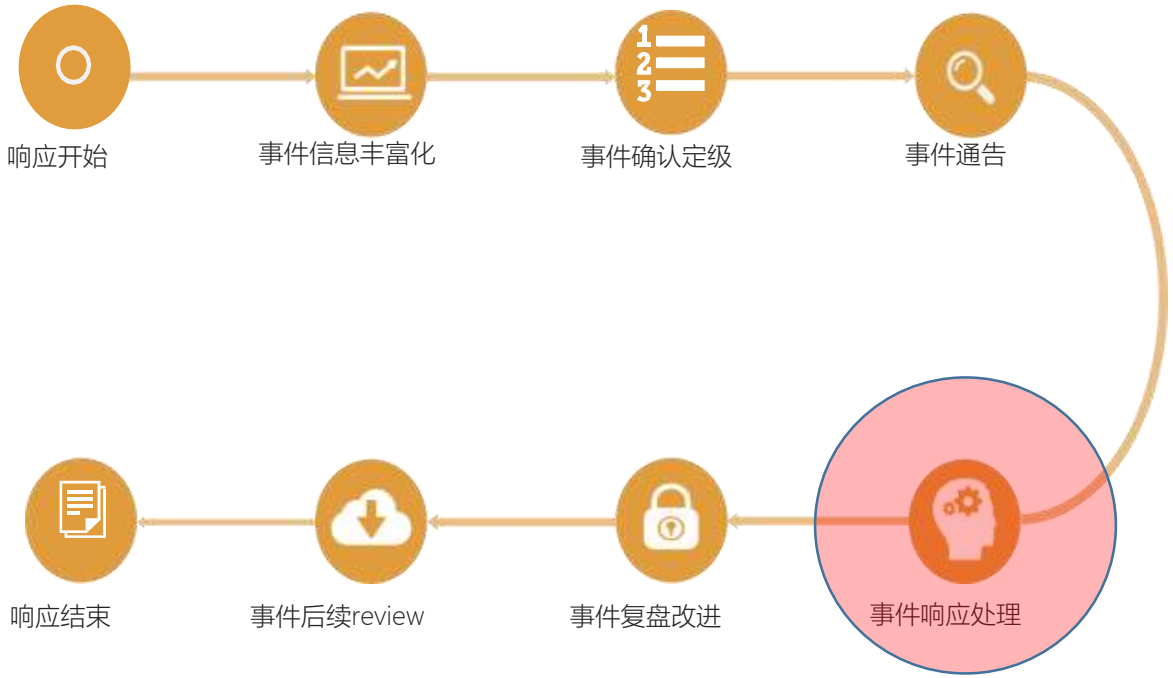
CASE STUDY： HIPS_MS17-010类响应剧本编排

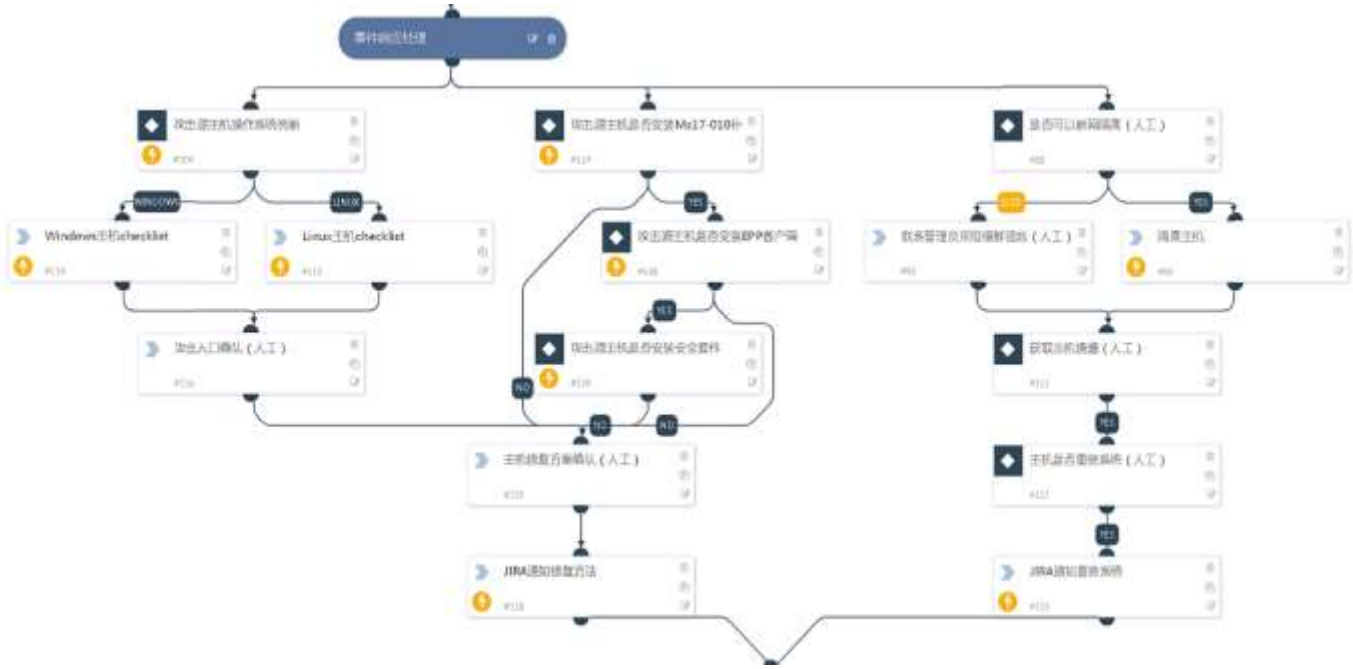




6、安全响应的playbook(续)

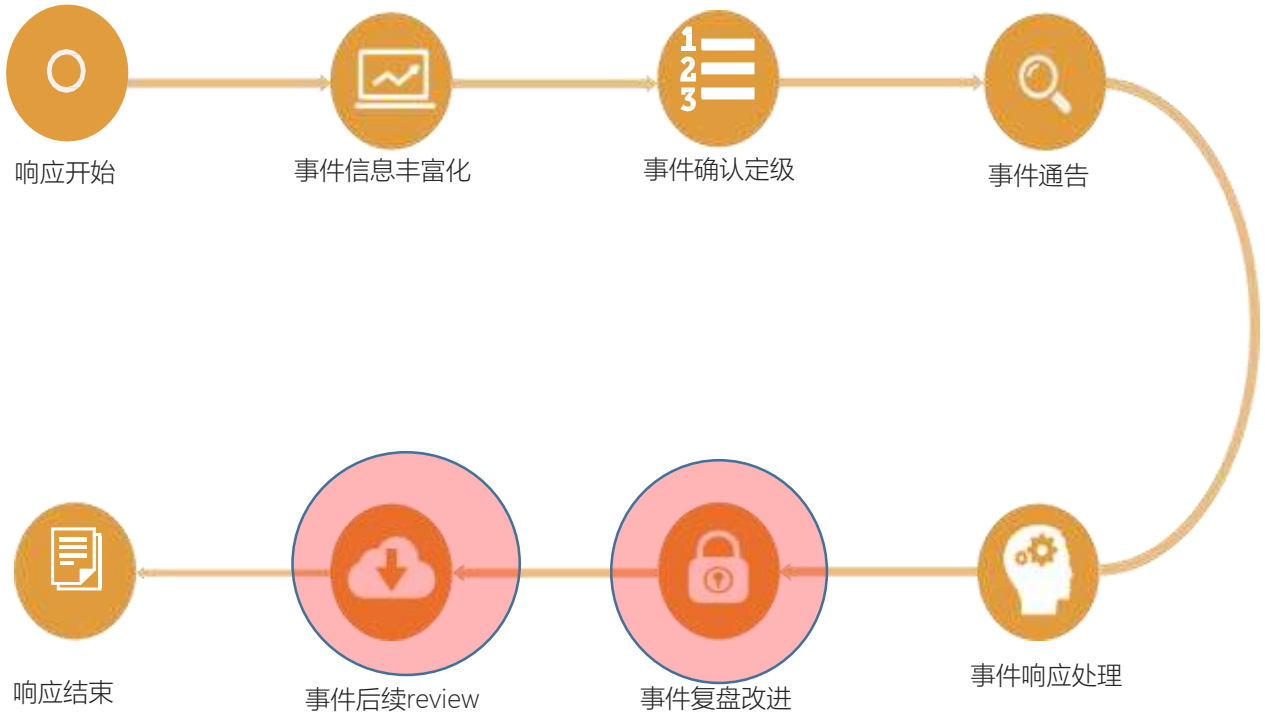
CASE STUDY： HIPS_MS17-010类响应剧本编排

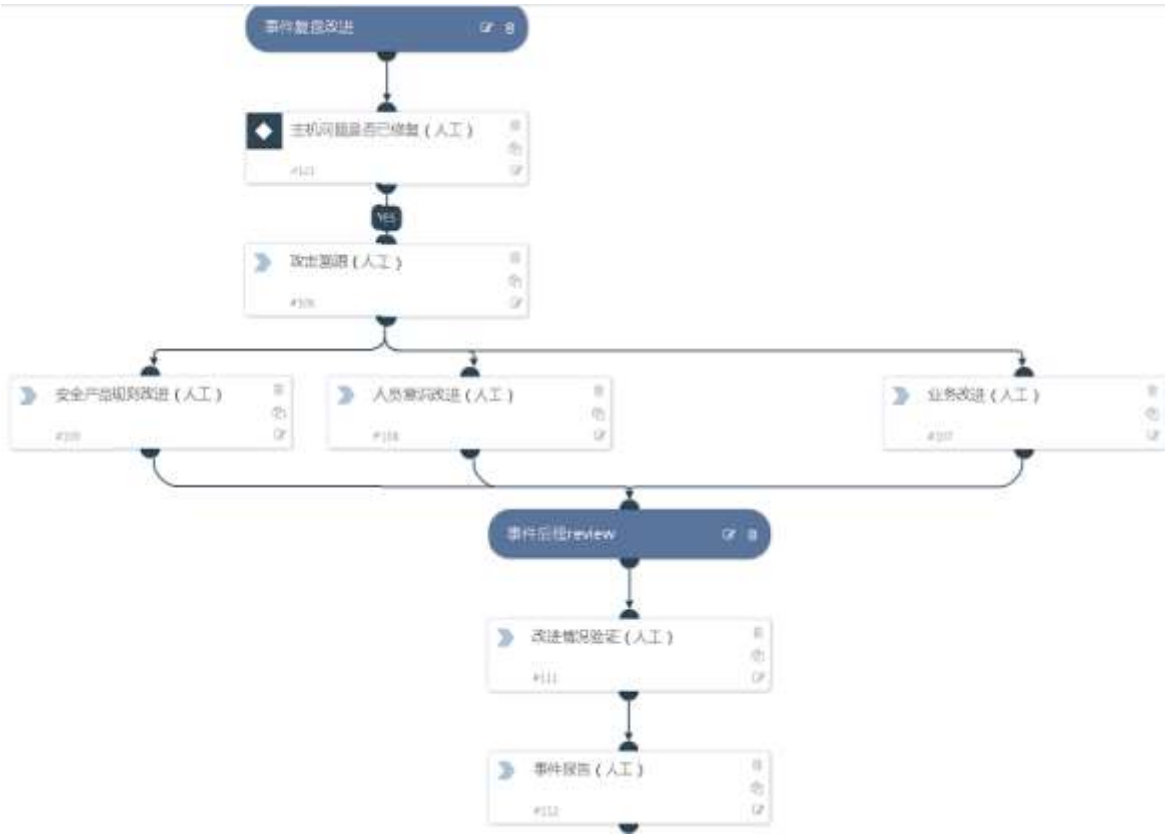




6、安全响应的playbook(续)

CASE STUDY： HIPS_MS17-010类响应剧本编排






ISC
2018
ISC互联网安全大会
360互联网安全中心
谢 谢!
ISC 互联网安全大会 中国 · 北京
Internet Security Conference 2018 Beijing · China
（原“中国互联网安全大会”）