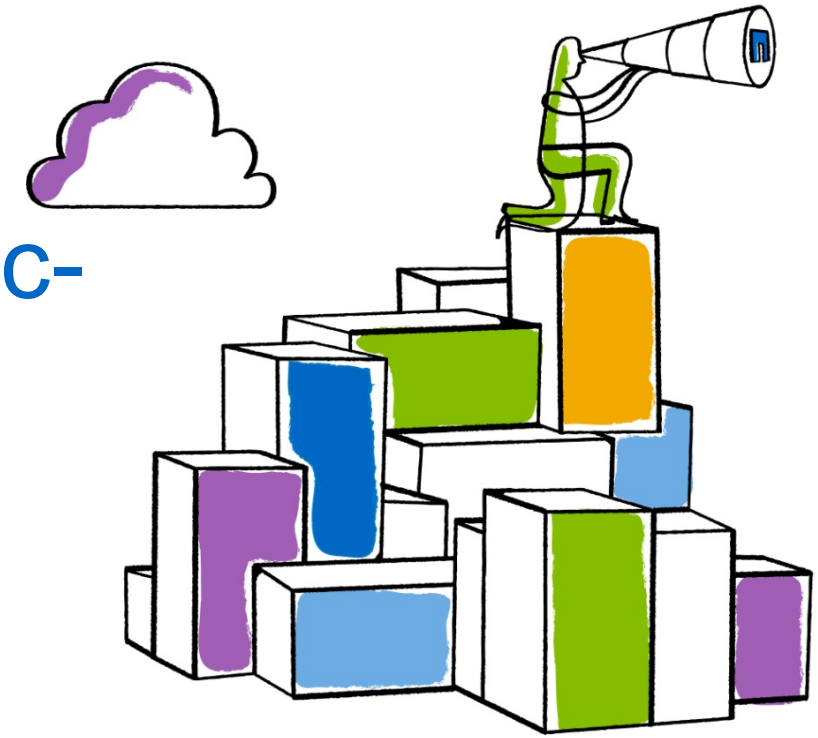
NetApp

Go further, faster®



Draft-ietf-nfsv4-rpcsec-gssv3-07

William A. (Andy) Adamson

andros@netapp.com

IETF 89 London

Motivation

- Refocus draft-ietf-nfsv4-rpcsec-gssv3 on NFSv4.2 use
  - Responded to draft-06 review comments
- NFSv4.2 Secure Inter-server SSC
  - IETF 88 discussion picked RPCSEC_GSSv3 for NFSv4.2 Inter server SSC
- NFSv4.2 Labeled NFS Full mode or server-guest mode
  - Unchanged from draft-06

Changes Between draft-06 and draft-07

- Responded to Mike Eisler's review comments
- RPCSEC_GSS version 3 now a proper super set of version 2
  - Uses RFC 5403 (GSSv2 draft) as a blue print
- Version 3 of the protocol adds two new **control** messages:
  - RPCSEC_GSS_LIST
    - Used to list server supported assertions (server side labels)
  - RPCSEC_GSS_CREATE
    - Creates a new '**child**' GSSv3 handle bound to CREATE assertions

Changes Between draft-06 and draft-07

- RPCSEC_GSS_BIND_CHANNEL not used
  - Use GSSv3 channel binding instead
- The RPC reply verifier is changed
  - Due to the “child” handle construction
  - NFSv4.1 SSV looks to have a similar issue
- Only RPCSEC_GSSv3 handles are used
  - No more mixed versions

Changes Between draft-06 and draft-07

- RPCSEC_GSSv3 credential has the same format as the version 1 and version 2 credential
- Setting rgc_version field to 3 indicates support for GSSv3 procedures

```
union rpc_gss_cred_t switch (unsigned int rgc_version) {
    case RPCSEC_GSS_VERS_1:
    case RPCSEC_GSS_VERS_2:
    case RPCSEC_GSS_VERS_3: /* new */
            rpc_gss_cred_vers_1_t rgc_cred_v1;
    };
```

RPCSEC_GSS_CREATE

- Requires an existing GSSv3 context (the “parent” context)
- Creates a ‘***child***’ GSSv3 handle bound to the assertion(s) in the payload.
  - Shares the parent context
- The NULLPROC payload is any combination of:
  - Channel binding
  - Compound authentication
  - Structured privileges (RPC application defined)
  - Security label assertions

RPCSEC_GSS Reply Verifier Change

- The call verifier is a MIC over the RPC header starting at the XID and including the credential
- The reply verifier is a MIC over just the RPCSEC_GSS sequence number
- Since the GSSv3 child handle shares a context with the GSSv3 parent handle and could have the ***same sequence number***, there is an opportunity for a ***man in the middle*** attack

Man in the Middle Attack

- Man in the middle caches <sequence number, reply verifier> tuples from an RPCSEC_GSSv3 session (parent handle)
- When child handle is created, attacker intercepts a child handle RPC call and grabs the sequence number from the call credential.
- If the child handle sequence number exists in the parent handle “tuple” cache, the attacker can craft a reply to the client using the associated verifier from the cache.

RPCSEC_GSS Reply Verifier Change

- The man in the middle attack is thwarted by changing the reply verifier to be the same as the call verifier: a MIC over the RPC header starting at the XID and including the credential:

```
msg_type mtype; /* set to REPLY */
unsigned int rpcvers;
unsigned int prog;
unsigned int vers;
unsigned int proc;
opaque_auth  cred; /* captures the RPCSEC_GSS handle */
```

Compound Authentication

- For authority assertions that the server may only grant if a user and a client are authenticated together to the server
- In addition to the parent context handle, the compound authentication payload contains a GSSv3 context handle called the '***inner***' handle along with a
  - Nonce
  - MIC of the nonce using the 'inner' GSS-API security context
- All uses of a child context handle that is bound to an inner context MUST be treated as speaking for the initiator principal of the inner context handle's GSS-API security context

Structured Privilege Assertion

- An RPC application defined opaque structure. It's encoding, server verification, and any server policies are described by the RPC application definition.
- NFSv4.2 Inter-server server side copy defines three structured privileges to authorize the destination server to copy a single file from the source server on behalf of the user principal. (*draft-ietf-nfsv4-minorversion2-21*)

Security Label Assertions

- The client performs a GSSv3 LIST control message asking the server which security labels it supports
- A GSS3 CREATE control message is sent to bind a set of security labels to the resultant GSS3 handle

Using a GSSv3 Context

- All NFS operations using the GSS3 handle assert all successful privileges and features associated with it's creation.
- Clients and servers therefore need to cache the specific privileges and features along with the GSS3 handles

Issues

- Parent and child handle creation and destruction
  - Can a child handle be a parent handle (NO!)?
  - Is the child handle destroyed when the parent is destroyed (YES!)?
- Review NFSv4.2 and GSSv3 security label handling
  - Ensure the two protocols handle the use cases
  - GSSv3 security label assertions should not define behavior
    - Just be the transport, let the labels define behavior
- Error messages need to be completed
- Have some draft-07 review comments
  - Note that GSSv3 reference to NFSv4.2 is now *informative*

NetApp

Thank you