



OpenSSH on Linux using Windows/Kerberos for Authentication

Published by the Open Source Software Lab at Microsoft. October 2007.

Special thanks to Chris Travers, Contributing Author to the Open Source Software Lab. Most current version will be maintained at <http://port25.technet.com>.



Abstract:

Secure remote access to UNIX and Linux systems is generally accomplished through SSH. The most frequent implementation of that protocol is OpenSSH, originally written for the OpenBSD project but now ported to a wide variety of platforms. This paper will show how to use OpenSSH with the Kerberos portion of Active Directory to automate authentication.

Information in this document, including URL and other Internet Web site references, is subject to change without notice and is provided for informational purposes only. The entire risk of the use or results from the use of this document remains with the user, and Microsoft Corporation makes no warranties, either express or implied. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

© 2008 Microsoft Corporation. This work is licensed under the Microsoft Public License. The Microsoft Public License is [available here](#).

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Microsoft, Windows, Windows XP, Windows Server, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are property of their respective owners.

1 Introduction

Secure remote access to UNIX and Linux systems is generally accomplished through SSH. The most frequent implementation of that protocol is OpenSSH, originally written for the OpenBSD project but now ported to a wide variety of platforms. This paper will show how to use OpenSSH with the Kerberos portion of Active Directory to automate authentication.

Identity management is beyond the scope of this paper. Kerberos provides authentication but not identity management. If OpenSSH cannot find an identity relating to the login credentials, access will be denied.

On the Linux system, I have installed the following tools (Package names from Fedora Core 5):

- openssh
- openssh-server
- samba-common
- samba-client
- krb5-workstation
- krb5-libs

On the Windows side, I installed the following tools:

- Windows Support Tools

In Windows Server 2003, the Support Tools are installed by running the program: D:\Support\Tools\supinst.msi (where D is the CDROM drive with the original Windows Server install disc). In addition, the Windows system was set up as a domain controller for the krbtest.local domain.

Also, note that the Administrator account was associated with the Kerberos realm using the User Manager tool on the Windows system.

1.1 Linux Configuration

In general, most aspects of the Linux configuration are best done first. All of the following actions are done with the superuser account (root).

1.1.1 Kerberos Setup

I added the following lines to the “[libdefaults]” section of the /etc/krb5.conf:

```
default_realm = KRBTEST.LOCAL
default_keytab_name = FILE:/etc/krb5.keytab
```

And the following lines to the “[realms]” section:

```
KRBTEST.LOCAL = {
```

```
kdc = kdc1.krbtest.local1:88
default_domain = krbtest.local
}
```

1.1.2 Samba Setup

The following lines were added to the `/etc/samba/smb.conf`² and are necessary for properly joining a domain and managing keytabs:

```
netbios name = chrislt
realm = KRBTEST.LOCAL
security = ADS
encrypt passwords = yes
password server = kdc1.krbtest.local
workgroup = KRBTEST
use kerberos keytab = yes
```

1.1.3 Joining the Domain

Joining the domain is simple:

```
bash# net ads join -U Administrator
```

This should add the machine entry (chrislt) to Active Directory, and create the Kerberos security principal. Active Directory will typically return a message indicating there was a successful join. If this does not work, the following troubleshooting steps may be of help:

1. Make sure the system clocks are synchronized between the two systems.
2. Try kinit Administrator. Enter the Administrator password when prompted. If you do not get any errors, then Kerberos is working.
3. Make sure your `resolv.conf`³ is pointing at the Active Directory domain controller as its DNS server.

1.1.4 Creating the Keytab

The command to create the [keytab](#) using the Samba tools is:

```
bash# net ads keytab create
```

This should create a keytab with 9 entries. At this point you can test this by using the following command:

```
bash# kinit -k -t /etc/krb5.keytab 'chrislt$'
```

If this command returns without error, it means that the host key is now working. Note that unlike Apache, OpenSSH will only use the Kerberos principal of `host/fqdn@REALM`. In this case, that resolves to `host/chrislt.krbtest.local@KRBTEST.LOCAL`. At the moment, if you try the following, you will get an error:

¹ Local was chosen as the default local host name. You may need to change this value to match your environment.

² This configuration file may also be found as `/etc/smb.conf` and `/usr/local/etc/smb.conf` depending on your installation of Samba.

³ The location of this file may vary depending on your Linux distribution.

```
bash# kinit -k -t /etc/krb5.keytab 'host/chrislt.krbtest.local'  
Error: Client not found in Kerberos database
```

Indeed this will not work until further steps are taken.

1.1.5 *Configuring SSH*

Make sure the following lines exist in your `/etc/ssh/sshd_config`⁴ file:

```
KerberosAuthentication yes  
KerberosOrLocalPasswd yes  
KerberosTicketCleanup yes  
GSSAPIAuthentication yes  
GSSAPICleanupCredentials yes
```

The Kerberos options allow users without Kerberos credentials to log in and get a ticket by presenting the appropriate password. These are also helpful when dealing with ssh clients that don't support Kerberos authentication. Unfortunately many of the more popular SSH clients for Windows do not support GSSAPI authentication.

The GSSAPI options allow a user with a Kerberos TGT to log in by presenting that TGT in lieu of a password or public key exchange.

1.2 **Windows Configuration**

Active Directory is LDAP-centric. Authentication accounts are stored first and foremost in the directory, and only user and computer accounts can have authentication information connected to them. For this reason, it is important to create a mapping between the host principal we need to use and the machine account. All of the actions below are done as an administrator.

1.2.1 *Creating the Host Principal*

From the command line (a CMD.EXE for example), use `ktpass` to create the host principal and map it to the machine account. Note that the command below is all one line.

```
ktpass /princ host/chrislt.krbtest.local@KRBTEST.LOCAL /mapuser  
chrislt$\KRBTEST
```

1.2.2 *A note on SNTP, NTP, and Clock Skew*

Kerberos requires that all system clocks involved are within a certain configurable range of each other. The default range is 5 minutes. If the clocks are different by a time greater than the configurable range then users will be unable to authenticate; therefore computers cannot join the domain.

When a Windows host joins the domain, it begins to synchronize its clock with the domain controller using `sntp`. The `sntp` client in a Windows domain controller cannot be configured to use an external source for time. Therefore, it is a common recommendation that a separate NTP client be installed so that all clocks can be authoritative. Failure to do so may cause a great deal

⁴ On some systems this file may be found as `/usr/local/etc/sshd_config`.
<http://port25.technet.com>

of frustration in troubleshooting the domain joining process because it is possible for the clocks to accumulate skew.

1.3 Trying It Out

Suppose you have an account stored locally named `chris` and a domain account with the same name, request a TGT with the `kinit` command and enter the password when prompted:

```
bash$ kinit chris
password for chris@KRBTEST.LOCAL:
```

ssh to the local 'chris' account:

```
bash$ ssh chris@chrislt.krbtest.local
```

Assuming you are not presented with an error message or a password prompt when connecting with `ssh`, this process has been successful.

1.4 Final Thoughts

Prior to Kerberos support for SSH, most large networks were using Kerberos to authenticate Telnet users and encrypt sessions. Although these configurations provide many of the benefits of the setup described above, the system was more complex to administer and, when improperly configured, less secure. Since many administrators currently use SSH for remote access management, utilizing Kerberos in this way allows an administrator to standardize to one remote access tool and centralize all authentication information.

One caveat I will offer is that Active Directory maps service principals to other account types. For this reason, various services on a system may be forced to share a key. If the keytab is compromised, an administrator must then assume that the machine account has been compromised and that an attacker may be able to spoof arbitrary services from that machine. While SSH offers some additional protection against these attacks, these protections rely on users being alert and reporting troublesome error messages to the IT staff.

If a machine account is compromised, the best action (in addition to other security measures depending on the type and extent of the compromise) is to delete the machine account, rejoin the domain, recreate the keytab, host service principals, etc.

1.5 About the Author

Chris Travers is the owner of Metatron Technology Consulting, a business devoted to helping people and businesses leverage open source software. He has extensive experience with OpenSSH, MIT Kerberos, and Active Directory.