反病毒技术发展四部曲

文 / 肖新光

2013年，尽管出现了斯诺登事件，让网络安全热度陡升，但从实际技术走向上看，并不是一个大场面年。人们在这一年更多是改进已经突破的技术，或知道早已存在的事实。在这个看似火爆、实则平淡的年份，作者回望反病毒技术的发展全程，不禁若有所思，遂总结其中较为熟悉的四个关键的发展阶段，献给读者。

现代反病毒的技术体系前端以反病毒引擎技术为核心，服务于主机与网络两个工作场景；而后端则靠大规模海量分析处理体系作为支持。前端所采集的文件和事件提交后端体系分析处理，而后端则为前端产品提供规则升级、模块分发和其他支撑能力，这个实时能力迭代构成了一个持续不竭的信息循环。

正如世界上一切漫长而精彩的过程一样，反病毒体系的形成也并非源自马基雅维利似的先验设计，而是在不断的威胁迁移和演进中逐渐成形的。其中既有大量规律和必然，也有部分巧合与偶然。

感染式病毒驱动的反病毒引擎的成熟（1988-1995）

在整个反病毒的技术体系中，反病毒引擎是最先成熟的。反病毒引擎是依赖于一组可扩展维护的数据结构，对待检测对象进行病毒检测和处理的程序模块的统称，而其所依赖的可维护数据结构的文件载体就叫做病毒库。由于其长期居于反病毒的核心位置，所以称其为引擎（Engine）。

随着需求病毒检测能力的场景日趋增多，反病毒引擎也逐渐成为一个与环境无关的检测体系，因此类似驱动保护等技术虽然也是反病毒的关键技术，但并非反病毒引擎的一部分。而网络的一些公开资料显然对此有所误解。

反病毒引擎是一个非常传统的概念，其在DOS时代依靠感染式病毒驱动已基本成形。在反病毒软件的早期经历了从专杀程序、组合工具、集成工具的过程，当数十个乃至上百个检测清除模块难以维护调试的时候，反病毒工作者开始将其检测方法形式化，将代码和数据分离。今天的反病毒引擎是由十几个主模块（用于关键格式解析、预处理、规则匹配等），成千上万个小模块（检测、处理大量不能采用通用规则和参数处理的）组成的，而与之配套的病毒库规则条数也以百万计、甚至超过千万。但如果按照不同模块在对象处理中扮演的角色，同样可以对反病毒引擎进行过程抽象，而将其划分为分流器、预处理器、匹配器、鉴定器、处置器几部分。反病毒引擎的基本原理正是分流器基于格式识别把不同的数据对象分流到不同的检测分支；预处理器则是对这些数据进行加工（如脱壳、解包裹、解码等）使它满足特征匹配的场景条件；匹配器是依托病毒库结构去完成特征匹配（如连续特征、正则特征、特征哈希等）；鉴定器则是实现未知检测和判定；处置器主要是进行感染式病毒的清除复原等工作，以及其他的一些场景处理工作。尽管实际的反病毒引擎机理要比其抽象过程更为复杂，但这种抽象思路其实在DOS时代已经完成，而后则是持续的能力迭代和归一化维护。其中令我经常感叹的一个掌故是，最早反病毒引擎开始脱壳处理并非是因为

病毒作者利用其进行免杀，而是一些软件作者的机器感染病毒导致所开发编译的软件工具也被感染，但因为作者自己没有发现，发布时又使用了压缩壳，导致病毒不能被检测，但却可以正常传播。而此时勇敢站出来、对看似并非很主流事件进行脱壳处理的厂商，由于建立范式后开始持续的相关能力积累，在木马时代到来时，就获得很大的技术优势。

表1 反病毒引擎抽象

抽象模块	DOS时代解决的问题
分流器	将文件划分成有毒和无毒格式，加快检测速度，降低误报。
预处理器	解ZIP、RAR等压缩包，脱PKlite等早期壳，基于虚拟机和相关模块对DOS时代加密、变形病毒的解密处理。
匹配器	基于KMP等算法的连续特征匹配、正则匹配的优化，基于缓冲区定位寻址和偏移的特征哈希检测。
鉴定器	对感染式病毒基于加权特点的启发式扫描。
处置器	基于参数和脚本的感染式病毒清除。

从表1即可看出，DOS时代后期本地反病毒引擎的模型已基本成形，到了1996年前后，对针对复杂结构的可执行体PE的检测处理和针对复合文档Office的宏病毒的检测处理方式库化后，本地反病毒引擎的基本思想就已彻底成熟。

而其后反病毒引擎的发展，亦出现了几个重要的节点。

■ 2002年开始的PUA（Potentially Unwanted Application，用户不需要的软件，国内也有称之为流氓软件、灰色软件等）的处理。由于PUA不是传统的单文件对象威胁，有其安装目录、文件列表、图标菜单项等，因此其并不适用传统反病毒的辨识和处理方式。反而看似粗糙的基于简单的路径和注册表特征的处理更为有效。

■ 而2005年之后，采用全可执行对象上报、云端全HASH匹配检测的新方法开始逐步走入产品实现当中。这种方法虽然失去了传统引擎的检测深度和规则鲁棒性，但亦有规则条数不再受限、规则即时生效的特点，逐渐成为了传统本地引擎的重要能力补充。而由于其不再依赖传统的思路和底蕴，这种方法也成为新兴互联网客户端安全厂商依靠二八定律赶超传统反病毒厂商的捷径。

■ 而Symbian、Android等智能手机系统的先后普及，也为反病毒引擎技术的发展带来了新的驱动力。由于在新的平台上可以抛开原有包袱，针对尚未海量膨胀的专有样本进行引擎重构，这给了新老厂商平等竞技的空间。而在这一轮循环中，国内安全厂商不再处于后发位置，这一点从安天、金山、360等厂商多次居于AV-TEST测试排名前列，即可看出。

蠕虫驱动网络侧检测技术的发展（1999-2006）

本世纪初，大规模蠕虫爆发的压力，迫使主机不再是反病毒的唯一战场，而必须延展到网络之上，之后对抗呈现出层次化和纵深化的特点，使恶意代码的网络侧检测技术逐渐成熟。

初始的恶意代码网络威胁，以Happy99等邮件蠕虫、CodeRed为代表的扫描溢出型蠕虫、BO为代表的后门控制为主。但此时网络安全的载体能力并不乐观，直路设备从吞吐量来说依然以百兆为主；过滤对象主要在包头层次，而非内容，更难以进行流的完整还原；而旁路设备所需要的零拷贝、并行协议栈等亦并不成熟；加之当时反病毒引擎自身的接口、移植性和检测速度等很多因素，也使其与设备的融合极为困难，而类似Snort早期病毒检测的IDS规则，则看起来那样粗糙。

此后，载体设备出现了两条不同的路线：一条仍坚守x86架构，坐享摩尔定律带动的能力成长，同时辅以专用加速硬件或GPU；另一条则走专用硬件道路——如FPGA乃至ASIC实现，也包括后来的Cavium为代表的MIPS多核平台。这些都为反病毒技术在网络上的延展建立了能力舞台。

检测方法上，在流还原与传统文件引擎结合的思路上，趋势科技在1995年就开始伏线，早早申请了基于代理机制进行流文件检测的专利，并得以在几年后持续挥舞专利的大棒打压倡导UTM的概念倡导者Fortinet和其他厂商。而安天等团队则从2002年起在包层次尝试重建一个不依赖预处理器的全规则引擎，并使之达到了千兆的线速，从而使引擎越过了协议还原的局限，直接与千兆零拷贝对接。但后来的发展证明，这是一个过渡型方法。随着规则急剧膨胀到百万量级，从包层次进行海量规则检测的软肋也开始显现出来。

随着硬件能力的增长，不仅流还原不再成为瓶颈，而且更为细粒度的协议解析也成为可能。而此时正值互联网客户端大发展，流氓软件和僵尸网络也跟随爆发。以Paloalto Network为代表的团队，提出了高精度防火墙的概念，并随后将之定名为NGFW（下一代防火墙）。下一代防火墙实现了更精细的协议，包括对大量小众应用协议的检测能力，以及对各种SNS实现了基于身份区分度的划分，这也应对了并不多见的SNS蠕虫。而这些技术积累在APT的应对中则呈现出后发优势。

综合来看，尽管经历了长期的发展演变，网络侧恶意代码检测的核心方法依然可以概括为以下三种。

- **载荷检测：**即对病毒体的检测，例如传输中的可执行文件、脚本、溢出文档和溢出包等。
- **行为检测：**对恶意代码心跳、控制、数据回传等的检测。
- **地址检测：**对域名、URL和C&C地址的检测等。

从网络侧技术的走向来看，一方面要不断细化检测粒度，增加检测维度；另一方面也要不断应对高带宽的影响。这种趋势今后还会持续下去。

木马数量膨胀驱动了反病毒后端分析体系的完善（1995-2010）

反病毒技术如同一座冰山，反病毒产品只是冰山之上水面的部分，而其更庞大的部分则在水下，那就是病毒后台分析机制。这是一个庞大的基于大量计算节点形成的分布式处理体系，我习惯性地将其称之为自动化分析流水线。而驱动这个流水线成熟的最重要驱动力则是恶意代码，特别是木马数量的爆发式增长。

后台系统的成熟可以分成以下几个阶段。

- 早期的反病毒团队都是规模较小的软件作坊，病毒样本都简单地以本地文件存放；一些简单的本地工具用于病毒分析；病毒库则依靠类似TXT2BIN的行命令工具手工生成，这是一个典型的手工作业阶段。一直到1995年，很多团队依然采用这种方式，但也有一些先发团队依托UNIX或者Novell系统，搭建了早期的协作分析系统。
- 随后，几乎是随着Win9x系统开始普及，TCP/IP协议得到广泛应用，相应数据库等系统都已成熟，建立协同体系的开发成本大大降低，而样本数量的继续增加和反病毒团队的规模成长，也将样本共享、任务分发的诉求提升。反病毒工作者将原有的单机分析工具进行整合，形成了一套集成化客户端，并从服务端获取样本与任务，再提交回分析结果。而在C/S模式的服务端，已能完成基于静态分析的基本的规则自动化提取循环。但由于此时虚拟机技术尚未成熟，大规模分布式行为分析并未得到普遍应用，后端分析则更多起到给前台分析员提供更多判定信息的作用。
- 2005年起，一方面是网络大发展拉动的应用数量快速膨胀，另一方面则是地下经济拉动其木马数量急剧地几何级数膨胀，反病毒厂商提取的每日未知文件数量迅速由万量级迅速提升到十万、百万量级。巨大的辨识压力横亘在反病毒工作者面前，依赖自动的方式对全部样本完成辨识，成为是硬性的要求。通过虚拟机技术、分布式计算、云计算技术和行为分析、对照扫描、文件信誉、终端信誉等方法的引入，整个体系的鉴定能力逐步得以对全量文件实现自动化的“鉴定-规则-检测”循环，而只会把少数疑难问题（如一些加密变形病毒的检测和复杂感染式的清除等）留给人工处理。而体系也不再是简单的前台产品与后端处理，而是融为一体。我把这个体系的价值，概括为三个字母：W、A、R。
  - **恶意代码的场景和来源：**When（传播和关联事件的时间），Where（传播感染范围和影响），Who（是谁编写的）。
  - **恶意代码的属性特点：**Attribute（属性），Action（行为），Name（分类、命名）。
  - **对应的处置响应方式：**Response。

这个三个字母恰好可以凑成war（战争）一词，而后台体系化的分析能力，也正是AVER与病毒之间旷日持久战争中最重要的战略支撑点。

APT驱动了安全能力前置化与新技术变局（2010至今）

2005年起，服务于美国空军的网络工程师为了更好地进行一些基于邮件前导投放的信息窃取攻

击，开始把这些攻击称为APT。2010年，随着震网（Stuxnet）病毒重挫伊朗核进程，APT终于成为了以国家和政经集团为支撑，对特定目标长期持续作业的网络新威胁的统称。

由于APT广泛使用0day漏洞、隐蔽通信、签名仿冒，加之攻击者承担成本能力之强大、攻击意志之坚决，前所未有，其对安全体系的冲击和造成的心理恐慌都到达了空前的程度。这种压力一方面驱动了传统反病毒厂商进行产品和技术的改进，另一方面也驱动了新兴厂商的出现。

FireEye倡导了传统网络侧检测设备与沙箱结合的产品形态，其核心价值不仅是可以将可执行对象直接投放到设备附带的虚拟环境中运行，进行行为判定，更重要的是利用这个虚拟环境实现利用不同的解析器、也包括不同的解析器版本打开，以诱发文件格式溢出。这样就可以让0day漏洞在数据向代码转换的过程中被显现出来。而国内瀚海源的星云、安天的追影等都是同类方向的产品。

图1　安天在分析Stuxnet过程中临时搭建的工控沙盘

Bit9则引领了企业级终端防护产品基于白名单和安全基线进行重构的浪潮。由于反病毒是一种易于获得的资源，导致其易于进行对抗测试的传统软肋难以改变，因此利用企业网络环境相对单纯的特点，建立一套自定义的白名单则成为一个新的安全选择。当然，如果只有单纯的相关机制只是一种静态执行体的可信，其对溢出、脚本等依然不能有效应对，需要传统的主动防御机制进行补充，在这一点上，无疑传统反病毒厂商更有优势。而在白名单线路上传统企业反病毒的成熟架构、公有云安全知识的积累，也都有独特的优势。因此我们也看到，国内的金山安全、360企业级产品线也纷纷跟进形成解决方案，不仅发布了私有云产品，也将基于沙箱的鉴定器前置。

总体来看，无论是网络侧与沙箱结合，还是私有安全云，都反映出在APT的高度定向化以及持续化攻击的压力下，安全解决方案呈现出了能力前置、知识私有的趋势。网络侧的沙箱与传统反病毒的后端自动化行为分析并无本质的差异，而私有安全云亦可看成是厂商安全云的微缩版本。其革命意义不在于技术点，而在于部署位置和安全资产所有者的变化。

结束语

反病毒技术和体系从20世纪80年代后期发展至今，如一道穿越时空的硝烟火线，是信息技术的保卫者和使用者联合与威胁对抗的不屈历史。每当恶意代码展现出新的趋势、威胁和压力时，反病毒工作者都在积极求变，作出应对。虽然魔道之高下，难有公论，但显然作为守方的我们，虽有短暂被动尴尬之时，但从无无计可施之日。在这种持续的对抗中，既形成了反病毒的体系能力和方法，也历练了反病毒团队和从业者的价值取向。

此间的精彩过程显然不止我所描述的四段，只是这些对于我而言参与更多而已。作为一名反病毒老兵，从20世纪90年代分析学习国外反病毒引擎起步，2001年正式开始反病毒引擎的设计工作，完整经历了团队建立网络恶意代码检测能力和驱动后台分析机制成熟的全程，今天亦与同事们依托这些工作积累，投身APT的检测与对抗。虽心力微薄，才华拙劣，依然虔诚前行，值2013岁尾临近，作此总结，希望能带领读者分享我们一直以来的经验与坚持、以及我们对这份正直而有原则之事业的热爱。

（文章系根据作者在ISF2013会议报告部分内容和国防科技大学同名报告录音整理稿整合并缩改而成。）

肖新光

网名江海客，安天实验室首席技术架构师，研究方向为反病毒等。新浪微博：@江海客