doi:10.11959/j.issn.1000-436x.2017244

# 网络协议隐形攻击行为的挖掘和利用

胡燕京[1,2]，裴庆祺[2]

（1. 武警工程大学网络与信息安全武警部队重点实验室，陕西 西安 710086；
2. 西安电子科技大学综合业务网理论及关键技术国家重点实验室，陕西 西安 710071）

**摘 要**：网络协议的隐形攻击行为生存性、隐蔽性和攻击性强，且不易被现有的安全防护手段检测到。为了弥补现有协议分析方法的不足，从实现协议程序的指令入手，通过动态二进制分析捕获协议的正常行为指令序列。然后通过指令聚类和特征距离计算挖掘出潜在的隐形攻击行为指令序列。将挖掘出的隐形攻击行为指令序列以内联汇编的方式加载到通用运行框架，在自主研发的虚拟分析平台 HiddenDisc 上动态分析执行，并评估隐形攻击行为的安全性。除了挖掘分析和有针对性的防御隐形攻击行为之外，还通过自主设计的隐形变换方法对隐形攻击行为进行形式变换，利用改造后的隐形攻击行为对虚拟靶机成功实施了攻击而未被发现。实验结果表明，对协议隐形攻击行为的挖掘是准确的，对其改造利用以增加信息攻防能力。

**关键词**：协议逆向分析；隐形攻击行为；指令聚类；隐形变换

**中图分类号**：TP393 **文献标识码**：A

# Mining and utilization of network protocol’s stealth attack behavior

HU Yan-jing[1,2], PEI Qing-qi[2]

(1.Network and Information Security Key Laboratory, Engineering University of the Armed Police Force, Xi’an 710086, China;
2.National Key Laboratory of Integrated Services Networks, Xidian University, Xi’an 710071, China)

**Abstract:** The survivability, concealment and aggression of network protocol’s stealth attack behaviors were very strong, and they were not easy to be detected by the existing security measures. In order to compensate for the shortcomings of existing protocol analysis methods, starting from the instructions to implement the protocol program, the normal behavior instruction sequences of the protocol were captured by dynamic binary analysis. Then, the potential stealth attack behavior instruction sequences were mined by means of instruction clustering and feature distance computation. The mined stealth attack behavior instruction sequences were loaded into the general executing framework for inline assembly. Dynamic analysis was implemented on the self-developed virtual analysis platform HiddenDisc, and the security of stealth attack behaviors were evaluated. Excepting to mining analysis and targeted defensive the stealth attack behaviors, the stealth attack behaviors were also formally transformed by the self-designed stealth transformation method, by using the stealth attack behaviors after transformation, the virtual target machine were successfully attacked and were not detected. Experimental results show that, the mining of protocol stealth attack behaviors is accurate, the transformation and use of them to increase information offensive and defensive ability is also feasible.

**Key words:** protocol reverse analysis, stealth attack behavior, instruction clustering, stealth transformation

## 1 引言

网络协议的隐形攻击行为是指通过网络协议成功实现攻击目标而难以被现有安防设备和安防技术所感知的攻击行为。近年来，针对特定计算机和目标网络的协议隐形攻击技术取得了长足的发展，成为网络空间安全最主要的威胁之一[1]。通过网络协议实施的隐形攻击也从最开始的独立简单行为迅速发展为数量巨大的复杂隐形恶意行为[2]。网络协议的隐形攻击行为旨在采用各种隐形技术手段，从目标节

**收稿日期**：2017-09-02
**基金项目**：国家自然科学基金资助项目（No.61373170, No.61402530, No.61309022, No.61309008）
**Foundation Item:** The National Natural Science Foundation of China (No.61373170, No.61402530, No.61309022, No.61309008)

点源源不断地秘密窃取高价值信息而不被发现。多数隐形攻击行为不对目标节点的软硬件造成明显的伤害，也不像病毒和恶意代码大量传播。相反，它们长期潜伏，默默监视着主机和网络的行为，只在条件成熟时发起短暂的攻击，如窃取重要文件和原始数据等敏感机密的信息，而后又迅速恢复潜伏状态，因此，现有的安全防护手段难以感知和应对这类隐形攻击行为。深入研究发现，我国互联网上已有数万台计算机被隐形攻击行为所控制，而且被检测到的主机上，平均每一台至少嵌入了 15 个不同类型的隐形攻击行为，一个攻击失效，可迅速切换到下一个隐形攻击。隐形攻击行为既能完成一次完整的攻击任务，又能为深入分析、深度攻击目标主机和网络收集攻击线索和信息。例如，隐形攻击行为可以劫持用户浏览器，也可以诱骗用户进入攻击者精心设计的“正常站点”（看似正常，实则是陷阱），从而秘密窃取用户隐私和敏感信息。

由于协议的所有行为，包括隐形攻击行为都不会超出实现它的程序代码的范畴，因此，从实现协议的程序代码分析协议的行为是最直接、最根本、最有效和最可靠的途径。通过对 2 316 个协议样本的实例分析，本文发现可以从协议的指令序列入手，分析和挖掘隐藏其中的隐形攻击行为。网络协议最主要的行为是接收和发送消息[3]，通过长期的分析，本文积累了大量这类指令序列的实例，对这些行为指令序列的特点也掌握得比较清楚和全面。协议的隐形攻击行为多种多样，短期内掌握所有隐形攻击行为的特点和行为模式也是不现实的。但是隐形攻击行为和正常协议行为相比，在指令序列上有明显的不同。根据大量实例分析的结果，本文提出了利用指令聚类挖掘协议隐形攻击行为的方法[4]。该方法能在短时间内，将潜在的隐形攻击行为从正常的协议行为中区分出来，并根据指令在类型、数量、执行频率等特征上的不同，自动生成不同的行为聚类。

指令聚类不但能快速准确地挖掘出潜在的隐形攻击行为，而且为进一步深入分析隐形攻击行为积累了宝贵的第一手资料。这些挖掘出的隐形攻击行为都是以一个二进制指令序列的形式保存的，通过在封闭的虚拟运行环境中触发其执行，可以掌握这些隐形攻击行为的具体功能。本文并不局限于挖掘、分析和防范隐形攻击行为。这些隐形攻击行为都是攻击者精心设计的，从这些挖掘出的实例中，本文可以探索隐形技术的一般规律。同时，对这些隐形攻击行为指令序列进行自主加工改造和自迷惑，改进其隐形能力，也可以利用它攻击敌人，本文不能被动防御，也要以攻对攻，用隐形攻击行为抵消攻击者的隐形攻击行为。这样既能充实信息攻防技术，提升信息攻防能力，也能在一定程度上打击攻击者的嚣张气焰。

## 2　相关工作

本节主要讨论隐形攻击行为挖掘问题的相关背景及本研究所涉及的工作。近年来，网络协议的隐形攻击行为，特别是对目标主机和网络具有恶意性的隐形行为，已经迅速发展成为网络安全面临的主要威胁之一。协议的隐形行为也从最初的单一简单行为快速发展成为复杂而隐蔽性强的恶意行为[5]。最基本的隐形是自迷惑技术，即协议程序在不改变自身原有功能的情况下，通过代码变换等手段来抵御逆向分析的一种反逆向技术[6]。迷惑技术可以多次使用，通过反复地迷惑能使相同的协议行为变得面目全非，看不出继承关系，实施逆向分析变得更加困难，也更加难以捕获和挖掘。但是协议的隐形攻击行为要付诸实施，必然需要相应指令序列的执行，只是这些指令序列的执行时机很难把握，指令序列也很难挖掘。本文研究在不攻击自迷惑技术的前提下，通过指令聚类来挖掘协议的隐形攻击行为，取得了进展。

对协议隐形攻击行为的挖掘分析大体上分为静态分析和动态分析 2 大类别。其困难主要体现在：1) 协议隐形攻击行为的种类复杂多样，难以穷尽，因此，很难定义隐形攻击行为的特征和行为模型；2) 对于不同种类的隐形攻击行为，特别是采用混淆技术后，其指令序列的长短、指令类型、调用频率等特征各不相同，难以统一规定和确定行为类型；3) 各行为指令序列之间可能存在复杂的依赖关系，而将隐形攻击行为的指令序列从海量数据中剥离和准确挖掘出来难度较大，而有效触发并分析其执行更加困难。传统的软件行为分析大多是手工作业，自动化程度低，工作强度大且容易出错。如 Samba 项目经过 12 年刻苦攻关，做了大量实验才完成了对其 SMB 协议的逆向分析[7]。

早期针对恶意行为的检测技术采用静态分析方法[8]，通过借助调试工具对协议程序进行反汇编分析和消息解析过程分析，从分析结果中提取恶意

行为的特征码。这种分析方法在基于静态特征码扫描的反病毒技术中被广泛应用，但是基于特征码的检测技术无法识别经过变形后的恶意行为。于是，人们将研究重点转向对协议程序语法和语义的分析，通过语义学对不同变体之间的特征码进行分析判定，从而检测出经过变形的恶意行为。然而，这些方法对于隐形行为的分析效果不佳。后来，针对通过模糊技术逃避检测的协议程序，提出了栈分析系统调用法和代码标准化法[9]。同时，有学者将工程学中的方法论运用到对协议程序代码的检测和分析中，在反恶意代码技术基于特征码检测的基础上提出了基于数据挖掘的代码检测法[10]。随着协议设计技术的不断进步，越来越多的隐形攻击行为不断涌现，国外学者开始对协议程序所包含的恶意行为进行自动化分析和研究，并取得了一定的进展。德国曼海姆大学可靠性分布式系统实验室的 CWSandbox 可将协议程序样本在虚拟机环境中运行，运用 API Hook 技术对协议行为进行追踪并对其恶意性进行分析，实现了恶意行为分析的自动化[11]。此外，Anubis 和 Norman Sandbox 使用沙箱技术提供在线分析服务，他们在沙箱中运行用户提供的协议程序样本，并对样本程序进行行为检测，成为检测和分析未知恶意行为的有效工具[12]。然而，这些方法只能对显式的恶意行为进行分析，且投入的人力物力巨大，面对未知协议的隐形攻击行为，这些方法均难以应对。

目前，协议行为分析相关的概念并没有完全统一，主要研究包括协议逆向工程、恶意软件行为分析、网络安全审计和网络行为分析等技术。国外学者提出了一些未公开的协议逆向分析方法，并开发了相应的分析工具和系统。而国内还罕有关于协议行为分析特别是隐形攻击行为方面的公开研究。本文将对未知协议隐形攻击行为的挖掘和利用问题进行探索性研究。

## 3 协议隐形攻击行为的挖掘

### 3.1 协议隐形攻击行为的描述

网络协议 $P$ 可看作由各种功能性指令序列 $C$ 组成的集合，即协议 $P=\{c_1, c_2,\cdots, c_n| n\in N\}$，其中，每一个 $c_i$ 表示一个特定功能的指令序列，所有指令序列的集合构成了该协议的全部功能。在这些指令序列中，公开执行的、能捕获到的正常行为的指令序列，称作协议的正常行为，记作 $P_{\text{nornal}}$，秘密执行的、只有在特殊条件下才触发执行的行为指令序列，称作协议的隐形攻击行为，记作 $P_{\text{stealth}}$。由此得出 $P=\{c_1, c_2,\cdots, c_n| n\in N\}\cong P_{\text{nornal}}\cup P_{\text{stealth}}$，即协议的行为由所有指令序列的集合组成，也由正常行为和隐形攻击行为的集合构成。协议若无隐形攻击行为，则 $P_{\text{stealth}}$=NULL。本文任务是挖掘深藏在协议中的隐形攻击行为，这是分析、防范和利用隐形攻击行为的基础和前提。

### 3.2 协议隐形攻击行为的挖掘方案

协议的隐形攻击行为虽然种类繁多，但在指令序列的特征上和正常行为具有显著的差异性，因此从指令序列层面将其挖掘出来是可行的。由于网络协议的主要功能都围绕着消息的发送和接收，这些正常行为的种类并不多，掌握其指令序列上的特点规律并不困难。所以，本文首先通过动态分析捕获协议样本的正常行为指令序列，将指令类型、执行顺序、指令数量和执行频率进行统计分析，生成协议正常行为在指令级的特征向量。然后利用本文自主设计的指令聚类算法，挖掘所有协议样本中潜在的隐形攻击行为指令序列。通过计算隐形攻击行为和正常行为之间的特征距离，将潜在的隐形攻击行为进一步划分，生成不同的行为聚类，这就为隐形攻击行为的分析、防范和利用准备了第一手的宝贵资料。协议隐形攻击行为的挖掘方案如图 1 所示。

该方案在本文自主研发的协议行为虚拟分析平台 HiddenDisc 上实施。协议程序集是近 5 年来收集到的 2 316 个协议程序样本。将其在虚拟分析平台上依次运行，通过动态二进制分析，可以捕获到协议运行时的行为指令序列。大量实例分析表明，这些大多是正常的网络通信行为，指令序列上虽然略有不同，但总体上差异不大，勉强可将其划分成 15 个不同的行为指令序列。通过指令序列对比，将捕获到的行为指令序列还原到原始协议程序的上下文中，精确定位这些指令序列在原协议程序中的地址，作为进一步分析的基础。将捕获到的行为指令序列和原始协议程序一起作为输入，根据本文提出的指令聚类算法，进行指令序列聚类。计算挖掘出的隐形攻击行为和已知行为之间的特征距离，根据特征距离的不同生成不同的隐形攻击行为聚类。隐形攻击行为的挖掘细节将在实验部分详细说明。

### 3.3 协议隐形攻击行为的挖掘算法

**算法 1** 协议隐形攻击行为挖掘

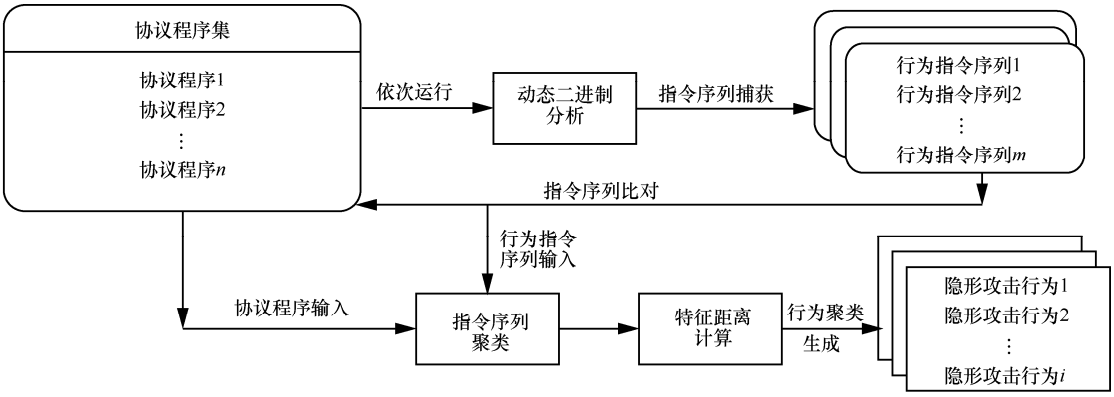



图 1　隐形攻击行为挖掘方案

mineStealthBehavior()

**输入**　已捕获的行为指令序列

协议程序集

**输出**　隐形攻击行为指令序列

parse(Instruction);//指令解析

judgeInstType;//判断指令类型

switch(InstType)

case:“push”, “jmp”, “call”…//若遇到“push”, “jmp”, “call”等

mark(F);//函数调用类指令，则将其标记为 F

break;

case: “add”, “mul”, “div”…//若遇到“add”, “mul”, “div” 等

mark(D);//数据处理类指令，则将其标记为 D

break;

case: “cmp”, “inc”, “xor”…//若遇到“cmp”, “inc”, “xor”等

mark(C);//流程控制类指令，则将其标记为 C

break;

instructionClustering(F,D,C);//对标记后的指令进行指令聚类

computeDistance($P_{\text{nornal}}$, $P_{\text{stealth}}$);//计算挖掘出的隐形攻击行为和捕获到的正常行为之间的特征距离

generateBehaviorClusters();//依据特征距离的不同，生成不同的隐形攻击行为聚类

算法 1 应用指令聚类的思想实现对隐形攻击行为的挖掘。首先通过指令解析，将所有的指令划分并标记为 3 类最基础的指令类型，即函数调用相关指令（F），数据处理相关指令（D）和流程控制相关指令（C）。这样，协议程序的所有指令就表示为 3 类基本指令 F、D 和 C 的排列组合。接着对标记后的指令进行 *n*-gram 聚类，计算挖掘出的隐形攻击行为和已捕获的正常行为之间的距离，根据隐形攻击行为偏离正常行为的不同距离，生成不同的行为聚类。至此，协议的潜在隐形攻击行为已经挖掘出来，但行为的具体内容，以及如何能够使用，还需要更进一步的研究。

## 4　协议隐形攻击行为的分析和利用

### 4.1　隐形攻击行为的触发和分析

指令聚类只能将潜在的隐形攻击行为区分出来，而要掌握该行为的具体功能则需要将其触发执行。挖掘出的隐形攻击行为指令序列通常具有难理解、难运行、少语义、不完整等特点。这些机器级的指令数量庞大，没有高级语言中类似函数、类型和变量等抽象表达，而且子功能之间没有明显的分隔标志。看不到变量和类型，只有寄存器和内存数据。从指令操作码只能掌握有限的功能信息，更加精确的数据结构和信息表示则较难获取。机器级指令不包含字符串等高级语言才拥有的数据类型信息，要获得较准确的数据类型信息通常需要分析人员的推断。由于指令聚类存在误差，这些指令序列的开始和结束部分并不一定完整和准确，而且机器级指令通常缺少高级语言所具有的语义信息，因此，这些指令序列如何执行成为首要和基本的问题。隐形攻击行为的触发和分析如图 2 所示。

将挖掘出的潜在隐形攻击行为指令序列和协议程序样本均作为输入。由于挖掘出的指令序列可能存在不完整、不准确和难以运行等问题，需要对其进行格式化。通过对大量行为指令序列的分析，本文开发了一个针对指令序列的可运行框架。将挖

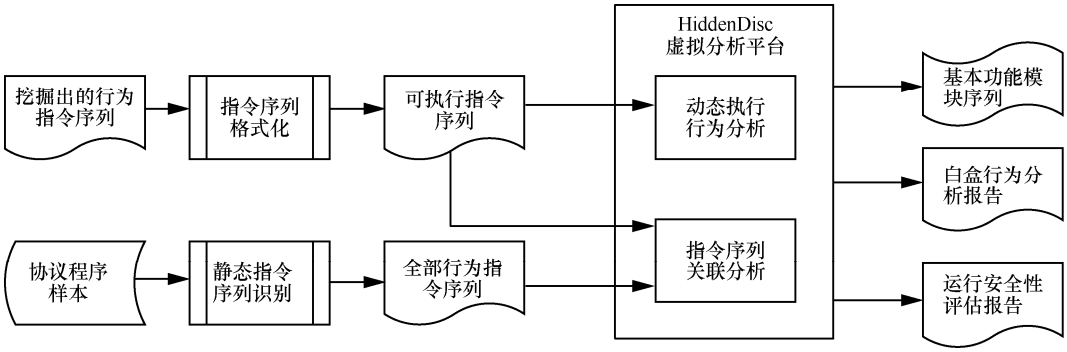



图 2　隐形攻击行为触发和分析

掘出的行为指令序列嵌入可运行框架中，就可以像调用函数一样触发其运行，这样就生成了可执行指令序列。另一个输入协议程序样本也需要进行一些预处理。通过静态指令序列识别模块，将所有的协议程序样本表示成一个个行为指令序列，全部行为指令序列就成为协议程序样本行为的静态表示。这里所谓的全部行为是指协议所有的显式行为，不包括隐形攻击行为。将可执行指令序列在虚拟分析平台 HiddenDisc 上动态执行，分析其行为；同时将可执行指令序列和全部行为指令序列一起进行指令序列关联分析，最终生成可表示隐形攻击行为的基本功能模块序列和白盒行为分析报告，并根据动态行为分析的结果生成可评估隐形攻击行为运行安全性的评估报告。有了这些结果，就可以掌握协议隐形攻击行为的具体功能及其运行危害，并可以有针对性地进行安全防范，甚至对攻击者采取反制措施。

### 4.2　隐形攻击行为的利用

隐形攻击行为大都精心设计、巧妙隐藏，危害是长期而隐蔽的，对于这类巧妙设计的攻击手段，本文不能满足于检测和防范，而应该进一步对其加以研究和利用、有效改造、充实信息攻防手段。指令聚类能够以较高的效率挖掘出隐形攻击行为指令序列，但这些指令序列通常只含有核心攻击指令，并不完整，也不能直接投入运行，需要将其加入本文的运行框架。运行框架主要完成三大功能：1) 识别隐形攻击行为的核心指令序列，抽取指令依赖和数据依赖，形成一个独立的行为指令序列；2) 利用 C 编译器为每一个隐形攻击行为生成一个函数，将隐形攻击行为作为内联汇编（inline assembly）函数体；3) 利用本文自主的隐形算法对可执行的隐形攻击行为实施隐形变换，让其快速变成本文的隐形攻击行为，并可利用它们对敌展开隐形攻击。对隐形攻击行为的利用将在实验部分详细讨论。

## 5　实验及分析

### 5.1　实验平台的搭建

实验平台由 4 台行为分析客户机、1 台控制服务器和 1 台分析服务器组成。鉴于隐形攻击行为可能会对真实的物理软硬件产生破坏作用，本文自主研发了能模拟真实硬件、操作系统和各类软件的虚拟分析平台 HiddenDisc。所有行为分析客户机均部署 HiddenDisc 虚拟分析系统，协议样本的运行、分析和利用均在虚拟分析平台上实施。实验平台的拓扑结构如图 3 所示。

各个行为分析客户机分别分析送入其中的协议程序样本，协议行为分析原始数据上传到控制服务器，控制服务器将所有客户机的协议行为分析数据汇总，生成全部协议样本的行为分析数据，并送达分析服务器。分析服务器根据指令聚类算法和虚拟分析平台运行、分析协议程序的结果，关联各部分数据，综合分析，生成隐形攻击行为分析报告。

### 5.2　隐形攻击行为分析实例

将 2 316 个协议样本依次在 HiddenDisc 虚拟分析平台上运行。其中已知正常协议 187 个，已知僵尸网络协议和恶意协议 273 个，其余 1 856 个为未知行为协议。在未知行为协议中，可能包含正常行为，也可能含有隐形攻击行为。隐形攻击行为指令序列表面上看没有明显的恶意行为，然而若将它们组合在一起却可能造成重大的安全隐患。图 4 是对本文捕获到的未知协议样本 CBot-3530 隐形攻击行为指令序列的分析。指令聚类挖掘出这 5 个行为和协议正常的收发消息不同，通过在虚拟分析平台 HiddenDisc 上动态执行和分析，发现这 5 个行为也

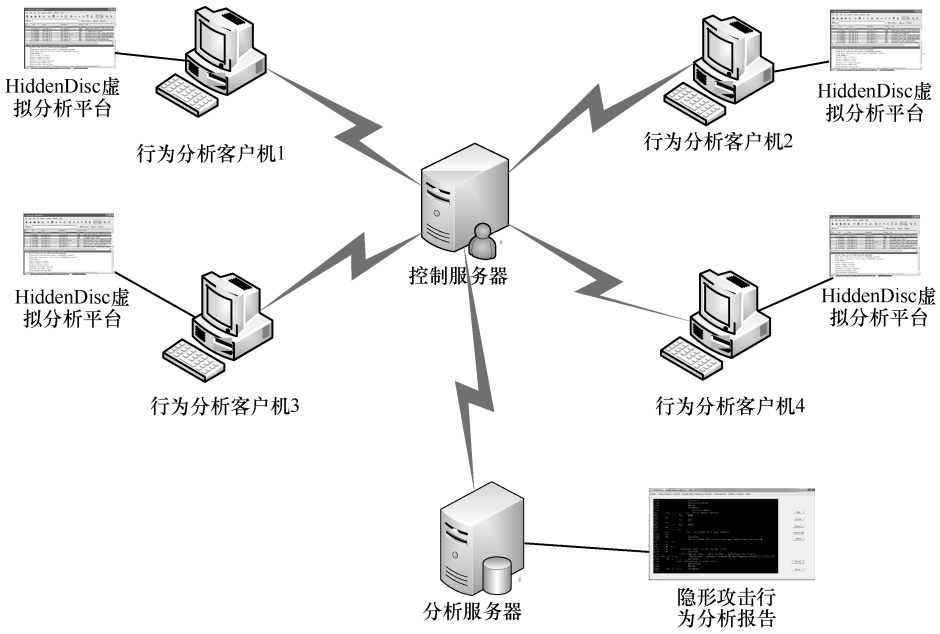



图 3　实验分析结构

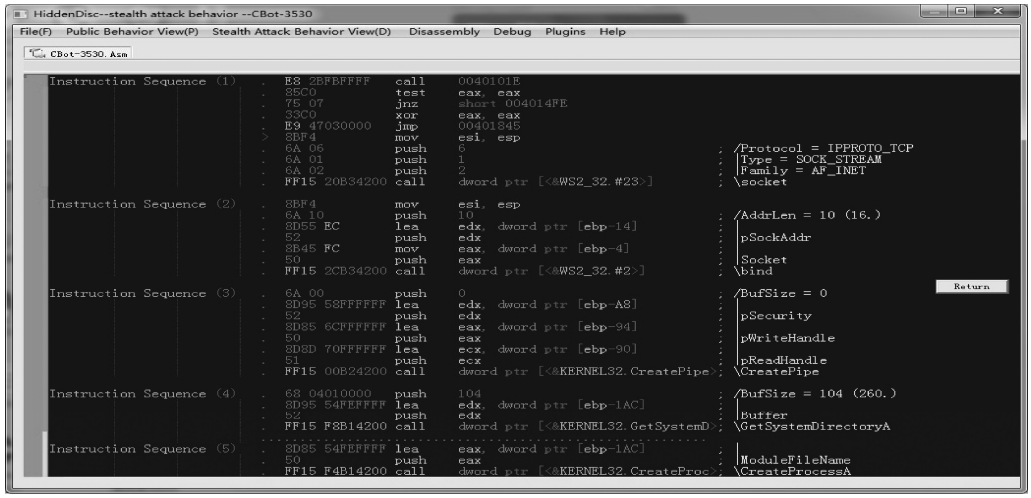



图 4　未知协议 CBot-3530 的隐形攻击行为分析

属于正常的通信行为。如指令序列（1）和（2）是 socket 网络通信的正常行为，这一类协议通常都具有类似的指令特点。指令序列（3）创建匿名管道，实现进程间通信，虽然和网络通信的行为相去甚远，但也属于正常的系统调用。指令序列（4）开始变得可疑，因为它要获得系统目录，一个正常的网络通信却要在背后执行获取系统目录的操作，显然不合理。指令序列（5）和随后的指令序列都是创建进程，虽然不是危险操作，却和正常的网络通信行为大相径庭。根据动态执行的结果，结合与其他指令序列的关联分析，最终确定 CBot-3530 是 Windows 系统下的一个后门程序，虽不进行传染和显性的破坏，但可以通过网络远程执行指令和代码，秘密窃取目标主机的数据。因此，这类隐形攻击行为表面上看来合法，而实际威胁和危害巨大，且可以长期潜伏而不被发现。

通过对 2 316 个协议样本的分析，发现协议的正常行为、恶意行为和隐形攻击行为在基因指令的分布上存在明显的差异性。行为分析的结果如图 5 所示。

图 5 显示，协议的正常行为中，函数调用类指令最少，而数据处理类指令最多，表明网络协议的正常功能主要表现在消息数据的接收和发送处理上。在本文的实验中，第 3 类恶意软件协议的行为特点非常明显，函数调用类指令最多，其次是流程转移类指令，再次是数据处理类指令，而数据处理类指令也不少，曲线的走势和协议的

正常行为差异巨大。这就在指令层面上又证明了流量分析的结果。协议的恶意行为通常会发送一定量的数据，如僵尸网络的命令控制协议（C&C），据此可以窃取大量僵尸主机的数据、发送垃圾邮件以及实施分布式拒绝服务攻击等。图 5 中上面 3 条曲线分别表示 3 类隐形攻击行为，这 3 类隐形攻击行为虽然功能各异，但基因指令的分布大体呈现出一致性。流程控制类指令占绝对多数，函数调用类指令也相当多，而数据处理类指令却非常少。一个网络协议一直在运行，却几乎不发送和接收任何数据，这就是协议分析仪通常很难发现隐形攻击行为的原因，也是协议隐形攻击行为的一个显著特点。

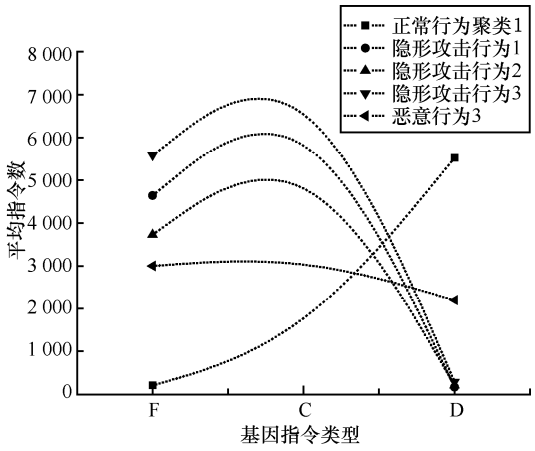



图 5　协议不同行为在基因指令上的分布特点

### 5.3　隐形攻击行为的利用实例

以样本 CBot-3530 为例，将指令聚类挖掘出的隐形攻击行为指令序列，作为内联汇编，加入运行框架。在 HiddenDisc 虚拟平台上编译成功后，就生成了可运行的攻击程序。再利用本文自主设计的隐形变换算法将攻击程序加密，生成可运行的隐形攻击程序。隐形变换前后的代码如图 6 所示。

如图 6 所示，隐形变换后的代码变得完全失去逻辑意义，不容易被反病毒软件和入侵检测系统识别。利用该隐形攻击程序控制一台 Windows 操作系统的靶机，可以秘密窃取虚拟靶机上的数据而不被防火墙、入侵检测和入侵防御系统拦截。

### 5.4　讨论

协议的隐形攻击行为隐蔽性、生存性和攻击性都很强，而且大都进行秘密渗透攻击，不对目标软硬件造成直接破坏，长期潜伏、短时行动，因此，常规安全防护手段很难检测到它。大量分析实例表明，协议行为的不同根源于其指令序列的不同，表现在 3 类基因指令的数量、执行频率和执行顺序的不同。因此，本文通过指令聚类能够较为高效地将未知协议的隐形攻击行为挖掘出来。挖掘出隐形攻击行为指令序列后，在 HiddenDisc 虚拟平台上运行、分析，可以有针对性地抵御隐形攻击行为。鉴于隐形攻击行为设计的巧妙性和特殊性，本文在安全防护的基础上更进一步，尝试将剥离出来的隐形

```
Instruction Sequence (1)    . E8 2BFBFFFF   call    0040101B
                            . 85C0          test    eax, eax
                            . 75 07         jnz     short 004014FE
                            . 33C0          xor     eax, eax
                            . E9 47030000   jmp     00401845
                            > 8BF4          mov     esi, esp
                            . 6A 06         push    6                               ; /Protocol = IPPROTO_TCP
                            . 6A 01         push    1                               ; |Type = SOCK_STREAM
                            . 6A 02         push    2                               ; |Family = AF_INET
                            . FF15 20B34200 call    dword ptr [<&WS2_32.#23>]       ; \socket

Instruction Sequence (2)    . 8BF4          mov     esi, esp
                            . 6A 10         push    10                              ; /AddrLen = 10 (16.)
                            . 8D55 EC       lea     edx, dword ptr [ebp-14]         ; |
                            . 52            push    edx                             ; |pSockAddr
                            . 8B45 FC       mov     eax, dword ptr [ebp-4]          ; |
                            . 50            push    eax                             ; |Socket
                            . FF15 2CB34200 call    dword ptr [<&WS2_32.#2>]         ; \bind

Instruction Sequence (3)    . 6A 00         push    0                               ; /BufSize = 0
                            . 8D95 58FFFFFF lea     edx, dword ptr [ebp-A8]         ; |
                            . 52            push    edx                             ; |pSecurity
                            . 8D85 6CFFFFFF lea     eax, dword ptr [ebp-94]         ; |
                            . 50            push    eax                             ; |pWriteHandle
                            . 8D8D 70FFFFFF lea     ecx, dword ptr [ebp-90]         ; |
                            . 51            push    ecx                             ; |pReadHandle
                            . FF15 00B24200 call    dword ptr [<&KERNEL32.CreatePipe>; \CreatePipe

Instruction Sequence (4)    . 68 04010000   push    104                             ; /BufSize = 104 (260.)
                            . 8D95 54FEFFFF lea     edx, dword ptr [ebp-1AC]        ; |
                            . 52            push    edx                             ; |Buffer
                            . FF15 F8B14200 call    dword ptr [<&KERNEL32.GetSystemD>; \GetSystemDirectoryA
                            ...............................................
Instruction Sequence (5)    . 8D85 54FEFFFF lea     eax, dword ptr [ebp-1AC]        ; |
                            . 50            push    eax                             ; |ModuleFileName
                            . FF15 F4B14200 call    dword ptr [<&KERNEL32.CreateProc>; \CreateProcessA
```

(a) 隐形变换前

```
   D93477           fstenv  (28-byte) ptr [edi+esi*2]
   15 DD347700      adc     eax, 7734DD
   0000             add     byte ptr [eax], al
   004B 45          add     byte ptr [ebx+45], cl
   52               push    edx
   4E               dec     esi
   45               inc     ebp
   4C               dec     esp
   3332             xor     esi, dword ptr [edx]
   2E:              prefix cs:
   64:6C            ins     byte ptr es:[edi], dx
   6C               ins     byte ptr es:[edi], dx
   0000             add     byte ptr [eax], al
   0000             add     byte ptr [eax], al
   47               inc     edi
   65:74 50         je      short 0042D0A2
   72 6F            jb      short 0042D0C3
   6341 64          arpl    word ptr [ecx+64], ax
   64:72 65         jb      short 0042D0BF
   73 73            jnb     short 0042D0CF
   0000             add     byte ptr [eax], al
   0047 65          add     byte ptr [edi+65], al
   74 4D            je      short 0042D0B0
   6F               outs    dx, dword ptr es:[edi]

   65:48            dec     eax
   61               popad
   6E               outs    dx, byte ptr es:[edi]
   64:6C            ins     byte ptr es:[edi], dx
   65:41            inc     ecx
   0000             add     byte ptr [eax], al
   004C6F 61        add     byte ptr [edi+ebp*2+61], cl
   64:4C            dec     esp
```

(b) 隐形变换后

图 6　示例代码隐形变换前后对比

**表 1**　　**相关主流技术方案对比**

| 方案 | 技术路线 | 分析对象 | 能否抵御加密/混淆技术 | 能否识别隐形攻击行为 | 成果 |
|---|---|---|---|---|---|
| 方案[13]方案 | 分析协议程序指令流 | 协议程序 | × | × | 推断出协议格式 |
| 文献[14]方案 | 使用试探性攻击挖掘协议漏洞 | 协议程序、协议消息 | √ | × | 挖掘出协议漏洞 |
| 文献[15]方案 | 逆向分析协议工作原理，伪装成被控端加入到控制网络 | 僵尸网络 C&C 协议 | √ | × | C&C 协议规格和工作原理 |
| 文献[16]方案 | 利用数据流图分析自动识别加密算法 | 二进制代码 | √ | × | 自动识别加密算法 |
| 文献[17]方案 | 采用 Win32 API 从大量二进制程序集中发现感兴趣的动态行为模式 | 二进制程序集 | × | × | 识别程序行为模式 |
| 方案[18]方案 | 逆向工程堡垒恶意软件，并掌握其内部结构和功能 | 堡垒恶意软件 | √ | × | 提高了逆向工程相似恶意软件变体的能力 |
| 文献[11]方案 | 使用蜜罐网络系统安全属性检测网络层和应用层攻击 | 蠕虫程序及网络流量 | √ | × | 检测网络层和应用层攻击 |
| 本文方案 | 使用指令聚类挖掘，执行模板执行，变形技术利用协议隐形攻击行为 | 协议程序及网络消息 | √ | √ | 挖掘出隐形攻击行为指令序列，并能够防范和利用隐形攻击行为 |

攻击行为指令序列再通过自主研发的隐形算法实施隐形变换，从而生成形式完全不同的、自主可用的隐形攻击程序，充实本文的信息攻防手段。初步实验表明，对隐形攻击行为的改造和利用是可行的，不但能有针对性地实施安全防护，而且有利于提高的信息进攻能力。和主流研究工作相比，本文研究的特点和优势较为明显。表 1 从技术路线、分析对象、可否抵御加密/混淆、可否识别隐形攻击行为等方面进行比较研究。

从表 1 可知，目前的研究在恶意软件分析、僵尸网络检测和协议逆向工程等方面都取得了丰硕的成果，但对协议隐形攻击行为的研究很少涉及。本研究恰好弥补了这一空白，不仅能够挖掘隐形攻击行为，还能对其加以利用，提高了本文的信息攻防能力。

## 6　结束语

本文通过动态二进制分析捕获协议暴露出来的行为指令序列，又采用新颖的指令聚类算法挖掘隐形攻击行为指令序列，进而对挖掘出的隐形攻击行为实施自主设计的隐形变换，对其加以利用，以充实信息攻防技术。实验表明，通过指令聚类来挖掘隐形行为是有效的，通过自主隐形变换利用该行为实施隐形攻击也是可行的。当前，对该课题的研究尚处于初级阶段，指令聚类的精确度如何，可否挖掘出所有的隐形攻击行为还有待进一步深入研究。另外，是否挖掘出的所有隐形攻击行为都能够有效利用还不能保证，有些挖掘出的指令序列甚至难以运行和分析，而且本文的隐形变换，只是实现了对靶机的攻击，隐形效果如何与是否能够逃避对手的检测和追踪，还没有进行充分的测试和验证。这些问题都是下一步研究的方向。

## 参考文献：


[1] HARALE S T A. Detection and analysis of network & application layer attacks using honey pot with system security features[J]. International Journal of Advance Research, Ideas and Innovations in Technology, 2017: 1-4.

[2] MING J, XIN Z, LAN P, et al. Impeding behavior-based malware analysis via replacement attacks to malware specifications[J]. Journal of Computer Virology and Hacking Techniques, 2016: 1-15.

[3] BOSSERT G, GUIHÉRY F, HIET G. Towards automated protocol reverse engineering using semantic information[J]. 9th ACM Symposium on Information, Computer and Communications Security, 2014.

[4] 胡燕京, 裴庆祺. 网络协议隐形攻击行为的聚类感知挖掘[J]. 通信学报, 2017, 38(6): 39-48.
HU Y J, PEI Q Q. Clustering perception mining of network protocol's stealth attack behavior[J]. Journal on Communicaitons, 2017, 38(6): 39-48.

[5] ANDERSON B, STORLIE C, LANE T. Improving malware classification: bridging the static/dynamic gap[C]//The 5th ACM Workshop on Security and Artificial Intelligence. 2012.

[6] EGELE M, SCHOLTE T, KIRDA E, et al. A survey on automated dynamic malware-analysis techniques and tools[J]. ACM Computing Surveys, 2012: 1-42.

[7] CABALLERO J, SONG D. Automatic protocol reverse-engineering: message format extraction and field semantics inference[J]. Computer Networks, 2013: 451-474.

[8] LCLI X D. A survey on methods of automatic protocol reverse engineering[C]//The 2011 Seventh International Conference on Computational Intelligence and Security. 2011: 685-689.

[9] CANFORA G, IANNACCONE A, VISAGGIO C. Static analysis for the detection of metamorphic computer viruses using repeated-instructions counting heuristics[J]. Journal of Computer Virology and Hacking Techniques, 2014: 11-27.

[10] KANG B, KIM T, KWON H, et al. Malware classification method via binary content comparison[C]//The 2012 ACM Research in Applied Computation Symposium. 2012.

[11] HAN K, LIM J H, IM E G. Malware analysis method using visualization of binary files[C]//The 2013 Research in Adaptive and Convergent Systems. Canada, 2013.

[12] QIAO Y, HE J, YANG Y, et al. A lightweight design of malware behavior representation[C]//IEEE International Conference on Trust Security and Privacy in Computing and Communications, IEEE Computer Society. 2013: 1607-1612.

[13] CABALLERO J, POOSANKAM P, KREIBICH C, et al. Dispatcher: enabling active botnet infiltration using automatic protocol reverse-engineering[C]//The 16th ACM Conference on Computer and Communications Security. Chicago, Illinois, USA, 2009.

[14] KANG J, PARK J H. A secure-coding and vulnerability check system based on smart-fuzzing and exploit[J]. Neurocomputing, 2017.

[15] BUCHLER M, HOSSEN K, MIHANCEA P F, et al. Model inference and security testing in the spacios project[C]//Software Maintenance, Reengineering and Reverse Engineering. 2014:411-414.

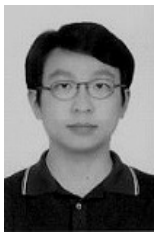
[16] CUI B, WANG F, HAO Y, et al. A taint based approach for automatic reverse engineering of gray-box file formats[J]. Soft Computing, 2015: 1-16.

[17] POLINO M, SCORTI A, MAGGI F, et al. Jackdaw: towards automatic reverse engineering of large datasets of binaries[J]. Detection of Intrusions and Malware, and Vulnerability Assessment. Springer International Publishing, 2015: 121-143.

[18] RAHIMIAN A, ZIARATI R, PREDA S, et al. On the reverse engineering of the citadel botnet[M]. Foundations and Practice of Security. Springer International Publishing, 2014: 408-425.

**作者简介：**

**胡燕京**（1980-），男，陕西西安人，博士，武警工程大学讲师，主要研究方向为信息系统安全防护、网络协议逆向分析。



**裴庆祺**（1975-），男，广西玉林人，西安电子科技大学教授、博士生导师，主要研究方向为数字内容保护与无线网络安全。