“云安全”检测技术安全性分析

许 蓉，吴 灏，张 航

（解放军信息工程大学 信息工程学院，河南 郑州 450002）

摘 要：“云安全”检测已成为病毒查杀领域发展的新趋势，为对其在病毒检测过程中的安全性有进一步了解，研究了“云安全”检测体系结构以及主流“云安全”策略，针对某“云安全”检测软件的文件样本提取方式和网络传输数据的特点，分析了检测流程中存在的安全隐患，基于这些安全隐患设计并实现了“云安全”检测的规避方案，针对规避方案提出了防护建议。实验结果表明，“云安全”检测在实际应用过程中仍可能被恶意程序绕过。

关键词：云安全；恶意程序；云查杀；特征码；检测规避

中图法分类号：TP309.5 **文献标识号**：A **文章编号**：1000-7024（2012）09-3309-04

Analysis of cloud-based malware detection security

XU Rong, WU Hao, ZHANG Hang

(Institute of Information Engineering, PLA Information Engineering University, Zhengzhou 450002, China)

Abstract: Cloud-based malware detection has already become the trend in antivirus. To dig further, the frameworks and policies of mainstream cloud antivirus are analyzed, and a method is designed and realized to escape from cloud antivirus according to its features and workflow. In the end some experiments are carried to verify that possibilities of bypassing cloud antivirus still exist, and safety suggestions are given based on this.

Key words: cloud security; malware; cloud antivirus; feature code; detection avoiding

0 引 言

传统病毒查杀机制主要通过目标文件特征码鉴定方式查杀木马病毒。这种方式主要是通过将目标文件的特征码和本地病毒特征库内病毒特征进行比对，若匹配特征库中某病毒特征则判定此文件为病毒文件，判断的准确性取决于特征库是否全面。然而新病毒不断涌现，用户需要不断更新病毒库才能保证病毒库的升级全面，这必然使得病毒库越来越庞大，占用用户内存和系统资源越来越多，导致杀毒软件的扫描效率不断下降，系统性能也受到极大影响[1-2]。《瑞星互联网安全报告》显示 2011 年上半年中国互联网安全领域病毒总量比去年同期上升 25.2%[3]。病毒数量呈爆炸式增长使得传统杀毒软件面临严重的困境，因此杀毒软件必须要建立一种全新的病毒查杀机制。“云安全”利用互联网的传输及计算功能，将原来放在客户端的分析计算能力转移到了服务器端，很大程度上弥补了传统病毒查杀机制的不足。目前对“云安全”检测技术的研究着重于检测性能的提升[4]和服务器集群的防攻击保护[5]方面，而本文针对“云安全”检测系统的客户端，对“云安全”结构和主流“云安全”策略进行了介绍，同时对某“云安全”检测流程及其检测流程中存在的安全隐患进行了分析，设计并验证了规避检测的方案，为“云安全”检测系统提供防护建议。

1 “云安全”检测结构分析及主流策略

1.1 “云安全”检测结构分析

“云安全”是“云计算”理论在安全领域的应用，融合诸多新兴技术，如网格计算、并行处理技术、未知病毒行为判断技术[6]等，通过互联网将用户和杀毒软件厂商的服务器集群进行连接，形成一个庞大的防毒杀毒系统，对用户机器中软件的异常行为进行监测，从中获取病毒木马的特征信息并向服务器端传送，服务器端对其进行分析处理后再向各个客户端发送该病毒木马的解决方案[7]。“云安全”检测系统结构组成如图 1 所示。

“云安全”将原先客户端的分析计算工作转移到了服务器端，这要求服务器端必须拥有快速响应客户端请求与快速分析处理可疑文件的能力。为提高分析查杀的准确率，服务器端拥有多个快速分析引擎、庞大的病毒特征库以及

收稿日期：2011-10-21；**修订日期**：2011-12-26

作者简介：许蓉（1986-），女，江苏大丰人，硕士研究生，研究方向为网络安全；吴灏（1965-），男，河南驻马店人，硕士，教授，研究方向为网络安全；张航（1986-），女，陕西西安人，硕士研究生，研究方向为网络安全。E-mail：xurongyancheng@126.com

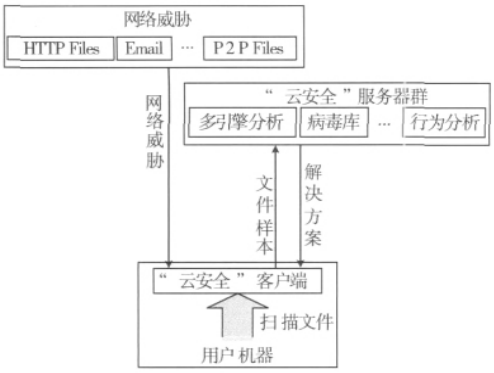

图 1 “云安全”系统结构组成

行为分析等多个分析技术。当用户访问网络信息时，“云安全”客户端首先将用户访问信息提交服务器进行安全评估，服务器快速响应分析后迅速将解决方案发送至客户端，客户端再根据解决方案提示用户该网络信息是否安全；当用户执行本地扫描时，客户端将可疑文件样本提交至服务器进行分析得到解决方案。

1.2 主流“云安全”策略

目前“云安全”的技术标准并不统一，各个杀毒软件厂商对“云安全”的理解各不相同，主要分为偏主动防御型和偏被动防御型两类“云安全”策略[8]。

偏主动防御型“云安全”以趋势科技为代表，其“云安全”使用大量的服务器构成一个具有庞大的黑白名单的“云”，通过 Web 信誉服务（web reputation services，WRS）、邮件信誉服务（email reputation services，ERS）和文件信誉服务（file reputation services，FRS）等核心技术，对网络访问信息进行安全评估，并拦截阻止 Web 威胁进入用户机器。该“云安全”系统的服务器数据库中存有大量的网络威胁特征值，客户端仅保存有少量的病毒特征码文件和 Web 信誉评级等数据用于本地验证。趋势云安全技术强调对 Web 威胁的拦截，通过动态分析，对网络访问信息进行安全等级评估，极大地降低了病毒对下载传染的依赖性，但是这种主动防御对本机上已经存在的未知威胁的有效感知能力相对较弱。

偏被动防御型“云安全”以瑞星为代表，其“云安全”拥有大量的客户端，每个客户端都通过已安装的“云安全的探针”对用户计算机进行扫描，截获、提取可能是恶意程序的文件样本，并将其上传至“云安全”服务器，服务器在自动分析和处理后再向每个客户端分发解决方案。这种被动防御模式能够对本地已经存在的未知病毒进行有效侦测和查杀，但对网络病毒的主动防御能力相对较弱。

1.3 “云安全”的安全防护

“云安全”检测采用轻客户端的策略，将分析计算工作以及病毒特征库等全部转移至服务器端，一旦云端服务器遭受攻击，比如篡改数据库等，“云安全”系统就无法对木马病毒进行正确判断，甚至无法正常运行，因此目前对于“云安全”的安全防护研究集中于对服务器集群的保护，对客户端的安全防护研究比较薄弱[9]。

2 “云安全”检测流程及安全隐患分析

2.1 “云安全”检测流程

以某“云查杀”（即“云安全”检测）软件为例分析其检测流程。该“云查杀”主要通过文件 hash 比对、文件样本启发式分析和行为分析三大检测技术对可疑文件进行分析判断。如图 2 所示，客户端软件首先查找可疑文件并上传该文件 hash 值至服务器集群，服务器端对该文件 hash 值进行黑白名单比对，发现异常则生成解决方案发送至客户端，未发现异常客户端则提取文件样本，经过服务器端对该文件样本的启发式分析检测后，发现异常则生成解决方案；若仍未发现异常，客户端软件则搜集该文件运行过程中的行为特征上传至服务器进行行为分析，服务器最终判定该文件是否可信并向客户端发送解决方案。

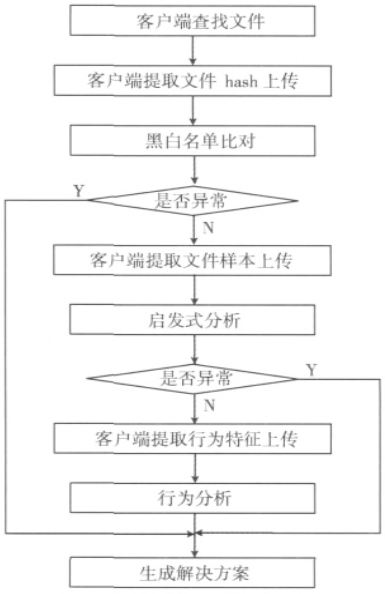

图 2 某“云查杀”系统检测流程

2.2 “云查杀”检测流程安全隐患分析

该“云查杀”系统依靠客户端上传的文件 hash 触发服务器端一系列的检测分析，在如图 2 所示的检测流程中，攻击任何一个环节都会影响最终解决方案的生成。但黑白名单比对、启发式分析和行为分析 3 个环节处于服务器端，由于“云安全”系统对服务器严密的安全防护，针对这 3 个环节的攻击相对比较困难，因此对处于客户端的各个环节进行分析发现，客户端查找文件和提取文件 hash 上传两个环节存在安全缺陷，极易遭受文件路径欺骗、通信欺骗

和断网攻击。

2.2.1　文件路径欺骗安全隐患分析

“云查杀”客户端对用户计算机进行实时监控，当有程序启动运行时，客户端将立即检查该程序的合法性，但客户端只是简单的根据启动项、系统服务等特定信息查找该程序位置并提取文件样本，恶意软件可以通过隐藏其真实文件路径来轻易地躲过客户端的查找。

2.2.2　通信欺骗安全隐患分析

该“云查杀”客户端在上传文件信息的过程中只对上传数据进行了压缩处理，并未对其进行加密和校验，这意味着攻击者一旦进入底层就可以轻易地拦截篡改上传数据，为恶意软件躲过服务器端的检测提供可能。通过嗅探获取该“云查杀”上传的数据包分析发现，上传数据采用 HTTP 包发送，数据包中除“md5s”数据段采用压缩外，其它数据均采用明文传送，如图 3 所示，该压缩数据段以“789C”开头，因此采用 Zlib 解压分析。

图 3　“云查杀”上传的数据内容

解压分析图 3 中被压缩数据段，得到被检测文件的关键信息，这些信息并未加密且无校验值，如图 4 所示，压缩数据主要为被检测文件的 hash 值、文件大小以及文件全路径。

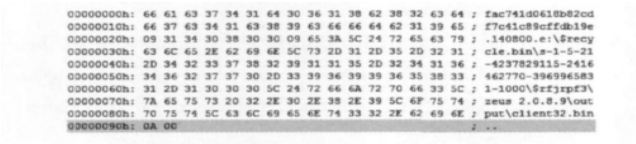

图 4　上传数据中压缩部分的内容

根据图 4 内容分析可知，“云查杀”对文件的初步判断主要仍是通过文件 hash 值的比对，因此木马可以底层拦截“云查杀”上传数据，将数据包中木马文件的 hash 值替换为正常文件的 hash 值，或者在有多条被检测文件记录的情况下直接删除木马文件的整条记录，再重新按照原数据包格式封装后发送至服务器端，从而达到躲避“云查杀”的目的。

2.2.3　断网安全隐患分析

尽管各杀毒软件厂商的“云安全”策略互不相同，但“云安全”终究依赖于互联网生存，网络一旦断开，“云查杀”就只能成为摆设。如果恶意程序运行时，网络处于断开状态，那么即使“云查杀”扫描提取到该恶意程序文件样本也无法上传至云端服务器进行分析处理。“云查杀”客户端在无法连接服务器的情况下，仍然会通过本地扫描检测生成解决方案，但由于本地病毒库并不全面，因此恶意程序可以很轻易地躲过“云查杀”软件的查杀。

3　实验验证方案

为了验证上文描述的安全隐患具有危害性，设计规避该“云查杀”的实验方案。下列方案以 PCShare 为例在该“云查杀”最新版上均测试通过。

3.1　文件路径欺骗

以上文中的文件路径欺骗安全隐患分析为理论依据，设计文件路径的隐藏方案。木马程序只需构建没有实际文件的进程即可使得杀毒软件根据启动项、系统服务等特定信息无法找到该进程实际存在的目录。首先为木马文件建立虚拟目录，使得木马在该虚拟目录里运行，再将创建的虚拟目录删除。这样，杀毒软件只能获取该木马运行的虚拟目录，根据该目录无法找到木马文件。具体实现步骤如下：

（1）首先利用 Windows 提供的 subst 命令给恶意程序实际存在的目录赋驱动器符，例如 G:，以磁盘驱动器符代替恶意程序实际存在的目录名称；

（2）打开第一步所创建的虚拟驱动器，运行其中的恶意程序；

（3）利用 subst 命令删除第一步所创建的虚拟驱动器。

3.2　“云查杀”通信欺骗

针对该“云查杀”软件对其上传文件样本未进行加密和校验的安全缺陷，设计篡改上传数据方案。该“云查杀”软件采用 HTTP 协议上传文件信息，HTTP 头部请求行包括请求方式、URL 和 HTTP 版本三项内容：

POST　/file_health_info.php　HTTP/1.1

该请求行中 URL 固定不变，可以作为数据包过滤条件，因此可以利用在用户层编写服务提供者接口（service provider interface，SPI）来 hook 系统 SPI 的方法截获修改数据包内容。首先将注册表中系统 SPI 的路径保存，编写新的具有截获篡改网络数据包功能的 SPI，再将系统 SPI 路径替换为新 SPI 路径；当有 Winsock 调用发生时，系统首先找到新 SPI 并执行，新 SPI 再加载原先保存的系统 SPI 并得到系统 SPI 函数 WSPSend，通过替换该 WSPSend 函数来完成数据封包的截获、修改与转发。WSPStartup 是 Windows Sockets 应用程序调用 SPI 的初始化函数，WSPSend 函数替换过程即在 WSPStartup 函数体内完成。

WSPStartup 函数具体实现步骤如下：

（1）根据参数 IpProtocolInfo 的协议信息栈找出服务提供者的 ID，根据 ID 取出原先保存的系统 SPI 程序的路径与文件名；

（2）Loadlibrary 加载系统 SPI，并通过 GetProcAddress 得到系统 SPI 的 WSPStartup 函数指针；

（3）通过系统 SPI 的 WSPStartup 函数得到系统 SPI 的

服务函数指针 IpProcTable 并保存；

（4）将 IpProcTable 结构中的 WSPSend 函数指针设置为新的 WSPSend 函数，在新的 WSPSend 函数中实现数据包的修改与转发。

WSPSend 函数具体实现步骤如下：

（1）以 HTTP 头部请求行中的 URL 为过滤条件，将符合条件的数据包内容保存，不符合条件的数据包调用系统 SPI 中的 WSPSend 函数直接转发；

（2）对于符合过滤条件的数据包，截取数据包中的压缩数据部分（如图 2 所示以“78 9C”开头，以“0D 0A”结尾），并采用 Zlib 解压该数据段；

（3）在解压数据中依据木马文件名查找该木马记录信息，将该条记录（以“0A”结尾）中的 hash 部分替换为正常文件 hash 值，并将解压修改后的数据重新压缩；

（4）将原数据包中的压缩数据替换为修改后重新压缩的数据，并计算数据包的数据部分长度，修改 HTTP 头部 Content-Length 值；

（5）调用系统 SPI 中的 WSPSend 函数转发修改后的数据包。

3.3 断网规避“云查杀”

以上文中的断网安全隐患分析为理论依据，设计断网规避“云查杀”方案。直接将用户网络关闭而不让用户发觉非常困难，因为直接断开网络连接，状态栏必然会提示网络连接被禁用。而且网络断开后必须要在短时间内恢复，否则必然会引起用户警觉。因此要做到短时间断网并且不提示，修改 IP 地址是个很有效的办法[10]。具体步骤如下：

（1）通过 netsh dump 命令导出当前网络设置进行备份；

（2）修改本地连接 IP 地址，例如修改为 10.1.1.111；

（3）IP 修改成功后，运行恶意程序；

（4）恶意程序成功运行后，通过 netsh-f 命令从备份文件中恢复网络设置。

4 规避“云查杀”行为的防护建议

针对文件路径欺骗行为，杀毒软件可以在获得病毒木马运行的虚拟目录后，进一步获取该虚拟目录所对应的物理路径。若无法获取物理路径，则可以根据病毒木马的文件名搜索扫描磁盘，对病毒木马的实际藏身之处进行定位。

针对“云查杀”通信欺骗行为，杀毒软件可以对上传数据进行加密，加大破解分析数据包的难度，并对数据包内容进行校验，防止恶意程序对数据包内容篡改。

针对断网规避“云查杀”的行为，杀毒软件可以对用户网络配置进行监控，对修改用户 IP 地址的行为发出提示告警；并针对“云查杀”对互联网的依赖性，加强客户端的分析处理技术，在无法连接服务器的情况下启用传统病毒查杀机制，能够对木马病毒进行有效地查杀。

5 结束语

本文针对某“云查杀”软件提出的规避方案主要是基于该“云查杀”客户端在查找文件路径和上传文件样本过程中存在的安全缺陷，这些安全缺陷产生的原因主要是由于该“云查杀”软件并未获得进程实际存在目录，并且并未对上传数据进行加密和校验。针对上述安全缺陷，本文提出的几点防护建议，对其它“云安全”检测软件也具有一定的借鉴参考作用。

参考文献：

[1] Jon Oberheide, Evan Cooke, Farnam Jahanian. Rethinking antivirus: Executable analysis in the network cloud [C]. USENIX Association, Boston, MA: 2nd USENIX Workshop on Hot Topics in Security, 2007-08.

[2] Yan W, Ansari N. Why anti-virus products slow down your machine? [C]. San Francisco, California, USA: Proceedings of 18th International Conference on Computer Communications and Networks, 2009.

[3] Rising. Rising internet security report of the first half of 2011 [R]. www.rising.com.cn/about/news/rising/2011-07-20/9802.html, 2011-07-20/2011-07-20 (in Chinese). [瑞星. 瑞星 2011 上半年互联网安全报告 [R]. www.rising.com.cn/about/news/rising/2011-07-20/9802.html, 2011-07-20/2011-07-20.]

[4] Yan W, Wu E. Toward automatic discovery of malware signature for anti-virus cloud computing [C]. Springer, Shanghai, China: First International Conference on Complex Sciences, 2009.

[5] Saurin K Shah. Exploring reliability of cloud antivirus solutions [EB/OL]. web.njit.edu/~sks36/resources/Final__Saurin__Shah.pdf, 2009/2009.

[6] Mortignoni L, Paleari R, Bruschi D. A framework for behavior-based malware analysis in the cloud [C]. Springer, Kolkata, India: 5th International Conference on Information Systems Security, 2009.

[7] Ismail Adel AL-Taharwa, Albert B Jeng, Hahn-Ming Lee, et al. Cloud-based anti-malware solution [C]. Academia Sinica, Taipei, Taiwan: The International Symposium on Grids and Clouds and the Open Grid Forum, 2011.

[8] Jon Oberheide, Evan Cooke, Farnam Jahanian. CloudAV: N-version antivirus in the network cloud [C]. USENIX Association, Boston, San Jose, CA, USA: Proceedings of the 17th USENIX Security Symposium, 2008.

[9] Igor Muttik, Chris Barton. Cloud security technologies [J]. Information Security Technical Report, 2009, 14 (1): 1-6.

[10] JIANG Xiaofeng. Research on open source project feature code anti-anti-virus and active defense avoiding [D]. Shanghai: Shanghai Jiaotong University, 2011 (in Chinese). [蒋晓峰. 面向开源程序的特征码免杀与主动防御突破研究 [D]. 上海：上海交通大学，2011.]