Using Kerberos security with Server for NFS

Article Number: 281 | Rating: Unrated | Last Updated: Wed, Jun 27, 2018 8:31 AM

Using Kerberos security with Server for NFS

May 13, 2010 by JoseBarreto // 1 Comments

- Share
- 0
- 0

I'll discuss how to configure a UNIX based NFS client to connect to Windows Server for NFS using Kerberos security with RPCSEC_GSS.

Traditionally NFS clients and servers use AUTH_SYS security. This essentially allows the clients to send authentication information by specifying the UID/GID of the UNIX user to an NFS Server. Each NFS request has the UID/GID of the UNIX user specified in the incoming request. This method of authentication provides minimal security as the client can spoof the request by specifying the UID/GID of a different user. This method of authentication is also vulnerable to tampering of the NFS request by some third party between the client and server on the network.

RPCSEC_GSS provides a generic mechanism to use multiple security mechanisms with ONCRPC (on which NFS requests are built). Server for NFS currently provides support for two Kerberos "flavors" over NFS using RPCSEC_GSS: krb5 and krb5i. krb5 provides Kerberos authentication at the RPC request level, while krb5i (Kerberos v5 with Integrity) also protects the NFS payload from tampering.

Explaining how to set up Kerberos security between a UNIX client and a Windows server running Server for NFS can best be accomplished by way of a simple example. In this tutorial we'll consider the following infrastructure scenario:

- Windows domain called NFSDOMAIN.COM running Active Directory on a DC named nfsdomain-dc.nfsdomain.com

- Windows server running Server for NFS: windowsnfsserver.nfsdomain.com
- UNIX client: unixclient.nfsdomain.com
- UNIX user on unixclient.nfsdomain.com: unixuser1 with UID 500
- UNIX group on uinxclient.nfsdomain.com: unixgroup1 with GID 500
- Windows user (NFSDOMAINunixuser1)
- Windows group (NFSDOMAINunixgroup1)

For the purposes of this configuration guide I'll be assuming unixclient.nfsdomain.com is running OpenSolaris.

Basics:

First, make sure that DNS name resolution is working properly using between the DC, the Windows NFS Server, and the UNIX client.

One caveat for the Solaris client is that the hostname should be set to just the first part of the FQDN. Running "hostname" on unixclient.nfsdomain.com should output only "unixclient". If not, set the hostname to just "unixclient".

Make sure that all the above users and groups have been created and that NFSDOMAINunixuser1 is a member of NFSDOMAINunixgroup1 group. Be sure that you've set a password for unixuser1. If the UID/GID numbers are different on the UNIX client, just substitute the appropriate values where I'm using 500.

Joining the UNIX client to AD:

Now we're going to configure unixclient to get Kerberos tickets from the NFSDOMAIN.COM domain. This is done by editing the /etc/krb5/krb5.conf file:

There should be an existing file with some placeholders which can be edited. We're going to fill in the following fields: "default_realm", "kdc", "admin_server", and "[domain_realm]". We're also going to add two lines under "[libdefaults]" for "default_tkt_enctypes" and "default_tgs_enctypes". I'll get to why these are added later. The end result should look something like:

```
<Begin /etc/krb5/krb5.conf snip>
[libdefaults]
    default_realm = NFSDOMAIN.COM
      default_tkt_enctypes = arcfour-hmac-md5 des3-cbc-sha1-kd des-cbc-md5
      default_tgs_enctypes = arcfour-hmac-md5 des3-cbc-sha1-kd des-cbc-md5
```

```
[realms]
      NFSDOMAIN.COM = {


                  kdc = nfsdomain-dc.nfsdomain.com
                  admin_server = nfsdomain-dc.nfsdomain.com
      }

[domain_realm]
      nfsdomain.com = NFSDOMAIN.COM
</snip>
```

At this point we should be able to test getting a ticket for NFSDOMAINunixuser1 from unixclient:

From unixclient run: “kinit unixuser1” and type in the user’s password.

Now run “klist”, you should have a ticket for unixuser1! Run “kdestroy” to destroy the ticket.

How does NFS use RPCSEC_GSS?

Ok, now that we have the basic Kerberos setup working I’ll explain a bit how authentication works from the NFS standpoint.

When unixclient wants to authenticate with windowsnfsserver, it needs some other “user” (in Kerberos this is called a “principal”) to authenticate with. The way this happens is that when a NFS share is mounted, the client looks at the FQDN of the NFS Server and expects to authenticate with the following principal: “nfs/FQDN@domain_realm”. In this case unixclient is going to look for “nfs/windowsnfsserver.nfsdomain.com@NFSDOMAIN.COM”. By convention, UNIX machines do the initial few NFS operations as “root”. In this context it’s the root account of the local machine and again, by convention, this principal is “root/unixclient.nfsdomain.com@NFSDOMAIN.COM”.

What does this mean? How do I set up these principals?

For “nfs/windowsnfsserver.nfsdomain.com@NFSDOMAIN.COM” we’re just going to user the existing “machine” account in AD and essentially create an alias for it.

From nfsdomain-dc run:

```
setspn -A nfs/windowsnfsserver windowsnfsserver
setspn -A nfs/windowsnfsserver.nfsdomain.com windowsnfsserver
```

then:

```
setspn -L windowsnfsserver
```

You should see all of the machine account’s “service principal names”.

What about the principals for unixclient?

We’re going to create some users in AD for these principals. Because “/” is not a valid character for AD account names we’ll have to pick a different name, then use the setspn utility to add the service principal names. For the sake of completeness, we’re going to create a couple more accounts than just the root account.

on nfsdomain-dc create the following users and set passwords for them:

- unixclienthost (represents the unixclient machine)
- unixclientroot (account for root on unixclient)
- unixclientnfs (used by the NFS Server on unixclient)

After creating these users, right click on each user, go to properties and under the “Account” tab change the “User logon name” to “host/unixclient.nfsdomain.com”, “root/unixclient.nfsdomain.com”, and “nfs/unixclient.nfsdomain.com” respectively.

Now we’re going to set the SPNs on these accounts:

```
setspn -A host/unixclient unixclienthost
setspn -A host/unixclient.nfsdomain.com unixclienthost
setspn -A root/unixclient unixclientroot
setspn -A root/unixclient.nfsdomain.com unixclientroot
setspn -A nfs/unixclient unixclientnfs
setspn -A nfs/unixclient.nfsdomain.com unixclientnfs
```

unixclient needs to use the root/unixclient.nfsdomain.com principal without actually typing in a password for that account. This is accomplished with a “keytab” file.

We’re going to export keytab files for these accounts. On nfsdomain-dc run:

```
ktpass -princ host/unixclient.nfsdomain.com@NFSDOMAIN.COM -mapuser unixclienthost -pass <insert password> -out
unixclienthost.keytab
ktpass -princ root/unixclient.nfsdomain.com@NFSDOMAIN.COM -mapuser unixclientroot -pass <insert password> -out
```

```
unixclientroot.keytab
ktpass -princ nfs/unixclient.nfsdomain.com@NFSDOMAIN.COM -mapuser unixclientnfs -pass <insert password> -out unixclientnfs.keytab
```

Now move these files from nfsdomain-dc to unixclient.

On unixclient we’re going to merge these files in the keytab file. From the directory where the files were copied run: “ktutil”. In this interactive tool run the following commands:

```
rkt /etc/krb5/krb5.keytab
rkt unixclienthost.keytab
rkt unixclientroot.keytab
rkt unixclientnfs.keytab
wkt /etc/krb5/krb5.keytab
q
```

Great, now unixclient should be able to get tickets for these accounts without typing any passwords. Test this out:

```
kinit -k host/unixclient.nfsdomain.com
kinit -k root/unixclient.nfsdomain.com
kinit -k nfs/unixclient.nfsdomain.com
```

Each of these commands should successfully get a ticket. This can be a bit tricky to get working in some configurations because the underlying encryption mechanism encoded in the ticket must be supported by both nfsdomain-dc and unixclient. You may also need to explicitly specify an encryption type when using the ktpass utility. “-crypto DES-CBC-MD5” seems to be widely supported. Depending on the version of AD in your environment, you may need to check the “Use Kerberos DES encryption types for this account” under the “Account” tab in the user property page. Check the security event log to help debug the logon attempts.

Mapping UID/GID to Kerberos principals on unixclient:

Ok, at this point we need to teach the NFS client on unixclient how to map its local UIDs and GIDs to Kerberos principals:

On unixclient run the following:

```
gsscred -m kerberos_v5 –a
gsscred -m kerberos_v5 –n host/unixclient.nfsdomain.com -u 0 –a
gsscred -m kerberos_v5 –n root/unixclient.nfsdomain.com -u 0 –a
gsscred -m kerberos_v5 –n nfs/unixclient.nfsdomain.com -u 0 –a
gsscred -m kerberos_v5 –n unixuser1 -u 500 –a
gsscred -m kerberos_v5 -n unixgroup1 -g 500 -a
```

These commands populated the GSS credentials table which is used to match incoming principals to local user accounts. Make sure that “gssd” is running on unixclient.

Enable Kerberos security for NFS:

OpenSolaris disables Kerberos over NFS by default. Obviously we need to enable it. Edit “/etc/nfssec.conf” and uncomment the “krb” lines. You should have the following:

```
<snip>
krb5          390003 kerberos_v5     default –              # RPCSEC_GSS
krb5i          390004 kerberos_v5     default integrity      # RPCSEC_GSS
krb5p          390005 kerberos_v5     default privacy        # RPCSEC_GSS
</snip>
```

Note that Server for NFS does not currently support krb5p.

Set up UID/GID mappings for Server for NFS:

On the windows side, you’ll still need to configure mapping between UNIX UIDs and GIDs to windows accounts. Server for NFS will always use the Kerberos principal it receives for each NFS request. However, the NFS protocol still only returns UIDs and GIDs for requests like GETATTR and READDIR. If no mapping is set up a “ls -l” on unixclient won’t show the correct UIDs and GIDs even though newly created files will have the correct security descriptor visible from windowsnfsserver.

Specify How Server for NFS Obtains Windows User and Group Information:
http://technet.microsoft.com/en-us/library/cc754514.aspx

This blog has useful information for using the Active Directory Lookup feature:
http://blogs.msdn.com/sfu/archive/tags/Active+Directory+Lookup/default.aspx

Provision a share on windowsnfsserver:

This configuration guide has details on how to set up a share:
http://technet.microsoft.com/en-us/library/cc770569.aspx

After creating the “share” directory, make sure that all intended users (including unixuser1) have access to it.

Mount the share:

Finally, we’re ready to mount a share!

Remember to use the FQDN of the server when mounting. This is so unixclient can figure out the proper principal name to look up for windowsnfsserver.

As root on unixclient run:

```
mount -o sec=krb5,vers=3,proto=tcp windowsnfsserver.nfsdomain.com:/share /mnt/share
```

Once the share has been mounted log i

Posted - Wed, Jun 27, 2018 8:31 AM. This article has been viewed 9847 times.

Online URL: http://kb.ictbanking.net/article.php?id=281