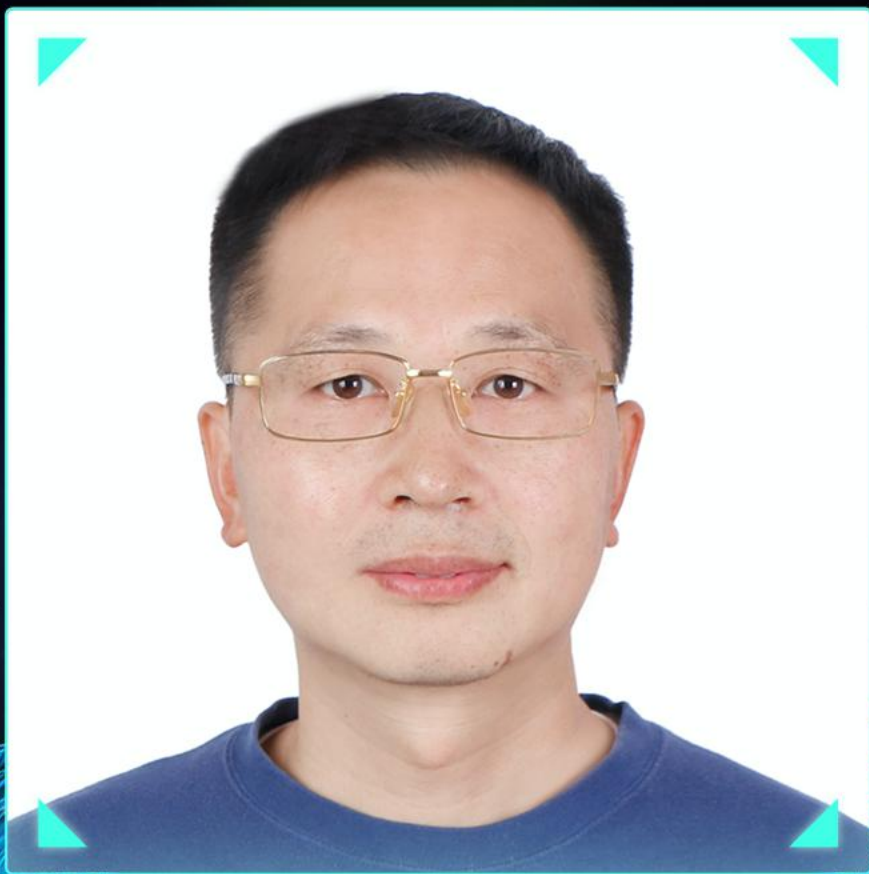


2021

58同城 第2届
安全技术沙龙

业务风控建设 & 应用安全实践





» 徐祖军 «

腾讯安全平台部技术专家

分享主题

流量分析在 应用安全中的 探索实践

议题介绍

安全问题的解决始于有效的发现，应用安全问题发现有多种不同的手段，这里将从流量分析的宏观角度就应用安全一些典型问题，包括漏洞发现等问题进行探索实践，就检测的实效性以及应用角度的多样性跟大家进行分享。

01

流量宝藏

目录

背景介绍

- 解决数据孤岛
- 时效性

重点目标

- 反入侵，防漏洞
- 态势感知，主动对抗高级威胁

实践探索

- 主机和应用层漏洞发现
- 挖矿，**DNS**安全，协议层面的分析

背景介绍

核心思想

- 多层防御体系
- 网络流量雁过留痕



02

重点目标

重点目标 逻辑架构

- 覆盖多种场景
- 不同的算法适配



重点目标 防漏洞

- 高危行为
- 高危组件覆盖



重点目标 反入侵

- 木马流量
- 隧道流量



03

实践尝试

实践尝试

主机安全回溯

传统主机安全检测响应EDR系统，通过Agent方式采集主机日志、命令操作等信息，然后上报到控制中心进行策略建模，从而发现主机入侵威胁。比如高风险命令注入执行，单纯基于主机端数据仅能知道发生了什么，若同时能在流量层面针对这些强特征命令字进行检测，进一步关联，就能溯源到攻击者是如何利用的。此外，流量层的检测能力，也能

源IP	域名	CGI	UA	uin	插入时间	http_param
	.qq.com	/getMd5	Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/38.0.2125.111 Safari/537.36 QQBrowser/4.2.4763.400 PCG Security Team	3464597487	2019-12-17 23:04:57	format-detection=&viewport=&apkUrl=ping.6.p1.apkUrl.345322.jinchi.w.a.cgi.pw? security.tencent.com

流量层感知的网络路径和请求，形成关联

出现可疑操作[]

服务器出现的行为: ping_inject

nobody在/ /front/public目录下执行了命令: ping 6.p1.apkUrl.345322.jinchi.w.a.cgi.pw

发生时间: 2019-12-17 23:04:37

security.tencent.com

主机层发现的告警

实践尝试

木马通信

ICMP木马，进程也不会监听端口，对主机层检测能力提出更大挑战，由于在协议栈之前对流量进行劫持处理，主机端tcpdump也无法抓取。而作为中间管道，在流量层进行检测分析，会是更好的补充手段。

New Tab × +

PRETTIFY

HISTORY

http://49.51.128.100/

3/

```
1 # Write your query or mutation here
2
```

```
root@VM-0-14-ubuntu:/home/ubuntu# tcpdump -i eth0 -s 0 -n host 115.159.102.100
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
```

security.tencent.com

security.tencent.com

在目标主机上使用 `tcpdump -i eth0 -s 0 -n host 115.159.102.100` 抓取源主机-115.159.102.100的包，无法抓获

```
[root@VM_196_165_centos ~]# ping 129.226.102.100
PING 129.226.102.100 (129.226.102.100) 56(84) bytes of data.
64 bytes from 129.226.102.100: icmp_seq=1 ttl=21 time=102 ms
64 bytes from 129.226.102.100: icmp_seq=2 ttl=21 time=102 ms
64 bytes from 129.226.102.100: icmp_seq=3 ttl=21 time=102 ms
64 bytes from 129.226.102.100: icmp_seq=4 ttl=21 time=102 ms
64 bytes from 129.226.102.100: icmp_seq=5 ttl=21 time=102 ms
64 bytes from 129.226.102.100: icmp_seq=6 ttl=21 time=102 ms
```

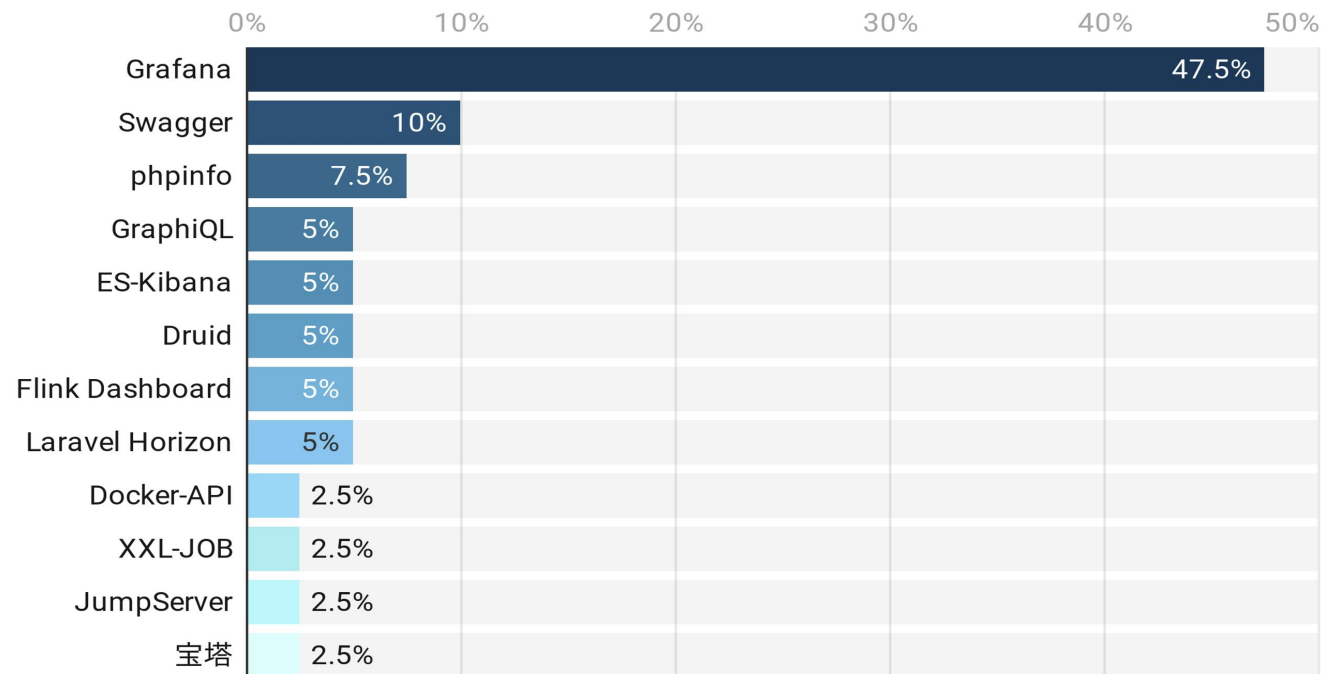
从源主机向目标主机129.226.x.x进行ping操作，可以正常ping通

实践尝试

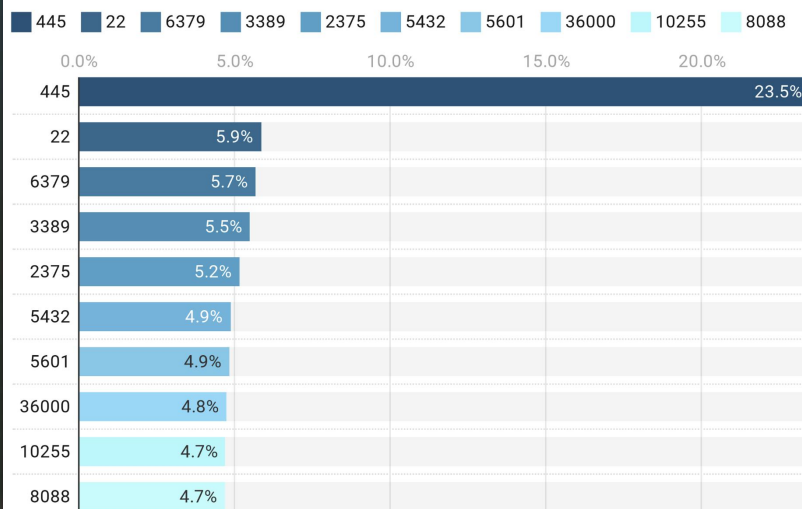
高危资产主动发现

- 1.高危端口、高危组件和高危服务的对外开放占据了漏洞攻击的很大部分，靠主动扫描来感知，存在扫描周期和扫描被屏蔽的问题。
- 2.通过流量建立相关通信特征的检测，则可以实现秒级响应。业务高危端口从对外开放到被探测利用的时间只有45秒，针对资产数量较多的大中型企业来说，很难在45秒内完成一轮扫描。
- 3.网络上的大量扫描和探测会带来误报，结合出流量关联，则能较好的解决误报问题

常见对外开放的高危组件分布



被扫描端口TOP10



实践尝试

脆弱点 / 漏洞主动发现

1.漏洞扫描一般重点关注在对特征类漏洞的发现，发送特定构造参数或payload以触发是否存在漏洞，对于业务层面的不合规行为或者逻辑类漏洞则覆盖不足，比如敏感信息明文传输、越权、管理后台类等。

2. 扫描方法基于关键字，灵活性和可维护性不高。流量分析，可以引入AI算法，模型可以自动学习页面特征，效果远超过传统方案。同时，引入正负反馈机制配合AI模型训练，提升模型识别率。此外这不但可以监测管理后台的对外开放问题，而且也可以对左右侧后台的管理后台进行告



The image shows a 'Login Form' with two input fields: 'Username' and 'Password'. The 'Username' field has a red error message below it: '用户名长度不能小于5!'. The 'Password' field has a toggle icon for visibility. Below the fields is a blue 'Login' button. At the bottom right, the URL 'security.tencent.com' is visible.

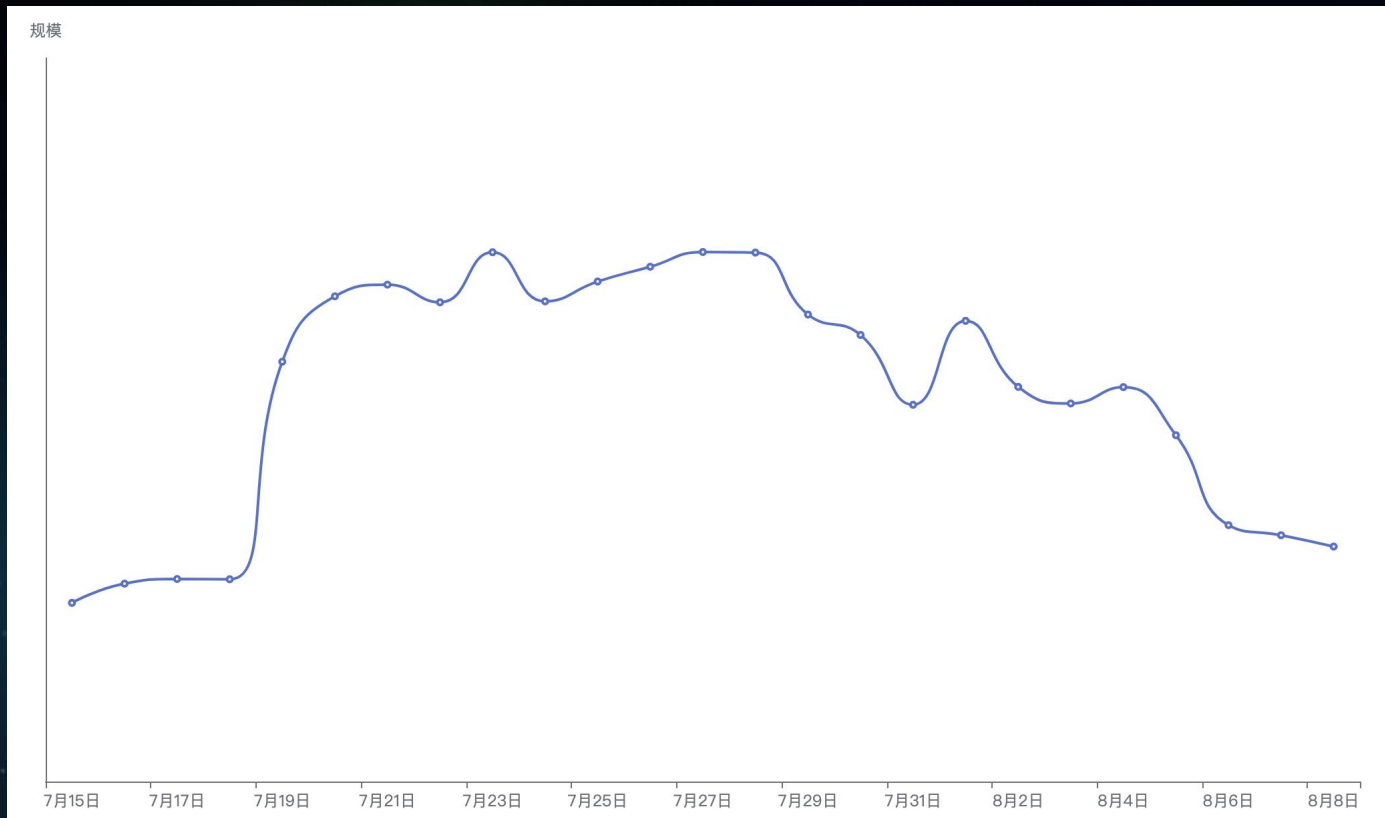
JS生成的管理页面，若扫描器不能解析JS则不能主动发现

实践尝试

Mirai僵尸网络

通过分析Mirai僵尸网络的命令与心跳包，提取出属于Mirai僵尸网络C&C服务器与被控端之间的通信流量，从而感知Mirai僵尸网络的规模变化

攻击指令：其中开头的"001a"代表攻击指令的长度，紧接着的"0000001e"代表了攻击的持续时间为30秒，"1e"之后的"00"代表了攻击的所采用的手法，根据对大量Mirai样本的逆向，"00"代表了使用udp进行攻击，后面的"d39fcd91"代表了攻击的目标IP，之后的几位是攻击的配置项，"000431343430"代表了攻击所使用的udp包大小为1400，"070439303033"代表了攻击的目标端口是9003



Mirai僵尸网络规模

0040	89 a8 00 1a 00 00 00 1e 00 01 d3 9f cd 91 20 02
0050	00 04 31 34 34 30 07 04 39 30 30 33	..1440.. 9003

Mirai攻击指令示例

实践尝试

SWARM挖矿

流量特征：在挖矿程序与交换端点进行通讯时，采用了jsonrpc协议，这点与门罗币挖矿相同，并且数据包的强特征明显，可以针对数据包中的关键词或者数据包中的结构进行检测

端口特征：Swarm项目在运行时默认使用1633，1634，1635这三个端口来进行数据交换

加密流量：能够在握手协议和证书两个层面来做一些事情。由于挖矿的特殊性，矿池的域名、证书是不会轻易进行变化的，并且矿池的具有聚集属性，即越大的矿池集合到的矿机越多，越能够保证收益的稳定性。所以也可以针对排名较为靠前的矿池进行域名和证书的收集，添加针对性的检测策略

```
POST / HTTP/1.1
Host: [REDACTED]
User-Agent: Go-http-client/1.1
Content-Length: 47
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip

{"jsonrpc":"2.0","id":1,"method":"eth_chainId"}HTTP/1.1 200 OK
content-type: application/json; charset=utf-8
content-length: 41
date: Tue, 29 Jun 2021 02:21:53 GMT

{"jsonrpc":"2.0","result":"0x64","id":1}
POST / HTTP/1.1
Host: [REDACTED]
User-Agent: Go-http-client/1.1
Content-Length: 51
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip

{"jsonrpc":"2.0","id":2,"method":"eth_blockNumber"}HTTP/1.1 200 OK
content-type: application/json; charset=utf-8
content-length: 46
date: Tue, 29 Jun 2021 02:21:53 GMT

{"jsonrpc":"2.0","result":"0x100aa19","id":2}
POST / HTTP/1.1
Host: [REDACTED]
User-Agent: Go-http-client/1.1
Content-Length: 85
Accept: application/json
Content-Type: application/json
Accept-Encoding: gzip

{"jsonrpc":"2.0","id":3,"method":"eth_getBlockByNumber","params":["0x100aa19",false]}HTTP/1.1 200 OK
content-type: application/json; charset=utf-8
content-length: 100
date: Tue, 29 Jun 2021 02:21:53 GMT
```

Jsonrpc协议

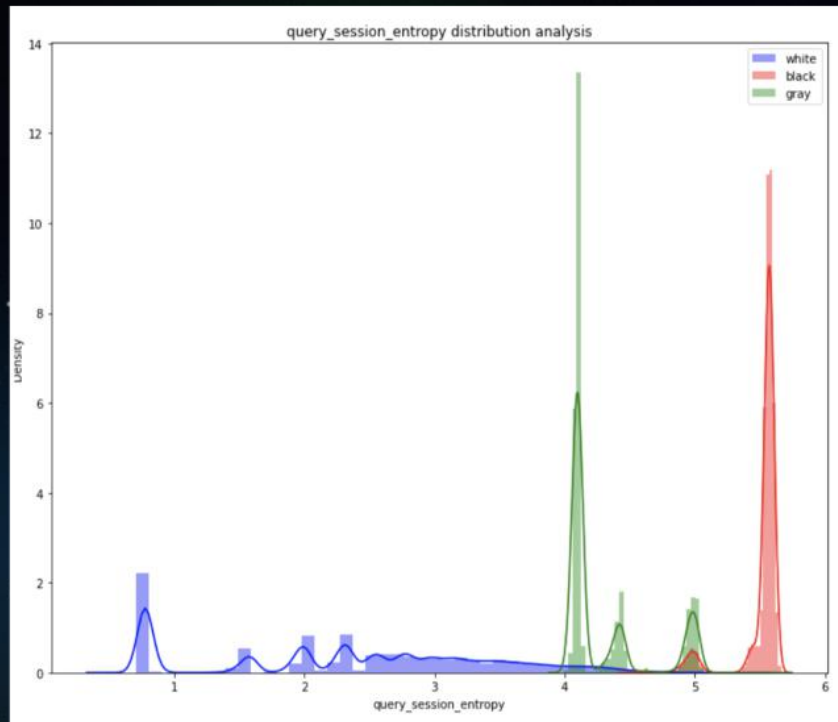
```
[2021-11-16 11:06:33.422] net use pool pool.hashvault.pro:80 TLSv1.3 125.253.92.50
[2021-11-16 11:06:33.422] net new job from pool.hashvault.pro:80 diff 895531 algo rx/0 height
2494252 (37 tx)
[2021-11-16 11:06:33.422] cpu use argon2 implementation AVX2
[2021-11-16 11:06:33.428] msr cannot read MSR 0xc0011020
[2021-11-16 11:06:33.428] msr FAILED TO APPLY MSR MOD, HASHRATE WILL BE LOW
[2021-11-16 11:06:33.430] randomx init dataset algo rx/0 (4 threads) seed 3185acac234a171b...
[2021-11-16 11:06:34.605] randomx allocated 2336 MB (2080+256) huge pages 11% 128/1168 +JIT (1175
ms)
```


实践尝试

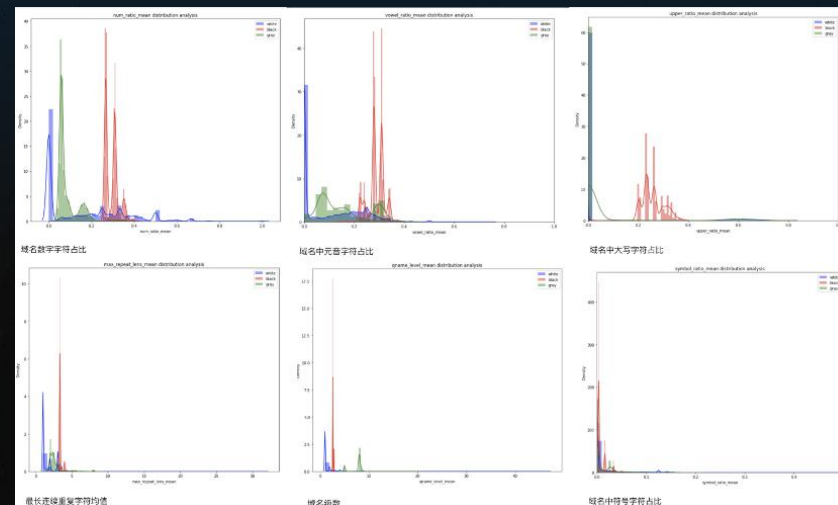
DNS隧道

三大类域名（黑，白，灰）的明显不同的统计特征

建立了一套基于AI+规则的DNS隧道流量检测系统。现网检出的隧道流量大多还是以正规厂商的DNS隧道及个人测试DNS隧道工具为主（示例如下表所示）。可以顺道发现一些与隧道相似（大量乱码域名请求）的DGA域名



隧道流量和普通流量信息熵



隧道流量和普通流量域名编码特征

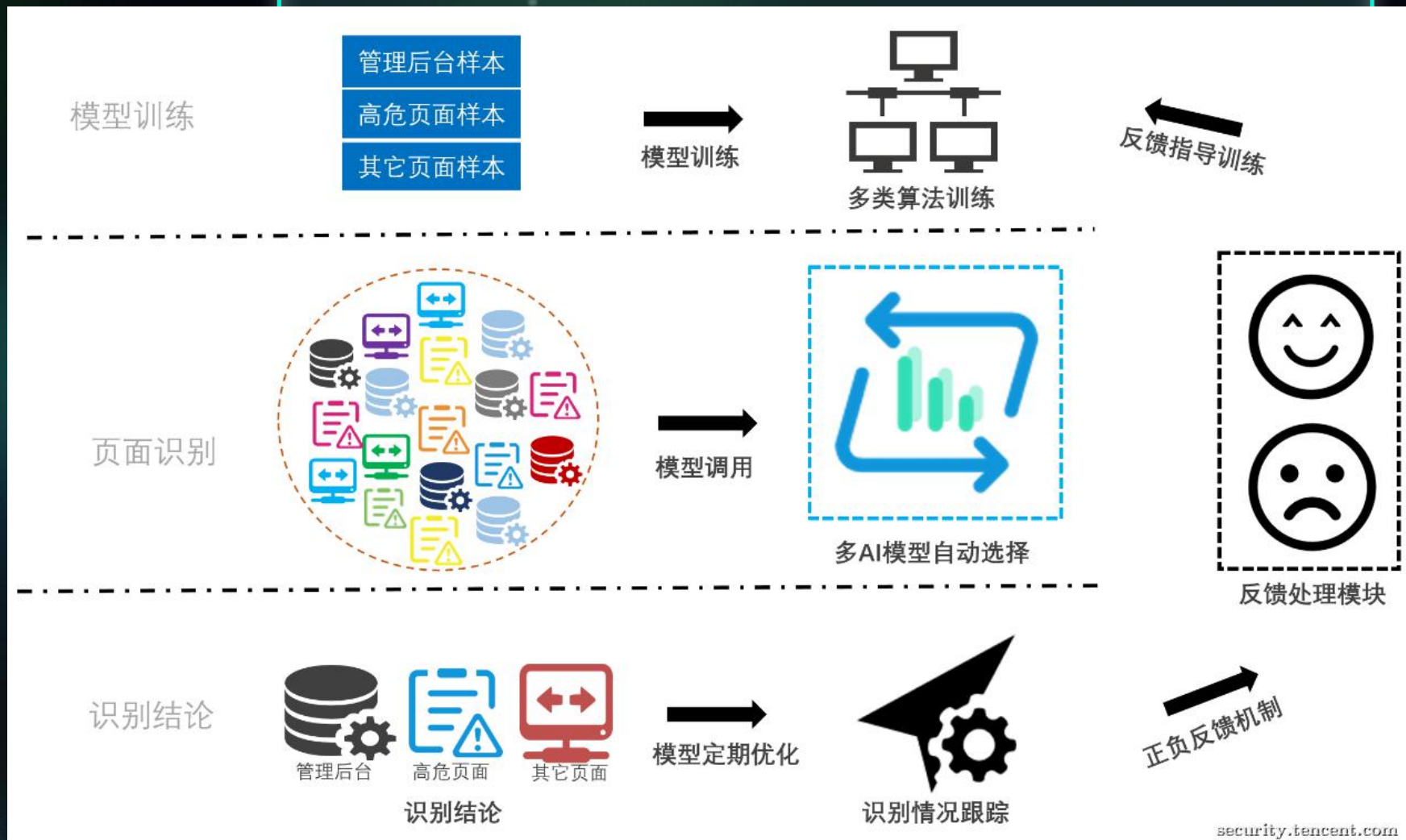
04

展望

展望

挑战

- 加密流量https,quic的解决思路
- 基于黑特征的检测算法的失效
- 解决方案：接入解密网关；同时对于某些场景，需要降低对黑特征、关键字的依赖，综合利用大数据、AI等手段，构建基于行为的检测机制，比如木马主控C&C行为的发现



THANKS

