公告

昵称：jankie
园龄：8年3个月
粉丝：56
关注：3
+加关注

< 2011年8月 >

日	一	二	三	四	五	六
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

搜索

找找看

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

随笔分类(223)

Active Directory(19)
Linux Service Mgt(12)
Linux Shell Mgt(31)
Linux System Mgt(77)
Linux Tools(3)
Linux管理及服务
MySQL&Storage(2)
Storage(1)
Virtualization_Cloud(6)
Windows Server(54)
系统管理脚本(14)
运维监控管理(4)

随笔档案(314)

2013年6月 (1)
2013年5月 (1)
2013年4月 (2)
2013年2月 (4)
2013年1月 (12)
2012年12月 (5)
2012年11月 (6)
2012年10月 (10)
2012年8月 (7)
2012年7月 (5)
2012年6月 (4)
2012年5月 (8)
2012年4月 (1)
2012年3月 (15)
2012年2月 (2)
2012年1月 (24)
2011年12月 (8)
2011年10月 (2)
2011年9月 (11)
2011年8月 (10)
2011年7月 (9)
2011年6月 (9)

Kerberos介绍（全）

微软Windows Server 2003操作系统实现Kerberos 版本5的身份认证协议。Windows Server 2003同时也实现了公钥身份认证的扩展。Kerberos身份验证的客户端实现为一个SSP（security support provider），能够通过SSPI（Security Support Provider Interface）进行访问。最初的用户身份验证是跟Winlogon的单点登录架构集成在一起的。Kerberos的KDC（Key Distribution Center）跟Windows Server 2003的域控制器DC（domain controller）上的安全服务集成在一起，KDC使用域的活动目录数据库作为它的安全帐户数据库，缺省的Kerberos实现要求活动目录的支持。

这个主题将解释Windows Server 2003是怎样支持Kerberos V5协议及其扩展的。

一、 什么是 Kerberos 身份验证

Kerberos V5身份验证协议提供一个在客户端跟服务器端之间或者服务器与服务器之间的身份验证机制 （并且是相互的身份验证机制）

Windows Server 2003把Kerberos V5身份验证协议实现为一个能够通过SSPS（Security Support Provider Interface）的SSP（security support provider），另外，Windows Server 2003还通过使用智能卡的公共密钥证书（public key certificates）进行初始身份验证来扩展此协议。

Kerberos的密钥分发中心 KDC（Key Distribution Center）使用活动目录的服务数据库作为自己安全帐户数据库。NTLM 和 Kerberos的缺省实现都需要活动目录的支持。

Kerberos V5协议假设客户端和服务端的最初信息交换发生在开放的网络环境中，在网上传输的数据包能够被监视并能被任意修改。这个假设的环境，跟现在的因特网非常相似，攻击者可以非常容易的伪装为一个客户端或者一个服务器，也能很容易的窃听和篡改合法客户端和服务端之间的通讯。

微软的Kerberos V5协议实现是：

Windows Server 2003的缺省身份认证

Kerberos V5协议成为Windows 2003的缺省身份验证，Windows Server 2003还为了能支持Windows NT Server 4.0等非Kerberos的操作系统还支持NTLM协议。

基于**RFC 1510**及其修订草案

Kerberos协议是成熟的、广泛应用的、开放的标准，微软Kerberos V5协议的实现遵循RFC的标准，因此能提供跟其他实现的互操作。

可扩展性

Kerberos 架构允许你指定另外的或者可以替换的安全方案。并且，可以通过智能卡的公钥/私钥来提供缺省的共享安全密钥过程。

1、 Kerberos身份认证带来的好处

Kerberos V5协议比NTLM协议更安全、更灵活，更有效，用Kerberos身份验证能够获得的好处是：

1.1、 身份委派

当Windows的服务为某个客户端访问资源时会扮演这个客户端，在多数情况下，在本机上一个服务能够为客户端完成访问资源的工作，因为NTLM和Kerberos都能为服务提供需要扮演客户端的信息。可是，在分布式应用被设计为前端服务扮演客户端连接到在其他服务器上的后端服务，Kerberos V5协议包括一个允许服务扮演客户端连接到其他服务器上的服务的代理机制，NTLM则没有这样的功能。

Interoperability.

1.2、 互操作

微软的Kerberos V5实现是基于IETF的推荐标准规范。这样，Windows Server 2003的Kerberos V5实现就为其他使用Kerberos V5协议的网络的互操作打下了基础。

1.3、 更高效率的身份验证

对于NTLM，为了验证每一个客户端，应用服务器必须连接到域控制器以证实客户端身份。对于Kerberos V5身份验证协议，服务器不用去连接域控制器，相应的，服务器可以检验客户端提供的验证票。客户端可以为特定的服务获取一次验证票并在一次登录过程中反复使用这个验证票。可更新的会话票据（session tickets）替代了pass-through authenticatio（不知道怎么翻译）。

1.4、 相互身份验证

通过使用Kerberos协议，在网络连接的一端都可以验证网络另一端的声明是它自己的实体。虽然NTLM允许服务器验证客户端的身份，但是它没有提供客户端验证服务端身份的功能，也没有提供服务器验证另一个服务器身份的功能。NTLM被设计为假设服务器都是真实的网络环境，Kerberos则没有这个假设。

2、 Kerberos V5协议标准

Kerberos身份验证协议几十年前起源于麻省理工学院，是由"Athena"项目的工程师开发的。第一个公开发行的版本是Kerberos版本4。在被广泛的使用后，协议的开发者发布了Kerberos第五版本。

Kerberos V5现在成为IETF的标准，Windows Server 2003中Kerberos V5的实现严格的遵循了RFC 1510定义的标准，另外，Kerberos消息中的安全令牌（security tokens）的格式和机制遵循RFC 1964定义的标准。

Kerberos V5协议规定了以下机制：

l 验证用户身份。当一个用户需要获取访问一个服务器的权利，服务器需要验证用户的身份，考虑一个场景，用户声称他是，比如，Alice@tailspintoys.com。因为访问资源是基于身份关联的许可，服务器必须确定用户就是他自己声称的用户。

l 安全的打包用户名，用户名(用户的主名，在本例中就是Alice@tailspintoys.com)，和用户的身份信任凭证（credentials）被打包在一个叫做票据（ticket）的数据结构中

l 安全的传送用户信任凭证。票据被加密后，Kerberos消息在网络上传送用户的信任凭证（credentials）。

注意：

虽然Kerberos协议验证用户的身份，它并不授权访问。这是个重要的区别。在其他情形中的票据，象驾驶执照，就同时提供了身份和驾驶车辆的许可。Kerberos的票据仅仅用来证明这个用户就是它自己声称的那个用户。在用户身份得以确认后，本地的安全权限将决定给予访问权限或者拒绝访问。

2.1、 密钥

Kerberos消息被多种加密密钥加密以确保没人能够篡改客户的票据或者Kerberos消息中的其他数据。

l 长期密钥（Long-term key）

一个密钥（只有目标服务器和KDC知道），并用来加密客户端访问这个目标服务器票据的密钥。

l Client/server会话密钥（session key）

一个短期的、单此会话的密钥，是在用户的身份和权限已经被确认后由KDC建立的用于这个用户的跟某个服务器之间的加密往来信息使用的密钥

l KDC/用户 会话密钥（session key）.

是KDC跟用户共享的一个密钥，被用于加密这个用户跟KDC之间的消息。

Kerberos V5协议使用了对称加密和非对称加密两种加密技术

因为大多数Kerberos的加密方式是基于只用于KDC和用户之间或者KDC和网络服务之间的密钥，Kerberos V5被设计为采用对称加密，即使用同一个密钥来加密和加密消息。

微软的Kerberos协议实现能够使用有限的非对称加密，一个私钥/公钥对被用于加密和解密来自客户端或者网络服务的初始验证信息。

2.2、 Kerberos 身份验证防止数据包重用

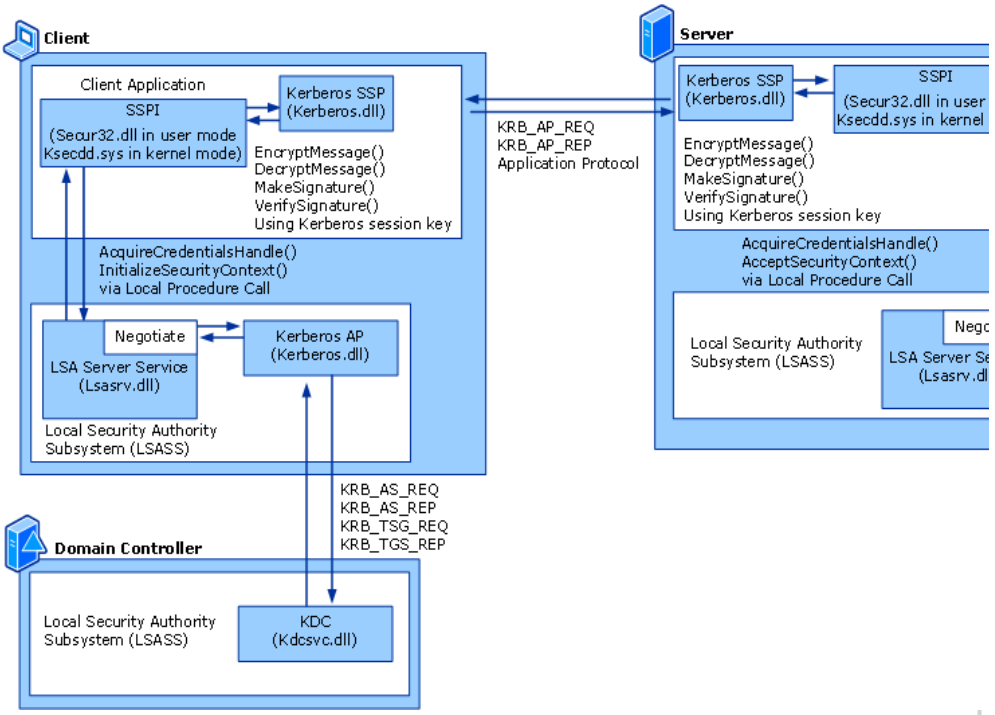
Kerberos身份验证机制建立并安全的传送一个带有客户票据的信任凭证（通常基于一个唯一的时间戳），信任凭证是唯一并且一次使用有效。这个限制使有人获取并重用客户端票据或者尝试偷取客户的身份的可能性降到最小。

3、 Kerberos V5协议的扩展

Windows Server 2003实现Kerberos V5协议的扩展，这个扩展在初始身份认证时采用公钥证书来替代常规的对称加密密钥。这个改进允许协议支持用智能卡交互登录。公钥身份验证扩展是基于IETF工作组的草案协议。

4、 Kerberos身份验证的相关技术

下图显示了Windows Server 2003中Kerberos身份验证同其它技术如何配合的。依赖是客户端或服务端应用是用户模式（user-mode）还是核心模式（kernel-mode）的应用，他们分别使用Secur32.dll 或者Ksecdd.sys，调用SSPI跟Local Security Authority Subsystem (LSASS)通讯



下表是参与kerberos身份验证的组件的描述

Component	Description
Kerberos.dll	被用来口令或者智能卡交互式登录实现工业标准的协议SSP。它也是Windows 2000 和 Windows Server 2003首选的身份验证方式。
Kdcsvc.dll	Kerberos密钥分发中心（KDC）服务，它回应客户端票据授权票（ticket-granting tickets）的申请
Ksecdd.sys	在核心模式（kernel-mode）下用户跟LSASS通讯的核心安全设备驱动
Lsasrv.dll	LSA服务，强制安全策略和担当LSA安全包管理器
Secur32.dll	Secur32.dll 是在用户模式（user mode）下实现SSPI的组件

Windows Server 2003是用SSP来实现Kerberos V5身份验证协议，是操作系统提供的一个动态链接库（DLL），系统使用Kerberos SSP, Kerberos.dll，是身份验证的第一选择。在LSA为一个交互式登录的用户建立了一个安全的上下文，为了支持Kerberos信息的签名和封装，正在运行的用户安全上下文装载另外一个Kerberos SSP实例

因为Kerberos身份验证协议是Windows Server 2003首选协议，所有域服务都支持Kerberos SSP，包括：

- AD活动目录要求使用LDAP（Lightweight Directory Access Protocol）
- 使用RPC的远程服务或workstation management
- 客户－服务器身份验证
- 使用Common Internet File System/server message block (CIFS/SMB)的远程文件访问
- 分布式文件系统管理
- IIS的intranet身份验证
- Internet Protocol security (IPSec)的安全验证
- 为域用户和计算机发放证书的请求

5、 Kerberos 身份验证依赖于

本节讨论和该书Kerberos身份验证依赖项以及和他们的关系

5.1、 操作系统

Kerberos 身份认证依赖于客户端功能，这些功能内建于Windows Server 2003、Windows XP、Windows 2000操作系统。如果一个客户端、域控制器或者目标服务器运行于更早的操作系统下，那它就不天然的支持Kerberos 身份验证。

5.2、 TCP/IP网络连通性

一旦Kerberos 身份认证发生，在客户端、域控制器和目标服务器之间必须有TCP/IP网络连接，关于TCP/IP更多的信息，参考"TCP/IP Technical Reference."

5.3、 域名系统

客户端使用全限定名fully qualified domain name（FQDN）访问域控制器，DNS必须能够保证客户端能够获得这个域控制器的地址。最好不要使用DNS主机文件，关于DNS的更多信息，参看"DNS Technical Reference."

5.4、 域活动目录

Kerberos 身份验证不支持更糟的操作系统，比如Windows NT 4.0。你必须使用活动目录服务中的用户和计算机，本地帐户和Windows NT域帐户不能被由于Kerberos 身份验证

5.5、 时间服务

为了Kerberos 身份验证能正常的发挥作用，在网络中的所有域和森林使用相同的时间源以保证网络中的所有计算机时间同步。一个活动目录域控制器担当权威的时间源，它保证所有的域具有相同的时间。更多信息，参看"Windows Time Service Technical Reference."

5.6、 服务主体名

服务主体名(SPNs) 是运行在服务器上服务的唯一标识符。每一个使用Kerberos 身份验证的服务都需要有个SPN以使客户端能够在网络上标识这个服务。没有正确的设置SPNs, Kerberos 身份验证就是不可能的

二、 Kerberos V5身份验证协议如何工作

Kerberos V5身份验证协议（RFC 1510定义），提供一个在开放的、潜在不安全的网络环境中验证主体身份的方法。这一节讨论RFC标准的Kerberos V5在Windows Server 2003如何使用

这节分为以下四个子章节：

- Kerberos SSP结构：说明Windows Server 2003的SSPI怎样提供一个访问SSP的机制。
- Kerberos 物理结构：讨论Windows Server 2003中实现Kerberos 身份验证的组件。这些组件包括密钥、票据和密钥分发中心（KDC）
- Kerberos V5身份验证协议过程和相互作用：说明Kerberos 身份验证在不同的情形下怎样被使用，Kerberos 消息的细节，并讨论其他相关技术。例子将给出完整的过程，包括没有被Kerberos 身份验证协议定义的相关的组件和过程。一些过程（比如怎样发现一个验证服务，什么信任证书会被通过，信任证书被存放在哪）在Windows系统是特殊的，可能与其他的Kerberos 协议的实现有所不同。
- Kerberos V5协议使用的网络端口：以表格形式展现在Kerberos 身份验证期间使用到的端口。

1、 Kerberos SSP架构

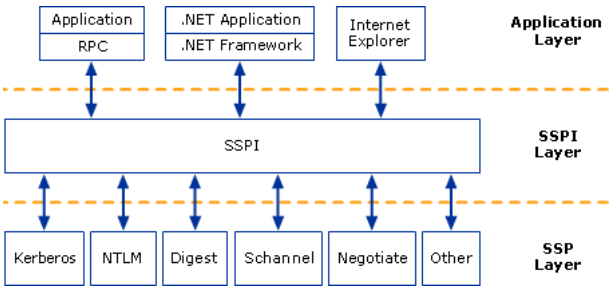
Windows Server 2003 以Security Support Provider (SSP)来实现Kerberos V5身份验证协议，一个操作系统的动态链接库（DLL）。Windows Server 2003另外还包括针对NTLM的SSP。缺省的启动时Windows Server 2003的Local Security Authority (LSA) 把两个SSP都装载进来，系统能够两个SSP中的任意一个来验证网络登录和客户端/服务器的连接。使用哪一个SSP取决于连接另一端的机器的能力和个别应用的参数选择。

Microsoft SSPI是Windows Server 2003中身份验证的基础，就是说，要求身份验证的应用和底层服务都使用SSPI接口。

SSPI是Generic Security Service API (GSSAPI)在Windows Server 2003中的实现，关于GSSAPI的更多信息，请参考RFC 2743 和 RFC 2744。

Windows Server 2003中默认的SSPs（Negotiate (SPNEGO), Kerberos, NTLM, Schannel, 和摘要身份验证（Digest authentication protocols））以DLL的形式插入到SSPI，其它的SSP如果能够同SSPI交互也能被插进来。

SSPI 架构图示：



Windows Server 2003中的SSPI提供一个在客户端和服务端已存在的连接上传送身份验证令牌的机制。当两个实体为了能相互安全的通讯需要身份验证时，身份验证的请求被路由到SSPI，SSPI不管当前正在使用什么网络协议完成身份验证过程，并返回一个透明的（transparent）两进制大对象给连接的另一边的应用。SSPI允许一个应用在一个机器上或网络上使用多种安全模型而不用改变安全系统的接口。

下表是被插入到SSPI的组件的描述。表中的Windows Server 2003中的每个协议都以不同方法来提升不安全网络环境中的通信的安全性。

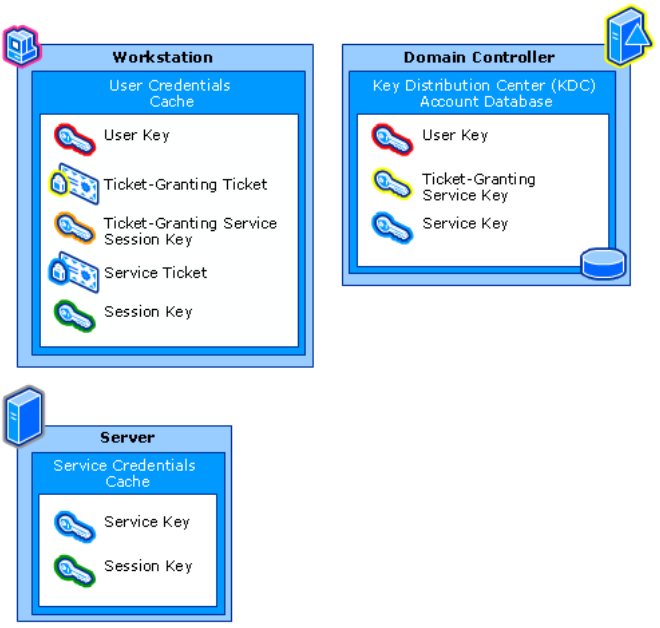
SSP 层组件：

组件	描述
Kerberos V5 身份验证	一个用口令或者智能卡交互登录的工业标准。它也是Windows 2000 and Windows Server 2003首选身份验证的方式。
NTLM 身份验证	一个质询回应（challenge-response）协议使用来提供比Windows 2000更早的系统的兼容性
摘要身份验证	用于Windows Server 2003 LDAP和web身份认证的工业标准，摘要身份的信任凭证被MD5散列后在网上传输。
Schannel	一个实现安全套接层（SSL）和传输层安全（TLS）因特网标准身份验证协议的SSP。Schannel被用于基于web的服务器身份验证，象当一个用户试图访问一个安全的web服务器。
Negotiate	被用于协商使用哪个特定协议的SSP。当一个应用通过SSPI登录到网络，它能够指定一个SSP处理这个登录请求，当应用指定Negotiate，Negotiate分析这个请求并选取对客户端配置的安全策略来说最好的SSP处理这个请求

2、 **Kerberos** 物理结构

Kerberos在开放的网络（在网络上传输的数据包可以被随意的监听和修改）上提供客户端和服务端双向的身份验证的机制。为了提供安全的身份验证，Kerberos身份验证使用对称密钥、加密的对象和Kerberos服务。

Kerberos 组件



2.1、 身份验证中使用的密钥

2.1.1 为什么需要密钥?

Kerberos协议严重的依赖于一个验证技术共享密钥。基本的共享密钥的概念十分简单：如果一个秘密只有两个人知道，任何一个人都可以通过他们之间共享的秘密来确定对方的身份。

比如，假设Alice经常给Bob传送信息，Bob在使用这个信息前需要确定这个信息真的就是Alice传送过来的。他们决定通过在他们选择一个只有他们两个人知道一个秘密来解决这个问题，如果一个信息声称来自Alice的，并能以某种方式出示发送者知道的密码，那Bob就能确信发送者就是Alice。

剩下的问题就是Alice和Bob需要解决Alice怎么出示她知道的密码。她可以把密码包含在她发送的信息里面，多半是在最后的签名 — Alice, Our$ecret。 这是简单而有效率的如果他们能确信没有人能够看到他们之间的传送的信息。不幸的是，他们的信息通过的网络上有个用户Carol，他用网络分析工具扫描网络通讯希望有一天能够发现这个密码。因此，Alice不能在她发送的信息里提供密码。为了保持密码的保密性，她必须要能出示这个密码但又不能暴露它。

Kerberos协议用密钥来解决这个问题。替代共享的密码，通讯的双方共享一个密钥，他们使用这个密钥的知识验证另一方的身份。为了使这个技术正常工作，他们共享的密钥必须是对称的，就是说，一个密钥既能够用来加密，又能够用于解密。密钥的一部分知识用来加密，另一部分知识用来解密。

Kerberos身份验证依赖于几个不同类型密钥，密钥的类型有长期对称密钥，长期非对称密钥和短期对称密钥。身份验证协议被设计为使用对称加密，意味着发送端和接收端使用共享的密钥加密和解密。

2.1.2 长期对称密钥：User, System, Service, Inter-realm Keys

长期对称密钥源于一个密码。明码文本的密码通过加密功能被转换为一个密钥（所有的Kerberos V5的实现必须支持DES-CBC-MD5），这里使用DES-CBC-MD5加密的方式把明码文本的密码转换一个密钥。

User keys

当建立一个用户时，用户口令将被用来建立一个user key。在活动目录域，user key和用户对象一起存放在活动目录里，在工作站，用户对象在用户登录时被建立。

System keys

当一个工作站或服务器加入到域，它会收到一个密码，跟用户帐户相似，这个密码用来建立一个system key。

Service keys

服务用的密钥基于登录到服务的帐户的密码

所有在同一领域的KDC使用同一个服务密钥。这个密钥基于跟krbtgt帐户关联的口令，每个活动目录月都有这个内建的帐户。

Inter-realm keys

为了跨领域的身份验证的需要，KDC必须共享一个inter-realm key，因为他们共享一个inter-realm key所以领域能够彼此信任，有父子关系的活动目录域共享一个inter-realm key。这个inter-realm key基于Windows 2000 和 Windows Server 2003的信任传递。如果一个shortcut trust关系被建立，两个域将会交换这个密钥

Kerberos SSP 加密密钥长度

Kerberos SSP支持不同的加密类型，密钥的长度也各不相同。虽然密钥的长度决定着这种加密方法保护程度，密钥的长度不意味着票据的强度。下表列出Kerberos SSP支持的不通加密方法的密钥长度。

不通加密类型的密钥长度

加密算法	密钥长度
RC4-HMAC	128

DES-CBC-CRC	56
DES-CBC-MD5	56

- 关于明文密码通过DES-CBC-MD5产生密钥：

Kerberos 定义了一种对用户密码进行处理以生成一个密钥的算法。在获得 TGT 的过程中 Kerberos 客户机将用这个密钥进行解密，这就是DES-CBC-MD5，CBC（密码分组链接 cipher block chaining）模式下的 DES（数据加密标准）。DES 是一个 FIPS（联邦信息处理标准 Federal Information Processing Standards）发表，它描述了一种将要加密的数据（纯文本）和密钥作为输入传递给加密过程的加密算法。根据 DES 算法对密钥和纯文本统一处理以生成一个加密的（密文）形式的纯文本数据。

CBC 是一种加密操作模式，其中纯文本数据分为同样大小的数据块。例如，在 64 位 DES-CBC 加密中，数据会分为 8 字节的块。如果纯文本数据中的字节数不是您希望每一个块所具有的字节数的整数倍，就要在最后一块中加上适当的数量的字节以使它的大小与其他的块相同。

然后创建一个与您的块具有同样大小的字节数组。这个字节数组称为 初始矢量 (IV)。Kerveros 规范定义了所有基于 Kerberos 的应用程序的初始矢量（类似地，其他使用 DES-CBC 的规范定义了它们使用的 IV 值）。之后，取这个 IV、纯文数据的第一块以及密钥并根据 DES 算法对它们共同进行处理，以构成对应于纯文本数据第一个数据块的密文。然后取第一个数据块的密文形式作为第二个块的初始矢量并进行同样的 DES 加密过程以生成第二个纯文本数据块的密文形式。以这种方式继续一块接一块地生成每一个块的密文形式。最后，串接所有密文块以得到全部纯文本数据的密文形式。

下面讨论用户密码经过DES-CBC-MD5处理生成密钥的过程

将用户密码、KDC 域名和用户的用户名串接到一起以构成一个字符串。Kerberos 利用这个串接的字符串而不仅仅是密码生成密钥。为什么要在密钥生成中加入域名和用户名呢？许多用户会在不同的服务器上使用同样的密码。如果我只使用密码生成密钥，那么一个给定的密码在所有 Kerberos 服务器上总是会生成同样的密钥。因而，如果一个黑客可以取得用户在一台 Kerberos 服务器上的密钥，那么，他就可以在所有 Kerberos 服务器上使用同一个密钥。另一方面，如果加入了域名和用户名，那么一个受到这种攻击的密钥将只会侵害特定的域。

得到第 1 步中串接的字符串的字节数组表达。

统计第 2 步中字节数组中的字节数。在这个字节串的后面附加适当数量的零字节以使它成为 8 的整数倍。例如，如果这个字节数组包含 53 个字节，那么就在这个字节数组的最后附加三个字节使它具有 56 个字节。

将第 3 步中附加了字节后的字节数组分为大小相同的块，每一块有 8 个字节。

每隔一个块倒转块的位顺序。换句话说，第一块保持不变，第二块的位顺序应该倒转，第三块应保持不变，第中块的位顺序应倒转，以此类推。

取第一个（未改变的）块并与第二个（倒转的）块进行每一位的 exclusive OR。然后将第一次 exclusive OR 操作得到的结果与第三个（未改变的）块进行另一次 exclusive OR 操作。继续 exclusive OR 操作直到完成了所有块。所有 exclusive OR 操作的最后结果是一个 8 字节长的块。

修正在第 6 步中得到的 8 字节块的奇偶性。每一块的最低有效位保留为奇偶位。统计 8 字节块中每字节中的 1 的个数，如果 1 的个数为偶数，那么就设置最低位为 1 使它成为奇数。例如，如果一个字节的值为 00000000，那么就要将它改为 00000001。如果一个字节中 1 的个数已经为奇数，那么就将它的最低位设置为零。例如，如果一个字节为 00000010，那么就不需要为修正其奇偶性做任何改变。

DES 定义了一些弱的、因而不适合用于加密的密钥。我们的密钥生成过程的第八步是要检查奇偶修正后的字节数组是否是一个弱的密钥。如果是的话，就要用 0xf0（11110000）与奇偶修正过的 8 字节块进行 exclusive OR。如果奇偶修正得到的不是弱密钥，那么就不需要进行这种 exclusive OR 操作。经过这种弱密钥处理的字节数组是一个临时密钥。

现在我要使用这个临时密钥以 DES-CBC 算法加密第 3 步中得到的附加后的字节数组。这个临时密钥同时作为密钥的值和 DES-CBC 加密的初始矢量的值。回想在前面的讨论中说过，CBC 要求密文块链接。第 9 步的结果是最后 8 字节块的加密结果（放弃所以以前的密文块）。因此，这一步的结果是另一个 8 字节块。

现在我修正第 9 步产生的 8 字节块中的每一个字节的奇偶性。在上面第 7 步中我解释了奇偶性修正。

现在再次检查第 10 步得到的经过奇偶修正的 8 字节块是不是弱密钥（就像在第 8 步中所做的那样）。

分类: Active Directory

jankie
关注 - 3
粉丝 - 56
+加关注

0 0

(请您对文章做出评价)

« 上一篇：RHEL6上安装和部署LVS服务
» 下一篇：Apache与Tomcat的区别

posted @ 2011-08-22 12:47 jankie 阅读(11427) 评论(0) 编辑 收藏

注册用户登录后才能发表评论，请 登录 或 注册，访问网站首页。