doi:10.3969/j.issn.1000-436x.2013.z1.026

基于分布式的僵尸网络主动探测方法研究

司成祥[1]，孙波[1]，杨文瀚[2]，张慧琳[2]，薛晓楠[2]
(1. 国家计算机网络应急技术处理协调中心，北京 100029；2. 北京大学 计算机科学技术研究所，北京 100871)

摘 要：僵尸网络是当前互联网上存在的一类严重安全威胁。传统的被动监控方法需要经过证据积累、检测和反应的过程，只能在实际恶意活动发生之后发现僵尸网络的存在。提出了基于僵尸网络控制端通信协议指纹的分布式主动探测方法，通过逆向分析僵尸网络的控制端和被控端样本，提取僵尸网络通信协议，并从控制端回复信息中抽取通信协议交互指纹，最后基于通信协议指纹对网络上的主机进行主动探测。基于该方法，设计并实现了 ActiveSpear 主动探测系统，该系统采用分布式架构，扫描所使用的 IP 动态变化，支持对多种通信协议的僵尸网络控制端的并行扫描。在实验环境中对系统的功能性验证证明了方法的有效性，实际环境中对系统扫描效率的评估说明系统能够在可接受的时间内完成对网段的大规模扫描。

关键词：僵尸网络；控制端；主动探测；分布式；协议分析
中图分类号：TP309 **文献标识码**：B **文章编号**：1000-436X(2013)Z1-0197-10

Active-probing based distributed malware master detection system

SI Cheng-xiang[1], SUN Bo[1], YANG Wen-han[2], ZHANG Hui-lin[2], XUE Xiao-nan[2]
(1.CNCERT/CC, Beijing 100029, China; 2.Institute of Computer Science and Technology, Peking University, Beijing 100871, China)

Abstract: Nowadays, botnet is still a kind of severe threat on the Internet. It wastes lots of time for traditional passive monitoring approaches to collect enough evidence, to detect and react. Only after real malicious activities occur can we find the existence of botnet. An active probing approach was proposed based on botnet controller's communication protocol fingerprint. Botnet samples including client and server were analyzed and the command and control protocol of the botnet were collected. The communication protocol fingerprint was also extracted from controller's response message and the host on the Internet was scanned with the communication protocol fingerprint. Active Spear active probing system was designed and implemented based on the approach. The system employs distributed architecture and IP used in the scanning is dynamic. The system supports to scan many botnets owning different types of protocols as their command and control protocols. The functional verification in the testing environment proves the effectiveness of the approach and the evaluation to scanning efficiency in the real network environment shows the ability that the system can finish task of scanning a large scale of IP section in an acceptable time.

Key words: botnet; server; active probe; distributed system; protocol analysis

1 引言

近年来，计算机、互联网成为人们日常生活不可或缺的一部分，与此同时，信息安全形势却越来越严峻，攻击者在传统恶意代码形态（包括计算机病毒、网络蠕虫、特洛伊木马和后门工具）的基础上融合、进化，开发出可以受控的木马程序，被木马感染的主机所组成的僵尸网络成为了个人隐私泄露、泄密、垃圾邮件和大规模拒绝服务攻击的重要原因之一[1]，已经成为互联网目前最严重的安全威胁。

中国国家互联网应急中心于 2013 年 2 月发布

收稿日期：2013-07-30
基金项目：国家 242 信息安全计划基金资助项目（2011A40）；国家自然科学基金资助项目（61003127）
Foundation Items: The National 242 Information Security Research Program of China (2011A40); The National Natural Science Foundation of China (61003127)

的网络安全信息与动态周报第四期[2]中指出，中国大陆被僵尸网络控制的主机约为 33.9 万个，感染 Conficker 病毒（一种 P2P 僵尸病毒）的主机 IP 约为 81.3 万个，其中广东、江苏和山东是僵尸病毒感染最多的地区。面对僵尸网络对网络安全的严重威胁，如何有效地防止和应对该类安全威胁已经成为了学术界和工业界的关注热点。

现有的研究工作大多数采用被动的检测、监控和追踪技术，无论是在主机层基于程序指纹的检测或基于程序行为分析的检测，还是在网络层的流量监控或基于主机行为时空关联性的检测方法，都存在 2 个缺点：1）“被动”意味着只有在僵尸网络发生实际恶意活动之后系统才可能发现；2）对于小规模使用私有协议作为命令与控制协议的僵尸网络，上述方法基本无效。

上述缺点使研究人员将目光投向主动式的检测和监控方法。文献[3,4]中分别提出了通过协议模拟进行主动探测的检测方法，前者检测以 IRC 协议作为控制与命令信道的僵尸网络，后者检测以 P2P 协议作为控制与命令信道的僵尸网络。2 种方法仍存在不足：1）检测针对单个协议，对其他协议的僵尸网络束手无策；2）检测能够发现被控主机，但不能直接发现控制端，威胁源头仍存在于网络中。

本文提出基于分布式的僵尸网络主动探测方法，力求克服上述不足，实现：1）具有在恶意活动发生之前发现僵尸网络控制端的能力；2）支持对以不同协议作为命令与控制信道的僵尸网络的扫描；3）检测能直接发现僵尸网络控制端。

该方法的基本假设是：1）控制端必须监听端口，向 bot 提供服务，如：下载二进制程序、发送最新活动命令等。如果能够确定某开放端口上运行服务的通信交互逻辑，那么就能唯一地标识该服务为某一僵尸网络控制服务，即确定该主机是某僵尸网络控制端。2）由于僵尸网络的通信交互逻辑在其传播时就已经固定，而僵尸网络形成一定的规模需要较长时间，黑客为了尽可能多地获得被控主机，会使对控制端服务的访问和通信交互逻辑在很长一段时间内保持稳定。

基于上述假设，本文设计了一个 2 阶段的检测框架：1）首先，在主机层和网络层使用半自动化的分析工具对僵尸网络的控制端和被控端样本进行分析，获取通信交互逻辑，提取通信交互过程中控制端返回的通信指纹；2）模拟所获取的通信交互逻辑，对目标主机的目标端口发起主动连接，在通信交互过程中尝试匹配所提取的通信指纹，从而识别该目标主机是否为某僵尸网络控制端。由于长期的、大规模的主动探测有可能会触发安全设备的误报或引起黑客的警觉，构建了一个分布式且 IP 动态变化的主动探测系统，将主动探测的流量分散，在提高效率的同时，降低了探测过程中网络被阻断的概率。

实验结果表明，相比之前的方法，本文的方法有 4 方面的优势：1）相比于被动的检测方法，本方法具有在僵尸网络进行大规模破坏之前发现僵尸网络控制端的能力；2）相比于主机级的基于程序指纹识别的检测方法，基于通信协议指纹识别僵尸网络更加准确有效；3）由于系统不需要在主机一级部署，因此更利于大规模部署实施、管理、升级；4）动态变化的 IP 和分布式扫描框架增加了扫描的效率，将流量分散，降低了网络被阻断的概率。

2 相关工作

2.1 网络层的僵尸网络检测

网络层的僵尸网络检测具有效率高、不需要大规模部署的优点。该类方法分为 2 类，一类通过网络流量监控发现网络活动中异常的流量、模式和结构，如文献[5]提出的基于机器学习的僵尸网络检测方法，该方法选取了几个通用的网络级流量特征描述网络聊天协议产生的流量；文献[6]研究了在主干网对 IRC 僵尸网络控制端进行网络流的检测；文献[7]设计并实现了通过追踪 IRC 僵尸程序昵称模式的基于指纹的 IRC 僵尸网络监测系统；文献[8]提到的 BotHunter 使用 IDS 对话关联将 IDS 事件和僵尸感染对话模型关联在一起。

另一类通过被控主机通信流量的相似性检测网络中僵尸网络的存在。如 BotSniffer[9]和 BotMiner[10]通过平行关联进行的时空相似性分析；TAMD[11]从目标地址、流量内容和平台 3 方面的相似性对网络流量进行聚合进行检测；文献[12]提出使用熵和基于机器学习的方法检测聊天室僵尸程序。

本文检测僵尸网络的方法属于网络层的僵尸网络检测，但并不归入上面 2 类，原理属于网络扫描的范畴（在 4.1 节中详述）。

2.2 主动探测的僵尸网络检测

由于被动监控的方法存在的不足，文献[3,4]提出了对 IRC 协议僵尸网络和 P2P 僵尸网络的主动探测方法。前者检测 IRC 信道内的僵尸程序，利用僵

尸程序在发送与接收指令时对字符错误的零容忍和人在聊天时对字符或语法错误所具有的强矫正能力来检测信道中参与聊天的对象是人还是僵尸程序，并使用假设检测和重复试验的方法，能够将系统的理论错误率限制在任意的精度范围内。后者检测 P2P 僵尸程序，首先使用污点分析、符号执行、路径检索等方法破解恶意程序的 MCB，之后主动连接指定的 IP 和端口，并使用破解得到的 MCB 进行主动探测，一旦通信交互过程中对方主机的回应与 MCB 吻合，则认为该主机是一个被控主机。本文主动探测方法的思想与文献[4]相似，不同之处在于：1）本文的方法支持对多种协议僵尸网络的检测；2）本文主动探测的对象是僵尸网络控制端而非被控主机。

2.3　通信协议逆向工程

许多研究者使用自动化协议逆向工程技术抽取未知和无文档的应用层网络协议。根据逆向的层次和目的，自动化协议逆向工程技术可以分为 3 类：1）分析一条信息，抽取域的结构[13~15]；2）分析多条信息，抽取通信交互协议的格式[16~18]；3）推导与协议等价的状态机[19,20]。根据输入的源，可以将其分为 2 类：1）以网络流量作为输入[14,16,20]；2）以执行踪迹作为输入[13,15,17~19]。一个详细的协议规范能显著提高许多安全应用的实用效果，如模糊测试、深入的分组分析等。文献[21]设计并实现了 Dispatcher，支持从应用发送的信息中抽取协议的格式和实现规范，推导出应用发送和接收信息中各个域的含义，并且支持对协议进行重写，而且允许进一步对活跃僵尸网络进行渗透。本文对样本通信交互协议逆向分析的目的是通过分析多条信息，抽取通信交互协议的格式，输入源为协议交互的网络流量。具体地，只关心通信交互协议的前几步，协议逆向的最终目的是获取能够在主动探测中唯一识别僵尸网络控制端服务的通信交互协议指纹。以下 2 种情况会导致自动化协议逆向工程技术在解决该类问题时失败：1）网络通信仅包含有限的语义信息；2）加密和混淆后的流量无法破解。因此，本文在实际逆向分析过程中，会结合污点分析、恶意软件动态调试等技术辅助理解僵尸网络控制端网络通信协议以及通信协议的加密方式。

3　方法概述

3.1　问题定义

假设：假设能够得到僵尸网络控制端和被控端的二进制文件。

定义样本 P 的通信协议指纹为一个元组：

$P=<L,N,S_1(.),R_1(.),S_2(.),R_2(.),\cdots,S_M(.),R_M(.)>$

其中，

L：被控端向控制端发送通信消息所依赖的变量；

N：通信协议指纹交互的长度（通信步数）；

$S_i(\cdot)$：第 i 次通信过程中被控端向控制端发送的消息；

$R_i(\cdot)$：第 i 次通信过程中控制端向被控端发送的消息。

举例说明上述定义及实例，具体如下。

$P=<\{IP,hostname\}, 2, S_1(.),R_1(.),S_2(.),R_2(.)>$

S_1(IP,hostname)=*IP*+“:”+hostname+“,aaa”

R_1(IP,hostname)=*IP*+“:”+hostname+“,AAA”

S_2(IP,hostname)=“bbb,”+*IP*+“:”+hostname

R_2(IP,hostname)=“BBB,”+*IP*+“:”+hostname

被控端 s 的 IP 地址为 172.30.40.10，hostname 为 inspur，s 与控制端 S 的交互过程为：

1) s 向 S 发送上线消息

172.30.40.10:inspur,aaa；

2) 控制端 S 接收到

172.30.40.10:inspur,aaa

控制端 S 向被控端 s 发送消息

172.30.40.10:inspur,AAA；

3) 被控端 s 接收收到

172.30.40.10:inspur,AAA

被控端接 s 向控制端 S 发送消息

bbb,172.30.40.10:inspur；

4) 控制端 S 接收到

bbb,172.30.40.10:inspur；

5) 控制端向被控端发送

BBB,172.30.40.10:inspur。

特别地，如果控制端首先发送上线信息，定义 S_1=Ø，如果被控端首先发送上线信息为空串，则 S_1=空串。

3.2　方法概述

所使用的方法如图 1 所示。

第一步，在隔离的安全分析环境中分析僵尸程序样本，提取用于识别控制端服务的通信协议指纹。使用动态分析工具分析样本，观察僵尸网络控制端和被控端的通信行为，提取通信交互协议，从控制端回复的反馈信息中抽取通信协议指纹。如果

通信交互协议是加密的且 L 不为空（通信内容依赖于被控端机器的某些信息），则使用自动化的工具和深度的人工分析进行进一步分析。分析完后，得到识别僵尸网络控制端的通信协议指纹。

第二步，在中心管控节点的调度下，分布式扫描节点使用第一步提取的通信协议指纹对网络上的主机进行主动探测。对于指定的 IP 和 Port，如果主机活跃且端口开放，则使用通信协议指纹逐条探测端口上是否开放了控制端服务。扫描发起连接使用动态 IP。扫描完毕后，扫描节点将扫描结果提交给中心管控节点。

4 关键技术

4.1 网络扫描与端口服务识别

网络扫描指通过网络通信探测远端网络或主机信息的一种技术。根据扫描的对象和目的进行分类，常见的扫描可划分为主机活跃性扫描、端口扫描、操作系统探测、服务识别扫描、漏洞扫描、防火墙规则刺探 6 种。本文工作涉及的内容包括主机活跃性扫描、端口扫描、服务识别扫描。

主机活跃性扫描判断某个 IP 或域名是否有主机开启，扫描通过发送探测分组到目标主机，如果收到回复，说明目标主机是开启的。端口扫描判断一台主机上的某些端口是否开放，以 TCP SYN 方式的端口扫描为例，该方式发送 SYN 到目标端口，如果收到 SYN/ACK 回复，那么判断端口是开放的；如果收到 RST 分组，说明该端口是关闭的；如果没有收到回复，那么判断该端口被屏蔽。服务识别扫描确定目标主机开放端口上运行的具体应用程序及版本信息，通过模拟某种服务的通信协议进行网络通信，如果发现对方主机的响应与该服务的通信协议指纹一致，说明扫描的端口上开放了该服务。

4 种扫描在使用时存在顺序依赖的关系，首先需要进行活跃性扫描，随后确定端口是否开放，最后确定端口上运行具体应用程序与版本的信息。

本文提出的基于通信协议指纹的主动探测属于服务识别扫描。与一般的服务识别扫描情形不同的是：主动探测识别的服务对象是僵尸网络控制端服务。

4.2 样本分析与通信协议指纹提取

通过分析僵尸网络样本、提取恶意服务的通信协议指纹是进行主动探测的前提。采用专家手工分析与自动化工具相结合、主机层和网络层相结合的方法进行样本分析。分析过程使用了动态行为分析、网络抓包、恶意软件调试、静态分析、污点分析和模糊测试等技术提取用于识别僵尸网络控制端的通信协议指纹。

本文使用了恶意代码动态行为监控工具对僵尸网络控制端的动态行为进行分析，工具对应用层 API 接口进行劫持（API inline hooking），同时对 Kernel 层 API 接口进行劫持（SSDT hooking），实现了完备的恶意代码动态行为监控。工具记录了控制端程序和被控端程序完整的通信过程，通过分析交互过程可以直接提取出通信未加密的样本和通信加密但前几步协议交互内容不变化样本的通信协议指纹。如 gh0st 程序的交互指纹为 P={Ø, 1, Ø,

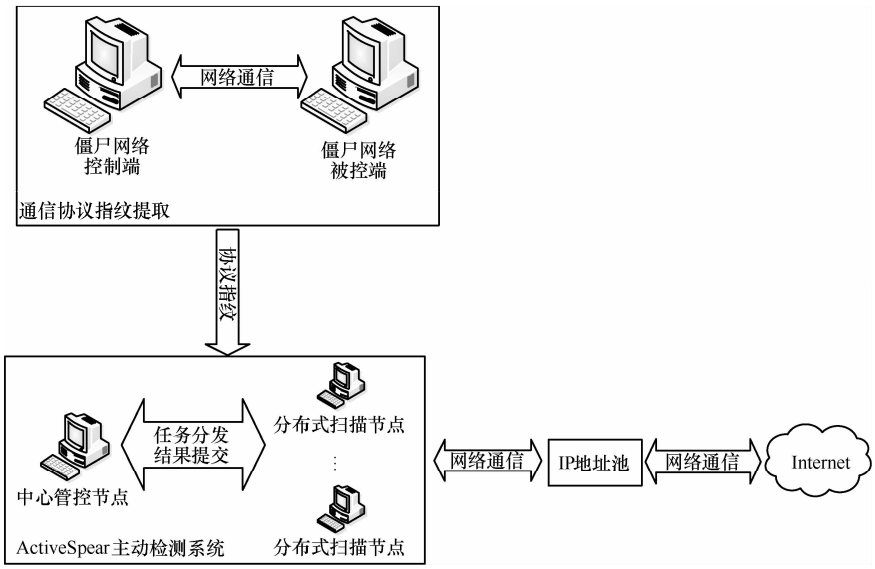


图 1 分布式的僵尸网络主动探测方法

R_1(.)=gh0st}。

对于少数隐蔽技术较高的僵尸网络程序，恶意代码动态行为监控工具无法截获其通信行为。这一情形下，通过抓分组分析捕获这些样本的通信分组，同样分析其通信记录可获取通信协议指纹。

对于通信加密且前几步协议交互内容变化的控制端程序，必须通过恶意软件调试和逆向分析技术找到其通信部分的代码，在加密前截获待加密的字符串，并识别加密函数。使用污点分析技术和静态分析技术可以辅助加快分析过程。污点分析通过标定发送至网络的数据或者从网络接收的数据作为“被污染”的源，由此产生的一系列算术和逻辑操作新生成的数据也会继承源数据的“是否被污染”的属性。污点分析标定的“被污染”的数据缩小了需要调试的代码范围。在静态分析中，通过锁定几个与网络通信相关的系统调用和 API 调用进一步缩小该范围，从而加快了人工分析的速度。

在分析的样本之前，使用模糊测试的方法提取样本的通信协议特征。将几个流行的扫描工具（如 nmap，scapy）中默认的服务识别指纹的试探分组发送到样本绑定的 IP 和端口上，如果对于某个特定的试探分组，端口上绑定的服务以某一特定的模式回应，那么可以确定该样本的通信协议指纹。

因此，根据通信是否加密、隐蔽技术以及前几步交互是否发生变化，可以将僵尸网络控制端程序分为 3 类，将提取 3 类样本通信协议特征的方法和技术归纳如表 1 所示。

表 1　　样本通信协议特征提取方法

类别	特点	提取方法
1	通信未加密或加密了协议交互前几步不变化	恶意代码动态行为监控
2	通信加密且协议交互前几步变化	恶意软件调试、静态分析、污点分析
3	能够躲避 API hook 和 SSDT hooking	抓分组分析+上述方法

5　系统实现

5.1　系统框架

如图 2 所示，ActiveSpear 基于分布式的僵尸网络主动探测系统搭建在 CNCERT\CC 提供的通用云平台上，该平台搭建在一个 B 类网段的 IP 地址池之上，由一个中心管控节点和超过 500 个分布式扫描节点组成，扫描节点每次进行网络通信都会随机分配一个 IP 和 Port。中心管控节点控制、管理所有的分布式扫描节点，进行任务下发和扫描节点调度，提供用户交互界面，提供对任务进度情况、主动探测结果、运行结果导出等内容的查询和统计功能。分布式扫描节点与中心管控节点交互获取中心管控节点分配给自己的任务，执行基于通信协议指纹的主动探测，最后将扫描结果上报给中心管控节点。

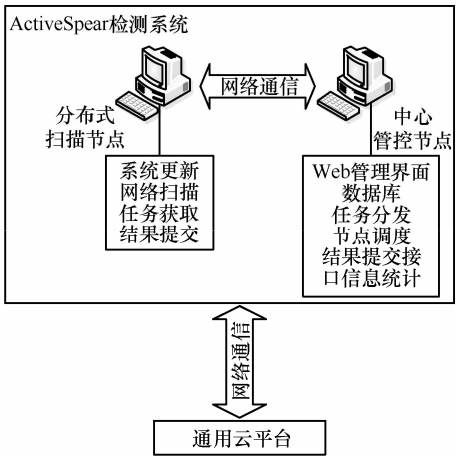


图 2　ActiveSpear 主动扫描系统框架

5.2　分布式扫描节点

5.2.1　工作流程

分布式扫描节点模拟僵尸网络被控端，进行基于通信协议指纹的主动探测，其工作流程如图 3 所示。每隔一段时间，系统会从中心管控节点获取分配给自己的任务，之后进行网络扫描（依次进行主机活跃性扫描、端口扫描、基于通信协议指纹的主动探测），最后向中心管控节点扫描的结果。

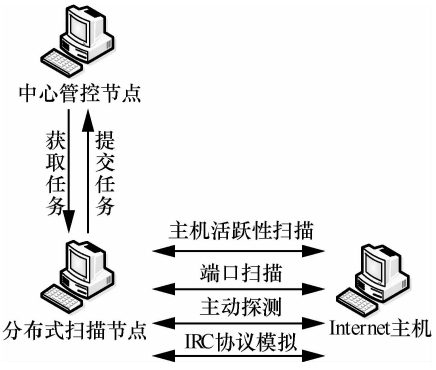


图 1 分布式扫描节点工作流程

5.2.2　主动探测

本文实现了支持发送 TCP SYN、TCP ACK、

SCTP INIT 等 8 种不同类型的分组进行主机活跃性扫描、进行 TCP SYN、TCP connect、TCP ACK 或 TCP Fin 这 4 种类型的端口扫描，使用 TCP 协议进行基于通信协议指纹的主动探测的并行扫描系统。基于通信协议指纹的主动探测过程如图 4 所示。

```
1) 发起 TCP 连接;
2) 等待片刻（S1=Ø）;
3) If 接收到目标主机发送的“WelcomeBanner”信息（R1） Then
        匹配通信协议指纹库中的指纹;
         If match:
              返回僵尸网络控制端的名字和版本信息;
         End IF
   End IF
4) 发送通信协议指纹库中的其他探测分组（S2）;
5) 等待片刻，接收回复分组;
6) If 接收到回复分组（R2） Then
        匹配通信协议指纹库中的指纹;
        If match:
             返回僵尸网络控制端的名字和版本信息;
        End IF
End IF
7) …
继续发送其他探测分组，将回复分组与指纹库中的指纹比较
…
8) 探测未识别任何具体的僵尸网络控制端，打印出服务返回的报文。
```

图 4　基于通信协议指纹的主动探测伪码

5.2.3　协议指纹匹配

需要特别说明的是，扫描过程中发送的数据分组完全按照 S_i 的规则精确生成，而在对回复报文 R_i 进行匹配的时候，为了使系统在通信协议交互出现微小变化情况（比如变种修改了通信协议）下仍能正确识别僵尸网络控制端，对通信协议指纹进行如下处理。

1) 协议指纹泛化。将 R_i 抽象成正则表达式的形式，匹配算法采用正则表达式匹配的算法。如 R_i 为 <MSG>006</MSG>，将指纹定义为^<MSG> [0-9]{2-4}</MSG>，该表达式匹配这样的字符串：以<MSG>开头，且在<MSG>和</MSG>之间包含 2 到 4 个 0 到 9 的数字组成的字符串。

2) 协议指纹分拆。对于通信加密但前几步交互不变的样本，将对应的通信协议指纹 P 中的 R_i 拆成均能大概率唯一标定该样本的几段，在匹配时只要匹配上其中某一段，即认定目标主机上开放了僵尸网络控制端服务。比如，R_i 为 01aec34jdf8jfdg4545，匹配的正则表达式为 01aec34jdf|34jdf8jfdg|fdg4545。

5.2.4　深度协议模拟

IRC 协议和 HTTP 协议是当前许多流行的僵尸网络命令与控制信道的通信协议，如 GT-Bot、Sdbot、Agobot、Spybot、Bobax、Rustock、Clickbot 等。由于这 2 类僵尸网络命令与控制信道的实现采用标准协议，仅通过服务识别并不能将僵尸网络控制端与正常应用的服务端区分开，因此本文方法在识别出端口运行的服务之后，进一步进行深度协议模拟，通过探测服务应用的逻辑判断目标主机是否为僵尸网络控制端。

IRC 协议模拟。对识别出运行 IRC 服务的端口，扫描节点通过 IRC 协议模拟接入到 IRC 服务器中，之后通过/List 命令查看是否存在 IRC 僵尸网络控制端常用的通信信道（如：#IRCbot、#SDbot）。如果存在，则认为该信道是僵尸网络命令与控制信道。

HTTP 协议模拟。对识别出运行 HTTP 服务的端口，扫描节点根据样本分析的结果构造 HTTP 请求，通过分析返回的 HTTP 页面判断对方主机是否为僵尸网络控制端。比如：分析发现某僵尸网络被控主机向控制端发送 Get 请求：http://domain/dirpath?request，控制端返回的 HTTP 页面中包含僵尸控制者下发的命令（scan 76.45.21.*），则针对该僵尸网络控制端的主动探测方法为：对识别出运行 HTTP 服务的端口，发送 Get 请求：http://ip:port/dirpath?request，看返回的页面是否包含命令 scan 76.45.21.*。如果包含，则对方主机为 HTTP 僵尸网络控制端。

5.3　中心管控节点

中心管控节点控制、管理所有的分布式扫描节点，进行任务下发和扫描节点调度，提供用户交互界面，提供对任务进度情况、主动探测结果、运行结果导出等内容的查询和统计功能。其中，中心管控节点任务下发的策略对扫描效率和扫描是否有可能造成网络拥塞产生影响。

5.3.1　扫描端口顺序

当前的僵尸网络控制端程序已经工具化，黑客在生成被控端的时候，能够随意配置被控端回连的端口，一台主机上的任何端口都可能成为控制端服务监听的端口。而相比于端口总数（65 536），一般主机开放的端口数量是很小的，因此，无序地扫描海量的端口是一件费时而无效率的事情。为了提高

扫描中开放端口的概率，提高扫描效率，将扫描端口分为 3 类。

1) 常用端口。一些知名的服务（比如：http、ssh、ftp 等）默认监听的端口，这些服务大量存在于互联网上，防火墙一般不会过滤掉这些端口，其开放的概率远远高于其他端口。

2) 僵尸网络控制端默认端口。僵尸网络控制端软件大多可以在生成被控端前配置反向连接的端口，但相信总存在一些黑客不修改这些控制端的默认端口号。因此，扫描这些端口发现僵尸网络控制端服务的概率大于其他端口。

3) IRC 端口和 Web 服务端口。很多僵尸网络采用 IRC 和 HTTP 协议构建控制与命令信道，因此 6666、6667、6679、80 和 8080 端口在扫描中给予更高的优先级。

对于需要扫描的网段，按 3)、2)、1）的次序依次扫描网段的对应端口，最后扫描网段内其他未扫描过的端口。

5.3.2 任务拆分策略

对于给定网段和端口范围的扫描任务，中心管控节点首先将任务分拆成许多子任务，之后将子任务分发给分布式扫描节点进行扫描。

为了防止系统在较短时间内对较小范围的网段进行大规模访问，引起网络拥塞或触发安全设备误报，系统采取如下拆分的策略：在一次任务中，每个扫描节点最多扫描 256 个 IP 地址，扫描的端口范围最多包含 50 个端口，拆分时优先从网段低位开始分拆，尽量避免同一网段的 IP 被分到一起。例如，扫描开头为 64 的 A 类网段，端口范围为 1~1 000，则分拆的子任务会被表示成许多如 64.*.23.54:51-100 的形式。

单个扫描节点对子任务进行扫描时，系统扫描 IP 和端口的顺序是随机的，使用临时从 IP 地址池中获得的一个随机 IP 和 Port 发起连接。系统在指定最大连接数的限制范围内并发地向多个IP和Port同时发起连接，提高扫描的效率。

6 实验与评估

本节使用真实的僵尸网络控制端对系统的有效性进行验证。这些控制端包括 IRC 僵尸网络控制端、HTTP 僵尸网络控制端和私有协议僵尸网络控制端。所有僵尸网络控制端和命令与控制信道协议类型如表 2 所示。

表 2　控制端和命令与控制信道协议类型

名称	协议类型	名称	协议类型
狐组远程控制	私有协议	Syla RAT	私有协议
黑太阳	私有协议	Trojan Nunks	私有协议
白金远程控制	私有协议	大白鲨远控	私有协议
DRAT	私有协议	哈迪斯远程协助	私有协议
51Remote	私有协议	华中帝国	私有协议
DarkStRat	私有协议	小花匠	私有协议
Freerat	私有协议	一笑江湖	私有协议
Nova Lite Rat	私有协议	终结者	私有协议
PaiN RAT	私有协议	Agobot	IRC 协议
S-xrat	私有协议	Rbot	IRC 协议
PcShare	私有协议	SdBot	IRC 协议
IRCbot	IRC 协议	poebot	IRC 协议
Phatbot	IRC 协议	Zeus	HTTP 协议
loic	IRC 协议	HFS malware download	HTTP 协议

6.1 通信协议分析

抽取了表 2 中僵尸网络通信协议交互的前几步，并根据控制端的回复信息提取主动探测时匹配的通信协议指纹。根据表 1 的分类，可以将上述控制端分类，如表 3 所示。

分析结果表明，大多使用私有协议作为命令与控制信道的僵尸网络属于第一类，容易提取通信协议指纹并应用于系统的主动探测中。匹配使用 IRC 协议作为命令与控制信道的僵尸网络控制端的通信协议指纹，是接入 IRC 服务器后发送 List 命令预期收到的回复；匹配使用 HTTP 协议作为命令与控制信道的僵尸网络控制端的通信协议指纹，是发送特定的 HTTP Get 请求后预期收到的回复页面。

6.2 靶机环境测试

为了验证系统功能的正确性，本文在互联网上部署了一台靶机，在该主机的 16 个常用端口上分别开放了 20 种不同的僵尸网络控制端服务（5 个 IRC 僵尸网络控制端运行在 6 667 端口提供的 IRC 服务的信道中），僵尸网络控制端服务与端口对应情况如表 4 所示。指定系统对包含该靶机的 B 类网段进行扫描，扫描的端口范围仅针对常用端口。扫描从 2013 年 3 月 18 日上午 9 点 12 分开始，于下午 13 点 17 分结束，扫描结果显示系统成功地检测了所有预置的僵尸网络控制端，说明系统能在短时间内成功检测分析过的僵尸网络控制端。

表 3　　控制端通信协议指纹及分类

名称	协议类型	通信协议指纹	名称	协议类型	通信协议指纹
狐组远程控制	私有协议	第一类	Syla RAT	私有协议	第一类
黑洞远程控制	私有协议	第一类	Trojan Nunks	私有协议	第一类
凝瑞远程控制	私有协议	第一类	大白鲨远控	私有协议	第二类
DRAT	私有协议	第一类	哈迪斯远程协助	私有协议	第二类
上兴远程控制	私有协议	第二类	华中帝国	私有协议	第三类
FeiMooMa	私有协议	第一类	小花匠	私有协议	第二类
Freerat	私有协议	第一类	一笑江湖	私有协议	第一类
Krist	私有协议	第一类	终结者	私有协议	第一类
PaiN RAT	私有协议	第一类	Agobot	IRC 协议	IRC List response
波尔远程控制	私有协议	第一类	Rbot	IRC 协议	IRC List response
PcShare	私有协议	第二类	SdBot	IRC 协议	IRC List response
IRCbot	IRC 协议	IRC List response	poebot	IRC 协议	IRC List response
Phatbot	IRC 协议	IRC List response	Zeus	HTTP 协议	HTTP Get request response
loic	IRC 协议	IRC List response	HFS malware download	HTTP 协议	HTTP Get request response

表 4　靶机开放的僵尸网络控制服务与对应端口

名称	绑定端口号	名称	绑定端口号
波尔远程控制	1 158	凝瑞远程控制	22
黑洞远程控制	1 521	IRCbot	6 667
一笑江湖	3 128	Phatbot	6 667
Syla RAT	8 081	loic	6 667
上兴远程控制	9 080	Agobot	6 667
终结者	1 080	Rbot	6 667
Freerat	21	Zeus	8 080
Krist	23	HFS malware download	80
PaiN RAT	443	狐组远程控制	2 100
FeiMooMa	69	Trojan Nunks	1 433

6.3　扫描速度测试

为了评估系统的扫描效率，指定系统对 3 个 B 类网段的 1 到 1 024 端口和 3 个 A 类网段的常用端口进行扫描。扫描统计信息见表格 5（年份默认为 2013 年）。

实验结果表明，系统能够在可接受的时间内完成对网段的大规模扫描。

7　不足与讨论

本文采用的方法对于通信协议未改变，但二进制代码发生了改变的僵尸网络变种，系统仍然能有效将其检测出，但对于通信协议发生变化的或者使用了新的通信协议的僵尸网络，本方法将无法识别。此外，本文方法无法对类似“黑洞”的僵尸网络控制端进行主动探测：该类僵尸网络的控制端只接收被控端发送的信息，而不给被控端发送反馈信息，因此无法获取足够的信息识别僵尸网络控制端。为了规避本文的探测方法，黑客可通过定期更改僵尸网络的通信交互协议，但让所有的被控主机频繁更改通信交互协议客观上增加了黑客的管理成本，在某种程度上也限制了僵尸网络的发展。另外，本文基于分布式的僵尸网络主动探测方法需要消耗一定的网络带宽，如果任务调度和分配不恰当，容易导致主动探测的 IP 范围太过集中，可能会触发安全设备的误报或导致黑客警觉，对于调度优化分配的方法，将在随后的工作中进行优化。

表 5　　系统扫描速度测试结果统计

任务类型	扫描 IP	端口范围	开始时间	截止时间
B 类网段 1-1024 端口	221.1.*.*	1~1 024	4 月 2 日 19 点 03 分	4 月 2 日 20 点 17 分
B 类网段 1-1024 端口	123.125.*.*	1~1 024	4 月 3 日 9 点 34 分	4 月 3 日 10 点 27 分
B 类网段 1-1024 端口	116.90.*.*	1~1 024	4 月 3 日 15 点 08 分	4 月 3 日 15 点 51 分
A 类网段常用端口	58.*.*.*	80, 8 080, 3 128, 8 081, 9 080, 1 080, 21, 23, 443, 69, 22, 25, 110, 7 001, 9 090, 3 389, 1 521, 1 158, 2 100, 1 433	3 月 18 日 9 点 12 分	3 月 18 日 14 点 17 分
A 类网段常用端口	222.*.*.*		3 月 19 日 12 点 05 分	3 月 19 日 16 点 51 分
A 类网段常用端口	121.*.*.*		3 月 20 日 2 点 09 分	3 月 20 日 6 点 49 分

8　结束语

本文提出基于僵尸网络控制端通信协议指纹的分布式主动探测方法，通过逆向分析僵尸网络的控制端和被控端样本，提取僵尸网络控制端通信协议，并从控制端回复信息中抽取通信协议交互指纹，最后基于通信协议指纹对网络上的主机进行主动探测。

基于该方法，本文设计并实现了 ActiveSpear 主动探测系统，该系统采用分布式架构，扫描所使用的 IP 动态变化支持对多种通信协议的僵尸网络控制端的并行扫描。

在实验环境中对系统的功能性验证证明了方法的有效性，实际环境中对系统扫描效率的评估说明系统能够在可接受的时间内完成对网段的大规模扫描。

参考文献：

[1] CNCERT/CC. CNCERT 互联网安全威胁报告[R]. 2011.
CNCERT/CC. CNCERT Internet Security Threat Report[R]. 2011.

[2] CNCERT/CC. CNCERT 网络安全信息与动态周报[R].2013.
CNCERT/CC. Weekly Report of CNCERT[R].2013.

[3] GU G F, YEGNESWARAN V, PORRAS P, *et al*. Active botnet probing to identify obscure command and control channels[A]. Proc of ACSAC '09[C]. Honolulu, HI, 2009. 241-253.

[4] XU Z Y, CHEN L F, GU G F, *et al*. Utilizing enemies' P2P strength against them[A]. Proc ACM CCS'12[C]. Raleigh, NC, USA, 2012. 581-592.

[5] STRAYER W T, WALSH R, LIVADAS C, *et al*. Detecting botnets with tight command and control[A]. 31st IEEE Conference on Local Computer Networks (LCN'06)[C]. Tampa, FL, 2006. 195-202.

[6] KARASARIDIS A, REXROAD B, HOEFLIN D. Wide-scale botnet detection and characterization[A]. USENIX Hotbots'07[C]. Cambridge, MA, 2007. 7.

[7] GOEBEL J, HOLZ T. Rishi: identify bot contaminated hosts by IRC nickname evaluation[A]. USENIX Workshop on Hot Topics in Understanding Botnets (HotBots'07)[C]. Cambridge, MA, 2007. 8.

[8] GU G, PORRAS P, YEGNESWARAN V, *et al*. Bothunter: detecting malware infection through IDS-driven dialog correlation[A]. 16th USENIX Security Symposium (Security'07)[C]. Boston, MA, 2007. 12.

[9] GU G, ZHANG J, LEE W. BotSniffer: detecting botnet command and control channels in network traffic[A]. Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS'08)[C]. San Diego, CA, USA, 2008. 17.

[10] GU G, PERDISCI R, ZHANG J, *et al*. BotMiner: clustering analysis of network traffic for protocol- and structureindependent botnet detection[A]. Proceedings of the 17th USENIX Security Symposium (Security'08)[C]. San Jose, CA, USA, 2008. 139-154.

[11] YEN T F, REITER M. Traffic aggregation for malware detection. assessment[A]. Proc of the Detection of Intrusions and Malwar, and Vulnerability[C]. Heidelberg, Berlin: Springer-Verlag, Paris, France, 2008. 207 - 227.

[12] GIANVECCHIO S, XIE M, WU Z, *et al*. Measurement and classification of humans and bots in internet chat[A]. Proceedings of the 17th USENIX Security Symposium (Security' 08)[C]. San Jose, CA, USA, 2008. 155-169.

[13] CABALLERO J, YIN H, LIANG Z, *et al*. Polyglot: automatic extraction of protocol message format using dynamic binary analysis[A]. ACM Conference on Computer and Communications Security[C]. Alexandria, VA, 2007. 317-329.

[14] CUI W, KANNAN J, WANG H J. Discoverer: automatic protocol description generation from network traces[A]. USENIX Security Symposium[C]. Boston, MA, 2007. 14.

[15] LIN Z, JIANG X, XU D, *et al*. Automatic protocol format reverse engineering through context-aware monitored execution[A]. Network and Distributed System Security Symposium[C]. San Diego, CA, 2008. 17.

[16] BEDDOE M A. Network protocol analysis using bioinformatics algorithms. http://www.baselineresearch.net/PI/.

[17] CUI W, PEINADO M, CHEN K, *et al*. Tupni: automatic reverse engi-

neering of input formats[A]. ACM Conference on Computer and Communications Security[C]. Alexandria, VA, 2008. 391-402.

[18] WONDRACEK G, COMPARETTI P M, KRUEGEL C, *et al*. Automatic network protocol analysis[A]. Network and Distributed System Security Symposium[C]. San Diego, CA, 2008. 16.

[19] COMPARETTI P M, WONDRACEK G, KRUEGEL C, *et al*. Prospex: protocol specification extraction[A]. IEEE Symposium on Security and Privacy[C]. Oakland, CA, 2009. 110-125.

[20] LEITA C, MERMOUD K, DACIER M. ScriptGen: an automated script generation tool for Honeyd[A]. Annual Computer Security Applications Conference[C]. Tucson, AZ, 2005. 203-214.

[21] CABALLERO J, POOSANKAM P, SONG D, *et al*. Dispatcher: enabling active botnet infiltration using automatic protocol reverse- engineering[A]. Proc of CCS'09[C]. Chicago, IL, USA, 2009. 621-634.

[22] DASWANI N, STOPPELMAN M. The anatomy of clickbot[A]. Proc of the 1st Conf on First Workshop on Hot Topics in Understanding Botnets[C]. Boston, MA, USA, 2007. 11.

[23] MILLER B, PEARCE P, GRIER C, *et al*. What's clicking what? Techniques and innovations of today's clickbots[A]. Proc of the Detection of Intrusions and Malware, and Vulnerability Assessment[C]. Amsterdam, Netherlands, 2011. 11.

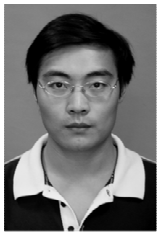
[24] DITTRICH D, DIETRICH S. Discovery Techniques for P2P Botnets, Technical Report[R]. 2008.

[25] KARTALTEPE E, MORALES J, XU S H, *et al*. Social network-based botnet command-and-control: emerging threats and countermeasures[A]. Proc of the Applied Cryptography and Network Security[C]. Beijing, China, 2010. 511-528.

[26] HOLZ T, GORECKI C, RIECK K, *et al*. Measuring and detecting fast-flux service networks[A]. Proc of the 15th Annual Network and Distributed System Security Symp[C]. San Diego, CA, 2008.19.

[27] 江健, 诸葛建伟, 段海新等. 僵尸网络机理与防御技术[J], 软件学报, 2012,23(1):82-96.
JIANG J, ZHUGE J W, DUAN H X, *et al*. Research on botnet mechanisms and defenses[J]. Journal of Software, 2012,23(1): 82-96.

作者简介：



司成祥（1982-），男，山东郯城人，博士，国家计算机网络应急技术处理协调中心工程师，主要研究领域为网络安全。

孙波[通信作者]（1972-），男，吉林桦甸人，博士，国家计算机网络应急技术处理协调中心教授级高级工程师，主要研究方向为网络安全。E-mail: sunbo1390123@139.com。

杨文瀚（1989-），男，湖北公安人，北京大学博士生，主要研究方向为恶意代码检测与防范。

张慧琳（1987-），女，河南淮阳人，北京大学博士生，主要研究方向为恶意代码检测与防范、互联网安全监测。

薛晓楠（1989-），男，吉林松原人，北京大学硕士生，主要研究方向为恶意代码检测与防范。