安全小课堂第128期【甲方威胁情报生存指南】

京东安全应急响应中心 1月28日

威胁情报最近概念很火，但是很多安全建设者在建设环节对威胁情报往往是泛泛的一笔带过，归根结底是威胁情报在运营和给企业带来收益的结果不是很明显，而企业内的一些数据往往是能够挖掘真实情报case的“金矿”

JSRC **安全小课堂第128期**，邀请到**elknot**作为讲师就**威胁情报相关的技术**为大家进行分享。同时感谢小伙伴们精彩讨论。

甲方安全里面威胁情报的定位和实际收益点?

京安小妹

elknot:

说之前先来说一下甲方对于威胁情报的定义，广义上的威胁情报实际上就是能为发现针对企业安全产生的威胁提供知识和处置方案的一种信息，所以威胁情报要拆开成威胁和情报，所谓威胁就是企业内部可能会发生的能够影响业务正常运转或者直接造成业务损失（可以是经济损失也可以是系统可用性损失），而情报则是能够辅助提前发现这些威胁知识和数据。

在甲方往往大家不太愿意谈论威胁情报的原因主要是因为威胁情报这个东西现在很多人还没有玩转，或者是直接去对接商用或者公开的数据到安全设备上，但是这种情况往往会出现很严重问题，比如低精度和高误报率，所以这个时候我们的情报实际上是被威胁情报给“威胁”了，变成了货真价实的情报威胁。所以甲方在建设威胁情报的角度上能够考察的点包括以下几点：

（1）情报的准确度：

情报准确度即情报数据能否去客观的发现该场景下正在发生的威胁，也就是精度和误报率的综合体，往往需要累计一段时间的运营数据才能够去描绘这一指标，这一指标直接体现出的就是情报源的质量和丰富度。

（2）所覆盖威胁的场景丰富度：

场景覆盖度指的是我能利用这些数据发现多少个不同场景下这在发生的威胁，这一指标可以指明情报发现的覆盖度（这里可能会和入侵检测的场景有些许重叠）。

（3）情报生命周期处理的自动化程度

情报生命周期的自动化程度则是工具、平台在情报处理方面的接入率，因为在企业安全中，威胁情报的生命周期为：采集->数据预处理->精细化场景处理->情报输送->持续化运营，在这五个环节中，每个环节如果都可以自动化完成了，就说明情报自动化运营的能力建设的非常好了，这一指标可以从侧面描述威胁情报在安全运营中的落地情况。

（4）面向威胁的止损时间SLA

止损时间SLA指的是在威胁情报介入前后，SLA能够缩减到什么程度，这一方面则可以很好的刻画情报驱动的应急响应是否能够产生实际的作用，ROI是否足够的高。

再来说收益点的问题，实际上威胁情报的收益点在于能够大幅度缩短止损时间和造成的业务损失（包括系统中断导致的损失如漏洞攻击，和在经济上的损失比如薅羊毛），许多威胁情报的PM在制定情报计划的时候可能会从这两个实际收益点以及以上的指标入手来制定今年的OKR。

威胁情报收集方法和挖掘思路?

京安小妹

elknot:

首先先来说威胁情报的收集方法，威胁情报收集的方法实际上还是依赖网络爬虫和公开的API进行收集，但是在收集之前一定要先做好情报计划，以保证自己收集过来的情报是满足之前设定的OKR的。

通过爬虫收集实际上分为两种场景，第一种是公开信息收集，即可以通过类似于selenium+scrapy这种组合对特定页面的数据进行爬取落库操作，当然这里其实有一种更简单的方法，Google开放服务里面提供一种服务叫做Google CSE(Custom SearchEngine)，这个东西可以利用Google的爬虫对特定的目标完成定制化的查找和排序，并且可以通过自带的API将搜索结果保存成JSON格式，以便和其他的平台进行衔接。用户只需要对CSE里面的规则进行编写即可完成这一部分的收集；除了互联网之外，一些业务相关的安全信息更多的可能会出现在暗网，暗网的话可以使用OnionProxy+OnionScan进行收集，注意点和之前的一样，一定要确定好这一部分数据是要符合之前制定的OKR的，不然收集回来的情报会浪费储存空间；鉴于黑产交易的渠道多为QQ群和微信群以及Telegram，对于这一部分需要去使用聊天机器人对特定的几个黑产群的聊天记录进行爬取，github上有很多公开的机器人code，这里就不举例了，在这里需要提醒一下大家，注意法律法规问题；另外的则是基于RSS、订阅这种的及时性很强的消息，这种的话由于格式很规范，解析后直接落库即可，不需要花费太多的精力，推荐使用rssparser这种库去解析RSS的订阅源，有时候会收集到一些有用的东西。

当然只是收集的话是远远不够的，更重要的是对于这些爬取回来的信息的处理，威胁情报数据处理分为两部分，粗处理和场景化精加工，所谓粗处理实际上是对爬取回来的信息进行格式化，比如说统一格式化成JSON格式或者是XML这种可以自动化处理的数据结构；第二部分是精加工，这一部分精加工主要是对情报进行更深层次的文本分析，可以利用诸如NLP技术、词频和词性分析、上下文理解的方法对情报信息进行提取，提取

出来的关键字应包含该情报所影响的作用点、作用范围、是否受到影响、影响程度、是否需要推动应急响应流程，我举个简单的例子，比如说我们爬取了Struts的安全公告，里面现在爆出来了一个CVE，这个时候我们需要对CVE的影响程度、影响的Struts的版本、是否受到影响（与CMDB或者资产大盘对接）、是否需要推动应急响应（和安全运营平台对接）、影响是否严重（CVSS评分、POC是否存在）等来描述我们爬到的漏洞威胁情报，当然这一部分数据是需要格式化成特定格式的，方便与系统的对接，这里推荐使用python的NLPIR、NLTK、Scikit-learn、Gensim这种NLP的库去处理，方便且高效。

接下来说的是挖掘问题，我们挖掘情报不一定需要从外部的数据去入手，相反实际上内部的数据往往是挖掘威胁情报的“金矿”。我们在对威胁情报挖掘之前需要理清楚两件事情，这两件事情是什么是正常的业务流程，业务会遭受何种类型的威胁，理清楚这一部分事情往往就完成了一半，接下来我们需要针对我们设计好的场景去进行日志和流量的梳理，日志分析的主要是业务系统日志和访问日志，流量则是进行通信的流量DPI数据。我们举个例子，假设有黑产利用Struts的漏洞去进行挖矿软件的种植，在这一环节中，我们可以通过分析web服务器的日志以及accesslog的日志判断在什么情况下会出现这种日志，进而通过Storm或者Flink的日志流处理对这一场景进行监控一旦发现后立刻报警，迅速进入应急响应流程。

通过持续化运营一段时间后，场景覆盖率提升到一定程度了，这个时候就完成了对于该种威胁的情报自动化运营，如果你要是不满足于这种现状的话，我们需要进一步挖掘潜在的没有发现的攻击，这个时候之前梳理出来的正常的业务逻辑就派上了用场，我们排除掉正常的业务逻辑中的日志和已经可以自动化发现的场景之后，剩下的日志就值得挖掘一下了，这种时候没准会挖掘出一些0day或者是在开发阶段的恶意脚本。

如果你的野心更大些，你可以在公网上部署蜜罐系统，去对蜜罐的日志进行挖掘，当然这里主要是涉及到一些样本层面上的东西，如果运气好的话，也能够挖掘出来一些有用的新样本或者是c2，取决于你有多少蜜罐的日志和蜜罐节点的分布。

讲师

提哪种类型的威胁情报SRC会更愉快的接纳?

京安小妹

elknot:

一般情况下SRC在接受威胁情报的时候会优先考虑已经对业务造成实际经济损失的情报，并且如果内部安全系统没有发现这种威胁的时候，奖励可能会double。第二种就是敏感信息和商业数据被恶意爬取的情况，这种相当于是触及了泄密，这种的SRC也会更愉快的接纳。第三种就是有一些黑产工具在逆向的过程中会发现一些厂商的url被挂在了黑产工具里面，这个时候如果造成了经济损失，SRC也会接纳。

所以总结一下就是：SRC更愿意接纳他们没有发现但是已经造成了实际损失（可能是经济损失也可能是系统可用性损失）的威胁情报。

讲师

如何从海量数据里面挖掘有价值的威胁情报？

京安小妹

elknot:

对于这个问题，首先我们需要明确的是何为有价值的威胁情报

有价值的威胁情报前提是ROI不能太低，要控制在一个可以接受的范围之内，其次要能够做到实时处理、实时传递和实时的运营，当然如果这部分没有条件的话可以人肉，但是SLA必须要在可接受范围内。

接下来我们来谈谈挖掘的事情，这部分挖掘一部分取决于数据，另一部分取决于算法，由于前面已经讲了太多抽象的东西，所以接下来我会举两个case来说明一下如何对这些数据进行情报挖掘

（场景1）：假设我们的业务系统很多都是web系统，所以accesslog是一个比较有价值的入手点，首先我们能确定accesslog可以记录的东西，一般的accesslog会记录访问

的url、请求方法、参数、返回字节、状态码、user-agent等信息，正常的访问情况下，这些数据大概率是会随机的。我们现在假设一个场景：黑产分子从我们的业务系统中准备批量爬取我们的UGC数据（用户的原创数据，比如商品评论）然后对用户进行画像，从而使用电话诈骗等手段进行不法谋利，但是由于是爬虫行为，所以我们完全可以从accesslog中访问进行分析，从而得到那一批IP或者哪个IP在频繁的对单个含有UGC内容的页面，这个时候我们根据返回字节数据可以判断是否为爬虫行为由于accesslog的量是非常巨大的，所以我们需要考虑对应的分析平台的基础架构，因为爬虫的止损时间直接关系到敏感数据被爬取的量，所以我们需要尽可能的实时，这个时候我们就会将日志接入spark streaming平台或者storm、flink这种实时平台进行实时的accesslog分析，这样好处是速度快且可以实时，我们通过判断一段时间内异常url在全部url中的占比作为阈值来判断是否被爬取，比如5分钟之内这个url的accesslog日志数量已经占到了70%，而且还在持续增加，user-agent是一样的，ip全是代理或者固定段，这个时候就可以告警，直接推动安全运营去应急止损。

（场景2）：假设由于用户和客户端的消息验证渠道有一部分是短信验证码，我们假设这个场景，如果一个或几个电话号码在频繁进行退换货和购买的情况，我们则会判定这些或这个手机号可能是羊毛党的作案资产，通过对该业务对应的业务日志和accesslog分析后发现，这些号码确实是羊毛党在作案，起因是因为这个业务本身存在逻辑漏洞，可以通过使用优惠券购买然后退货按照原价退的思路把优惠券部分进行套现，针对这个场景我们可以将这种行为编写成实时计算的作业对接到storm、flink这些平台，然后监测到对应的行为后立即推送到业务进行止损，减少因这种情况造成的损失。

以上两个例子其实都是基于真实场景去构造对应的情报分析模型来发现威胁的case，在实际的情报运营当中，对待威胁的运营方式都是case by case的运营，case积累的多了，能够发现的威胁也多，进而能够得到相对的安全。所以在挖掘的过程中需要对攻击手段和正常的业务进行理解，区分正常和异常的表现，进一步进行威胁建模，推动模型的落地与运营，实现情报运营的自动化和标准化。

讲师

互动问答环节：

1.威胁有哪些类？有哪些威胁是很多企业都有的呢？

讲师:

其实说句公道话，不同类型的企业威胁往往是不同的。但是其中又有一些共同点，首先成规模的企业必定有大量的基础设施，这些基础设施会存在漏洞、版本过旧、安全策略不当等问题，这种问题往往就是企业中的威胁，这种威胁称之为基础设施威胁。另一种则是业务层面的，比如说银行、电商企业面临的羊毛党、自身电商系统的稳定性、系统逻辑有问题被人实锤了恶意利用这种，这种统称为业务层面上的威胁。

2.老师挖过哪些让你记忆犹新的威胁情报呢？

讲师:

记忆犹新的情报，之前还在前司的时候我自己部署了一套蜜罐系统，当时主要的目的是收集一些样本并且提取对应的yara规则做防御，但是直到有一天我发现了一个样本绕过了杀软的检测，同时用IDA打开之后发现里面内置了很多函数，但是这些函数都是直接return了，当时觉得很奇怪，后来通过使用virustotal的数据和把c&c打下来之后看了一下，这个样本是Gafgyt家族的一个最新变种，只是还在待开发的阶段，后来在一个月的时间内这个样本连续迭代了8次，每次都只增加1-2个功能，然后再根据一些测绘的情报和信息，我挖到了一个专门做botnet的一个工作室，地点在北京回龙观那一块。

本期JSRC 安全小课堂到此结束。更多内容请期待下期安全小课堂。如果还有你希望出现在安全小课堂内容暂时未出现，也欢迎留言告诉我们。
安全小课堂的往期内容开通了自助查询，点击菜单栏进入“安全小课堂”即可浏览。

简历请发送：cv-security@jd.com

微信公众号：jsrc_team

新浪官方微博：京东安全应急响应中心