

**毕业设计（论文）**

题 目 企业服务器文件完整性检测

系统的设计与实现

专 业 网络空间安全

学　　 号 160700226

学 生 陈锦前

指 导 教 师 董开坤

答 辩 日 期 2020.06.09

**摘 要**

计算机网络技术的不断发展，使得人们的工作和生活发生了巨大的变革。网络已经渗透到生活的各个领域，随之而来的是个人隐私的泄露、重要文件的篡改等安全问题。文件完整性检测系统作为入侵检测运行系统必不可少的一个子模块，可以保证系统运行安全。

本论文研究了基于MD5算法的文件检测技术，设计并实现了针对企业服务器的文件完整性检测系统，详细阐述了系统构建过程，并通过实验的方式对系统进行评估。

该系统的文件数据均来自于Windows环境下的文件。该系统首先使用操作系统的接口获取文件的属性值以及文件内容，再使用MD5算法生成文件内容的摘要值，最终将文件的属性值及文件内容摘要值存储到数据库从而得到文件的初始消息。该系统分为检测端与服务器端，使用基于TCP协议的通信技术，对企业服务器文件完整性进行检测，并将检测结果可视化输出。

通过对检测结果的分析，该系统可以实现对多台企业服务器中的不同文件进行检测，并将检测出被篡改后的文件进行记录及展示。系统在文件完整性检测上整体较为有效。

**关键词**：入侵检测；文件完整性检测；摘要值计算；MD5算法；

**（内容及关键词用小4号字）**

**Abstract**

With the continuous development of computer network technology, great changes have taken place in people's work and life. The network has penetrated into all areas of life, followed by personal privacy disclosure, important documents tampering and other security issues. As an essential sub module of IDS, file integrity detection system can ensure the security of the system.

This paper studies the file detection technology based on MD5 algorithm, designs and implements the file integrity detection system for enterprise server, describes the system construction process in detail, and evaluates the system through experiments.

The file data of this system are all from files under Windows environment. The system first uses the interface of the operating system to obtain the attribute value and the content of the file, then uses the MD5 algorithm to generate the summary value of the file content, finally stores the attribute value and the summary value of the file content to the database to get the initial message of the file.The system is divided into detection end and server end.The system is divided into detection end and server end. It uses the communication technology based on TCP protocol to detect the integrity of enterprise server files, and visualizes the detection results.

Through the analysis of the detection results, the system can detect different files in multiple enterprise servers, and record and display the tampered files. The system is more effective in document integrity detection.

***空一行***

**Keywords:** Intrusion detection，Integrity detection，summary value calculation，MD5 algorithm

**（内容及关键词用Times New Roman**

**注意事项***中各章题序及标题用黑体小4号字，其余用宋体小4号字，数字及符号用新罗马字体*

# 第1章 绪 论

## 1.1 课题背景及研究的目的和意义

### 1.1.1 课题背景

随着现代网络技术的快速发展和网络规模的不断扩大，计算机网络已渗透到了社会的各个领域，人们在享受网络带来的资源共享及信息交流方便快捷的同时，也不得不面对越来越多的来自网上的恶意攻击和各种黑客入侵等，也存在个人隐私泄露、重要文件篡改等安全问题，而且日新月异，人们对于网络安全的需求与日俱增[1]。

互联网与生俱有的开放性、交互性和分散性特征，使人类所憧憬的信息共享、开放、灵活和快速等需求得到满足。网络环境为信息共享、信息交流、信息服务创造了理想空间，为人类社会的进步提供了巨大推动力。然而也正是由于互联网的上述特性，产生了许多安全问题[2]。尤其当今网上支付、数字货币技术不断成熟，保护企业的文件至关重要。

攻击者越来越多,攻击工具和手法日趋复杂多样,单防火墙技术已经无法完全阻止网络入侵这一日益严重的网络安全问题。所以需要在防火墙后提供新的安全保障，网络的防卫必须采用一种纵深的、多样的手段[3]。入侵检侧系统(IDS)是一种不同于防火瑞的主动保护网络资派的网络安全系统,是防范网络攻击的最后一道防城,是作为防火堵有益补充而出现的网络第二道防线产品,在网络安全技术中起着不可替代的作用[4]。

### 1.1.2 研究目的和意义

按照美国国防部1983年制定的可信计算机系统评测标准，信息安全应当确保信息的机密性(Confidentiality)、完整性(Integrity)和可用性(Availability)这三个基本标准要求。其中,完整性指防止信息遭受以非授权方式所作的篡改或破坏,它作为信息安全三个基本标准要求之一,占有举足轻重的作用[5]。

系统的安全是一个体系,建立在各种安全机制集成的基础上,入侵检测系统(IntrusionDeteetionSystem,简称IDS)作为系统安全的一道重要防线,主要是通过多种手段监控系统来发现入侵。文件完整性检查是IDS中的关键部分之一,它通过检查系统关键资源的变化情况来为入侵的发现提供依据。文件完整性检查系统的目的在于:保证被保护主机的关键资源不被恶意更改。所谓关键资源是指系统中不应也不能被经常更改的关键成分,如:内核、配置文件、可执行程序、库等,以及那些由第三方提供的可能影响到主机运行的资源[6]。

文件完整性检查系统的优点：①从数学上分析，攻克文件完整性检查系统，无论是时间上还是空间上都是不可能的。文件完整性检查系统是非常强劲的检测文件被修改的工具。②文件完整性检查系统具有相当的灵活性，可以配置成为监测系统中所有文件或某些重要文件。③当一个入侵者攻击系统时，他会干两件事：首先，他要掩盖他的踪迹，即他要通过更改系统中的可执行文件、库文件或日志文件来隐藏他的活动；其次，他要作一些改动保证下次能够继续入侵。这两种活动都能够被文件完整性检查系统检测出[7]。

通过研究企业服务器文件完整性检查系统的设计与实现，可以有效地检测攻击，可以更有效地保护系统关键文件不被篡改，防止木马、病毒等对系统的破坏，保障企业正常运行[8]。……

## 1.2 国内外在该方向的研究现状及分析

### 1.2.1 国外的研究现状

2014年Mark Kedgley通过分析当前黑客的攻击行为以及网络环境，指出了当前杀毒软件的不足之处，例如无法有效抵御基于0-day漏洞的攻击行为，同时阐明了文件完整性检查系统的优点及可靠性[9]。

2014年，日本学者Watanabe,H根据在日本使用广泛的Gfarm分布式文件管理系统，在已有的基于时间戳的文件完整性检查系统基础上，设计了一套改进系统，大幅度提升了该系统的效率[10]。

云存储系统中数据的所有权和管理权分离,导致用户数据隐私泄露、完整性破坏等一系列安全问题。2018年Pinheiro Alexandre等人设计了一套基于实时监控云端数据及服务器行为的系统用于保护用户隐私[11]。

### 1.2.2 国外的研究现状

近年来中国学者在文件完整性检查研究方向取得了不断进步，尤其在云计算以及云存储方向，根据云计算环境下平台系统的特点,刘海永提出了一套新型的文件完整性检测系统的设计方案，采用了新型的SHA-3安全算法生成文件摘要，将安全系统与被检测主机相互分离，将安全检测负载与被检测主机适当分离。实现了检测系统的运行的大部分负载从被监控主机转移,从而适用于云计算环境[12]。

随着云端存储技术的不断成熟，目前已有的数据完整性证明机制还存在不支持全动态操作,通信和计算开销大等各种问题。因此贾倍提出了支持数据隐私的数据完整性证明机制、一个带秩的多分支路径结构(MBT,Multi-branch Tree)和数据完整性检测算法、以及基于MBT的数据完整性证明机制，该方案不仅保护了数据对第三方验证者的隐私性,而且支持全动态操作与批量审计操作,简化了数据完整性证明信息的获取过程[13]。

刘广沛利用移动Agent技术在云端部署分布式虚拟机代理模型,同时通过虚拟机代理模型构建基于区块链的完整性保护框架,使用默克尔哈希树生成文件对应的唯一散列哈希值,借助区块链上的智能合约监控数据变化,及时向数据拥有者发出数据篡改的警告信息，另外,利用“挑战-应答”模式构造基于区块链的云数据完整性验证方案[14]。

移动可信平台需要经常进行大量完整性校验,然而其各种资源却十分有限。张永棠教授提出了一种可以配合各种成熟的HASH算法使用的完整性校验方法——RMAC(Random-MAC)。从不同版本的Linux系统中,搜集了不同格式的ELF文件,随机抽取粒度样本,进行完整性校验分析。由于其引入随机性,每次计算产生的摘要都不同,使目前已有的病毒都无法做到每次都能通过RMAC校验。RMAC提供的障碍有效地降低了病毒的繁殖速度,可以阻止病毒的大规模爆发[15]。

## 1.3 论文主要内容与结构安排

本次课题的主要研究内容是设计一个企业服务器文件完整性检查系统，对企业服务器中的文件信息进行提取、分析、检查，以实现在复杂的网络环境下，根据文件属性或者文件摘要进行文件完整性检查，判断文件是否受到恶意篡改。具体研究内容如下：

（1）对现有的文件完整性检测技术进行调研并完成综述

（2）设计针对企业服务器的文件完整性检测系统，使用改进后的MD5算法进行文件初始信息获取，并基于MySQL数据库进行信息存储。

（3）基于TCP协议，使用多进程/多线程编程技术，实现通信系统完成检测请求发送于检测结果接收。

（4）利用Python编程实现所设计的企业服务器文件完整性检测技术。

（5）使用该系统对多台服务器的不同文件进行完整性检测，通过检测结果得出结论。

本文结构及各章节所述信息具体如下：

第1章是绪论，本章主要介绍了本课题研究的背景、课题研究需要达到的目的，以及国内外从大数据、云存储、可信计算等不同方向对文件完整性检测系统的研究现状、详细说明了本文要研究的具体内容。

第2章是相关理论概述以及系统需求分析，本章对企业服务器文件完整性检测技术的相关的理论进行介绍，主要由基于TCP协议的多进程/多线程通信技术和文件完整性检测技术这两方面构成，其中文件完整性检测技术作为本章的重点内容，详细介绍了基于MD5算法的文件完整性检测技术以及MD5算法的优化技术。对系统进行了功能性需求分析以及可行化分析。

第3章是企业服务器文件完整性检测系统设计，本章对此系统进行了总体设计和功能化模块概要设计。本章共设计了三个功能模块，并对各个模块的具体功能进行了阐述。

第4章是企业服务器文件完整性检测系统的系统实现，本章描述上一章所设计内容的实现过程，包括文件初始信息获取与存储模块、文件完整性检测模块、检测结果可视化模块。

第5章是系统测试，本章设计了对企业服务器文件完整性检测系统的功能测试与性能测试，根据测试结果进行分析得出结论。

结论总结了本次课题的全部研究工作、各项工作的完成状态，对接下来的研究工作进行了展望。

# 第2章 系统需求分析

本章首先对论文研究的课题进行概述，主要对课题中的通信技术与文件完整性检测技术进行解释说明。根据系统概述中所描述的内容对系统的功能性需求和可行性进行分析，最后总结本章内容。

## 2.1 系统概述

本论文研究了文件完整性检测技术和基于TCP协议的通信技术，根据研究内容设计和实现了企业服务器文件完整性检测系统。那么首先对该系统所使用的相关技术进行概述，之后再分析得出系统需求。

## 2.1 文件完整性检测技术

完整性指网络信息的真实可信性，即网络上的信息不会被偶然或者蓄意地进行删除、修改、伪造、插入等破坏，保证授权用户得到的信息是真实的。只有具有修改权限的实体才能修改信息，如果信息被未经授权的实体修改了或者在传输过程中出现了错误，信息的使用者应该能够通过一定的方式判断出信息是否真实可靠[17]。目前主要使用密码学中的各种算法对文件完整性进行检测。

### 2.1.1 基于MD5算法的文件完整性检测技术

美国密码学学家罗纳德（Ronald Linn Rivest）在MD5算法的基础上，于1992年设计出更为趋于成熟的MD5算法。使用该算法可以将任意长度的文本转换为一个固定长度的字符串。由于同一文件产生出相同字符串的概率极小，可将该字符串视为该文件的摘要信息。基于MD5算法的文件的完整性检测是比较文件前后的摘要值是否相同，如果被篡改，摘要值将发生变化，因此可以有效检测出文件是否完整，其检测模型图见图2-1所示。

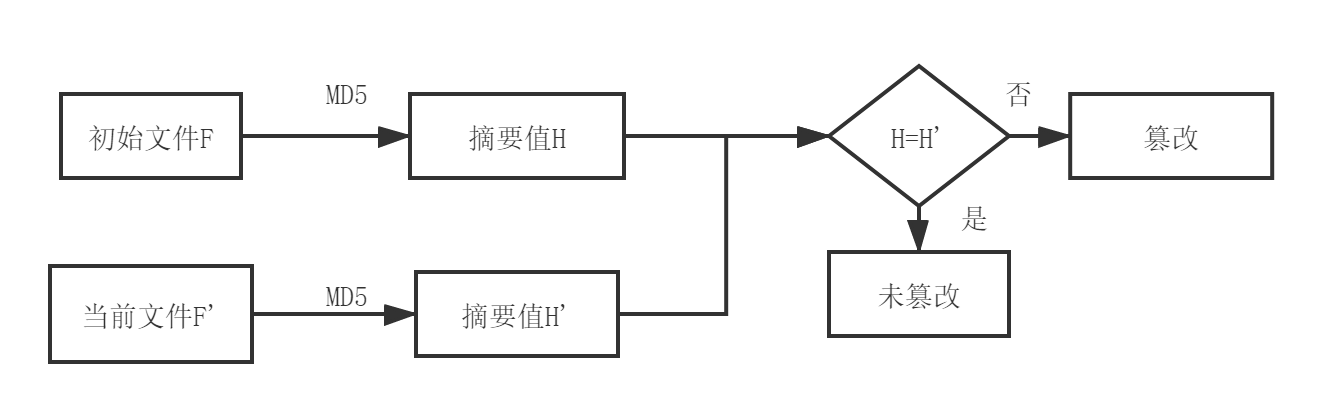


图2-1 基于MD5算法的文件完整性检测模型图

自2004年8月，王小云在美国加州圣芭芭拉召开的国际密码大会上首次宣布了她及她的研究小组破译了MD5算法后用MD5做数字签证、电子签名已经不安全了，但是在普通的应用开发中MD5还是被开发者广为应用，虽然随着海量存储技术和分布式计算的发展，在如今的大数据时代，可以通过创建名文库组建反向查询库对MD5值进行反向查询，只要拥有足够大的反向查询库，就有可能直接查询出器明文值。但是通过用户和系统对MD5算法作部分修改或辅助改进则能有效加大反向查询的难度，从而避免攻击者从数据库中的文件摘要值直接获取文件内容导致文件泄露。例如以下几种方案：

（1）多次加密

进行一次MD5加密后，可对生成的MD5密码的全部或部分再进行一次或多次加密，增加在字典中查找的难度。如md5(md5(“cjqhit”))是对MD5码的全部进行再一次加密；md5（right（md5（“cjqhit”），10））是对MD5码的右10位进行再加密。经实验证明多次MD5加密是不会影响其时间性能的。

（2）加盐法

若想解决短明文摘要加密安全性不高的问题，可以增加原始明文长度, 该方法又称为加盐法。这个“盐”常为字符串数据，将盐与密码一起合并计算摘要，可增加查询破解的难度，从而提升安全性。破解网站的暴力查询破解只能查询常见的短明文的摘要，如果加长原文数据长度，摘要结果则很难查到[22]。

（3）对MD5算法的步函数和消息分量计算方法进行改进

由于HASH函数本身的限制，使用HASH函数加密会出现碰撞的现象，其抗碰撞性不足的重要原因之一是高位不能充分混淆，为了使得MD5抗碰撞的能力得以提升，我们可以对明文进行消息预处理，根据多次实验探索，发现对Mi的复杂化是十分有效的改进方式，改进方法为把原先只有一个参数的32位Mi改成有三个参数取三个提取出来的32位明文消息段相乘取后32位的合成参数[23]。

（4）散列值的长度增加

虽然128位散列值空间搜索的复杂度已经很大了，但是随着各类算法的不断优化，寻找碰撞变得越来越简单。增加散列值的长度虽然会对运算速度造成影响，但对于增加安全性考虑，这些改进是很有意义的[24]。

该系统存在大量、同时的使用MD5算法对文件内容进行摘要值的计算，MD5算法过程繁琐、算法复杂，在具体实现时必须考虑其计算性能，如果加密时间过长将会为该系统带来非常差的用户体验。所以对于MD5算法的计算性能的优化分析具有非常大的现实意义[25]。这里介绍4种优化MD5算法计算性能的方法。

（1）展开MD5算法的循环过程

由于MD5算法需要很多循环, 而且很多时候采用多层循环嵌套来实现。对于计算机体系来说, 多次的循环与多层次的循环嵌套，加上变量的地址寻址，自身在CPU执行时，会浪费大量的时间，加上多次循环和多层循环嵌套，与变量地址寻址使得CPU指令流水线的预取与阻断的机制失效，从而增加了大量的计算时间。对于大数量级和多层循环嵌套的MD5算法来说，可以通过展开循环过程来提升其计算时间。所以在编程实现MD5算法时，尽量将每个循环体采用5到10次的循环次数，通过多个循环程序块来完成整个MD5算法。同时，为了减少内存寻址的时间浪费，能够采用常量的变量，尽量在预定义时采用常量的形式定义。

（2）避免指令跳转

在当前CPU的计算机制中，执行固定内存块的指令速度是最块的，如果发生程序地址指针改变，则CPU将会浪费一部分时间去内存寻址或从虚拟交换空间或磁盘中读取数据，而这将会浪费大量的计算时间。对于常见的编程语言, if…else或者for、while循环，甚至是goto语句，都是常见的指令跳转语句，在实现MD5算法时，尽量避免使用产生指令跳转的语句，从而减少CPU执行代码时的寻址时间。

（3）变量长度CPU寄存器匹配

不同计算机的CPU寄存器大小是不一样的。对于长度大于CPU寄存器长度的变量，CPU将会分多次进行计算，最后将结果进行整合来完成计算。如果变量长度小于CPU寄存器长度，那么CPU将会在寄存器后附加其他的数据或者执行来完成计算。对于一个32位寄存器的CPU来说，对一个32位变量的计算时间将比对一个16位变量的计算时间块近一倍，所以在复杂繁琐的MD5算法实现过程中, 定义变量或常量时，尽量与当前执行算法的CPU寄存器长度一致，从而增加整个代码中数值的计算速度。

（4）减少变量个数

由于操作系统对于变量的存放机制，使得CPU在对某个变量进行计算时，需要按照变量指针从内存中寻址读取，存放到寄存器中进行计算。整个计算机框架CPU从内存读取和写入的时间对于CPU计算来说慢很多，所以在MD5算法实现时, 尽量减少变量的个数。此外，常量与变量的存放机制也有所不同，所以对于固定不变的数据尽量使用常量类型代替。

### 2.1.2 基于DES算法的文件完整性检测技术

DES是一种典型的对称加密算法，将CBC模式的反馈机制与传统的加密算法DES结合起来，产生文件相应的校验码，这样不仅可以验证了数据的有效性，也可以防止信息被恶意篡改、伪造或假冒。

DES是一种分组密码加密算法，它以64位分组对数据加密。它的密钥长度是56位（因为每个第8位都用作奇偶校验），密码可以是任意的56位的数，而且可以任意时候改变。其中有极少量的数被认为是弱密钥，但是很容易避开它们，所以保密性依赖于密钥。DES算法的加密过程大致可分成四步:初始置换、迭代过程、子密钥生成和逆置换。该加密算法就是将两种基本的加密方法—替换和移位精心且完美的结合起来。通过有意识的将一些位置替换为其他位的方法使得原明文结构混乱，再通过重新安排达到扩散的目的，安全强度也通过重复应用这种技术来保证。分组密码工作模式是利用分组密码加密消息的技术，当消息的长度大于分组的长度时，首先把消息按分组的长度进行分块，然后利用分组密码进行加密。CBC是一种密码分组链接模式，即是将一种反馈机制引入分组密码子，前一个分组的加密结果被反馈机制引入到当前分组的加密中，换而言之，每一个分组被用于修改下一个分组的加密。每一个密文不仅依赖于产生他的明文分组，而且依赖于所有前面的明文分组。改变任一铭文组中一个字节的内容，都会影响明文的产生。

根据以上理论。首先将8字节的文件校验码附加于文件末尾，当加密后，会产生相对应的密文校验码，然后将其与原始密文校验码进行比较，就可以判断文件是否遭受篡改，保证文件的完整性。其检测模型图见图2-2所示。

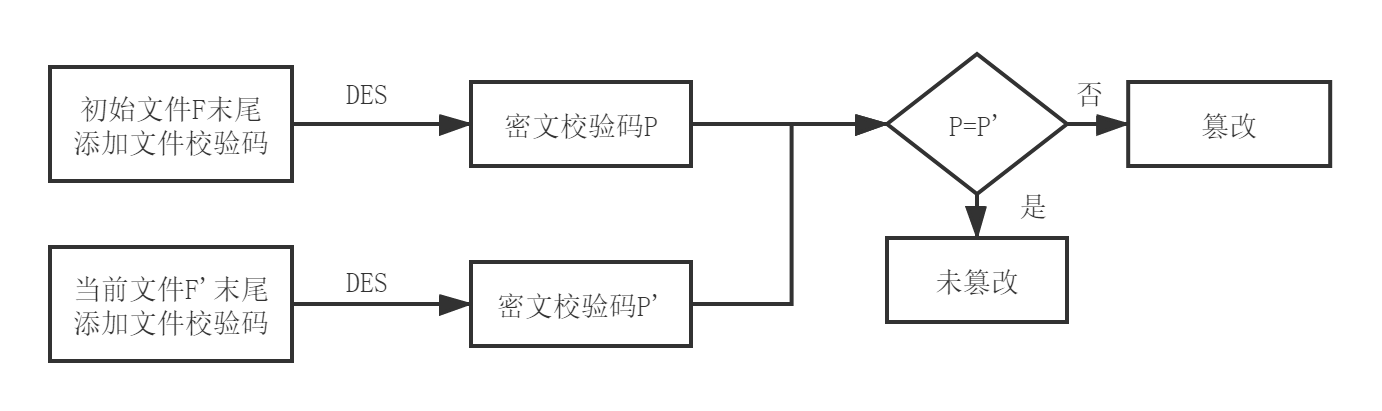


图2-2 基于DES算法的文件完整性检测模型图

因为基于DES的文件完整性检测技术相对繁琐，且消耗系统资源较大，只适用于少量、小型的文件进行文件完整性检测，所以本次课题使用基于MD5算法的文件完整性检测技术。

## 2.2 基于TCP协议的通信技术

随着互联网的普及和信息技术的发展，现代社会已经变成一个由网路服务器和主机节点组成的地球村。网络设备之间通信常用采用的常用模型是ISO（国际标准化组织）定义的七层网络互联模型。目前主流的技术是基于TCP协议的Socket通信，在该系统中企业服务器要接收大量的文件完整性检测请求，需要对服务器的并发处理能力进行良好的设计。本节将详细介绍利用Socket和并发编程技术实现检测端与服务器端的高效可靠通信。

TCP是一种面向连接的、可靠的、基于IP的全双工传输层协议，当应用层向TCP层发送8位字节表示的数据流时，TCP将数据流分割成适当长度的报文段，然后TCP把数据包传给IP层，由它来通过网络将包传送给接收端实体的IP层，再由对端IP层传输TCP层。

基于TCP的Socket通信结构是一个5元组：（协议、源IP地址、源端口、目的IP地址、目的端口）。Socket通信本质上是端到端通信（从源端到目的段），人们通过一个Socket连接就唯一确定了通信的双方及通信的协议，所以一旦Socket连接后，通信两端就可以通过这个Socket连接进行通信。TCP通信典型时序图见图2-3所示。

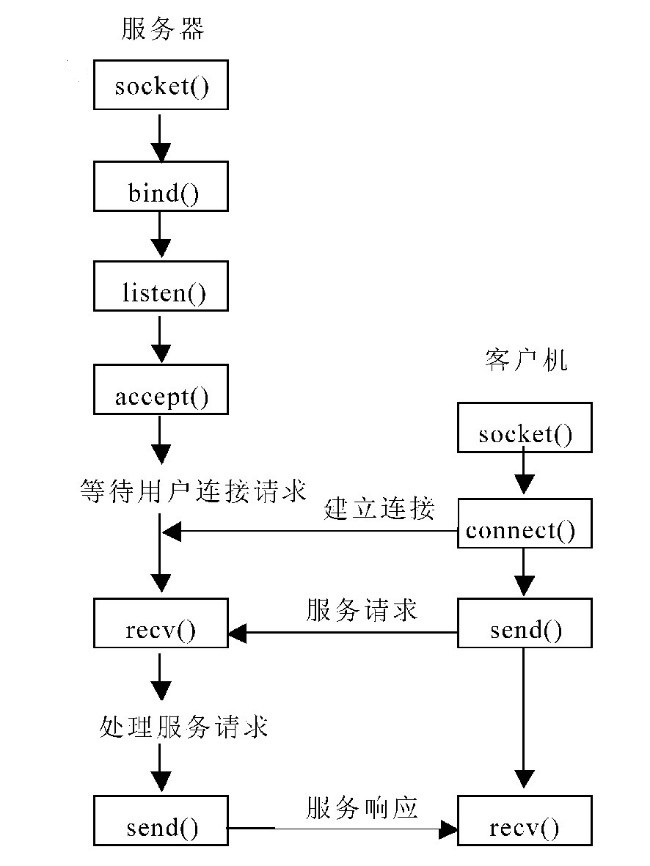


图2-2 TCP通信典型时序图

该系统中建立高性能的通信系统非常有必要，在服务器方面常见的编程模型包括：迭代模型、并发模型、异步模型。下面分别介绍这几个模型。

（1）迭代服务器

迭代服务器模型有唯一的侦听进程，它accept新的Socket连接之后，通过fork新的子进程的方式为这个请求进行专门的服务，直至请求处理完成，直至请求处理完成，之后才会释放在处理该请求过程中被占用的各种资源（包括进程、内存、Socket句柄等）。采用迭代型服务器模型编程最常见的代码是Apache服务器。这种模型的优点是开发简单，缺点是并发处理能力有限（受限服务器的资源），而且服务完成后的资源释放也是一个容易出错的地方，处理不当容易造成资源泄漏， 从而最终导致服务器无法响应新的服务。其流程图见图2-3。

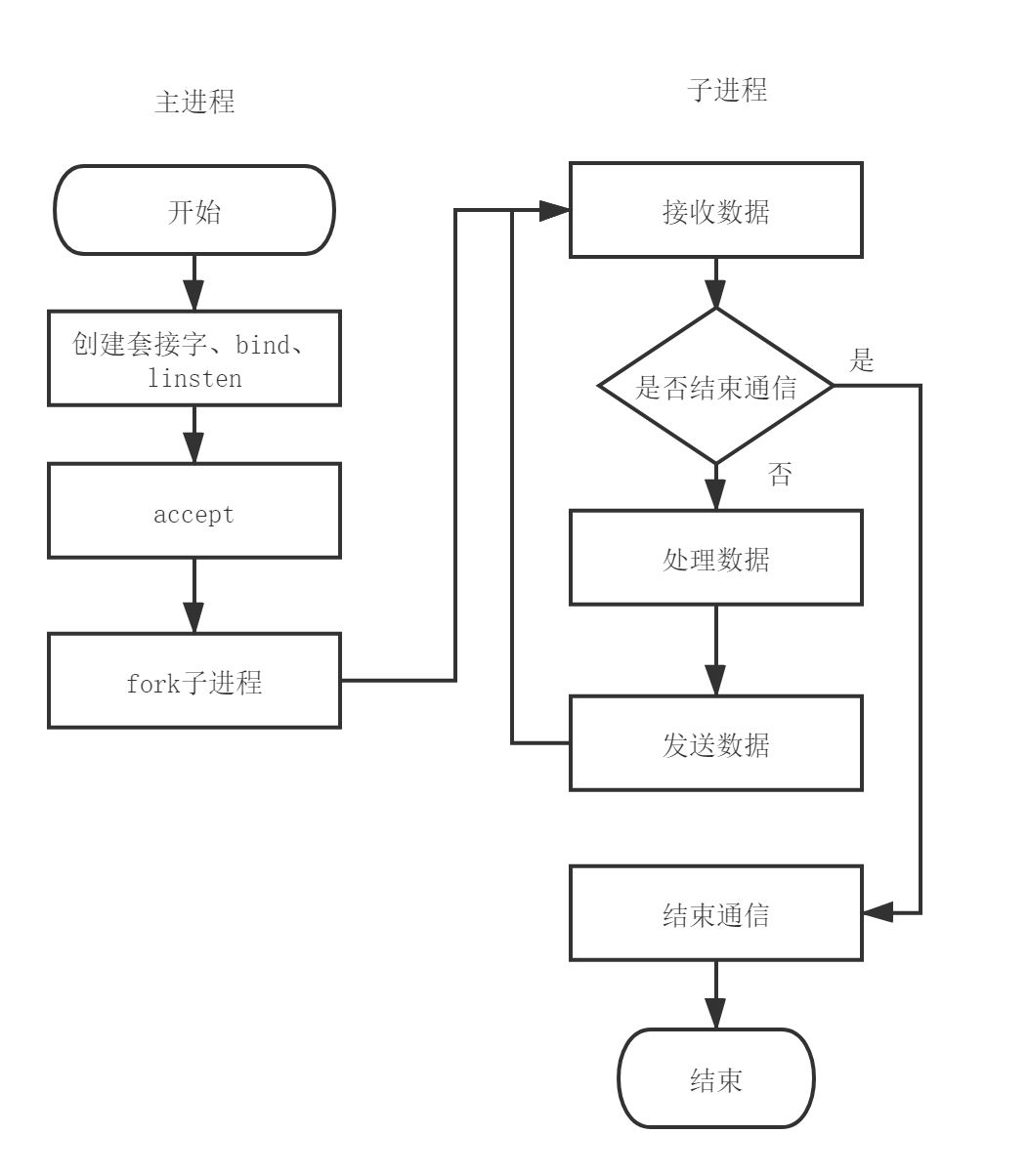


图2-3迭代服务器流程图

（2）并发处理器

并发服务器模型可以同时接受多个客户端的请求。通常有多线程和多进程两种模型。下面以多线程并发为例，在并发服务模型中，由主进程进行侦听和Accept客户端请求，然后把请求放入一个服务队列，然后由有一个处理请求的线程池从请求队列不断地读取Socket请求进行处理，处理完成后线程继续读取队列，直到队列为空。这种服务器开发模型的优点是极大地提高了服务器的并发能力，并且编程也比较简单，不存在迭代模型的进程资源释放问题。缺点是该模型依然受限于服务器本地的资源，并发处理的线程数和请求服务队列受限于服务器的CPU和内存资源情况。

（3）异步服务器

异步服务器模型使用网络异步模型Select或Epoll（目前广泛使用的服务器一般采用Epoll），该模型单个进程就可以同时等待多个Socket 的读写事件，大大地提高了网络操作的效率，目前常见的异步模型典型有：libevent。采用该网络模型的优点是网络操作高效，进程或线程切换开销，可以处理比并发模型更大的并发请求，缺点是编程复杂度较高，容易出错，另外还有一个缺点是业务代码和网络操作代码耦合，维护成本较高。

在本次课题中服务器端使用迭代服务器模型实现通信，这种模型兼顾并发能力高，实现简单的特点。同时检测端发送检测请求将使用多线程技术实现并发地发送检测请求对多台企业服务器的不同文件进行完整性检测。

2.3 系统功能性需求分析

本论文的课题是企业服务器文件完整性检测系统的设计与实现，主要工作是基于MD5算法的文件完整性检测系统和基于TCP协议的通信系统，经过分析得到此系统的功能需求如下：

（1）文件初始信息获取与存储：能够根据客户需要将文件的初始信息，包括文件的大小、修改时间、文件的文件内容摘要值等准确存储在数据库中，以便系统进行文件完整性检测时能将文件的初始信息与文件的当前信息进行对比，从而得到检测结果。文件初始信息获取与存储用例规约如表2-1所示。

表2-1 文件初始信息获取与存储规约

|  |  |
| --- | --- |
| 用例名称 | 结果可视化用例规约 |
| 用例描述 | 该用例规定了获取与存储文件初始信息的过程 |
| 参与者 | 模型部署者 |
| 前置条件 | 已经完成数据库搭建以及编写好相关算法 |
| 后置条件 | 排除程序异常 |
| 基本事件流 | 从企业服务器中获取文件信息并存储至数据库 |

（2）检测结果获取：从企业服务器获取文件的当前信息并与数据库中文件的初始信息进行对比，并得到检测结果。检测结果获取的用例规约如表2-2所示。

表2-2 检测结果获取用例规约

|  |  |
| --- | --- |
| 用例名称 | 检测结果获取用例规约 |
| 用例描述 | 该用例规定了文件前后信息对比的过程 |
| 参与者 | 模型部署者 |
| 前置条件 | 文件初始信息已存储入数据库  已编好当前文件信息获取的算法 |
| 后置条件 | 排除程序异常 |
| 基本事件流 | 将文件前后的信息进行对比并得到检测结果 |

（3）通信系统搭建：检测端能够把检测请求放入一个服务队列，然后由有一个处理请求的线程池从请求队列不断地读取Socket请求进行处理，实现与服务器端的双向信息交互。服务器端使用迭代服务器模型实现与检测端的信息交互。通信系统搭建的用例规约如表2-3所示。

表2-3 通信系统搭建用例规约

|  |  |
| --- | --- |
| 用例名称 | 通信系统搭建用例规约 |
| 用例描述 | 该用例规定了检测端与服务器端信息交互的过程 |
| 参与者 | 模型部署者 |
| 前置条件 | 已编好通信程序 |
| 后置条件 | 排除程序异常 |
| 基本事件流 | 完成信息交互 |

（3）检测请求发送与存储：检测端能够使用搭建好的通信系统向服务器端发送检测请求，并将检测请求信息存储在数据库中，包括检测开始时间、检测方式、检测目录、检测频率、检测服务器IP地址。检测请求发送与存储的用例规约如表2-4所示。

表2-4 检测请求发送用例规约

|  |  |
| --- | --- |
| 用例名称 | 检测请求发送与存储用例规约 |
| 用例描述 | 该用例规定了检测请求的发送与存储过程 |
| 参与者 | 模型部署者 |
| 前置条件 | 已搭建好通信系统及数据库 |
| 后置条件 | 排除程序异常 |
| 基本事件流 | 发送与存储检测请求 |

（4）检测请求接收与检测结果发送：服务器端能够使用搭建好的通信系统接收检测请求并将检测结果发送给检测端，检测请求接收与检测结果发送的用例规约如表2-5所示。

表2-5 检测请求接收与检测结果发送用例规约

|  |  |
| --- | --- |
| 用例名称 | 检测请求接收与检测结果发送用例规约 |
| 用例描述 | 该用例规定了检测请求的接收与检测过程 |
| 参与者 | 模型部署者 |
| 前置条件 | 已搭建好通信系统及数据库 |
| 后置条件 | 排除程序异常 |
| 基本事件流 | 接收检测请求与发送检测结果 |

（5）检测结果可视化：能够对接收到的检测检测进行可视化，根据检测结果发出弹窗警告，最终能够使用该系统将检测结果展示在程序图形界面中。检测结果可视化用例规范如表2-6所示。

表2-6 检测结果可视化用例规范

|  |  |
| --- | --- |
| 用例名称 | 检测结果可视化用例规范 |
| 用例描述 | 该用例规定了检测结果可视化过程 |
| 参与者 | 模型部署者 |
| 前置条件 | 已搭建好通信系统及数据库 |
| 后置条件 | 排除程序异常 |
| 基本事件流 | 展示接收到的检测结果；发出弹窗警告 |

2.4 系统可行性分析

可行化分析主要分析现有技术水平、经济条件等能否顺利完成企业服务器文件完整性检测系统的开发工作，各种配置能否满足开发的需求等。

企业服务器文件完整性检测系统涉及到大量的加密计算，网络交互的信息量较大，而这两个特点非常适合计算机和计算机网络的特点，发挥计算机的数据处理速度快、准确度高、计算机网络信息传输速度快的优势。

本系统使用Python语言进行企业文件完整性检测系统的程序编写，并基于MySQL数据库实现信息存储与提取。这几项相关技术现在均已成熟，发展使用都较好，所以本系统从技术角度来看是可实现的。

本系统的开发，可以通过虚拟机的技术模拟真实企业服务器存储环境，同时MySQL数据库也可部署在本地，因此只需要一台普通PC电脑即可，无需额外的成本投入。综上可行性分析，本论文的企业服务器文件完整性检测系统是可行的。

2.5 本章小结

本章对企业服务器文件完整性检测系统进行了需求分析，介绍了功能需求分析以及系统可行性分析。

其次，对本论文工作相关的理论知识进行了介绍，首先介绍了两种不同的文件完整性检测技术，其中重点分析了MD5算法的改进方法，这是本论文的基础性技术。

之后有介绍了基于TCP的通信技术，对于系统如何使用TCP协议进行信息交互，并对服务器并发模型进行了详细的介绍。

第3章 基于微博的性格分析系统设计

本章主要对企业服务器文件完整性系统进行设计，主要从系统的整体架构、功能设计、功能模块的概要设计、系统数据架构设计、检测端与服务器端设计这些方面展开说明。

3.1 系统整体设计

3.1.1 系统架构设计

从系统架构方面来看，企业服务器文件完整性检测系统自下而上可以划分为数据层、文件信息存储层、信息交互层这三层，具体的层次架构图如图3-1所示。

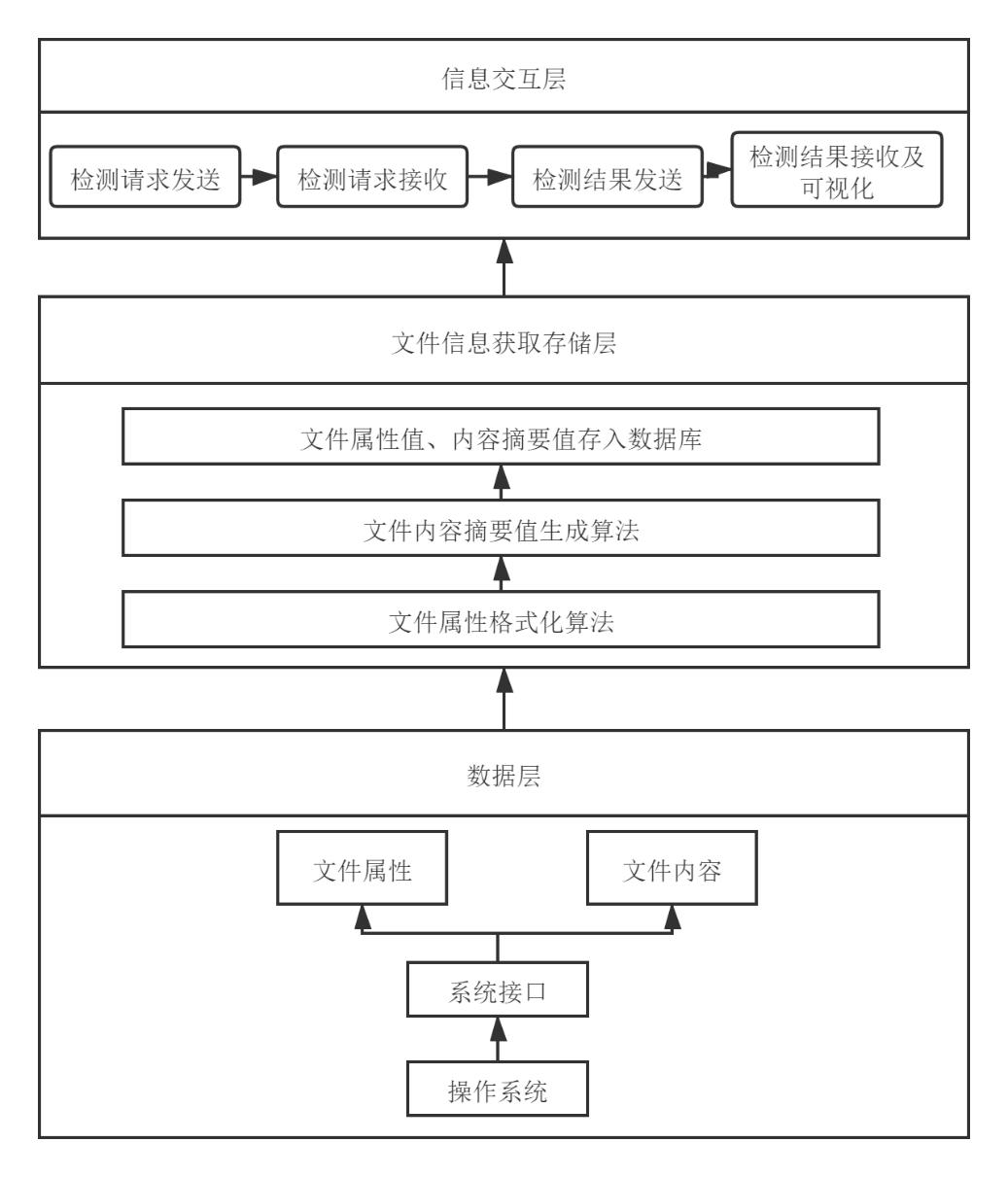


图3-1 系统架构设计图

数据层主要是用来解决文件初始信息获取的问题，根据图3-1所知是调用操作系统的接口来实现的，通过使用read()、stat()、walk()等系统函数可以获取文件目录下的文件名、文件属性、文件内容等信息。将获取的文件属性存储为List类型、并将文件内容分块读取到内存中，等待下一步处理。

文件信息获取存储层主要是利用各种相关算法获取文件初始信息，并存储以适当的结构存储在MySQL关系型数据库中。根据文件数据的处理过程，主要划分为文件属性格式化处理、基于MD5算法的文件内容摘要值生成、文件属性值、内容值存储这三个步骤。

信息交互层主要用来实现检测端与服务端之间的信息交互，具体包括检测端的检测请求发送与检测结果接收、服务器端的检测请求接收与检测结果发送、检测结果可视化。

3.1.2 系统功能设计

根据此系统对于功能性需求方面的分析，可以将系统划分为文件信息获取和存储、文件完整性检测模块和检测结果可视化这三大模块，具体划分如图3-2所示。

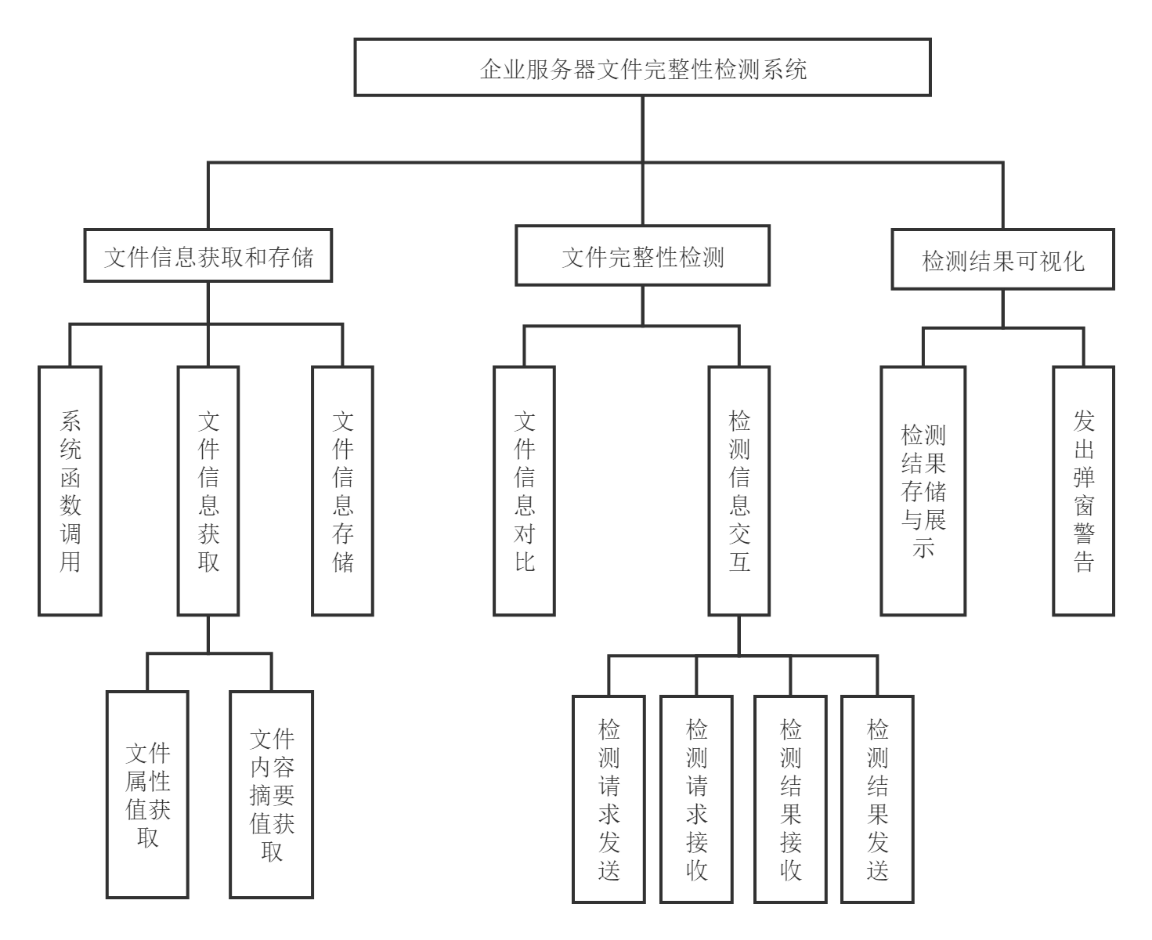


图3-2 系统功能模块划分示意图

文件信息获取和存储模块主要包括系统函数调用、文件信息获取、文件信息存储这三个部分内容。系统函数的功能主要是获取文件的相关属性、将文件内容读入内存。文件信息获取包括对得到的文件的属性进行格式化处理、生成文件内容摘要值。文件信息存储用来实现将文件属性信息合理存储，方便以后进行文件完整性检测使用。

文件完整性检测模块主要包括文件信息对比、检测信息交互这两部分内容。文件信息对比是实现将文件的当前信息与文件的初始信息进行对比，从而得到检测结果。检测信息交互是利用基于TCP的通信系统实现的，利用搭建好的通信系统检测端可以实现检测请求的发送、检测结果的接收，服务器端可以实现检测检测结果的