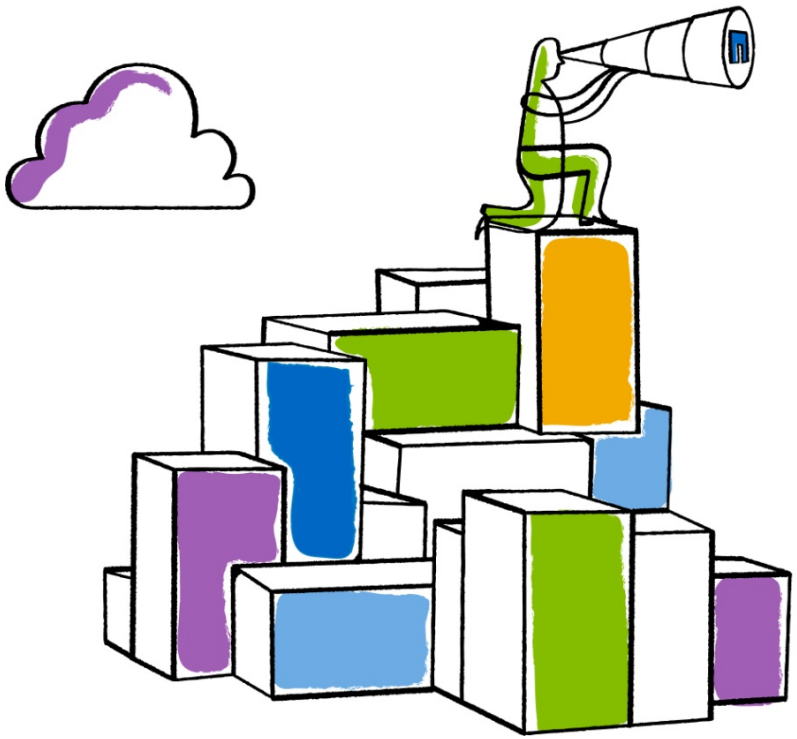
NetApp

Go further, faster®



RPCSEC_GSSv3 use in NFSv4.2

William A. (Andy) Adamson

andros@netapp.com

IETF 89, London

History

- IETF87: No progress on draft-ietf-nfsv4-rpcsec-gssv3
- Discussion on list of draft-20, which removed the GSSv3 requirement, exposed several issues with non-GSSv3 secure inter-server server side copy
  - Several choices but no clear solution from list discussions
- IETF88: Deciphered and presented choices from list to WG
  - We decided to pursue RPCSEC_GSSv3
- IETF89: New draft-ietf-nfsv4-rpcsec-gssv3-07
  - GSSv3 used in NFSv4.2 secure inter-server server side copy
  - GSSv3 used to complement NFSv4.2 LNFS
    - To add server-guest and full mode labeling

Secure Inter-server SSC Goals Review

1. Source server properly authenticates the destination server
2. Destination server READ is associated with the copy and is handled in a special manner by the source (see READ stateid issue slide)
3. Destination server is granted the privilege to act on behalf of the user-principal to READ.
4. Limit the ability of the destination server to act as the user-principal (e.g. a single copy)

Inter server SSC READ Stateid Issue

- COPY ca_src_stateid is from the client OPEN to the source server verified against the client clientid (NFSv4.1)
- Destination (acting as a client) to perform ‘normal’ READs from the source
  - No OPEN from the destination server to avoid (share) locking issues
  - READ with ca_src_stateid and the COPY SAVE_FH
- Source needs to know the READ ca_src_stateid is special
  - So as not to verify it against the destination server clientid

NFSv4.2 SSC use of RPCSEC_GSSv3

- Used for ***inter-server*** server-side copy
- A generated 'shared secret' is distributed via NFSv4.2 defined RPCSEC_GSSv3 structured privileges
  - The 'shared secret' is distributed first to the source, then to the destination, and finally is presented by the destination to the source as an identifier for the particular copy
- Compound authentication is also required to authorize the destination server to act on behalf of the user principal
  - User principal information required for compound authentication is passed from the client to the destination and then from the destination to the source.
- Privacy is used for all SSC RPCSEC_GSS_CREATE calls

NFSv4.2 Inter SSC Step 1

- The user principal establishes an RPCSEC_GSSv3 context with the source server (**princ-src** context)
- The user principal OPENs the file to be copied on the source server using the princ-src context.
- A copy_from_auth privilege is established on the source
  - A user principal is authorizing a source principal to allow a destination principal to setup the copy_confirm_auth privilege required to copy a file from the source to the destination on behalf of the user principal.
- A COPY_NOTIFY is sent to the source server using the copy_from_auth structured privilege GSSv3 handle.

copy_from_auth: RPCSEC_GSS_CREATE

- Client establishes this privilege on the source server

```
struct copy_from_auth_priv {
        secret4         cfap_shared_secret;
        netloc4         cfap_destination;
        /* the NFSv4 user name that the user principal maps to */
        utf8str_mixed   cfap_username;
  };
```

NFSv4.2 Inter SSC Step 2

- The user principal establishes an RPCSEC_GSSv3 context with the destination server (**princ-dst** context)
- The user principal OPENs the file to be copied to on the destination server using the princ-dst context.
- A copy_to auth privilege is established on the destination
  - A user principal is authorizing a destination principal to setup a copy_confirm_auth privilege with a source principal
- A COPY is sent to the destination server using a copy_to_auth structured privilege GSSv3 handle.
  - The copy_to_auth privilege grants the destination server the ability to setup a **compound authentication** assertion with the source server.

GSSv3 Compound Authentication

- In general, compound authentication is used for assertions that the server may only grant if a user and a client are authenticated together to the server.
- An established <client, server> GSSv3 protected connection is used as the ‘parent’ context
- A user principal’s context handle (‘inner handle’) , a nonce, and a MIC of the nonce using the user principals context is sent in an RPCSEC_GSS_CREATE rgss3_gss_binding payload
- The server verifies the inner handle by locating the inner handle context, and calling GSS_VerifyMIC on the nonce

NFSv4.2 Inter Server SSC & Compound Authentication

- We want to establish a GSSv3 compound authentication assertion using the user principal context as the “inner handle” from the destination (acting as a client)
  - The user principal has no context established with the source on the destination
- What we need is to use the user principal **princ-src** context which was used for the OPEN of the source file to be copied.
- The copy_to_auth privilege provides the information that the destination (as a client) will use to establish a compound authentication with the source

copy_to_auth: RPCSEC_GSS_CREATE

```
struct copy_to_auth_priv {
        /* equal to cfap_shared_secret */
        secret4         ctap_shared_secret;
        netloc4         ctap_source;    Note: needs to be the same list as in COPY
        /* the NFSv4 user name the user principal maps to */
        utf8str_mixed   ctap_username;
        opaque          ctap_handle;  ←princ-src context handle
        /* A nounce and a mic of the nounce using ctap_handle */
        opaque          ctap_nounce;
        opaque          ctap_nounce_mic;
};
```

NFSv4.2 Inter SSC Step 3

- The destination (as a client) establishes an RPCSEC_GSSv3 context with the source server (**src-dst** context).
  - Similar to a client to pNFS file layout data server connection
- A copy_confirm_auth privilege is established on the source
  - A destination principal is confirming with the source principal that it is authorized to copy data from the source.
  - A MIC of the shared secret identifies the particular copy
- READs are sent to the source server using the resultant copy_confirm_auth privilege GSSv3 handle.
- The resultant GSSv3 handle MUST be destroyed by the destination if the copy_to_auth privilege handle is destroyed

copy_to_auth: RPCSEC_GSS_CREATE

```
struct copy_confirm_auth_priv {
        /* equal to GSS_GetMIC() of cfap_shared_secret */
        opaque          ccap_shared_secret_mic<>;
        /* the NFSv4 user name that the user principal maps to */
        utf8str_mixed   ccap_username;
};
struct rgss3_gss_binding {
        opaque      rgb_handle<>; /* inner handle */  <- princ-src handle
        opaque      rgb_nonce<>;
        opaque      rgb_nounc_mic<>;
};
```

GSSv3 & Secure Inter-server SSC Goals

- Authenticates the destination server
  - YES, via the shared secret distributed via GSSv3
- Destination READ special handling at source
  - YES, using the copy_confirm_auth GSSv3 handle for READs
- Act on behalf of the user-principal
  - YES, via the use of compound authentication in the copy_confirm_auth privilege
- Limit the destination server
  - YES, client destroys the copy_from_auth and copy_to_auth GSSv3 context handles -> destination destroys copy_confirm_auth GSSv3 handle

NFSv4.2 LNFS and RPCSEC_GSS3

- NFSv4.2 LNFS without GSSv3 achieves client-guest labeling
  - Label asserted by the client, stored on the server
- A GSS3 CREATE control message is sent to bind a set of security labels to the resultant GSS3 handle
  - The client first performs a GSSv3 LIST control message asking the server which security labels it supports
- Resultant GSSv3 handle used for NFSv4.2 LNFS calls to achieve full mode labeling
- Or GSSv3 handle used without NFSv4.2 LNFS calls to achieve server-guest labeling

Thank you