Kerberos简介
Posted on 2006-03-20 15:24 idior 阅读(60155) 评论(26) 编辑 收藏

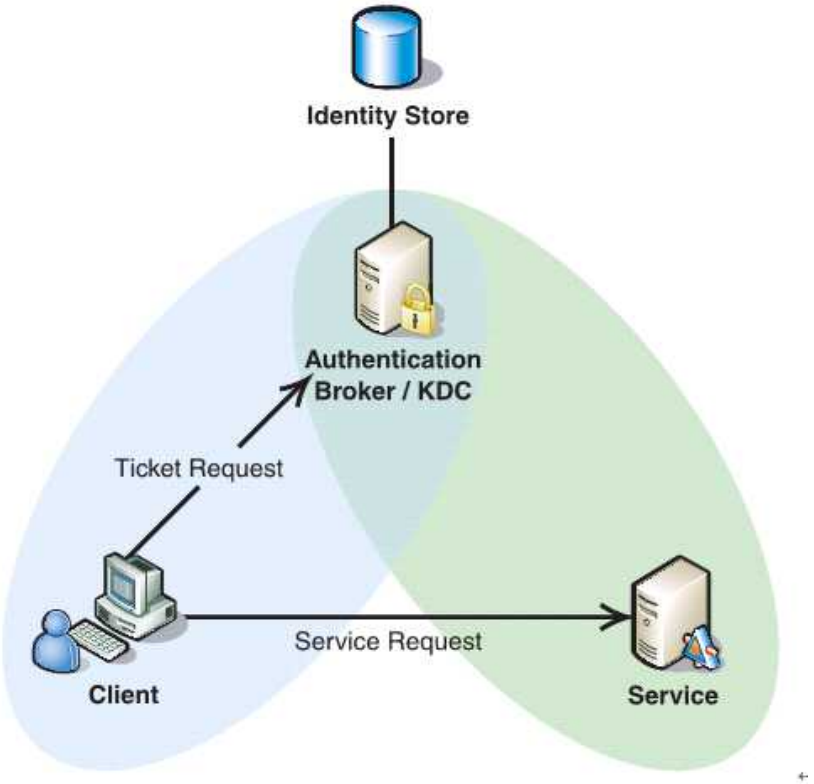
Kerberos协议：

Kerberos协议主要用于计算机网络的身份鉴别(Authentication), 其特点是用户只需输入一次身份验证信息就可以凭借此验证获得的票据(ticket-granting ticket)访问多个服务，即SSO(Single Sign On)。由于在每个Client和Service之间建立了共享密钥，使得该协议具有相当的安全性。

条件

先来看看Kerberos协议的前提条件：

如下图所示，Client与KDC， KDC与Service 在协议工作前已经有了各自的共享密钥，并且由于协议中的消息无法穿透防火墙，这些条件就限制了Kerberos协议往往用于一个组织的内部， 使其应用场景不同于X.509 PKI。



过程

Kerberos协议分为两个部分：

1．Client向KDC发送自己的身份信息，KDC从Ticket Granting Service得到TGT(ticket-granting ticket)， 并用协议开始前Client与KDC之间的密钥将TGT加密回复给Client。

此时只有真正的Client才能利用它与KDC之间的密钥将加密后的TGT解密，从而获得TGT。

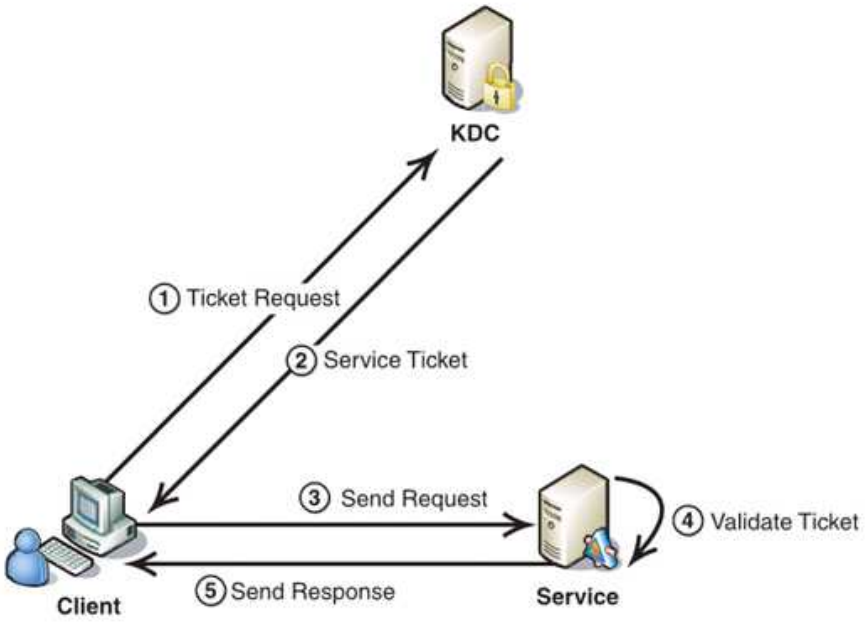
（此过程避免了Client直接向KDC发送密码，以求通过验证的不安全方式）

2. Client利用之前获得的TGT向KDC请求其他Service的Ticket，从而通过其他Service的身份鉴别。

Kerberos协议的重点在于第二部分，简介如下：

随笔分类(228)
.Net (42)
AOP(4)
Code Review(4)
DataBase(1)
DDD(1)
Design(43)
Distributed App(3)
infinoor box(20)
Lucene(7)
Misc(33)
O/R Mapping(7)
Orcas(3)
Security(2)
Sementic web(2)
SICP(3)
Software Engineer(2)
TDD(9)
Tool(4)
Web(8)
Web Services & SOA(10)
Windbey(16)
WPF(4)

随笔档案(213)
2013年7月 (1)
2013年6月 (4)
2010年6月 (8)
2010年5月 (2)
2007年11月 (1)
2007年9月 (2)
2007年8月 (1)
2007年7月 (2)
2007年3月 (2)
2007年1月 (3)
2006年12月 (1)
2006年11月 (1)
2006年10月 (3)
2006年9月 (3)
2006年8月 (2)
2006年7月 (3)
2006年6月 (6)
2006年5月 (8)
2006年4月 (1)
2006年3月 (3)
2006年2月 (10)
2006年1月 (15)
2005年12月 (2)
2005年10月 (3)
2005年9月 (7)
2005年8月 (18)
2005年7月 (8)
2005年6月 (1)
2005年5月 (3)
2005年4月 (18)
2005年3月 (17)
2005年2月 (25)
2005年1月 (18)



1. Client将之前获得TGT和要请求的服务信息(服务名等)发送给KDC，KDC中的Ticket Granting Service将为Client和Service之间生成一个Session Key用于Service对Client的身份鉴别。然后KDC将这个Session Key和用户名，用户地址（IP），服务名，有效期, 时间戳一起包装成一个Ticket(这些信息最终用于Service对Client的身份鉴别)发送给Service， 不过Kerberos协议并没有直接将Ticket发送给Service，而是通过Client转发给Service.所以有了第二步。

2. 此时KDC将刚才的Ticket转发给Client。由于这个Ticket是要给Service的，不能让Client看到，所以KDC用协议开始前KDC与Service之间的密钥将Ticket加密后再发送给Client。同时为了让Client和Service之间共享那个秘密(KDC在第一步为它们创建的Session Key)， KDC用Client与它之间的密钥将Session Key加密随加密的Ticket一起返回给Client。

3. 为了完成Ticket的传递，Client将刚才收到的Ticket转发到Service. 由于Client不知道KDC与Service之间的密钥，所以它无法算改Ticket中的信息。同时Client将收到的Session Key解密出来，然后将自己的用户名，用户地址（IP）打包成Authenticator用Session Key加密也发送给Service。

4. Service 收到Ticket后利用它与KDC之间的密钥将Ticket中的信息解密出来，从而获得Session Key和用户名，用户地址（IP），服务名，有效期。然后再用Session Key将Authenticator解密从而获得用户名，用户地址（IP）将其与之前Ticket中解密出来的用户名，用户地址（IP）做比较从而验证Client的身份。

5. 如果Service有返回结果，将其返回给Client。

总结

概括起来说Kerberos协议主要做了两件事

1. Ticket的安全传递。

2. Session Key的安全发布。

再加上时间戳的使用就很大程度上的保证了用户鉴别的安全性。并且利用Session Key，在通过鉴别之后Client和Service之间传递的消息也可以获得Confidentiality(机密性), Integrity(完整性)的保证。不过由于没有使用非对称密钥自然也就无法具有抗否认性，这也限制了它的应用。不过相对而言它比X.509 PKI的身份鉴别方式实施起来要简单多了。

推荐资料：

Kerberos的原理

Kerberos: An **Authentication Service for Computer Networks**

Web Services Security系列文章

分类: Security

2004年12月 (11)

文章档案(45)
2007年11月 (1)
2007年8月 (4)
2006年12月 (1)
2006年9月 (1)
2006年8月 (1)
2006年6月 (5)
2006年5月 (4)
2006年4月 (6)
2006年3月 (2)
2006年1月 (1)
2005年9月 (2)
2005年8月 (1)
2005年7月 (2)
2005年4月 (1)
2005年3月 (4)
2005年2月 (2)
2005年1月 (7)

相册(11)
myself(11)

Friends
alittlefish
Artech
cc
DuDu
RobinZhong
Samuel
Study English
wayfarer
古惑狼
吕震宇
维生素C.NET

Resource
Alphatom
Artima Forums
Book Store
ChannelDispatcher
CodeBetter Blog
CodeProject
Control templates in WPF
Date Format
Dive in Python
Domain Driven
ErrorBank
Get running instance of VS
HTTP Made Really Easy
Jason Haley (interesting thing)
Message Pump
PPT FAQ
ShortCuts of Word

idior
关注 - 1
粉丝 - 262
荣誉：推荐博客
+加关注

1 0

(请您对文章做出评价)

« 上一篇：重构－－－Who are you?!
» 下一篇：RhinoMock2 续(TDD Mock)

Feedback

#1楼
2006-03-20 17:28 by U2U

在《Distributed Programming》一书中也看过，顶一下

支持(0) 反对(0)

#2楼
2006-12-16 19:50 by ss[匿名]

有些地方翻译得比较差,明显译者在翻完没有认真审核,不过能翻过来共享也算相当不错了

支持(0) 反对(0)

#3楼[楼主]
2006-12-16 22:43 by idior

@ss[匿名]
这不是翻译的，汗 -_-!!我的语文这么差？

支持(0) 反对(0)

#4楼
2007-02-28 15:37 by aaa

谢谢，不错，我看懂了

支持(0) 反对(0)

#5楼
2007-03-15 15:07 by Y-Aries

很有帮助，感谢共享！
PS：ss很不厚道的说

支持(0) 反对(0)

#6楼
2007-04-26 13:24 by test

KDC
k1| \k2
client--service
k3

总而言之有3个session key，KDC利用k1 和k2发布了session key k3。

支持(0) 反对(0)

#7楼
2007-05-24 16:00 by kevin

smart software
SOA Biztalk
Spoken English
String Format
Terms Trans
WCF Extension
WCF Runtime
Word FAQ
xpah
江南白衣@ITO
商业意识
数码产品

SOA
Advanced Web Services
stateful webservice
WCF Samples
WCF Wiki
Web Service Factory
Web Services Security(ms)
WS-Addressing
WSE 3.0 Class Lib

Util
Box
code format
CopySourceAsHtml
Email Icon
FaveCave
VB2C#

积分与排名
积分 - 1023173
排名 - 47

最新评论
1. Re:质疑国内.Net社区
2015-11-24
--缘续成
2. Re:Kerberos简介
翻译的很好，感谢分享
--旧收音机
3. Re:Kerberos简介
很好，学习了。谢谢！！
--hbg-rohens
4. Re:RhinoMock2 续
it's crazy！这篇文章还是06年的，年代久远啊！写的不错，到处了rhino.mock 很多重点的东西！
--unbreakable
5. Re:Delegate比较全面的例子(原创)
好文，循序善诱！
--wzStyle
6. Re:Remoting基本原理及其扩展机制（上）
Mark

个人认为，写的非常简练。谢谢了

支持(0) 反对(0)

#8楼
2007-05-26 23:32 by Silent Void

"Client与KDC，KDC与Service在协议工作前已经有了各自的共享密钥"

请问它们之间是如何协商共享密钥的？有没有相关的参考资料？Thanks:)

支持(0) 反对(0)

#9楼
2007-05-27 12:17 by idior

文中已有概述,并且在文章最后我有给出推荐资料.

支持(0) 反对(0)

#10楼
2007-05-28 00:56 by baoni

楼主写的不错呢。是认真学习的好孩子。：）

支持(0) 反对(0)

#11楼
2007-07-02 21:25 by robin5475

非常感谢
我们要考信息安全了~多亏了这篇文章

支持(0) 反对(0)

#12楼
2007-07-18 15:15 by ld

清晰透彻！

支持(0) 反对(0)

#13楼
2007-08-15 14:28 by fangsang

不错！值得推荐

支持(0) 反对(0)

#14楼
2007-09-20 17:19 by coolzhang001

感谢

支持(0) 反对(0)

#15楼
2007-09-21 14:10 by road

write very well.

--VAllen

7. Re:Kerberos简介

文章浅显易懂，把最主要的要点都讲到了，大赞一个！！

--elf_tech

8. Re:Kerberos简介

首先，用户使用客户机（用户自己的机器）上的程序登录：用户输入用户ID和密码到客户机。客户机程序运行一个单向函数（大多数为杂凑）把密码转换成密钥，这个就是客户机（用户）的"用户密钥"（K_client）.....

--huokong

9. Re:Kerberos简介

写的很精炼！！

--xfile

10. Re:你了解创建者模式了吗? --- 创建者模式详解

@nx这个本文里头没有涉及到吧...

--谁说我不是会员

支持(0) 反对(0)

#16楼
2008-01-08 12:02 by fuhongwei041

很好...

支持(0) 反对(0)

#17楼
2008-11-07 15:45 by 皇上爱累了

--引用----------
ss[匿名]: 有些地方翻译得比较差,明显译者在翻完没有认真审核,不过能翻过来共享也算相当不错了

--引用----------
idior: @ss[匿名]

这不是翻译的，汗 -_-!!我的语文这么差?

哈哈!这段真搞笑

支持(0) 反对(0)

#18楼
2009-12-10 16:04 by 222fgewrf

2sdfsfsfsdfsfsdfsdfsdf

#19楼
2009-12-10 20:31 by coodoing

我觉得重要的是理论应用于实践。。。。

支持(0) 反对(0)

#20楼
2009-12-22 13:13 by gzpenghaifeng

这是我见过的最清晰易懂的解释，多谢楼主！
Kerberos标准协议描述的文档多数只是涉及到第二步的内容，看这篇算是把疑虑搞通了。只是又问：那Service和KDC之间的共享密钥又是如何传递的呢？再请教了。:)

另外，不知是否已经有基于Kerberos的通用的SSO解决方案包？

#21楼
2009-12-22 13:22 by IDIOR_ONLINE

它们是在一个信任域中的，密钥共享不存在什么太大的安全问题。

#22楼
2014-06-24 21:17 by xfile

写的很精炼！！

支持(0) 反对(0)

12. 征集译者(27)
13. Kerberos简介(26)
14. .Net给我们带来了什么？(25)
15. Delegate和Command Pattern(24)

推荐排行榜

1. Remoting基本原理及其扩展机制（上）(4)
2. Covariance and Contravariance(4)
3. How does ElementName Binding work? - Part 1 Logical Tree & NameScope(4)
4. Webservice 的设计和模式(3)
5. Lambda表达式的应用(2)
6. How does ElementName Binding work – Part 3 InheritanceContext(2)
7. Memory leak caused by EventHandle - weak event(2)
8. O/R Mapping 基本概念(欢迎指正)(2)
9. 你了解创建者模式了吗? --- 创建者模式详解(2)
10. Transaction in ADO.net 2.0(2)
11. Web Services Security(2)
12. webservice Quiz（Wsdl & Soap）(1)
13. Enterprise Test Driven Develop (1)
14. Rhino Mocks (RhinoMock)2(1)
15. .Net2.0 的新线程 ParameterizedThreadStart &BackgroundWorker(1)

#23楼
2014-06-25 19:11 by huokong

首先，用户使用客户机（用户自己的机器）上的程序登录：

用户输入用户ID和密码到客户机。
客户机程序运行一个单向函数（大多数为杂凑）把密码转换成密钥，这个就是客户机（用户）的"用户密钥"（K_client）。受信任的AS通过某些安全的途径也获取了与此密钥相同的密钥。

支持(0) 反对(0)

#24楼
2014-06-29 10:33 by elf_tech

文章浅显易懂，把最主要的要点都讲到了，大赞一个！！

支持(0) 反对(0)

#25楼
2015-08-26 14:11 by hbg-rohens

很好，学习了。谢谢！！

支持(0) 反对(0)

#26楼
2015-10-09 14:49 by 旧收音机

翻译的很好，感谢分享

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部