5-2 win系统安全

2019年4月10日 23:36

3 Windows本地安全机制

- 3.1 用户管理机制
- 3.2 Windows本地身份认证机制
- 3.3 windows访问控制实现方式
- 3.4 Windows中的安全审计机制
- 3.5 文件加密机制

3.1 用户管理机制

- 用户账户(Account)
  - 本地用户账户创建于网络客户机，作用范围限于创建它的计算机，用于控制用户对该计算机上资源的访问。
  - 全局用户账户（域用户账户）创建于服务器（域控制器），可以在网络中任何计算机上登录，使用范围是整个网络。
- 账户类型
  - 内置账户：
    - Administrator:管理员账户，具有最高权限。
    - Guest：来宾账户。
  - 普通用户：由系统管理员建立，通过用户的配置文件存储账户的唯一安全标识SID（SecurityIdentifier）和权限。

安全标识符（SID）

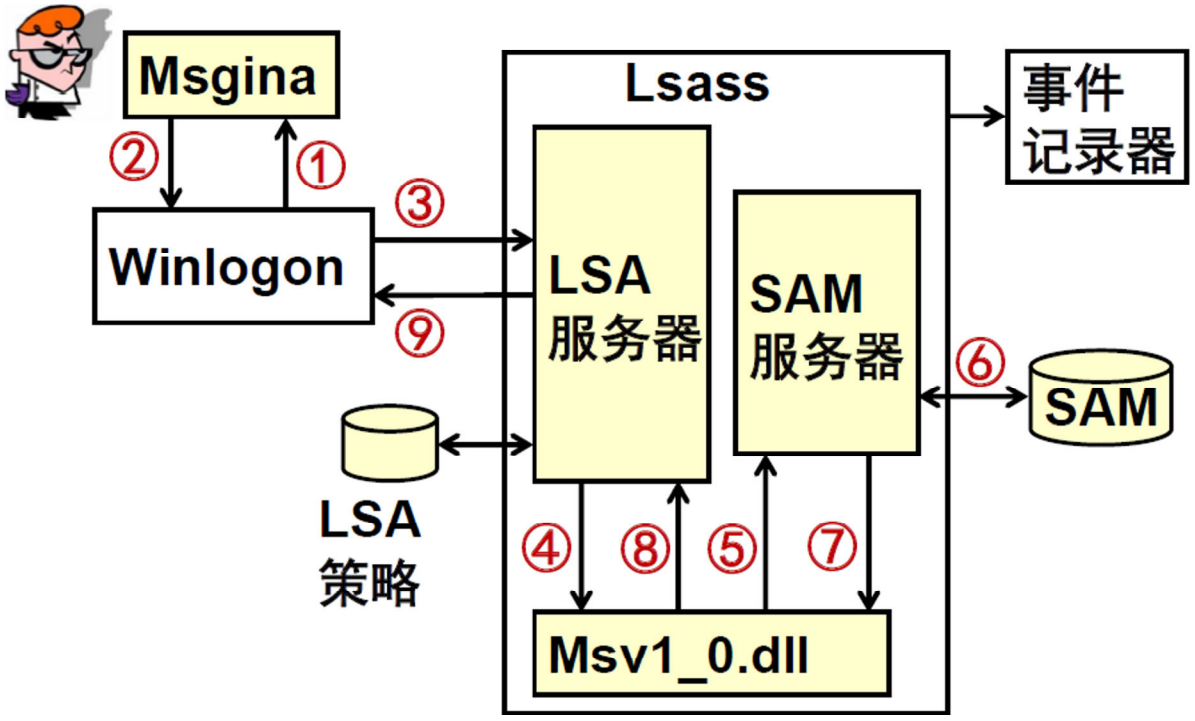
- 每次创建一个用户或一个组的时候，系统会给它分配一个唯一的SID。
  - 用户名与SID一一对应对应。
  - 删除用户后，其SID不会被重用。
- SID的唯一性
  - SID永远都是唯一的，由如下三个参数共同确定以保证唯一性：
    - 计算机名

        - 当前时间
        - 当前用户态线程的CPU耗费时间的总和
    - 不同于名称，永远不会更改
- SID命名规则
    - 例： S-1-5-21-31044058 8- 2 500 36847- 5 803 895 05-500
    -

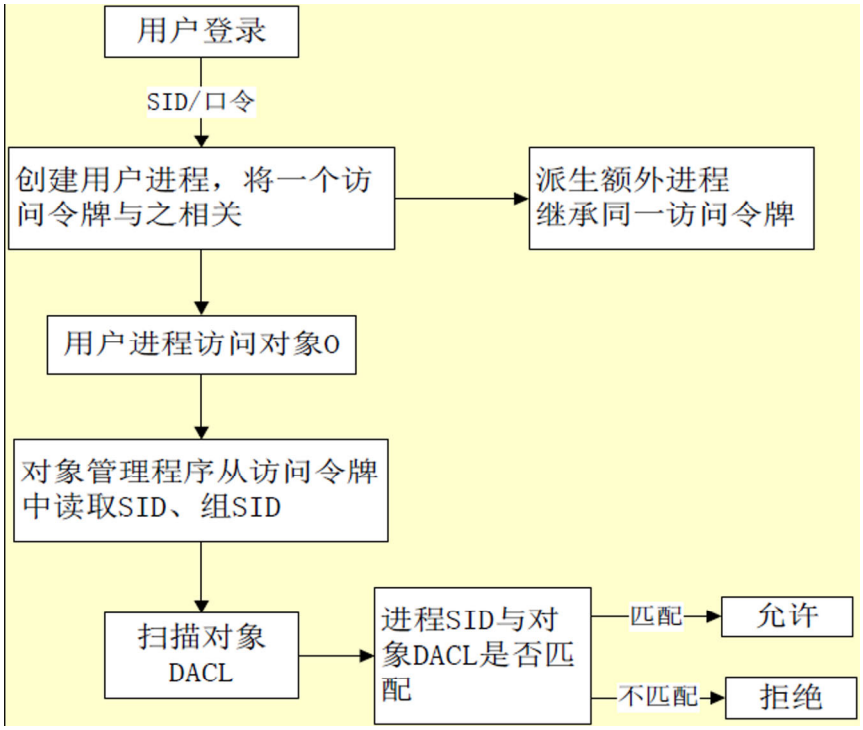
S	1	5	21-31044058 8-2 500 36847- 5 803 895 05	500
表明这一串是SID	SID的版本号	指颁发机构，这里就是NT，值是5	表示一系列的子颁发机构，前面几项是标志域的	标志着域内的帐户和组。一个不是内置账户的值将大于等于1000

- SID查看
    - 开始--运行--输入cmd，在cmd窗口中输入whoami/user，即可查询到当前用户的SID。

3.2 Windows本地身份认证机制



3.3 windows访问控制实现方式

- 

访问令牌（AccessToken）

- 用户通过身份认证后，登录进程会为其创建一个访问令牌，该令牌相当于用户访问系统资源的票证。
- 当用户试图访问系统资源时，将访问令牌提供给Windows系统。
- 由于访问令牌是用户在通过认证的时候由登录进程所提供的，所以改变用户的权限需要注销后重新登录，重新获取访问令牌。
- 用来标识进程或线程（主体）的安全属性
  - 用户SID和组SID
  - 进程权限信息
  - 信息字段
  - 默认的自主访问控制列表（DACL）

安全描述符（SecurityDescriptor）

- 安全描述符是Windows创建对象时所基于结构的一个主要部分，与每个被访问的对象相关联，包含了安全对象的安全信息。
- Windows可识别的每一个对象（文件对象、注册表键、网络共享、互斥量、信号灯、进程、线程、令牌、硬件、服务、驱动）都可以保证其安全。
- 安全描述符包含的安全信息
  - 标记：一组控制位集合，说明安全描述符的含义
  - 所有者：拥有者或基本组对象的安全ID（SID）
  - 自主访问控制列表DACL：指定特殊用户或组的允许或拒绝的访问权限
  - 系统访问控制列表SACL：指定该对象上的哪些操作需要产生审计信息

访问控制列表(AccessControlLists)

- ACL是Windows访问控制机制的核心。

- 当某进程要访问一个对象时，进程的SID将与对象的ACL进行比较，决定是否可以访问。
- ACL：附加到保护对象上的零个或多个访问控制项的顺序列表。
- 每个访问控制项标识用户和工作组对该对象的访问权限。
- 访问控制列表结构：首部＋0或多个访问控制项

ACL的首部

```
typedef struct_ACL{
    BYTE AclRevision;
    BYTE Sbz1;
    WORD AclSize;
    WORD AceCount;
    WORD Sbz2;
}ACL,*PACL;
```

访问控制项ACE(AccessControlEntries)

- ACE由对象的权限以及用户或者组的SID组成。
  - 拒绝访问
  - 允许读取和写入
  - 允许执行
- 结构

```
typedefstruct_ACE_HEADER{
    BYTE AceType;（标识ACE的类型）
    BYTE AceFlags;
    WORD AceSize;
}ACE_HEADER,*PACE_HEADER;
typedefstruct_ACCESS_ALLOWED_ACE{
    ACE_HEADER Header;
    ACCESS_MASK Mask;
    ULONG SidStart;
}ACCESS_ALLOWED_ACE;
```

自主访问控制列表DACL(DiscretionaryACL)

- 由对象所有者控制，决定了用户或用户组可以对该对象执行的操作。
- DACE的类型

  允许

  拒绝

系统访问控制列表SACL(SystemACL)

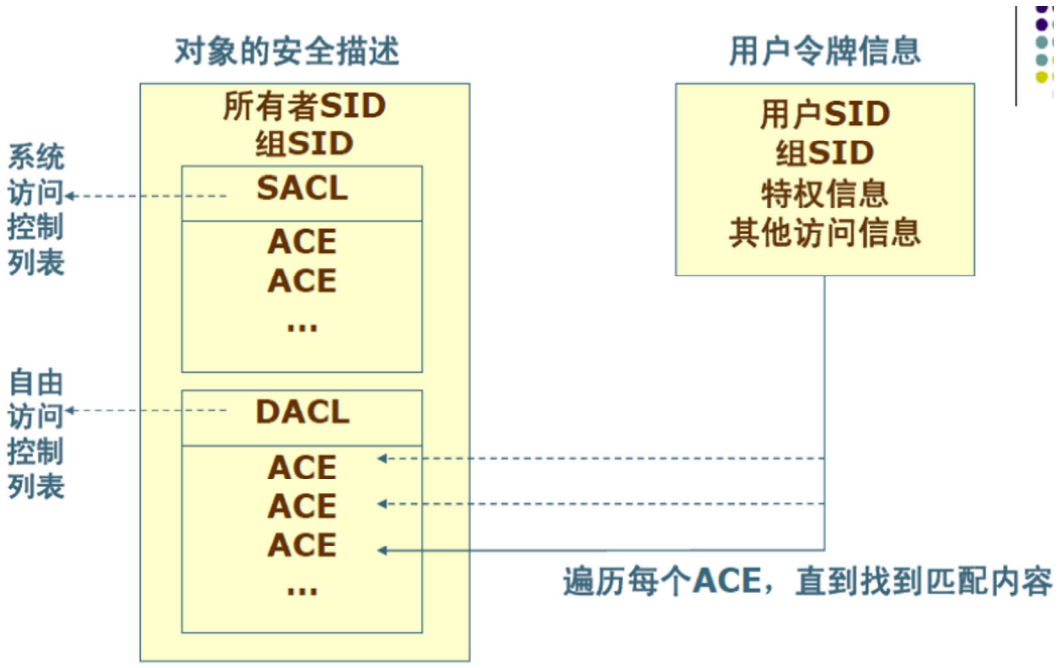
- 确定安全资源的审计策略，SACL描述了哪些类型的访问请求需要被系统记录，包含对象

被访问的时间。

- SACE的类型

  成功

  失败

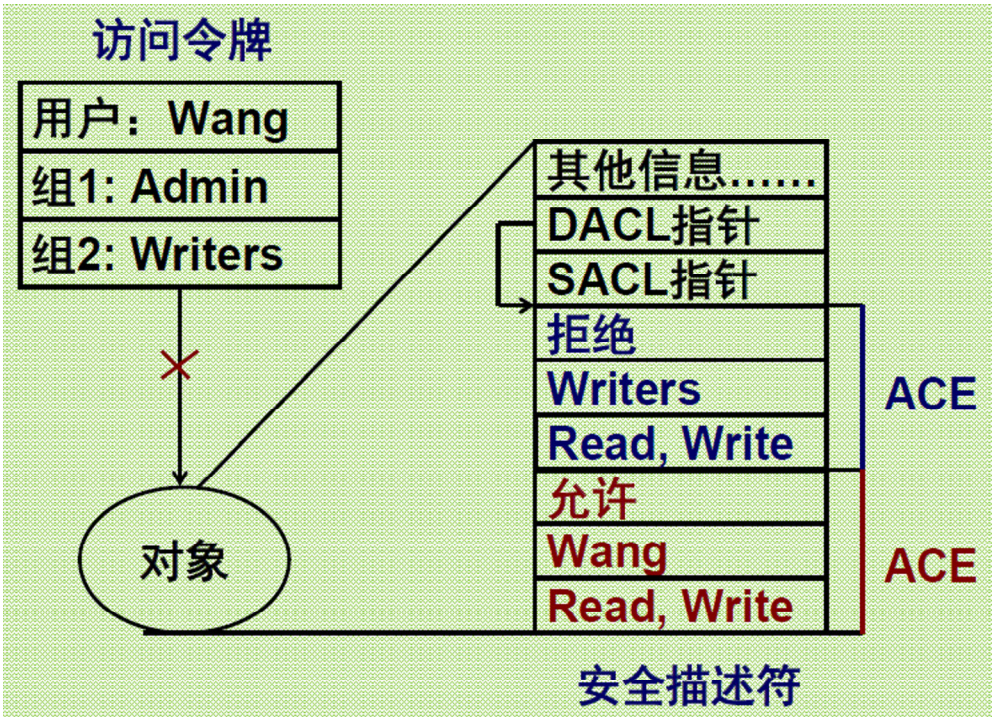


空值

注意区分下述情形

- SD中没有DACL，即DACL=null
- SD中有DACL，但该DACL中的ACE数量为0
- SD中没有SACL，即SACL=null

访问控制过程



3.4 Windows中的安全审计机制

- Windows系统的审计进程默认关闭，需要手动操作打开审计进程，审计目标对象，指定审计失败还是成功的事件，或两者都审计。
    - 成功事件
    - 失败事件

审计事件类型

- 登录事件
    - 如果对登录事件进行审计，用户每次在计算机上登录或者注销时，都会在安全日志中生成一个事件。
        - 本地登录
        - 远程登录
- 账户登录事件
    - 如果审计域控制器上的账户登录事件，就会看到对账户进行验证的域控制器上记录的登录尝试
- 账户管理事件
    - 账户管理事件是指用户或者组的创建、更改或删除事件。
    - 可以确定何时创建了安全主体、什么人执行了该任务
- 对象访问
    - 可以用系统访问控制列表SACL对基于Windows的网络中的所有对象启用审计
- 目录服务访问
    - Windows的活动目录对象也有与其关联的SACL，可以对其进行审计
- 特权使用
    - 每次一个用户尝试使用用户权限都会生成一个事件。
    - 通常只对特权使用失败进行审计
- 进程跟踪
    - 进程跟踪审计可以详细审计计算机上进程的跟踪信息。
    - 事件日志将显示创建进程和结束进程的尝试。
    - 事件日志还会记录一个进程尝试生成一个对象的句柄或者尝试获取对一个对象的间接访问权限的时间
- 系统事件
    - 一个用户或者进程尝试改变计算机环境的某些方面时，会生成系统事件。例如关闭计算机或者更改系统时间
- 策略更改
    - 由于攻击者可能会设法改变审计策略本身，以使其进行的任何操作不会被系统审计到，因此对审计策略被更改的事件进行审计，有助于确定用户的行为是否有攻击环境的企图

(不考)日志文件

- 审计信息以二进制结构形式记录在磁盘文件中。

- 包括：事件名称、事件源、事件号、事件类别等。
    - 系统日志
        - 跟踪各种各样的系统事件，记录由Windows系统组件产生的事件。例如，启动过程加载驱动程序错误或其它系统组件的失败记录在系统日志中
    - 应用程序日志
        - 记录由应用程序或系统程序产生的事件，比如应用程序产生的装载dll(动态链接库)失败信息将出现在日志中
    - 安全日志
        - 记录登录上网、下网、改变访问权限以及系统启动和关闭等事件以及与创建、打开或删除文件等与资源使用相关联的事件。利用系统的“事件管理器”可以指定在安全日志中记录需要记录的事件，安全日志的默认状态是关闭的。

禁止Guest访问日志

- 禁止Guest访问系统日志
    - 修改注册表,HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System下添加键值名称为RestrictGuestAccess，类型为DWORD，将值设置为1
- 禁止Guest访问应用日志
    - 修改注册表,HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application下添加键值名称为：RestrictGuestAccess，类型为：DWORD，将值设置为1
- 禁止Guest访问安全日志
    - 修改注册表,HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security下添加键值名称为RestrictGuestAccess，类型为DWORD，将值设置为1

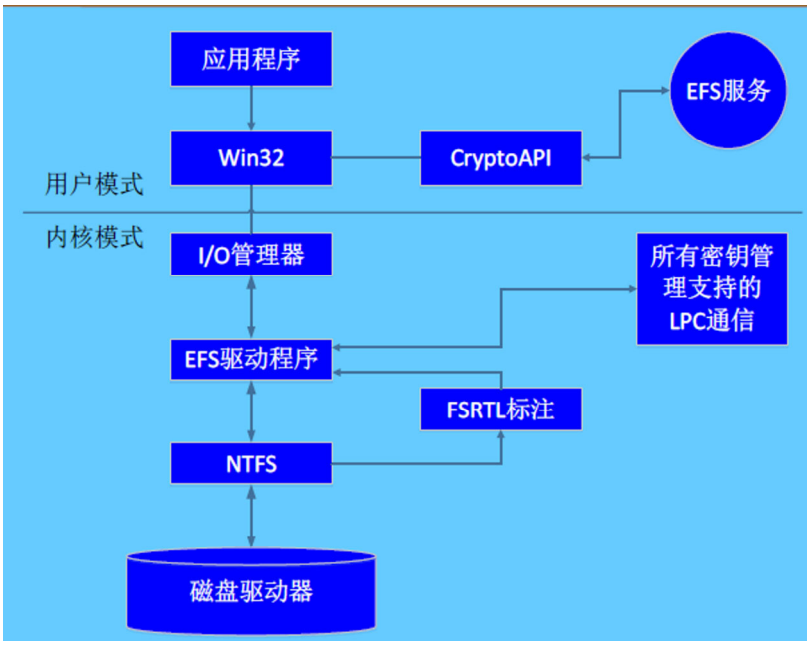
3.5 文件加密机制

Windows的文件系统类型

- FAT16(FileAllocationTable)
    - Microsoft较早推出的文件系统，具有高度兼容性，被用于Windows95或早期系统上。
    - FAT16使用16位的空间来表示每个扇区配置文件的情况，磁盘分区最大达到2GB。
- FAT32
    - FAT32采用32位的文件分配表，提供了比FAT16更为先进的文件管理特征，可以支持的磁盘大小达到2TB。
    - FAT32文件系统可以重新定位根目录、使用FAT的备份副本，分区的启动记录被包含在一个含有关键数据的结构中，减少了计算机系统崩溃的可能性。
- NTFS(NewTechnologyFileSystem)

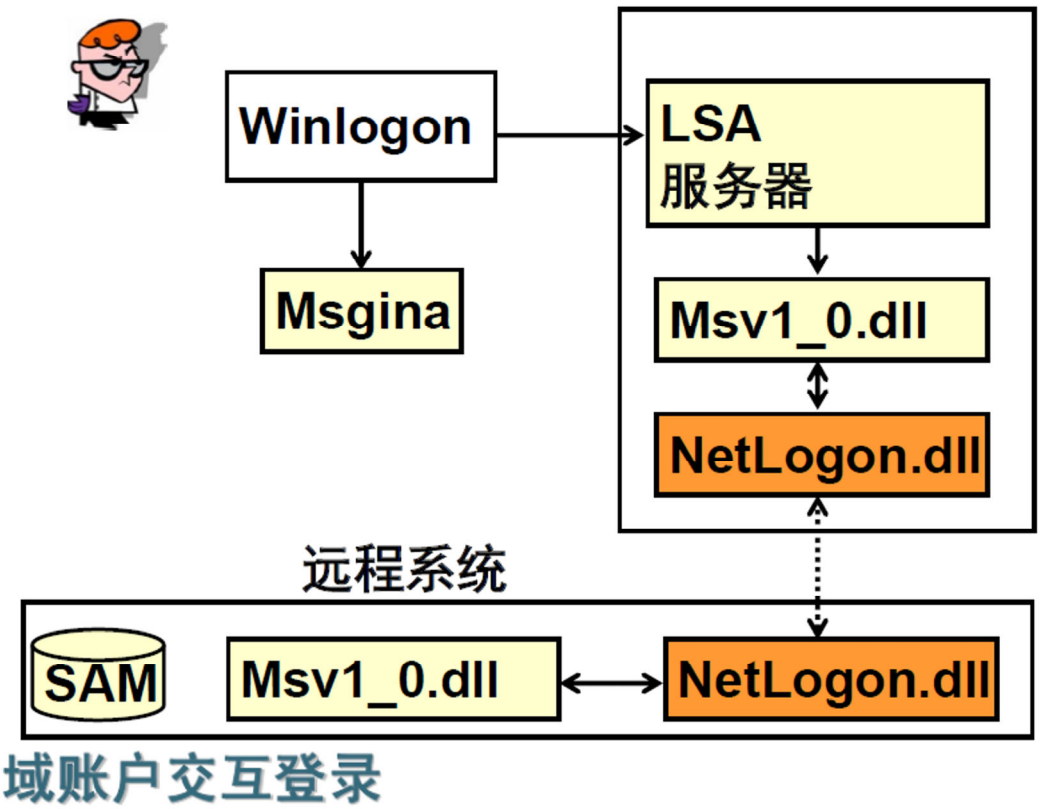
- 伴随WindowsNT的产生而出现和发展的，提供了FAT16和FAT32文件系统没有的可靠性和兼容性，提供了服务器或工作站所需的安全保障。
- NTFS分区支持自主访问控制和拥有权，对共享文件夹无论采用FAT还是NTFS文件系统都可以指定权限，以免受到本地访问或远程访问的影响。
- NTFS支持对分区、文件夹和文件的压缩。

加密文件系统EFS(EncryptingFileSystem)

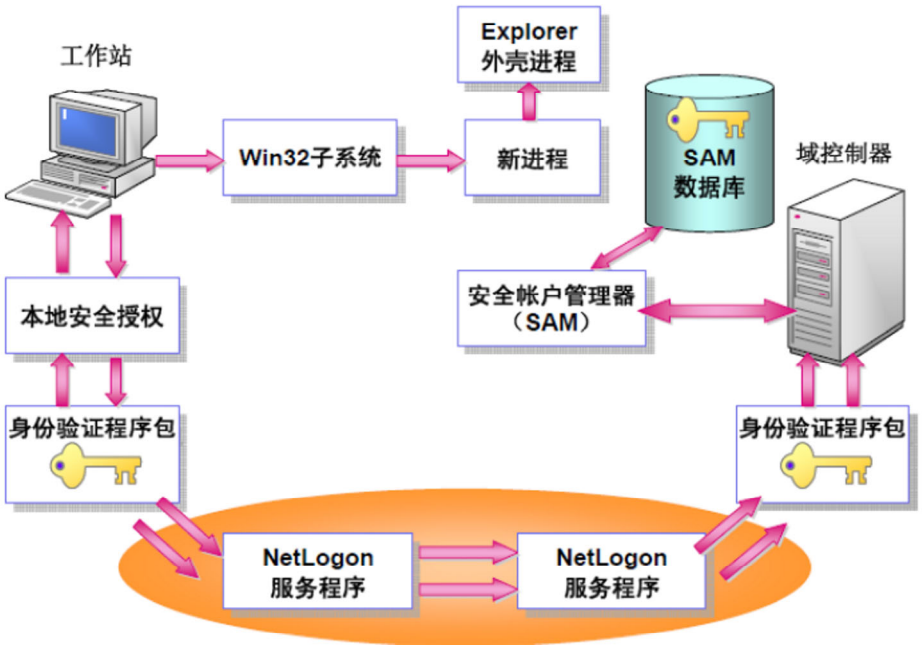
- 用于NTFS文件系统，使用户在本地计算机上安全存储数据。
- 加密用户使用透明，其他用户被拒。
- 不能加密压缩和系统文件，加密后不能被共享，可以被删除、列出文件或目录。
- 结构图
  - 

4 Windows网络安全技术

- 活动目录
  - 实现Windows系统分布式联网的基础。
  - 活动目录是一个包含网络资源（例如计算机、用户、文件、打印机）的数据库，也是一种分布式的目录服务系统。
  - 在分布式环境中，提供给用户和程序访问读取。
  - 按照层次式、面向对象的方式存储信息。
- 网络身份认证



域账户交互登录



- 公钥基础设施
  - Windows从Windows2000起，对公钥基础设施（PKI）做了全面支持。
  - 核心：微软证书服务系统MicrosoftCertificateSerivices。
- IPSec（InternetProtocolSecurity）
  - IPSec是一种开放标准的框架结构，通过使用加密安全服务，确保在Internet协议(IP)网络上进行保密而安全的通讯。
  - Windows系统支持IPSec，与Windows域和活动目录服务集成。
- IIS安全(InternetInformationServer)
- Windows防火墙

    - Internet链接防火墙
    - 只对进入计算机的流量进行过滤
    - 防火墙在安全体系结构中的层次

Windows服务器的安全配置

- 使用NTFS文件系统，便于文件和目录管理
- 关闭默认共享
- 修改共享权限
    - 建立新的共享后立即修改Everyone的缺省权限，不让Web服务器访问者得到不必要的权限。
- 为系统管理员账号更名，避免非法用户攻击。
- 禁用TCP/IP上的NetBIOS。
- 对进站连接进行控制。
    - 只允许80端口
- 修改注册表，减小拒绝服务攻击的风险。
    - 打开注册表：将HKLM\System\CurrentControlSet\Services\Tcpip\Parameters下的SynAttackProtect的值修改为2，使连接对超时的响应更快。

IIS安全配置

- 安全安装
    - 不要将IIS安装在域控制器上。
    - 不要将IIS安装在系统分区上。
    - 修改IIS的安装默认路径。
    - 打上Windows和IIS的最新补丁。
- 用户控制
    - 对匿名用户的权限进行控制。
    - 如无必要，取消Web的匿名服务。
- 登录认证安全
    - 匿名访问：无认证
    - 基本认证：明文认证
    - 请求/响应方式：加密认证，安全性较高，推荐
- 访问权限控制
    - 控制文件夹/文件的访问权限和WWW目录的访问权限。
    - 利用NTFS审核功能。
    - 为IIS中的文件分类设置权限。为Web站点上不同类型的文件建立目录，给它们分配适当权限。例如：静态文件文件夹允许读、拒绝写；ASP脚本文件夹允许执行、拒绝写和读取；EXE等可执行程序允许执行、拒绝读写。
- IP地址控制
- 转发安全
    - 如无必要，禁用转发功能。

- SSL(SecureSocketsLayer安全套接层)安全机制
  - 基于数字证书，加密通讯
  - 基于HTTPS，系统开销大。

5 安全开发

传统软件开发流程的局限性

- 需求分析：功能越多越好、越方便越好
- 软件设计：假设来自相关模块的数据可信
- 软件编码：力图高效、正确地实现功能
- 软件测试：根据功能文档设计测试数据
- 以产品功能为中心，缺乏安全方面的考虑

安全开发生命周期Security Development Lifecycle(SDL)

- 指导原则:把安全特性的考虑渗透到产品生命周期的每一个阶段。SD3+C原则：
  - 安全设计（SecurebyDesign）
  - 安全配置（SecurebyDefault）
  - 安全部署（SecureinDeployment）
  - 交流（Communications）
- 需求分析阶段
  - 建立产品的威胁模型
  - 设计产品的安全功能
  - 制定安全实施计划
- 设计阶段
  - 为每个软件模块建立对应的威胁模型
  - 预测可能的攻击方式
  - 设计相应的防范措施
  - 安全团队进行设计审查
- 编码阶段
  - 遵循严格的编码规范

    StyleCop
  - 禁止使用容易误用的API和库函数
  - 使用分析工具检查代码的安全性

    PREfix/PREfast/FxCop
- 测试阶段
  - 严格测试威胁模型指出的高风险代码
  - 广泛采用“FUZZ”测试
  - 进行渗透测试