用 KERBEROS 实现网络计算的安全认证

李成斌　熊华平　刘万伟　（大庆勘探开发研究院　163712）

摘要：Kerberos 是为分布式计算应用提供基于密钥加密的强度认证协议。通信双方通过 kerberos 相互认证身份，且其通信内容可以采用加密方式传输。本文介绍了 Kerberos 的原理，提出了 Kerberos 的实施过程和配置方法。

关键词：Kerberos　认证　配置

一、引言

在分布式计算环境中，服务器主要是 UNIX 系统。从操作系统本身的安全性能来讲，多用户的操作系统都是符合 C2 级安全规范。由于操作系统对用户的认证普遍采用口令认证，当在网上存取系统时，口令直接以 ASCII 码的形式在网上广播，非法用户通过截取用户口令就可对系统进行非法侵入；同时，也存在非法用户对通信网络的数据进行数据截取及数据操作的可能。这种威胁对系统数据和系统本身来说都是致命的，往往造成系统的严重破坏。因此，在分布式计算环境下系统面临的重大安全隐患是如何对用户实现有效的认证和访问服务的控制。

目前，在远程访问控制上主要采取两种技术，一是防火墙技术，它控制从外网对内网的存取，但解决不了内网用户的非法侵入。事实上，内网用户往往是主要的非法侵入者。二是 Kerberos 技术。Kerberos 是由 MIT 开发，为分布式计算应用提供基于密钥加密的强度认证协议。通信双方通过 kerberos 相互认证身份，其通信内容也可以采用加密方式传输。

二、Kerberos 的工作原理

Kerberos 针对非法用户的网上窃听或伪造，通过认证服务器（AS）和入场券授予服务（TGS）采取认证信息加密和规定认证有效期的措施，确保认证信息不被窃听或伪造后重发。KERBEROS 认证的过程如下：

1. 用户（Client）想申请向 TGS 访问的 TGS 入场卷 TGT，先向 AS 发送报文。

as－req＝c，tgs，expiry，n。

2. AS 在 Kerberos 数据库中查找用户的口令字对应的散列函数值，并用之构成 K_c，用 K_c 加密回应信息给用户。回应信息中还包含密钥 $K_{c,tgs}$（用于 client 和 TGS 交互的），并用 K_{tgs} 加密令牌 $T_{c,tgs}$。

as－resp＝$K_c(K_{c,tgs})$，$K_{tgs}(T_{c,tgs})$。

其中 $T_{c,tgs}$＝(c，tgs，tstmp，expiry，$K_{c,tgs}$)。

用户使用根据口令字计算的密钥 K_c，解密从 AS 接收的报文，得到 $K_{c,tgs}$，并比较报文中的 chksm 和自己计算的 chksm。如一致，则认为用户提供给 AS 的口令正确的。

3. client 向 TGS 发报文

tgs－req＝s，expiry，n，$K_{tgs}(T_{c,tgs})$，$K_{c,tgs}$(AC)

其中 AC＝(c，IPc，tstmp，chksm)。TGS 用 K_{tgs} 解密 $T_{c,tgs}$，并进行校验保证其完整性。TGS 从 $T_{c,tgs}$ 中取出 $K_{c,tgs}$ 用它解密得到 AC。

4. TGS 向 client 发送用 $K_{c,tgs}$ 加密的报文，给出会话密钥 $K_{c,s}$。

tgs－resp＝$K_{c,tgs}(K_{c,s})$，$K_s(T_{c,s})$

其中 $T_{c,s}$＝(c，s，IP_c，tstmp，chksm，expiry，$K_{c,s}$)。

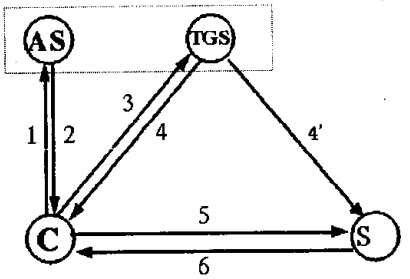
5. client 向服务器 s 发送 AC，和从 TGS 获得的加密的服务入场卷 $T_{c,s}$。

s－req＝$K_s(T_{c,s})$，$K_{c,s}$(AC)

6. 当双方验证均要求时，服务器 s 向 client 发从 $T_{c,s}$ 中取出的 tstmp 信息，并将值＋1，用 $K_{c,s}$ 加密返回。如服务器 s 不能产生 K_s（K_s 基于服务器 s 的口令，为 s 和 TGS 共享），则说明是顶替者。

s－resp＝$K_{c,s}$(tstmp＋1)。

其中，tstmp 为时间片，chsum 为检测和，K 为密钥，T 代表入场卷，c 代表 client，tgs 代表 TGS 服务器，n 是随机数，expiry 是密钥生存期限，s 是应用服务器。



上图显示了完整的 kerberos 认证过程。1 和 2 只发

生在用户首次登入时;3 和 4 在每次用户向一个新的应用系统请求服务时都建立这种连接,4'与 4 同时发生,封装的内容包括用户名和主机名;信息 5 仅用于向应用服务器认证自己的身份;信息 6 只有当用户请求对应用服务器实行相互认证时出现。

三、Kerberos 的实施

1. 设置认证服务器

由于 Kerberos 认证服务器要维护一个场地用户的口令(加密密钥)数据库,因此,认证服务器应安装在一个相对安全可靠的机器上。在可能的情况下,认证服务器应用于专门运行认证服务,且限制可存取的用户数量。

在初始化时,用户应合理地选择口令,因为 Kerberos 没有提供有效抵御口令猜测攻击的手段。

2. Kerberos 实用程序

以下几个实用程序用于用户入场券的管理,必须安装在用户工作站上。

(1)Kinit:用于申请 TGS 入场券

(2)Kdestroy:删除入场券

(3)Klist:列出入场券

(4)Kpasswd:修改用户 Kerberos 口令

为方便用户的使用,通常的做法是把 kinit 和 kdestroy 分别放到用户登录程序和用户退出程序中,使申请和删除入场券的操作自动进行,一方面避免用户输入两次口令,另一方面根除用户忘记删除入场券造成的隐患。

另外,为防止"特洛伊木马"的侵入,要把上述实用程序放在较为安全的路径下,减小非法用户通过改写 kinit 程序俘获用户口令的可能。

3. 支持 Kerberos 认证的应用及开发

网络应用要使用 kerberos 认证,必须经过重新开发或修改。MIT 提出的参考实施方案中包括较流行的 rcp/rsh/rlogin/ftp/telnet 的 kerberos 版本。

(1) 直接支持 Kerberos 的应用

①rsh, rlogin, and rcp。在分布时计算环境中,为方便使用,这三个以 r 开头的命令在 UNIX 中常配置成"可信任"的状态,即在目标机中由.rhosts 文件说明"可信任"的用户或主机,这样在进行 r 命令操作时就不必输入用户口令。显然,提供方便的同时却留下了安全隐患。对支持 Kerberos 认证的三种服务,用户可以选择加密传输或非加密传输两种方式。

②telnet 和 ftp。由于 telnetd 总是采用同一个端口进行通信,在/etc/inetd.conf 文件的配置时应采用"-a user"选项,使当 kerberos 认证失败时可执行普通的 telnet 命令。ftp 服务与 telnet 服务的工作方式类似。

③pop。POP 用于从邮件服务器提取邮件。使用'-k'选项运行该协议的 kerberos 版本。该服务仅仅在邮件服务器上运行。

(2)支持 Kerberos 的应用程序的开发。显然,支持 kerberos 认证的应用应具有两个方面的基本功能:

①应用程序(客户端)必须能够产生并发送 kerberos 应用请求给应用服务器(服务端);

②应用服务器应能够验证认证信息。

kerberos 提供了产生和验证认证信息的例程库 KAPI,它是一个完全基于 kerberos 认证机制的编程接口。另外,最新的 kerberos 实现中提供了通用安全服务应用编程接口(GSSAPI)。GSSAPI 是一个独立于认证机制的编程接口,开发人员可以使用包括 kerberos 在内的多种不同的认证技术。当然,GSSAPI 并不支持全部 kerberos 提供的功能特性,如用户到用户之间的认证。

4. 跨域(cross-realms)认证

在规模较大的分布式计算环境中,存在跨越多个组织(或应用)边界的情况,这时就应考虑规划多个认证服务器。当存在多个认证服务器时,每个认证服务器对应一个用户或应用服务器的子集,称为管理域(realms)。跨域认证允许用户向注册在不同域中的应用服务器证明其身份。其认证过程如下:

为了向远程域中的应用服务器认证自己,用户首先向本地域认证服务器申请一个远程域的 TGT,这就要求本地认证服务器与应用服务器所在域的认证服务器共享一个跨域密钥。接着用户用该 TGT 向应用服务器对应域的认证服务器申请应用服务器的入场券。

四、Kerberos 的配置

下面结合 eBones Kerberos V4 说明 kerberos 的配置过程。

1. Kerberos 服务器的配置

服务器上运行 Kerberos 服务程序。Kerberos 服务器实现了 Kerberos 技术中 AS 和 TGT 的功能。

设所用的机器名是 server1,域名是 dq.cnpc.com.cn,安装目录是/usr/athena/bin。

(1) 管理域(realm)的配置。Kerberos 的管理域以大写字母表示,如果没有特殊原因,管理域通常由 Internet 的域名组成,相同管理域下的机器使用同一个认证服务器。配置文件/etc/krb.conf 和 /etc/krb.realms 与管理域的配置紧密相关。

①krb.conf 文件决定哪台机器服务于哪个管理域,文件的格式如下:

THIS.REALM

THIS. REALM kerberos. this. realm admin server

THIS. REALM kerberos－1. this. realm

ANOTHER. REALM kerveros. another. realm

②krb. realms 指出特定的机器属于哪个管理域。文件的格式如下：

client1 THIS. REALM

. this. realm THIS. REALM

当然，应修改/etc/services 文件，确定 Kerberos 服务的端口。

(2) 建立 Kerberos 数据库。以 root 用户登录，在环境变量 PATH 中，加入/usr/athena/bin 和/usr/athena/sbin 两条路径。运行 kdb-init 命令，给出管理域(如 DQ. CNPC. COM. CN)及 kerberos 口令。这个口令用来加密 kerberos 数据库。

(3) 运行 kstash 命令，把数据库密钥放入密钥缓冲文件/. k 中

server# kstash

Enter Kerberos master password : <password>

Curent Kerberos master key version is 1.

Master key entered. BEWARE!

Wrote master key to /. k

(4) 添加管理用户(principal)。在服务器上使用 kdb-edit 命令，编辑数据库。该命令执行结果是把用户名(Principal，如 itisme)、类型(Instance，如 admin)、用户 Kerberos 口令、有效日期及入场券最大有效时间等信息写入数据库。该命令是一个循环的过程，一次可以建立多个用户。

(5) 启动服务器进程。在后台运行 kerberos 进程。

Server1# /usr/athena/libexec/kerberos &

.....

(6) 为管理用户赋予权限。用户的添加、读取、修改、删除权限分别由 admin-acl. add、admin-adl. get、admin. acl. mod 和 admin-acl. del 四个文件给出。例如为 itisme. admin 分配添加权限时，只要把 itisme. admin@YJY. DQ. CNPC. COM. CN 放在文件/var/kerberos/admin-acl. get 中即可。

(7) 启动管理进程。在服务器上，以后台方式启动管理进程/usr/athena/libexec/kadmind 。

2. kerberos 客户机的配置

(1) 修改配置文件。编辑/etc/krb. conf 和/etc/krb. realms 文件，并修改/etc/services 文件。

(2) 验证客户机的时钟。客户机的时钟要和服务器的时钟始终保持同步，最大允许误差 5 分钟，最好采用时钟自同步方法(NTP)，使网络上的机器自动对时。

(3) 修改/etc/inetd. conf 文件。确定那些服务需要进行认证，并修改/etc/inetd. conf 文件。

(4) 为每种服务创建认证。像每个用户都需要在认证服务器上注册一样，每种服务也需要在服务器上进行认证，使应用服务与 kerberos 服务器之间产生一个共享的密钥。rcp/rsh/rlogin/ftp/telnet 远程访问服务建立的注册名的名字是'rcmd. hostname'

client1# ksrvutil －p itisme. admin get

Name [rcmd] : <>

Instance [jsz1] : <>

Realm [DQ. CNPC. COM. CN] : <>

Is this correct ? (y,n) [y] : <>

Add more keys ? (y,n) [n] : <>

Password for itisme. admin@DQ. CNPC. COM. CN < itisme. admin's password>

Written rcmd. jsz1

五、结束语

由于 Kerberos 系统具有安全性高、透明性高和可扩展性好的特点，因此在分布式计算环境中得到了广泛的应用。但是，在使用中应注意几个问题。(1)Kerberos 服务器与用户共享的秘密是用户的口令字，服务器在回应时不验证用户的真实性，假设只有合法用户拥有口令字。如攻击者记录申请回答报文，就易形成代码本攻击。(2) AS 和 TGS 是集中式管理，容易形成瓶颈，系统的性能和安全也严重依赖于 AS 和 TGS 的性能和安全。(3)随用户数增加，密钥管理较复杂。Kerberos 拥有每个用户的口令字的散列值，AS 与 TGS 负责用户间通信密钥的分配。当 N 个用户想同时通信时，仍需要 N*(N－1)/2 个密钥。

参考文献

[1] B. Clifford Neuman and Theodore Ts'o. Kerberos: An Authentication Service for Computer Networks, IEEE Communications, 32(9):33－38. September 1994

[2] John T. Kohl, B. Clifford Neuman, and Theodore Y. T'so, The Evolution of the Kerberos Authentication System. In Distributed Open Systems, pages 78－94. IEEE Computer Society Press, 1994.

[3] 《计算机世界》1997 年第 40 期

(来稿时间:1999 年 4 月)