

windows的认证

windows的认证.....	1
1. 认证基础.....	15
1.1. 概述.....	15
1.1.1.	
身份验证技术包括简单的账号密码，手机短信验证码，指纹，人脸识别等等。	
在windows域环境中，用户可能会在一个位置或多个位置访问多个服务器上的多个应用程序。鉴于这些原因，身份验证必须支持其他平台和其他 Windows 操作系统的环境。	
所以在域环境中，将加密密钥存储在安全的中心位置中可以确保身份验证过程可伸缩且可维护。在Windows域中默认使用Active Directory	
是用于存储标身份验证信息，其中包括用户的凭据的加密密钥。在 Active Directory 使用默认 NTLM 和 Kerberos 实现需要的认证。	16
1.1.1.1.	
我们可以理解为我们所有的账号密码都存在域控制服务器中，需要认证就问一下域控制器，“那个xxx是不是xxx”。域控制服务器就去查活动目录，然后跟我们说“xxx是不是xxx”.....	17
1.2. 身份验证和授权.....	17
1.2.1. 可传递信任	17
1.2.1.1. 定义	18
1.2.1.1.1.	
信任是什么，我们来可以这样理解：当我们在去国内任何地方旅游时，我们的身份证就是一种“信任“，用来证明我们的身份，所以我们使用身份证可以中国任意转。那么那么可传递信任是什么呢,我们都知道国内我们拿身份证就可以了，但是当我们出国的呢，我们得拿身份证去申请护照，那么身份证的“信任“就会传递到护照中，我们拿着护照就可以出国玩耍了。	18
1.2.1.2. 作用	18
1.2.1.2.1. 可传递信任是 Windows	
客户端/服务器体系结构中网络安全的基础。	
信任关系在一组域（如域树）中流动，并形成域与信任该域的所有域之间的关系。例如，如果域 A 具有域 B 的可传递信任，并且域 B 信任域	
C，则域 A 信任域 C。	19
1.2.2. 身份验证和授权之间的区别	19

1.2.2.1. 身份验证和授权之间存在差异。	
对于身份验证，系统会证明你是你。	
在授权下，系统会验证您是否有权执行您想要执行的操作。	19
1.3. 凭据	19
1.3.1. 在 Windows	
中，可以对凭据进行管理，使帐户持有者可以通过网络访问资源，而无需反复提供凭据。	
这种类型的访问权限允许用户一次对用户进行身份验证，以访问他们有权使用的所有应用程序和数据源，而无需输入其他帐户标识符或密码。 Windows	
平台通过在操作系统的本地安全机构 (LSA)	
中以本地方式缓存用户凭据，从而使使用单个用户标识 (通过网络 Active Directory) 维护的能力。当用户登录到域时， Windows	
身份验证包在对网络资源的凭据进行身份验证时，透明地使用凭据来提供单一登录。 20	
1.3.1.1. 账号密码就是一个凭据	21
1.4. Windows 身份验证中的凭据进程	21
1.4.1. 定义	21
1.4.1.1. Windows	
凭据管理是操作系统从服务或用户接收凭据的过程，并保护该信息以供将来呈现到身份验证目标。	
对于已加入域的计算机，身份验证目标为域控制器。	
身份验证中使用的凭据是将用户标识关联到某种形式的身份验证（如证书、密码或 PIN）的数字文档。	
默认情况下，将根据本地计算机上的安全帐户管理器 (SAM) 数据库或通过 Winlogon 服务在加入域的计算机上 Active Directory 验证 Windows 凭据。	
凭据是通过登录用户界面上的用户输入收集的，或者通过应用程序编程接口以编程方式收集的， (API) 提供给身份验证目标。	
本地安全信息存储在注册表中 HKEY_LOCAL_MACHINE \security 下。	
存储的信息包括策略设置、默认安全值和帐户信息，如缓存的登录凭据。	
SAM 数据库的副本也存储在此处，	22
1.4.2. 成功登录的路径	22
1.4.2.1. 如图	23
1.4.3. 进程	23
1.4.3.1.1. 用户登录 Winlogon.exe	
是负责管理安全用户交互的可执行文件。 Winlogon	
服务通过将用户操作在安全桌面 (登录)	
用户界面上收集的凭据传递给本地安全机构来启动 Windows	
操作系统的登录过程， (LSA) 到 Secur32.dll。	24

1.4.3.1.2.	应用程序登录 不需要交互式登录的应用程序或服务登录。 用户使用 Secur32.dll 在用户模式下运行的大多数进程，而在启动时启动的进程（如服务）使用 Ksecdd.sys 在内核模式下运行。	24
1.4.3.1.3.	Secur32.dll 构成身份验证过程基础的多身份验证提供程序。 ..	24
1.4.3.1.4.	Lsasrv.dll LSA 服务器服务，它强制实施安全策略，并充当 LSA 的安全包管理器。 LSA 包含 Negotiate 函数，该函数在确定要成功的协议后选择 NTLM 或 Kerberos 协议。	24
1.4.3.1.5.	安全支持提供程序 一组可单独调用一个或多个身份验证协议的提供程序。 默认的提供程序集可以随每个版本的 Windows 操作系统一起更改，并可以编写自定义的提供程序。	24
1.4.3.1.6.	Netlogon.dll Net Logon 服务执行的服务如下所示： - 维护计算机的安全通道 (不会与域控制器的 Schannel) 混淆。 - 通过安全通道将用户的凭据传递到域控制器，并为用户返回域安全标识符 (Sid) 和用户权限。 -在域名系统 (DNS) 中发布服务资源记录，并使用 DNS 将名称解析为域控制器 (IP) 地址的 Internet 协议。 -根据远程过程调用 (RPC) 实现复制协议，以便同步主域控制器 (Pdc) 和备份域控制器 (Bdc) 。 25	
1.4.3.1.7.	Samsrv.dll 安全帐户管理器 (SAM) ，它存储本地安全帐户，强制实施本地存储的策略并支持 Api 。	25
1.4.3.1.8.	注册表 注册表包含 SAM 数据库的副本、本地安全策略设置、默认安全值和仅可用于系统的帐户信息。 25	
1.5.	安全主体和帐户	25
1.5.1.	在 Windows 中，可以启动操作的任何用户、服务、组或计算机都是安全主体。 安全主体具有可在计算机本地或基于域的帐户。 例如，即使没有人为用户登录， Windows 客户端加入域的计算机也可以通过与域控制器通信来加入网络域。 若要启动通信，计算机必须在域中具有活动帐户。 在接受来自计算机的通信之前，域控制器上的本地安全机构会对计算机的标识进行身份验证，然后定义计算机的安全上下文，就像对安全主体而言。 此安全上下文定义特定计算机或网络上的用户、服务、组或计算机上的用户或服务的标识和功能。 例如，它定义了可以访问的资源（例如文件共享或打印机）以及可以由用户、服务或该资源上的计算机执行的操作，如读取、写入或修改。	26
1.6.	委托身份验证	26

1.6.1. 在 Windows

中，当网络服务接受来自用户的身份验证请求并假定该用户的标识以便启动到第二个网络服务的新连接时，将进行委派的身份验证。

若要支持委派的身份验证，必须建立前端或第一层服务器（如 web 服务器），这些服务器负责处理客户端身份验证请求以及用于存储信息的后端或 n 层服务器（如大型数据库）。

你可以委派权限，以便为组织中的用户设置委派的身份验证27

1.6.2.

通过将服务或计算机建立为受信任的委派，可让服务或计算机完成委托身份验证，接收发出请求的用户的票证，然后访问该用户的信息。此模型仅将后端服务器上的数据访问限制到那些使用正确的访问控制令牌提供凭据的用户或服务。此外，它还允许对这些后端资源进行访问审核。通过要求通过委托给服务器的凭据访问所有数据以代表客户端使用，你可以确保服务器不会泄露，并且你可以获取对存储在其他服务器上的敏感信息的访问权限。

对于设计为在多台计算机上使用单一登录功能的多层应用程序，委派身份验证非常有用。27

1.7. 域之间的信任关系中的身份验证27

1.7.1.

拥有多个域的大多数组织都有权访问位于不同域中的共享资源控制此访问要求一个域中的用户也可以通过身份验证和授权使用另一个域中的资源。若要在不同域中的客户端和服务端之间提供身份验证和授权功能，两个域之间必须存在信任。信任是 Active Directory 通信的基本技术，是 Windows Server 网络体系结构的一个有机安全组件。

如果两个域之间存在信任，则每个域的身份验证机制将信任来自其他域的身份验证。

信任帮助提供对资源域中共享资源的受控访问权限（信任域），验证传入的身份验证请求是否来自受信任的颁发机构--受信任的域。

通过这种方式，信任充当桥，只允许验证的身份验证请求在域之间传输。

特定信任通过身份验证请求的方式取决于其配置方式。

信任关系可以是单向的，即提供从受信任域到信任域中资源的访问权限，或通过提供从每个域到其他域中资源的访问权限来实现双向关系。

信任也是不可传递的，在这种情况下，信任关系仅存在于两个信任伙伴域之间，或者是可传递的，在这种情况下，信任会自动扩展到合作伙伴信任的任何其他域。 28

1.8. 协议转换29

1.8.1.

协议转换通过允许应用程序在用户身份验证层支持不同的身份验证机制，并通过在后续应用程序层切换到 Kerberos

协议（如相互身份验证和约束委派）来帮助应用程序设计人员。29

1.9.	约束委派	29
1.9.1.		
	约束委派使管理员能够通过限制应用程序服务可以代表用户操作的范围来指定和强制实施应用程序信任边界。	
	你可以指定特定服务，通过这些服务，可委托的计算机可以请求资源。	
	限制服务的授权权限的灵活性有助于通过减少不受信任的服务的泄露机会来改善应用程序安全设计。	29
2.	认证分类	29
2.1.		
	windows的认证方式主要有NTLM认证、kerberos认证两种。同时，windows Access Token记录着用户的SID、组ID、Session、及权限等信息，也起到了权限认证的作用。 30	
2.1.1.	所以一般我们可以说windows的认证有3种	30
3.	NTLM认证	30
3.1.	描述	30
3.1.1.		
	随着更安全的身份验证协议（例如Kerberos协议）的出现，业界对在其环境中更好地管理NTLM协议的能力的要求不断提高。减少IT环境中NTLM协议的使用，既需要了解NTLM上已部署的应用程序要求，也需要配置计算环境以使用其他协议所需的策略和步骤	31
3.1.1.1.	微软官方看到的，但其实国内还是使用的	31
3.2.	认证描述	31
3.2.1.		
	NTLM认证主要有本地认证和网络认证两种方式。本地登陆时用户密码存储在SAM文件中，可以把它当作一个存储密码的数据库，所有的操作都在本地进行的。它将用户输入的密码转换为NTLM Hash,然后与SAM中的NTLM Hash进行比较。而网络认证则是基于一种Challenge/Response认证机制的认证模式。 32	
3.3.	本地认证	32
3.3.1.	描述	32
3.3.1.1.		
	在本地登陆的情况下，操作系统会使用用户输入的密码作为凭据去与系统中的密码进行校验，如果成功的话表明验证通过。操作系统的密码存储在C盘的目录下： %SystemRoot%\system32\config\sam SAM用于储存本地所有用户的凭证信息，但是这并不代表着你可以随意去查看系统密码。我们登陆系统的时候系统会自动读取SAM文件中的密码与我们输入的密码进行比对，如果认证相同则可以使用该机器。 windows自身是不会保存明文密码的，也就是说SAM中保存的不是明文而是	

Hash。

Hash，一般翻译做散列、杂凑，或音译为哈希，是把任意长度的输入（又叫做预映射pre-image）通过散列算法变换成固定长度的输出，该输出就是散列值。这种转换是一种压缩映射，也就是，散列值的空间通常远小于输入的空间，不同的输入可能会散列成相同的输出，所以不可能从散列值来确定唯一的输入值。简单的说就是一种将任意长度的消息压缩到某一固定长度的消息摘要的函数。

33

3.3.2. NTLM Hash生成.....34

3.3.2.1. NTLM

Hash是怎么样生成的呢？当用户注销、重启、锁屏后，操作系统会让winlogon显示登陆界面，当winlogon.exe接收到账号密码输入之后，会将密码交给lsass进程，这个进程会存一份明文密码，将明文密码加密成NTLM

Hash，对SAM数据库比较认证。(winlogon.exe即Windows Logon Process，是Windows

NT用户登陆程序，用于管理用户登录和退出。LSASS用于微软Windows系统的安全机制。它用于本地安全和登陆策略。)

比如当用户输入密码admin的时候，操作系统会将admin转换为16进制，经过Unicode转换后，再调用MD4加密算法加密，这个加密结果的十六进制就是NTLM Hash.....34

3.3.3. 本地认证的流程34

3.3.3.1. winlogon.exe -> 接收用户输入 -> lsass.exe -> (认证) Windows Logon Process(即 winlogon.exe)，是Windows NT 用户登陆程序，用于管理用户登录和退出。LSASS用于微软Windows系统的安全机制。它用于本地安全和登陆策略35

3.3.4. NTLM前身LM Hash.....35

3.3.4.1.

LM与NTLM协议的认证机制相同，但是加密算法不同。目前大多数的windows系统都采用了NTLM协议认证，LM协议现在基本已经淘汰了。LM协议认证过程中需要LM Hash作为根本凭证进行参与认证。LM

Hash产生原理：35

3.4. 网络认证35

3.4.1. 基础35

3.4.1.1.

在工作组中，无论是局域网中的一台机器还是很多机器，它们能够通信的话都无法相互建立一个完美的信任机制。只要有一个可以信任的信托机构，对两方进行认证，这样就有第三方来证实双方的可信任性。36

3.4.2. SMB协议36

3.4.2.1.	在了解认证之前先了解一些SMB协议：SMB（ServerMessage Block）通信协议是微软（Microsoft）和英特尔(Intel)在1987年制定的协议，主要是作为Microsoft网络的通讯协议。SMB是在会话层（session layer）和表示层（presentation layer）以及小部分应用层（application layer）的协议。SMB使用了NetBIOS的应用程序接口（Application Program Interface，简称API），一般端口使用为139，445。另外，它是一个开放性的协议，允许了协议扩展——使得它变得更大而且复杂；大约有65个最上层的作业，而每个作业都超过120个函数，甚至Windows NT也没有全部支持到，最近微软又把SMB改名为CIFS（CommonInternet File System），并且加入了许多新的特色。早期SMB协议在网络上传输的是明文口令。后来出现LAN Manager ChallengeReponse验证机制，简称LM。微软提出了windowsNT挑战/响应验证机制，简称MTLM。现在已经更新到了V2版本以及加入了Kerberos验证体系	36
3.4.3.	NTLM 协议	37
3.4.3.1.	NTLM是一种网络认证协议，它是基于挑战（Challenge）/响应（Response）认证机制的一种认证模式。(这个协议只支持Windows) NTLM协议的认证共需要三个消息完成：协商 --> 挑战 --> 认证。 协商：主要用于确认双方协议版本、加密等级等 挑战：服务器在收到客户端的协商消息之后，会读取其中的内容，并从中选择出自己所能接受的服务内容，加密等级，安全服务等等。并生成一个随机数challenge, 然后生成challenge消息返回给客户端。该消息就是挑战/响应认证机制的主要功能体现。 认证：验证主要是在挑战完成后，验证结果，是认证的最后一步。	37
3.4.4.	认证流程	37
3.4.4.1.	首先客户端向服务器发送一些用户信息以及主机信息（包含用户名，如果没有这个用户的话认证就失败），服务器接收到请求之后会生成一个16为的随机数称之为"Challenge"(挑战)，使用登陆用户名对应的NTLM Hash加密Challen(其实就是验证它是否在我本地认证的数据库中)生成Challenge1，同时将Challenge发送给客户端。 Net NTLM Hash = NTLM Hash(Challenge) Challenge是服务端随机生成的、NTLM Hash是服务器根据客户端提供的用户名寻找出来的Hash加密生成Net NTLM Hash(这个是在服务端这边) 客户端接收到Challenge后使用将要登陆到账户对应的NTLM Hash加密Challenge生成Response之后发给服务端，服务器接收到客户端发过	

来的Response后，对比Challenge1与Response是否相等，从而进行验证。

38

3.4.5. 流程解读38

3.4.5.1. 过程： 第一步：输入密码,然后LSASS会把密码的NTLM

Hash后的值先存储到本地。 第二步：客户端把用户名的明文发送给服务端

第三步：服务端接收到用户名之后会判断用户名是否存在，不存在则代表认证失败。存在的话服务端会生成一个16位的随机数,并且从本地查找share_user对应的NTLM Hash，使用NTLM Hash加密Challenge，生成一个Net-NTLM Hash存在内存中，并将Challenge发送给客户端。

第四步：当客户端收到challenge后,用在第一步中存储的NTLM

Hash对其加密，然后再将加密后的challenge发送给服务器，也就是response，表现形式是Net-NTLM Hash。

第五步：服务端在收到response后，会向DC发送针对客户端的验证请求。该请求主要包含以下三方面的内容：客户端用户名、客户端NTLM

Hash加密的Challenge、原始的Challenge。

第六步：当DC接到过来的这三个值的以后,会根据用户名到DC的账号数据库(ntds.dit)里面找到该用户名对应的NTLM

Hash,然后把这个hash拿出来和传过来的challenge值进行比较,相同则认证成功,反之,则失败。38

3.5. NTLM 协议 V1 与 V2的区别39

3.5.1. NTLM 协议 V1 与 V2的区别 NTLM v1与NTLM

v2最显著的区别就是Challenge与加密算法不同，共同点就是加密的原料都是NTLM Hash。 Challenge:NTLM v1的Challenge有8位，NTLM v2的Challenge为16位。

Net-NTLM Hash:NTLM v1的主要加密算法是DES，NTLM

v2的主要加密算法是HMAC-MD5。39

4. Kerberos认证39

4.1. 定义40

4.1.1. Windows Server 操作系统可实现 Kerberos 版本 5

身份验证协议和对公钥身份验证的扩展，用于传输授权数据和委派。Kerberos身份验证客户端作为安全支持提供程序 (SSP 实现)

，并且可通过安全支持提供程序接口 SSPI 进行访问()。初始用户身份验证与Winlogon 单一登录 - 体系结构集成。Kerberos 密钥发行中心 (KDC)

与域控制器上运行的其他 Windows Server 安全服务相集成。KDC 使用域的

Active Directory 域服务数据库作为其安全帐户数据库。Active Directory

域服务是域或林中的默认 Kerberos 实现所必需的41

4.2. 作用41

4.2.1. 委托身份验证。42

4.2.1.1.	在代表客户端访问资源时，在 Windows 操作系统上运行的服务可以模拟客户端计算机。通常，服务通过访问本地计算机上的资源为客户端完成工作。当客户端计算机向服务进行身份验证时，NTLM 和 Kerberos 协议都可以提供服务在本地模拟客户端计算机所需的授权信息。但是，某些分布式应用程序的设计使前端 - 服务在连接到其他计算机上的后端服务时，必须使用客户端计算机的标识 - 。 Kerberos 身份验证支持一种委派机制，使服务在连接到其他服务时可以代表其客户端进行操作。	42
4.2.2.	单一登录。	42
4.2.2.1.	在域或林中使用 Kerberos 身份验证将允许用户或服务访问管理员允许访问的资源，而无需多次请求凭据。在通过 Winlogon 第一次登录域之后，Kerberos 将在每次尝试访问资源时管理整个林中的凭据。	43
4.2.3.	对服务器更高效的身份验证。	43
4.2.3.1.	在 Kerberos 出现之前，可以使用 NTLM 身份验证，它要求应用程序服务器必须连接到域控制器，以便验证每个客户端计算机或服务的身份。使用 Kerberos 协议时，可续订会话票证会替代 pass - 通过身份验证。服务器不需要使用域控制器，(除非它需要验证特权属性证书 (PAC)) 。服务器可以通过检查客户端出示的凭据来验证客户端计算机的身份。客户端计算机在获得一次特定服务器的凭据后，即可在整个网络登录会话期间重复使用这些凭据。	43
4.2.4.	相互身份验证。	43
4.2.4.1.	通过使用 Kerberos 协议，网络连接两端的每一方可验证另一方所宣称的身份。 NTLM 不允许客户端验证服务器的身份，也不允许一个服务器验证另一个服务器的身份。 NTLM 身份验证旨在用于服务器假定为真的网络环境。 Kerberos 协议不进行此假设。	44
4.3.	Kerberos 约束委派	44
4.3.1.	定义	44
4.3.1.1.	在 Windows Server 2003 中引入的 Kerberos 约束委派，为服务所使用的委派提供了一种更安全的形式。配置了 Kerberos 约束委派后，它限制了指定服务器可代表用户执行的服务。这需要域管理员权限来为服务配置一个域账户，并把该账户限制到单个域。在如今的企业中，前端服务的设计并不局限于仅与域中的服务进行集成。在域管理员配置了服务的早期操作系统中，服务管理员没有有效途径来了解哪些前端服务委派给了其拥有的资源服务。	

并且可委派给资源服务的任何前端服务都代表了一个潜在的攻击点。 如果托管前端服务的服务器受到安全威胁，并且它已配置为委派给资源服务，则资源服务也会受到安全威胁。在 Windows Server 2012 R2 和 Windows Server 2012 中，为服务配置约束委派的能力已从域管理员转移给服务管理员。 这样，后端服务管理员可以允许或拒绝前端服务。	44
4.3.2. 应用	45
4.3.2.1. 约束委派让服务管理员能够通过限制应用程序服务可以代表用户的范围来指定和强制应用程序信任边界。 服务管理员可以配置哪些前端服务账户能委派到其后端服务。通过支持 Windows Server 2012 R2 和 Windows Server 2012 中跨域的约束委派，可以将前端服务（例如 Microsoft Internet 安全和加速 (ISA) Server、Microsoft Forefront 威胁管理网关、Microsoft Exchange Outlook Web 访问 (OWA) 和 Microsoft SharePoint Server）配置为使用约束委派对其他域中的服务器进行身份验证。 这将通过使用现有的 Kerberos 基础结构来支持跨域的服务解决方案。 域管理员或服务管理员可以管理 Kerberos 约束委派。	45
4.3.3. 跨域的基于资源的约束委派	45
4.3.3.1. 描述	46
4.3.3.1.1. Kerberos 约束委派可以在前端服务与资源服务不在同一域中时用于提供约束委派。服务管理员能通过指定可以在资源服务账户对象上代表用户的前端服务域账户来配置新委派。	46
4.3.3.1.2. 作用	46
4.3.3.1.2.1. 通过支持跨域约束委派，可以将服务配置为使用约束委派（而不是非约束委派）来对其他域中的服务器进行身份验证。 这将通过使用现有的 Kerberos 基础结构来支持跨域的服务解决方案身份验证，而不需要信任委派到任何服务的前端服务。 这还决定了服务器是否应该信任委派的标识的源，从委派的域管理员到资源所有者。	46
4.3.3.1.3. 基于资源的约束委派的安全含义	46
4.3.3.1.3.1. 基于资源的约束委派会将委派控制置于拥有所访问资源的 管理员。它依赖于资源服务的属性，而不是受信任的服务委托。 因此，基于资源的约束委派不能使用以前控制的协议转换的受信任身份验证委托位。当执行基于资源的约束委派时，KDC 始终允许协议转换，就像设置了位一样。因为 KDC	

不会限制协议转换，所以引入了两个新的已知 Sid，以将此控件授予资源管理员。这些 Sid 确定是否发生了协议转换，并可与标准访问控制列表结合使用来根据需 要授予或限制访问权限。	47
4.4. Kerberos域认证解读.....	47
4.4.1. 定义	47
4.4.1.1. Kerberos 是一种网络认证协议，其设计目标是通过密钥系统为客户 机 / 服务器应用程序提供强大的认证服务。该认证过程的实现不 依赖于主机操作系统的认证，无需基于主机地址的信任，不要求 网络上所有主机的物理安全，并假定网络上传送的数据包可以被 任意地读取、修改和插入数据。在以上情况下， Kerberos 作为一 种可信任的第三方认证服务，是通过传统的密码技术(如:共享 密钥)执行认证服务的。	48
4.4.2. 作用	48
4.4.2.1. 在域中，网络对象可以相互访问，但是在真实情况中，需要对某 些部门的计算机进行限制，例如：销售部门不能访问技术部门的服务器。 这个中间就需要Kerberos认证协议来验证网络对象间的权限。	49
4.4.3. 基础	49
4.4.3.1. Kerberos的标志是三只狗头，狗头分别代表以下角色： Client Server KDC(Key Distribution Center) = DC(Domain Controller) kerberos使用了一个包含客户端、应用服务器和一个kerberos服务器的协议 ，这个协议的设计就是对抗客户端/服务器对话安全的多种威胁。在一个不 受保护的网路中，任何一个客户端可以使用任意一台服务器提供的服务。很 明显的安全威胁就是伪装，对方可以扮演另一个客户端并在服务器上获取没 有经过验证的权限！所以服务器必须能确认请求服务的客户端的身份进行验 证。为了避免给服务器更多的访问压力和每次和客户端交互的风险，使用认 证服务器(AS),它存储了所有用户的口令并集中在一个数据库中，然后用户就 可以登陆AS进行验证身份，如果验证通过的话它就可以把信息传达到一个 应用服务器。	49
4.4.3.2. 名词基本概念： KDC: Key Distribution Center，密钥分发中心，负责管理票据、认证票据、分发票据，但是KDC不 是一个独立的服务，它由AS和TGS组成。 AS: Authentication Service，验证服务，为client生成TGT的服务 TGS: Ticket Granting Service，票据授予服务，为client生成某个服务的ticket TGT: Ticket Granting Ticket，入场券，通过入场券能够获得票据，是一种临时凭证的存在。 Ticket:票据，是网络中各对象之间互相访问的凭证 AD: Account Database，存储所有client的白名单，只有存在于白名单的client才能顺利申 请到TGT。 DC: Domain Controller，域控 KRBtgt:	

每个域控制器都有一个krbtgt账户，是KDC的服务账户，用来创建TGS加密的密钥。 50

4.4.4. 域认证流程:50

4.4.4.1. client向kerberos服务请求，希望获取访问server的权限。

kerberos得到了这个消息，首先得判断client是否是可信赖的，也就是白名单黑名单的说法。这就是AS服务完成的工作，通过在AD中存储黑名单和白名单来区分client。成功后，返回AS返回TGT给client。client得到了TGT后，继续向kerberos请求，希望获取访问server的权限。kerberos又得到了这个消息，这时候通过client消息中的TGT，判断出了client拥有了这个权限，给了client访问server的权限ticket。

client得到ticket后，终于可以成功访问server。这个ticket只是针对这个server，其他server需要向TGS申请。51

4.4.4.1.1.

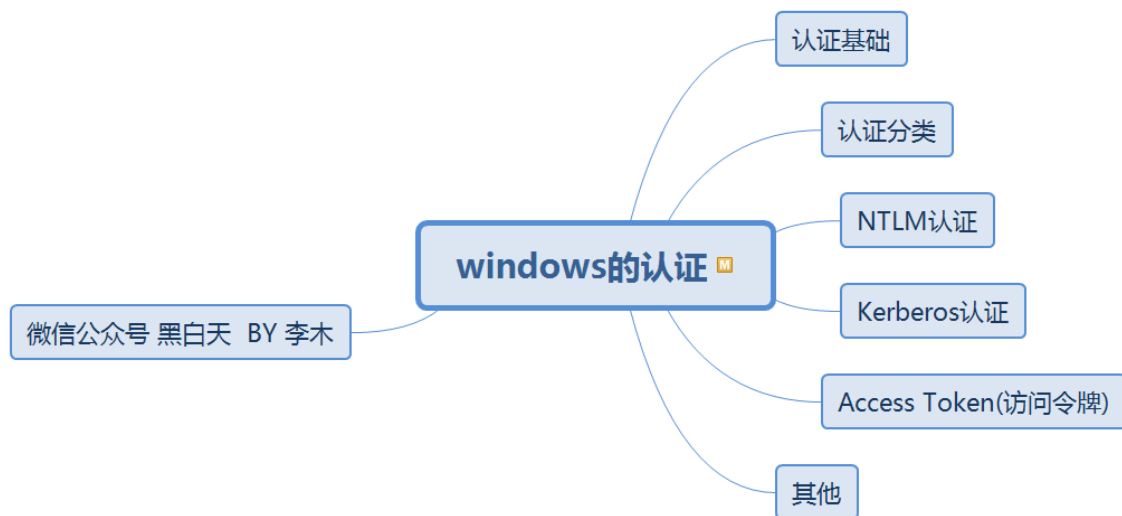
首先用户登陆到一个工作站并请求访问一个特定的服务器，客户端把一个包含用户ID和被称为TGT(Ticket-Granting Ticket,票据授予票据，也可也称为入场券)请求的消息发送到AS。其中，TGT的到期时间为8小时，如果超过了8小时，还需要重新申请TGT，不能之间进入下一步获取Ticket；AS在它的数据库中查找用户的口令，然后AS回复一个TGT和一个称为会话密钥的一次性加密密钥(可以称之为Session Key)给客户端。这两个加密都是使用用户口令作为加密密钥。然后发送给客户端，这个时候会提示客户端输入口令，产生密钥，并且解开发来的信息，如果提供了正确的口令，票据(ticket)和会话密钥就会被恢复。票据组成了一个客户端用来请求服务的信任证书的集合，票据显示AS已经接收了这个客户端和用户。票据包含了用户ID、一个时间戳、票据的失效时间。整个票据使用AS和服务共享的DES密钥加密。这个时候客户端会向AS发送TGT和解密的Session Key。Session Key用于客户端向TGS服务通信。域内所有网络对象的凭证都在AD中保存 KDC中某个用户指的是krbtgt52

4.4.4.1.2.

这个时候Kerberos与客户端已经建立起来了，客户端需要提供TGT与第一步中使用自己NTLM Hash解密出来的Session Key加密的客户端信息跟时间戳；如果假设这个数据被中间人窃取到，也无法在段时间内破解，因为KDC会校验时间戳。KDC接到TGT与其他内容后，会首先解密TGT，只有KDC可以解密TGT，从TGT中提取到Session Key，再使用Session Key解密其他内容，解密出来的内容同TGT中的信息进行校验来确认客户端是否受信；验证通过后，就会生成一个新的Session Key，我们称之为Server Session Key，这个Server Session

Key主要用于和服务器进行通信。同时还会生成一个Ticket，也就是最后的票据了。 53	
4.4.4.1.3. 第三步里，客户端向服务器请求，需要提供Ticket，Server Session Key加密的客户端信息与时间戳。 Ticket客户端无法解密服务器端通过解密Ticket解密Server Session Key(Client info + Timestamp)比较时间长度	
校验通过后，认证成功，该票据会一直存在客户端内存中。53	53
5. Access Token(访问令牌)	54
5.1. 基础	54
5.1.1. Windows Token其实叫Access Token(访问令牌)，它是一个描述进程或者线程安全上下文的一个对象。不同的用户登录计算机后，都会生成一个Access Token，这个Token在用户创建进程或者线程时会被使用，不断的拷贝，这也就解释了A用户创建一个进程而该进程没有B用户的权限。令牌就是系统的临时密钥，相当于用户名和密码，用来决定是否允许这次请求和判读这次请求属于那个用户，它允许你不提供凭证的前提下访问网络和系统资源。 Windows Access Token(访问令牌)有两种，一种是Delegation token(授权令牌)，主要用于交互会话登录(例如本地用户直接登录、远程桌面登录)，另一种是Impersonation token(模拟令牌)，主要用于非交互登录(利用net use访问共享文件夹)	54
5.2. Access Token种类.....	54
5.2.1. 主令牌	55
5.2.2. 模拟令牌	55
5.3. Windows Access Token组成.....	55
5.3.1. 用户帐户的安全标识符(SID) 用户所属的组的SID 用于标识当前登录会话的登录SID 用户或用户组所拥有的权限列表 所有者SID 主要组的SID 访问控制列表 访问令牌的来源 令牌是主要令牌还是模拟令牌 限制SID的可选列表 目前的模拟等级 其他统计数据	55
5.4. Windows Access Token SID (安全标识符).....	56
5.4.1. 安全标识符是一个唯一的字符串，它可以代表一个账户、一个用户组、或者是一次登录。通常它还有一个SID固定列表，例如 Everyone这种已经内置的账户，默认拥有固定的SID。 SID的表现形式: 域SID-用户ID 计算机SID-用户ID SID列表都会存储在域控的AD或者计算机本地账户数据库中。56	56
5.5. Windows Access Token产生的过程.....	56
5.5.1. 每个进程创建时都会根据登录会话权限由LSA(Local Security Authority)分配一个Token。如果CreaetProcess时自己指定了 Token, LSA会用该Token,	

否则就用父进程Token的一份拷贝。当用户注销后，系统将会使授权令牌切换为模拟令牌，不会将令牌清除，只有在重启机器后才会清除。	56
6. 其他	57
6.1. 内网渗透常用端口	57
6.1.1. 53 DNS服务，在使用中需要用到TCP/UDP	
53端口，AD域的核心就是DNS服务器，AD通过DNS服务器定位资源	88
Kerberos服务，在使用中需要用到TCP/UDP	
88端口，Kerberos密钥分发中心(KDC) 在该端口上侦听Ticket请求	135
135端口主要用于使用RPC协议并提供DCOM服务。	137 NetBIOS-
NS(名称服务)，在使用中需要用到TCP/UDP	137端口
139 Session Server(会话服务),在使用中需要用到TCP/UDP	
139端口，允许两台计算机建立连接	389
LDAP服务(轻量级目录访问协议)，在使用中需要用到TCP/UDP	
389端口，如果需要使用SSL，需要使用636端口，	445
主要用于共享文件夹或共享打印，存在较多漏洞，如MS08-067、MS17-010	
3268 Global Catalog(全局编录服务器)，如果需要使用SSL，需要用到3269端口，主要用于	
用户登录时，负责验证用户身份的域控制器需要通过防火	
墙，来向“全局编录”查询用户所隶属的通用组	57
6.2. https://payloads.online/archivers/2018-11-30/1#0x03-windows-access-token	
https://www.cnblogs.com/artech/archive/2011/01/24/kerberos.html	
https://www.cnblogs.com/artech/archive/2011/01/25/NTLM.html	58
6.3.	58
7. 微信公众号 黑白天 BY 李木.....	58



1. 认证基础



1.1. 概述

概述

身份验证技术包括简单的账号密码，手机短信验证码，指纹，人脸识别等等。在windows域环境中，用户可能会在一个位置或多个位置访问多个服务器上的多个应用程序。鉴于这些原因，身份验证必须支持其他平台和其他 Windows 操作系统的环境。

所以在域环境中，将加密密钥存储在安全的中心位置中可以确保身份验证过程可伸缩且可维护。在Windows域中默认使用Active Directory 是用于存储标身份验证信息，其中包括用户的凭据的加密密钥。在 Active Directory 使用默认 NTLM 和 Kerberos 实现需要的认证。

1.1.1.

身份验证技术包括简单的账号密码，手机短信验证码，指纹，人脸识别等等。在windows域环境中，用户可能会在一个位置或多个位置访问多个服务器上的多个应用程序。鉴于这些原因，身份验证必须支持其他平台和其他 Windows 操作系统的环境。

所以在域环境中，将加密密钥存储在安全的中心位置中可以确保身份验证过程可伸缩且可维护。在Windows域中默认使用Active Directory 是用于存储标身份验证信息，其中包括用户的凭据的加密密钥。在 Active Directory 使用默认 NTLM 和 Kerberos 实现需要的认证。

身份验证技术包括简单的账号密码，手机短信验证码，指纹，人脸识别等等。在windows域环境中，用户可能会在一个位置或多个位置访问多个服务器上的多个应用程序。鉴于这些原因，身份验证必须支持其他平台和其他 Windows 操作系统的环境。

所以在域环境中，将加密密钥存储在安全的中心位置中可以确保身份验证过程可伸缩且可维护。在Windows域中默认使用Active Directory 是用于存储标身份验证信息，其中包括用户的凭据的加密密钥。在 Active Directory 使用默认 NTLM 和 Kerberos 实现需要的认证。

我们可以理解为我们所有的账号密码都存在域控制服务器中，需要认证就问一下域控制器，“那个xxx是不是xxx”。域控制服务器就去查活动目录，然后跟我们说“xxx是不是xxx”

1.1.1.1. 我们可以理解为我们所有的账号密码都存在域控制服务器中，需要认证就问一下域控制器，“那个xxx是不是xxx”。域控制服务器就去查活动目录，然后跟我们说“xxx是不是xxx”

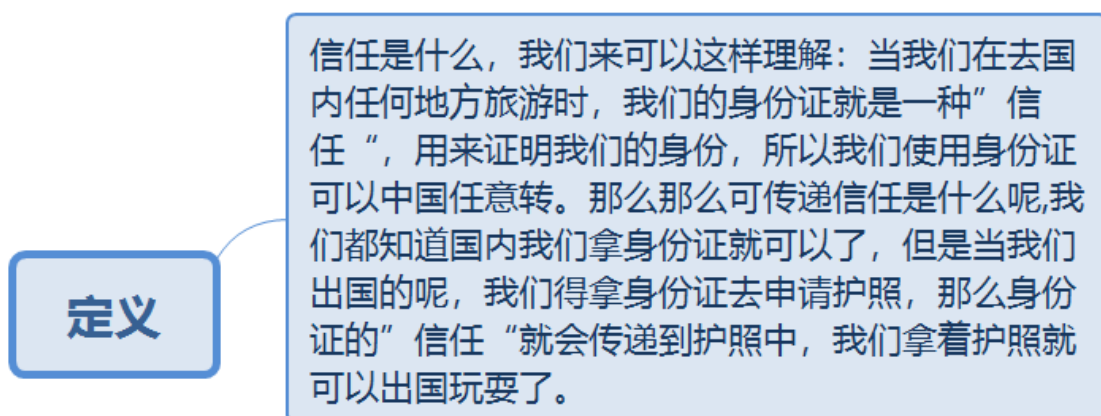
1.2. 身份验证和授权



1.2.1. 可传递信任



1.2.1.1. 定义



1.2.1.1.1. 信任是什么，我们来可以这样理解：当我们在去国内任何地方旅游时，我们的身份证就是一种“信任”，用来证明我们的身份，所以我们使用身份证可以中国任意转。那么那么可传递信任是什么呢，我们都知道国内我们拿身份证就可以了，但是当我们出国的呢，我们得拿身份证去申请护照，那么身份证的“信任”就会传递到护照中，我们拿着护照就可以出国玩耍了。

1.2.1.2. 作用

作用

可传递信任是 Windows 客户端/服务器体系结构中网络安全的基础。信任关系在一组域（如域树）中流动，并形成域与信任该域的所有域之间的关系。例如，如果域 A 具有域 B 的可传递信任，并且域 B 信任域 C，则域 A 信任域 C。

1.2.1.2.1. 可传递信任是 Windows

客户端/服务器体系结构中网络安全的基础。

信任关系在一组域（如域树）中流动，并形成域与信任该域的所有域之间的关系。例如，如果域 A 具有域 B 的可传递信任，并且域 B 信任域 C，则域 A 信任域 C。

1.2.2. 身份验证和授权之间的区别

身份验证和授权之间的区别

身份验证和授权之间存在差异。对于身份验证，系统会证明你是你。在授权下，系统会验证您是否有权执行您想要执行的操作。

1.2.2.1. 身份验证和授权之间存在差异。

对于身份验证，系统会证明你是你。

在授权下，系统会验证您是否有权执行您想要执行的操作。

1.3. 凭据

凭据

在 Windows 中，可以对凭据进行管理，使帐户持有者可以通过网络访问资源，而无需反复提供凭据。这种类型的访问权限允许用户一次对用户进行身份验证，以访问他们有权使用的所有应用程序和数据源，而无需输入其他帐户标识符或密码。Windows 平台通过在操作系统的本地安全机构 (LSA) 中以本地方式缓存用户凭据，从而使使用单个用户标识 (通过网络 Active Directory) 维护的能力。当用户登录到域时，Windows 身份验证包在对网络资源的凭据进行身份验证时，透明地使用凭据来提供单一登录。

1.3.1. 在 Windows

中，可以对凭据进行管理，使帐户持有者可以通过网络访问资源，而无需反复提供凭据。

这种类型的访问权限允许用户一次对用户进行身份验证，以访问他们有权使用的所有应用程序和数据源，而无需输入其他帐户标识符或密码。Windows 平台通过在操作系统的本地安全机构 (LSA) 中以本地方式缓存用户凭据，从而使使用单个用户标识 (通过网络 Active Directory) 维护的能力。当用户登录到域时，Windows 身份验证包在对网络资源的凭据进行身份验证时，透明地使用凭据来提供单一登录。

在 Windows 中，可以对凭据进行管理，使帐户持有者可以通过网络访问资源，而无需反复提供凭据。这种类型的访问权限允许用户一次对用户进行身份验证，以访问他们有权使用的所有应用程序和数据源，而无需输入其他帐户标识符或密码。Windows 平台通过在操作系统的本地安全机构 (LSA) 中以本地方式缓存用户凭据，从而使使用单个用户标识 (通过网络 Active Directory) 维护的能力。当用户登录到域时，Windows 身份验证包在对网络资源的凭据进行身份验证时，透明地使用凭据来提供单一登录。

账号密码就是一个凭据

1.3.1.1. 账号密码就是一个凭据

1.4. Windows 身份验证中的凭据进程



1.4.1. 定义

定义

Windows 凭据管理是操作系统从服务或用户接收凭据的过程，并保护该信息以供将来呈现到身份验证目标。对于已加入域的计算机，身份验证目标为域控制器。身份验证中使用的凭据是将用户标识关联到某种形式的身份验证（如证书、密码或 PIN）的数字文档。

默认情况下，将根据本地计算机上的安全帐户管理器 (SAM) 数据库或通过 Winlogon 服务在加入域的计算机上 Active Directory 验证 Windows 凭据。凭据是通过登录用户界面上的用户输入收集的，或者通过应用程序编程接口以编程方式收集的，(API) 提供给身份验证目标。

本地安全信息存储在注册表中

HKEY_LOCAL_MACHINE \security 下。存储的信息包括策略设置、默认安全值和帐户信息，如缓存的登录凭据。SAM 数据库的副本也存储在此处，

1.4.1.1. Windows

凭据管理是操作系统从服务或用户接收凭据的过程，并保护该信息以供将来呈现到身份验证目标。对于已加入域的计算机，身份验证目标为域控制器。身份验证中使用的凭据是将用户标识关联到某种形式的身份验证（如证书、密码或 PIN）的数字文档。

默认情况下，将根据本地计算机上的安全帐户管理器 (SAM) 数据库或通过 Winlogon 服务在加入域的计算机上 Active Directory 验证 Windows 凭据。

凭据是通过登录用户界面上的用户输入收集的，或者通过应用程序编程接口以编程方式收集的，(API) 提供给身份验证目标。

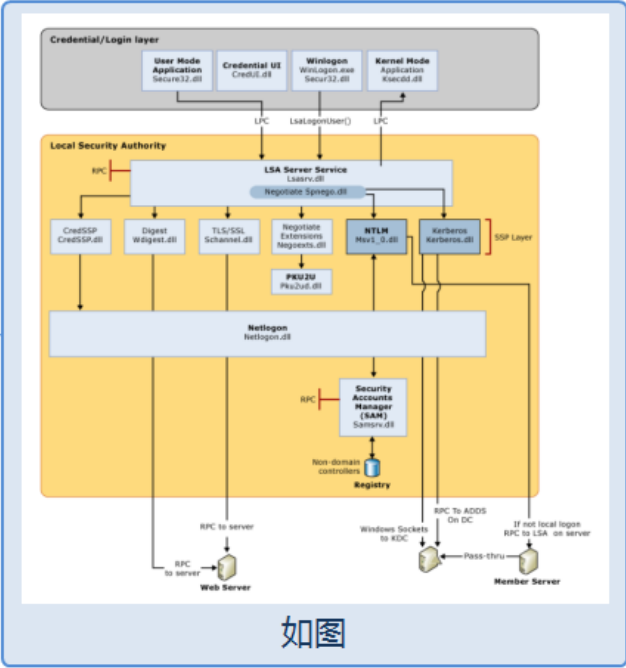
本地安全信息存储在注册表中 HKEY_LOCAL_MACHINE \security 下。

存储的信息包括策略设置、默认安全值和帐户信息，如缓存的登录凭据。

SAM 数据库的副本也存储在此处，

1.4.2. 成功登录的路径

成功登录的路径



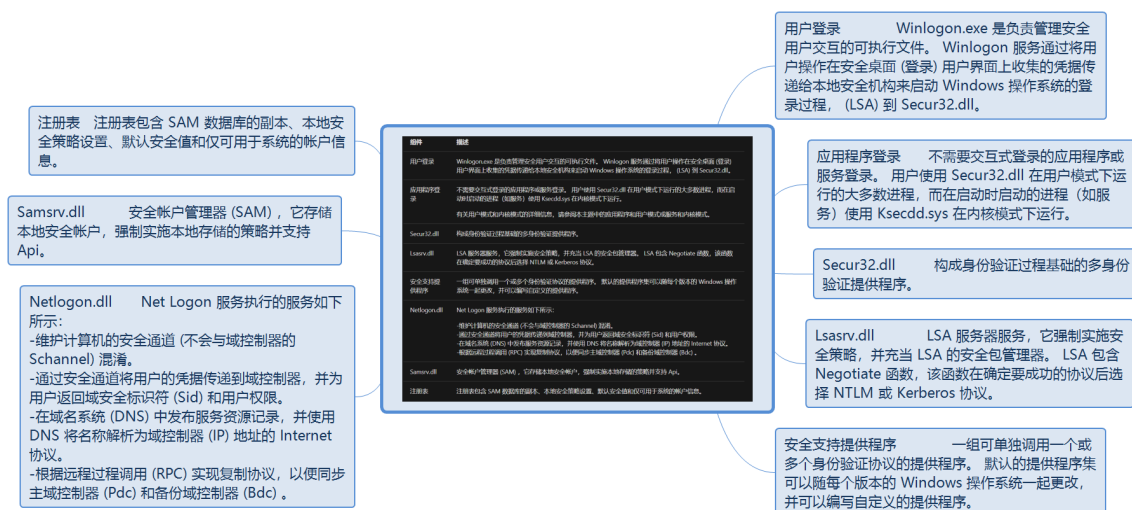
1.4.2.1. 如图

1.4.3. 进程

进程

组件	描述
用户登录	Winlogon.exe 是负责管理安全用户交互的可执行文件。Winlogon 服务通过将用户操作在安全桌面 (登录) 用户界面上收集的凭据传递给本地安全机构来启动 Windows 操作系统的登录过程, (LSA) 到 Secur32.dll。
应用程序登录	不需要交互式登录的应用程序或服务登录。用户使用 Secur32.dll 在用户模式下运行的大多数进程, 而在启动时启动的进程 (如服务) 使用 Ksecdd.sys 在内核模式下运行。 有关用户模式和内核模式的详细信息, 请参阅本主题中的应用程序和用户模式或服务模式和内核模式。
Secur32.dll	构成身份验证过程基础的多身份验证提供程序。
Lsasrv.dll	LSA 服务器服务, 它强制实施安全策略, 并充当 LSA 的安全包管理器。LSA 包含 Negotiate 函数, 该函数在确定要成功的协议后选择 NTLM 或 Kerberos 协议。
安全支持提供程序	一组可单独调用一个或多个身份验证协议的提供程序。默认的提供程序集可以随每个版本的 Windows 操作系统一起更改, 并可以编写自定义的提供程序。
Netlogon.dll	Net Logon 服务执行的服务如下所示: -维护计算机的安全通道 (不会与域控制器的 Schannel) 混淆。 -通过安全通道将用户的凭据传递到域控制器, 并为用户返回域安全标识符 (Sid) 和用户权限。 -在域名系统 (DNS) 中发布服务资源记录, 并使用 DNS 将名称解析为域控制器 (IP) 地址的 Internet 协议。 -根据远程过程调用 (RPC) 实现复制协议, 以便同步主域控制器 (Pdc) 和备份域控制器 (Bdc)。
Samsrv.dll	安全帐户管理器 (SAM), 它存储本地安全帐户, 强制实施本地存储的策略并支持 Api。
注册表	注册表包含 SAM 数据库的副本、本地安全策略设置、默认安全值和仅可用于系统的帐户信息。

1.4.3.1.



1.4.3.1.1. 用户登录 Winlogon.exe 是负责管理安全用户交互的可执行文件。

Winlogon 服务通过将用户操作在安全桌面 (登录)

用户界面上收集的凭据传递给本地安全机构来启动 Windows

操作系统的登录过程, (LSA) 到 Secur32.dll。

1.4.3.1.2. 应用程序登录 不需要交互式登录的应用程序或服务登录。

用户使用 Secur32.dll

在用户模式下运行的大多数进程, 而在启动时启动的进程 (如服务) 使用

Ksecdd.sys 在内核模式下运行。

1.4.3.1.3. Secur32.dll 构成身份验证过程基础的多身份验证提供程序。

1.4.3.1.4. Lsasrv.dll LSA 服务器服务, 它强制实施安全策略, 并充当 LSA

的安全包管理器。LSA 包含 Negotiate

函数, 该函数在确定要成功的协议后选择 NTLM 或 Kerberos 协议。

1.4.3.1.5. 安全支持提供程序

一组可单独调用一个或多个身份验证协议的提供程序。

默认的提供程序集可以随每个版本的 Windows 操作系统一起更改，并可以编写自定义的提供程序。

1.4.3.1.6. Netlogon.dll Net Logon 服务执行的服务如下所示：

- 维护计算机的安全通道 (不会与域控制器的 Schannel) 混淆。
- 通过安全通道将用户的凭据传递到域控制器，并为用户返回域安全标识符 (Sid) 和用户权限。
- 在域名系统 (DNS) 中发布服务资源记录，并使用 DNS 将名称解析为域控制器 (IP) 地址的 Internet 协议。
- 根据远程过程调用 (RPC) 实现复制协议，以便同步主域控制器 (Pdc) 和备份域控制器 (Bdc) 。

1.4.3.1.7. Samsrv.dll 安全帐户管理器 (SAM)

，它存储本地安全帐户，强制实施本地存储的策略并支持 Api。

1.4.3.1.8. 注册表 注册表包含 SAM

数据库的副本、本地安全策略设置、默认安全值和仅可用于系统的帐户信息。

1.5. 安全主体和帐户

安全主体和帐户

在 Windows 中，可以启动操作的任何用户、服务、组或计算机都是安全主体。安全主体具有可在计算机本地或基于域的帐户。例如，即使没有人为用户登录，Windows 客户端加入域的计算机也可以通过与域控制器通信来加入网络域。若要启动通信，计算机必须在域中具有活动帐户。在接受来自计算机的通信之前，域控制器上的本地安全机构会对计算机的标识进行身份验证，然后定义计算机的安全上下文，就像对安全主体而言。此安全上下文定义特定计算机或网络上的用户、服务、组或计算机上的用户或服务的标识和功能。例如，它定义了可以访问的资源（例如文件共享或打印机）以及可以由用户、服务或该资源上的计算机执行的操作，如读取、写入或修改。

1.5.1. 在 Windows

中，可以启动操作的任何用户、服务、组或计算机都是安全主体。

安全主体具有可在计算机本地或基于域的帐户。

例如，即使没有人为用户登录，**Windows**

客户端加入域的计算机也可以通过与域控制器通信来加入网络域。

若要启动通信，计算机必须在域中具有活动帐户。

在接受来自计算机的通信之前，域控制器上的本地安全机构会对计算机的标识进行身份验证，然后定义计算机的安全上下文，就像对安全主体而言。

此安全上下文定义特定计算机或网络上的用户、服务、组或计算机上的用户或服务的标识和功能。

例如，它定义了可以访问的资源（例如文件共享或打印机）以及可以由用户、服务或该资源上的计算机执行的操作，如读取、写入或修改。

1.6. 委托身份验证

通过将服务或计算机建立为受信任的委派，可让服务或计算机完成委托身份验证，接收发出请求的用户的票证，然后访问该用户的信息。此模型仅将后端服务器上的数据访问限制到那些使用正确的访问控制令牌提供凭据的用户或服务。此外，它还允许对这些后端资源进行访问审核。通过要求通过委托给服务器的凭据访问所有数据以代表客户端使用，你可以确保服务器不会泄露，并且你可以获取对存储在其他服务器上的敏感信息的访问权限。对于设计为在多台计算机上使用单一登录功能的多层应用程序，委派身份验证非常有用。

委托身份验证

在 Windows 中，当网络服务接受来自用户的身份验证请求并假定该用户的标识以便启动到第二个网络服务的新连接时，将进行委派的身份验证。若要支持委派的身份验证，必须建立前端或第一层服务器（如 web 服务器），这些服务器负责处理客户端身份验证请求以及用于存储信息的后端或 n 层服务器（如大型数据库）。你可以委派权限，以便为组织中的用户设置委派的身份验证

1.6.1. 在 Windows

中，当网络服务接受来自用户的身份验证请求并假定该用户的标识以便启动到第二个网络服务的新连接时，将进行委派的身份验证。

若要支持委派的身份验证，必须建立前端或第一层服务器（如 web 服务器），这些服务器负责处理客户端身份验证请求以及用于存储信息的后端或 n 层服务器（如大型数据库）。

你可以委派权限，以便为组织中的用户设置委派的身份验证

1.6.2. 通过将服务或计算机建立为受信任的委派，可让服务或计算机完成委托身份验证，接收发出请求的用户的票证，然后访问该用户的信息。

此模型仅将后端服务器上的数据访问限制到那些使用正确的访问控制令牌提供凭据的用户或服务。此外，它还允许对这些后端资源进行访问审核。

通过要求通过委托给服务器的凭据访问所有数据以代表客户端使用，你可以确保服务器不会泄露，并且你可以获取对存储在其他服务器上的敏感信息的访问权限。

对于设计为在多台计算机上使用单一登录功能的多层应用程序，委派身份验证非常有用。

1.7. 域之间的信任关系中的身份验证

域之间的信任关系中的身份验证

拥有多个域的大多数组织都有权访问位于不同域中的共享资源控制此访问要求一个域中的用户也可以通过身份验证和授权使用另一个域中的资源。若要在不同域中的客户端和服务器之间提供身份验证和授权功能，两个域之间必须存在信任。信任是 Active Directory 通信的基本技术，是 Windows Server 网络体系结构的一个有机安全组件。如果两个域之间存在信任，则每个域的身份验证机制将信任来自其他域的身份验证。信任帮助提供对资源域中共享资源的受控访问权限（信任域），验证传入的身份验证请求是否来自受信任的颁发机构--受信任的域。通过这种方式，信任充当桥，只允许验证的身份验证请求在域之间传输。特定信任通过身份验证请求的方式取决于其配置方式。信任关系可以是单向的，即提供从受信任域到信任域中资源的访问权限，或通过提供从每个域到其他域中资源的访问权限来实现双向关系。信任也是不可传递的，在这种情况下，信任关系仅存在于两个信任伙伴域之间，或者是可传递的，在这种情况下，信任会自动扩展到合作伙伴信任的任何其他域。

1.7.1. 拥有多个域的大多数组织都有权访问位于不同域中的共享资源控制此访问要求一个域中的用户也可以通过身份验证和授权使用另一个域中的资源。

若要在不同域中的客户端和服务器之间提供身份验证和授权功能，两个域之间必须存在信任。信任是 **Active Directory** 通信的基本技术，是 **Windows Server** 网络体系结构的一个有机安全组件。

如果两个域之间存在信任，则每个域的身份验证机制将信任来自其他域的身份验证。

信任帮助提供对资源域中共享资源的受控访问权限（信任域），验证传入的身份验证请求是否来自受信任的颁发机构--受信任的域。

通过这种方式，信任充当桥，只允许验证的身份验证请求在域之间传输。

特定信任通过身份验证请求的方式取决于其配置方式。

信任关系可以是单向的，即提供从受信任域到信任域中资源的访问权限，或通过提供从每个域到其他域中资源的访问权限来实现双向关系。

信任也是不可传递的，在这种情况下，信任关系仅存在于两个信任伙伴域之间，或者是可传递的，在这种情况下，信任会自动扩展到合作伙伴信任的任何其他域。

1.8. 协议转换

协议转换

协议转换通过允许应用程序在用户身份验证层支持不同的身份验证机制，并通过在后续应用程序层切换到 Kerberos 协议（如相互身份验证和约束委派）来帮助应用程序设计人员。

1.8.1. 协议转换通过允许应用程序在用户身份验证层支持不同的身份验证机制，并通过在后续应用程序层切换到 Kerberos 协议（如相互身份验证和约束委派）来帮助应用程序设计人员。

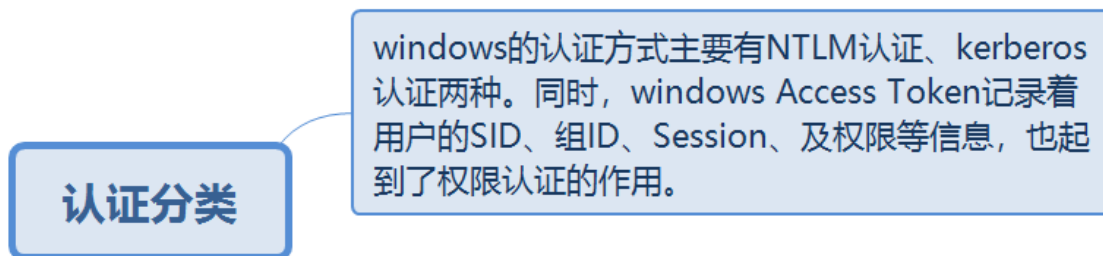
1.9. 约束委派

约束委派

约束委派使管理员能够通过限制应用程序服务可以代表用户操作的范围来指定和强制实施应用程序信任边界。你可以指定特定服务，通过这些服务，可委托的计算机可以请求资源。限制服务的授权权限的灵活性有助于通过减少不受信任的服务的泄露机会来改善应用程序安全设计。

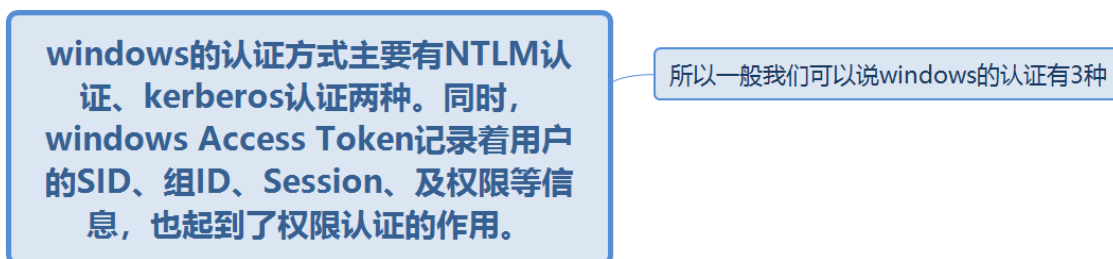
1.9.1. 约束委派使管理员能够通过限制应用程序服务可以代表用户操作的范围来指定和强制实施应用程序信任边界。
你可以指定特定服务，通过这些服务，可委托的计算机可以请求资源。
限制服务的授权权限的灵活性有助于通过减少不受信任的服务的泄露机会来改善应用程序安全设计。

2. 认证分类



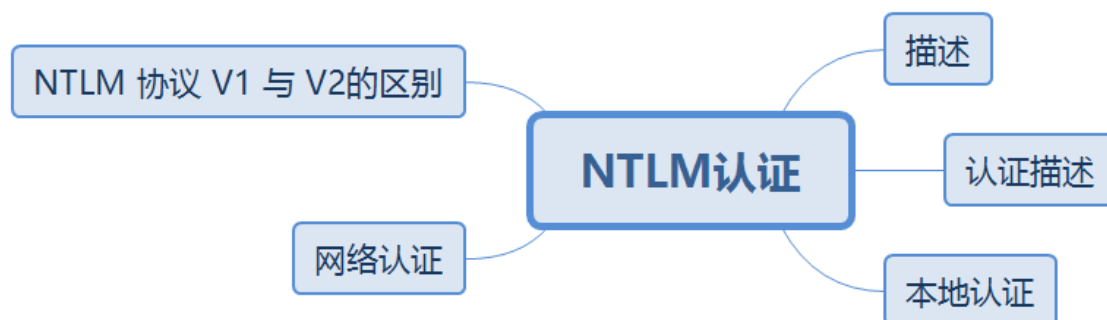
2.1.windows的认证方式主要有NTLM认证、kerberos认证两种。同时， windows Access

Token记录着用户的SID、组ID、Session、及权限等信息，也起到了权限认证的作用。



2.1.1. 所以一般我们可以说windows的认证有3种

3. NTLM认证



3.1. 描述

描述

随着更安全的身份验证协议（例如Kerberos协议）的出现，业界对在其环境中更好地管理NTLM协议的能力的要求不断提高。减少IT环境中NTLM协议的使用，既需要了解NTLM上已部署的应用程序要求，也需要配置计算环境以使用其他协议所需的策略和步骤

3.1.1. 随着更安全的身份验证协议（例如Kerberos协议）的出现，业界对在其环境中更好地管理NTLM协议的能力的要求不断提高。减少IT环境中NTLM协议的使用，既需要了解NTLM上已部署的应用程序要求，也需要配置计算环境以使用其他协议所需的策略和步骤

随着更安全的身份验证协议（例如Kerberos协议）的出现，业界对在其环境中更好地管理NTLM协议的能力的要求不断提高。减少IT环境中NTLM协议的使用，既需要了解NTLM上已部署的应用程序要求，也需要配置计算环境以使用其他协议所需的策略和步骤

微软官方看到的，但其实国内还是使用的

参见：

NTLM是一种网络认证协议，它是基于挑战(Challenge)/响应(Response)认证机制的一种认证模式。(这个协议只支持Windows)

NTLM协议的认证共需要三个消息完成：协商 --> 挑战 --> 认证。

协商：主要用于确认双方协议版本、加密等级等

挑战：服务器在收到客户端的协商消息之后，

会读取其中的内容，并从中选择出自己所能接受的服务内容，加密等级，安全服务等等。并生成一个随机数challenge，

然后生成challenge消息返回给客户端。该消息就是挑战/响应认证机制的主要功能体现。

认证：验证主要是在挑战完成后，验证结果，是认证的最后一步。

3.1.1.1. 微软官方看到的，但其实国内还是使用的

3.2. 认证描述

认证描述

NTLM认证主要有本地认证和网络认证两种方式。本地登陆时用户密码存储在SAM文件中，可以把它当作一个存储密码的数据库，所有的操作都在本地进行的。它将用户输入的密码转换为NTLM Hash,然后与SAM中的NTLM Hash进行比较。而网络认证则是基于一种Challenge/Response认证机制的认证模式。

3.2.1. NTLM认证主要有本地认证和网络认证两种方式。本地登陆时用户密码存储在SAM文件中，可以把它当作一个存储密码的数据库，所有的操作都在本地进行的。它将用户输入的密码转换为NTLM Hash,然后与SAM中的NTLM Hash进行比较。而网络认证则是基于一种Challenge/Response认证机制的认证模式。

参见:

NTLM是一种网络认证协议,它是基于挑战(Challenge)/响应(Response)认证机制的一种认证模式。(这个协议只支持Windows)

NTLM协议的认证共需要三个消息完成:协商 --> 挑战 --> 认证。

协商:主要用于确认双方协议版本、加密等级等

挑战:服务器在收到客户端的协商消息之后,

会读取其中的内容,并从中选择出自己所能接受的服务内容,加密等级,安全服务等等。并生成一个随机数challenge,

然后生成challenge消息返回给客户端。该消息就是挑战/响应认证机制的主要功能体现。

认证:验证主要是在挑战完成后,验证结果,是认证的最后一步。

3.3. 本地认证



3.3.1. 描述

描述

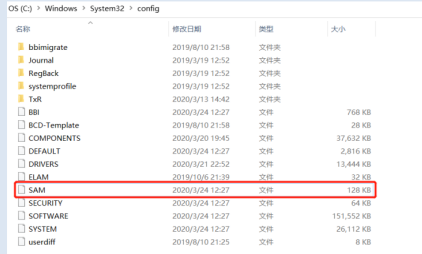
在本地登陆的情况下，操作系统会使用用户输入的密码作为凭据去与系统中的密码进行校验，如果成功的话表明验证通过。操作系统的密码存储在C盘的目录下：

`%SystemRoot%\system32\config\sam`

SAM用于储存本地所有用户的凭证信息，但是这并不代表着你可以随意去查看系统密码。我们登陆系统的时候系统会自动读取SAM文件中的密码与我们输入的密码进行比对，如果认证相同则可以使用该机器。

windows自身是不会保存明文密码的，也就是说SAM中保存的不是明文而是Hash。

Hash，一般翻译做散列、杂凑，或音译为哈希，是把任意长度的输入（又叫做预映射pre-image）通过散列算法变换成固定长度的输出，该输出就是散列值。这种转换是一种压缩映射，也就是，散列值的空间通常远小于输入的空间，不同的输入可能会散列成相同的输出，所以不可能从散列值来确定唯一的输入值。简单的说就是一种将任意长度的消息压缩到某一固定长度的消息摘要的函数。



3.3.1.1. 在本地登陆的情况下，操作系统会使用用户输入的密码作为凭据去与系统中的密码进行校验，如果成功的话表明验证通过。操作系统的密码存储在C盘的目录下：

`%SystemRoot%\system32\config\sam`

SAM用于储存本地所有用户的凭证信息，但是这并不代表着你可以随意去查看系统密码。我们登陆系统的时候系统会自动读取**SAM**文件中的密码与我们输入的密码进行比对，如果认证相同则可以使用该机器。

windows自身是不会保存明文密码的，也就是说**SAM**中保存的不是明文而是**H**ash。

Hash，一般翻译做散列、杂凑，或音译为哈希，是把任意长度的输入（又叫做预映射**pre-image**）通过散列算法变换成固定长度的输出，该输出就是散列值。这种转换是一种压缩映射，也就是，散列值的空间通常远小于输入的空间，不同的输入可能会散列成相同的输出，所以不可能从散列值来确定唯一的输入值。

简单的说就是一种将任意长度的消息压缩到某一固定长度的消息摘要的函数

。

3.3.2. NTLM Hash生成

NTLM Hash生成

NTLM Hash是怎样生成的呢？当用户注销、重启、锁屏后，操作系统会让winlogon显示登陆界面，当winlogon.exe接收到账号密码输入之后，会将密码交给lsass进程，这个进程会存一份明文密码，将明文密码加密成NTLM Hash，对SAM数据库比较认证。（winlogon.exe即Windows Logon Process，是Windows NT用户登陆程序，用于管理用户登录和退出。LSASS用于微软Windows系统的安全机制。它用于本地安全和登陆策略。）
比如当用户输入密码admin的时候，操作系统会将admin转换为16进制，经过Unicode转换后，再调用MD4加密算法加密，这个加密结果的十六进制就是NTLM Hash

admin -> hex(16进制编码) = 61646d696e
61646d696e -> Unicode = 610064006d0069006e00
610064006d0069006e00 -> MD4 = 209c6174da490caeb422f3fa5a7ae634

3.3.2.1. NTLM

Hash是怎么样生成的呢？当用户注销、重启、锁屏后，操作系统会让winlogon显示登陆界面，当winlogon.exe接收到账号密码输入之后，会将密码交给lsass进程，这个进程会存一份明文密码，将明文密码加密成NTLM

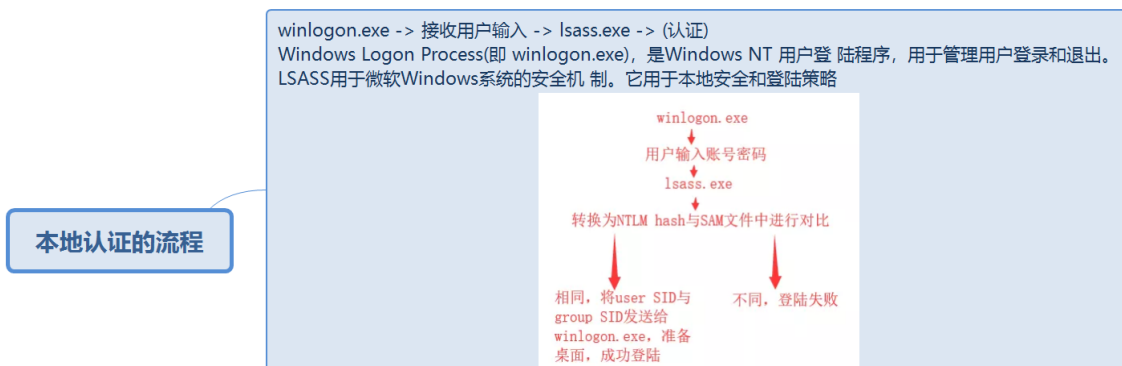
Hash，对SAM数据库比较认证。（winlogon.exe即Windows Logon

Process，是Windows

NT用户登陆程序，用于管理用户登录和退出。LSASS用于微软Windows系统的安全机制。它用于本地安全和登陆策略。）

比如当用户输入密码admin的时候，操作系统会将admin转换为16进制，经过Unicode转换后，再调用MD4加密算法加密，这个加密结果的十六进制就是NTLM Hash

3.3.3. 本地认证的流程



3.3.3.1. winlogon.exe -> 接收用户输入 -> lsass.exe -> (认证)

Windows Logon Process(即 winlogon.exe), 是Windows NT 用户登陆程序, 用于管理用户登录和退出。

LSASS用于微软Windows系统的安全机制。它用于本地安全和登陆策略

3.3.4. NTLM前身LM Hash

NTLM前身LM Hash

LM与NTLM协议的认证机制相同, 但是加密算法不同。目前大多数的windows系统都采用了NTLM协议认证, LM协议现在已经基本已经淘汰了。LM协议认证过程中需要LM Hash作为根本凭证进行参与认证。LM Hash产生原理:

- 将所有小写字母转换为大写字母
- >123ABC // 未达16个字符
- 将密码转换为16进制, 分两组, 填充为16个字符, 空位使用0x00字符填充
- >31212341242430000000000000000000
- 将密码分为两组7个字符的块
- >312123412424300 0000000000000000 // 16进制
- 将每块转换为比特流, 不足56位1补在左边加0
- >312123412424300 -> 1(转换为二进制) 11000110011000110011011000001010000100100001100000000 -> (补) 15
- 将比特流按照7比特一组, 分出4组, 求反加0

由于后者都为0, 结果可想而知, 那就都是0;

将每块比特流转换为16进制作为加密的key, 使用DES加密, 字符串 "KGS{045K}"为key(0x04047532140232425), 得到:

-> 001100001100110011000110001101100000010100000100100001100 00000000

-> 300000003014120000 -> DES(300000003014120000) -> 48-07-EB-91-2F-5E-69-7C

由于我们的密码不超过7个字节, 所以后面的一半是固定的:

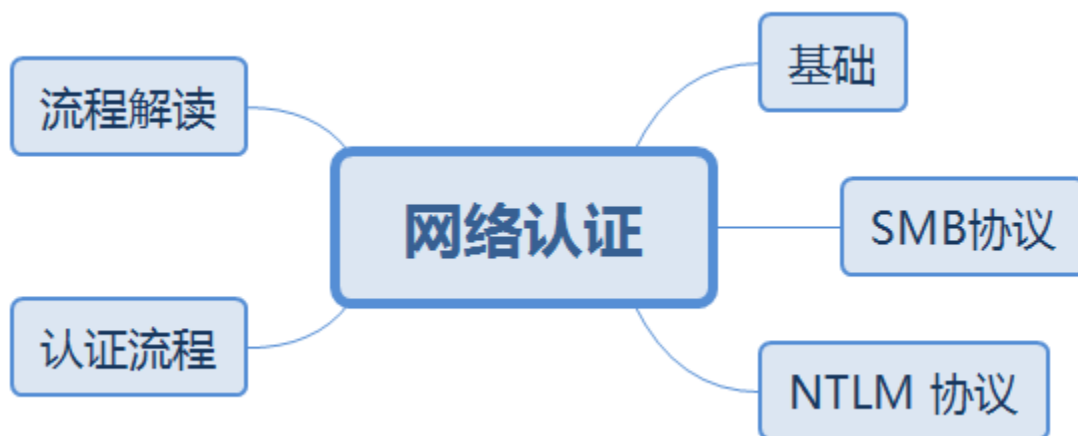
AA-03-84-35-85-14-04-EE

最后两个0x35和0x84是固定的, 这是LmHash。

48-07-EB-91-2F-5E-69-7C-AA-03-84-35-85-14-04-EE

3.3.4.1. LM与NTLM协议的认证机制相同, 但是加密算法不同。目前大多数的windows系统都采用了NTLM协议认证, LM协议现在已经基本已经淘汰了。LM协议认证过程中需要LM Hash作为根本凭证进行参与认证。LM Hash产生原理:

3.4. 网络认证



3.4.1. 基础

在工作组中，无论是局域网中的一台机器还是很多机器，它们能够通信的话都无法相互建立一个完美的信任机制。只要有一个可以信任的信托机构，对两方进行认证，这样就有第三方来证实双方的可信任性。

3.4.1.1. 在工作组中，无论是局域网中的一台机器还是很多机器，它们能够通信的话都无法相互建立一个完美的信任机制。只要有一个可以信任的信托机构，对两方进行认证，这样就有第三方来证实双方的可信任性。

3.4.2. SMB协议

在了解认证之前先了解一些SMB协议：

SMB (ServerMessage Block) 通信协议是微软 (Microsoft) 和英特尔(Intel)在1987年制定的协议，主要是作为Microsoft网络的通讯协议。SMB 是在会话层 (session layer) 和表示层 (presentation layer) 以及小部分应用层 (application layer) 的协议。SMB使用了NetBIOS的应用程序接口 (Application Program Interface, 简称API)，一般端口使用为139, 445。另外，它是一个开放性的协议，允许了协议扩展——使得它变得更大而且复杂；大约有65个最上层的作业，而每个作业都超过120个函数，甚至Windows NT也没有全部支持到，最近微软又把 SMB 改名为 CIFS (CommonInternet File System)，并且加入了许多新的特色。

早期SMB协议在网络上传输的是明文口令。后来出现LAN Manager ChallengeReponse 验证机制，简称LM。微软提出了windowsNT挑战/响应验证机制，简称MTLM。现在已经更新到了V2版本以及加入了Kerberos验证体系

3.4.2.1. 在了解认证之前先了解一些SMB协议：

SMB (ServerMessage

Block) 通信协议是微软 (Microsoft) 和英特尔(Intel)在1987年制定的协议，主要是作为Microsoft网络的通讯协议。SMB 是在会话层 (session layer) 和表示层 (presentation layer) 以及小部分应用层 (application layer) 的协议。SMB使用了NetBIOS的应用程序接口 (Application Program Interface, 简称API)，一般端口使用为139, 445。另外，它是一个开放性的协议，允许了协议扩展——

使得它变得更大而且复杂；大约有65个最上层的作业，而每个作业都超过120个函数，甚至Windows NT也没有全部支持到，最近微软又把 SMB 改名为 CIFS (CommonInternet File System)，并且加入了许多新的特色。

早期SMB协议在网络上传输的是明文口令。后来出现LAN Manager ChallengeReponse 验证机制，简称LM。

微软提出了windowsNT挑战/响应验证机制，简称MTLM。现在已经更新到了V2版本以及加入了Kerberos验证体系

3.4.3. NTLM 协议

NTLM 协议

NTLM是一种网络认证协议，它是基于挑战（Challenge）/响应（Response）认证机制的一种认证模式。（这个协议只支持Windows）
NTLM协议的认证共需要三个消息完成：协商 --> 挑战 --> 认证。

协商：主要用于确认双方协议版本、加密等级等

挑战：服务器在收到客户端的协商消息之后，会读取其中的内容，并从中选择出自己所能接受的服务内容，加密等级，安全服务等等。并生成一个随机数challenge，然后生成challenge消息返回给客户端。该消息就是挑战/响应认证机制的主要功能体现。

认证：验证主要是在挑战完成后，验证结果，是认证的最后一步。

3.4.3.1. NTLM是一种网络认证协议，它是基于挑战（Challenge）/响应（Response）认证机制的一种认证模式。（这个协议只支持Windows）

NTLM协议的认证共需要三个消息完成：协商 --> 挑战 --> 认证。

协商：主要用于确认双方协议版本、加密等级等

挑战：服务器在收到客户端的协商消息之后，

会读取其中的内容，并从中选择出自己所能接受的服务内容，加密等级，安全服务等等。并生成一个随机数challenge，

然后生成challenge消息返回给客户端。该消息就是挑战/响应认证机制的主要功能体现。

认证：验证主要是在挑战完成后，验证结果，是认证的最后一步。

参见：

随着更安全的身份验证协议（例如Kerberos协议）的出现，业界对在其环境中更好地管理NTLM协议的能力的要求不断提高。减少IT环境中NTLM协议的使用，既需要了解NTLM上已部署的应用程序要求，也需要配置计算环境以使用其他协议所需的策略和步骤，

NTLM认证主要有本地认证和网络认证两种方式。本地登陆时用户密码存储在SAM文件中，可以把它当作一个存储密码的数据库，所有的操作都在本地进行的。它将用户输入的密码转换为NTLM Hash,然后与SAM中的NTLM Hash进行比较。而网络认证则是基于一种Challenge/Response认证机制的认证模式。

3.4.4. 认证流程

认证流程

首先客户端向服务器发送一些用户信息以及主机信息（包含用户名，如果没有这个用户的话认证就失败），服务器接收到请求之后会生成一个16为的随机数称之为"Challenge"(挑战)，使用登陆用户名对应的NTLM Hash加密Challenge(其实就是验证它是否在我本地认证的数据库中)生成Challenge1，同时将Challenge发送给客户端。

Net NTLM Hash = NTLM Hash(Challenge)

Challenge是服务端随机生成的、NTLM Hash是服务器根据客户端提供的用户名寻找出来的Hash加密生成Net NTLM Hash(这个是在服务端这边)

客户端接收到Challenge后使用将要登陆到账户对应的NTLM Hash加密Challenge生成Response之后发给服务端，服务器接收到客户端发过来的Response后，对比Challenge1与Response是否相等，从而进行验证。

3.4.4.1. 首先客户端向服务器发送一些用户信息以及主机信息（包含用户名，如果没有这个用户的话认证就失败），服务器接收到请求之后会生成一个16为的随机数称之为"Challenge"(挑战)，使用登陆用户名对应的NTLM Hash加密Challenge(其实就是验证它是否在我本地认证的数据库中)生成Challenge1，同时将Challenge发送给客户端。

Net NTLM Hash = NTLM Hash(Challenge)

Challenge是服务端随机生成的、NTLM

Hash是服务器根据客户端提供的用户名寻找出来的Hash加密生成Net NTLM Hash(这个是在服务端这边)

客户端接收到Challenge后使用将要登陆到账户对应的NTLM

Hash加密Challenge生成Response之后发给服务端，服务器接收到客户端发过来的Response后，对比Challenge1与Response是否相等，从而进行验证。

3.4.5. 流程解读

流程解读

过程:

第一步：输入密码,然后LSASS会把密码的NTLM Hash后的值先存储到本地。

第二步：客户端把用户名的明文发送给服务端

第三步：服务端接收到用户名之后会判断用户名是否存在，不存在则代表认证失败，存在的话服务端会生成一个16位的随机数,并且从本地查找share_user对应的NTLM Hash，使用NTLM Hash加密Challenge，生成一个Net-NTLM Hash存在内存中，并将Challenge发送给客户端。

第四步：当客户端收到challenge后,用在第一步中存储的NTLM Hash对其加密，然后再将加密后的challenge发送给服务器，也就是response，表现形式是Net-NTLM Hash。

第五步：服务端在收到response后，会向DC发送针对客户端的验证请求。该请求主要包含以下三方面的内容：客户端用户名、客户端NTLM Hash加密的Challenge、原始的Challenge。

第六步：当DC接收过来的这三个值的以后,会根据用户名到DC的账号数据库(ntds.dit)里面找到该用户名对应的NTLM Hash,然后把这个hash拿出来和传过来的challenge值进行比较,相同则认证成功,反之,则失败。

3.4.5.1. 过程:

第一步：输入密码,然后LSASS会把密码的NTLM Hash后的值先存储到本地。

第二步：客户端把用户名的明文发送给服务端

第三步：服务端接收到用户名之后会判断用户名是否存在，不存在则代表认

证失败。存在的话服务端会生成一个16位的随机数,并且从本地查找share_user对应的NTLM Hash,使用NTLM Hash加密Challenge,生成一个Net-NTLM Hash存在内存中,并将Challenge发送给客户端。

第四步:当客户端收到challenge后,用在第一步中存储的NTLM Hash对其加密,然后再将加密后的challenge发送给服务器,也就是response,表现形式是Net-NTLM Hash。

第五步:服务端在收到response后,会向DC发送针对客户端的验证请求。该请求主要包含以下三方面的内容:客户端用户名、客户端NTLM Hash加密的Challenge、原始的Challenge。

第六步:当DC接到过来的这三个值的以后,会根据用户名到DC的账号数据库(ntds.dit)里面找到该用户名对应的NTLM Hash,然后把这个hash拿出来和传过来的challenge值进行比较,相同则认证成功,反之,则失败。

3.5. NTLM 协议 V1 与 V2的区别

NTLM 协议 V1 与 V2的区别

NTLM 协议 V1 与 V2的区别
NTLM v1与NTLM v2最显著的区别就是Challenge与加密算法不同,共同点就是加密的原料都是NTLM Hash。
Challenge:NTLM v1的Challenge有8位, NTLM v2的Challenge为16位。
Net-NTLM Hash:NTLM v1的主要加密算法是DES, NTLM v2的主要加密算法是HMAC-MD5。

3.5.1. NTLM 协议 V1 与 V2的区别

NTLM v1与NTLM

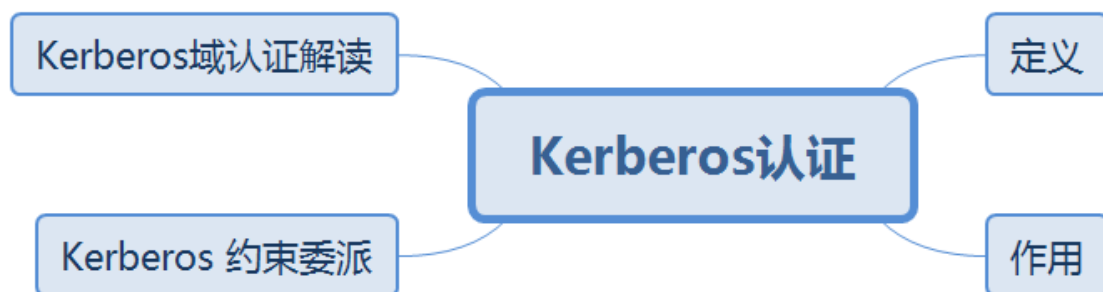
v2最显著的区别就是Challenge与加密算法不同,共同点就是加密的原料都是NTLM Hash。

Challenge:NTLM v1的Challenge有8位, NTLM v2的Challenge为16位。

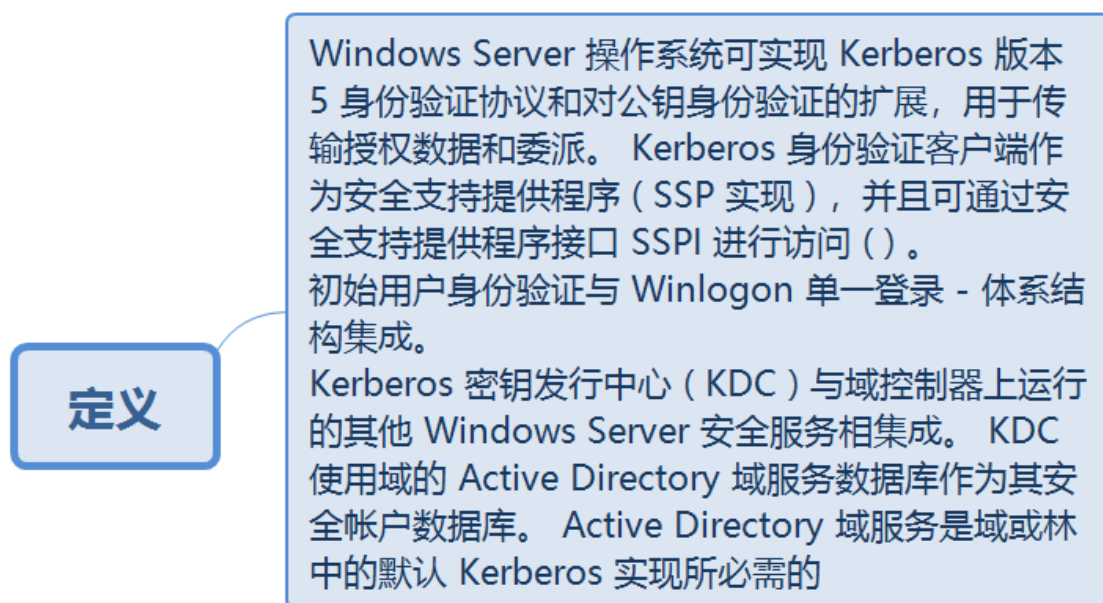
Net-NTLM Hash:NTLM v1的主要加密算法是DES, NTLM

v2的主要加密算法是HMAC-MD5。

4. Kerberos认证



4.1. 定义



参见: [定义](#), 在 [Windows Server 2003 中引入的 Kerberos](#)

[约束委派](#), 为服务所使用的委派提供了一种更安全的形式。配置了 Kerberos

[约束委派](#)后, 它限制了指定服务器可代表用户执行的服务。

[这需要域管理员权限来为服务配置一个域账户, 并把该账户限制到单个域。](#)

[在如今的企业中, 前端服务的设计并不局限于仅与域中的服务进行集成。](#)

[在域管理员配置了服务的早期操作系统中, 服务管理员没有有效途径来了解哪些前端服务委派给了其拥有的资源服务。](#)

[并且可委派给资源服务的任何前端服务都代表了一个潜在的攻击点。](#)

[如果托管前端服务的服务器受到安全威胁, 并且它已配置为委派给资源服务, 则资源服务也会受到安全威胁。](#)

[在 Windows Server 2012 R2 和 Windows Server 2012 中, 为服务配置约束委派的能力已从域管理员转移给服务管理员。这样, 后端服务管理员可以允许或拒绝前端服务。](#)

4.1.1. Windows Server 操作系统可实现 Kerberos 版本 5

身份验证协议和对公钥身份验证的扩展, 用于传输授权数据和委派。Kerberos 身份验证客户端作为安全支持提供程序 (SSP 实现)

, 并且可通过安全支持提供程序接口 SSPI 进行访问 ()。

初始用户身份验证与 Winlogon 单一登录 - 体系结构集成。

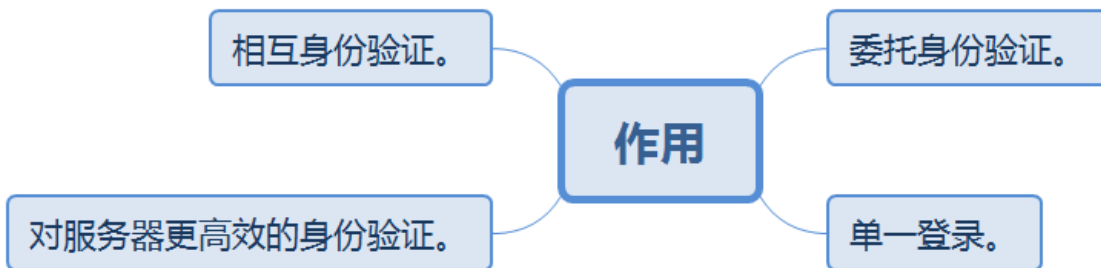
Kerberos 密钥发行中心 (KDC) 与域控制器上运行的其他 Windows Server

安全服务相集成。KDC 使用域的 Active Directory

域服务数据库作为其安全帐户数据库。Active Directory

域服务是域或林中的默认 Kerberos 实现所必需的

4.2. 作用



参见: [作用](#),

[约束委派让服务管理员能够通过限制应用程序服务可以代表用户的范围来指定和强制应用程序信任边界。](#)

[服务管理员可以配置哪些前端服务账户能委派到其后端服务。](#)

[通过支持 Windows Server 2012 R2 和 Windows Server 2012](#)

[中跨域的约束委派, 可以将前端服务 \(例如 Microsoft Internet 安全和加速 \(ISA\)](#)

[Server、Microsoft Forefront 威胁管理网关、Microsoft Exchange Outlook Web 访问](#)

[\(OWA\) 和 Microsoft SharePoint](#)

[Server\)配置为使用约束委派对其他域中的服务器进行身份验证。](#)

[这将通过使用现有的 Kerberos 基础结构来支持跨域的服务解决方案。](#)

[域管理员或服务管理员可以管理 Kerberos 约束委派。](#)

4.2.1. 委托身份验证。

委托身份验证。

在代表客户端访问资源时，在 Windows 操作系统上运行的服务可以模拟客户端计算机。通常，服务通过访问本地计算机上的资源为客户端完成工作。当客户端计算机向服务进行身份验证时，NTLM 和 Kerberos 协议都可以提供服务在本地模拟客户端计算机所需的授权信息。但是，某些分布式应用程序的设计使前端 - 服务在连接到其他计算机上的后端服务时，必须使用客户端计算机的标识 - 。Kerberos 身份验证支持一种委派机制，使服务在连接到其他服务时可以代表其客户端进行操作。

4.2.1.1. 在代表客户端访问资源时，在 Windows

操作系统上运行的服务可以模拟客户端计算机。

通常，服务通过访问本地计算机上的资源为客户端完成工作。

当客户端计算机向服务进行身份验证时，NTLM 和 Kerberos

协议都可以提供服务在本地模拟客户端计算机所需的授权信息。

但是，某些分布式应用程序的设计使前端 -

服务在连接到其他计算机上的后端服务时，必须使用客户端计算机的标识 -

。Kerberos

身份验证支持一种委派机制，使服务在连接到其他服务时可以代表其客户端进行操作。

4.2.2. 单一登录。

单一登录。

在域或林中使用 Kerberos 身份验证将允许用户或服务访问管理员允许访问的资源，而无需多次请求凭据。在通过 Winlogon 第一次登录域之后，Kerberos 将在每次尝试访问资源时管理整个林中的凭据。

4.2.2.1. 在域或林中使用 Kerberos

身份验证将允许用户或服务访问管理员允许访问的资源，而无需多次请求凭据。在通过 Winlogon 第一次登录域之后，Kerberos 将在每次尝试访问资源时管理整个林中的凭据。

4.2.3. 对服务器更高效的身份验证。

对服务器更高效的身份验证。

在 Kerberos 出现之前，可以使用 NTLM 身份验证，它要求应用程序服务器必须连接到域控制器，以便验证每个客户端计算机或服务的身份。使用 Kerberos 协议时，可续订会话票证会替代 pass-through 身份验证。服务器不需要使用域控制器，（除非它需要验证特权属性证书（PAC））。服务器可以通过检查客户端出示的凭据来验证客户端计算机的身份。客户端计算机在获得一次特定服务器的凭据后，即可在整个网络登录会话期间重复使用这些凭据。

4.2.3.1. 在 Kerberos 出现之前，可以使用 NTLM

身份验证，它要求应用程序服务器必须连接到域控制器，以便验证每个客户端计算机或服务的身份。使用 Kerberos 协议时，可续订会话票证会替代 pass-through 身份验证。服务器不需要使用域控制器，（除非它需要验证特权属性证书（PAC））。

服务器可以通过检查客户端出示的凭据来验证客户端计算机的身份。

客户端计算机在获得一次特定服务器的凭据后，即可在整个网络登录会话期间重复使用这些凭据。

4.2.4. 相互身份验证。

相互身份验证。

通过使用 Kerberos 协议，网络连接两端的每一方可验证另一方所宣称的身份。NTLM 不允许客户端验证服务器的身份，也不允许一个服务器验证另一个服务器的身份。NTLM 身份验证旨在用于服务器假定为真的网络环境。Kerberos 协议不进行此假设。

4.2.4.1. 通过使用 Kerberos

协议，网络连接两端的每一方可验证另一方所宣称的身份。NTLM 不允许客户端验证服务器的身份，也不允许一个服务器验证另一个服务器的身份。NTLM 身份验证旨在用于服务器假定为真的网络环境。Kerberos 协议不进行此假设。

4.3. Kerberos 约束委派



4.3.1. 定义

定义

在 Windows Server 2003 中引入的 Kerberos 约束委派，为服务所使用的委派提供了一种更安全的形式。配置了 Kerberos 约束委派后，它限制了指定服务器可代表用户执行的服务。这需要域管理员权限来为服务配置一个域账户，并把该账户限制到单个域。在如今的企业中，前端服务的设计并不局限于仅与域中的服务进行集成。在域管理员配置了服务的早期操作系统中，服务管理员没有有效途径来了解那些前端服务委派给了其拥有的资源服务。并且可委派给资源服务的任何前端服务都代表了一个潜在的攻击点。如果托管前端服务的服务器受到安全威胁，并且它已配置为委派给资源服务，则资源服务也会受到安全威胁。在 Windows Server 2012 R2 和 Windows Server 2012 中，为服务配置约束委派的能力已从域管理员转移给服务管理员。这样，后端服务管理员可以允许或拒绝前端服务。

4.3.1.1. 在 Windows Server 2003 中引入的 Kerberos

约束委派，为服务所使用的委派提供了一种更安全的形式。配置了 Kerberos 约束委派后，它限制了指定服务器可代表用户执行的服务。

这需要域管理员权限来为服务配置一个域账户，并把该账户限制到单个域。

在如今的企业中，前端服务的设计并不局限于仅与域中的服务进行集成。

在域管理员配置了服务的早期操作系统中，服务管理员没有有效途径来了解

哪些前端服务委派给了其拥有的资源服务。

并且可委派给资源服务的任何前端服务都代表了一个潜在的攻击点。

如果托管前端服务的服务器受到安全威胁，并且它已配置为委派给资源服务，则资源服务也会受到安全威胁。

在 **Windows Server 2012 R2** 和 **Windows Server 2012**

中，为服务配置约束委派的能力已从域管理员转移给服务管理员。

这样，后端服务管理员可以允许或拒绝前端服务。

参见: [定义](#)

4.3.2. 应用

应用

约束委派让服务管理员能够通过限制应用程序服务可以代表用户的范围来指定和强制应用程序信任边界。服务管理员可以配置哪些前端服务账户能委派到其后端服务。通过支持 Windows Server 2012 R2 和 Windows Server 2012 中跨域的约束委派，可以将前端服务（例如 Microsoft Internet 安全和加速 (ISA) Server、Microsoft Forefront 威胁管理网关、Microsoft Exchange Outlook Web 访问 (OWA) 和 Microsoft SharePoint Server）配置为使用约束委派对其他域中的服务器进行身份验证。这将通过使用现有的 Kerberos 基础结构来支持跨域的服务解决方案。域管理员或服务管理员可以管理 Kerberos 约束委派。

4.3.2.1. 约束委派让服务管理员能够通过限制应用程序服务可以代表用户的范围来指定和强制应用程序信任边界。

服务管理员可以配置哪些前端服务账户能委派到其后端服务。

通过支持 **Windows Server 2012 R2** 和 **Windows Server 2012**

中跨域的约束委派，可以将前端服务（例如 **Microsoft Internet 安全和加速 (ISA) Server**、**Microsoft Forefront 威胁管理网关**、**Microsoft Exchange Outlook Web 访问 (OWA)** 和 **Microsoft SharePoint**

Server）配置为使用约束委派对其他域中的服务器进行身份验证。

这将通过使用现有的 **Kerberos** 基础结构来支持跨域的服务解决方案。

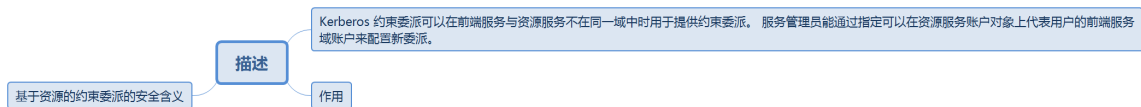
域管理员或服务管理员可以管理 **Kerberos** 约束委派。

参见: [作用](#)

4.3.3. 跨域的基于资源的约束委派



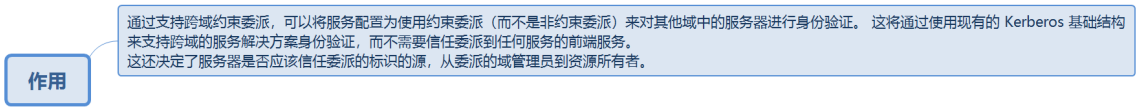
4.3.3.1. 描述



4.3.3.1.1. Kerberos

约束委派可以在前端服务与资源服务不在同一域中时用于提供约束委派。服务管理员能通过指定可以在资源服务账户对象上代表用户的前端服务域账户来配置新委派。

4.3.3.1.2. 作用



4.3.3.1.2.1. 通过支持跨域约束委派，可以将服务配置为使用约束委派（而不是非约束委派）来对其他域中的服务器进行身份验证。这将通过使用现有的 Kerberos 基础结构来支持跨域的服务解决方案身份验证，而不需要信任委派到任何服务的前端服务。这还决定了服务器是否应该信任委派的标识的源，从委派的域管理员到资源所有者。

4.3.3.1.3. 基于资源的约束委派的安全含义

基于资源的约束委派的安全含义

基于资源的约束委派会将委派控制置于拥有所访问资源的管理员。它依赖于资源服务的属性，而不是受信任的服务委托。因此，基于资源的约束委派不能使用以前控制的协议转换的受信任身份验证委托位。当执行基于资源的约束委派时，KDC 始终允许协议转换，就像设置了位一样。因为 KDC 不会限制协议转换，所以引入了两个新的已知 Sid，以将此控件授予资源管理员。这些 Sid 确定是否发生了协议转换，并可与标准访问控制列表结合使用来根据需要授予或限制访问权限。

SID	描述
AUTHENTICATION_AUTHORITY_ASSERTED_IDENTITY S-1-18-1	一个 SID，表示根据客户端凭据所有权验证，身份验证颁发机构对客户端的标志进行断言。
SERVICE_ASSERTED_IDENTITY S-1-18-2	一个 SID，表示服务对客户端的标志进行断言。

后端服务可以使用标准 ACL 表达式来确定如何对用户进行身份验证。

4.3.3.1.3.1. 基于资源的约束委派会将委派控制置于拥有所访问资源的管理员。它依赖于资源服务的属性，而不是受信任的服务委托。因此，基于资源的约束委派不能使用以前控制的协议转换的受信任身份验证委托位。当执行基于资源的约束委派时，KDC 始终允许协议转换，就像设置了位一样。因为 KDC 不会限制协议转换，所以引入了两个新的已知 Sid，以将此控件授予资源管理员。这些 Sid 确定是否发生了协议转换，并可与标准访问控制列表结合使用来根据需要授予或限制访问权限。

4.4. Kerberos域认证解读



4.4.1. 定义

定义

Kerberos 是一种网络认证协议，其设计目标是通过密钥系统为客户机 / 服务器应用程序提供强大的认证服务。该认证过程的实现不依赖于主机操作系统的认证，无需基于主机地址的信任，不要求网络上所有主机的物理安全，并假定网络上传送的数据包可以被任意地读取、修改和插入数据。在以上情况下，Kerberos 作为一种可信任的第三方认证服务，是通过传统的密码技术(如:共享密钥)执行认证服务的。

参见: [定义](#)

4.4.1.1. Kerberos 是一种网络认证协议，其设计目标是通过密钥系统为客户机 / 服务器应用程序提供强大的认证服务。该认证过程的实现不依赖于主机操作系统的认证，无需基于主机地址的信任，不要求网络上所有主机的物理安全，并假定网络上传送的数据包可以被任意地读取、修改和插入数据。在以上情况下，**Kerberos** 作为一种可信任的第三方认证服务，是通过传统的密码技术(如:共享密钥)执行认证服务的。

4.4.2. 作用

作用

在域中，网络对象可以相互访问，但是在真实情况中，需要对某些部门的计算机进行限制，例如：销售部门不能访问技术部门的服务器。这个中间就需要Kerberos认证协议来验证网络对象间的权限。

参见: [作用](#)

4.4.2.1. 在域中，网络对象可以相互访问，但是在真实情况中，需要对某些部门的计算机进行限制，例如：销售部门不能访问技术部门的服务器。

这个中间就需要Kerberos认证协议来验证网络对象间的权限。

4.4.3. 基础

名词基本概念：

KDC: Key Distribution Center, 密钥分发中心, 负责管理票据、认证票据、分发票据, 但是KDC不是一个独立的服务, 它由AS和TGS组成。
AS: Authentication Service, 验证服务, 为client生成TGT的服务
TGS: Ticket Granting Service, 票据授予服务, 为client生成某个服务的ticket
TGT: Ticket Granting Ticket, 入场券, 通过入场券能够获得票据, 是一种临时凭证的存在。
Ticket: 票据, 是网络中各对象之间互相访问的凭证
AD: Account Database, 存储所有client的白名单, 只有存在于白名单的client才能顺利申请到TGT。
DC: Domain Controller, 域控
KRBtgt: 每个域控制器都有一个krbtgt账户, 是KDC的服务账户, 用来创建TGS加密的密钥。

基础

Kerberos的标志是三只狗头, 狗头分别代表以下角色:

Client
Server

KDC(Key Distribution Center) = DC(Domain Controller)

kerberos使用了一个包含客户端、应用服务器和一个kerbroes服务器的协议, 这个协议的设计就是对抗客户端/服务器对话安全的多种威胁。在一个不受保护的的网络中, 任何一个客户端可以使用任意一台服务器提供的服务。很明显的安全威胁就是伪装, 对方可以扮演另一个客户端并在服务器上获取没有经过验证的权限! 所以服务器必须能确认请求服务的客户端的身份进行验证。为了避免给服务器更多的访问压力和每次和客户端交互的风险, 使用认证服务器(AS), 它存储了所有用户的口令并集中在一个数据库中, 然后用户就可以登陆AS进行验证身份, 如果验证通过的话它就可以把信息传达到一个应用服务器。

4.4.3.1. Kerberos的标志是三只狗头，狗头分别代表以下角色：

Client

Server

KDC(Key Distribution Center) = DC(Domain Controller)

kerberos使用了一个包含客户端、应用服务器和一个kerbroes服务器的协议, 这个协议的设计就是对抗客户端/服务器对话安全的多种威胁。在一个不受保护的的网络中, 任何一个客户端可以使用任意一台服务器提供的服务。很明显的安全威胁就是伪装, 对方可以扮演另一个客户端并在服务器上获取没有经过验证的权限! 所以服务器必须能确认请求服务的客户端的身份进行验证。为了避免给服务器更多的访问压力和每次和客户端交互的风险, 使用认证服务器(AS), 它存储了所有用户的口令并集中在一个数据库中, 然后用户就可以

登陆AS进行验证身份，如果验证通过的话它就可以把信息传达到一个应用服务器。

4.4.3.2. 名词基本概念:

KDC: Key Distribution

Center，密钥分发中心，负责管理票据、认证票据、分发票据，但是KDC不是一个独立的服务，它由AS和TGS组成。

AS: Authentication Service，验证服务，为client生成TGT的服务

TGS: Ticket Granting Service，票据授予服务，为client生成某个服务的ticket

TGT: Ticket Granting

Ticket，入场券，通过入场券能够获得票据，是一种临时凭证的存在。

Ticket:票据，是网络中各对象之间互相访问的凭证

AD: Account

Database，存储所有client的白名单，只有存在于白名单的client才能顺利申请到TGT。

DC: Domain Controller，域控

KRBtgt:

每个域控制器都有一个krbtgt账户，是KDC的服务账户，用来创建TGS加密的密钥。

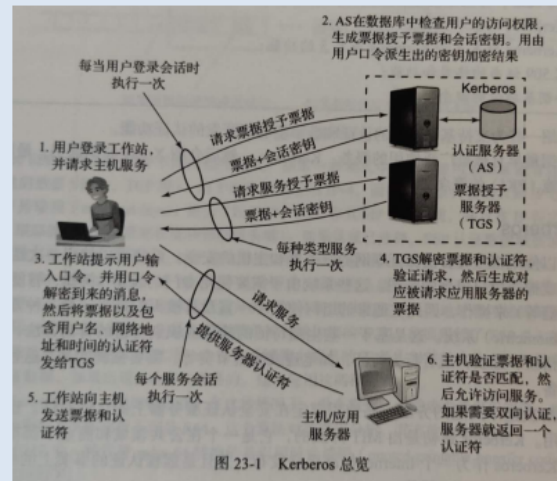
4.4.4. 域认证流程:

域认证流程:

client向kerberos服务请求, 希望获取访问server的权限。kerberos得到了这个消息, 首先得判断client是否是可信赖的, 也就是白名单黑名单的说法。这就是AS服务完成的工作, 通过 在AD中存储黑名单和白名单来区分client。成功后, 返回AS返回TGT给client。

client得到了TGT后, 继续向kerberos请求, 希望获取访问 server的权限。kerberos又得到了这个消息, 这时候通过client 消息中的TGT, 判断出了client拥有了这个权限, 给了client访 问server的权限ticket。

client得到ticket后, 终于可以成功访问server。这个ticket只是 针对这个server, 其他server需要向TGS申请。



4.4.4.1. client向kerberos服务请求, 希望获取访问server的权限。

kerberos得到了这个消息, 首先得判断client是否是可信赖的, 也就是白名单黑名单的说法。这就是AS服务完成的工作, 通过在AD中存储黑名单和白名单来区分client。成功后, 返回AS返回TGT给client。

client得到了TGT后, 继续向kerberos请求, 希望获取访问server的权限。kerberos又得到了这个消息, 这时候通过client消息中的TGT, 判断出了client拥有了这个权限, 给了client访

问server的权限ticket。

client得到ticket后，终于可以成功访问server。这个ticket只是

针对这个server，其他server需要向TGS申请。

第三步里，客户端向服务器请求，需要提供Ticket，Server Session Key加密的客户端信息与时间戳。

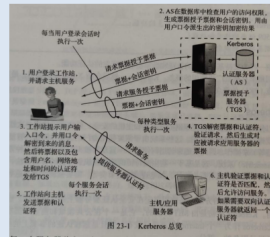
Ticket客户端无法解密
服务器端通过解密Ticket解密Server Session Key(Client info + Timestamp)
比较时间长度
校验通过后，认证成功，该票据会一直存在客户端内存中。

这个时候Kerberos与客户端已经建立起来了，客户端需要提供TGT与第一步中使用自己NTLM Hash解密出来的Session Key加密的客户端信息跟时间戳；如果假设这个数据被中间人窃取到，也无法在段时间内破解，因为KDC会校验时间戳。KDC接到TGT与其他内容后，会首先解密TGT，只有KDC可以解密TGT，从TGT中提取到Session Key，再使用Session Key解密其他内容，解密出来的内容同TGT中的信息进行校验来确认客户端是否受信；验证通过后，就会生成一个新的Session Key，我们称之为Server Session Key，这个Server Session Key主要用于和服务器进行通信。同时还会生成一个Ticket，也就是最后的票据了。

client向kerberos服务请求，希望获取访问server的权限。kerberos得到了这个消息，首先得判断client是否是可信的，也就是白名单黑名单的说法。这就是AS服务完成的工作，通过 在AD中存储黑名单和白名单来区分client。成功后，返回AS返回TGT给client。

client得到了TGT后，继续向kerberos请求，希望获取访问server的权限。kerberos又得到了这个消息，这时候通过client 消息中的TGT，判断出了client拥有了这个权限，给了client访问server的权限ticket。

client得到ticket后，终于可以成功访问server。这个ticket只是 针对这个server，其他server需要向TGS申请。



首先用户登陆到一个工作站并请求访问一个特定的服务器，客户端把一个包含用户ID和被称为TGT(Ticket-Granting Ticket,票据授予票据，也可也称为入场券)请求的消息发送到AS。其中，TGT的到期时间为8小时，如果超过了8小时，还需要重新申请TGT，不能之间进入下一步获取Ticket；AS在它的数据库中查找用户的口令，然后AS回复一个TGT和一个称为会话密钥的一次性加密密钥(可以称之为Session Key)给客户端。这两个加密都是使用用户口令作为加密密钥。然后发送给客户端，这个时候会提示客户端输入口令，产生密钥，并且解开发来的信息，如果提供了正确的口令，票据(ticket)和会话密钥就会被恢复。票据组成了一个客户端用来请求服务的信任证书的集合，票据显示AS已经接收了这个客户端和用户。票据包含了用户ID、一个时间戳、票据的失效时间。整个票据使用AS和服务共享的DES密钥加密。这个时候客户端会向AS发送TGT和解密的Session Key。

Session Key用于客户端向TGS服务通信。
域内所有网络对象的凭证都在AD中保存
KDC中某个用户指的是krbtgt

4.4.4.1.1. 首先用户登陆到一个工作站并请求访问一个特定的服务器，客户端把一个包含用户ID和被称为TGT(Ticket-Granting Ticket,票据授予票据，也可也称为入场券)请求的消息发送到AS。其中，TGT的到期时间为8小时，如果超过了8小时，还需要重新申请TGT，不能之间进入下一步获取Ticket；AS在它的数据库中查找用户的口令，然后AS回复一个TGT和一个称为会话密钥的一次性加密密钥(可以称之为Session Key)给客户端。这两个加密都是使用用户口令作为加密密钥。然后发送给客户端，这个时候会提示客户端输入口令，产生密钥，并且解开发来的信息，如果提供了正确的口令，票据(ticket)和会话密钥就会被恢复。票据组成了一个客户端用来请求服务的信任证书的集合，票据显示AS已经接收了

这个客户端和用户。票据包含了用户ID、一个时间戳、票据的失效时间。整个票据使用AS和服务器共享的DES密钥加密。这个时候客户端会向AS发送TGT和解密的Session Key。

Session Key用于客户端向TGS服务通信。

域内所有网络对象的凭证都在AD中保存

KDC中某个用户指的是krbtgt

4.4.4.1.2. 这个时候Kerberos与客户端已经建立起来了，客户端需要提供TGT与第一步中使用自己NTLM Hash解密出来的Session

Key加密的客户端信息跟时间戳；

如果假设这个数据被中间人窃取到，也无法在段时间内破解，因为KDC会校验时间戳。KDC接到TGT与其他内容后，会首先解密TGT，只有KDC可以解密TGT，从TGT中提取到Session Key，再使用Session

Key解密其他内容，解密出来的内容同TGT中的信息进行校验来确认客户端是否受信；验证通过后，就会生成一个新的Session

Key，我们称之为Server Session Key，这个Server Session

Key主要用于和服务器进行通信。同时还会生成一个Ticket，也就是最后的票据了。

4.4.4.1.3. 第三步里，客户端向服务器请求，需要提供Ticket，Server Session Key加密的客户端信息与时间戳。

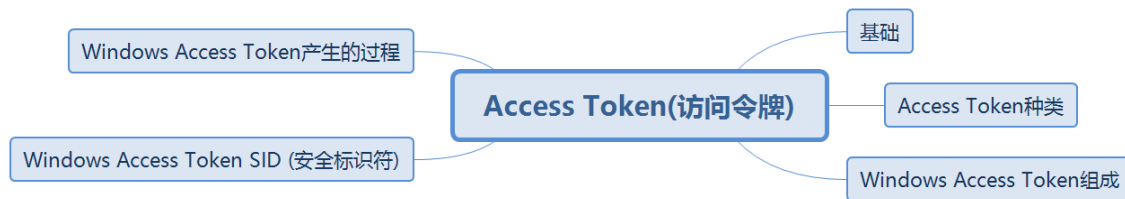
Ticket客户端无法解密

服务器端通过解密Ticket解密Server Session Key(Client info + Timestamp)

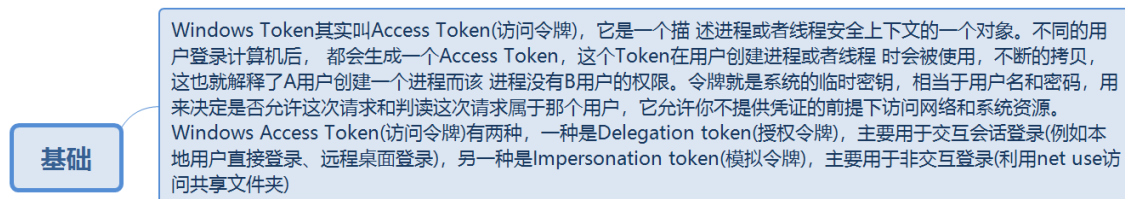
比较时间长度

校验通过后，认证成功，该票据会一直存在客户端内存中。

5. Access Token(访问令牌)



5.1. 基础



5.1.1. Windows Token其实叫Access Token(访问令牌)，它是一个描述进程或者线程安全上下文的一个对象。不同的用户登录计算机后，都会生成一个**Access Token**，这个**Token**在用户创建进程或者线程时会被使用，不断的拷贝，这也就解释了**A**用户创建一个进程而该进程没有**B**用户的权限。令牌就是系统的临时密钥，相当于用户名和密码，用来决定是否允许这次请求和判读这次请求属于那个用户，它允许你不提供凭证的前提下访问网络和系统资源。

Windows Access Token(访问令牌)有两种，一种是**Delegation token(授权令牌)**，主要用于交互会话登录(例如本地用户直接登录、远程桌面登录)，另一种是**Impersonation token(模拟令牌)**，主要用于非交互登录(利用**net use**访问共享文件夹)

5.2. Access Token种类



5.2.1. 主令牌

5.2.2. 模拟令牌

5.3. Windows Access Token组成



5.3.1. 用户帐户的安全标识符(SID)

用户所属的组的SID

用于标识当前登录会话的登录SID

用户或用户组所拥有的权限列表

所有者SID

主要组的SID

访问控制列表

访问令牌的来源
令牌是主要令牌还是模拟令牌
限制SID的可选列表
目前的模拟等级
其他统计数据

5.4. Windows Access Token SID (安全标识符)

Windows Access Token SID (安全标识符)

安全标识符是一个唯一的字符串，它可以代表一个账户、一个用户组、或者是一次登录。通常它还有一个SID固定列表，例如 Everyone这种已经内置的账户，默认拥有固定的SID。
SID的表现形式：
域SID-用户ID
计算机SID-用户ID
SID列表都会存储在域控的AD或者计算机本地账户数据库中。

5.4.1. 安全标识符是一个唯一的字符串，它可以代表一个账户、一个用户组、或者是一次登录。通常它还有一个SID固定列表，例如 Everyone这种已经内置的账户，默认拥有固定的SID。
SID的表现形式：

域SID-用户ID

计算机SID-用户ID

SID列表都会存储在域控的AD或者计算机本地账户数据库中。

5.5. Windows Access Token产生的过程

Windows Access Token产生的过程

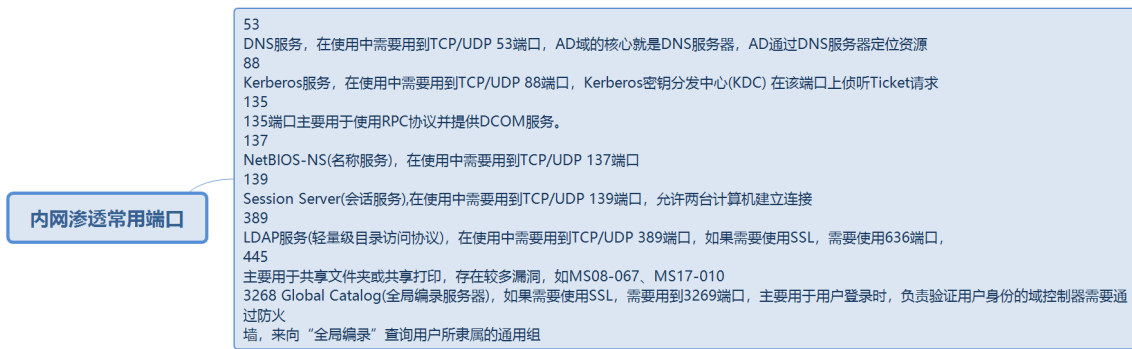
每个进程创建时都会根据登录会话权限由LSA(Local Security Authority)分配一个Token。如果CreateProcess时自己指定了Token，LSA会用该Token，否则就用父进程Token的一份拷贝。当用户注销后，系统将会使授权令牌切换为模拟令牌，不会将令牌清除，只有在重启机器后才会清除。

5.5.1. 每个进程创建时都会根据登录会话权限由LSA(Local Security Authority)分配一个Token。如果CreateProcess时自己指定了Token，LSA会用该Token，
否则就用父进程Token的一份拷贝。当用户注销后，系统将会使授权令牌切换为模拟令牌，不会将令牌清除，只有在重启机器后才会清除。

6. 其他



6.1. 内网渗透常用端口



6.1.1. 53

DNS服务, 在使用中需要用到TCP/UDP

53端口, AD域的核心就是DNS服务器, AD通过DNS服务器定位资源

88

Kerberos服务, 在使用中需要用到TCP/UDP

88端口, Kerberos密钥分发中心(KDC) 在该端口上侦听Ticket请求

135

135端口主要用于使用RPC协议并提供DCOM服务。

137

NetBIOS-NS(名称服务), 在使用中需要用到TCP/UDP 137端口

139

Session Server(会话服务),在使用中需要用到TCP/UDP

139端口, 允许两台计算机建立连接

389

LDAP服务(轻量级目录访问协议), 在使用中需要用到TCP/UDP

389端口，如果需要使用SSL，需要使用636端口，

445

主要用于共享文件夹或共享打印，存在较多漏洞，如MS08-067、MS17-010

3268 Global

Catalog(全局编录服务器)，如果需要使用SSL，需要用到3269端口，主要用于用户登录时，负责验证用户身份的域控制器需要通过防火墙，来向“全局编录”查询用户所隶属的通用组

6.2. <https://payloads.online/archivers/2018-11-30/1#0x03-windows-access-token>

<https://www.cnblogs.com/artech/archive/2011/01/24/kerberos.html>

<https://www.cnblogs.com/artech/archive/2011/01/25/NTLM.html>

6.3. . . .

7. 微信公众号 黑白天 BY 李木



7.1.