# 分布式端口扫描检测系统的设计与实现

梁锦华 （中国科技大学研究生院 100039）
蒋建春 文伟平 卿斯汉 （中国科学院信息安全技术工程研究中心 100080）

摘要：本文提出了一种基于MPI分布式的分析主机应答包而进行端口扫描检测的方法。扫描者在扫描网络时需要得到目标的应答，根据应答的情况才能对目标的端口开放情况等进行分析，这种方法不分析进入网络中数据包，减轻了数据分析强度，巧妙地解决了一些对扫描探测包进行检测时难以解决的问题（如对分片扫描包检测）。检测规则使用了基于时间窗、端口策略等方法，较好地解决了对分布式扫描、慢速扫描检测的难题。

关键词：端口扫描 端口扫描检测 MPI (Message-Passing Interface)

## 1 当前端口扫描检测分析

### 1.1 NSM (The Network Security Monitor)

网络安全监测器 (NSM)[3]，第一个网络入侵检测系统 (NIDS)，也是第一个检测扫描的网络入侵检测系统。如发现任意一个IP地址在一个时间段内与另外超过15个的IP地址进行连接，则认为是扫描行为。这个规则至今仍被许多系统使用。

### 1.2 GrIDS (The Graph Based Intrusion Detectiong System)

基于图形的入侵检测系统 (GrIDS)[4]，通过建立节点之间的连接图表来代表网络中的主机的连接，这样，来自同一个地址的扫描就被发现了。可以用很大的比例来进行图表查看，能发现随机性较强的扫描，但不能处理不正常和随机的数据包，所以不能检测秘密扫描，另外，其原型实现使用了Perl，相对较慢，不能适应现在的大型快速网络。

### 1.3 Snort 预处理程序

Snort端口扫描预处理程序用于向标准记录设备中记录从一个源IP地址来的端口扫描的开始和结束。端口扫描定义为在时间T (秒) 之内向超过P个端口进行TCP连接尝试，或者在时间T (秒) 之内向超过P个端口发送UDP数据包。缺点是不能侦测分布式扫描、慢速扫描，不能处理分片。

### 1.4 Emerald

Emerald系统使用异常检测技术，使用行为规则匹配和流量监控来发现扫描。缺点是不能侦测慢速扫描、分布式扫描和新的IP地址。

## 2 分布式环境MPI简介

消息传递机制是广泛应用于并行机的一种模式，其基本概念是通过消息完成进程通信。消息传递接口MPI (Message Passing Interface) 是为编写消息传递程序而开发的一个广泛使用的标准。

消息传递标准化涉及到大约60个国家的专家学者，他们主要来自于美国和欧洲的40个组织，这包括并行计算机的多数主要生产商，还有来自大学、政府实验室和工厂的研究者们。标准化开始于分布存储环境中消息传递标准的讨论会，这个会议是由并行计算研究中心支助的，于一九九二年四月二十九日至三十日在威吉尼亚，威廉姆斯堡召开。会议上讨论了标准消息传递的必要的、基本的特点，并建立了工作组继续进行标准化工作。

建立消息传递标准的主要优点是可移植性和易于使用，以低级消息传递程序为基础的较高级和 (或) 抽象程序所构成的分布存储通信环境中，标准化的效益特别明显。而且，消息传递标准的定义能提供给生产商清晰定义的程序库，以便他们能有效地实现这些库或在某些情况下为库程序提供硬件支持，因此加强了可扩展性。简单地说，消息传递接口的目的是为编写消息传递程序而开发的广泛使用的标准。象这个接口一样，应为消息传递建立一个实际的、可移植的、有效的和灵活的标准。它的目标如下：

(1) 设计一个应用编程接口 (不必为编译器或一个系统实现库)。

(2) 允许有效的通信：避免存储器到存储器的拷贝，而允许计算和通信的重叠，尽可能给通信协同处理器减少负担。

(3) 对于接口，允许C语言和Fortran 77进行方便的联接。

(4) 设定一个可靠的通信接口：用户不必处理通信失败，这些失败由基本的通信子系统处理。

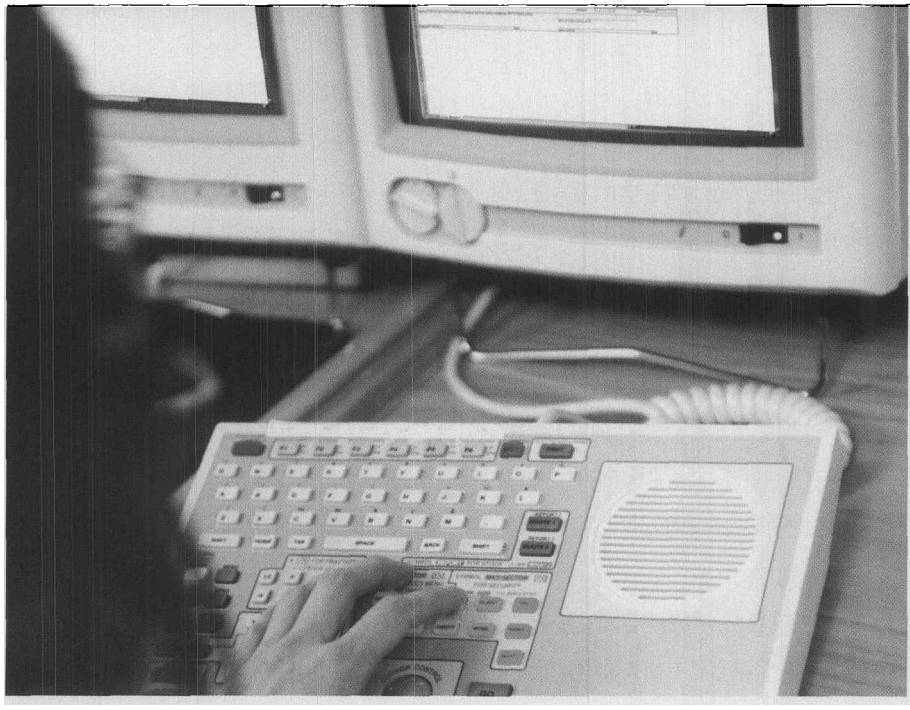
(5) 定义一个接口，并非不同于现在的实践，如：PVM,NX,Express,p4等，而是提供更大灵活性的扩展。

(6) 定义一个接口，它能在基本的通信和系统软件无重大改变时，在许多生产商的平台上实现。接口的语义是独立于语言的。

(7) 接口应设计成允许线程－安全 (thread-safety)。

## 3 MPI构建的分布式环境

在Linux环境下安装MPI，首先解压源程序包，生成目录／mpich。在Linux环境下ch-p4设备被设为缺省，通常ch-p4设备在网络的启动进程是rsh (remote shell)。但是rsh的使用要求在相应的主机上设置权限，而这种权限



的设置与某些网络的安全要求冲突。因此，考虑到主机之间通信的安全，将用 ssh (secure shell) 取代 rsh。

### 3.1 ssh 的设置

当 ssh 安装后，用 ssh-keygen 生成自己的加密密钥及对外公开使用的公钥，将生成一对公钥和私钥，公钥保存在~/.ssh2/id-dsa-1024-a.pub里，私钥保存在~/.ssh2/id-dsa-1024-a里。然后将要远程访问的主机的公钥加入本机 ~/.ssh2/authorization 中，即加入一行 Key remotecomputer.pub。为了避免每次登录要输入密码，可以创建一个 ssh-agent 并加入密码。

### 3.2 MPI 的设置

配置好 ssh 后，进入到/mpich 目录下，运行./configure-rsh=ssh-prefix=/usr/local/mpich,这样ssh将取代 rsh 并且指定了安装目录。当编译安装完成后，到/usr/local/mpich/share 下，将分布式环境中各个主机名写入到文件 machine.LINUX 中。

通过以上的设置，MPI 环境初步建立，保证了分布式环境中各主机可以安全的通信。

## 4 分布式的端口扫描检测系统

### 4.1 分布式端口扫描检测系统结构

图1显示了扫描检测代理的分布式配置和管理结构，检测代理处于网络连接节点处，便于监视网络流量和信息交换。用基于主机的扫描检测系统保护网络中的关键系统，基于网络业务分析的检测代理保护网络中的指定网段中的系统等。管理控制器则负责所辖网络范围内的扫描检测代理的配置、管理、报警信息的汇总、融合分析以及激活响应机制等，对网络行为进行全局性的监控、分析、报告或处理。

### 4.2 消息通信机制

MPI 消息传递接口为分布式扫描的各扫描进程间的通信提供了一个标准，进程由一个唯一的"标识数"(整数) 表示，标识数为数0、1、2、……、N-1。0进程发送消息，利用MPI-Send()函数通知各进程需扫描的端口及扫描间隔，并接收 MPI-Recv()各子进程返回的扫描结果。其通信流程如图 2 所示。

(1) MPI 初始化各进程，进程标识数为 0 则为主进程，对其他从属进程进行控制。

(2) 首先发送消息到各从属进程 MPI-Send (buffer,count,datatype,destination,tag,MPI-COMM-WORLD)。

(3) 各从属进程接收从主进程发来的消息 MPI-Recv(buffer,maxcount,datatype,source,tag,MPI-COMM-WORLD,&status)，取得所需的各项参数。

(4) 各从属进程根据接收的参数执行各自的扫描进程。

(5) 各从属进程将扫描结果发送给主进程。

(6) 主进程确定所有进程执行完毕后，发送结束标志。

(7) 各从属进程退出，主进程退出。

### 4.3 分布式的端口扫描检测原理

一般来说检测基于以下两类信息：

(1) 统计信息：连接访问间隔、地址和端口信息等都是基本的被统计的信息。最简单的例子，扫描探测行为是阶段性的而被扫描的端口是连续的。

(2) 应答包特征：扫描时会使用一些特殊类型的包进行探测，如 TCP FIN、TCP SYN、UDP

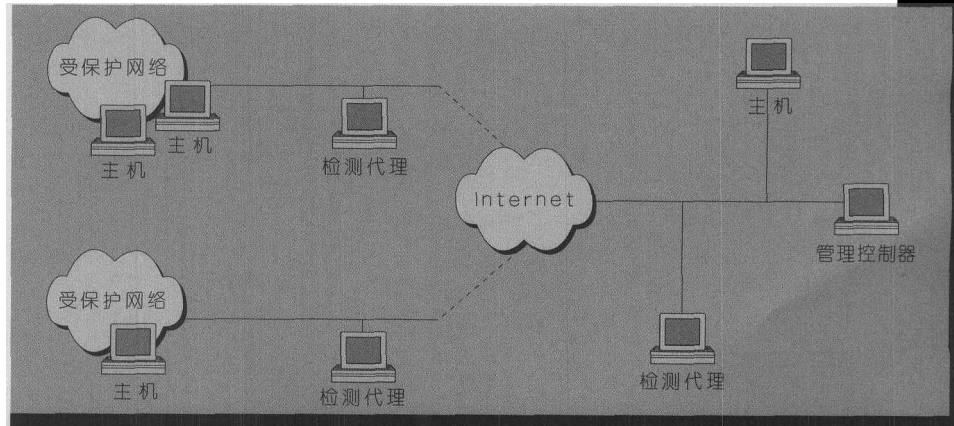



图 1 扫描检测代理的分布式配置和管理结构图

响应、各种标志位设置的包等，对这些包的应答特征也有所不同。

当扫描者发出探测包进行探测时，被扫描网络中的设备如路由器、防火墙、终端主机等会自动发出应答包进行应答，如果扫描者穷尽一个网段进行扫描，必然会导致大量包含错误信息的包的应答。我们这里使用对系统应答包的监视来检测扫描行为。当系统发出大量的应答包尤其是大量包含错误信息如ICMP目标不可达等信息的应答包时，系统可能正在被扫描，分布式的检测代理应记录这些应答以便分析。如图3所示应答包的收集示例。

这部分功能由两个子模块组成，一个包窥探器，负责从以太网上选择性地抓取特定类型的IP包，另一块负责分析IP包，维护统计数据，并据此判断是否有扫描行为出现。

### 4.4 网络数据的获取

将网络接口设置为混杂模式，将网络上向外传输的数据包截取下来，供分析使用。由于效率的需要，有时要根据设置过滤网络上的一些数据包，如特定IP，特定MAC地址、特定协议的数据包。数据包获取模块的过滤功能的效率是该网络监听的关键，因为对于网络上的每一数据包都会使用该模块过滤，判断是否符合过滤条件。低效率的过滤程序会导致数据包丢失、分析部分来不及处理等。

为提高效率，数据包过滤在系统内核里来实现。我们采用了专门为数据监听应用程序设计的开发包里Wincap来实现这模块，开发包中内置的内核层实现的BDF过滤机制和许多接口函数不但能够提高监听部分的效率，也降低了我们开发的难度。同时wincap是从UNIX平台上的LIPCAP移植过来的，它们具有相同的接口，减轻了不同平台上开发网络代理的难度。

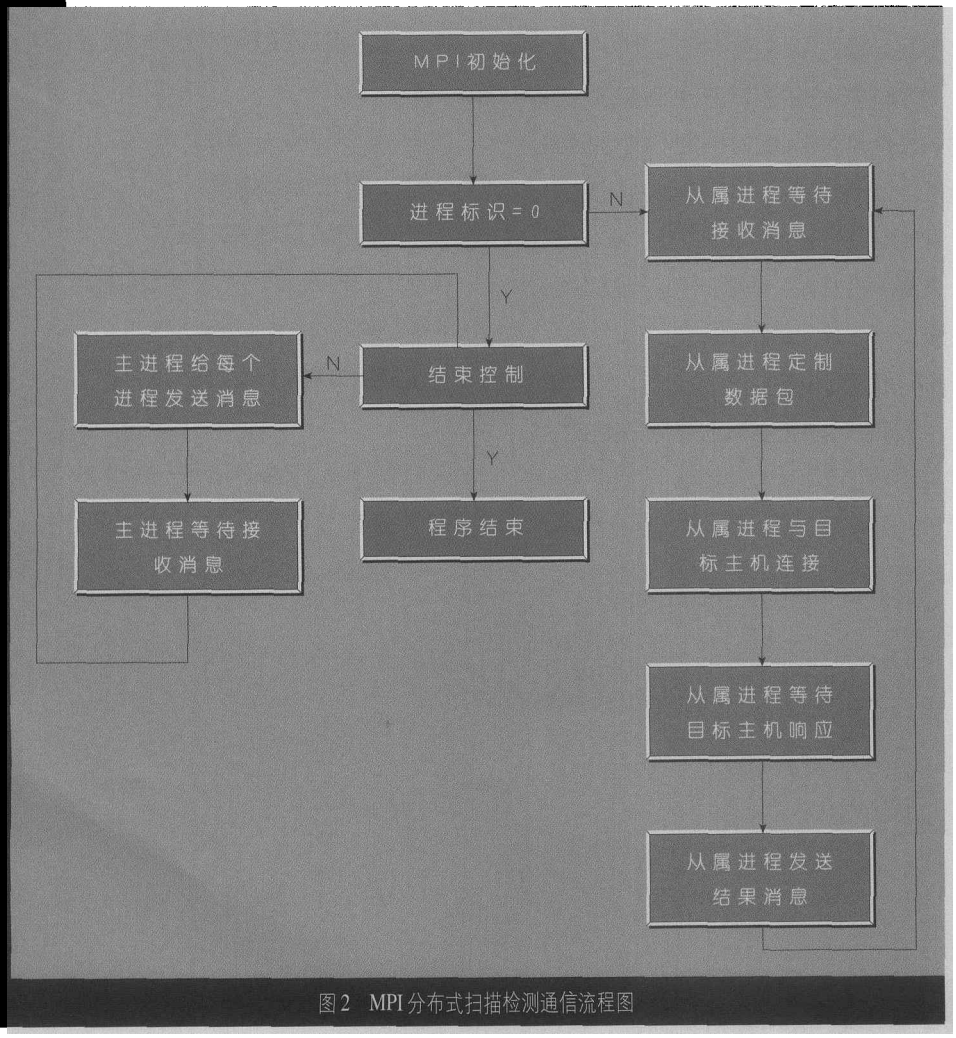



图2 MPI分布式扫描检测通信流程图

### 4.5 数据分析

#### 4.5.1 基于时间窗的检测

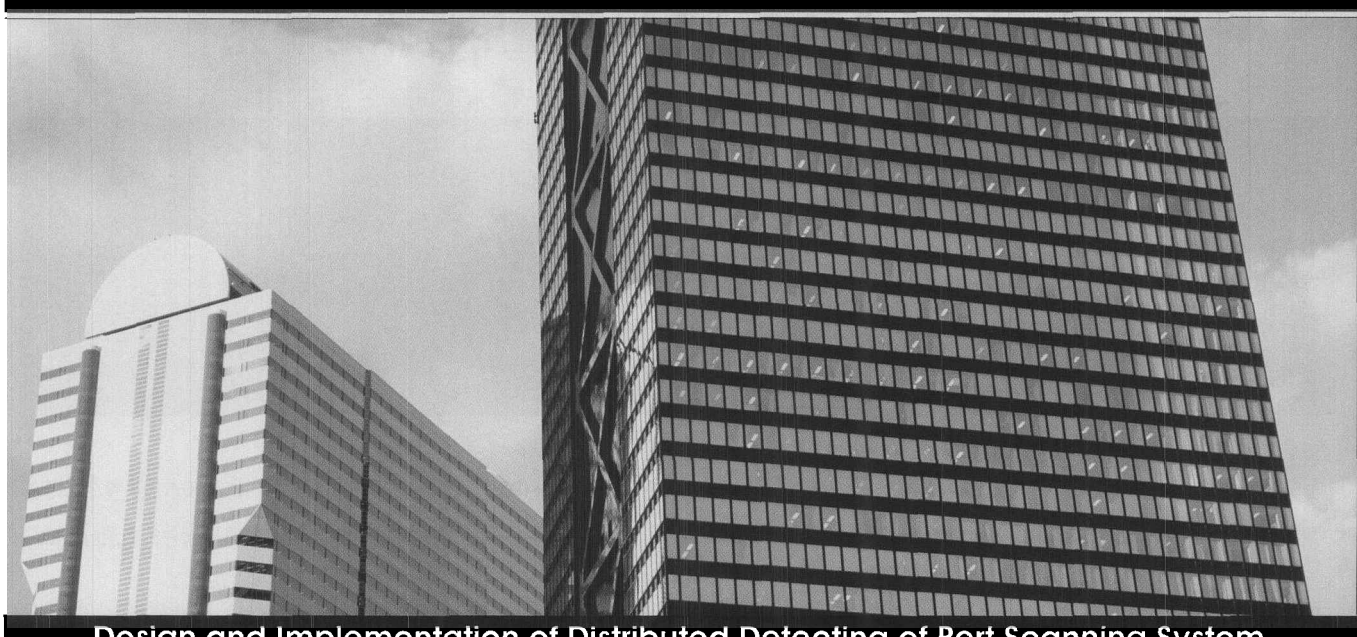
对TCP和UDP分别维护一张固定大小的表，表中每一项记录包括一个端口号以及从该端口发出最后一个TCP RST包（或ICMP Port Unreachable包）的时间，表的大小（记录数）就是认为端口扫描发生了的那个阈值；每当从网上抓取到一个这样的包，如果端口号是新的，就在对应表中加入一个条目，如同端口号与已经存在的某表目中的端口号相同，则不增加条目，只更新该条目中的时间；一个定时器激发检查条目中的时间，超过一定期限就把条目从表中除去，这样，该表的实际大小就反映了最近一段时间内，可能遭到了扫描的端口数目，如果一个新的端口号出现，在试图加入表中时发现因为再没有空闲的表目而无法加入时，则说明，根据我们的模型，可以认为该主机确实遭到了TCP（或UDP）端口扫描。

#### 4.5.2 基于策略的检测

为保证较高的检测效率，检测系统不对所有的连接端口进行检查，只对安全风险端口进行检查。因为根据Honeynet统计分析，端口扫描往往使用聚焦于单个脆弱性的策略，针对这个脆弱性扫描到尽可能多的系统。对此脆弱性端口我们也作重点防御。

将端口区分为正常服务端口和安全风险端口，即服务器确实开放服务的端口号记录在正常服务端口表中，其他易发生端口扫描的端口号（如本机未开放的常用服务、木马端口等）列在安全风险端口表中，定期对这两个表进行维护。

如果连接的端口是正常服务端口则用较高



Design and Implementation of Distributed Detecting of Port Scanning System

阀值时间窗的检测规则进行检测，如果连接的端口是风险端口则用一个较低阀值的时间窗规则进行检测。

4.5.3 检测慢速扫描

在一个较长的时间窗内，对发起连接的IP进行记录，如发现其连接的风险端口数目达到阀值（如20分钟内4次），我们就认为有扫描行为的发生。

4.5.4 处理扫描淹没

欺骗的"端口扫描"能被用来填满日志，真正的攻击就是后继的真正的端口扫描，这很可能是成功的扫描。

使用针对攻击类型的概率限制可以解决这个问题。(比如——每0秒的信息不能超过5个)。概率限制发生时，把这个事件记录下来，同时暂时地停止记录这种类型的攻击。

解决这个问题的另外一个方法是给每一种攻击一个各自的日志限额。

## 5 实验分析

发送TCP SYN探测包对HTTP服务（80端口），UDP探测包对DNS服务（53端口）进行探测，以及用分布式扫描方式同时对多个网络进行扫描的结果分析，能够准确检测出扫描行为。

## 6 结束语

分布式端口扫描检测系统对主机的应答包进行检测分析，巧妙地解决了一些检测扫描包难以解决的问题（如分片扫描包检测）。检测规则使用了基于时间窗、端口策略等方法较好地解决了对分布式端口扫描、慢速扫描检测的难题。存在的问题是容易产生漏警，对使用特殊标志位的探测包的检测效果不明显，易受具体网络环境如防火墙、IDS的设置的影响。进一步的工作可以将对应答包的检测和对扫描者发出探测包的分析检测结合起来，以实现高效、灵活的检测。


参考文献

1 Honeynet http://project.honeynet.org/

2 Honeynet Know Your Enemy: Statistics http://project.honeynet.org/papers/stats/ 23 July, 2001。

3 Heberlein, L.T.,G.Dias,K. Levitt,B.Mukherjee,J. Wood, and D.Wolber,network security moniter,Proc., 1990 Symposium on Research in Security and Privacy,pp.296-304,Oakland,CA,May1990。

4 Stantiford-Chen S.,S.Cheung,R.Crawford,M. Dilger,J.Frank,J.Hoagland,K.Levitt,C.Wee,R.Yip,D. Zerkle,rIDS-A Graph-Based Intrusion Detection System for Large Networks The 19th National Informaion Systems Security Conference。

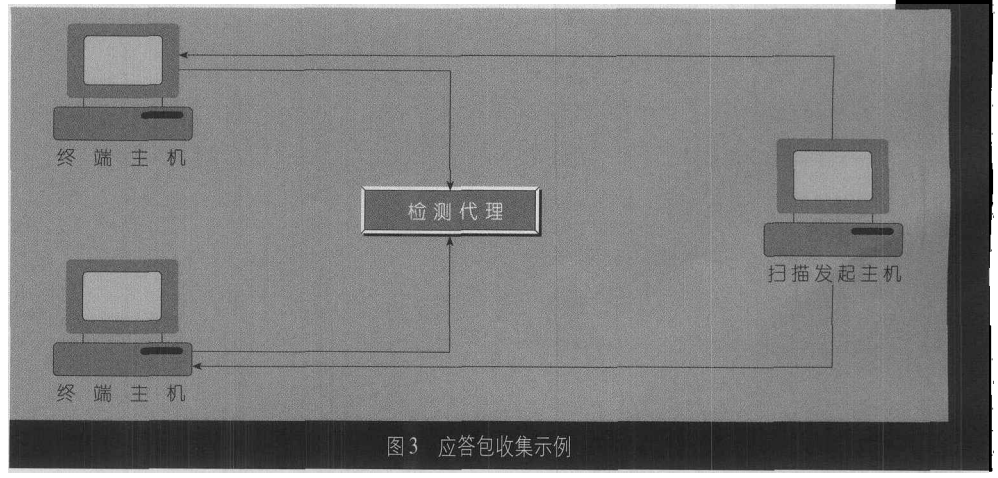




图3 应答包收集示例