HoneyBow：一个基于高交互式蜜罐技术的恶意代码自动捕获器

诸葛建伟[1]，韩心慧[1]，周勇林[2]，宋程昱[1]，郭晋鹏[1]，邹维[1]

(1. 北京大学 计算机科学技术研究所，北京 100871；2. 国家计算机网络应急技术处理协调中心，北京 100029)

摘 要：恶意代码已成为互联网最为严重的安全威胁之一，自动化捕获恶意代码样本是及时有效地应对恶意代码传播的必要前提，提出了一个基于高交互式蜜罐技术的恶意代码自动捕获器 HoneyBow。相比较于基于低交互式蜜罐技术的 Nepenthes 恶意代码捕获器，HoneyBow 具有恶意代码捕获类型更为全面、能够捕获未知恶意代码的优势，互联网上的实际恶意代码捕获记录对比和 Mocbot 蠕虫的应急响应处理实例对其进行了充分验证。

关键词：恶意代码；恶意代码捕获；蜜罐；蜜网

中图分类号：TP309.5 **文献标识码**：A **文章编号**：1000-436X(2007)12-0008-06

HoneyBow: an automated malware collection tool based on the high-interaction honeypot principle

ZHUGE Jian-wei[1], HAN Xin-hui[1], ZHOU Yong-lin[2], SONG Cheng-yu[1], GUO Jin-peng[1], ZOU Wei[1]

(1. Institute of Computer Science and Technology, Peking University, Beijing 100871, China;

2. National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China)

Abstract: Malware has become one of the severest threats to the public Internet. To deal with the malware breakout effectively as early as possible, an automated malware collection solution must be implemented as a precondition. An automated malware collection tool was presented based on the high-interaction honeypot principle called HoneyBow. Comparing with the Nepenthes platform based on the low-interaction honeypot principle, HoneyBow has its advantages on wilder range of captured malware samples and the capability of collecting unknown malware samples, which are validated by the experiment results from wild malware collection and the case of Mocbot dealment.

Key words: malware; malware collection; honeypot; honeynet

1 引言

自从 1988 年莫里斯蠕虫爆发以来，恶意代码已成为互联网上影响最为广泛且最为严峻的安全威胁，特别是具有主动传播能力的恶意代码，不仅对单独受感染主机造成危害，同时对公共互联网的整体安全构成了严重损害。为了应对恶意代码对互联网所带来的安全威胁，计算机应急响应部门、反病毒厂商和安全研究人员必须获取恶意代码的样本，通过深入分析了解其传播和感染机理，然后才能给出准确的检测特征码和适宜的处理措施。传统的恶意代码样本捕获途径一般包括现场取证采集、用户上报、与其他厂商单位交换等，这些传统渠道一般需要人工介入，而这种需要人工介入的恶意代码样本采集方式并不能够适应目前恶意代码变种的生产时间不断的减少，以及网络传播型恶意代码爆发

收稿日期：2007-09-03；修回日期：2007-11-20

基金项目：国家高技术研究发展计划基金资助项目(“863”计划)(2006AA01Z445)；国家 242 信息安全计划基金资助项目(2006A30)

Foundation Items: The National High Technology Research and Development Program of China (863 Program) (2006AA01Z445); The National 242 Information Security Research Program of China (2006A30)

传播速度不断提高的趋势。因此需要引入完全自动化的恶意代码样本捕获方法。

蜜罐作为一种新兴的攻击诱骗技术，已经在互联网安全威胁监测方向上得到了较为广泛的应用[1,2]。蜜罐技术一般按照蜜罐系统提供给攻击者的交互等级划分为低交互式蜜罐技术和高交互式蜜罐技术 2 类。低交互式蜜罐一般采用模拟或仿真网络服务的方式，为攻击者提供有限的交互，一般只能诱骗自动化的攻击方式。低交互式蜜罐的部署和维护均较为简单，所带来的安全风险也较低。最为典型的低交互式蜜罐为开源的 Honeyd 工具[3]。高交互式蜜罐则使用完全真实的系统为攻击者提供更为全面和真实的交互性，与低交互式蜜罐相比，高交互式蜜罐的部署和维护较为困难，被攻击者攻陷后也将带来更大的安全风险。目前最普遍使用的高交互式蜜罐系统构建方式即蜜网项目组所提出的第三代蜜网技术[4]。

正是由于蜜罐技术在安全威胁监测方面存在的价值，研究者也开始基于蜜罐技术构建恶意代码样本自动捕获和监控方法，其中最为著名的是由德国蜜网项目组所研发的 Nepenthes 恶意代码捕获器[5]。Nepenthes 采用低交互式蜜罐技术思想，通过模拟存有漏洞的网络服务诱骗并捕获针对这些已知漏洞的恶意代码样本，由于本文所提出的基于高交互式蜜罐技术的 HoneyBow 恶意代码捕获器将与其进行针对性的比较，因此将在第 2 节中对 Nepenthes 工具进行详细介绍。此外 iDefense 实验室的 David Zimmer 也实现了一个类似的基于低交互式蜜罐技术的恶意代码捕获器 Multipot[6]。在高交互式蜜罐技术方向上，GA Tech 大学的 Levine 等人曾利用蜜网技术捕获并深入分析内核套件类恶意代码[7]，但在本文之前，尚未有基于高交互式蜜罐技术思想的自动化恶意代码捕获方法。

本文介绍了一个基于高交互式蜜罐技术的恶意代码自动捕获器 HoneyBow，与低交互式蜜罐技术通过模拟存有安全漏洞的网络服务诱骗和捕获恶意代码不同，HoneyBow 在高交互式蜜罐系统上直接构建，通过文件系统实时监控和文件列表交叉对比方法捕获感染高交互式蜜罐的恶意代码样本。相比较于 Nepenthes 工具，HoneyBow 存在捕获恶意代码更具全面性，能够捕获未知恶意代码样本的优势，实际恶意代码捕获记录的统计分析结果和 Mocbot 蠕虫应急响应处理实例都充分验证了这一点；此外 HoneyBow 的部署更具灵活性，其更新代价较低，无需深入调查安全漏洞的底层机制和实现特定安全漏洞的模拟脚本。

本文的结构如下：第 2 节介绍基于低交互式蜜罐技术的 Nepenthes；第 3 节详细阐述基于高交互式蜜罐技术的 HoneyBow 恶意代码捕获器的组成结构和实现技术，以及其与 Nepenthes 相比的优势和劣势；第 4 节则根据实际恶意代码捕获情况的统计分析和 Mocbot 蠕虫应急响应处理事件比较 HoneyBow 和 Nepenthes 在恶意代码捕获方面的效果；第 5 节为结束语。

2　Nepenthes——基于低交互式蜜罐技术的恶意代码捕获器

本节将介绍由德国蜜网项目组中的 Paul Baecher 和 Markus Koetter 等人所开发的 Nepenthes 工具[5]——开源的基于低交互式蜜罐技术的恶意代码捕获器。

Nepenthes 工具的基本设计思想是通过模拟存有漏洞的网络服务，构建低交互式的蜜罐系统，使其能够与网络上正在传播的恶意代码进行一定程度上的交互，并从交互过程中分析获取恶意代码样本的感染源位置信息，从而对这些恶意代码样本进行自动化地捕获和收集。基于这样的设计思想，Nepenthes 工具实现为 Linux 平台下的守护进程，Nepenthes Core 负责通过网络接口与恶意代码感染源进行交互，并协调其他各类组件模块共同完成恶意代码样本捕获和收集任务，目前 Nepenthes 工具中包含如下几类组件模块：1) 漏洞模拟模块：模拟各种已知的网络服务中存在的安全漏洞，如影响广泛的 LSASS 漏洞（MS04-011）、RPC-DCOM 漏洞（MS03-026）、ASN1 漏洞（MS04-007）等；2) Shellcode 分析模块：分析由漏洞模拟模块所接收的攻击负载（payload），并从中提取恶意代码感染源的位置信息；3) 获取模块：实现各种协议的恶意代码样本获取功能，通过提取的 URL 位置信息从远程位置下载恶意代码样本；4) 提交模块：实现各种形式的恶意代码样本提交功能，包括写入本地文件系统，写入数据库，提交远程收集服务器，以及提交给反病毒厂商进行扫描和分析等；5) 日志模块：记录恶意代码捕获过程中的信息，为进一步的恶意代码传播模式分析提供基础数据；6) 其他模块：包

括地理位置查询模块和端口监听模块等。

Nepenthes 工具作为 Mwcollect.org 组织主要的开源恶意代码捕获器，已经得到了较为广泛的认可和应用。Mwcollect.org 组织在 Nepenthes 工具基础上，已构建了恶意代码收集联盟，吸引了较多的反病毒厂商、研究团队参与该联盟。此外 Nepenthes 工具也被德国蜜网项目组[1]、JHU、ShadowSever.org 团队等应用于对僵尸网络的捕获和分析。

Nepenthes 工具存在的不足之处包括：对攻击未知漏洞的恶意代码无法进行有效捕获；需要部署方持续跟踪新发现的安全漏洞，并为其实现漏洞模拟脚本，维护工作量较大；捕获的恶意代码类型和数量受限于模拟的安全漏洞数量和质量。

3 HoneyBow——基于高交互式蜜罐技术的恶意代码捕获器

针对 Nepenthes 的不足之处，在本节将详细描述 HoneyBow 恶意代码捕获器，包括其组成结构以及如何利用高交互式蜜罐的思想来构建一个完全自动化的恶意代码样本捕获和收集方法，特别是针对未知的 0-day 恶意代码。HoneyBow 使用了真实的存有安全漏洞的服务来诱骗恶意代码感染，而非 Nepenthes 工具采用的模拟存有漏洞网络服务的方法。因此，HoneyBow 并不需要去调查网络服务安全漏洞的机理并实现其模拟版本，HoneyBow 背后的思想也就变得非常简单，它依赖于高交互式的蜜罐技术，部署 HoneyBow 恶意代码捕获器，我们只需要先在一个蜜网环境中部署一个或多个高交互式的蜜罐，并根据恶意代码捕获的需求定制这些蜜罐系统的补丁等级、所安装的网络服务和存有的已知安全漏洞等，然后在蜜罐系统及其他们的宿主操作系统上安装 HoneyBow 恶意代码捕获器的组件，就可以通过文件系统监控和交叉对比方法自动获取感染高交互式蜜罐的恶意代码样本，包括通过攻击未知安全漏洞感染蜜罐的 0-day 恶意代码。

3.1 HoneyBow 恶意代码捕获器组成结构

蜜网研究领域中存在 2 种普遍使用的部署高交互式蜜罐的方法，传统蜜网（traditional honeynet）与虚拟蜜网（virtual honeynet），传统蜜网使用物理硬件来安装蜜网网关和高交互式蜜罐系统，并使用物理的网络连接设备构建一个蜜网体系框架，而与之相反，虚拟蜜网则是一个在单独主机上安装整个蜜网体系的解决方案，部署虚拟蜜网最普遍的技术是使用虚拟机软件，如 VMware、VirtualPC 和 User Mode Linux 等。这 2 种高交互式蜜罐部署方法存有各自的优势和弱势，在实际的蜜网部署上也都得到普遍的使用。

HoneyBow 恶意代码捕获器同时支持这 2 种高交互式蜜罐的部署方式，如图 1 所示，HoneyBow 恶意代码捕获器由以下 3 个组件构成：MwWatcher、MwFetcher 和 MwSubmitter。

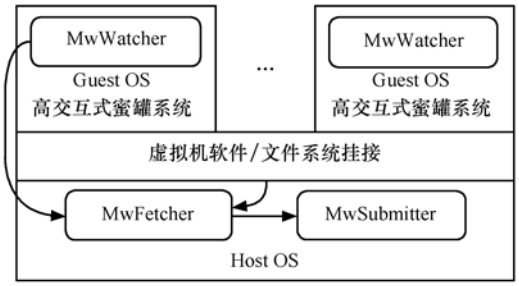


图 1 HoneyBow 恶意代码捕获起的组成结构

MwWatcher 是 HoneyBow 中实现的 2 个恶意代码捕获模块之一，它基于蜜罐系统的本质特性——“蜜罐系统没有任何的业务活动，因此蜜罐系统中发生的任何行为都预示着恶意”，通过实时监控蜜罐系统文件系统的变化，从而发现和捕获恶意代码样本。MwWatcher 在高交互式蜜罐系统中安装并运行，当互联网上传播的恶意代码攻击高交互式蜜罐系统上存有安全漏洞的网络服务，并感染蜜罐系统，在大多数情况下，恶意代码样本将会被传播并保存在蜜罐主机的文件系统中，MwWatcher 将通过截获文件系统调用，实时捕获新建的或修改的恶意代码样本文件。由于 Win32 平台恶意代码数量占据了流行恶意代码的绝大多数，因此目前 MwWatcher 只实现了 Win32 平台版本，而在其他操作系统平台如 Linux、FreeBSD 等，尽管具体实现机制不同，但仍可以借鉴 MwWatcher 中采用的实时文件系统监控机制进行实现。

MwFetcher 是 HoneyBow 中实现的另外一个恶意代码捕获模块，通过交叉对比受感染的蜜罐文件系统列表和之前保存的干净文件系统列表间的差异，提取可疑的恶意代码样本。MwFetcher 借鉴了在 Rootkit 检测领域中经典的交叉视图对比方法，但与其在同一时间选择底层和高层文件系统视图并进行比对的方法不同，MwFetcher 则是选取了感染前和感染后不同时间点上的底层文件系统视图

进行差异比对，通过差异比对 MwFetcher 能够完备地获取这段时间内感染蜜罐文件系统的恶意代码，包括对高层文件系统隐藏的 Rootkit。

MwSubmitter 则是将 MwWatcher 和 MwFetcher 捕获的恶意代码样本提交到恶意代码收集服务器的提交程序。作为客户端通过 Socket 连接恶意代码收集服务器上实现的 MwCollector 守护进程，提交每次恶意代码捕获实例和新的（未捕获过的）恶意代码样本。

MwFetcher 和 MwSubmitter 在宿主操作系统上安装并运行，目前均实现在 Linux 平台上。

在虚拟蜜网的部署方式中，容易理解宿主操作系统指的是宿主主机上所安装的操作系统，即虚拟机软件所在的操作系统，而高交互式蜜罐系统则使用虚拟机软件构建并安装，当这些蜜罐系统被恶意代码所感染后，可以周期性地、或按需地调用 MwFetcher 程序通过文件系统交叉对比方法捕获恶意代码样本(其中包括 MwWatcher 所实时捕获并存放在蜜罐文件系统中的恶意代码样本文件)，然后通过大部分虚拟机软件所支持的镜像恢复命令（revert）即可完成蜜罐系统的恢复，从而进入新的恶意代码捕获周期。

但在传统蜜网部署方式中，一旦物理蜜罐被恶意代码感染，由于恶意代码运行所带来的大量资源消耗，将使得物理蜜罐上的操作系统变得非常缓慢并可能导致无法响应，同时有些恶意代码感染主机后，会通过修补系统漏洞等方式排斥其他恶意代码。为了能够达成使用物理蜜罐进行自动化的恶意代码捕获，我们设计了一套基于智能平台管理接口（IPMI，intelligent platform management interface）和预启动执行环境（PXE，preboot execution environment）协议的针对物理蜜罐的自动恢复流程方案，该方案需要物理蜜罐使用具有支持 IPMI 接口主板和支持 PXE 启动网卡的硬件平台进行部署。在这种部署方式中，HoneyBow 恶意代码捕获器中的 MwFetcher 和 MwSubmitter 安装并运行在植入的 Linux 启动微内核中，在物理蜜罐自动恢复流程中首先调用 MwFetcher 恶意代码捕获模块，通过对 FAT32 或 NTFS 文件系统的挂接访问，从受感染的蜜罐操作系统文件系统中获取捕获的样本，并提交到中央恶意代码收集服务器。

3.2　HoneyBow 恶意代码捕获器的优势和局限

与基于低交互式蜜罐技术的 Nepenthes 恶意代码捕获器相比，HoneyBow 所具有的优势在于：1）对攻击未知安全漏洞的 0-day 恶意代码同样具有捕获能力，更加适用于计算机应急响应部门和反病毒厂商，进行大规模恶意代码爆发监测、恶意代码新变种的样本捕获等任务；2) 对 Downloader 类恶意代码下载并执行的二次注入代码样本也具有捕获能力；3) 无需调查安全漏洞的内部机制以及实现其模拟的漏洞模块，对部署方的技术水平要求程度较低，维护工作量小；4) 可通过定制高交互式蜜罐系统的补丁等级、安装的网络服务范围和修补补丁情况灵活的满足恶意代码捕获的不同范围需求。

同时 HoneyBow 也存在着较 Nepenthes 恶意代码捕获器的局限性，包括：1) 虽然 HoneyBow 恶意代码捕获器也可以通过在高交互式蜜罐系统上绑定多个 IP 地址等方法拓广其监测范围，但较 Nepenthes 能够同时绑定和监测 1 个/18 网段（大约 16 000 个 IP 地址）的高度可扩展性相比，由于高交互式蜜罐本身具有的特性，使得 HoneyBow 恶意代码捕获器无法达到同等级别的可扩展性。2) 以传统蜜网方式部署的 HoneyBow 恶意代码捕获器对硬件设备条件存在较强依赖，安装物理蜜罐系统的硬件设备必须拥有支持 IPMI 接口的主板和支持 PXE 启动的网卡才能实现完全自动化的恶意代码捕获流程，但目前只有较高端的服务器才支持 IPMI 接口，使用这种硬件设备的部署成本较高。若以虚拟蜜网方式部署，则存在虚拟机软件可能被恶意代码所识别的风险，虽然目前已经存在一些消除虚拟机软件（如 VMWare）指纹信息的补丁，但还无法完全抹去虚拟机软件的痕迹。3) HoneyBow 恶意代码捕获器只能够捕获成功攻陷安全漏洞并感染主机文件系统的恶意代码，而 Nepenthes 则能够从失败的破解攻击中同样分析出恶意代码下载源信息，从而对其进行获取。

HoneyBow 相比较于 Nepenthes 的优势和局限性的根源在于高交互式蜜罐技术和低交互式蜜罐技术两者本身各自存在的优势和劣势，因此在实际运用中，我们应充分地结合这两种恶意代码捕获方法，发挥各自的长处而抑制缺陷，以达到更好的自动化恶意代码样本捕获效果。

4　实验结果与分析

我们已经实际部署了一个基于分布式蜜网的

恶意代码自动捕获和监控系统，在此系统中，同时部署了 Nepenthes 和 HoneyBow 工具，因此可以实际数据对比两者在恶意代码样本捕获方面之间的性能差异。为了方便统计与保证数据之间的可比性，选取了分布式蜜网系统中 4 个同时部署和运行 HoneyBow 和 Nepenthes 的蜜网站点在 2006 年 12 月份的恶意代码捕获记录，进行统计对比分析。

如表 1 所示，4 个站点上部署的 HoneyBow 恶意代码捕获器共计有 40 077 次恶意代码捕获，消除重复后则为 13 140 个恶意代码样本，其中捕获后 Kaspersky 扫描标识为未知恶意代码的有 546 个 4 090 次，而 Nepenthes 则共计有 24 479 次恶意代码捕获，但是由于 Nepenthes 模拟脚本的单一性，这 24 479 次恶意代码捕获消除重复后仅有 683 个不同的恶意代码样本，在未知恶意代码捕获方面，Nepenthes 也仅捕获了 41 个 954 次，远小于 HoneyBow 的捕获量。

表 1　2006 年 12 月份 4 个站点 HoneyBow 和 Nepenthes 捕获情况对比

捕获情况	恶意代码捕获次数	恶意代码捕获个数	未知恶意代码捕获次数	未知恶意代码捕获个数
HoneyBow	40 077	13 140	4 090	546
Nepenthes	24 479	683	954	41

图 2~图 5 给出了其中一个典型站点的 HoneyBow 和 Nepenthes 的恶意代码捕获情况对比，HoneyBow 捕获的恶意代码次数平均在 410 次/天，略高于 Nepenthes 的 305 次/天，而在每日捕获的不同恶意代码样本个数方面，如图 3 所示，HoneyBow 大幅度领先于 Nepenthes，平均每日捕获 70 个不同样本，而 Nepenthes 则每日平均仅捕获 31.7 个，这一数据说明 HoneyBow 在恶意代码捕获的全面性方面要优于 Nepenthes，这也是 HoneyBow 所基于的高交互式蜜罐本身特性所决定的。此外，使用 Kaspersky 病毒引擎对捕获的恶意代码样本进行扫描，从而得到如图 4 和图 5 所示的未知恶意代码样本的捕获趋势，可以明显看出 HoneyBow 无论在未知恶意代码每日捕获次数和捕获个数（平均 103.3 次/天、15.3 个/天）方面均大大优于 Nepenthes（平均 49.3 次/天、2.3 个/天），这也充分体现了基于高交互式蜜罐技术的 HoneyBow 恶意代码捕获器在未知恶意代码样本收集方面的优势。

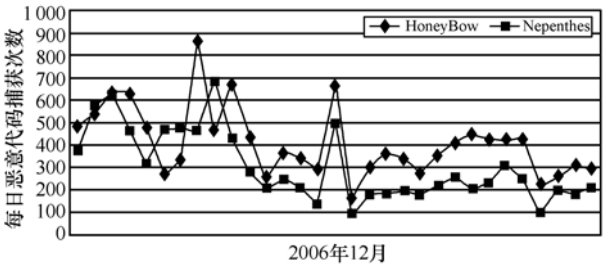


图 2　每日恶意代码捕获次数对比

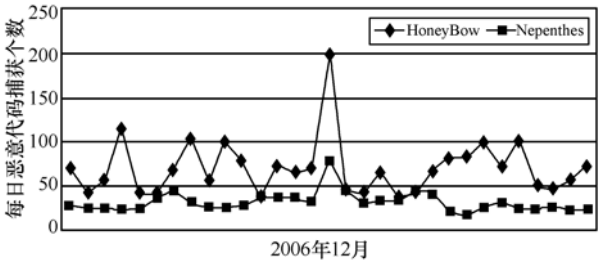


图 3　每日恶意代码捕获个数对比

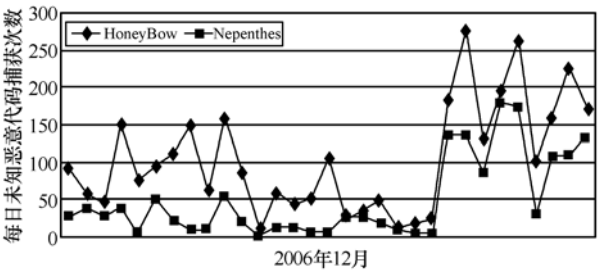


图 4　每日未知恶意代码捕获次数对比

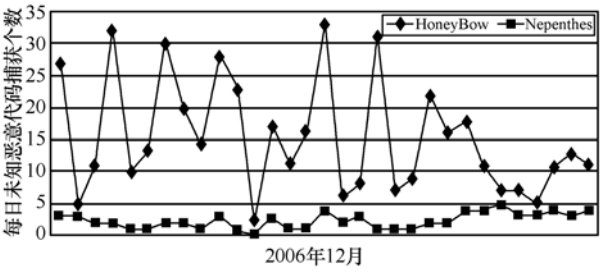


图 5　每日未知恶意代码捕获个数对比

对这些由 HoneyBow 和 Nepenthes 所捕获的恶意代码样本，进一步使用多个目前在市场上普遍应用的病毒扫描引擎(包括 Kaspersky, Kingsoft, Rising, Trend 和 BitDefender)进行恶意代码类型标识，从表 2 中可以看出 HoneyBow 捕获的恶意代码类型数量要多于 Nepenthes，但两者捕获恶意代码样本类型的并集分别较 HoneyBow 和 Nepenthes 有平均 22.87%和 78%的提升，这也说明了基于高交互式蜜罐技术的 HoneyBow 恶意代码捕获器和基于低交互式蜜罐技术的 Nepenthes 恶意代码捕获器具有一定的互补性，因此有必要在实际部署中结合 2 种恶意

代码捕获器进行更为全面的恶意代码样本收集和监控。

表 2　HoneyBow 捕获恶意代码与 Nepenthes 捕获恶意代码类型间的互补性

	AV Engine 1	AV Engine 2	AV Engine 3	AV Engine 4	AV Engine 5
HoneyBow 捕获类型数量	178	233	247	300	139
Nepenthes 捕获类型数量	113	194	180	233	76
两者并集	213	290	324	361	165

在 2006 年 8 月份 Mocbot 蠕虫的应急响应处理事件中，在互联网上实际部署的恶意代码样本自动捕获体系也起到了关键性作用，当 Mocbot 蠕虫爆发初期，基于高交互式蜜罐技术的 HoneyBow 恶意代码捕获器在北京时间 2006 年 8 月 13 日凌晨 3 点 54 分即获取了该蠕虫的样本，而该样本在 2006 年 8 月 13 日凌晨 4 点 10 分第一次调用 Kaspersky 病毒扫描引擎时的扫描结果为未知(Kaspersky 病毒扫描引擎在 8 月 16 日(捕获三日后)的扫描中识别 Mocbot 蠕虫为 Backdoor.Win32.IRCBot.uv)，Nepenthes 由于未实现 Mocbot 蠕虫所利用的 MS06-040 安全漏洞模拟脚本，因此没有及时捕获该蠕虫样本。

5　结束语

本文关注于使用蜜罐技术构建恶意代码自动捕获和收集体系，引入了基于高交互式蜜罐技术的 HoneyBow 恶意代码自动捕获器。与基于低交互式蜜罐技术 Nepenthes 相比，HoneyBow 在捕获恶意代码全面性较高、能够捕获未知恶意代码、部署维护代价较小以及部署方式更为灵活等方面存在优势，而在部署的可扩展性等方面存在劣势。我们进一步对实际部署蜜网站点的恶意代码捕获技术进行了统计分析，结果表明，HoneyBow 恶意代码捕获器在捕获样本的全面性、对未知恶意代码样本的捕获能力方面要强于 Nepenthes，HoneyBow 在第一时间对 2006 年 8 月大规模爆发的 Mocbot 蠕虫的及时捕获也体现了 HoneyBow 在快速传播型恶意代码应急响应中的关键作用。同时 HoneyBow 和 Nepenthes 两者之间捕获的样本类型存在一定的互补性，这体现了在实际部署中结合两者的必要性。

参考文献：

[1] BAECHER P, HOLZ T, KOETTER M, *et al*. Know your enemy: tracking botnets, using honeynets to learn more about bots[EB/OL]. http://www.honeynet.org/papers/ bots/, 2005. Accessed March 2007.

[2] WATSON D, HOLZ T, MUELLER S. Know your enemy: phishing[EB/OL]. http://www.honeynet. org/papers/phishing/,2005. Accessed March 2007.

[3] PROVOS N. A virtual honeypot framework[A]. Proceedings of 13th USENIX Security Symposium[C]. San Diego, CA, USA, 2004. 1-14.

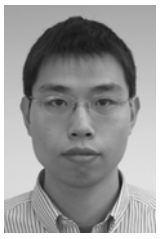
[4] BALAS E, VIECCO C. Towards a third generation data capture architecture for honeynets[A]. Proceeedings of the 6th IEEE Information Assurance Workshop[C]. West Point, NY, USA, 2005.

[5] BAECHER P, KOETTER M, HOLZ T, *et al*. The nepenthes platform: an efficient approach to collect malware[J]. Lecture Notes in Computer Science 4219, 2006, 165-184.

[6] ZIMMER D. Multipot[EB/OL]. http://labs.idefense.com/software/malcode. php, 2006. Accessed March 2007.

[7] LEVINE J, GRIZZARD J, OWEN H. Application of a methodology to characterize rootkits retrieved from honeynets[A]. Proceedings of the Fifth Annual Information Assurance Workshop[C]. West Point, NY, USA, 2004. 15-21.

作者简介：



诸葛建伟（1980-），男，浙江瑞安人，博士，北京大学助理研究员，主要研究方向为入侵检测与关联、蜜罐与蜜网技术、网络攻防技术、恶意代码分析与防范。

韩心慧（1969-），男，河南开封人，北京大学博士生、高级工程师，主要研究方向为网络与信息安全。

周勇林（1974-），男，北京人，国家计算机网络应急技术处理协调中心博士生、高级工程师，主要研究方向为互联网安全监测、应急响应处理。

宋程昱（1984-），男，云南昆明人，北京大学硕士生，主要研究方向为恶意代码分析与防范。

郭晋鹏（1982-），男，山西阳泉人，北京大学工程师，主要研究方向为蜜罐与蜜网技术、恶意代码分析与防范。

邹维（1964-），男，重庆人，硕士，北京大学研究员，主要研究方向为网络与信息安全。