doi:10.3969/j.issn.1002-0802.2017.01.020

面向威胁情报的攻击指示器自动生成*

徐文韬，王轶骏，薛 质

（上海交通大学 电子信息与电气工程学院，上海 200240）

摘 要：为了应对更加复杂的攻击，如何便捷地分享交流安全情报成为针对特定攻击目标检测、响应和防止的关键问题。基于国内外共享的威胁情报和 OpenIOC 框架，实时获取国内外海量共享的威胁情报数据，对其进行爬取、解析和分类，通过基于沙箱的恶意代码分析平台进行威胁行为的检测分析，最后结合机器学习算法自动生成可机读、共享的 IOC 文件，从而对最新广泛流行的攻击行为做出快速响应。

关键词：威胁情报；共享；机器学习；分析平台

中图分类号：TP393.08 **文献标志码**：A **文章编号**：1002-0802(2017)-01-0116-08

Indicator Autogeneration of Compromise Oriented to Threat Intelligence

XU Wen-tao, WANG Yi-jun, XUE Zhi

(School of Electronic Information and Electrical Engineering, Shanghai Jiaotong University, Shanghai 200240, China)

Abstract: How to deal with more complex attack and easily share the security information now becomes the key point of detection , response and prevention of specific target. Based on the threat intelligence and OpenIOC framework, real-time access to and analysis on the massive threat intelligence data both at home and abroad are done with the sandbox malware analyzer named cuckoo. Finally by machine learning algorithm, the indicator of compromiseis automatically generated, which can be shared and machine readable. And it is thus possible to make guick response to the latest and most popular attacks.

Key words: threat intelligence; share; machine learning; malware analyzer

0 引 言

近年，“威胁情报”成为信息安全界被提及最多的词汇之一[1]。早在 2013 年，Gartner 就对威胁情报做出了定义：“威胁情报是基于证据的知识，包括场景、机制、指标、含义和可操作的建议，这些知识是关于现存的、即将出现的针对资产的威胁；或者是危险的、可为主体响应相关威胁提供决策的信息。[2]”

如今，APT 攻击的泛滥使传统设备和方案很难对抗不断升级、持续变化的攻击手段，而威胁情报在防御过程中会不断被收集、分析、丰富，并在此基础上不断扩大，以动态的方式来对抗攻击者。例如，僵尸网络地址情报（Zeus/SpyEye Tracker）、0day 漏洞信息、恶意 URL 地址情报等。这些情报对于防守方进行防御很有帮助[1]。然而，传统的威胁情报共享方案主要通过人工收集信息并加以分析，然后采用特定的技术规范进行描述，形成书面的报告分发共享给其他人员，时延非常高[3]。因此，如何便捷地分享交流威胁情报，成为针对特定目标攻击检测、响应和防止的关键问题。

本文结合威胁情报共享和 OpenIOC 框架，通过恶意软件自动化分析平台，分析目标并提取特征，运用机器学习算法，自动生成可机读的攻击指示器（Indicator of Compromise，IOC），实现了威胁情报

* 收稿日期：2016-09-05；修回日期：2016-12-09 Received date:2016-09-05;Revised date:2016-12-09

便捷、快速的共享。

1　系统框架

“OpenIOC”作为现实可用的威胁情报共享规范，是一个记录、定义以及共享威胁情报的格式。它可以借助机器可读的形式，实现不同类型威胁情报的快速共享[3]。

1.1　威胁情报共享框架 OpenIOC

首先，需要明确以下几个定义：

（1）表达式（Expression）：定义了一个条件，当为真值时，表明存在一个入侵行为；

（2）简单表达式（Simple Expression）：没有使用 AND 或 OR 两种逻辑运算符的表达式；

（3）复杂表达式（Complex Expression）：多个简单表达式通过 AND 或 OR 连接；

（4）攻击指示器（IOC）：多个表达式的连接，可以是简单表达式，也可以是复杂表达式[4]。

IOC（Indicator of Compromise） 是 MANDIANT 在长期的数字取证实践中定义的、可以反映主机或网络行为的攻击指示器。它使用XML进行描述(XML 提供了灵活、丰富的格式，将数据表示成可机读的形式）。OpenIOC 是一套威胁情报共享的标准，通过该标准，可以建立 IOC 的逻辑分组，从而实现威胁情报的交流共享。

在使用 OpenIOC 时，通常会定义自己的指示器属性。这些属性由 OpenIOC 提供，总共包含了 27 个大类，每一类又包含了很多具体的属性。同时，OpenIOC 还支持用户以标准的格式自定义属性，从而实现了属性的扩展。每个 IOC 都是一个复合指示器，是由多个指示器组合到一起形成的。因此，IOC 实质上就是一个复合表达式。当表达式的值为真时，则该IOC命中。一个 IOC 的例子，如图 1 所示。

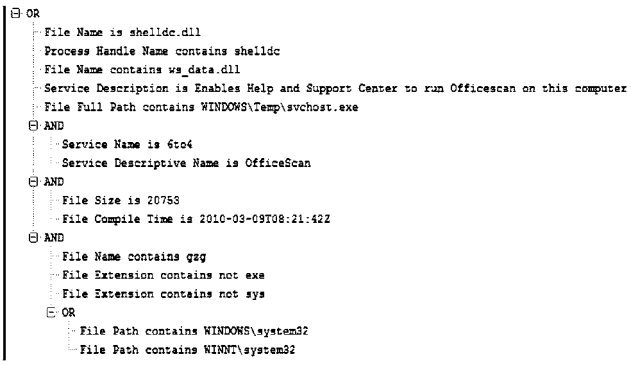

图 1　攻击指示器例子

如图 1 所示，IOC 的顶级逻辑为 OR 运算，下层的每个逻辑运算符 AND 或 OR 只作用于其子元素。每个 IOC 表达式的条件有 containslcontains notlislis not 四种，刻画的细度主要通过设置不同的属性（IndicatoreItem）来实现。通过模块化的逻辑结构，可以随时根据获得的信息和知识进行 IOC 的调整和优化[4]。

本文的主要目标是对给定的一些恶意软件样本，可以自动生成它们的 IOC 描述，从而快速进行威胁情报共享。

1.2　系统模块

IOC 自动化生成平台主要分为资源获取模块、分析模块、特征提取模块和 IOC 生成模块。整体框架如图 2 所示。

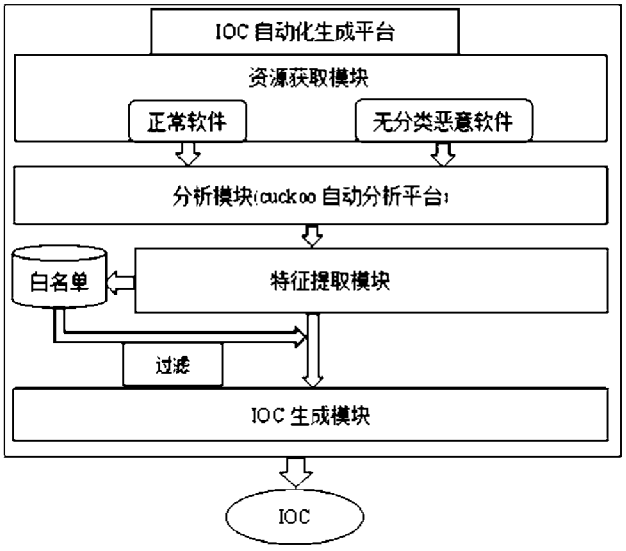

图 2　OC 自动化生成平台的系统框架

1.3　详细流程

IOC 自动化生成平台的详细工作过程如下：

（1）资源获取模块使用网络爬虫，从正规的软件下载中心下载合法的常用软件，同时从威胁情报共享网站获取恶意软件样本，并将这些获取到的资源分别投入到分析模块。

（2）分析模块对这两类资源分别进行分析，并将分析的结果投入到特征提取模块。

（3）特征提取模块对两类资源的分析结果分别采取以下方案进行处理：

①对正常软件的分析结果提取和统计特征，生成白名单。

②对恶意软件的分析结果，先提取特征，然后通过白名单过滤掉正常行为特征，得到恶意行为特征集合，最后将提取到的恶意特征投入 IOC

生成模块。

（4）IOC 生成模块处理特征，生成 IOC 复合指示器。通过聚类算法，将这些特征生成几类特征簇；对于每个特征簇，使用关键词树算法生成逻辑树；根据逻辑树每个节点的特征的描述，选取对应的指示器标签，调用 OpenIOC 的 ioc_writer 模块，生成攻击指示器 IOC。

2 自动化分析平台

本文在第 1 章提出了 IOC 自动生成平台的框架，其中分析平台主要基于 Cuckoo 恶意软件自动化分析平台。以下将对 Cuckoo 恶意软件自动化分析平台进行相关介绍。

2.1 Cuckoo 恶意软件自动化分析平台

Cuckoo 是一款开源的自动化恶意软件分析系统 [5]，目前主要用于分析 windows 平台下的恶意软件。Cuckoo Sandbox 已经于 2010 年开始成为谷歌代码之夏的项目之一，其框架同时支持 Linux 和 Mac OS[6]。

Cuckoo 能够自动获取如下信息：

（1）跟踪恶意软件进程及其产生的所有进程的 win32 API 调用记录 [7]；

（2）检测恶意软件的文件创建、删除和下载；

（3）获取恶意软件进程的内存镜像；

（4）获取系统全部内存镜像，方便其他工具进行进一步分析；

（5）以 pacp 格式抓取网络数据；

（6）抓取恶意软件运行时的截图 [8]。

Cuckoo 系统的架构如图 3 所示。

Cuckoo Host 负责 Guest 端与分析报告的管理，包括提交任务、打开合适的虚拟机进行分析、生成分析报告等。Guest 端通常是一个干净的系统，负责实际运行分析 Host 端提交的样本。Host 与 Guest 通过虚拟网络建立连接，所以 Cuckoo 的运行需要至少一个虚拟化环境。目前，cuckoo 能够支持 vmware、virtualbox、kvm、qemu、xen、avd 等主流虚拟化平台 [5]。

Cuckoo 支持分析多种文件格式，包括 windows 可执行文件、DLL 文件、PDF 文档、Office 文档、恶意 URL、HTML 文件、PHP 文件、CPL 文件、VBS、ZIP 压缩文件、jar 文件、python 程序等。它的架构高度模块化，只要添加不同的分析模块，Cuckoo 就能够完成不同系统平台下的分析工作 [5]。

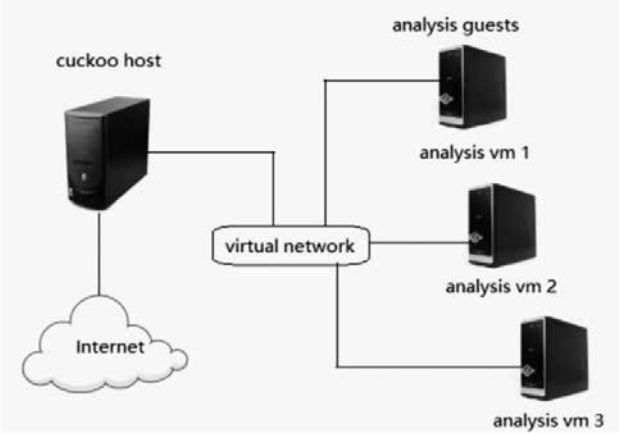

图 3 Cuckoo 系统框架

2.2 分析报告

Cuckoo 生成的数据包括本地功能与 Windows API 调用追踪、被创建及被删除的文件副本以及分析机内存转储数据。用户可以对该项目的处理与报告机制进行定制，从而将报告内容生成不同格式，包括 JSON 与 HTML[8]。

当一个样本通过 Host 添加到任务列表中后，Host 会根据样本运行的系统平台，启动相应的空闲虚拟机进行分析 [5]。虚拟机在分析完样本后，会将分析结果通过虚拟网络传回给 Host，分析结果记录在 report 文件夹下，同时会导入到数据库。report 的结构如图 4 所示。

Cuckoo 提供了一份 html 格式的报告，可以使用 web 浏览器打开，如图 5 所示。

图 4 Cuckoo 分析结果结构

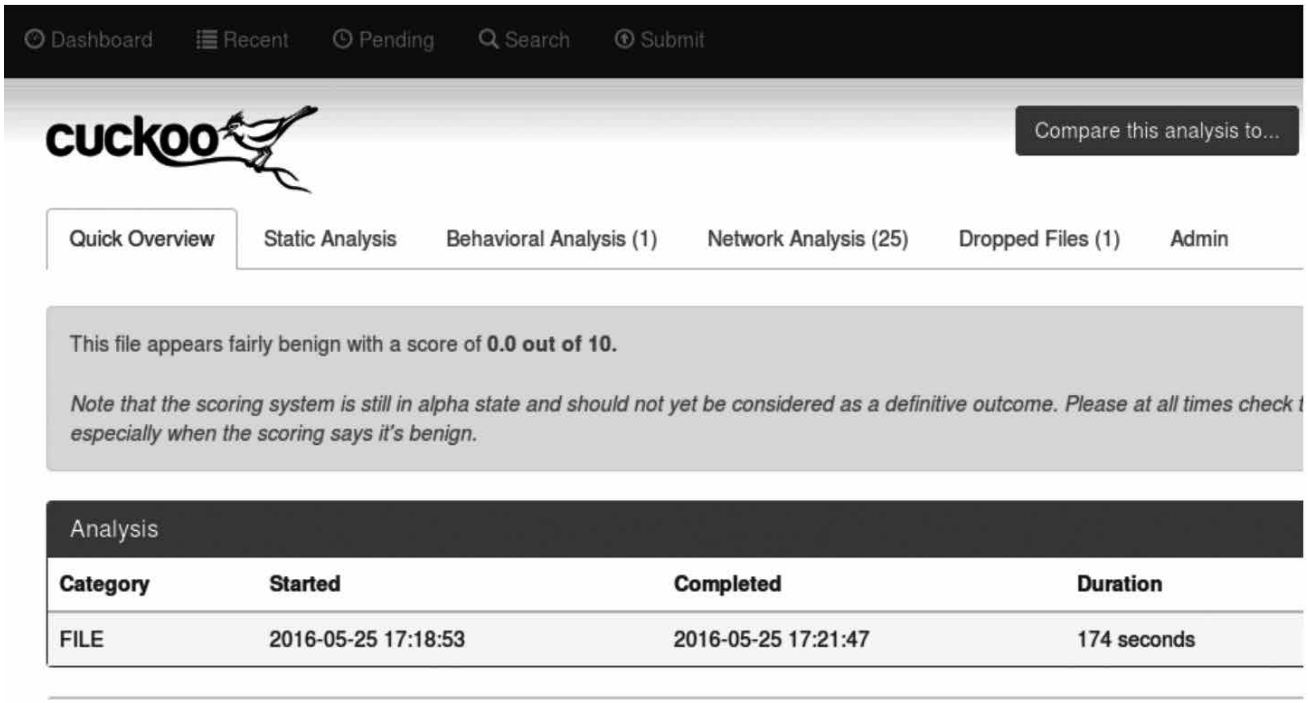

图 5　Cuckoo 分析报告

3　特征提取与 IOC 生成

3.1　特征提取

通过分析报告，可以很直观地获取恶意软件的一些静态特征，如文件名、MD5 值、文件编译的时间、调用哪些 dll 等。这些静态特征可以直接从分析报告中提取。

3.1.1　行为操作

恶意代码与正常程序的区别在于其运行后会对计算机及网络系统产生危害[9]。因此，除了与正常程序相近的行为操作外，对于可能会对系统产生危害的恶意行为应给予重点关注[10]。这些恶意行为操作往往依赖操作系统的函数调用，并读取、修改、操作系统的相关资源对象，如文件、进程、网络、注册表[11]。因此，定义原子操作为一个五元组：

$$\alpha=\{H,I,P,A,O\} \quad (1)$$

其中：H 为恶意代码的 MD5 Hash，用作该恶意软件的唯一标识[11]；I 为操作方式标识，包括 file、process、network、registry、directory 等；P 为操作进程标识；A 为操作行为标识；O 为操作对象标识。具体的，部分操作标识符如表 1 所示。

Cuckoo 生成分析报告后，通过文本分析，从 report.json 中提取相关信息，并按照上述行为操作的描述，将其转化为特征五元组的形式。这样，每个样本都会得到一个行为特征集合：

$$\theta=\{\alpha_1,\alpha_2,\cdots\alpha_n\} \quad (2)$$

表 1　操作标识符

操作方式标识 I	操作行为标识 A	
File	读文件	file_read
	写文件	file_written
	打开文件	file_opened
	创建文件	file_exists
	创建文件夹	directory_created
	删除文件	file_deleted
Process	加载 dll	dll_loaded
	创建进程	process_created
	杀死进程	process_killed
Registry	打开注册表	regkey_opened
	查询注册表	regkey_read
	删除注册表	regkey_deleted
	写入注册表	regkey_written
Network	DNS	dns
	User agent	user_agent
	Host	host_ip

需要说明的是，样本的动态特征由这一系列五元组进行描述。

3.1.2　白名单

通过前期工作，所提取到的特征中可能包含了大量正常的、无用的特征[12]，造成特征操作集合规模过大，不利于之后的整理工作。为了提高 IOC 的准确度和效率，需要对这些特征进行过滤、筛选。本文采用白名单的方法实现特征的过滤。

通过网络爬虫，在如常用软件下载中心等网站

爬取下载一些正常软件。通过 cuckoo 自动化分析平台的检测后，将通过的软件作为白名单的生成样本。将这些样本的分析报告按照上述方法提取特征集合：

$$\{\theta_1,\theta_2,\cdots,\theta_n\} \quad (3)$$

并求这些特征集合的并集，得到白名单：

$$W=\theta_1 \cup \theta_2 \cup \theta_3 \cup \cdots \theta_n \quad (4)$$

在对恶意软件提取到的特征进行分析前，首先通过白名单进行过滤：

$$\theta=\theta \cap W \quad (5)$$

这样，大量无用的特征就会被过滤掉，从而留下恶意性较高的特征。

3.2 聚类算法

大多数威胁情报共享网站只是提供了一个最新发现的威胁情报汇总的展示平台，然后由用户提交潜在的威胁，网站做出检测后，生成报告并将样本共享。以某网站为例，每天会更新几百上千条信息，包含了一些基本信息和样本下载。这些威胁情报可能毫无联系，也可能存在很多相同或者相似的类型，如同源恶意代码、恶意代码家族等。如果对每一条威胁情报都生成 IOC，工作量巨大，而生成 IOC 之间也必然存在许多重复描述，达不到便捷、快速共享的目的。研究发现，恶意软件表现出的恶意行为存在一定的规律性 [13]。因此，需要寻求一种方案，能够将相似较高，具有同源性、家族性的恶意软件聚合到一起，对某一类恶意软件生成 IOC 描述，从而提高效率，同时挖掘到威胁情报背后的联系。

通过特征提取模块提取恶意软件的特征后，本文采用 DBSCAN 聚类算法，对这些样本进行聚类，形成具有较高相似性的类簇。

DBSCAN 是一种基于密度的聚类算法。该算法能将具有足够高密度的区域划分为簇，并在具有噪声的空间数据库中发现任意形状的簇。它的优势在于其聚类时不需要事先指定聚类的簇数，聚类时可根据样本之间的差异性，将相似的样本聚合成一类 [14]。

这里使用 Jaccard 系数作为相似性的度量：

$$J(A,B)=\frac{|A\cap B|}{|A\cup B|} \quad (6)$$

$$d_J(A,B)=1-J(A,B) \quad (7)$$

Jaccard 相似系数被定义为两个集合的交集个数除以并集个数，而两个集合之间的 Jaccard 距离则为 1 减去 Jaccard 系数，且 Jaccard 系数具有以下特性：① $0 \leqslant J(A,B) \leqslant 1$；② A、B 都为空集，则 $J(A,B)=1$[15]。

将 $d_J(A,B)$ 作为相似性度量用于 DBSCAN 算法进行聚类。将相似性较高的恶意代码聚合到一起，便于提取共同特征，生成 IOC。

主要步骤如下：

（1）设置最小密度（min_pts），扫描半径（eps）；

（2）对于待测特征集合中尚未检查过的对象 θ_i，如果未被处理过（归为某个簇或者被标记为噪声），则检查其邻域（半径为 eps）；若其邻域中包含的对象数大于 min_pts，则建立新簇 C，并将其中的所有点加入候选集 P；

（3）对于数据集 P 所有尚未被处理的对象 q，检查其邻域，若至少包含 min_pts 个对象，则将这些对象加入 P；如果 q 未归入任何一个簇，则将 q 加入 C；

（4）重复步骤（3），继续检查 P 中未被处理的对象，直到当前候选集 P 为空；

（5）重复（2）~（3），直到所有对象都被归到某个簇或者被标记为噪声 [14]。

通过 DBSCAN 聚类算法，可以将经过分析的样本聚类成具有相似特征的样本簇，并对每个样本簇生成一条 IOC 描述。

3.3 IOC 生成

3.3.1 属性对照

OpenIOC 提供了一系列的指示器属性标签。为了能够使用 IOC 描述一个样本簇，需要将上述特征描述转化为与 OpenIOC 指示器属性，部分操作行为标识与指示器属性的对应关系，如表 2 所示。

表 2 IOC 属性与特征标识符

File FileName Accessed	file_read
File FileName Modified	file_writed
File FileName Created	file_exsit
Registry Path Accessed	regkey_opened
Registry Key Path Modified	regkey_writed
Registry Path Deleted	regkey_deleted
Network DNS	dns
Network User Agent	user_anget

OpenIOC 自身有 27 个大类属性，每个大类又包含许多具体的属性。每个属性又默认对应了 contains|contains not|is|is not 四个条件中的一种 [3]。ioc_common 提供了一个编程规范，使得用户可以自定义属性，从而形成了对 OpenIOC 的扩展。

因此，要在系统中维持这张映射表。当完成聚类工作后，通过表所提供的映射关系，将特征的描述转化为 OpenIOC 指示器属性。

3.3.2　关键词树算法

关键词树（Keyword-Based Signature Tree）是一棵以 IOC 属性出现频率为依据而建立的树。它的每个节点都是一个 IOC 属性[16]。出现频率越高的 IOC 属性，越靠近根节点；出现频率越低的，越靠近叶子节点[16]。它根据 IOC 属性出现的频率、内容，将特征逐层次分类。建好的关键词树从根节点到任意非根节点路径上的所有 IOC 属性，都可以组成一个正则表达式[13]。

建立一棵关键词树的算法如下：

（1）建立一个根节点。

（2）对于提出来的 IOC 属性，计算它们出现的频率。选出频率最高的属性，建立一个子节点，其内容为当前 IOC 属性与其匹配的所有特征集，而已匹配的 IOC 属性不能在该路径的后面再次被匹配。

（3）对剩下的网络会话继续建立子节点，直到所有父节点所包含的会话都被这一层的子节点所匹配。接下来，建立下一层子节点，直到节点所有的网络会话没有可扩展的 IOC 属性。

（4）根据关键词树生成 IOC 逻辑关系。父子结点表示逻辑层次，兄弟结点之间的逻辑关系为“OR”。若父子结点所包含特征集相同，则将该子节点的逻辑层次提高到与其父节点一致，逻辑关系为“AND”。

在生成关键词树之后，每个节点就是一条简单表达式，而从根节点到叶子结点的任何一条路径代表了一条复杂表达式。因此，将关键词树所有的路径复合到一起，便可以生成一条攻击指示器（IOC）。

3.3.3　ioc_writer

ioc writer 是 OpenIOC 生成 IOC 的 API，提供了一系列函数[4]，部分函数的功能如表 3 所示。

表 3　ioc_writer 函数表

函数	功能
make_ioc()	创建一个新的空白的 IOC 文件
make_indicator_node()	创建一个逻辑节点（and/or）
make_indicatoritem_node()	创建表达式条件
make_description_node()	创建描述
add_parameter()	添加参数
ioc_writer()	保存文件
make_[item]_[indicatoritem]()	自定义属性

4　实验结果与验证

本次实验的恶意软件样本来源于“www.malshare.com”。该网站提供了最新发现的恶意软件样本，并提供样本下载。

实验环境：操作系统 ubuntu 14.04，硬件 CPU Intel core i3、16GB RAM，编程 python，数据库 MongoDB。

从 malshare 获取 800 多个样本，使用卡巴斯基对样本进行检测。根据恶意代码命名规则，挑选出 11 个恶意软件家族共 438 个样本，如表 4 所示。

表 4　恶意软件样本统计

家族名	数目
Trojan.Win32.BHO	62
Trojan-Downloader.Win32	42
Backdoor.Win32.Hupigon	39
Trojan-Spy.Win32.Zbot	81
HEUR:Trojan.Win32.Generic	8
Backdoor.Win32.Ruskill	27
Packed.Win32.Krap	34
Trojan-PSW.Win32	73
Backdoor.Win32.BlackHole	51
Trojan.Win32.StartPage	13
AdWare.WinLNK.Clicke	11

将这些样本通过恶意软件自动分析模块进行分析，并通过特征提取模块进行特征提取和样本聚类，结合 pylab，将聚类结果绘制成图[17]，如图 6 所示，共得到 10 个簇和 27 个噪声点。

图 6　聚类结果

将聚类得到的结果与上述 11 个家族进行比较，得到各家族样本的聚类情况，如图 7 所示。

图 7 中，柱状图的横轴表示各个家族的恶意软件，纵轴表示每个家族包含恶意软件的数量，用 11 种不同的灰度分别表示聚类后的 10 个聚类簇和噪声点。每一条柱表示了一个家族中的恶意软件的聚类情况。以 Trojan.Win32.BHO 为例，总共包含了 62 个样本，其中 56 个样本属于聚类 1，3 个样本属于聚类 2，3 个样本属于噪声点。由此可见，Trojan.Win32.BHO 家族中的样本绝大部分都被聚类到了一起。同理，其他各家族的情况也是如此。需要说明的是，HEUR:Trojan.Win32.Generic 这一家族，是由卡巴斯基使用了启发式扫描技术所产生的通用名称，加上样本数量较少，因此并没有单独聚为一类。

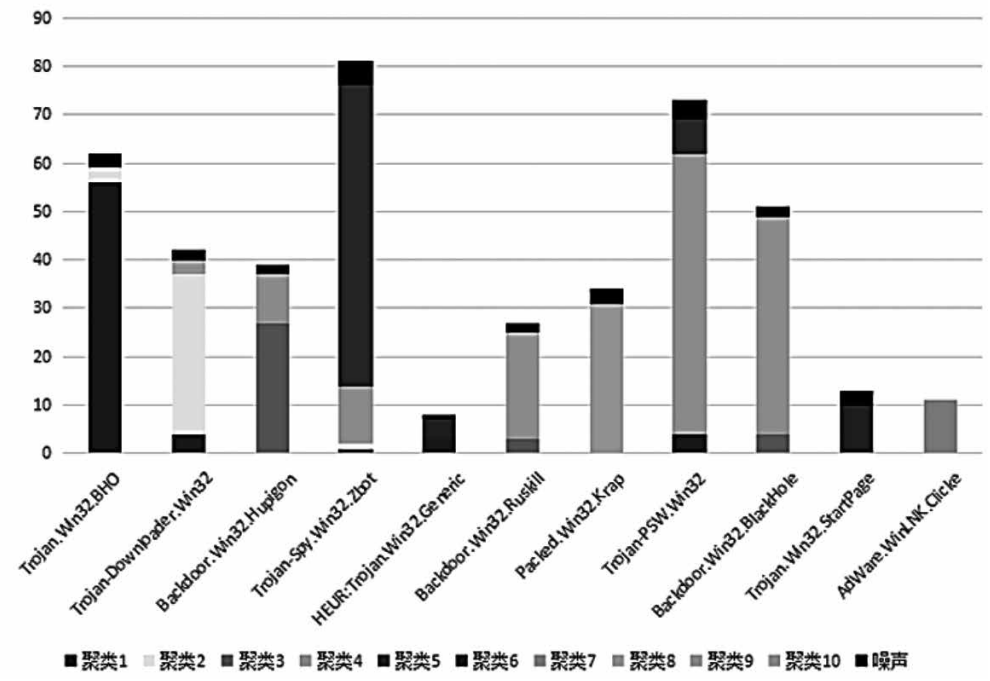

图 7　恶意软件家族聚类结果统计

因此，定义一个聚类簇中份额最大的恶意软件家族的样本数量占聚类簇样本总数的百分比为该聚类簇的准确率。于是，得到每个聚类簇的准确率，如表 5 所示。

从实验结果可以看出，本文采用的方法可以有效将相似性较高的恶意软件聚合在一起，提取它们的共有特征，方便生成高度概括性的 IOC。

以聚类 1（主要为 Trojan.Win32.BHO）为例，通过使用本文提出的关键词树算法，结合 ioc writer，生成 IOC 表达式，部分描述如图 8 所示。

表 5　恶意软件聚类准确性

类簇	代表家族	准确率 /（%）
1	Trojan.Win32.BHO	82.4
2	Trojan–Downloader.Win32	89.2
3	Backdoor.Win32.Hupigon	79.4
4	Trojan–PSW.Win32	82.9
5	Trojan–Spy.Win32.Zbot	89.9
6	Trojan.Win32.StartPage	71.4
7	AdWare.WinLNK.Clicke	100
8	Packed.Win32.Krap	96.8
9	Backdoor.Win32.Ruskill	69.5
10	Backdoor.Win32.BlackHole	81.2

```
<?xml version="1.0" encoding="us-ascii"?>
<ioc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2
  <short_description>Trojan.Win32.BHO</short_description>
  <authored_by>xwt</authored_by>
  <authored_date>2016-06-15T06:42:18</authored_date>
  <links />
  <definition>
    <Indicator operator="OR" id="ba371626-a478-4ea3-aac9-e23487f8a8bc">
      <IndicatorItem id="a16be557-afe5-4265-99e6-839bd2c1406b" condition="is">
        <Context document="FileItem" search="FileItem/FilenameCreated" type="mir" />
        <Content type="date">%Program Files%\NetMeeting\lawugeqi</Content>
      </IndicatorItem>
      <IndicatorItem id="b5662cf2-e25b-4ad4-b425-c5577c97ed8d" condition="is">
        <Context document="FileItem" search="FileItem/FilenameCreated" type="mir" />
        <Content type="date">%Program Files%\NetMeeting\ progyrtaky.html</Content>
      </IndicatorItem>
      <Indicator operator="OR" id="06ed2fbb-0363-4b2f-9004-9306cd5e9c3a">
        <Indicator operator="AND" id="1bdd86bf-1ab5-4bc0-aa48-f167d08abf55">
          <IndicatorItem id="dd36ddb7-ado7-40c0-9d2c-65c83c263f71" condition="contains">
            <Context document="RegistryItem" search="RegistryItem/KeyPath" type="mir" />
            <Content type="string">HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{CB44128D-95
          </IndicatorItem>
```

图 8　IOC 的 XML 形式

5 结　语

本文以 OpenIOC 框架和 Cuckoo 恶意软件自动化分析平台为基础，设计并实现了能够自动生成 IOC 文件的恶意代码自动化分析系统。该系统能够获取最新的威胁情报，自动对获取到样本进行动态分析，生成分析报告，并根据分析报告对样本进行聚类，生成高度概括、可机读、可实时共享的攻击指示器 IOC，实现了威胁情报快速、便捷的共享。

参考文献：

[1] FireEye.Poison Ivy:Assessing Damage and Extracting Intelligence[EB/OL].(2014-05-29)[2016-05-26].http://www.fireeye.com/resources/pdfs/fireeye-poison-ivy-report.pdf.

[2] Aron Stark.Introduction to Intelligence Threat[EB/OL].(2016-02-18)[2016-06-26].http://mt.sohu.com/20160218/n437788851.shtml.

[3] Windhawk.The Intelligence Threat Sharing Framework: OpenIOC[EB/OL].(2015-11-24)[2016-06-26].http://www.freebuf.com/sectool/86580.html.

[4] Mandiant.An Open Framework for Sharing Threat Intelligence[EB/OL].(2015-11-28)[2016-06-26].http://www.openioc.org/.

[5] Cuckoo official website.Automated Malware Analysis: Cuckoo Sandbox[EB/OL].(2016-03-28)[2016-06-27].http://docs.cuckoosandbox.org/en/latest/.

[6] 凌群 . 恶意程式分析沙箱 [EB/OL].(2015-02-15)[2016-06-28].http://www.syscom.com.cn/ePaper_New_Content.aspx?id=446&EPID=208&TableName=sgEPArticle.
LING Qun.The Sandbox for Analysis Malware[EB/OL].(2015-02-15)[2016-06-28].http://www.syscom.com.cn/ePaper_New_Content.aspx?id=446&EPID=208&TableName=sgEPArticle.

[7] 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(02):1-42.

[8] John Smith.Cuckoo Sandbox Part3:Testing[EB/OL].(2015-11-09)[2016-06-29]https://gwallgofi.com/cuckoo-sandbox-part-3-testing/.

[9] 徐小琳 , 云晓春 , 周勇林 . 基于特征聚类的海量恶意代码在线自动分析模型 [J]. 通信学报 ,2013,34(08):146-153.
XU Xiao-lin,YUN Xiao-chun,ZHOU Yong-lin.Huge Amount of Malcode Online Analysis based on Characteristic Cluster[J].Journal on Communications,2013,34(08):146-153.

[10] QIAN Yu-cun,PENG Guo-jun,WANG Ying,et al.Homology Analysis of Malicious Code and Family Clustering. [J]. Computer Engineering and Applications,2015,51(18):76-81.

[11] Mandiant.Tracking Malware Import Hashing[EB/OL].(2014-05-14)[2016-07-01].https：//www.mandiant.com/blog/tracking-malware-import-hashing.

[12] Alienvault.Tracking Down the Author of the Plugx Rat[EB/OL].(2014-05-19)[2016-07-01].http://www.alienvault.com/openthreat-exchange/blog/tracking-down-the-author-of-theplugx-rat.

[13] 纪晓宇 , 冷冰 , 周洁 . 一种基于行为的可信计算动态度量方法 [J]. 通信技术 ,2015,48(11):1290-1294.
JI Xiao-yu,LENG Bing,ZHOU Jie.A Trusted Computing Dynamic Measurement Method based on Behavior[J]. Communication Technology,2015,48(11):1290-1294.

[14] Wikipedia.DBSCAN[EB/OL].(2016-05-11)[2016-07-01].https://en.wikipedia.org/wiki/DBSCAN.

[15] 陈家豪 . 通过机器学习进行恶意软件分析 [EB/OL].(2016-01-11)[2016-07-02].http://www.freebuf.com/articles/network/92472.html.
CHEN Jia-hao.Do Malware Analysis by Machine Learning[EB/OL].(2016-1-11)[2016-07-02]http://www.freebuf.com/articles/network/92472.html.

[16] 白巍 , 潘道欣 , 贺宇轩 . 基于网络特征的恶意代码自动分析平台 [R]. 课程设计 ,2013.
BAI Wei,PAN Dao-xin,HE Yu-xuan.Malicious Code Auto Analysis Platform based on the Network Characteristics[R]. Curriculum Design,2013.

[17] CESARE S,XIANG Y,ZHOU W.Malwise-An Effective and Efficient Classification System for Packed and Polymorphic Malware[J].IEEE Trans on Computer,2013,62(06):1193-1206.

作者简介：

徐文韬（1992—），男，硕士，主要研究方向为网络攻防及渗透测试；

王轶骏（1980—），男，硕士，讲师，主要研究方向为网络攻防及系统安全；

薛　质（1971—），男，博士，教授，主要研究方向为计算机通信网及信息安全。