基于特征阈值的恶意代码快速分析方法

齐法制[1,2] 孙智慧[1]
(中国科学院高能物理研究所 北京 100049)[1] (中国科学院大学 北京 100049)[2]

摘 要 当前恶意代码具有种类多、危害大、复杂程度高、需要的应急响应速度快等特点,针对现有恶意代码分析方法难以适应现场快速分析处置与应用实践的需求的问题,研究了基于特征阈值的恶意代码分析方法,构建了恶意代码快速分析处置的具体环节,包括环境分析、文件细化、静态分析、动态分析,并通过构建的阈值判断来定位代码的功能和家族属性,并给出清除恶意代码的具体方法。实际应用结果证明,此方法对恶意代码安全特性相关的意图、功能、结构、行为等因素予以综合,实现在现场处置层面上对恶意代码安全性的分析研究,为当前网络安全恶意代码的现场快速响应和处置提供了重要支撑。

关键词 信息安全,恶意代码,现场处置,阈值分析,快速处置
中图法分类号 TP391 **文献标识码** A

Rapid Analysis Method of Malicious Code Based on Feature Threshold

QI Fa-zhi[1,2] SUN Zhi-hui[1]
(Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049, China)[1]
(University of Chinese Academy of Sciences, Beijing 100049, China)[2]

Abstract Nowadays, malicious code has many characteristics, such as multiple types, harm, high complex and needing fast response to handle it. Because the existing method for the analysis of malicious code is difficult to adapt to rapidly analyzing and disposing at the scene and the needs of application practice, this paper proposed the analysis method of malicious code based on feature threshold and constructed the details of the rapid analysis and disposal of malicious code. It contains the environmental analysis, file refinement, static analysis and dynamic analysis. By constructing the threshold determination, locating the function and family properties of code, we provided the specific method of removing the malicious code. The result of practical application proves that this method combines intention, function, structure and behavior of malicious code, and realizes the research about the analysis of the security of malicious code at the level of the disposal site. It provides important support for the fast response and disposal of the current network security of malicious code.

Keywords Information security, Malicious code, Site disposal, Threshold analysis, Rapid disposal

1 引言

恶意代码是一种通过漏洞或欺骗行为在不被察觉的情况下侵入用户电脑的程序,达到破坏被感染电脑数据、运行具有入侵性或破坏性的程序、破坏被感染电脑数据的安全性和完整性的目的。恶意代码传染的结果有浪费资源、破坏系统和一致性、数据丢失、被窃等。当今主流的恶意代码有:木马、病毒、后门、蠕虫、广告件、流氓软件、间谍软件、有害工具、风险程序。

同时,随着互联网的迅猛发展,伴随而来的网络安全局面日益复杂,网络用户受到的安全威胁日益增大,用户利益受到损害的安全事件不断增多,其中由恶意代码引发的用户信息泄漏、网银被盗、主机被控等问题层出不穷。据迈克菲发布的2012年第四季度威胁报道,恶意代码不仅在数量上大幅上升,而且呈现变种多、更加智能化的特点。这为分析工作带来了严峻挑战,主要表现在两个方面:1)如何能自动提取反映代码本质的特征,为下一步进行自动分析或人工分析提供更为全面的描述信息;2)在面对实时出现的大量未知恶意代码时,如何能更加快速地对未知样本进行判定,从而提升分析速度和处置效率。

本文针对当前未知恶意代码现场分析规范技术手段和处置手段缺乏问题,提出了针对未知恶意代码的快速分析和处置方法,构建了恶意代码快速分析处置的具体环节,包括环境分析、文件细化、静态分析、动态分析;经过具体环节分析后,得出恶意代码的全部核心行为与恶意代码的功能与最终意图,并通过构建的阈值判断来定位代码的功能和家族属性,并

本文受战略性先导科技专项(A类)(XDA100109),国家自然科学基金项目(11305196),中科院青年创新促进会(29201431231100102)资助。
齐法制(1978-),男,博士生,正高级工程师,主要研究方向为高性能计算网络、可信网络、信息安全,E-mail:qfz@ihep.ac.cn;孙智慧(1985-),男,硕士,助理研究员,主要研究方向为高性能计算网络、未来网络、信息安全。

给出清除恶意代码的具体方法；最后以具体的恶意代码案例分析对恶意代码现场分析流程进行实际应用说明。

2 相关工作

分析恶意代码的功能和危害是恶意代码分析的基础，目前主要有静态分析方法和动态分析方法。静态分析方法指不运行恶意代码而是通过文件结构分析、反汇编、反编译等方法对恶意代码的可执行文件进行分析的方法。该方法可以了解恶意代码的程序流程和功能，获取用于检测和查杀恶意代码的静态特征。动态分析方法指在一个可控的环境内运行恶意代码，分析恶意代码与运行环境之间交互行为的方法。该方法通过捕捉环境在运行恶意代码前后发生的变化，在不同层次上给出恶意代码的指令或系统调用描述，从而近似还原恶意代码实际功能。

随着恶意代码数量的急剧增加，以前主要依靠长时间机器分析的方法无法应对快速检测处置的现实需求，因此出现了一些融合实时捕获、个体分析、群体聚类或家族特征提取功能为一体的自动分析系统[1]。文献[2]总结并分析了当前主流的动静态分析工具，例如静态分析方法常用的反汇编工具 IDA 和 PE 文件解析器，可以从系统调用、API 操作、文件系统操作等角度对恶意代码进行分析。文献[3,4]等提出用 API 作为恶意代码特征的方法，即通过分析源码或者监测系统调用提取恶意代码样本的 API 集合或序列，辅以调用次数或参数构成表示样本的特征向量；同时，恶意代码当前逃避基于特征码机制的检测的机制更为成熟，因为特征库中的特征码来源于已知的恶意代码，恶意代码使用代码模糊、封装、嵌入垃圾指令等机制可轻而易举地逃避检测。例如，嵌入 non-ops指令和代码重组生成的变体可以逃避检测[5]。恶意代码的开发者已经开始使用这些技术逃避检测。

上述方法为恶意代码分析的研究做出了贡献。但是，当前恶意代码具有种类多、危害大、复杂程度高、需要的应急响应速度快等特点，现有方法难以适应现场快速分析处置与应用实践的需求。

针对上述问题，本文提出了利用特征阈值方法进行快速恶意代码分析和处置的模型，提出了针对未知恶意代码的快速分析和处置方法，构建了恶意代码快速分析处置的具体环节，分为环境分析、文件细化、静态分析、动态分析 4 步分析方法，结合快速阈值检测理论，作出病毒处置的快速响应，最终为当前网络安全恶意代码的现场快速响应和处置提供重要支撑。

3 恶意代码快速分析方法

3.1 恶意代码环境分析

在分析恶意代码之前，要对此恶意代码提取的环境进行综合分析，这是进行恶意代码分析的关键。恶意代码提取的终端环境、网络状态、业务试用期情况等信息都给分析此恶意代码带来直接影响。恶意代码环境分析主要可分为现场环境综合分析和环境重构两个部分。

通过大量的信息安全应急响应服务与恶意代码分析经验，总结出了恶意代码环境分析的分类。本文从以下 5 个部分进行分析与定位（见图 1），可以全面并快速地进行恶意代码环境的信息统计与分析。

(1)服务对象分析

是指针对服务的类别是企业、政府、金融、个人或是其它，针对不同的对象给出该对象重视的是恶意代码的哪些环节等。

(2)人员环境分析

是对要现场服务的工作人员进行调查，如平时的上网习惯、平时网络总出现什么问题等。

(3)网络环境分析

是指对现场的网络进行详细查看，例如其拓扑结构、是否具有硬件防火墙等，在关键部位进行网络监控。

(4)软硬件资源整合分析

对现场的计算机的硬件与应用软件进行统计，例如装有什么杀毒软件、病毒库日期是多少等。

(5)其它环境分析

针对其它环境周边进行分析，例如是否经常被盗、是否经常有不是本部门的其他人使用机器等。

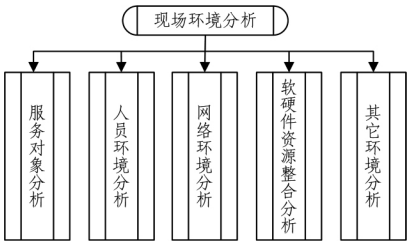

图 1 恶意代码环境分析

恶意代码环境完整性分析方法的特征在于使用动态二进制分析平台 Pin 和反编译工具 IDA 获取恶意代码执行过程中感知寻找的环境信息，并以获取的环境信息为基础进行恶意代码最佳执行环境的构建，最后将恶意代码投入动态构建的环境中运行实现完整性分析。采用粗粒度环境数据提取与细粒度敏感环境识别相结合的方式，共同确定恶意代码隐藏行为触发所需要的执行环境，其具体执行步骤如下。

步骤 1 恶意代码输入，开始进行行为完整性分析，分别执行步骤 2 和步骤 8；

步骤 2 使用动态分析方法提取恶意代码执行过程中使用的字符串型环境数据，并对使用字符串的函数类型进行判别并确定字符串环境类型，同时对恶意代码进行脱壳处理；

步骤 3 使用静态分析方法提取脱壳代码中的静态字符串信息，并对各字符串的引用位置及方式进行分析并确定字符串环境类型；

步骤 4 整合步骤 2 和步骤 3 中提取出的环境数据，填充环境数据提取信息库；

步骤 5 使用决策树分析环境数据提取信息库中环境类型为其它的数据，确定数据的环境类型；

步骤 6 对环境数据提取信息库中的数据进行冗余删减、负面数据去除处理，获得构建环境所需的最终数据；

步骤 7 使用步骤 6 中获得的数据从预先搭建的虚拟机镜像群中选择合适的镜像作为恶意代码基础执行环境，转步骤 11；

步骤 8 采用基于动态二进制分析平台 Pin 的数据流跟

踪程序，跟踪记录环境敏感数据的传播过程；

步骤 9　判断分支点是否属于环境敏感分支点，如果是，则提取分支靶项并执行下一步操作；否则继续跟踪程序执行；

步骤 10　执行分支靶项回溯，获取与之对应的环境敏感 API 函数规则；

步骤 11　解析 API 模式规则提取靶项类型信息，如果靶项类型为二元靶项，则确定强制修改分支条件取值的位置与方式；如果靶项类型为三元靶项，则获取需要添加的环境数据；

步骤 12　继续跟踪程序执行流程，判断程序是否结束，如果程序结束，则执行下一步操作；如果遇到分支点，转步骤 9；

步骤 13　使用步骤 11 中获取的环境数据与配置文件完善步骤 7 中构建的基础环境，获得恶意代码最佳执行环境；

步骤 14　将恶意代码置入步骤 13 中构建的执行环境，启动恶意代码行为分析；

步骤 15　解析分析结果，获得恶意代码完整性分析结果。

3.2　静态代码分析

静态分析是指在尽量不运行调试恶意代码的前题下，尽可能多地收集此恶意代码的相关信息，以找出此恶意代码的功能与作用或为动态分析找出恶意代码的功能做准备。静态分析主要分为两个方面：静态常规信息分析和静态代码分析。

静态常规分析主要查看文件的文件属性，可以对文件的正常程度加权，加权方式可以效仿阈值检测部分，以达到对恶意代码的属性信息等进行分析。每种程序根据自己的功能类型、版权所属或者个性特色，都会使用醒目的图标来区分自己，而盗用文件的图标也是迷惑用户的一个重要方式，利用用户对系统文件不熟悉、害怕进行修改的心理弱点，使用系统文件的图标让用户不敢轻易改动，因此使得恶意代码能够长期驻留在系统中。在恶意代码的程序形态特性分析过程中，会对文件名欺骗特性、文件位置特性、文件版本特性、数字签名等进行详细分析和验证。

同时，在不需要运行病毒的情况下，尽量多地获取 API、字符串、原始文件信息识别等。主要有如下几个技术手段。

通过反汇编工具对静态的 PE 文件进行反汇编：反汇编后可以通过分析反汇编后的文件中动态加载的 API，来判断此样本的基本功能和特点。

API 操作特性：恶意代码往往调用一些特殊的 API 来实现其非法功能，通过对其 API 函数的调用，能大致分辨其所具备的功能。通常带有恶意行为的程序都会调用一些特定的 API 来组合出其要实现的功能作用，比如木马要进行网络连接，就必定调用网络 API 中的 recv 和 send 函数。

API 数量特性：恶意代码为了压缩或者保护自己，通常采用特殊隐藏处理，这个处理是隐藏文件内部资源，主要是 API 函数的调用，阻止分析人员对其进行反汇编而暴露自身。经过这种处理后，使用反汇编程序查看程序内部资源时，往往只有少数几个 API 能被显示出来，除非有一定的原因才需要对资源做这样的保护。

部分恶意代码常用 API 如表 1 所列。此处列举的 API 包括写注册表、遍历文件、结束进程、获得当前路径、文件操作等，这些 API 在正常文件中有时也会出现。

表 1　恶意代码常用 API

注册表操作 API	RegCreateKey　创建注册表键 RegCloseKey　关闭注册表键 RegDeleteValue　删除键值 RegEnumKey　列举注册表键值 RegOpenKey　打开注册表键 RegQueryValue　查询注册表键 RegSetValue　设置注册表键值 …
敏感操作 API	GetActiveWindow　获得当前窗口 keybd_event　制造键盘敲击事件 OepnService　打开服务 RemoveDirectory　删除目录 SetSystemTime　设置系统时间 …

3.3　动态代码分析

简单的静态分析可能不足以将恶意代码感染、运行的全过程进行重放，分析人员也仅只能停留在表面现象，而不能知晓其中的真正操作、调用，需要一种能够对恶意代码进行更接近驱动级的监控工具来辅助。所以一定要在恶意代码分析实验室中进行深入的动态分析，来一步一步地剖析出恶意代码的各个功能。动态分析主要分为两个部分处理：行为分析与动态代码分析。

行为分析指的是从外部观察病毒运行后对系统所做出的各种改动的一种表象总结，由于病毒运行必然对系统做出相应改动，因此可针对病毒行为进行修复，对于一般恶意代码行为分析可按以下步骤进行：

(1)注册表监控：对注册表的修改进行动态分析，对于每一个键值的操作(包括查询、访问、修改、删除等)进行详细分析。

(2)文件监控：对文件写入硬盘或删除等进行实时监控，动态地分析一个文件是如何被创建、运行、结束的。特别关注 \winnt(\windows)、\winnt\system32 等几个系统的核心位置。

(3)网络监控：对网络端口、传输文件等进行实时分析。

(4)进线程监控：对进程与线程的创建与消亡进行监控。

(5)API 跟踪：动态地跟踪 API 函数，来获取有用的信息。API 函数是 Windows 下编程实现文件功能的重要工具，恶意代码也是通过对 API 函数的调用和组合，来完成其对文件的管理操作、对注册表的编辑、对信息的截取，以及网络行为的发生。

(6)消息相关分析：判断是否存在消息钩子等。Windows 应用程序的运行模式是基于消息驱动的，任何线程只要注册了窗口类都会有一个消息队列来接收用户的输入消息和系统消息。为了取得特定线程接收或发送的消息就要用到 Windows 提供的钩子。钩子(Hook)是 Windows 消息处理机制中的一个监视点，应用程序可以在这里安装一个子程序(钩子函数)以监视指定窗口某种类型的消息，所监视的窗口可以是其他进程创建的。当消息到达后，在目标窗口处理函数处理之前，钩子机制允许应用程序截获它进行处理。病毒作者利用此技术截获相关 API 函数来实现某种目的。例如：截获用户的鼠标和键盘消息来窃取用户敏感信息；隐藏进程、进程保护、获取窗口标题相关信息等。

对某些恶意代码，如需详细了解则可对其进行动态代码分析。因为有时恶意代码的某些功能在特定条件下才会得到激活。这种情况下行为分析有时不能激活，往往会漏掉这部分的分析；通过动态代码分析找到激发条件，然后模拟该激发条件，使恶意代码的特定条件得以满足，从而捕获相关行为。

所以动态代码分析是恶意代码分析的最关键部分，也是其核心部分。

分析时需要在恶意代码分析实验室中利用 Ollydbg 等工具动态加载，逆向反汇编跟踪调试分析，如表 2 所列，用调试器载入恶意代码后，在恶意代码的各个可疑的地方下断点，根据代码来确定此恶意代码的有害操作。

表 2　恶意代码逆向反汇编跟踪调试分析

```
0043673A   8945 FC
MOV DWORD PTR SS:[EBP-4],EAX
0043673D   E8 C8FFFFFF
CALL v.0043670A
////以下逐页比较验证，找到 Kernel32 的基地址
0043676D 8B41 18 MOV EAX,DWORD
PTR DS:[ECX+18]
////以 NumberOfName 的值做循环，查找想要的函数入口地址
004368BC   FF53 30
```

3.4　快速阈值检测

阈值是用来区分一个文件是否为正常文件或者是恶意代码的指标。给定文件的某些特殊行为和特征一定的分值，最后综合得出所有被考核项的分值总和，将其与阈值进行比较，当超过阈值时，就报警提示该文件可能为恶意代码，以此来进行未知恶意代码的检测。阈值检测标准包含：1）文件名分析结果；2）文件使用系统图标；3）API 数量异常；4）文件在系统路径下等 16 项主要分析项目及其结果。

下面采用德尔菲法模型对阈值检测的 16 项标准进行打分。对于同一层次的各元素，以相邻上一层支配元素为准则使用德尔菲法，专家参照如表 3 所列的 Satty 9 级分制两两比较相对重要性。

表 3　Satty 9 级分制

标度	含义
1	两因素相比，同等重要
3	两因素相比，前者比后者稍重要
5	两因素相比，前者比后者明显重要
7	两因素相比，前者比后者强烈重要
9	两因素相比，前者比后者极端重要
2,4,6,8	表示上述判断的中间值
倒数	相应两因素交换次序比较的重要性

快速阈值过程主要包含单个事件定义、事件集的定义和系统环境定义 3 个过程。其中，单个事件定义主要对事件的危害性进行量化定义，事件集定义是对系统环境中可能出现的事件进行分级，而系统环境定义则选择不同级别的事件完成各个事件情景的原子任务。

得出相对权值的比值$\frac{\omega_i}{\omega_j}$，以此建立一个判断矩阵，$A$ 为 $n\times n$方阵，主对角线为 1，其具有性质：对 $\forall i,j\in N\Delta\{1,2,\cdots,n\}$，有

$$a_{ij}>0,a_{ij}=\frac{1}{a_{ji}},a_{ii}=1 \tag{1}$$

层次单排序及一致性检验。对每一个判断矩阵求解最大特征根 λ_{max}及对应的最大特征向量 w，进而得出针对上层某一准则的各元素相对权重，之后做一致性检验。根据需要，本文采用和积法计算 λ_{max}与 w 的近似值。

1）对 A 按列规范化：

$$\overline{a_{ij}}=\frac{a_{ij}}{\sum_{i=1}^{n}a_{ij}}(i,j=1,2,\cdots,n) \tag{2}$$

2）将规范化后的判断矩阵按行相加：

$$\widetilde{\omega_i}=\sum_{j=1}^{n}\overline{a_{ij}}(i=1,2,\cdots,n) \tag{3}$$

3）对向量 $w=\{\widetilde{\omega_1},\widetilde{\omega_2},\cdots,\widetilde{\omega_n}\}^{\mathrm{T}}$ 规范化：

$$\omega_i=\frac{\widetilde{\omega_1}}{\sum_{i=1}^{n}\widetilde{\omega_i}} \tag{4}$$

则 $w=\{\widetilde{\omega_1},\widetilde{\omega_2},\cdots,\widetilde{\omega_n}\}^{\mathrm{T}}$ 即为最大特征向量的近似值。

4）利用最大特征向量求最大特征根的近似值：

$$\lambda_{max}=\frac{1}{n}\frac{(AW)_i}{\omega_i} \tag{5}$$

其中，$(AW)_i$ 为矩阵 A 与乘积向量的第 i 个元素。

当因素较多时，由专家经验给出的两两比较的主观判断矩阵计算的最大特征向量与最大特征根可能不一致，这需要由 λ_{max}检验 A 是否存在严重的非一致性。为此，使用 Satty 定义的一致性检验指标 CI：

$$CI=\frac{(\lambda_{max}-n)}{(n-1)} \tag{6}$$

其中，n 为 A 的阶数，CI 越小，说明一致性越高。为判断矩阵是否具有满意的一致性，可将 CI 与平均随机一致性指标 RI 相比，得到随机一致性比率 CR：

$$CR=\frac{CI}{RI} \tag{7}$$

未知恶意代码检测标准是归纳绝大多数恶意代码的静态属性和动态行为，以及对每种行为制定相应的危害数值。在对可疑文件进行考核时，严格对照以上策略标准，记录每一项的分数，最后综合得出该文件的参考分数，并与阈值进行比较，如果超过该阈值，则可以判断该文件的恶意行为将会对系统安全构成一定威胁，从而确定该文件为一个可疑文件，然后提取恶意代码样本进行分析，得出处置方案。

4　恶意代码分析应用与处置

整个过程主要包括 5 个步骤：第 1 步为恶意代码环境评估，主要定义恶意代码快速分析需要的评估值，或者需要分析的相关内容；第 2 步的静态分析和第 3 步的动态分析综合起来就是对恶意代码进行定义，描述恶意代码的行为、特征及危害；第 4 步的快速阈值检测，实际上就是对恶意代码的最终属性进行定义；第 5 步为恶意代码处置，利用分析得出综合分析结论，基于分析过程进行恶意代码风险抵抗。

4.1　分析应用实例

当文件中嵌入 PE 程序时，很有可能这是一个能够自解压或者自身能够释放出体内可执行程序的程序。将真正的恶意代码体隐藏在外部文件中，这样能躲避杀毒软件的监控扫描。当该外部程序传播到目的地后，自行释放并运行体内的 PE 程序，达到病毒传播运行的目的。

本文使用 Satty 给出的 1-12 阶 RI 参考值，如表 4 所列。

表 4　随机一致性指标 RI 参考值

n	1	2	3	4	5	6
RI	0	0	0.58	0.90	1.12	1.24
n	7	8	9	10	11	12
RI	1.32	1.41	1.45	1.49	1.51	1.54

表 4 列举了阈值检测的 16 项标准，下面对这 16 项标准

（下转第 367 页）

测控技术,2013,32(8):137-140

[11] 曹嘉容,王瑛,支乐.基于二元语义与灰色关联决策的改进FMEA方法[J].空军工程大学学报(自然科学版),2014(5):92-95

[12] Chang K H,Chang Y C,Wen T C,et al. An innovative approach integrating 2-tuple and LOWGA operators in process failure mode and effects analysis [J]. International Journal of Innovative Computer Information & Control,2012,8(1):747-761

[13] Fuller R,Majlender P. An analytic approach for obtaining maximal entropy OWA op erator weights [J]. Fuzzy Sets & Systems,2001,124(124):53-57

(上接第 345 页)

进行打分,该打分是经过 XX 公司多年的应用总结出的特别精确的判定分数,如表 5 所列。

表 5 阈值检测 16 项标准打分值

sp_id	sp_score	sp_expr
26112	30	文件名有数字前缀
26113	30	文件使用系统图标
26114	60	API 数量异常
26115	20	文件在系统路径下
26116	30	文件无版权信息
26117	10	文件长度过大
26118	60	文件有多扩展名
26119	10	有非正常节名
26120	70	处于启动项中
26121	30	文件最近创建
26122	50	仿造系统文件名
26123	30	文件使用加壳
26124	50	文件监听端口
26125	40	文件包含恶意 API
26126	20	文件中包含邮件信息
26127	50	嵌入 PE 程序
36866	150	阈值

该方法应用时主要面向实际系统中需要高响应速度、高准确率的恶意代码分析。分析时的样本数据集用于特征向量空间构建,主要包含白样本和黑样本,其中,白样本集主要从 Windows 系统提取相关文件,黑样本集主要通过反病毒厂商样本交换及网络蜜罐捕获积累的样本,用作快速恶意代码分析的参考空间和佐证。

实际 Backdoor. Win32. IRCBot. bad 恶意代码快速分析过程如下。

(1)恶意代码环境

病毒名称为 Backdoor. Win32. IRCBot. bad,病毒类型为后门类,文件 MD5 为 87A7C70B7295C5CE90E171447243E875,公开范围为完全公开,危害等级为 3,文件长度为 43520 字节,感染系统为 Windows98 以上版本,开发工具为 Microsoft Visual C++,加壳类型为 PECompact V2. X-> Bitsum Technologies。

(2)静态分析

观察文件,该文件为一个 43kB 的 exe 文件。该文件没有版本信息,文件属性为系统隐藏,文件路径为系统备份文件目录。文件创建进程、开启服务、打开端口、连接网络。初步分析为一个后门程序。用 autoruns 查看其创建的服务。

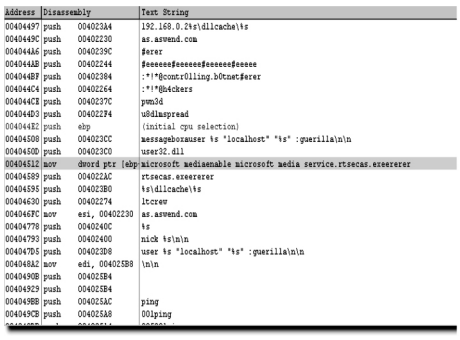

图 2 恶意代码静态分析

从恶意代码程序的字符串中可以看到恶意代码释放文件的路径,创建服务名称及其描述,如图 2 所示。

(3)动态分析

////病毒运行后把自身拷贝为%System%\dllcache\Rtsecas. exe,删除自身,设置文件属性为系统、隐藏、只读。

0040460B 8D85 E4FEFFFF LEA EAX,DWORD PTR SS:[EBP-11C]

00404611 6A 07 PUSH 7;参数:隐藏、系统、只读

00404613 50 PUSH EAX

00404614 FF15 60204000 CALL DWORD PTR DS:[<&kernel32. S>];调用 SetFileAttributesA 设置文件属性为系统、隐藏、只读。

该病毒为后门类,病毒运行后把自身拷贝为%System%\dllcache\Rtsecas. exe,并删除自身;创建名为 Microsoft Media 的服务,链接 IRC 服务器接受控制,创建线程扫描本地网络,并对其进行攻击。

4.2 恶意代码处置

先对恶意代码进行非感染式恶意代码手动处置,然后再对感染的正常文件进行程序清除,但很容易在手动处置的同时运行已被感染的正常文件。因此针对感染式恶意代码一般不进行手动处置,而是直接根据感染式恶意代码的分析,编写程序进行程序处置。

程序的编写除了进行非感染式恶意代码程序处置部分,还要编写清除模块。清除模块包括两个最主要的部分:感染式特征匹配和针对此恶意代码编写修复模块。

结束语 本技术的工作流程与实现技术面向恶意代码分析的实际需求,研究了进行现场快速分析和处置恶意代码的方法,构建了环境分析、文件细化、静态分析、动态分析恶意代码快速分析处置的具体模型,结合快速阈值检测理论,达到了基于快速阈值分析适合现场恶意代码快速分析的和处置的目的。在此基础上,实现了对笔者所在单位遇到的恶意代码的实例分析。

参 考 文 献

[1] Branco R R,Shamir U. Architecture for automation of malware analysis[C]// The 5th International Conference on Malicious and Un-wanted Software (MALWARE). 2010:106-112

[2] Egele M,Scholte T,Kirda E,et al. A survey on automated dynamic malware-analysis techniques and tools[J]. ACM Computing Surveys (CSUR),2012,44(2):1-42

[3] Sathyanarayan V S,Kohli P,Bruhadeshwar B. Signature generation and detection of malware families[C]// Information Security and Privacy. 2008:336-349

[4] Staish S,Pereira S. Behavioral Signature Generation Using Clustering:WIPO Patent 2011137083[P]. 2011

[5] http://www. m86security. com/newsimages/trace/Marshal8e6__TRACE_Report_July_2015. pdf