Table of Contents

1 文档目标读者与使用场景

本文旨在为我司内部提供威胁情报与安全产品深度结合的研发指导，主要面向安全产品经理、研发工程师、安全架构师及企业安全负责人。目标读者需具备网络安全基础认知，关注攻防实战能力提升与产品创新方向。具体适用场景如下：

◆ 安全产品能力迭代：

为流量威胁检测、安全运营与自动化分析响应平台、边界防护与阻断、主机与终端安全、邮件安全等安全防护场景的产品提供威胁情报集成方案，具体方式包括情报源管理，威胁情报检测模块配置，告警分析及响策略配置等，可有效增强安全产品的威胁检出率与拦截精度。

◆ 实网攻防与应急响应：

指导防守方在日常安全运营、攻防演练、重保防护等场景中发挥安全产品威胁情报模块的实战价值，提效威胁事件鉴定流程，改善分析方法，支撑事件响应团队制定行动策略。

◆ 安全能力体系规划：

为安全团队提供威胁情报驱动的技术架构设计参考，涵盖威胁数据运营流程，安全事件与告警分析，关键威胁提炼方式等场景。具体产品规划与设计需要满足实际业务场景下的相关规范和要求（如行业性标准，业务风险特征），本文档主要包含方法论框架，威胁情报数据选型，场景化功能设计等内容。读者可通过本文系统性掌握威胁情报与产品研发的融合路径，提升攻防对抗中的主动防御能力与业务风险管控效率。

2 威胁情报概述

2.1 威胁情报定义与价值

威胁情报（ThreatIntelligence，TI），是指通过收集、处理和分析来自多个来源的安全数据，结合上下文、技术和攻击者意图，提取出可用于支撑安全决策与行动的知识体系。

根据 Gartner（2013）的定义：

- 威胁情报是基于证据的知识集合，能够对现有或新兴的威胁或攻击者进行描述，旨在通过告知安全决策，来改善防御方的风险应对能力。
- Threat intelligence is evidence-based knowledge, including context, mechanisms, indicators, implications and actionable advice, about an existing or emerging menace or hazard to assets that can be used to inform decisions regarding the subject's response to that menace or hazard.

从业界主流共识来看，威胁情报不仅包含已知威胁的可识别指标（如恶意 IP、域名、文件 Hash），还包括攻击者使用的策略、战术、技术手段（TTPs），以及攻击团伙背景、行业目标、地理活跃区等上下文画像信息。详细见第 2 节描述。

网络威胁情报（CyberThreatIntelligence，CTI）是威胁情报的子集，Gartner 并没有给出明确定义，通常我们特指围绕网络资产相关的情报为网络威胁情报。

开源情报（Open-Source Intelligence）指基于某种目的对公开来源数据进行收集和处理,而得到的某种显式或隐含的信息。

其核心价值体现为构建分层次、全周期的网络安全决策支撑框架：

战术层	通过自动化威胁检测与阻断技术（如终端威胁检测与响应系统 EDR、防火墙产品进行集成），实现基于 IOC 的实时攻击拦截，将平均威胁响应时间缩短至秒级
运营层	结合 MITRE ATT&CK 框架进行攻击模式分析，动态优化安全设备策略配置，持续降低安全运维复杂度
战略层	通过攻击趋势预测和风险建模，支撑企业安全资源投入决策，形成攻防成本的非对称优势

2.2 威胁情报的数据分类

在具体的产品场景化应用中，由于需要考虑安全产品的实际功能和应用场景，采用的威胁情报数据会有较大差异。本章节从最佳实践出发，结合目前市面主流威胁情报数据分类经验，整体概述主要情报类型的定义和应用价值。

2.2.1 攻击来源入站情报

此类情报一般用于与入站请求的来源地址 IP（互联网 IP）进行匹配，以协助识别有风险的外部访问请求来源。可检出的威胁类型包括：WEB 攻击 、网络爆破、网络扫描、网络蜜罐、DDOS、垃圾邮件等攻击迹象的来源。常用于匹配的日志源包括 web 防火墙、流量检测、防火墙等。

2.2.2 失陷检测出站情报

此类情报一般用于与出站请求目的地址匹配，一般包括 IP、域名等维度，以协助识别内网是否存在失陷主机。可检出 APT、网银木马、窃密木马、勒索软件、僵尸网络、挖矿软件、常规木马、漏洞利用、SinkHole、DGA、远控木马、黑灰产等威胁类型造成的攻击事件。常用于匹配日志源包括防火墙、安全 DNS 日志、流量检测等。

2.2.3 文件信誉情报

此类情报一般用于与网络中传输的文件 Hash 进行匹配，以协助识别有风险的文件样本。查询值为 Hash（MD5、SHA1 等）。通过情报查询可获取文件黑白判定，风险等级，对应家族团伙信息，关联 IOC，相关 ATT&CK 攻击手法。常用的匹配的对象包括风险邮件附件，U 盘导入的可疑可执行文件等的 Hash 值。

2.2.4 风险 URL 链接情报

此类情报一般用于办公网终端上网行为中的风险识别，通过带协议头（如 https、http 等）的全地址匹配实现查询。此类情报一般更新频率高，可精准识别目标地址存在的业务风险（如电信金融欺诈）、攻击威胁（如钓鱼、恶意木马下载）、内容风险（如黄赌

毒）。常用的匹配对象包括风险邮件正文的外部链接，网站被篡改的链接，浏览器访问目标地址等。

2.3 情报层级分类

威胁情报可分为四个主要层级，支撑不同层面的安全活动于决策支撑：

层级	内容特征	主要作用
技术情报（TechnicalIntelligence）	恶意 IP、域名、URL、Hash 等 IOC 指标	驱动检测与阻断设备，支撑安全防御动作
战术情报（TacticalIntelligence）	攻击技术、行为模式、TTP（映射 ATT&CK）	支撑威胁建模、规则设计与检测机制优化
操作情报（OperationalIntelligence）	当前攻击活动、目标行业、攻击流程链路	支撑响应行动、研判分析与防护调整
战略情报（StrategicIntelligence）	国家级威胁组织、产业趋势、攻击动机、宏观情报图谱	为管理层制定战略规划、投资决策提供依据

例如，一个 IP 被标记为“疑似 C2 服务器”是技术情报；该 IP 属于使用 Beacon 模块的攻击，是战术情报；它近期攻击了金融行业多个客户，是操作情报；其背后为 APT41 团伙，目的为国家情报收集，是战略情报。对于日常 SOC 过程中，通常为了沟通方便可以简略成战术情报与战略情报。

2.4 关键概念与惯用词

2.4.1 惯用词表

在详细介绍威胁情报之前，下列提供了一个包含了技术、运营、产品三个方向的惯用词表，解释了在威胁情报领域内常用词的含义，以减少信息传递过程中的因概念偏差引起的不必要问题。

简称/缩写	英文全称	说明
APT	Advanced Persistent Threat	高级持续性威胁（组织）
CTI	Cyber Threat Intelligence	网络威胁情报
IOC	Indicators of Compromise	失陷指标
IOA	Indicators of Attack	描述攻击行为、策略和技术模式的情报，用于预测或检测攻击活动。
TIP	Threat Intelligence Platform	威胁情报平台
CVE	Common Vulnerabilities and Exposures	国际通用的漏洞编号与描述标准。
SIEM	Security Information and Event Management	收集、分析、关联安全日志与事件的安全管理平台。
SOAR	Security Orchestration, Automation and Response	用于自动化安全事件处理流程的系统。（安全编排、自动化与响应）
C2	Command and Control	攻击者指挥与控制服务器(CC\CnC\C&C)
IOC Feed	Indicators of Compromise Feed	威胁情报数据源
TTP	Tactics, Techniques, and Procedures	攻击者在网络攻击中的作战模式与手法，用于威胁建模与溯源
IR	Incident Response	针对网络安全事件进行检测、分析、遏制、根除和恢复的全过程。
EDR	Endpoint Detection and Response	终端检测与响应
EPP	Endpoint Protection Platform	终端防护平台
XDR	Extended Detection and Response	扩展检测与响应
MDR	Managed Detection and Response	托管式检测与响应

MTTD	Mean Time to Detect	从威胁或事件发生到被检测出的平均时间。
MTTR	Mean Time to Respond	从发现安全事件到完成处置的平均时间。
PoC	Proof of Concept	验证某技术、漏洞或攻击方法可行性的测试实现
SOC	Security Operations Center	集中监控、分析、响应网络安全事件的团队与平台。安全运营中心。
Inbound	Inbound	入站，外部网络（如互联网）发起，进入本地网络或内部系统的流量
Outbound	Outbound	出站，本地网络或内部系统发起，发送到外部网络的流量。
Threat Actor	Threat Actor	威胁行为者，泛指任何实施恶意活动的个人、团体或国家实体
Campaign	Campaign	攻击行动，威胁组织针对特定目标或目标集群开展的一系列攻击活动
Threat Hunting	Threat Hunting	威胁狩猎，利用海量未知数据挖掘威胁情报
OSINT	Open-Source Intelligence	开源情报
ATT&CK	ATT&CK	特指 MITRE 提出的威胁行为者攻击矩阵模型。

2.4.2 失陷指标（Indicators of Compromise）

失陷指标是用于识别系统或网络上潜在恶意活动的证据数据片段。这些指标可以帮助信息安全和 IT 专业人员检测数据泄露、恶意软件感染或其它威胁活动。失陷指标通常有两大特点，具体性与时效性：

- 失陷指标是具体的、可观察的数据片段，例如恶意 IP 地址、域名、文件哈希值等。
- 失陷指标通常与特定的时间相关，用于检测当前或最近的攻击。

常见的失陷指标如异常的出站网络流量，异常 DNS 请求，可疑的文件 HASH，恶意软件签名等。通过监控失陷指标，组织可以及时发现潜在的攻击，并在攻击的早期阶段检测到威胁采取响应，从而限制损害。

失陷指标与失陷状态是不同的内容，失陷指标是依据与参考，失陷状态是结果。

2.4.3 Tactics, Techniques, and Procedures（TTP）

TTP 是网络空间中对手或对手群体行为的重要概念，用于描述威胁行为者的所作所为及其方式。TTP 包含特定的对手行为，例如攻击模式、恶意软件、漏洞利用等，涉及利用的资源（如工具、基础设施、身份）、针对的受害者信息（谁、什么、在哪里）、目标的漏洞利用、预期效果、相关的杀伤链阶段、处理指南、TTP 信息来源等内容。

在网络安全领域，TTP 详细刻画了攻击者的行为，涵盖战术层面的意图、技术层面的实现方式和具体的操作程序。例如，攻击者可能采用的战术是

- 利用恶意软件窃取信用卡凭证：
    - 相关技术可能包括向潜在受害者发送定向电子邮件
    - 邮件附件包含恶意代码
    - 代码在打开时执行，通过记录键盘输入窃取信用卡信息
    - 使用 HTTP 协议与命令控制服务器通信传输信息
    - 相关程序则可能涉及进行开源研究以识别容易上当的个人，然后制作具有说服力的社会工程电子邮件和文档
    - 创建能够绕过当前防病毒检测的恶意软件/漏洞利用程序
    - 注册名为 mychasebank.org 的域名建立命令控制服务器
    - 从名为 accountsmychasebank@gmail.com 的 Gmail 账户向受害者发送邮件

2.4.4 出入站（Inbound/Outbound）

出入站特指产品适用过程中的检测方向场景，基于不同场景下使用威胁情报做出不同的决策判断。

I. 入站场景（InboundThreatIntelligence）

聚焦于外部威胁源发起的攻击，如互联网到内部网络的攻击流量、外部攻击者对组织网络边界的扫描、漏洞利用等行为，帮助组织提前发现并防御外部入侵。适用于防火墙、IPS、WAF、边界 DPI 设备等安全组件。下列表格是基于入站场景下，不同类型情报的描述和应用举例：

类型	描述	应用举例
恶意 IP 情报	指访问请求来自的 IP 地址具有攻击历史或恶意行为记录	来自黑产扫描器、DDoS 发起源、Bot 网络控制 IP
攻击指纹特征	来自 HTTP/FTP/DNS 等流量中的异常 Payload、协议变形	SQL 注入 Payload、非法 User-Agent 特征
主机画像类情报	来访 IP 为 IDC、VPN、动态住宅 IP、代理节点	用于辅助判别是否为伪装攻击或匿名访问
0day 攻击源	指曾发起 0day 利用攻击的 IP	结合沙箱、蜜罐溯源生成

II. 出站场景（OutboundThreatIntelligence）

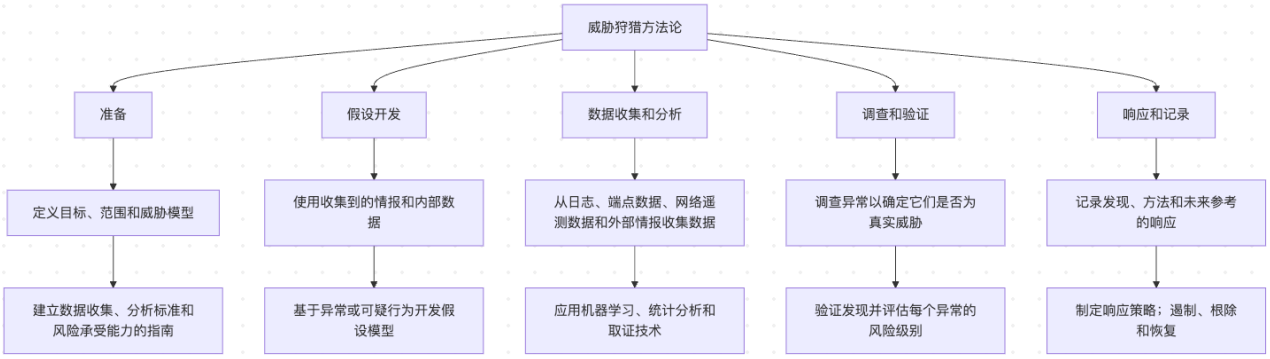
关注组织内部系统或网络向外部发送的异常信息，如被恶意软件感染的主机向外部 C2 服务器传输数据、内部用户泄露敏感信息等，主要用于检测内部失陷主机和数据泄露风险。下列是表格是基于出站场景下，不同类型情报的描述和应用举例：

类型	描述	应用举例
远控 C2 地址	与木马、勒索软件、Bot 等后门通信的控制端地址	beacon.cobalt-host[.]net,xyz.c2server[.]cc
失陷域名/IP	黑产平台、挖矿域名、僵尸网络指挥地址	与“Mirai”、“XLoader”通信域名
风险 URL	被用于钓鱼、诈骗、诱导、下载恶意文件的网址	http[:]//123.aa.cc/offer.exe，假冒银行
DNS 通道域名	高熵、疑似 FastFlux、DGA 生成的出站域名	jfj3kdsa.asd83dj33[.]info,短周期 TTL 快速变更
恶意下载链接	文件哈希、URL 指向恶意载荷	山石沙箱样本来源追溯

2.4.5 威胁狩猎（Threat Hunting）

威胁狩猎是一种通过主动搜索网络和系统中的潜在威胁来识别和中和恶意活动的过程，这些恶意活动可能会被传统的自动化工具（如 SIEM 和 EDR）所遗漏。威胁狩猎的核心在于发现异常和隐藏的威胁，不仅仅是发现新的威胁，还包括改进检测机制和增强威胁情报。通过这种方法，组织可以提前识别潜在的攻击，并在攻击造成严重损害前采取措施进行防御。

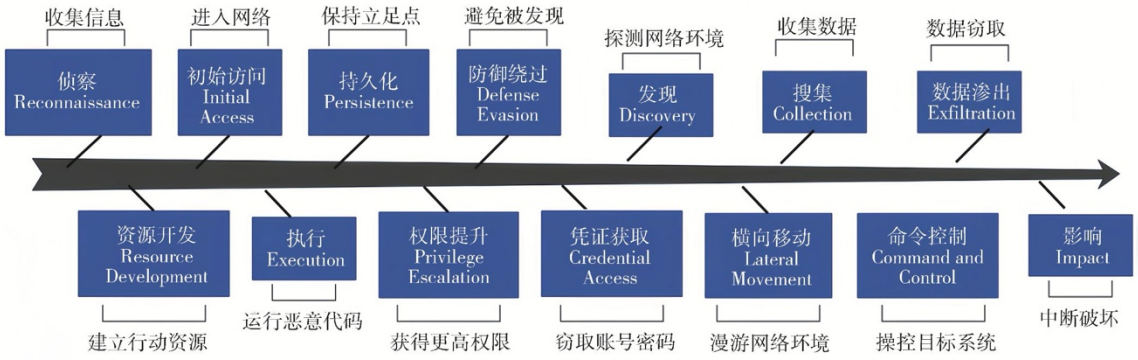
威胁狩猎方法论：



详细威胁狩猎方法见第 X 章。

2.4.6 MITRE ATT&CK

ATT&CK（Adversarial Tactics, Techniques, and Common Knowledge）是一个广泛接受的威胁模型，由美国 MITRE 公司于 2013 年发起，于 2015 年 5 月正式发布。该框架基于真实世界中的攻击事件创建和维护，是一个公开免费、全球可访问的知识库该。框架旨在帮助安全专业人员更好地理解和应对网络威胁，提供了一种标准化的方法来描述攻击者的行为和技术。



MITRE 公司在 1958 年从麻省理工学院林肯实验室分离出来后参与了许多政府项目，2013 年为解决防守方面临的困境，基于真实攻击事件创建了 ATT&CK 框架。自 2015 年发布以来，该框架迭代更新较快，每隔 3 至 6 个月就会完成一次更新，更新内容主要包括战

术、技术、攻击组、软件、缓解措施等。最初专注于攻击者入侵系统后的行为，后来随着企业上云和容器技术的发展，新增了云环境和容器相关的内容，还创建了 ATT&CK for ICS 框架以应对工业控制系统中的网络攻击。

3 山石网科威胁情报服务

3.1 威胁情报的集成模式

3.1.1 威胁情报云查服务

云查服务是指威胁情报的消费方（主要是指安全产品）可通过 Restful 标准的 API 模式，批量化高并发的使用云端情报查询服务的一种机读接口调用机制。优先采用云查 API 接口而非私有化部署模式的场景包括：

- 实时性与快速响应：云查服务依托云端实时更新的全球威胁数据库，能够提供最新的情报数据，帮助企业实现对最新攻击特征和恶意 IP/域名的快速识别。这种服务模式相较于传统的私有化部署，能够显著减少因情报滞后带来的安全风险。
- 低成本与低门槛：对于资源有限的中小企业，云查服务提供了一种低成本、低门槛的安全能力集成方式。通过 SaaS 模式，企业无需承担高昂的硬件成本和复杂的运维压力，即可实现高效的安全能力集成。
- 自动化整合多来源数据：云查服务能够自动整合来自不同来源的威胁情报，形成统一的查询服务。这种服务模式不仅提高了情报查询的实时性，还降低了私有化部署所需的额外采购成本、定制化开发成本和时间成本。

与此同时，对于提供云查服务的供应者也有相应的要求，包括：需要坚实的云端计算、存储、网络、软件资源为基础，以便支撑情报业务所需的高频更新，高并发，可弹性扩展性能。

3.1.2 私有化对接方案

旨在通过将云端威胁情报数据进行本地化存储，实现在隔离网络环境下的情报查询应用。该方案采用威胁情报静态特征库，以适应不同的安全产品需求。

该方案下，防火墙上的情报应用引擎被称为 C2 引擎（Command and Control 引擎）和 AV 引擎（Antivirus 引擎）。C2 引擎专注于识别和阻断命令与控制流量，而 AV 引擎则

负责检测和清除恶意软件。这种双引擎架构提供了更全面的威胁检测和防护能力。主要适用场景包括：

- 网络环境严格隔离：对于国内部分行业和企业，由于较高的网络连接管控条件，安全产品需要部署在无法直连互联网的环境下。山石威胁情报私有化对接方案能够满足这些环境下情报数据的更新需求，确保安全产品在本地完成且使用过程全程无需联网。
- 情报查询性能要求较高：部分安全设备（如防火墙）主要应用模式为大吞吐量的数据通场景，威胁情报需要满足低性能占用情况下的高速查询能力。山石方案通过本地化部署的静态特征库，提供了高性能的情报查询服务，满足了实时检测需求。
- 情报数据的本地管理运营：除了外部威胁情报源提供的互联网风险监测视野，部分大中型企业会结合自身的安全运营需求，以及特殊场景（如重保、攻防演练），增补部分本地自运营的威胁情报数据。山石方案支持情报数据的本地化管理和运营，以实现威胁检测效能的提升。

3.2 山石云查核心服务矩阵

数据类型	服务内容	接口类型	接口等级
Ipv4 情报	静态信誉情报	特征库 API	基础
	基础设施画像情报	API	基础
	攻击源情报（Inbound）	API	高级
	出站信誉情报（Outbound）	API	高级
	失陷情报（Outbound）	API	高级
	归属威胁行为者	特征库 API	高级
	关联文件情报与 TTPs	API	高级
域名情报	静态信誉情报	特征库	基础
	基础设施画像情报	特征库 API	基础
	web 站点分类	API	基础

	出站信誉情报（Outbound）	API	高级
	归属威胁行为者	特征库 API	高级
	关联文件情报与 TTPs	API	高级
Hash 情报	静态信誉情报	特征库 API	基础
	分析情报	API	基础
	深度分析情报（SandBox）	API	高级
URI	静态信誉情报（Outbound）	特征库 API	基础

详细赋能场景见第 x 章。接口详细描述见 x 节。

3.2.1 静态信誉情报（Static Reputation Intelligence）

静态信誉情报是指基于长期监测与情报积累，对 IP、域名、Hash、URI 等对象进行的威胁信誉（黑、灰、白或未知）评分（置信度与生命周期）与威胁分类（Malware、Trojan、Virus、Scanner 等），不依赖即时流量分析即可判断其潜在风险。

典型场景：

- 边界防御：防火墙、IPS 在未建立会话前即拦截高风险 IP/域名
- 邮件安全：在邮件网关拦截来自低信誉域名的钓鱼邮件
- Web 应用防护：WAF 在 URL 请求到达业务前阻断已知恶意 URI

3.2.2 基础设施画像情报（Infrastructure Fingerprint Intelligence）

通过长期互联网测绘、开放端口扫描、证书指纹、DNS 记录等技术手段，对 IPv4、域名的网络基础设施进行归属和功能识别。如云服务 IP、IDC、代理 IP、VPN、CDN 等归属信息。

典型场景：

- 攻击面管理：帮助企业识别暴露在公网的云服务、VPN、CDN、代理 IP 等
- 威胁归因：通过基础设施特征将不同攻击活动关联到同一威胁组织

3.2.3 「IPv4」攻击源情报（Inbound）

针对入站方向（Inbound）检测到的恶意 IPv4 地址的情报数据，这些地址往往为攻击源、扫描器、僵尸网络节点等。

典型场景：

- 入侵防御：在边界防护设备上快速阻断来自高风险攻击源的连接
- 安全运营中心（SOC）：事件关联分析，识别是否与近期攻击活动相关
- 蜜罐分析：基于蜜罐捕获的源 IP 直接与威胁情报比对，缩短溯源时间

3.2.4 出站信誉情报（Outbound）

针对内部资产访问外部网络时的目的 IP/域名进行风险评估，用于检测内部主机是否访问恶意资源。如 APT 攻击、蠕虫木马、僵尸网络、勒索软件等的远控服务器地址情报数据库。

典型场景：

- 数据外泄检测：识别内网主机是否与已知 C2（Command & Control）服务器通信
- 恶意软件通信阻断：阻止恶意程序与外部下载站点或更新服务器连接

3.2.5 「IPv4」失陷情报（Outbound Compromise Intelligence）

监测内部资产是否被攻击者控制并对外发起恶意行为，例如参与 DDoS、暴力破解、垃圾邮件发送等。是出站信誉情报基础上的高精 IP 情报，用于主机失陷确认。

3.2.6 归属威胁行为者

将观测到的恶意活动、基础设施、恶意文件等与已知威胁行为者（APT、网络犯罪团伙等）建立关联，提供威胁画像。

典型场景：

- 高级威胁检测：发现与特定 APT 组织相关的攻击行为
- 威胁狩猎（Threat Hunting）：基于归因结果主动搜寻潜伏在环境中的同类活动
- 战略情报分析：帮助管理层理解当前面临的主要威胁对手

3.2.7 关联文件情报与 TTPs

分析威胁资产及其关联的文件和公开报告、行为模式，提炼战术、技术与程序（TTPs），用于检测与防御类似攻击。

典型场景：

- 攻击链还原：重构威胁事件的完整流程
- 规则生成：根据 TTPs 制定检测规则（如 YARA、Sigma）
- 跨事件关联：发现不同事件中重复使用的工具链或策略

3.2.8 「HASH」分析与深度分析情报

山石内部拥有亿级威胁样本量，同时拥有高效的动静态沙箱。通过样本 Hash 快速定位已知威胁文件检测结果，同时支持实时静态、动态（沙箱）分析，提供恶意软件的行为分析报告、黑白判定、风险等级和报毒名等。

典型场景：

- 样本鉴定：快速判断文件是否为已知恶意样本
- 自动化响应：基于分析结果阻断或清除相关文件

3.3 云查（API）服务接口

3.3.1 云查服务授权与接入方式

API 体验版

标签：免费使用、查询次数有限、不可商用

使用方式：所有注册用户可免费使用，只需注册山石云平台并获取 API 密钥即可；部分功能仅限 API 正式版客户使用。

权限说明

- 支持 信誉查询 和 高级查询；
- 每分钟查询速率、每天最大查询次数有限制，有效期无限制；
- 不可用于商业产品或服务。

API 正式版

标签：查询频率不限、可商用

使用方式：通过以下方式联系获取：

- 电子邮件：tac@hillstonenet.com
- 客户服务热线：400-693-0555

权限说明

- 支持 信誉查询 和 高级查询；
- 查询速率无限制，每天最大查询次数由许可证规格决定；
- 可用于商业产品或服务；
- 仅限配合智源产品使用。

请求结构

- 服务地址：接口接入域名 ti.hillstonenet.com.cn
- 通信协议：所有接口均通过 HTTPS 进行通信，提供高安全性的通信通道
- 请求方法：仅支持 Get 的 HTTP 请求方法
- 字符编码：均使用 UTF-8 编码

认证方式

Header 部分

参数名称	说明	是否必须	类型	参数位置
X-Auth-Token	用户 apiKey	是	String	Headers
ACCEPT	返回参数格式，目前只支持 json（application/json）	否	String	Headers
X-API-Version	获取版本（当前值为 1.0.0），不携带默认最新版本	否	String	Headers
X-API-Language	获取的语言（en）	否	String	Headers

URL 部分

参数名称	说明	是否必须	类型	参数位置
key	查询值	是	String	url 中

3.3.2 IPv4 静态信誉情报

3.3.2.1 查询方法

请求地址：/api/ip/reputation

请求方法：GET

3.3.2.2 请求参数说明

#	参数名称	必选	类型	描述	示例
1	key	是	String	ip 值	1.1.1.1

3.3.2.3 响应参数说明

#	参数名称	类型	描述
1	response_code	Int	详见附录 8.1
2	response_msg	String	返回结果描述信息，和返回码对应
3	ip_address	String	ip 值
4	result	String	**恶意程度**

			取值恶意、可疑、正常、未知 ● malicious = 恶意 ● suspicious = 可疑 ● normal = 正常 ● unknown =未知
5	threat_type	Array(Stirng)	情报对应的**威胁标签**，英文名 威胁标签属于标签的一级分类 取值详见附录 8.2
6	flow_direction	Int	情报对应的出入站分类 出站-1，不确定 0，入站 1

3.3.2.4 请求示例（Curl/Java/Pyhton）

- Curl

```
curl "https://ti.hillstonenet.com.cn/api/ip/reputation?key={key}" \
 -H 'X-Auth-Token: <your API key>' \
 -H 'ACCEPT: application/json' \
 -H 'X-API-Version: 1.0.0' \
 -H 'X-API-Language: en'
```

- Java

```
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;

public class IpReputationReport {
   public static void main(String[] args) throws Exception{
      HttpClient httpClient;
      HttpGet getMethod;
      HttpResponse response;
      String reponseContent;
      httpClient = HttpClients.createDefault();
      getMethod = new HttpGet("https://ti.hillstonenet.com.cn/api/ip/reputation?key={key}");
      getMethod.addHeader("X-Auth-Token", "{your api key}");
      getMethod.addHeader("ACCEPT", "application/json");
      getMethod.addHeader("X-API-Version", "1.0.0");
      getMethod.addHeader("X-API-Language", "en");
```

```
        response = httpClient.execute(getMethod);
        HttpEntity httpEntity = response.getEntity();
        reponseContent = EntityUtils.toString(httpEntity);
        EntityUtils.consume(httpEntity);
        System.out.println(reponseContent);
    }
}
```

- Pyhton

```
import urllib2

url = 'https://ti.hillstonenet.com.cn/api/ip/reputation?key={your key}'
headers = {'X-Auth-Token':'{your api key}',
          'ACCEPT':'application/json',
          'X-API-Version':'1.0.0',
          'X-API-Language':'en'}
data = None
req = urllib2.Request(url, data, headers)
response = urllib2.urlopen(req)
report = response.read()
print report
```

3.3.2.5 响应示例（JSON）

```
{
        "data": {
                "result": "normal",
                "threat_type": [
                        "CnC"
                ],
                "ip_address": "101.132.109.94",
                "flow_direction": -1
        },
        "response_code": 0,
        "response_msg": "OK"
}
```

3.3.3 DOMAIN 静态信誉情报

3.3.3.1 查询方法

请求地址：/api/domain/reputation
请求方法：GET

3.3.3.2 请求参数说明

#	参数名称	必选	类型	描述	示例
1	key	是	String	域名	baidu.com

3.3.3.3 响应参数说明

#	参数名称	类型	描述
1	response_code	Int	详见附录 8.1
2	response_msg	String	返回结果描述信息，和返回码对应
3	domain_name	String	域名
4	result	String	**恶意程度** 取值恶意、可疑、正常、未知 ● malicious = 恶意 ● suspicious = 可疑 ● normal = 正常 ● unknown =未知
5	threat_type	Array(String)	情报对应的**威胁标签**，英文名 威胁标签属于标签的一级分类 取值详见附录 8.2
6	flow_direction	Int	情报对应的出入站分类 出站-1，不确定 0，入站 1

3.3.3.4 请求示例（Curl/Java/Python）

- Curl

```
curl "https://ti.hillstonenet.com.cn/api/domain/reputation?key={key}" \
  -H 'X-Auth-Token: <your API key>' \
  -H 'ACCEPT: application/json' \
```

```
  -H 'X-API-Version: 1.0.0' \
  -H 'X-API-Language: en'
```

- Java

```
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;

public class IpReputationReport {
   public static void main(String[] args) throws Exception{
      HttpClient httpClient;
      HttpGet getMethod;
      HttpResponse response;
      String reponseContent;
      httpClient = HttpClients.createDefault();
      getMethod = new HttpGet("https://ti.hillstonenet.com.cn/api/domain/reputation?key={key}");
      getMethod.addHeader("X-Auth-Token", "{your api key}");
      getMethod.addHeader("ACCEPT", "application/json");
      getMethod.addHeader("X-API-Version", "1.0.0");
      getMethod.addHeader("X-API-Language", "en");
      response = httpClient.execute(getMethod);
      HttpEntity httpEntity = response.getEntity();
      reponseContent = EntityUtils.toString(httpEntity);
      EntityUtils.consume(httpEntity);
      System.out.println(reponseContent);
   }
}
```

- Python

```
import urllib2

url = 'https://ti.hillstonenet.com.cn/api/domain/reputation?key={your key}'
headers = {'X-Auth-Token':'{your api key}',
         'ACCEPT':'application/json',
         'X-API-Version':'1.0.0',
         'X-API-Language':'en'}
data = None
req = urllib2.Request(url, data, headers)
response = urllib2.urlopen(req)
report = response.read()
print report
```

3.3.3.5 响应示例（JSON）

```
{
        "data": {
                "result": "normal",
                "threat_type": [
                        "CnC"
                ],
                "domain_name": "baidu.com",
                "flow_direction": -1
        },
        "response_code": 0,
        "response_msg": "OK"
}
```

3.3.4 HASH 静态信誉情报

3.3.4.1 查询方法

请求地址：/api/file/reputation

请求方法：GET

3.3.4.2 请求参数说明

#	参数名称	必选	类型	描述	示例
1	key	是	String	文件哈希值 md5/sha1/sha256/sha512	b0b89921493c71fdc9477313f5c90fd0

3.3.4.3 响应参数说明

#	参数名称	类型	描述
1	response_code	Int	详见附录 8.1
2	response_msg	String	返回结果描述信息，和返回码对应
3	md5	String	文件 md5 值
4	sha1	String	文件 sha1 值

5	sha256	String	文件 sha256 值
6	result	String	**恶意程度** 取值恶意、可疑、正常、未知 ● malicious = 恶意 ● suspicious = 可疑 ● normal = 正常 ● unknown =未知
7	threat_type	Array(String)	情报对应的**威胁标签**，英文名 威胁标签属于标签的一级分类 取值详见附录 8.2
8	flow_direction	Int	情报对应的出入站分类 出站-1，不确定 0，入站 1

3.3.4.4 请求示例（Curl/Java/Python）

- Curl

```
curl "https://ti.hillstonenet.com.cn/api/file/reputation?key={key}" \
 -H 'X-Auth-Token: <your API key>' \
 -H 'ACCEPT: application/json' \
 -H 'X-API-Version: 1.0.0' \
 -H 'X-API-Language: en'
```

- Java

```
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;

public class IpReputationReport {
   public static void main(String[] args) throws Exception{
      HttpClient httpClient;
      HttpGet getMethod;
      HttpResponse response;
      String reponseContent;
      httpClient = HttpClients.createDefault();
      getMethod = new HttpGet("https://ti.hillstonenet.com.cn/api/file/reputation?key={key}");
```

```
        getMethod.addHeader("X-Auth-Token", "{your api key}");
        getMethod.addHeader("ACCEPT", "application/json");
        getMethod.addHeader("X-API-Version", "1.0.0");
        getMethod.addHeader("X-API-Language", "en");
        response = httpClient.execute(getMethod);
        HttpEntity httpEntity = response.getEntity();
        reponseContent = EntityUtils.toString(httpEntity);
        EntityUtils.consume(httpEntity);
        System.out.println(reponseContent);
    }
}
```

- Python

```
import urllib2

url = 'https://ti.hillstonenet.com.cn/api/file/reputation?key={your key}'
headers = {'X-Auth-Token':'{your api key}',
          'ACCEPT':'application/json',
          'X-API-Version':'1.0.0',
          'X-API-Language':'en'}
data = None
req = urllib2.Request(url, data, headers)
response = urllib2.urlopen(req)
report = response.read()
print report
```

3.3.4.5 响应示例（JSON）

```
{
        "data": {
                "result": "normal",
                "threat_type": [
                        "CnC"
                ],
                "md5": "b10a8db164e0754105b7a99be72e3fe5",
                "sha1": "0a4d55a8d778e5022fab701977c5d840bbc486d0",
                "sha256": "a591a6d40bf420404a011733cfb7b190d62c65bf0bcda32b57b277d9ad9f146",
                "flow_direction": -1
        },
        "response_code": 0,
        "response_msg": "OK"
}
```

3.3.5 URL 静态信誉情报

3.3.5.1 查询方法

请求地址：/api/url/reputation

请求方法：GET

3.3.5.2 请求参数说明

#	参数名称	必选	类型	描述	示例
1	key	是	String	url 值	http://anguiillanet.com/freeme/fre.php

3.3.5.3 响应参数说明

#	参数名称	类型	描述
1	response_code	Int	详见附录 8.1
2	response_msg	String	返回结果描述信息，和返回码对应
3	url	String	url 值
4	result	String	**恶意程度** 取值恶意、可疑、正常、未知 ● malicious = 恶意 ● suspicious = 可疑 ● normal = 正常 ● unknown =未知
5	threat_type	Array(String)	情报对应的**威胁标签**，英文名 威胁标签属于标签的一级分类 取值详见附录 8.2
6	flow_direction	Int	情报对应的出入站分类 出站-1，不确定 0，入站 1

3.3.5.4 请求示例（Curl/Java/Python）

- Curl

```
curl "https://ti.hillstonenet.com.cn/api/url/reputation?key={key}" \
 -H 'X-Auth-Token: <your API key>' \
 -H 'ACCEPT: application/json' \
 -H 'X-API-Version: 1.0.0' \
 -H 'X-API-Language: en'
```

- Java

```
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;

public class IpReputationReport {
   public static void main(String[] args) throws Exception{
      HttpClient httpClient;
      HttpGet getMethod;
      HttpResponse response;
      String reponseContent;
      httpClient = HttpClients.createDefault();
      getMethod = new HttpGet("https://ti.hillstonenet.com.cn/api/url/reputation?key={key}");
      getMethod.addHeader("X-Auth-Token", "{your api key}");
      getMethod.addHeader("ACCEPT", "application/json");
      getMethod.addHeader("X-API-Version", "1.0.0");
      getMethod.addHeader("X-API-Language", "en");
      response = httpClient.execute(getMethod);
      HttpEntity httpEntity = response.getEntity();
      reponseContent = EntityUtils.toString(httpEntity);
      EntityUtils.consume(httpEntity);
      System.out.println(reponseContent);
   }
}
```

- Python

```
import urllib2

url = 'https://ti.hillstonenet.com.cn/api/url/reputation?key={your key}'
headers = {'X-Auth-Token':'{your api key}',
         'ACCEPT':'application/json',
         'X-API-Version':'1.0.0',
         'X-API-Language':'en'}
data = None
```

```
req = urllib2.Request(url, data, headers)
response = urllib2.urlopen(req)
report = response.read()
print report
```

3.3.5.5 响应示例（JSON）

```
{
        "data": {
                "result": "normal",
                "threat_type": [
                        "CnC"
                ],
                "url": "http://anguiillanet.com/freeme/fre.php",
                "flow_direction": -1
        },
        "response_code": 0,
        "response_msg": "OK"
}
```

3.3.6 IPv4 高级接口

3.3.6.1 查询方法

请求地址：/api/ip/detail

请求方法：GET

3.3.6.2 请求参数说明

#	参数名称	必选	类型	描述	示例
1	key	是	String	ip 值	1.1.1.1

3.3.6.3 响应参数说明

#	参数名称	类型	描述

1	response_code	Int	详见附录 8.1
2	response_msg	String	返回结果描述信息，和返回码对应
3	ip_address	String	ip 值
4	basic_info	Object	ip 基本信息，包括： network：网络 carrier：运营商 location：地理位置 country：国家 country_code：国家代码 province：省份 city：城市 longtitude：维度 latitude：维度
5	asn	String	自治系统信息
6	tags	Array(String)	标签
7	rdns_list	Array(Object)	可逆 dns 记录，包括两个字段： domain_name：域名 lookup_time：时间戳（最多 10 个）
8	current_domains	Array(String)	当前域名
9	history_domains	Array(Object)	历史域名 domain_name：域名 date：时间戳（最多 10 个）
10	download_files	Array(String)	下载的文件样本 hash 值（最多 10 个）
11	referer_files	Array(String)	存在相关 ip 的文件 hash 值（只返回恶意的，最多 10 个）
12	connect_files	Array(String)	相关文件 hash 值（只返回恶意的，最多 10 个）
13	ports	Array(Object)	端口信息： port：端口 module：应用协议 product：应用名称 version：应用版本 update_time：更新时间
14	additional_info	Object	额外信息

15	result	String	**恶意程度** 取值恶意、可疑、正常、未知 ● malicious = 恶意 ● suspicious = 可疑 ● normal = 正常 ● unknown =未知
16	threat_type	Array(String)	情报对应的**威胁标签**，英文名 威胁标签属于标签的一级分类 取值详见附录 8.2
17	flow_direction	Int	情报对应的出入站分类 出站-1，不确定 0，入站 1

3.3.6.4 请求示例（Curl/Java/Python）

- Curl

```
curl "https://ti.hillstonenet.com.cn/api/ip/detail?key={key}" \
 -H 'X-Auth-Token: <your API key>' \
 -H 'ACCEPT: application/json' \
 -H 'X-API-Version: 1.0.0' \
 -H 'X-API-Language: en'
```

- Java

```
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;

public class IpReputationReport {
   public static void main(String[] args) throws Exception{
      HttpClient httpClient;
      HttpGet getMethod;
      HttpResponse response;
      String reponseContent;
      httpClient = HttpClients.createDefault();
      getMethod = new HttpGet("https://ti.hillstonenet.com.cn/api/ip/detail?key={key}");
      getMethod.addHeader("X-Auth-Token", "{your api key}");
      getMethod.addHeader("ACCEPT", "application/json");
      getMethod.addHeader("X-API-Version", "1.0.0");
```

```
        getMethod.addHeader("X-API-Language", "en");
        response = httpClient.execute(getMethod);
        HttpEntity httpEntity = response.getEntity();
        reponseContent = EntityUtils.toString(httpEntity);
        EntityUtils.consume(httpEntity);
        System.out.println(reponseContent);
    }
}
```

- Python

```
import urllib2

url = 'https://ti.hillstonenet.com.cn/api/ip/detail?key={your key}'
headers = {'X-Auth-Token':'{your api key}',
          'ACCEPT':'application/json',
          'X-API-Version':'1.0.0',
          'X-API-Language':'en'}
data = None
req = urllib2.Request(url, data, headers)
response = urllib2.urlopen(req)
report = response.read()
print report
```

3.3.6.5 响应示例（JSON）

```
{
        "data": {
                "result": "normal",
                "ip_address": "101.132.109.94",
                "basic_info": {
                        "location": {}
                },
                "current_domains": [],
                "history_domains": [
                        {
                                "date": 1595289600000,
                                "domain_name": "www.hillstonenet.com"
                        },
                        {
                                "date": 1595116800000,
                                "domain_name": "vpn.tac.hillstonenet.com"
                        },
                        {
                                "date": 1594080000000,
```

```
                        "domain_name": "vpnsh.tac.hillstonenet.com"
                    },
                    {
                        "date": 1571185998000,
                        "domain_name": "docs.hillstonenet.com.cn"
                    },
                    {
                        "date": 1571185998,
                        "domain_name": "docs.hillstonenet.com.cn"
                    }
                ],
                "download_files": [],
                "referer_files": [],
                "connect_files": []
            },
            "response_code": 0,
            "response_msg": "OK"
}
```


3.3.7 DOMAIN 高级接口

3.3.7.1 查询方法

请求地址：/api/domain/detail

请求方法：GET

3.3.7.2 请求参数说明

#	参数名称	必选	类型	描述	示例
1	key	是	String	域名	baidu.com

3.3.7.3 响应参数说明

#	参数名称	类型	描述
1	response_code	Int	详见附录 8.1
2	response_msg	String	返回结果描述信息，和返回码对应

3	domain_name	String	域名
4	current_whois	String	Whois 信息
5	tags	Array(String)	标签
6	dns_records	Array(String)	dns 解析记录（最多 10 个）
7	current_ips	Array(String)	当前解析的 ip（最多 10 个）
8	history_ips	Array(Object)	历史域名 ip_address：ip 地址 date：时间戳（最多 10 个）
9	sub_domains	Array(String)	子域名（最多 10 个）
10	domain_siblings	Array(String)	相关域名（最多 10 个）
11	download_files	Array(String)	下载的文件样本 hash 值（最多 10 个）
12	referer_files	Array(String)	存在相关 ip 的文件 hash 值（只返回恶意的，最多 10 个）
13	connect_files	Array(String)	相关文件 hash 值（只返回恶意的，最多 10 个）
14	additional_info	Object	额外信息
15	result	String	**恶意程度** 取值恶意、可疑、正常、未知 ● malicious = 恶意 ● suspicious = 可疑 ● normal = 正常 ● unknown =未知
16	threat_type	Array(String)	情报对应的**威胁标签**，英文名 威胁标签属于标签的一级分类 取值详见附录 8.2
17	flow_direction	Int	情报对应的出入站分类 出站-1，不确定 0，入站 1

3.3.7.4 请求示例（Curl/Java/Python）

- Curl

```
curl "https://ti.hillstonenet.com.cn/api/domain/detail?key={key}" \
 -H 'X-Auth-Token: <your API key>' \
 -H 'ACCEPT: application/json' \
 -H 'X-API-Version: 1.0.0' \
```

```
  -H 'X-API-Language: en'
```

- Java

```
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;

public class IpReputationReport {
   public static void main(String[] args) throws Exception{
      HttpClient httpClient;
      HttpGet getMethod;
      HttpResponse response;
      String reponseContent;
      httpClient = HttpClients.createDefault();
      getMethod = new HttpGet("https://ti.hillstonenet.com.cn/api/domain/detail?key={key}");
      getMethod.addHeader("X-Auth-Token", "{your api key}");
      getMethod.addHeader("ACCEPT", "application/json");
      getMethod.addHeader("X-API-Version", "1.0.0");
      getMethod.addHeader("X-API-Language", "en");
      response = httpClient.execute(getMethod);
      HttpEntity httpEntity = response.getEntity();
      reponseContent = EntityUtils.toString(httpEntity);
      EntityUtils.consume(httpEntity);
      System.out.println(reponseContent);
   }
}
```

- Python

```
import urllib2

url = 'https://ti.hillstonenet.com.cn/api/domain/detail?key={your key}'
headers = {'X-Auth-Token':'{your api key}',
         'ACCEPT':'application/json',
         'X-API-Version':'1.0.0',
         'X-API-Language':'en'}
data = None
req = urllib2.Request(url, data, headers)
response = urllib2.urlopen(req)
report = response.read()
print report
```

3.3.7.5 响应示例（JSON）


```
{
        "data": {
                "result": "normal",
                "domain_siblings": [
                        "partner.hillstonenet.com",
                        "swupdate.hillstonenet.com",
                        "exchange.hillstonenet.com",
                        "support.hillstonenet.com",
                        "sandbox.hillstonenet.com",
                        "images.hillstonenet.com",
                        "demo3.hillstonenet.com",
                        "exam1.hillstonenet.com",
                        "docs.hillstonenet.com",
                        "bor.hillstonenet.com"
                ],
                "domain_name": "www.hillstonenet.com",
                "current_whois": "Creation Date: 2003-10-25T00:17:21Z\nDNSSEC: unsigned\nDomain Name: HILLSTONENET.COM\nDomain Status: ok https://icann.org/epp#ok\nName Server: NS1.DNSV3.COM\nName Server: NS2.DNSV3.COM\nRegistrar Abuse Contact Email: DomainAbuse@service.aliyun.com\nRegistrar Abuse Contact Phone: +86.95187\nRegistrar IANA ID: 420\nRegistrar URL: http://www.net.cn\nRegistrar WHOIS Server: grs-whois.hichina.com\nRegistrar: Alibaba Cloud Computing (Beijing) Co., Ltd.\nRegistry Domain ID:105573727_DOMAIN_COM-VRSN\nRegistry Expiry Date: 2022-01-31T04:59:59Z\nUpdated Date: 2019-03-25T14:22:59Z",
                "dns_records": [
                        "45.56.122.29"
                ],
                "current_ips": [
                        "45.56.122.29"
                ],
                "history_ips": [
                        {
                                "date": 1595289600000,
                                "ip_address": "101.132.109.94"
                        },
                        {
                                "date": 1592956800000,
                                "ip_address": "45.56.122.29"
                        },
                        {
                                "date": 1589155200000,
                                "ip_address": "45.56.122.29"
                        },
                        {
```

```
                        "date": 1542153600000,
                        "ip_address": "221.224.30.134"
                },
                {
                        "date": 1527897600000,
                        "ip_address": "221.224.30.138"
                },
                {
                        "date": 1394064000000,
                        "ip_address": "162.144.47.50"
                },
                {
                        "date": 1370908800000,
                        "ip_address": "121.52.208.205"
                },
                {
                        "date": 1575020870,
                        "ip_address": "45.56.122.29"
                },
                {
                        "date": 1394064000,
                        "ip_address": "162.144.47.50"
                },
                {
                        "date": 1370908800,
                        "ip_address": "121.52.208.205"
                }
        ],
        "sub_domains": [
                "partner.hillstonenet.com",
                "swupdate.hillstonenet.com",
                "exchange.hillstonenet.com",
                "support.hillstonenet.com",
                "sandbox.hillstonenet.com",
                "exam1.hillstonenet.com",
                "docs.hillstonenet.com",
                "bor.hillstonenet.com",
                "partners.hillstonenet.com",
                "images-en.hillstonenet.com"
        ],
        "download_files": [
                "bfe0e842c98c86ab26f26808cdec7ba9"
        ],
        "referer_files": [],
        "connect_files": []
},
"response_code": 0,
"response_msg": "OK"
```

```
}
```

3.3.8 HASH 高级接口

3.3.8.1 查询方法

请求地址：/api/file/detail
请求方法：GET

3.3.8.2 请求参数说明

#	参数名称	必选	类型	描述	示例
1	key	是	String	文件哈希值 md5/sha1/sha256/sha512	b0b89921493c71fdc9477313f5c90fd0

3.3.8.3 响应参数说明

#	参数名称	类型	描述
1	response_code	Int	详见附录 8.1
2	response_msg	String	返回结果描述信息，和返回码对应
3	sha256	String	文件 sha256 值
4	sha1	String	文件 sha1 值
5	md5	String	文件 md5 值
6	basic_info	Object	文件基本信息，包括： file_size：文件大小 file_type：文件类型 first_seen：首次发现时间 last_seen：最后发现时间 scan_time：扫描时间
7	tags	Array(String)	标签
8	malware_family	String	恶意家族
9	connect_ips	Array(String)	该文件连接过的 ip（最多 10 个）

10	download_ips	Array(String)	下载过该文件的 ip（最多 10 个）
11	referer_ips	Array(String)	该文件包含的 ip（最多 10 个）
12	download_domains	Array(String)	下载过该文件的 domain（最多 10 个）
13	referer_domains	Array(String)	该文件包含的 domain（最多 10 个）
14	connect_domains	Array(String)	该文件连接过的 domain（最多 10 个）
15	additional_info	Object	额外信息
16	result	String	**恶意程度** 取值恶意、可疑、正常、未知 ● malicious = 恶意 ● suspicious = 可疑 ● normal = 正常 ● unknown =未知
17	threat_type	Array(String)	情报对应的**威胁标签**，英文名 威胁标签属于标签的一级分类 取值详见附录 8.2
18	flow_direction	Int	情报对应的出入站分类 出站-1，不确定 0，入站 1

3.3.8.4 请求示例（Curl/Java/Python）

- Curl

```
curl "https://ti.hillstonenet.com.cn/api/file/detail?key={key}" \
 -H 'X-Auth-Token: <your API key>' \
 -H 'ACCEPT: application/json' \
 -H 'X-API-Version: 1.0.0' \
 -H 'X-API-Language: en'
```

- Java

```
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;

public class IpReputationReport {
   public static void main(String[] args) throws Exception{
```

```
        HttpClient httpClient;
        HttpGet getMethod;
        HttpResponse response;
        String reponseContent;
        httpClient = HttpClients.createDefault();
        getMethod = new HttpGet("https://ti.hillstonenet.com.cn/api/file/detail?key={key}");
        getMethod.addHeader("X-Auth-Token", "{your api key}");
        getMethod.addHeader("ACCEPT", "application/json");
        getMethod.addHeader("X-API-Version", "1.0.0");
        getMethod.addHeader("X-API-Language", "en");
        response = httpClient.execute(getMethod);
        HttpEntity httpEntity = response.getEntity();
        reponseContent = EntityUtils.toString(httpEntity);
        EntityUtils.consume(httpEntity);
        System.out.println(reponseContent);
    }
}
```

- Python

```
import urllib2

url = 'https://ti.hillstonenet.com.cn/api/file/detail?key={your key}'
headers = {'X-Auth-Token':'{your api key}',
          'ACCEPT':'application/json',
          'X-API-Version':'1.0.0',
          'X-API-Language':'en'}
data = None
req = urllib2.Request(url, data, headers)
response = urllib2.urlopen(req)
report = response.read()
print report
```

3.3.8.5 响应示例（JSON）

```
{
        "data": {
                "result": "malicious",
                "sha256":
"c759874d5935483b385e5c04d928944f726ccd6b4c5f3c5ca73cfabc908d2a39",
                "sha1": "c99f4500a4c830784246b95caf4bc9aec5d12db0",
                "md5": "b0b89921493c71fdc9477313f5c90fd0",
                "basic_info": {
                        "file_size": 2831625,
                        "file_type": "Win32 EXE",
                        "first_seen": 1593271898000,
```

```
                        "last_seen": 1593396305000,
                        "scan_date": 1593396305000
                },
                "connect_ips": [
                        "114.114.114.114"
                ],
                "download_ips": [],
                "referer_ips": [],
                "connect_domains": [],
                "download_domains": []
        },
        "response_code": 0,
        "response_msg": "OK"
}
```


3.3.9 URL 高级接口

3.3.9.1 查询方法

请求地址：/api/url/detail

请求方法：GET

3.3.9.2 请求参数说明

#	参数名称	必选	类型	描述	示例
1	key	是	String	url 值	http://anguiillanet.com/freeme/fre.php

3.3.9.3 响应参数说明

#	参数名称	类型	描述
1	response_code	Int	详见附录 8.1
2	response_msg	String	返回结果描述信息，和返回码对应
3	url	String	url 值
4	hash_sha256	String	url 的 hash 值

5	first_seen	Long	首次发现时间
6	last_seen	Long	最后发现时间
7	scan_date	Long	扫描时间
8	scan_result	Object	扫描结果，包括： positives：返回为黑的引擎个数 total：返回检测结果的引擎个数 total2：部署的引擎个数
9	scan_detail	Array(Object)	扫描细节，包括： engine_name：扫描引擎名称 detected：检测是否恶意 category：检测类型 result：恶意软件类型 version：引擎版本 scan_time：结果记录时间
10	resolution	String	该 url 解析出的 ip
11	related_ips	Array(String)	关联的 ip
12	related_domains	Array(String)	关联的 domain
13	related_files	Array(String)	关联的 file
14	additional_info	Object	额外信息，包括： content_hash_sha256：响应内容 sha256 response_code：响应码 html_meta：html 数据 response_length：响应长度 response_header：响应头额外信息
15	result	String	**恶意程度** 取值恶意、可疑、正常、未知 ● malicious = 恶意 ● suspicious = 可疑 ● normal = 正常 ● unknown =未知
16	threat_type	Array(String)	情报对应的**威胁标签**，英文名 威胁标签属于标签的一级分类 取值详见附录 8.2
17	flow_direction	Int	情报对应的出入站分类

			出站-1，不确定 0，入站 1

3.3.9.4 请求示例（Curl/Java/Python）

- Curl

```
curl "https://ti.hillstonenet.com.cn/api/url/detail?key={key}" \
 -H 'X-Auth-Token: <your API key>' \
 -H 'ACCEPT: application/json' \
 -H 'X-API-Version: 1.0.0' \
 -H 'X-API-Language: en'
```

- Java

```
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;

public class IpReputationReport {
   public static void main(String[] args) throws Exception{
      HttpClient httpClient;
      HttpGet getMethod;
      HttpResponse response;
      String reponseContent;
      httpClient = HttpClients.createDefault();
      getMethod = new HttpGet("https://ti.hillstonenet.com.cn/api/url/detail?key={key}");
      getMethod.addHeader("X-Auth-Token", "{your api key}");
      getMethod.addHeader("ACCEPT", "application/json");
      getMethod.addHeader("X-API-Version", "1.0.0");
      getMethod.addHeader("X-API-Language", "en");
      response = httpClient.execute(getMethod);
      HttpEntity httpEntity = response.getEntity();
      reponseContent = EntityUtils.toString(httpEntity);
      EntityUtils.consume(httpEntity);
      System.out.println(reponseContent);
   }
}
```

- Python

```
import urllib2
```

```
url = 'https://ti.hillstonenet.com.cn/api/url/detail?key={your key}'
headers = {'X-Auth-Token':'{your api key}',
        'ACCEPT':'application/json',
        'X-API-Version':'1.0.0',
        'X-API-Language':'en'}
data = None
req = urllib2.Request(url, data, headers)
response = urllib2.urlopen(req)
report = response.read()
print report
```

3.3.9.5 响应示例（JSON）

```
{
        "data": {
                "result": "malicious",
                "url": "http://anguiillanet.com/freeme",
                "hash_sha256":
"e543179563f7f0d3deb2ebe5ca2a2983b1c5fd540c9293fac984076dc311cb74",
                "related_domains": [
                        "anguiillanet.com"
                ]
        },
        "response_code": 0,
        "response_msg": "OK"
}
```

4 专项场景最佳情报实践

4.1 安全运营与自动化分析响应（XDR+SOAR）

4.1.1 挑战与痛点

目前安全运营遇到的挑战与痛点如下：

- **告警疲劳，响应效率低下** 互联网的威胁无处不在，在不同业务场景的攻击事件产生的威胁情报数据也会大相径庭。只有覆盖了足够多的主要业务场景，才能保证应用情报时威胁事件的检出率。完善的威胁情报数据运营来源，需要覆盖第一手的公有云防护场景，办公网防护场景，互联网业务防护场景，并基于互联网基础关联数据（如 PDNS、Whois 等）和开源情报线索的挖掘。
- **难以关联，缺失跨系统事件分析的关联性** 孤立的安全设备日志无法还原复杂的攻击过程各阶段，但通过威胁情报识别攻击者要素实体（如文件、域名、IP）并进行有效关联，可以实现跨系统攻击模式评估（如通用 ATT&CK 框架，恶意家族攻击团伙归属），实现分散事件间的逻辑链路、攻击阶段重新整合。
- **缺乏上下文，告警难以区分真实攻击与误报** 在告警运营中，加密流量或业务异常行为易引发误判。例如，Cobalt Strike 木马使用动态指纹变种进行 C2 通信，威胁情报可通过特征库 IOC 精准识别，但若仅依赖流量特征检测，可能会导致漏报或误报。
- **依赖人工经验，无法确定响应动作和形成自动化剧本（Playbook）** 威胁情报为自动化剧本提供标准化线索输入，增加条件判断维度，覆盖更多场景，并在业务场景、攻防对抗等多种威胁分析环节，提供丰富的上下文信息，以便触发更多有效响应机制，如：基于 C&C 识别自隔离受感染主机；基于可疑来源 IP，下发阻断清单至防火墙。补充的自动化的剧本，能有效提升企业运营效率和降低应急响应时间。

4.1.2 推荐威胁情报类型

1. 本地威胁情报引擎+云查服务；
2. 全量出入站网络威胁情报与文件信誉查询接口；
3. 情报 TTPs，包含 ATTCK 技术点，关联恶意家族团伙，关联进程信息等

4.1.3 三大目的

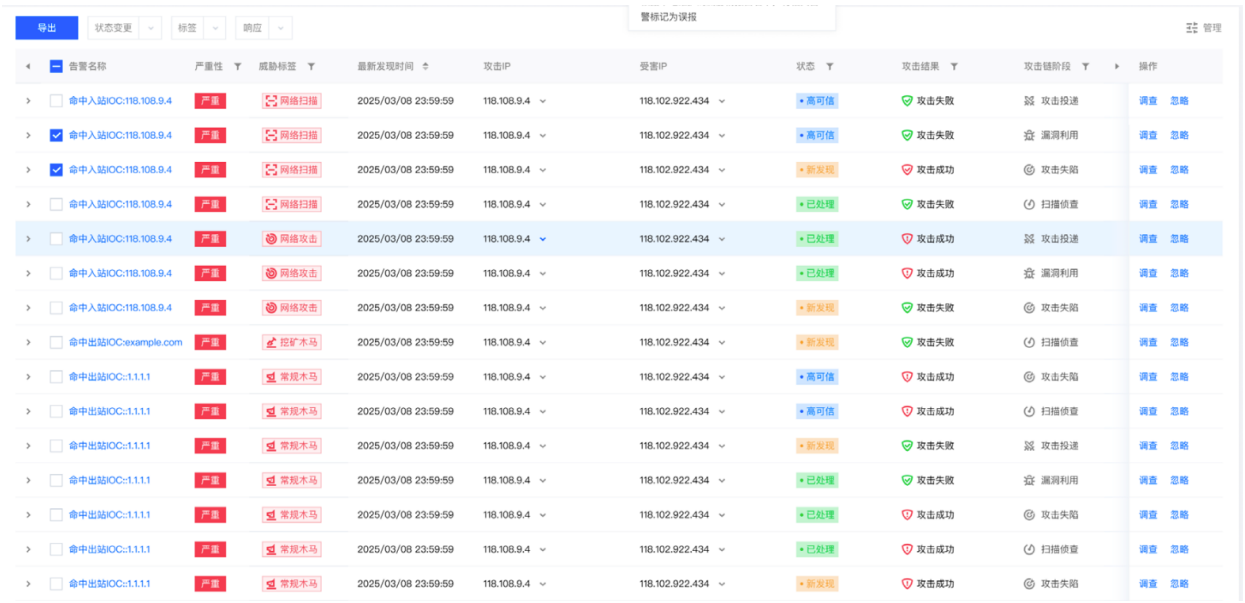
优化告警分诊的策略配，安全运营平台的首要挑战往往是告警过多的问题。对于部署在头部大型国有企业总部的安全运营平台来说，每日要处理的安全告警往往是数以千万计，即便通过关联分析引擎等模块，告警也只能减少至万级。通过威胁情报的降噪和关联威胁提炼，可有效提升安全运营人员的效率处理关键事件。对于其他非关键告警，也可以根据威胁情报的风险等级、置信度、威胁类型进行分类分级处理，实现告警运营的有序性。

丰富化告警的上下文，安全运营平台会对各类型安全设备的告警进行统一整理，但因为格式不同，检测规则存在局限性等原因，安全运营人员很难基于告警和安全事件做深度分析。但威胁情报自带上下文，可以有效补充告警中所需的决策依据。

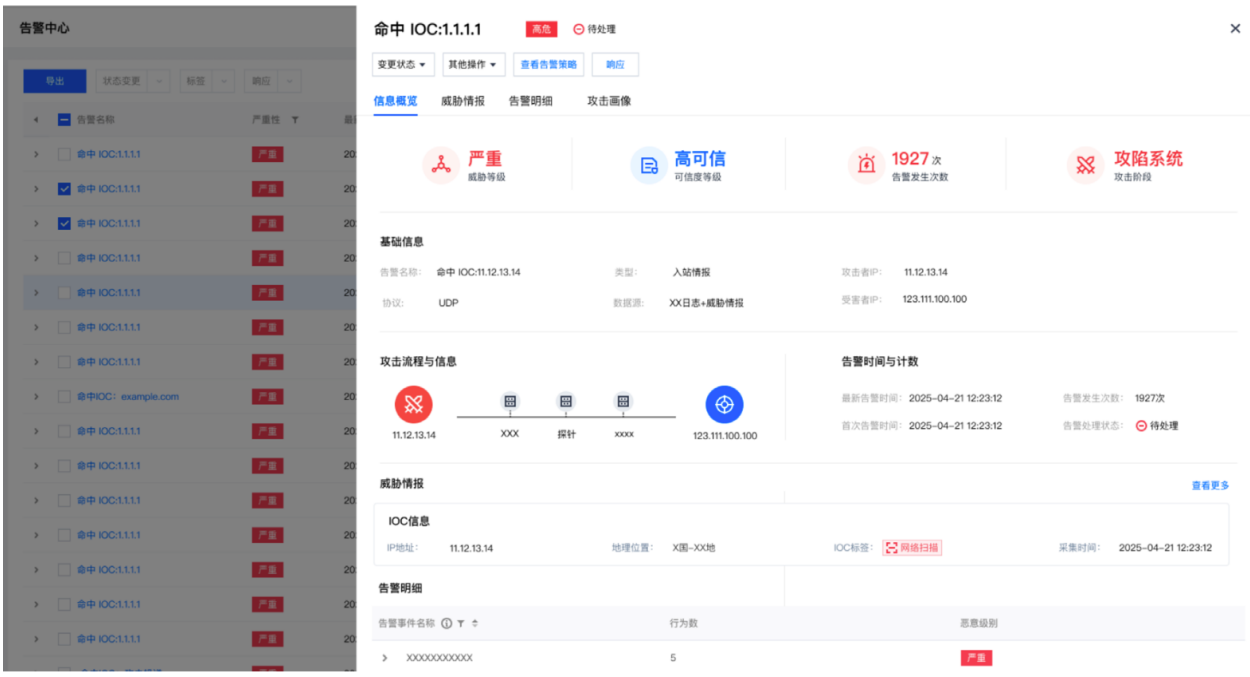
注入情报到知识库供 SOAR 的 Playbook 的调用，对于检测设备来说，更多是判断恶意等级的高低。但对于自动化分析响应流程来说，需要的是场景化的分析和灵活的配置。威胁情报不仅可作为检测规则的IOC模块，还提供地理位置、家族团伙信息、归属者性质（IDC、VPN、TOR）等关联上下文信息，以便作为自动化分析响应模块的知识库。

4.1.4 示例：XDR 出入站匹配与告警富化

XDR 区别于流量检测（NDR）设备直接针对网络行为请求进行匹配检测，安全运营平台更多是针对**收集到的各类安全设备的告警进行研判**，需要基于告警格式去建立映射关系。如：通过告警中出站目的地址匹配 IOC 情报，发现内网失陷主机；**日志中入站请求的来源地址匹配 IP 入站情报**，可以有效识别可疑来访源和相关风险标签。



在查看各类安全产品的告警时，仅依赖威胁情报本地引擎可能无法形成立体的分析评估，故建议在产生告警后，调用**字段更多的情报源进行二次富化**。解决 agent 情报不及时和信息不丰富的问题。因为 Agent 只拿情报检测，运营需要更多的上下文进行参考。如基于情报富化与调查：在告警列表中，**点击查看富化情报，可弹出整体情报的描述信息。至少包括命中情报值，风险等级，置信度，威胁类型（一级、二级标签），相关恶意家族和攻击团伙，相关告警次数，技战术和状态描述（如参考 ATT&CK 字段），其他基础信息（如 IP 可提供地理位置，域名可提供注册者），命中情报源信息（包括命中时间），告警相关网络信息（来源地址、访问地址、相关协议、事件时间等）。建议支持外联互联网云查情报以保障数据的时效性和上下文的丰富度**。



4.1.5 示例：XDR 的 SOAR 剧本

下面是 XDR 上基于 4.1.4 富化情报做“挖掘”和高层次检测的示例：如果情报有 TOR 和 VPN 标签，标记为潜在的威胁，降低等级：

对于入站类（扫描）的，可以根据 NDR 的频率检测，自动化降低一批告警等级：

4.2 流量威胁检测（NDR）

典型产品：网络威胁分析（NTA）、网络威胁检测与响应（NDR）、入侵检测系统（IDS）、入侵防御系统（IPS）、安全 DNS 产品等。

4.2.1 推荐情报类型

全量出入站的网络威胁情报

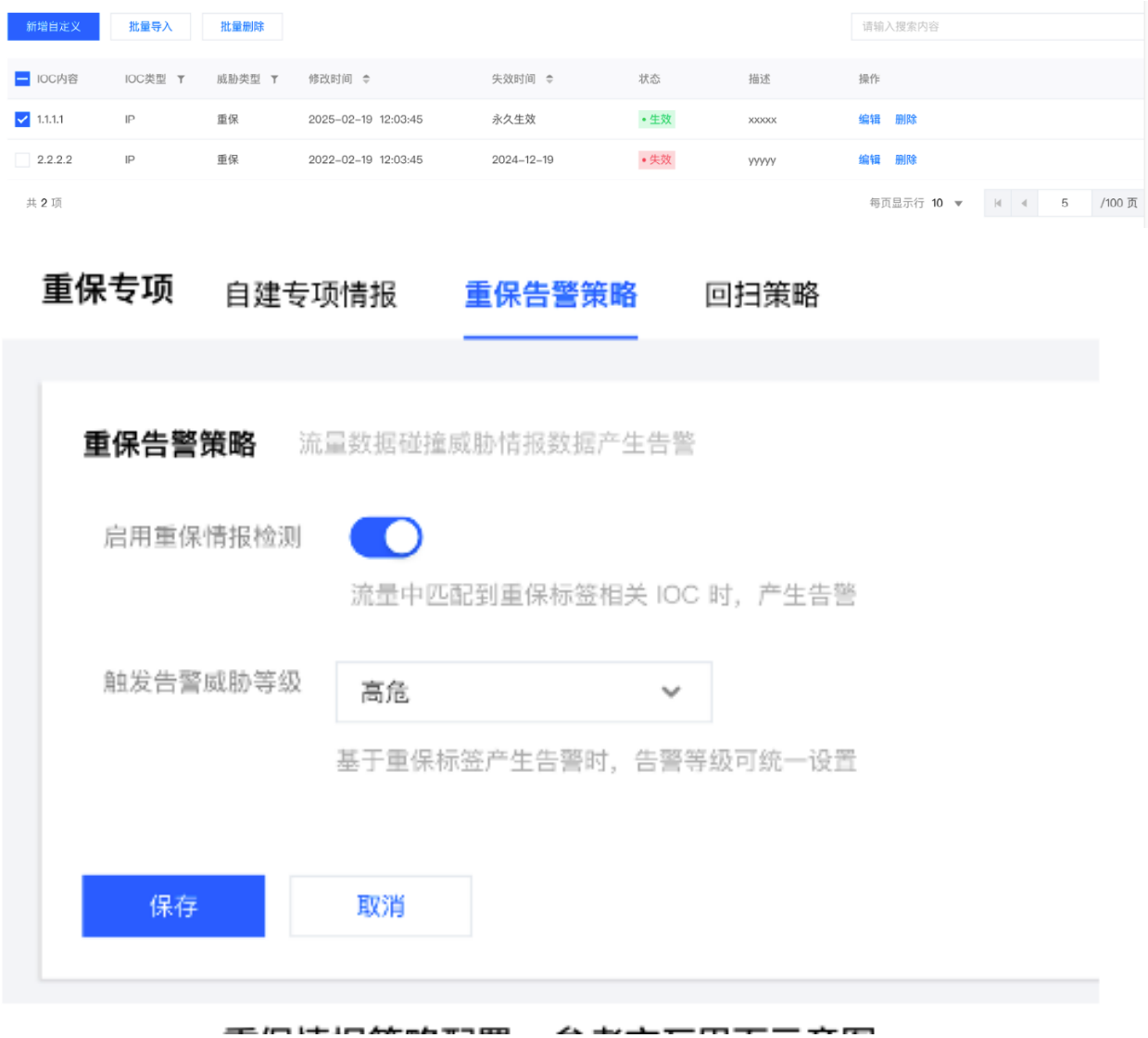
4.2.2 示例：轻量出入站检测

NDR 先不分方向，全部检测出来告警数据，然后在轻量分析时候，依据出入站情报，再匹配出入站方向的资产，判断失陷与攻击来源。并基于

4.2.3 示例：重保期间入站策略提高

区分日常运营，重保期间会存在大量的入站流量，根据入站情报，可提高拦截优先级，并依据捕捉到的态势，快速拦截同一区域类的其他资产。提供已导入重保情报的数据列表，支持增删改查。注意，导入的重保情报需要与现有情报格式匹配，如至少包括 IOC

值，威胁标签为“重保”，更新时间，当前状态，威胁等级等。导入时需要支持单条录入，推荐支持批量导入（如基于模板）以便于保障运营效率。



4.3 边界防护与阻断（WAF，FW）

4.3.1 挑战与痛点

在网关侧，难以发现高级持续性威胁 传统网关类产品，依赖静态规则检测已知威胁，但面对高级持续性威胁（APT），往往难以应对低频加密流量或合法协议混淆这样的复杂攻击行为。例如，针对低频出现的 C2 通信链接，边界防护设备很难基于规则去针对性识别造成漏报，但威胁情报因为具有广泛的威胁监测视野，确定性的IOC 可以直接作为高置信的黑名单支撑边界防护设备识别对应远控服务器的 IP 或域名。

缺乏识别矿池等类型威胁事件的手段 挖矿类软件的通信行为常通过加密流量伪装，传统网关难以通过端口或协议特征识别。但对于网关类产品，其实需要的只是高危来源、回连黑地址的识别即可。

无法建立确定性威胁的阻断能力 防火墙一般会通过下发黑名单的方式确定威胁阻断范围，但内部运营难以持续更新，且作为大流量吞吐的设备，更需要精准的判断进行阻断支撑。而高精准的 IOC 则可有效弥补网关类设备的这一短板。

性能有限无法承载大规模黑名单数据 传统网关硬件性能无法支撑大规模黑名单数据，且更新模式上会给性能带来更大挑战。这亟待一种原生性的威胁情报结合模式，在达到百万级数据加载的同时，支撑大吞吐量的访问业务控制。同时，威胁情报数据可基于威胁等级、置信度、威胁类型等维度进行分级分类整理，满足在总量较大情况下的分层下沉，以保障应用产品的数据量在性能可控范围内。

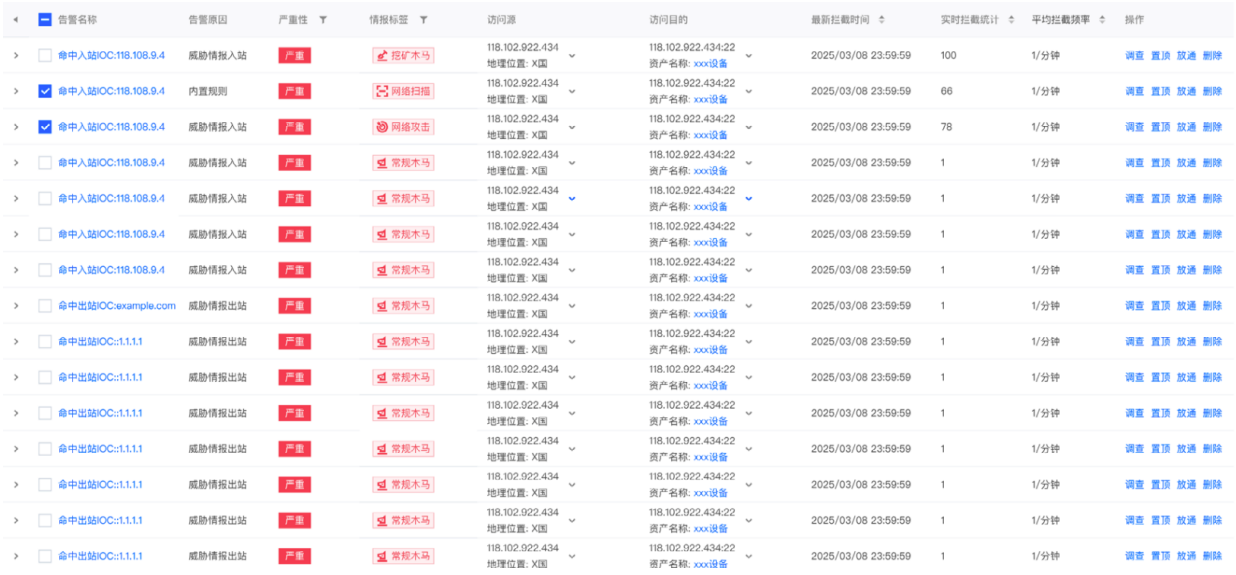
BOT 防护依赖规则容易出现漏报 对于 WAF 类设备，常规防御 BOT 攻击主要依赖行为分析、黑名单等简单规则进行识别，以保障业务性能的正常。但这种方式存在更新要求高，检测准确率较低等问题，缺少对攻击来源身份、威胁类型的鉴定能力。如果能基于威胁情报的威胁标签、置信度、风险等级、活跃时间等进行画像补全，可有效弥补识别 bot 的盲区。

4.3.2 推荐情报类型

高精静态出入站情报

4.3.3 示例：检测静态，信息动态 API

在检测时候使用出入站静态情报，在信息展示时候可以根据动态 API 获取丰富信息：

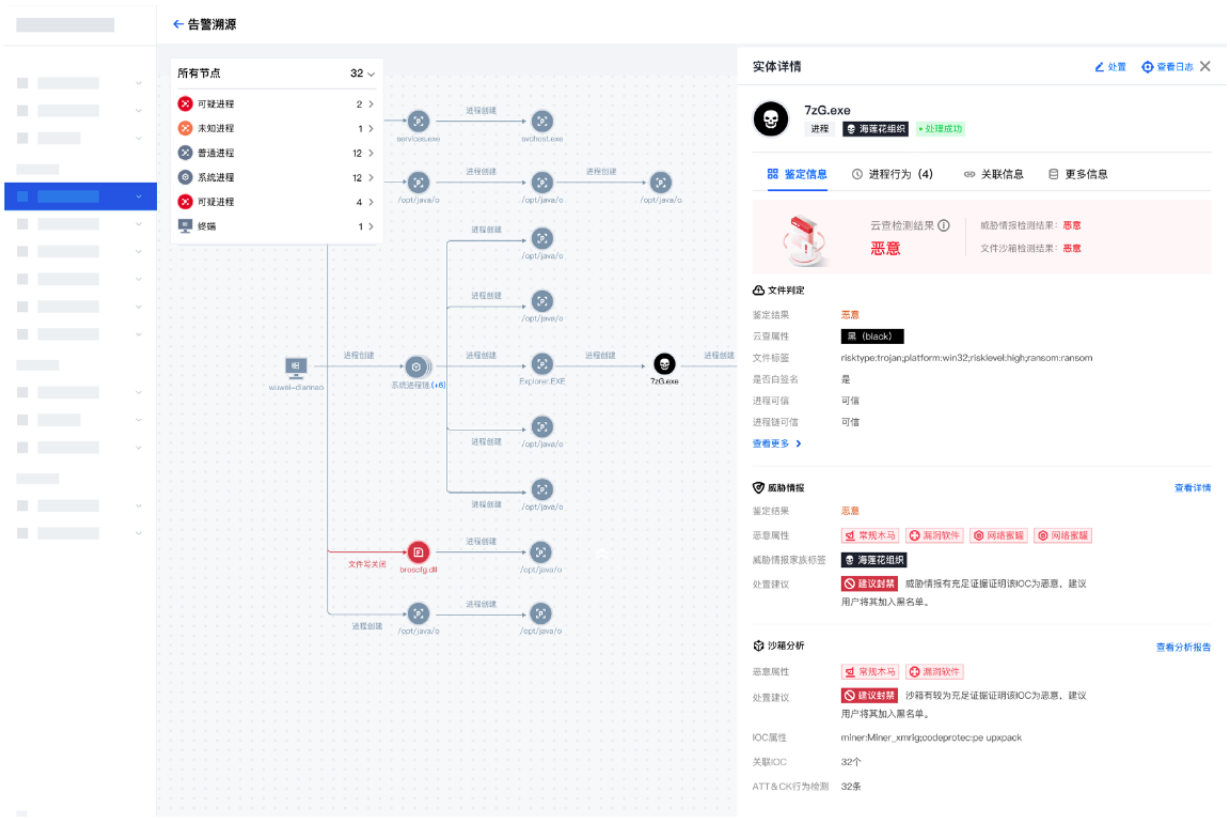


告警名称	告警原因	严重性	情报标签	访问源	访问目的	最新拦截时间	实时拦截统计	平均拦截频率	操作
命中入站IOC:118.108.9.4	威胁情报入站	严重	挖矿木马	118.102.922.434 地理位置: X国	118.102.922.434:22 资产名称: xxx设备	2025/03/08 23:59:59	100	1/分钟	调查 置顶 放通 删除
命中入站IOC:118.108.9.4	内置规则	严重	网络扫描	118.102.922.434 地理位置: X国	118.102.922.434:22 资产名称: xxx设备	2025/03/08 23:59:59	66	1/分钟	调查 置顶 放通 删除
命中入站IOC:118.108.9.4	威胁情报入站	严重	网络攻击	118.102.922.434 地理位置: X国	118.102.922.434:22 资产名称: xxx设备	2025/03/08 23:59:59	78	1/分钟	调查 置顶 放通 删除
命中入站IOC:118.108.9.4	威胁情报入站	严重	常规木马	118.102.922.434 地理位置: X国	118.102.922.434:22 资产名称: xxx设备	2025/03/08 23:59:59	1	1/分钟	调查 置顶 放通 删除
命中入站IOC:118.108.9.4	威胁情报入站	严重	常规木马	118.102.922.434 地理位置: X国	118.102.922.434:22 资产名称: xxx设备	2025/03/08 23:59:59	1	1/分钟	调查 置顶 放通 删除
命中入站IOC:118.108.9.4	威胁情报入站	严重	常规木马	118.102.922.434 地理位置: X国	118.102.922.434:22 资产名称: xxx设备	2025/03/08 23:59:59	1	1/分钟	调查 置顶 放通 删除
命中入站IOC:118.108.9.4	威胁情报入站	严重	常规木马	118.102.922.434 地理位置: X国	118.102.922.434:22 资产名称: xxx设备	2025/03/08 23:59:59	1	1/分钟	调查 置顶 放通 删除
命中出站IOC:example.com	威胁情报出站	严重	常规木马	118.102.922.434 地理位置: X国	118.102.922.434:22 资产名称: xxx设备	2025/03/08 23:59:59	1	1/分钟	调查 置顶 放通 删除
命中出站IOC::1.1.1.1	威胁情报出站	严重	常规木马	118.102.922.434 地理位置: X国	118.102.922.434:22 资产名称: xxx设备	2025/03/08 23:59:59	1	1/分钟	调查 置顶 放通 删除
命中出站IOC::1.1.1.1	威胁情报出站	严重	常规木马	118.102.922.434 地理位置: X国	118.102.922.434:22 资产名称: xxx设备	2025/03/08 23:59:59	1	1/分钟	调查 置顶 放通 删除
命中出站IOC::1.1.1.1	威胁情报出站	严重	常规木马	118.102.922.434 地理位置: X国	118.102.922.434:22 资产名称: xxx设备	2025/03/08 23:59:59	1	1/分钟	调查 置顶 放通 删除
命中出站IOC::1.1.1.1	威胁情报出站	严重	常规木马	118.102.922.434 地理位置: X国	118.102.922.434:22 资产名称: xxx设备	2025/03/08 23:59:59	1	1/分钟	调查 置顶 放通 删除
命中出站IOC::1.1.1.1	威胁情报出站	严重	常规木马	118.102.922.434 地理位置: X国	118.102.922.434:22 资产名称: xxx设备	2025/03/08 23:59:59	1	1/分钟	调查 置顶 放通 删除
命中出站IOC::1.1.1.1	威胁情报出站	严重	常规木马	118.102.922.434 地理位置: X国	118.102.922.434:22 资产名称: xxx设备	2025/03/08 23:59:59	1	1/分钟	调查 置顶 放通 删除
命中出站IOC::1.1.1.1	威胁情报出站	严重	常规木马	118.102.922.434 地理位置: X国	118.102.922.434:22 资产名称: xxx设备	2025/03/08 23:59:59	1	1/分钟	调查 置顶 放通 删除

4.4 主机与终端安全

4.4.1 示例：信息富化

在投递样本，文件执行，C&C 回连环节，威胁情报和反病毒引擎可对各类威胁实体（如文件 Hash、域名、IP 等）进行上下文富化，在此可提供详情展开页面图形化展示信息。文件类威胁实体可统一提供威胁情报、反病毒引擎、云沙箱等多个维度评估结果，具体样例效果如下所示。



5 基于情报多场景联动威胁狩猎示例

5.1 内网资产出站失陷检测与 EDR 精准隔离

目标是在攻击者完成 C2 回连、数据外传之前，将失陷主机从网络层与进程层同时“双杀”。

1. 情报输入：
   - TTP 级情报：APTxx 组织惯用 DNS-over-HTTPS(DoH) 隧道 + 特定 JA3/JA4 指纹。
   - IOC 级出站情报：30 个 DoH 域名、15 个回连 IP、4 个恶意进程签名哈希。

2. 产品联动：
   - NDR 镜像核心交换机流量 → 发现异常 DoH 隧道（JA4 匹配）。
   - NDR 将告警同步给 XDR，XDR 自动查询 EDR 的进程-网络关联树。
   - EDR 确认恶意进程 PID 后，下发“网络隔离+进程终止+注册表免疫”策略。

3. 运营流程（SOAR 剧本）：
   - T0：NDR 告警 → SOAR 开 Case；
   - T+30 秒：XDR 判定主机失陷 → SOAR 调用 EDR API 隔离；
   - T+2 分钟：SOAR 拉取威胁情报报告，自动追加至 Case；
   - T+10 分钟：值班人员复核，SOAR 关闭或升级。

度量指标：MTTD < 60 秒，MTTI（隔离）< 3 分钟，误报率 < 1%。

5.2 入站钓鱼域名前置拦截 + NDR 深度 SSL 解密

在员工点击恶意链接前完成域名封禁，并留存完整攻击链 PCAP。

1. 情报输入
   - 云端每日推送“今日新增钓鱼域名”列表。 WHOIS 创建时间 < 24h、DV 证书颁发者 = Let’s Encrypt、Alexa 排名 > 1M。

2. 产品联动
   - XDR → SOAR → 防火墙自动下发域名黑名单。
   - NDR 触发 SSL 解密 → 还原完整 HTTP 载荷。
   - XDR 将钓鱼域名关联至邮件网关日志，定位收件人邮箱与终端。

3. 运营流程
   - 情报命中 → 1 分钟内全网封禁；
   - NDR 提取钓鱼页面中植入的 JavaScript 重定向 → 提取新 IOC；
   - 反向推送至 XDR，实现“自反馈”情报循环。

度量指标：钓鱼域名存活时间 ≤ 30 分钟，NDR PCAP 留存率 100%。

5.3 挖矿木马横向移动狩猎（NDR + EDR + XDR 三联）

示例，识别由永恒之蓝、Kerberoasting 触发的横向挖矿行为。

1. 情报输入
   - 行为特征：SMB 445 大量 200K 字节写操作、Kerberos TGS-REQ 异常加密类型。
   - 矿池域名：*.minexmr.com*、.nanopool.org。
2. 产品联动
   - NDR 发现横向 SMB 爆破→XDR 关联同一 AD 账户登录事件。
   - EDR 在目标主机发现可疑 PowerShell 下载 cradle→触发 YARA 扫描。
   - XDR 将 IOC、TTP 自动写入威胁狩猎查询模板，供分析师每日滚动运行。
3. 运营流程
   - NDR 告警→XDR 生成“可疑横向移动”时间线；
   - 分析师一键下发“EDR 隔离 + NDR 流量镜像到沙箱”；
   - 挖矿木马样本回传后，自动提取新 C2→更新情报库。

度量指标：横向移动检测窗口 ≤ 5 分钟，狩猎查询复用率提升 40%。

5.4 勒索软件双重勒索外泄通道阻断（XDR 数据流视角）

目标：在文件外泄阶段而非加密阶段止损。

1. 情报输入
   - 已知勒索软件外泄域名 mega.nz 特定子域、User-Agent = “ransom-leak-v2”。
   - JA3 指纹对应 Go-http-client/2.0 固定版本。
2. 产品联动
   - XDR 融合 EDR 的文件访问日志 + NDR 的上传流量→发现异常 10GB 外泄。
   - NDR 实时阻断 TLS 连接并注入 TCP-RST；EDR 同时阻止 winhttp.dll 调用。
   - SOAR 工单自动通知法务、公关，启动数据泄露应急预案。
3. 运营流程
   - 情报命中外泄域名→30 秒内阻断；
   - XDR 输出“外泄文件列表 + 受影响用户”CSV；
   - 每 15 分钟滚动对比情报库，防止攻击者切换域名。

度量指标：外泄中断率 ≥ 95%，文件级溯源精度 100%。

5.5 失陷资产判断（EDR + NDR + XDR 三联）

示例，识别由恶意软件感染触发的失陷资产行为。

1. 情报输入
   - 行为特征：主机短时间内向多个陌生 IP 发起连接请求，且连接端口为已知恶意软件常用端口（如 4444、5555 等）
   - 进程特征：出现未知的可疑进程，且该进程频繁读写特定文件（如随机命名的临时文件）
   - 域名特征：主机访问的域名具有高熵特性（如长度 > 30 位、含特殊字符），或已知黑域名
2. 产品联动
   - NDR 发现异常连接行为 → XDR 关联同一主机的网络连接事件。
   - EDR 在目标主机发现可疑进程 → 触发深度扫描。
   - XDR 将 IOC、TTP 自动写入威胁狩猎查询模板，供分析师每日滚动运行。

3. 运营流程
   - NDR 告警 → XDR 生成“可疑失陷资产”时间线；
   - 分析师一键下发“EDR 隔离 + NDR 流量镜像到沙箱”；
   - 恶意软件样本回传后，自动提取新 IOC → 更新情报库。

6 参考信息

7 附录

7.1 1.0.0 版本服务响应码对照

返回状态码 (Response Code)	描述 (Description)
1	无报告
0	正常
-1	无权限
-2	查询超限
-3	查询频率过高
-4	查询值异常
-5	请求无效

7.2 威胁标签取值

#	标签名称（中文）	标签名称（英文）
1	CnC	CnC
2	暴力破解器	BruteForcer
3	扫描器	Scanner
4	僵尸网络	Botnet
5	APT 攻击	APT
6	撞库攻击	CredentialStuffing
7	挖矿	Miner
8	外泄数据站点	DropSite
9	木马	Trojan
10	后门	Backdoor
11	蠕虫	Worm
12	恶意软件	Malware
13	勒索软件	Ransomware
14	广告软件	Adware
15	伪杀毒软件	FakeAV
16	灰色软件	GrayWare

17	风险软件	RiskWare
18	黑客工具	HackTool
19	网络钓鱼	Phishing
20	发送垃圾邮件	Spam
21	拒绝服务攻击	Dos
22	漏洞利用	Exploit
23	引擎测试程序	TestFile
24	初始访问	Initial-Access
25	执行	Execution
26	持久化	Persistence
27	权限提升	Privilege-Escalation
28	防御绕过	Defense-Evasion
29	横向移动	Lateral-Movement
30	命令与控制	Command-and-Control
31	Web 攻击	WebAttacker
32	Keylogger	Keylogger
33	访问控制	AccessControl
34	缓冲区溢出	BufferOverfloe
35	中间人攻击	MITM
36	DNS 攻击	DNS-ATTACK
37	web 应用漏洞攻击	Web-Vulnerability-Attack
38	身份验证攻击	Authentication-Attack
39	异常流量	Anomalous-Traffic
40	协议异常	Protocol-Anomaly
41	网络攻击	CyberAttack