DOI: 10.13245/j.hust.160304

基于主机与云分析结合的轻量级威胁感知系统

彭国军[1] 王泰格[1] 刘 焱[2] 张焕国[1]

(1 武汉大学 a 空天信息安全与可信计算教育部重点实验室, b 计算机学院,
湖北 武汉 430072; 2 百度在线网络技术(北京)有限公司, 北京 100085)

摘要 提出了一套基于主机与云分析相结合的轻量级威胁感知系统,该系统从主机捕获敏感行为日志,然后在云端对其进行分析处理.该系统的优势在于行为捕获过程实现用户无感知,将复杂的分析过程放到云端实现,既能捕获到进程级别的主机行为信息,又不会对主机产生较大性能压力,还能从云端进行主机间的关联分析.该系统已部署 1.763 6×10⁴ 台客户主机,经过实际运行检测,发现 114 个未知恶意程序,对未知恶意软件具有良好的检测效果,同时有效降低人均样本分析压力,显著提升了人工分析效率.

关键词 恶意软件; 进程行为; 关联分析; 云分析; 异常检测; 威胁感知
中图分类号 TP309 **文献标志码** A **文章编号** 1671-4512(2016)03-0017-05

Lightweight threat awareness system based on combination of host and cloud analysis

Peng Guojun[1] *Wang Taige*[1] *Liu Yan*[2] *Zhang Huanguo*[1]

(1 a Key Laboratory of Aerospace Information Security and Trusted Computing of Ministry of Education, b School of Computer School, Wuhan University, Wuhan 430072, China;
2 Baidu Online Network Technology (Beijing) Co. Ltd., Beijing 100085, China)

Abstract A lightweight threat awareness system based on the combination of host and cloud analysis was proposed in this paper. The system captured sensitive behavior from hosts, and then analyzed the log in the cloud server. The advantage is that the process of capturing user's behavior is imperceptible, and the complex analysis is achieved in server. The solution can reduce the pressure of performance on host, and implement the correlation analysis in cloud as well. Our system has been deployed into 1.763 6×10⁴ clients, and 114 malwares that are failed to be declared by current commercial antivirus software has been detected.

Key words malware; process behavior; correlation analysis; cloud analysis; anomaly detection; threat perception

由于恶意程序种类和数量的急速增长,现有主机及网络防御手段无法很好地应对所有攻击行为. Anderson[1] 首先提出了入侵检测的概念, Denning[2] 提出异常检测统计模型,自此异常检测方法的研究一直受到学术界的广泛关注.常见的异常检测系统可以分为基于主机的异常检测方法[3-6]和基于流量的网络异常检测方法[7-9]两大类.前者可能对主机性能造成较大影响,而后者则无法准确定位到进程并获取其所有恶意行为.目前国内外对恶意程序检测技术大致分为基于特征码的检测技术[10]、基于行为的检测技术[11]、基于内存完整性的木马检测技术[12]和基于人工免疫

收稿日期 2015-05-28.
作者简介 彭国军(1979-),男,副教授,E-mail: guojpeng@whu.edu.cn.
基金项目 国家自然科学基金资助项目 (61202387, 61373168, 61202385); 中国博士后科学基金资助项目 (2012M510641); 高等学校博士学科点专项科研基金资助项目 (20120141110002); 武汉市青年科技晨光计划资助项目 (201271031367).

的检测技术[13]四类,它们各具优势也各有其局限性.其中基于行为的检测技术近年来被学术界广泛研究.

然而对该类技术的研究中,很少有研究者将其与云分析相结合,因此无法对不同主机的行为做关联分析.较高的误报率也是该方法无法运用到实际检测系统中的一个重要原因.

为此,提出了一套基于主机与云分析相结合的轻量级威胁感知系统,并将其部署在国内某大型互联网企业内网中.本系统在客户端搜集所有进程的敏感行为,并上传到服务器端进行分析处理,借用已有的商业沙盒做辅助分析以减轻人工处理的压力.

经过在 $1.763\ 6\times10^4$ 台真实员工电脑中稳定运行 60 d 后,本系统成功发现 114 个现有安全机制没有发现的未知恶意程序.

1 系统架构

本系统由基于主机的轻量级敏感行为捕获模块和基于服务端的威胁感知模块两个主要模块组成,整个系统的架构如图 1 所示.

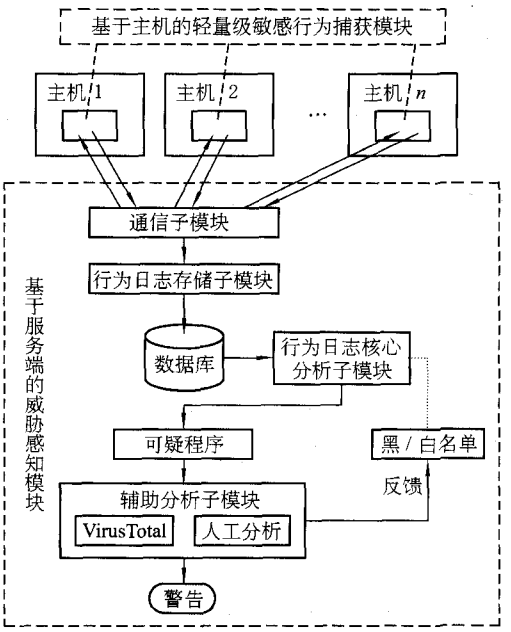

图 1 系统总体架构

基于主机的轻量级敏感行为捕获模块运行在员工电脑中,其在不影响用户使用的前提下捕获系统中所有进程产生的敏感行为,将行为日志发送到服务端存储分析.

基于主机的轻量级敏感行为捕获模块的内部结构如图 2 所示,图中 DLL(dynamic link library)为动态链接库.

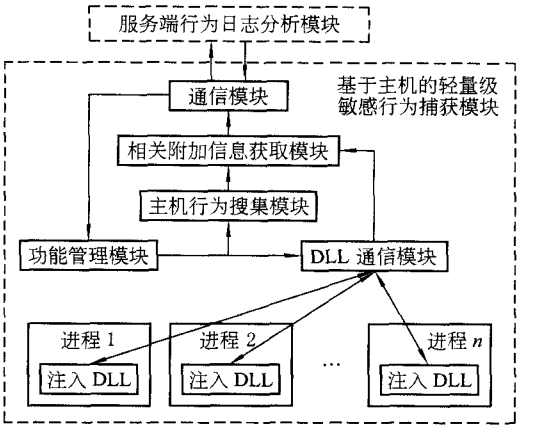

图 2 基于主机的轻量级敏感行为捕获模块框架

2 系统实现

2.1 基于主机的轻量级敏感行为捕获模块

本模块用于搜集客户端主机上各进程的敏感行为,并将行为的详细日志回传给服务端程序进行存储、处理和分析.目前行为捕获的方法分进程行为监测和系统环境监测两类.在捕获到进程行为后,本模块会将该进程的相关信息发送给服务端程序,其中包括进程名、进程号、进程主模块文件详细信息、相关行为的附加参数、父进程名、父进程号、父进程主模块文件详细信息.文件的详细信息包括文件名、文件路径、文件 MD5、文件大小、文件属性(隐藏/系统)、文件签名信息.

2.1.1 进程行为监测方式

本行为捕获方式须将 DLL 注入被监控的进程,使用应用程序编程接口(application programming interface, API) HOOK 技术对相关系统 API 函数进行挂钩,从而捕获敏感行为.出于性能和稳定性方面的考虑,本系统中采取用户层 Inline HOOK 技术进行行为捕获,而放弃使用更底层的驱动实现.使用此方式捕获的敏感行为包括开启摄像头、开启录音、截屏、创建服务、开启服务、创建文件、端口转发、远程注入 DLL、提升权限.

2.1.2 系统环境监测方式

对进程注入 DLL 挂钩 API 的方式一定程度上会对系统性能产生影响,为尽量减小此影响,本系统的部分行为捕获使用直接对系统环境进行监测的方式完成.系统环境检测方法和检测行为如表 1 所示.

2.2 基于服务端的威胁感知模块

本模块主要负责将客户端程序收集到的行为日志以及相应文件详细信息并存储到数据库中,

然后统一进行分析. 其中包括四个子模块：通信子模块，行为日志存储子模块，行为日志核心分析子模块，辅助分析子模块.

表 1　系统环境检测方法和检测行为

行为	检测方法
隐藏界面的 cmd 程序	查找 cmd. exe 进程界面是否可见
自启动程序	定期扫描与自启动相关的注册表项等
运行 bat	查找 cmd. exe 进程的参数列表中是否包含 bat 文件
rundll32 运行 DLL	查找 rundll32. exe 进程的参数列表中是否包含 DLL 文件
使用可疑文件名	遍历所有进程，查找进程主模块可执行程序文件名，查看其中异常字符的比例
连接网络	扫描系统内 TCP/UDP 连接的方式获取连接内外网的程序列表
加壳程序	判断导入函数个数等
创建傀儡进程	对比进程主模块内存与相应磁盘文件中文件头部分结构是否相同等
产生异常流量	使用 WinPcap 工具捕获数据流量，从而进行流量分析，根据木马流量特征模型进行判定[14]

行为日志核心分析模块和辅助分析子模块负责对行为日志进行分析从而实现威胁感知功能. 本系统对恶意程序的确定性判断主要经过人工分析实现，由于整个企业每天产生的行为日志量太大，超过人工能够处理的极限；因此，行为日志核心分析子模块的功能就是从海量日志数据中筛选出高度可疑的少量数据，然后在辅助分析子模块中使用 VirusTotal 对确定正常的程序进行进一步剔除，从而极大降低人工分析的压力.

通信子模块. 通信子模块用于将服务端命令（开启/关闭某功能、更新客户端程序、卸载客户端程序）发送给所有客户端主机，并接收客户端程序的定期心跳包连接，以及客户端程序发回的敏感行为日志.

行为日志存储子模块. 除了行为自身之外，不同的行为还包含不同的附加参数，行为日志存储子模块负责将收到的敏感行为日志根据不同的行为进行字段解析，解析完成后再将日志写入到数据库.

行为日志核心分析子模块. 经过对行为日志数据长时间的分析总结，提炼出以下分析方法，旨在从海量日志数据中筛选出在量级上可进行人工分析的少量高度可疑的行为数据. 本模块首先使用白名单过滤，然后使用两个并列的分析方法同时判定筛选可疑程序，下面分别对这两种方法进行简单描述.

a. 主机间关联分析方法. 该方法旨在筛选出文件大小小于阈值、具有特殊属性，以及在多台电脑上存储的文件名不同的程序. 根据对恶意样本的分析得知：常见恶意样本的文件都比较小，其中绝大多数小于 1 M，因此通过文件大小可以排除一部分正常程序. 具有隐藏或者系统属性的文件一般是系统程序或者恶意程序，由于具有此特点的系统程序的数量很有限，通过白名单可以将其排除. 恶意程序常为了隐藏，在每次感染过程中产生不同的文件名，因此当同一个 MD5 的 exe 文件在不同用户电脑出现且文件名不同时，排除明显重命名程序（两个不同文件名差别为“一副本”、“(1)”等）后，可筛选出部分感染多台电脑的恶意程序.

b. 进程树威胁度评分方法. 对捕获到的所有行为设定不同威胁值，并将所有行为共分为间谍行为、驻留行为、隐藏行为、网络行为和其他行为五类，计算进程树威胁评分时将所有节点中的行为全部归总到整棵树的行为，即 $T=5(C-1)+\sum_{i=1}^{N}T_i$，其中：$T$ 为进程树威胁度评分；C 为所有行为涉及到的类别数；N 为涉及到的行为总数；T_i 为第 i 种行为的威胁值. 各行为的威胁值和类别如表 2 所示.

表 2　敏感行为威胁值对照表

序号	行为	威胁值	类别
1	截屏	3	Ⅰ
2	开启摄像头	3	Ⅰ
3	开启录音	3	Ⅰ
4	开启服务	1	Ⅴ
5	频繁访问文件	7	Ⅰ
6	创建服务	6	Ⅱ
7	远程注入 DLL	8	Ⅲ
8	端口转发	5	ⅤⅤ
9	提升权限	7	Ⅴ
10	文件敏感属性	8	Ⅲ
11	隐藏界面的 cmd 程序	5	Ⅰ
12	自启动程序	7	Ⅱ
13	运行 bat 程序	1	Ⅴ
14	使用 rundll32 运行 DLL	4	Ⅴ
15	使用可疑文件名	9	Ⅲ
16	连接内网/外网	5	Ⅳ
17	加壳程序	8	Ⅴ
18	创建傀儡进程	8	Ⅲ
19	产生异常流量	7	Ⅳ

本系统总体架构属于模块化设计，行为日志核心分析子模块逻辑上独立于系统其他模块，因此当发现新的检测方法时，无须对主机上客户端程序进行任何修改，直接在此模块中添加新的云

分析子模块即可实现其性能的提升.

2.2.4 辅助分析子模块

经过行为日志核心分析模块对大量日志数据进行分析筛选,会得到少量高度可疑的文件以及对应的行为.此模块通过辅助手段对高度可疑的文件进行进一步分析,从而对其恶意性进行判定.本模块主要包括以下两个检测步骤.

a. 使用 VirusTotal 对文件 MD5 进行检测. VirusTotal 是一款提供免费的可疑文件分析服务的网站,其使用全球 50 款以上知名反病毒引擎对样本进行扫描.为评估 VirusTotal 对文件 MD5 的检测效果,本系统将一段时间内收集到的总共 $6.005\ 5\times10^4$ 个文件的 MD5 提交到 VirusTotal 上进行扫描.其中有 $1.010\ 9\times10^4$ 个文件存在于 VirusTotal 的数据库中,这些文件的扫描结果如表 3 所示,表中 N 为判定此文件是恶意程序的反病毒引擎的个数.结果表明:绝大部分的程序判定时都只有不到半数的杀软报警,因此这里不依此进行确定性判断.

表 3 VirusTotal 扫描结果

N	报警数	比例/%
$N=0$	1 248	12
$N=1$	3 838	38
$2\leqslant N<10$	3 184	32
$10\leqslant N<20$	567	6
$20\leqslant N<30$	601	6
$30\leqslant N<40$	526	5
$N\geqslant40$	145	1

由于不同杀毒软件对恶意性的定义不同,对于分析测试人员平时可能使用的黑客工具等程序,大部分反病毒引擎会将其判定为恶意,但是实际不会对员工电脑产生威胁,因此即使大部分杀毒软件认为恶意的程序本系统也不能直接将其判定为恶意软件.经过分析,有理由假设所有反病毒引擎都认为正常的程序就是正常程序,其他情况则须进一步分析.

为此,本模块中先使用 VirusTotal 对高度可疑的文件进行 MD5 扫描,当发现报警数为 0 时直接排除此文件,不进行人工分析.而对于所有反病毒引擎都认为正常的 MD5,则直接将其加入白名单.

b. 人工检测.由于很多误报的正常程序都属于常见的服务性后台程序、常见的黑客测试工具、部分安全工具以及系统管理工具,凭借简单的人工查看就能进行判断其是否是误报.而对于无法确定的可疑文件,分析人员会对特定客户端发送命令将文件收取到服务器,然后进行进一步静态及动态分析.人工分析结束后,将结果记录到黑白名单中.

3 系统测试

3.1 样本测试

样本测试主要针对进程树威胁度评分方案进行,本检测方法对远控型木马等功能复杂的恶意程序具有很好的检测效果,这里主要使用木马程序进行测试分析.样本测试过程使用了 10 款常见的木马程序和 10 款常用的正常程序进行测试.参与测试的恶意样本及检测结果如表 4 所示,正常样本及检测结果如表 5 所示.

表 4 恶意样本检测结果

序号	恶意样本名	威胁度评分
1	贝壳安全网远控第二版	30
2	黑防灰鸽子专版	54
3	BlackHoleV1.98	31
4	Gh0st RAT beta 3.6	30
5	上兴 2014	42
6	暗组远控	16
7	大白鲨	26
8	华中帝国技术论坛系列远控第28版	28
9	IM 远控	51
10	PI2.3.2	33

表 5 正常样本检测结果

序号	正常样本名	威胁度评分
1	Office Word	1
2	迅雷 7	14
3	迅雷看看播放器	14
4	优酷客户端	5
5	百度云管家	5
6	Evernote	5
7	Chrome	5
8	有道词典	5
9	QQ	5
10	网易云音乐	5

测试结果显示:正常程序的威胁度评分绝大部分情况下均远小于木马程序的威胁度评分.经过对测试样本进一步分析得知:6 号木马程序(暗组远控)的威胁度评分没有表现出明显高于正常程序,其主要原因在于网上下载到的该样本部分功能不完整,在 Windows 7 操作系统下没有表现出隐藏驻留等行为.

3.2 实际使用测试

本系统在国内某大型互联网企业员工电脑中成功部署运行,安装部署的客户端数量总共为

1.763 6×10⁴ 台，部署两个月后共有活跃客户端 1.207 6×10⁴ 台，客户端统计数据如表 6 所示.

表 6　客户端统计列表

操作系统(OS)	总客户端个数	活跃客户端个数
Windows XP	178	47
Windows7	1.739 4×10⁴	1.199 9×10⁴
Windows8	56	26
Windows 2003	4	2
Windows 2008	3	1
Windows 2012	1	1

该企业的员工电脑中已经安装了全套企业级安全防护工具，其中主机上安装了某国际知名商业杀毒软件以及 Windows 主机防火墙，网络层部署了先进的 IDS，IPS 以及网络防火墙.

本系统持续运行时间为两个月，客户端及服务端程序均持续稳定运行. 通过对用户抽样调查得知：本系统客户端程序运行时平均 CPU 占用率为 0%～3%，内存占用小于 20 MB.

经过 2 个月的运行，本系统在检测效果方面表现出较好结果. 工作日内平均每天收到行为日志近 2.5×10⁶ 条，总共收集到行为日志量超过 1.5×10⁹ 条. 平均每天检测到有敏感行为的文件约 1.5×10⁴ 个，系统稳定运行后，经过行为日志核心分析模块以及 VirusTotal 过滤，平均每天所需人工分析的文件约为 20 个，其中平均每天查找到 2 个新的恶意程序. 2 个月总共检查出恶意程序 114 个，总共检测到 190 次感染. 大部分恶意程序当被释放到操作系统时就会被商业杀毒软件处理删除，而本系统只针对已经运行的进程进行检测，因此本系统检测出的恶意程序均是该企业已有安全防护机制检测到的.

经测试分析可知：本系统运行稳定性良好，系统资源消耗小. 作为现有安全防护机制的一种补充，本系统能够很好地发挥其未知威胁感知的功能，进一步提升企业内部环境安全.

然而本系统仍存在一些不足，下一步的工作主要包括以下几个方面：

a. 部分恶意程序的主进程执行完部分恶意行为(设置自启动、创建服务等)后立即退出，由于该系统在应用层实现(保障系统稳定性)，无法及时获取进程创建的通知，导致对此类程序可能不能及时注入，从而可能遗漏一部分行为，该问题须进一步研究解决方案，必要时可以在确保稳定性的前提下考虑引入简单驱动实现注入操作；

b. 还有其他常见敏感行为(如：文件上传、修改系统账户、远程文件浏览等)有待继续添加；

c. 行为日志分析模块中分析方法还有较大改进空间，可考虑将人工判定后的结果反馈给分析模块，进而对进程树威胁度评分方案的参数进行修正和优化.

参　考　文　献

[1] Anderson J P. Computer security threat monitoring and surveillance[R]. Washington: Anderson Company, 1980.

[2] Denning D E. An intrusion-detection model[J]. IEEE Transactions on Software Engineering, 1987 (2): 222-232.

[3] Mustafa A, Solaimani M, Khan L, et al. Host-based anomaly detection using learning techniques[C]// Proc of 13th International Conference on Data Mining Workshops (ICDMW). Washington: IEEE, 2013: 1153-1160.

[4] Murtaza S S, Khreich W, Hamou-Lhadj A, et al. A host-based anomaly detection approach by representing system calls as states of kernel modules[C]// Proc of 24th International Symposium on Software Reliability Engineering (ISSRE). Washington: IEEE, 2013: 431-440.

[5] Murtaza S S, Khreich W, Hamou-Lhadj A, et al. A trace abstraction approach for host-based anomaly detection[C]// Proc of Symposium on Computational Intelligence for Security and Defense Applications (CISDA). Washington: IEEE, 2015: 1-8.

[6] Xie M, Hu J, Yu X, et al. Evaluating host-based anomaly detection systems: application of the frequency-based algorithms to ADFA-LD[C]// Network and System Security. Berlin: Springer, 2014: 542-549.

[7] Lu W, Ghorbani A A. Network anomaly detection based on wavelet analysis[J]. Eurasip Journal on Advances in Signal Processing, 2009, 12(5): 1234-1249.

[8] Fiore U, Palmieri F, Castiglione A, et al. Network anomaly detection with the restricted Boltzmann machine[J]. Neurocomputing, 2013, 122: 13-23.

[9] Bhuyan M H, Bhattacharyya D K, Kalita J K. Network anomaly detection: methods, systems and tools [J]. IEEE Communications Surveys and Tutorials, 2014, 16(1): 303-336.

[10] 陈健，范明钰. 基于恶意软件分类的特征码提取方法[J]. 计算机应用，2011，31(A01)：83-84.

[11] Hughes K. Detecting malware using behavior-based aggregated signature[D]. Colorado Springs: College of Information Technology, Colorado Technical University, 2014.

(下转第 27 页)

[5] Xie J Y, Jiang S. A simple and fast algorithm for global K-means clustering[J]. 2010 Second International Workshop on Education Technology and Computer Science, 2010, 45(90): 77-86.

[6] Raghavan U N, Albert R, Kumara S. Near linear time algorithm to detect community structures in large-scale networks [J]. Physical Review, 2007, 76(23), 36-41.

[7] Jϕsang A. A logic for uncertain probabilities[J]. International Journal of Uncertainty, Fuzziness and Knowledge-based Systems, 2001, 9(3): 24-48.

[8] Jnanamurthy H K, Singh S. Detection and filtering of collaborative malicious users in reputation system using quality repository approach[C] // 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI). Santa Clara Marriott: IEEE, 2013: 304-312.

[9] Leung I X Y, Crowcroft J. Towards real-time community detection in large networks[J]. Physical Review, 2009, 79(43): 66-70.

[10] Moon S, Lee J G, Kang M. Scalable community detection from networks by computing edge betweenness on MapReduce[J]. BigComp, 2014, 33(12): 44-52.

[11] Zhang Y, Zheng Z B, Lyu M R. WSPred: a time-aware personalized QoS prediction framework for Web services[C] // Proc of 22th IEEE Symposium on Software Reliability Engineering. Washington: IEEE, 2011: 475-486.

[12] Jiang Y C, Liu J X, Tang M D, et al. An effective Web service recommendation method based on personalized collaborative filtering[C] // Proc of IEEE International Conference on Web Services. Washington: IEEE, 2011: 211-218.

[13] 贾冬艳，张付志. 基于双重邻居选取策略的协同过滤推荐算法[J]. 计算机研究与发展，2013，50(5)：1076-1084.

(上接第 21 页)

[12] 芦天亮. 基于人工免疫系统的恶意代码检测技术研究[D]. 北京邮电大学计算机学院，2013.

[13] Bezobrazov S, Golovko V. Artificial immune systems approach for malware detection: neural networks applying for immune detectors construction [J]. International Journal of Computing, 2014, 7(2): 44-50.

[14] 彭国军，王泰格，邵玉如，等. 基于网络流量特征的未知木马检测技术及其实现[J]. 信息网络安全，2012(10)：5-9.

[15] Liang Y, Peng G, Zhang H, et al. An unknown trojan detection method based on software network behavior[J]. Wuhan University Journal of Natural Sciences, 2013, 18(5): 369-376.