



Linux/Windows Security Interop: Apache with mod_auth_kerb and Windows Server 2003 R2

Published by the Open Source Software Lab at Microsoft. January 2008.

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:

The Apache authentication module mod_auth_kerb allows Apache to authenticate users against a Kerberos KDC including one from ActiveDirectory. Kerberos itself can be fairly complex to set up. This guide will attempt to show the specific steps required to make this possible as well as discuss security limitations specific to the interoperability matters. This guide assumes a basic understanding of Kerberos V and that the Active Directory domain controller is properly configured prior to starting this process.

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.

© 2007 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

The Apache authentication module `mod_auth_kerb` allows Apache to authenticate users against a Kerberos KDC including one from ActiveDirectory. Kerberos itself can be fairly complex to set up. This guide will attempt to show the specific steps required to make this possible as well as discuss security limitations specific to the interoperability matters. This guide assumes a basic understanding of Kerberos V and that the Active Directory domain controller is properly configured prior to starting this process.

The test environment consisted of a Fedora Core 5 Linux system running Apache 2.2.x and `mod_auth_kerb` authenticating against a Windows Server 2003 R2 domain controller. The basic steps should form an outline for working with other distributions but package names and other minor details may require slight adjustment.

In addition to being a fairly useful topic in itself, this paper also provides a basis for what exactly is involved in getting Linux server software, which supports Kerberos, to work Active Directory if no local authentication or identity management is required.

I found that the following packages were required on the Linux system (names from Fedora Core 5, other distros may change names slightly):

- `krb5-libs-1.4.3-5.3`
- `krb5-workstation-1.4.3-5.3`
- `samba-client-3.0.23c-1.fc5`
- `samba-common-3.0.23c-1.fc5`
- `openldap-clients-2.3.19-4`
- `openldap-2.3.19-4`

Also, note that this has not been tested with any Kerberos implementations on the Linux side other than the MIT version.

On the Windows side, I named my domain `krbtest.local` and the domain controller `kdc1`. I set up Active Directory and DNS, and I installed the Windows support tools from the CDROM (from `D:\Support\Tools\subtools.msi` where `D:` is the CDROM drive. While I also installed SUA¹, I am not using any of the features of that component and therefore that step is optional. The support tools, while not used in this paper, provided valuable information about how Active Directory works, and hence may be helpful in troubleshooting issues. Although not strictly required, I would recommend installing them for such an environment.

1.1 Joining the Linux Server to the Domain using `samba-client`

Joining an Active Directory domain is done in three stages. First a Kerberos TGT² is requested for the joining user. Secondly, the joining user logs into the LDAP³ directory using the TGT and creates the LDAP directory entry. Third, the "computer account" is activated (and the service

¹ Subsystem for Unix-based Applications

² Ticket Granting Ticket

³ Lightweight Directory Access Protocol

<http://port25.technet.com>

principle added to the Kerberos database). Problems can occur at any stage and so it is often important to recognize that some degree of trial and error may be required to make this work.

1.1.1 Configuration Files Required to Join the Domain

First, I had to configure the realm in the `/etc/krb5.conf`. The realm was added and properly configured. My domain was named `krbtest.local`, so the relevant realm name was `KRBTEST.LOCAL`.

The following lines were added to the `[realms]` section:

```
KRBTEST.LOCAL = {  
    kdc = kdc1.krbtest.local:88  
    default_domain = krbtest.local  
}
```

Next, I added the following lines to the `[global]` section of the `/etc/samba/smb.conf`:

```
netbios name = chrislt  
realm = KRBTEST.LOCAL  
security = ADS  
encrypt passwords = yes  
password server = kdc1.krbtest.local  
  
# workgroup = NT-Domain-Name or Workgroup-Name, eg: MIDEARTH  
workgroup = KRBTEST
```

Note that it is possible, though more complex, to join the domain without resorting to Samba tools. However, I have generally found that this represents the easiest way of automating the setup once the configuration issues are taken care of.

1.1.2 A note on SNTP, NTP, and Clock Skew

Kerberos requires that all clocks involved are within a certain configurable range of each other. The default is 5 minutes. If the clocks are off greater than the configured 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 SNTP⁴. The SNTP client in a Windows domain controller cannot be also configured to use an external source for time. It is therefore 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 of frustration in troubleshooting the domain joining process as it is possible for the clocks accumulate skew.

1.1.3 Testing the Kerberos Setup

In general, I have found that the standard MIT Kerberos libraries provide better error reporting than either Windows or Samba. Therefore, it is useful to try to authenticate first using `kinit`. The syntax is:

⁴ Simple Network Time Protocol
<http://port25.technet.com>

```
kinit username@REALM
```

For example, to request a Kerberos TGT for the Administrator account for the domain above, use:

```
kinit Administrator@KRBTEST.LOCAL
```

You should be prompted for the password and once you enter it, you should get no further errors. You can verify that you have received the TGT by using the `klist` command.

1.1.4 Joining the Domain

Once you have verified that Kerberos is working, you can join the domain with the following command on the Linux system:

```
net join -U Administrator
```

The following diagnostics may be helpful in troubleshooting errors:

1. Can you get a TGT using `kinit`?
2. Can the `host` command pull the proper IP for the server?

1.2 Creating the Keytab

I used the following command to generate a keytab from ActiveDirectory:

```
bash# net ads keytab create -U Administrator
```

This step gave me a lot of trouble. One of the issues is that the `default_keytab_name` line in the `krb5.conf` must be set up to use the following syntax: `FILE:/path/to/file`

Once my `krb5.conf` contained the following line, I had no further issues:

```
default_keytab_name = FILE:/etc/krb5.keytab
```

1.3 Configuring Apache

Fedora Core includes all files in `/etc/httpd/conf.d`⁵ in its main configuration. When `mod_auth_kerb` is installed, it adds a file `auth_kerb.conf` to that directory. All that is needed is to adjust the directives accordingly and uncomment them. My file, which requires authentication for the entire installation includes the following directives:

```
LoadModule auth_kerb_module modules/mod_auth_kerb.so
<Location />
  AuthType Kerberos
  AuthName "Kerberos Login"
  KrbMethodNegotiate On
  KrbMethodK5Passwd Off
  KrbAuthRealms KRBTEST.LOCAL
  Krb5KeyTab /etc/httpd/conf/keytab
  require valid-user
</Location>
```

⁵ The location of this configuration file will vary with different Linux and Unix distributions
<http://port25.technet.com>

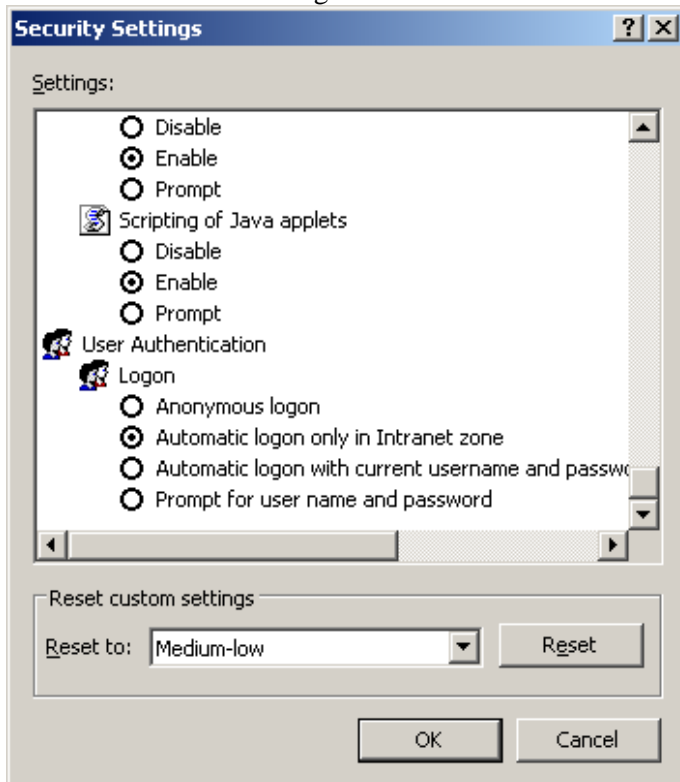
Other distributions may need to add similar provisions in the httpd.conf.

1.4 Configuring Firefox and Internet Explorer

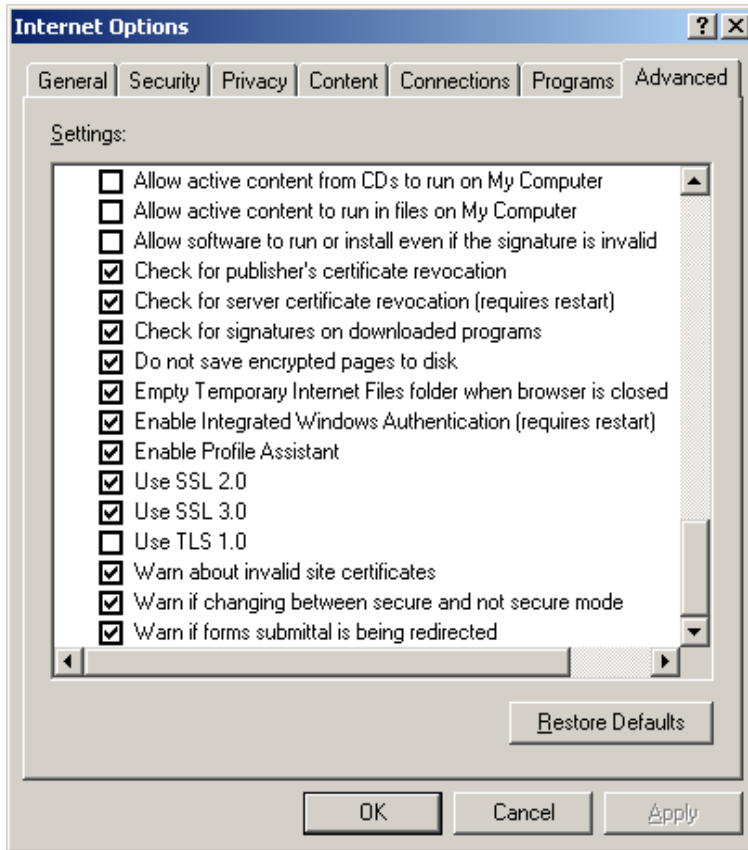
Firefox 1.5 and higher support Kerberos authentication out of the box. Internet Explorer 6.0 also supports Kerberos authentication. Though in both cases, it may be initially disabled.

In Internet Explorer, you must allow the site to log in automatically, and you must also enable Windows Integrated Authentication. If either of these fail, the authentication request will not be successful.

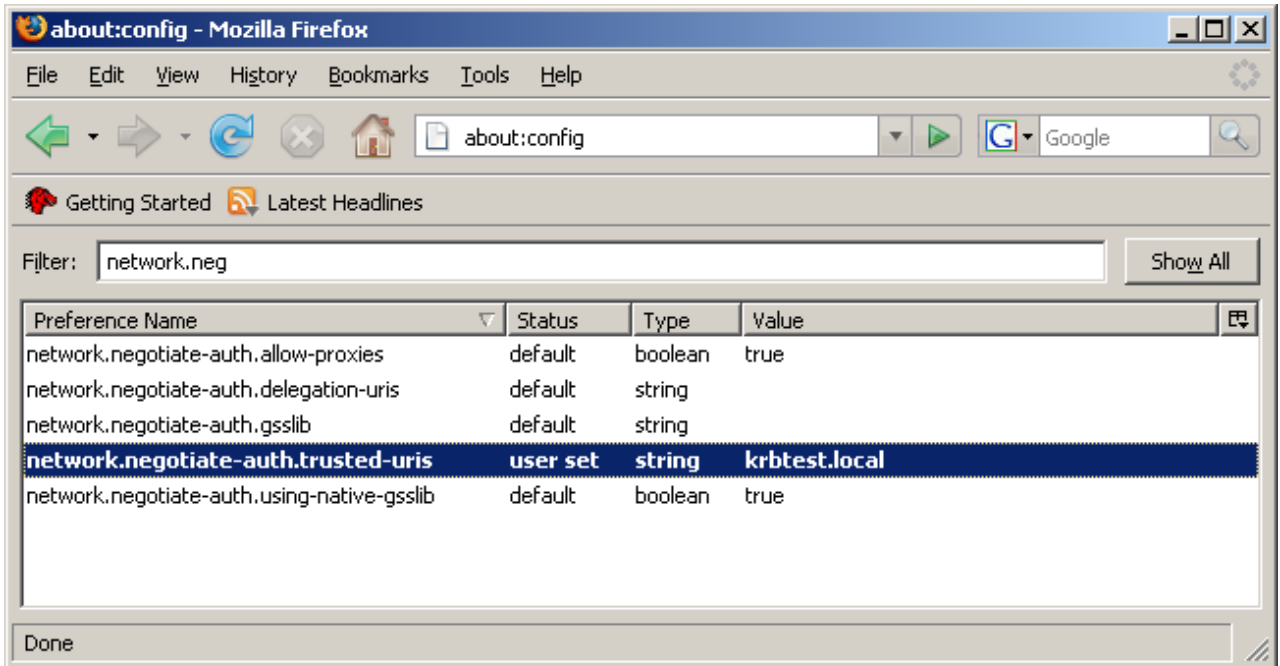
To enable the support in Internet Explorer, select the intranet zone, click custom level, and set the User Authentication settings as seen below. Then click **OK**.



Next, click on the Advanced tab. Ensure that the **Enable Windows Integrated Authentication** checkbox is checked as seen below. Click OK. If you had to enable this option, you will need to restart your computer.



To configure Firefox, type "about: config" in the address bar. Then type "network.negotiate-auth.trusted-uris" in the filter. Double click on the **network.negotiate-auth.trusted-uris** entry. enter the domain name. Click OK. The screen should look similar to that below:



1.5 Security Considerations

In Windows, every service principle account authenticates against the computer account in Active Directory. In the event where the keytab is compromised, the machine account should at least be disabled, if not deleted. It is also important to guard the keytabs very carefully because compromising one service's keytab might allow an attacker to spoof other services from the same server.

1.6 Further Reading

Microsoft has also published a knowledge base article outlining a different approach at: <http://support.microsoft.com/kb/555092>

The above article describes a completely different approach which would likely be useful in entirely different environments. Unlike the approach documented here, where the server is a full member server, and only Kerberos tickets are exchanged, the article details how to use the same software to accomplish "single password" capabilities. That approach may be useful when the target server is on the internet, perhaps where Kerberos negotiation authentication is not practical. However, in these cases it is extremely important that this be paired with SSL because otherwise domain passwords would be passed across the network in plain text.

1.7 Final Thoughts

Once Kerberos authentication is set up it is secure, reliable, and quite powerful. Unfortunately it can also be somewhat complex initially. This paper provides a basis for understanding how to set up Apache, and perhaps other services, to use Kerberos for authentication against an Active Directory domain controller. Such techniques allow Windows servers to play greater roles in

identity management and highlight how Windows authentication can still be used when the target software is running on a Linux platform.

1.8 About the Author

Chris Travers is the owner of Metatron Technology Consulting, a business dedicated to helping people and businesses leverage open source software. He has six years of experience with Kerberos and Windows Server technologies.