APT攻击检测与反制技术体系的研究

陈瑞东[1]，张小松[1*]，牛伟纳[2]，蓝皓月[1]

(1. 电子科技大学网络空间安全研究中心 成都 611731；2. 四川大学网络空间安全学院 成都 610044)

【摘要】高级持续威胁(APT)是近年兴起的新型网络攻击，一直受到网络安全界的重视。该文通过研究近十年150余项典型APT案例，形成针对APT攻击的分析模型，提出了当前APT攻击检测与反制亟需解决的4项问题，即：渗透防护脆弱、检测精度低、攻击范围取证困难、未知新型攻击响应慢。同时，该文对近年来典型性APT攻击事件进行取样分析，以攻击组织使用的工具集为基础，对攻击工具集进行关联挖掘。实验得出，同一组织使用的工具集间存在相似性规律。综上所述，该文研究的APT整体防御方案包括了4类防御方案的最新成果分析及归纳，对于构建统一的攻击检测与溯源反制平台起到支撑作用。

关 键 词 APT攻击； 攻击检测； 攻击技术分类； 防御方案

中图分类号 TP393 **文献标志码** A doi:10.3969/j.issn.1001-0548.2019.06.011

A Research on Architecture of APT Attack Detection and Countering Technology

CHEN Rui-dong[1], ZHANG Xiao-song[1*], NIU Wei-na[2], and LAN Hao-yue[1]

(1. Center for Cyber Security, University of Electronic Science and Technology of China Chengdu 611731;
2. College of Cyber Security, Sichuan University Chengdu 610044)

Abstract Advanced persistent threat (APT) is a new kind of cyber-attack as a growth security events. This paper analysis more than 150 typical APT cases happened during last decade, and constructs the analytical model of APT attack, indicates 4 major problems of APT attack detection and countering: the fragile penetration protection problem, the low detection accuracy, the difficulty of determining the attack forensic, and the slow response to the unknown attack problem. In the meanwhile, this paper analyzes typical APT attacks in recent years, mines the association based on attacking tools. According to the experiments, there are similarity patterns between the tools used by the same organization. In summary, the integral APT defense scheme in this paper includes the latest achievements of four types of defense schemes, plays an academic supporting role in building a unified attack detection and traceability countermeasure platform.

Key words APT attacks; attack detection; attacking technology classification; defense scheme

自网络空间成为第五战场以来，APT逐渐浮出水面。与早期个人兴趣推动的网络攻击不同，APT攻击的背后通常存在组织提供资助，具有复杂的背景和目的[1-3]。APT发起者具有很强的专业性，使用的攻击技术手法成熟、攻击方案较为先进。相对而言，目前对抗APT攻击的防御方案存在多方面的薄弱点，并缺乏完整的解决方案。

对APT攻击方来说，APT行为通常有相对固定的生命周期[4-5]。该生命周期被文献[6]称为网络攻击杀伤链，并分为6个阶段：侦察跟踪阶段、武器构建阶段、载荷投递阶段、漏洞利用阶段、安装植入阶段、命令与控制阶段。在不同的阶段，APT攻击的目的、危害性和被检测防御的难度都不同[7-10]。在侦察跟踪阶段和武器构建阶段，攻击者的目的主要是调查攻击目标和确定攻击手段，此时攻击者行事隐秘，往往不容易暴露且不留下证据，因此在此阶段几乎没有有效的防御手段，且防御的效率较低，意义也不大。在载荷投递阶段和漏洞利用阶段，攻击者的目标是入侵目的系统，此时APT攻击暴露在防御系统面前，对抗者可以初步对APT攻击进行检测和防御。此阶段APT攻击尚未形成实质性的破坏，因此在此阶段进行防御是较为理想的选择。到了命令与控制阶段，攻击者的主要目标在于横向和纵向移动以提升权限，最终达到破坏或窃取的目的。此

收稿日期：2019－02－26；修回日期：2019－03－16
基金项目：国家自然科学基金面上项目(61572115)；国家重点研发计划网络空间安全重点专项(2016QY04W800)
作者简介：陈瑞东(1985－)，男，博士生，主要从事信息安全方面的研究.
通信作者：张小松，教授，E-mail: johnsonzxs@uestc.edu.cn

时虽然检测难度较低，但由于进行攻击和防御所付出的代价较大，攻击已对目标系统造成了实质性的伤害。因此，现行的防御方案主要集中在载荷投递阶段、漏洞利用阶段和命令与控制阶段。

本文通过研究近10年来150多个APT攻击案例，提出了APT防御亟需解决的4项难题，即：渗透防护脆弱；检测精度低；攻击取证困难；对新型攻击响应慢。并依次阐述解决这4个难题的3项关键技术，即：系统防渗透技术；动态变形攻击模型与检测技术；攻击溯源与反制技术。并形成包括了恶意代码检测类防御技术、主机应用保护类防御技术、网络入侵检测类防御技术、大数据分析检测类技术的APT整体防御方案。

1　国内外研究趋势

1.1　APT攻击检测研究趋势

APT的检测和防御涵盖了APT攻击生命周期的各个阶段，但主要集中在中后期，所采用的检测和防御技术主要分为虚拟执行分析检测和异常检测。虚拟执行通过在虚拟机上执行样本，基于其程序控制流等样本信息判断攻击是否发生。

当前的APT攻击为了抵抗此种检测技术，常采用代码混淆的方法躲避检测。在代码混淆领域，普遍采用Collberg的分类方法，将代码混淆分为布局混淆、控制混淆、数据混淆以及预防混淆4种。布局混淆主要是减少代码中可供攻击者阅读和理解的信息，对程序的执行不会带来任何影响。控制混淆的主要目的是隐藏程序真正的控制流，数据混淆的对象是程序中的数据域。控制混淆和数据混淆都是通过提高程序的复杂度来增加混淆强度，它们的混淆强度及耐受性相对较好，但会增加大量程序代码，并引入额外的执行开销。预防混淆是针对特定的反编译器和反混淆工具设计的，适用范围较窄。

代码混淆不仅使虚拟执行的时间和空间开销倍数增大，而且通过隐藏控制流降低了虚拟执行检测的准确率。为对抗代码混淆、降低代码执行的时间复杂度，选取准确率较高的控制流特征来减少代码混淆带来的检测难度[11]。

APT异常检测通过检测网络中的异常流量判断APT攻击的回传行为。通常情况下，APT进行回传时，需要传输海量的数据，因此网络流量突然增大是一种明显的异常流量。但当前的APT攻击为了隐藏自己的流量，通常采用P2P(peer-to-peer)的自组织网络、匿名名网、加密网及隧道技术，实现自适应网络拥塞过程，在网络拥塞的情况下减小自己传输流量的速度，因此异常流量检测难以发现网络环境的变化。为更精确地检测到异常流量，研究人员需要设立多维度的流量特征，对网络中的流量进行全面监控。

同时，APT攻击离不开对漏洞的利用，对漏洞进行分析和检测能帮助APT攻击技术分类。漏洞分析检测的主要技术有基于源代码、基于补丁对比、基于模糊测试、基于代码特征等方法[12-18]。基于源代码的漏洞分析检测技术要求分析者拥有软件的源代码，对程序进行数据流和控制流分析或符号执行，检测漏洞是否存在。基于补丁对比的方法是通过对比打补丁前后程序的差异，找出所修复漏洞所在的位置。基于模糊测试的漏洞挖掘是一种动态的软件测试技术，该技术向程序中输入随机生成的数据并监视其运行，若程序出现异常，则判断程序中存在漏洞。基于代码特征的方法则是预先在已知漏洞中提取漏洞模型，并和目标软件进行匹配而发现漏洞的方法。该方法与机器学习结合得到了较多的应用。

在发掘方面，符号执行近年来也成果显著。文献[19]首次设计了一个支持安卓框架的符号执行系统，标志着符号执行在安卓系统中的起步。文献[20]将符号执行和数据流分析结合，用于快速查找补丁相关的代码段，并自动检测补丁是否带来新的漏洞。实验表明该方法能检测到内存相关的典型漏洞，提高了补丁的安全性。

文献[21]提出一种基于覆盖的灰盒测试方法，通过改变输入来改变测试生成的路径。当输入足够多时，可以用马尔可夫链来解释输入运动轨迹的概率。实验表明，该方法比AFL(american fuzzy lop)模糊器发现公共漏洞和暴露(common vulnerabilities & exposures, CVE)的效率高了很多，并在漏洞检测方面比符号执行工具Klee更有效。

1.2　溯源反制技术研究趋势

APT溯源反制技术主要包括分类技术和溯源技术。分类技术指根据恶意代码的特征将代码进行同源性分类，进而将APT攻击进行分类。在新型的恶意代码样本中，很多都是已有恶意代码的变种。这是恶意代码的创造者为躲避恶意代码检测技术，对旧的恶意代码进行变形加壳等扰乱技术[22-23]。因此在对代码进行同源性分类后，将新型的恶意代码归入其中一类，可提高新型恶意代码的分析效率。研究人员通过提取代码特征并建模对恶意代码进行分类，大致可分为基于序列的方法和基于图的方法，

需要达到的目标为在兼容一定代码变化的情况下，在海量代码库中快速定位同源代码。该技术一方面可以找出相似攻击行为的恶意代码，帮助研究人员分析新型代码，另一方面可以为APT攻击的变化过程提供检测依据。

APT溯源技术指的是通过攻击行为的特征追踪到APT行为主体的攻击主机、攻击控制主机、攻击者、攻击组织机构等信息。这些被利用的特征包括邮件、水坑、渗透等攻击习惯，所利用的漏洞、平台、持久化方法、检测对抗等技术特点，控制与通信的特点以及攻击者对区域或国家、行业或领域的目标偏好等。根据不同的特征，APT溯源可以从两个方面入手，一方面是从网络侧溯源，另一方面是从样本侧溯源。其通常利用的溯源特征如图1所示。

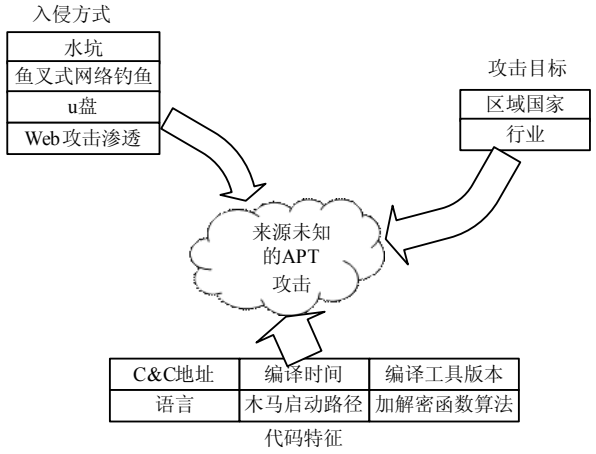

图1 几种常用的溯源特征示意图

在APT溯源技术兴起的同时，攻击者的行为也逐渐变得审慎，通过伪造虚假的网络痕迹来隐藏自己的行踪。如防御者通过代码修改和链接的时间情况判断攻击者所处的时区，相应地，攻击者通过伪造时间信息隐藏自己。

在国内，溯源技术已存在一定的应用，如360团队监控到了利用“双杀”零日漏洞发起的一例APT攻击，通过攻击流程分析、漏洞分析、powershell荷载分析、用户账户控制(user account control, UAC)绕过载荷分析以及归属关联分析确认了该攻击与APT-C-06组织的关联性[24]。

2 APT攻击检测与溯源反制技术路线

2.1 系统防渗透技术

渗透指通过对特定系统进行攻击形成对网络层、应用层、传输层或物理层等层次的控制能力，获得管理员等控制权限的过程。一方面，渗透是一个攻击行为，通过渗透可以入侵受害者的系统。另一方面，对我方系统进行渗透测试，全面检查系统的脆弱性，预防性地弥补系统的薄弱之处，能使系统保持一定程度的安全性。

美国是一个很早就研究发展渗透测试技术的国家。根据文献[25]的描述，美国在2005～2006年就已经启动类似项目并于2009年11月在纳坦兹铀浓缩工厂成功投放该数字弹头。目前主流的渗透技术主要集中在应用系统渗透测试技术、传输协议数据包的注入攻击技术、网络层设施的固件攻击技术、基于可替换物理设备的渗透测试技术、具有反取证能力的远程侦控技术、基于身份匿名化的海量数据续传技术以及载荷装配技术等7个方面。

文献[26]通过向Linux内核层中注入恶意代码来分析安卓操作系统中有关Wi-Fi的新型漏洞。文献[27]提出物联网所带来的新型网络威胁和渗透测试技术。文献[28]比较了两个自动化工具寻找Web程序漏洞的准确性，并提出自动化工具有效性的两个方面，即必须尝试检测Web应用的所有漏洞和必须报告该工具所检测到的所有漏洞。

2.1.1 虚拟执行分析检测

虚拟执行分析检测又称为沙盒检测。这种检测技术是通过在虚拟环境中执行检测，基于运动行为是否发生判定攻击。FireEye公司提出的基于终端异常行为检测的FireEye安全保护平台就是由多矢量虚拟执行(multi-vector virtual execution, MVX)引擎、动态威胁信息(dynamic threat information, DTI)模块以及共享威胁信息的DTI云协同工作。MVX引擎能够动态地、少签名地、虚拟化地分析网络攻击新品种。DTI是一种以多向量威胁信息为特征的新威胁保护模型，提供面向未知APT威胁的网络攻击的保护。

为克服动态恶意软件分析耗时较多的困难，文献[29]提出一种基于虚拟时间控制的沙盒系统。该系统通过生成虚拟的时钟，使沙盒系统能够加速运行，减少了恶意软件分析的时间，提高了检测效率。

2.1.2 异常流量检测

异常检测通常是对异常流量进行检测，如在短时间内剧增的流量以及不合理长度的流量包等。这些异常流量有可能就是攻击流量。

文献[30]提出一种精确的实时网络测量系统，运用此系统可对网络进行实时监控和流量分析。该系统内置了异常流量检测机制，并被证明是可用于实时流量监控的高性能且可扩展的解决方案。

2.1.3 基于流量和深度内容的检测

该检测技术需要对全流量进行审计和对内容进

行深度分析，并对异常行为进行识别。在工业界，RSA实验室提出了一种基于网络异常行为检测的模型Beehive，它通过对网络环境中各类日志的大范围收集和分析进行异常检测，但检测结果有效性的确认仍然需要大量的人工鉴定工作。文献[31]提出了一种蜜罐和网络入侵检测相结合的APT防御框架。由于APT攻击手法复杂且种类繁多，APT的检测与防御技术也趋向于动态检测与防御。而结合大数据和机器学习以及人工智能的APT攻击异常检测能有更好的准确率。文献[32]采用基于机器学习的方法对数据包流量进行分类，从而识别出APT流量特征。文献[33]则通过数学建模来生成检测APT攻击的模型。

文献[34]提出一种基于蜜罐的入侵检测和预防系统，该系统结合了低交互蜜罐和高交互蜜罐的优点，能实时检测网络流量并在零日攻击检测方面有一定效果。

2.1.4　事件回溯和关联分析

该技术是发现可疑对象后，利用快速查找的方式对具有同样身份对象参与的事件进行回溯，查找是否存在其他可疑记录，并将这些可疑记录进行关联分析。

布隆过滤器在事件回溯和关联分析中是一个常见的应用。使用布隆过滤器能快速检索一个元素是否在一个集合中，因此在事件回溯中应用布隆过滤器能通过极少时间检索到目标对象。文献[35]提出一种基于前缀树的利用布隆过滤器查找IP地址记录的新方法。该方法通过改进识别误报的方式，显著减少了片外访问前缀树的数量，进而减少了使用布隆过滤器搜索IP地址记录在最坏情况下的时间。

2.2　动态变形攻击模型与检测机制

在APT攻击模型提出之前，针对P2P蠕虫、自动化攻击型病毒等较为复杂的网络攻击已经设计了攻击树、攻击图等较为普适性的攻击模型，以应用于病毒防治等领域。基于攻击树模型，洛克希德-马丁公司提出了网络杀伤链(Cyber-kill-chain)模型[6]，如图2所示。

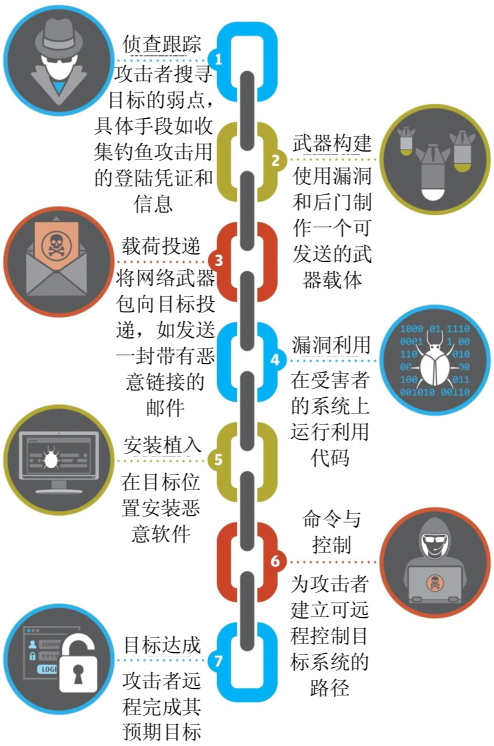

图2　网络杀伤链(Cyber-kill-chain)模型

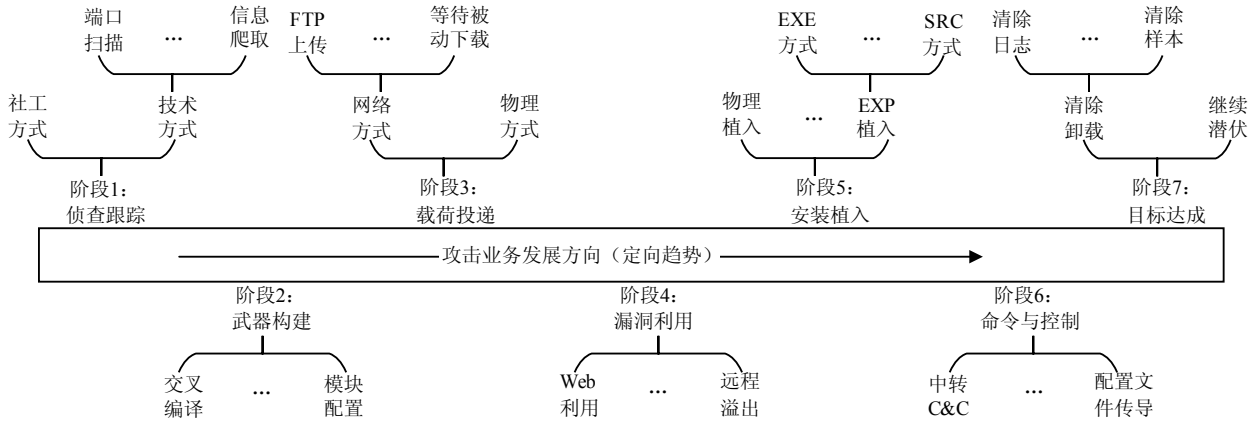

图3　网络杀伤链(cyber-kill-chain)的展开模型-A

上述模型的本质是将军事技战术思维和网络攻击过程结合在一起，共划分为7个常规阶段：侦查跟踪、武器构建、载荷投递、漏洞利用、安装植入、命令与控制、目标达成。每个阶段均包含一系列操作，使整个APT攻击犹如一个拥有总线的攻击树结构，如图3所示。但通过2016～2018年期间发现的发生于2008～2010年期间APT事件Valut7已经存在Cyber-kill-chain模型的例外实例，即不一定要走完其

7个阶段。例如，通过100余次定制化的无害漏洞样本利用来替代侦察跟踪等步骤，或者直接将安装植入和载荷投递过程合并为一个过程，典型的就是Valut8企业安全事件中的键盘芯片后门案例。因此，本文基于APT事件在攻防博弈过程中存在的局部不确定特点，提出了基于金字塔的展开模型，如图4所示。将网络杀伤链的7个环节置于金字塔的各层，然后将该棱锥在*XY*平面进行投射，相应的APT攻击案例则可以由“域三角、域内阶段节点、节点间过程箭头”三元组进行表述，进而完成对Valut7/8等案例的特殊情况的抽象表达。该模型将应用于未知类型的攻击检测以及可能的攻击路径预测等领域。

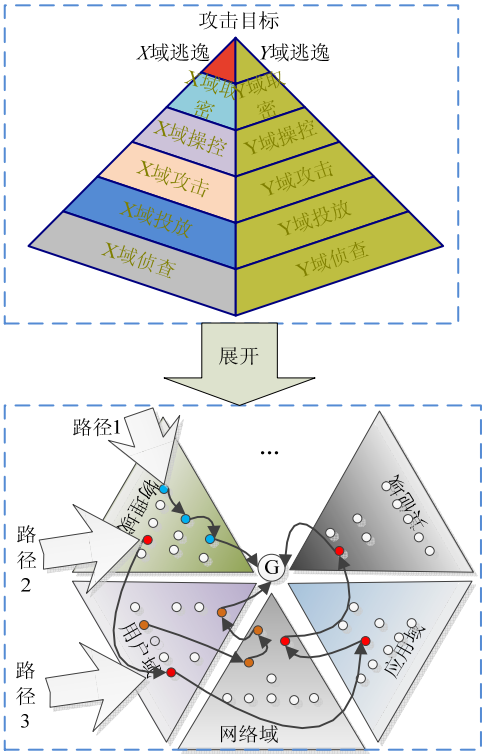

图4 基于金字塔的展开模型

2.3 攻击溯源与反制技术

攻击溯源与反制技术最早由拒绝服务攻击(denial of service, DOS)事件引发学术与工程界的关注，主要是通过网络报文的首发者追溯来实现拒绝服务攻击案件的破获，因其发展迅速后逐渐将DOS的对抗延伸到分布式拒绝服务攻击(distributed denial of service, DDOS)。近年来结合APT等复杂攻击的追踪溯源、分析画像等研究，逐渐发展成一门新兴科学技术，防御检测方通过对攻击事件链条中各个环节的线索信息，最终可以定位到被攻击目标、攻击行为、攻击路径、攻击参与者甚至最终的攻击幕后发起者[36-37]；溯源技术也被广泛应用到国家公共安全执法、网络犯罪取证等领域，与网络取证、终端设备与系统软件取证、情报线索取证等3类，权威且强大的相关取证平台包括了FireEye NX/EMPS、惠普的WebInspect和Guidance公司的EnCase等系统。本文提出一种基于APT组织采用的工具集特征的攻击溯源方法，详见2.5节。

2.4 攻击技术分类研究

通过对恶意代码分类的研究，进行同类代码的分类识别，可进一步对APT攻击进行分类。目前主流的同源性分类技术如表1所示。

表1 主流的同源性分类技术

技术类型	使用的方法	方法原理	优缺点
静态技术	基于序列的方法	对比样本函数序列间的相似性	有扩展性，误报率较高
	基于图的方法	对比样本调用图间的相似性	结果较准确，缺少扩展性
	污点传播	标记并跟踪数据传播	结果较准确，消耗资源多
动态技术	动态特征建模	对样本执行时的动态特征建模	分类结果依赖于特征选取

早期的恶意代码同源性分类技术大多是静态的分类技术，大致分为基于序列的同源性分类技术和基于图的同源性分类技术[38-40]。基于序列的方法将恶意样本转化为伪代码，并从其函数序列判断不同样本间的相似性。基于图的方法则通过更高层的结构判定，使用函数调用图分析不同样本间的相似性。传统的基于字符串、令牌、抽象语法树和哈希等基于序列的方法能处理百万行级别的代码，但有较高误报率。为克服这些不足，研究人员在这些方法上提出改进，并提出了一些基于机器学习的同源性分类方法。文献[41]提出了基于应用程序接口(application programming interface, API)调用的一种自动恶意软件同源性识别方法。通过编程习惯，定义了6种类型的API调用行为，基于这6种调用行为使用Jaccard相似系数计算不同恶意软件的同源程度，并通过经验设立了一个阈值和该同源程度比较，得出样本间是否相似的结论。该方法在真实样本中的准确率较高，其系统框图如图5所示。当前主流的基于特征的同源性识别方法框图均与此框图类似。

文献[42]采用机器学习的方法，基于恶意软件关系图提出一种新型的评估模型。该模型中图形的构造不受大型数据库的影响，只需要测试集中样本的anti-virus(AV)标签信息，并对这些信息进行聚类分析。实验证明该模型具有抗恶意软件家族分类不一

致性和AV标签粒度不一致的抗噪能力，并能完成不同粒度的聚类分析。

文献[43]同样采用机器学习的方法，提出一种基于纹理指纹的聚类方法来分析恶意代码的同源性，并用实验验证了该方法的有效性和准确性。该方法可以分析缺少源代码的二进制恶意程序并分析其图像纹理指纹信息，并对其指纹信息进行聚类。

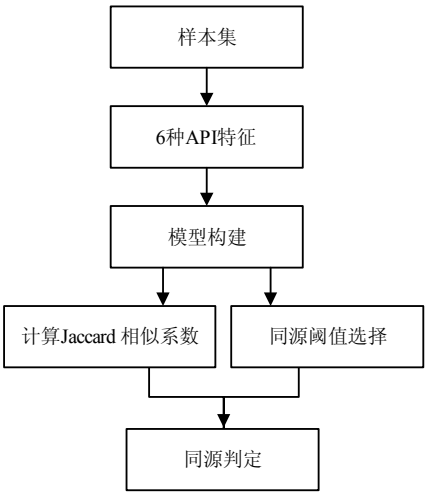

图5　基于API调用的一种自动恶意软件同源性识别方法系统框图

2.5　工具集溯源分析

2.5.1　攻击组件溯源分析

APT攻击团队的技术门槛和协作默契要求很高，最终都会体现在其发出的攻击样本工具、攻击策略等内容上，因此对这些事件中的工具集进行分析，寻找其关联规律、应用方式可以对现有的APT组织后续的作案手法进行有效提取。

工具集的初级形态是可执行样本，中高级形态是可被变形的组件化代码，以实现攻击任务中不同组件的灵活组合，同时也实现了开发团队和测试应用团队之间的解耦问题。相同功能的组件通常用Git、SVN进行版本控制，不同版本的相似度较高，将不同的工具集进行组件间的关联分析，并通过相关系数矩阵的数值分析来量化它们之间的相似性是比较好的方法。

本文通过生成$(2*n+1)$维向量集{工具集：组件1；组件1技术；组件2；组件2技术；…组件n；组件n技术}来表示一个工具集。当某个工具集未使用组件i时，定义该工具集组件i和组件i技术的值为0。

接着，使用n个$m\times m$维类相关系数矩阵$\{\boldsymbol{A}_{ij}\}_{m\times m}$表示不同工具集间相同组件的技术相似程度。其中，$m$为样本数，$a_{ij}$表示工具集$i$和工具集$j$间组件$a$的相似值，$a_{ij}=\dfrac{\mathrm{Cov}(I,J)}{\sigma_i\sigma_j}$。一般地，有$a_{ij}=a_{ji}$。该协方差通过组件的原始代码间的距离进行计算。两个组件使用的技术越相似，其代码间距离越小，相似程度也更高。最不相似的组件间相似值为0，完全相同的组件间相似值为1。特殊地，若某组件没有该工具集，则它和其他组件的该工具集相似度为–1，且不计入矩阵的数学特征统计。

通过统计协方差矩阵的数学特征，包括矩阵的秩、最大相关系数、最小相关系数和平均相关系数等，可以得出该组织使用的工具集的相似度统计。相似度越高，越能认为这些工具集间存在亲缘关系。

对APT29的工具集进行了关联分析。APT29常用的工具集如表2所示。

表2　工具集相关信息

工具集	开始时间	结束时间	C&C通信方式
PINCHDUKE	2008.10	2010.12	HTTP(S)
GEMINDUKE	2009.01	2012.12	HTTP(S)
COSMICDUKE	2010.01	2015.12	HTTP(S), BotgenStudios, NemesisGemina
MINDUKE	2010.07	2011.05	HTTP(S), Twitter
COZYDUKE	2010.01	2015.12	HTTP(S), Twitter (backup)
ONIONDUKE	2013.02	2015.12	HTTP(S), Twitter (backup)
SEADUKE	2014.10	2017.12	HTTP(S)
HAMMERDUKE	2015.01	2017.12	HTTP(S), Twitter
CLOUDDUKE	2015.06	2018.12	HTTP(S), Microsoft OneDrive

一个工具集通常包括数个加载器、窃取器，以及其他组件。如PINCHDUKE仅由几个加载器和一个信息窃取器组成，它通过窃取器收集系统配置信息、窃取用户凭证，并从受害主机收集用户文件，这些文件通过HTTP(S)传输到C&C服务器。而GEMINDUKE除了加载器和窃取器之外，还有多个持久性相关的组件，这些组件在受害计算机上嵌入额外的可执行文件，用来进行持久性的攻击。

通过组件的关联分析发现，在APT29的工具集中，尽管使用的工具不同，但工具间存在着很强的关联性。如GEMINDUKE和COSMICDUKE采用了相同的持久性组件技术。以加载器为例，得出的协方差矩阵的数学特征如表3所示。

从数据可以看出，不同的工具集对之间相关系数存在明显差异。理论上，两组完全不相关的工具集间的相关系数应不大于实验结果中的最小相关系

数。从最大最小相关系数商远大于1可以看出，关联最大的两组工具集组件间存在的关联远大于关联最小的两组工具集组件间存在的关联，且最大相关系数接近1，表明这两组工具集间关系非常密切，相似度超过了80%。由此可得，在本次实验分析的工具集中，至少有一对工具集的组件存在关联关系。又因为平均相关系数远大于最小相关系数，说明组件之间的关联关系在该工具集间广泛存在，相同组织内的组件相似度大于不同组织间的组件相似度。对不明来源的未知工具集，通过将其组件和不同组织的组件进行对比分析，组件相似度越高的工具集越有可能和未知工具集属于同一来源。

表3 loader的协方差矩阵数学特征

数学特征	特征值
秩	9
最大相关系数	0.824
最小相关系数	0.273
平均相关系数	0.563
最大最小相关系数商	3.018

2.5.2 信息窃取行为溯源分析

绝大部分工具集在渗入计算机系统后都会进行信息的窃取。这些被窃取的信息类型并不完全相同，但存在较大的相似性。可以认为，窃取信息类型的相似性越高，工具集的相似性也就越高。本文对工具集的信息窃取数据做出Apriori算法的关联分析。该算法是一种挖掘关联规则的频繁项集算法，其核心思想是通过候选集生成和情节的向下封闭检测两个阶段来挖掘频繁项集。

借鉴该算法的频繁项集定义，定义两组件窃取相同类型信息的数量为它们之间的相似频数，采用相似频数定义不同组件间的关联度。设组件 i 窃取了 m 项用户信息，组件 j 窃取了 n 项用户信息，且这两个组件间窃取信息的相似频数为 k，则有关联度 $a_{ij}=\dfrac{k}{m+n-k}$。一般地，有 $a_{ij}=a_{ji}$。其中，$m+n-k$ 是两个组件窃取用户信息项的合集，而 k 为两个组件窃取用户信息项的交集。

使用关联度组成相关系数矩阵，进行如组件分析中所示的矩阵数学特征分析，包括矩阵的、最大相关系数、最小相关系数和平均相关系数等，得到不同工具集间信息窃取的关联分析结果。

不同的工具集可能收集不同的系统配置信息或软件或服务相关的用户凭证，并通过HTTP(S)等方式传输到C&C服务器。这些系统配置信息包括但不限于：

本地用户账户、网络设置、网络代理设置、安装驱动、进程、用户最近执行程序、开机自启动程序和服务及环境变量值。

本文得出的关于信息窃取的协方差矩阵的数学特征如表4所示。

表4 信息窃取协方差矩阵数学特征

数学特征	特征值
秩	9
最大相关系数	0.836
最小相关系数	0.507
平均相关系数	0.674
最大最小相关系数商	1.650

从实验结果可以看出，不同的工具集对之间关于信息窃取的相关系数也存在明显差异。从最大和最小相关系数判断，这些工具集对之间存在不同程度的相似，且相似度在50%～83%之间。

实验结果表明了相同APT组织的不同工具集之间，信息窃取类组件的代码和行为相似度较高，应是其工具研发任务集中在特定工程师中所致。

而在不同APT组织间的工具集中信息窃取类组件的关联性较低，甚至所采用的基础平台和开发语言有巨大差异，这说明了其团队研发环境、思路和攻击类产品确实存在较小交集，而不同APT组织间所采用的EXP样本的相关系数较高且关系数标准差较低，说明其共同参考了同一个交叉漏洞来源组织。

2.5.3 幕后组织分析与溯源反制的研究

将相同APT组织内的工具集关联分析推广到不同APT组织间的工具集中。据经验，相同背景的APT组织间工具集的关联度较大，不同背景的APT组织间工具集的关联度较小。分析方法同上述的组件分析和信息窃取分析。

2.5.4 通信特征的关联分析

通过APT29事件中的工具集通信组件进行分析，发现攻击者的控制通道均在HTTP+TLS模式下进行了封装，导致与常见Mail等协议无法区分，同时攻击者还结合了第三方的公共网络服务(如：Twitter、微软云服务等)。

本文尝试采用更细粒度的通信特征来进行取样，例如：Twitter作为控制命令通道的工具集占约40%的归纳为APT组织X，而另一组事件中采用Twitter为控制命令通道的工具集占约67%的则为APT组织W。同时分析控制指令集的指纹，基本可

以确定相应的组织。

2.5.5　工具集关系分析

在360威胁情报中心编写的2017年中国高级持续威胁研究报告中，各个工具集之间存在一定的关系，如图6所示。

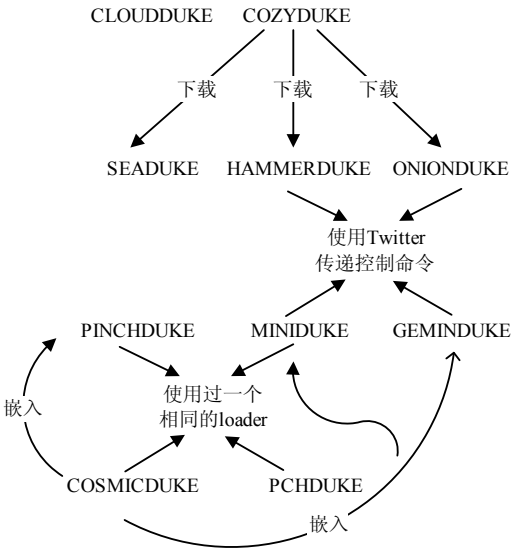

图6　工具集关系

可以看出，一个攻击组织内部工具集之间通常会有一定的联系，一个工具集入侵成功后，通常会下载同一组织下另外的攻击集作为第二后门使用；在组件利用方面，不同的工具集很有可能复用一些攻击组件，如图6中的loader。当然，样例集中的CLOUDDUKE工具集，根据收集的信息未发现与其他工具集的交集，所以不排除一个工具集可能会被单独开发使用。

3　APT整体防御方案

基于上述实际APT案例的分析，本文提出相对应的APT攻击快速检测与防御解决方案，大致分为4个方面，如图7所示。

1)恶意代码检测类方案：恶意代码是APT网络攻击过程中不可或缺的战略工具，也是基于这个原因，针对APT的恶代检测技术集中在未知恶意代码的表象特征和未知行为特征的自适应技术方面。大部分的工程方案采取了攻击者思维建模的方法来进行加权及动态加权的方式来检测假定动作，如ROP提权行为的可疑度较高而发生应用程序UPDATE协议的特征则可疑度较低等。除此之外，近年来国外高校提出的内建安全(build security in DNA, BSI)来智能化的分析新样本、新变种也是发展前景较好的方向。

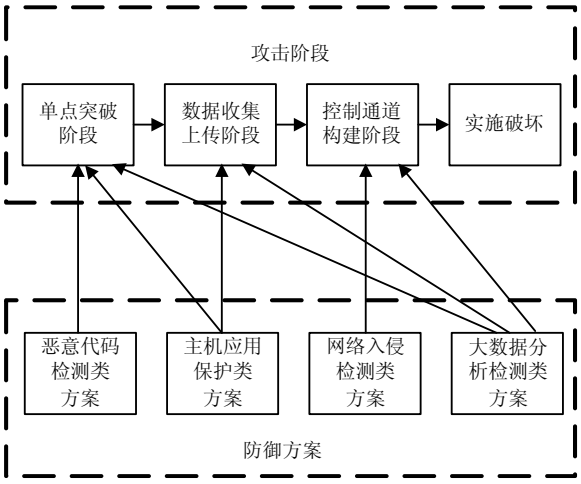

图7　APT阶段性防御方案示意图

2) 主机应用保护类方案：此类方案与恶意代码检测和防御等思路不同，重点不在于分析对手，而将重点置于自身系统资产的保护方面。事实证明这种思路虽然传统，但可以非常有效地对抗APT攻击以及大规模的病毒爆发等事件。近年来由Bit9、可信计算联盟等发起的白名单机制、虚拟隔离机制、可信基和可信应用框架技术均属于该方案的最新进展，核心是实现对主机及异构的嵌入式系统中所需运行的软件、固件、APP应用进行精细化管理、签名和认证、防篡改与运行环境保护来实现APT的抵御。

3) 网络入侵检测类方案：该类技术的研究周期较长且成果显著，主要集中在防火墙策略、网络入侵检测策略集等技术类别。但随着近年来越来越多的APT攻击引入了密码学协议和多通道技术，网络层的攻击检测开始向控制通道、僵尸网络Command&Control协议及网络协议漏洞攻击等方面倾斜发展。近年也开始逐渐发展出了网络核心资产与入侵检测联合分析防御以及网络内生安全等新领域，这些方向一旦落地应用不仅对APT检测有巨大作用，还将会大范围提升公共网络的治理能力。

4) 大数据分析检测类方案：采用大数据进行APT检测重点是利用了APT攻击周期长的弱点，在漫长的试探性攻击中存在较多被取证痕迹的可能，结合国内外的威胁情报数据库，可以快速定位类似行为集在几年跨度下的检索、关联性事件及样本。因此，大数据分析检测类方案也是APT防御的中台技术，是调度恶意代码检测流水线、主机安全防御、网络入侵检测以及威胁情报分析中枢性系统。核心技术重点还是集中在未知事件的无标度分类与聚类算法，事件关联性分析与样本亲缘度计算，基于日志数据的场景重构和安全大数据分析的实时性能力

等方面；国外的安全大数据巨头Palantir公司除了在上述先进技术方面有突出成果外，还提出了图分析应用、安全可视化等方案，将人机交互在安全风险识别与研判方面提升到了更高层次。

综上所述，由于APT攻击的复杂性和专业性，且过往研究人员所研究的内容较为分散，尚未形成针对APT攻击的完整解决方案，给网络空间安全造成了巨大的挑战。未来的APT安全解决方案应该会覆盖APT攻击的所有阶段，并拥有良好的可扩展性，以应对全面检测和防御APT攻击的能力。同时，这种方案应具有实时的检测和防御能力，将安全风险的检测与响应内置到网络业务当中，才能及时制止APT攻击，而不是在攻击发生后亡羊补牢。

4 结 束 语

APT作为网络攻防领域的重要课题，针对我国目前渗透防护脆弱、异常检测精度低、攻击发生后取证困难和对新型攻击响应慢等难题，需要尽快形成相应的防控体系。本文基于学术和工程领域的最新成果，重点突破在零日漏洞破门技术的实时检测与自学习进化等技术，持续升级攻击过程中所用样本的检测与防御能力，有效分析与阻断各类APT相关流量并降低误报、误杀等事件，力求研究形成一套具有扩展性的，且能投入实际业务应用的APT检测与防御性整体方案。

参 考 文 献

[1] LANGNER R. Stuxnet: Dissecting a cyberwarfare weapon[J]. IEEE Security & Privacy, 2011, 9(3): 49-51.

[2] TANKARD C. Advanced persistent threats and how to monitor and deter them[J]. Network security, 2011, 2011(8): 16-19.

[3] AUTY M. Anatomy of an advanced persistent threat[J]. Network Security, 2015(4): 13-16.

[4] CHEN P, DESMET L, HUYGENS C. A study on advanced persistent threats[C]//IFIP International Conference on Communications and Multimedia Security. Aveiro, Portugal: Springer, 2014: 63-72.

[5] VIRVILIS N, VANAUTGAERDEN B, SERRANO O S. Changing the game: The art of deceiving sophisticated attackers[C]//2014 6th International Conference On Cyber Conflict (CyCon 2014). Tallinn, Estonia: IEEE, 2014: 87-97.

[6] MUCKIN M, FITCH S C. A threat-driven approach to cyber security[J]. Lockheed Martin Corporation, 2015: 3(1): 1-8.

[7] GHAFIR I, PRENOSIL V. Advanced persistent threat attack detection: An overview[J]. International Journal of Advancements in Computer Networks and Its Security, 2014, 4(4): 50-54.

[8] VUKALOVIĆ J, DELIJA D. Advanced Persistent Threats-detection and defense[C]//38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). Opatija, Croatia：IEEE, 2015: 1324-1330.

[9] USSATH M, JAEGER D, CHENG F, et al. Advanced persistent threats: Behind the scenes[C]//2016 Annual Conference on Information Science and Systems (CISS). Princeton, NJ, USA: IEEE, 2016: 181-186.

[10] CHEN P, DESMET L, HUYGENS C. A study on advanced persistent threats[C]//IFIP International Conference on Communications and Multimedia Security. Aveiro, Portugal: Springer, 2014: 63-72.

[11] COLLBERG C, THOMBORSON C, LOW D. A taxonomy of obfuscating transformations[R]. New Zealand: Department of Computer Science, The University of Auckland, 1997.

[12] ANTUNES N, VIEIRA M. Assessing and comparing vulnerability detection tools for web services: benchmarking approach and examples[J]. IEEE Transactions on Services Computing, 2015, 8(2): 269-283.

[13] WILLIAMS L, MCGRAW G, MIGUES S. Engineering security vulnerability prevention, detection, and response[J]. IEEE Software, 2018, 35(5): 76-80.

[14] LI Z J, ZHANG J X, LIAO X K, et al. Survey of software vulnerability detection techniques[J]. Chinese Journal of Computers, 2015, 38(4): 717-732.

[15] NAKAJIMA A, IWAMURA M, TAKESHI Y. Vulnerability detection device, vulnerability detection method, and vulnerability detection program[DB/OL]. [2017-04-26]. http://www.freepatentsonline.com/WO2012097678.html.

[16] CHERNIS B, VERMA R. Machine learning methods for software vulnerability detection[C]//Proceedings of the 4th ACM International Workshop on Security and Privacy Analytics. Tempe, AZ, USA: ACM, 2018: 31-39.

[17] CHEN C, ZHANG F, YU H. Design of vulnerability detection system for web application program[J]. Computer Technology and Development, 2017, 27(9): 101-105.

[18] JI T, WU Y, WANG C, et al. The coming era of alphaHacking?: A survey of automatic software vulnerability detection, exploitation and patching techniques[C]//IEEE 3rd International Conference on Data Science in Cyberspace (DSC). Guangzhou, China: IEEE, 2018: 53-60.

[19] LIU P, ZENG Q, CAO C, et al. System service call-oriented symbolic execution of android framework with applications to vulnerability discovery and exploit generation[C]//Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services. Niagara Falls, New York, USA: ACM, 2017: 225-238.

[20] QIANG W, LIAO Y, SUN G, et al. Patch-related vulnerability detection based on symbolic execution[J]. IEEE Access, 2017, 5: 20777-20784.

[21] MARCEL BÖHME, PHAM V T, ROYCHOUDHURY A. Coverage-based greybox fuzzing as Markov Chain[J]. IEEE Transactions on Software Engineering, 2017, 45(5):

489 - 506.

[22] SHABTAI A, MOSKOVITCH R, FEHER C, et al. Detecting unknown malicious code by applying classification techniques on OpCode patterns[J]. Security Informatics, 2012, 1(1): 1-22.

[23] NATARAJ L, KARTHIKEYAN S, JACOB G, et al. Malware images: Visualization and automatic classification [C]//8th International Symposium on Visualization for Cyber Security. Pittsburgh, Pennsylvania, USA: ACM, 2011: 4.

[24] 360 Threat Intelligence Center. The APT-C-06 organization's first APT attack analysis and traceability initiated by the "double kill" 0-day vulnerability (CVE-2018-8174)[EB/OL]. [2018-08-01]. http://zt.360.cn/1101061855.php?dtid=1101062370&did=210645168.

[25] ZETTER K. Countdown to zero day: Stuxnet and the launch of the world's first digital weapon[M]. New York: Broadway Books, 2015.

[26] MOHAMMED B. Penetration testing of vulnerability in android linux kernel layer via an open network (Wi-Fi)[J]. International Journal of Computer Applications, 2016, 134(6): 40-43.

[27] CHEN C K, ZHANG Z K, LEE S H, et al. Penetration testing in the IoT age[J]. Computer, 2018, 51(4): 82-85.

[28] MUNOZ F R, VEGA E A A, VILLALBA L J G. Analyzing the traffic of penetration testing tools with an IDS[J]. The Journal of Supercomputing, 2018,74(12): 6454-6469.

[29] LIU J, DU Y, YANG J, et al. SONAR: A scalable stream-oriented system for real-time network traffic measurements[C]//IEEE 16th International Conference on High Performance Switching and Routing (HPSR). Budapest, Hungary: IEEE, 2016.

[30] LIN C H, PAO H K, LIAO J W. Efficient dynamic malware analysis using virtual time control mechanics[J]. Computers & Security, 2018, 73: 359-373.

[31] SAUD Z, ISLAM M H. Towards proactive detection of advanced persistent threat (APT) attacks using honeypots[C]//Proceedings of the 8th International Conference on Security of Information and Networks. Sochi, Russia: ACM, 2015: 154-157.

[32] SIDDIQUI S, KHAN M S, FERENS K, et al. Detecting advanced persistent threats using fractal dimension based machine learning classification[C]//Proceedings of the 2016 ACM on International Workshop on Security and Privacy Analytics. New Orleans, Louisiana, USA: ACM, 2016: 64-69.

[33] CHANDRAN S, HRUDYA P, POORNACHANDRAN P. An efficient classification model for detecting advanced persistent threat[C]//2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI). Kerala, India: IEEE, 2015: 2001-2009.

[34] BAYKARA M, DAS R. A novel honeypot based security approach for real-time intrusion detection and prevention systems[J]. Journal of Information Security and Applications, 2018, 41: 103-116.

[35] MUN J H, LIM H. New approach for efficient ip address lookup using a bloom filter in trie-based algorithms[J]. IEEE Transactions on Computers, 2016, 65(5): 1558-1565.

[36] KHAN S, GANI A, WAHAB A W A, et al. Software-defined network forensics: Motivation, potential locations, requirements, and challenges[J]. IEEE Network, 2016, 30(6): 6-13.

[37] ACHLEITNER S, LA PORTA T, JAEGER T, et al. Adversarial network forensics in software defined networking[C]//Proceedings of the Symposium on SDN Research. Santa Clara, CA, USA: ACM, 2017: 8-20.

[38] NIKOLOPOULOS S D, POLENAKIS I. A graph-based model for malware detection and classification using system-call groups[J]. Journal of Computer Virology and Hacking Techniques, 2017, 13(1): 29-46.

[39] MOSKOVITCH R, ELOVICI Y, ROKACH L, et al. Detection of unknown computer worms based on behavioral classification of the host[J]. Computational Statistics & Data Analysis, 2008, 52(9): 4544-4566.

[40] ANDERSON B, QUIST D, NEIL J, et al. Graph-based malware detection using dynamic analysis[J]. Journal of Computer Virology and Hacking Techniques, 2011, 7(4): 247-258.

[41] QIAO Y, YUN X, ZHANG Y. How to automatically identify the homology of different malware[C]//2016 IEEE Trustcom/BigDataSE/ISPA. Tianjin, China: IEEE, 2016: 929-936.

[42] CHEN Y, LIU F, SHAN Z, et al. MalCommunity: A graph-based evaluation model for malware family clustering[C]//International Conference of Pioneering Computer Scientists, Engineers and Educators. Zhengzhou, China: Springer, 2018: 279-297.

[43] XIAOLIN Z, YIMAN Z, XUHUI L, et al. Research on malicious code homology analysis method based on texture fingerprint clustering[C]//17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). New York, USA: IEEE, 2018: 1914-1921.

编 辑 刘飞阳