RPCSEC_GSSv3 Update

Andy Adamson
andros@netapp.com
Connectathon March 2016
Presenter Title
Date

State of Draft

- Remote Procedure Call (RPC) Security Version 3
- Current draft
  - draft-ietf-nfsv4-rpcsec-gssv3-17 of the
- WG State: Submitted to IESG for Publication

Overview

- GSS-API mechanisms are insufficient for communicating certain authorization and authentication information to a server.
- GSS-API and its mechanisms could be extended to address this shortcoming.
- Chose to address them at the application layer in RPCSEC_GSS

Motivation

- Secure server-side copy for NFSv4.2
  - Use of structured privileges to authorize the destination to copy the file from the source on behalf of the user.
  - REQUIRED to implement for NFSv4.2
- Support for multi-level (labeled) security
  - Assert Mandatory Access Control (MAC) subject labels
  - Enables full mode MAC when combined with NFSv4.2 LNFS

Motivation

- Multi-principal assertions: client host and user
  - Address shared client datat cache poisoning attacks
    - One user caches data from server, re-display it for another user without fetching data from server, allows bogus data to be introduced into the cache
- Simplified Channel Bindings
  - Replaces RPCSEC_GSSv2

RPCSEC_GSSv3

- Two new control operations
  - RPCSEC_GSS_CREATE
    - Establish a label, structured privilege, multi-principal, or channel binding assertion.
    - No assertion payload is REQUIRED to implement
  - RPCSEC_GSS_LIST
    - List established assertions or Label Format Specifiers supported by the server, not REQUIRED
- MUST implement the new version, the new reply verifier, and the new auth stats

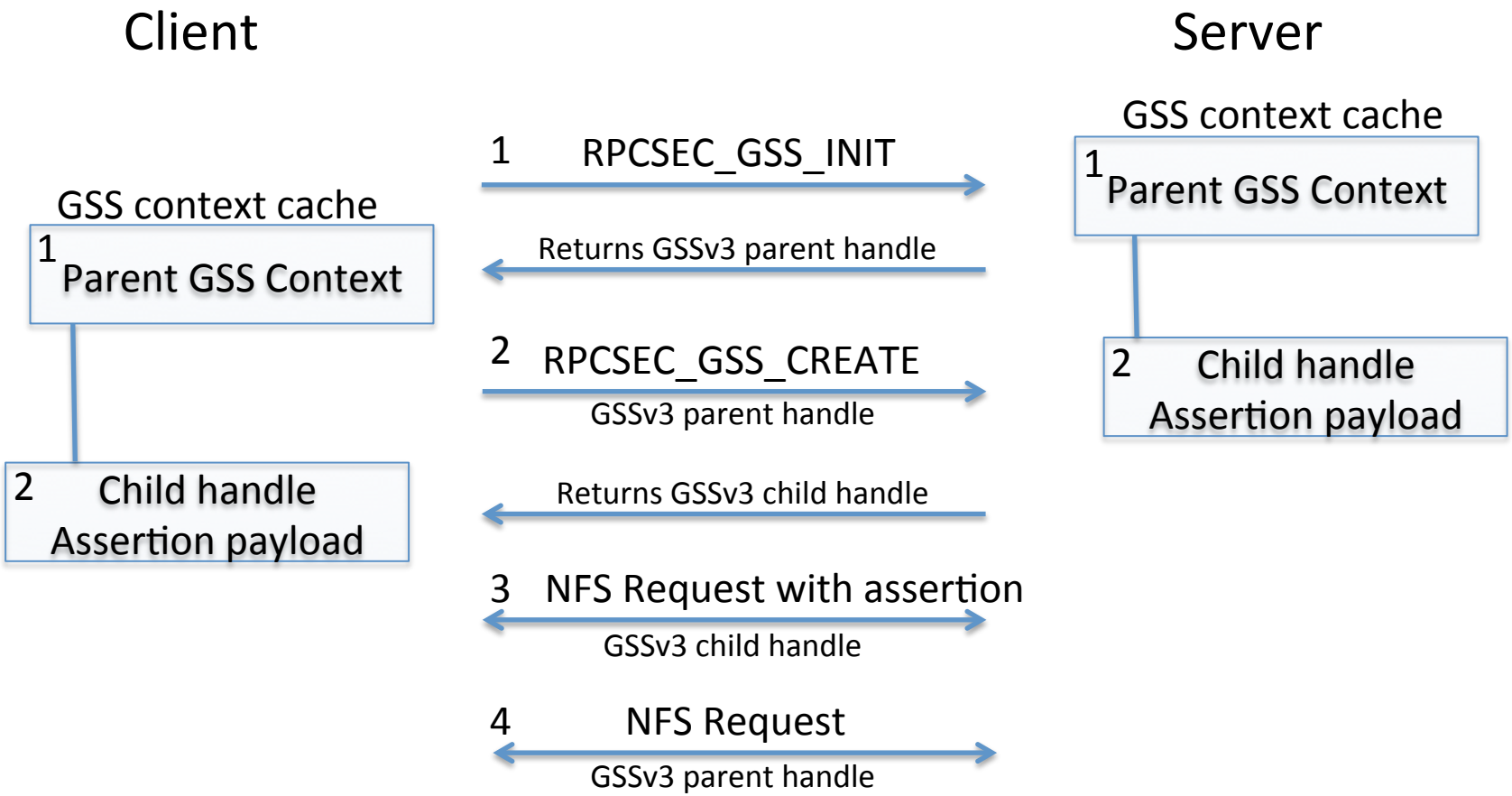
RPCSEC_GSS_CREATE

- Client uses an existing RPCSEC_GSSv3 context handle established in the usual manner
  - called the 'parent' handle
- Client sends RPCSEC_GSS_CREATE with assertion payload
  - NFS NULL call
- Server verifies create request and on success replies with a 'child' GSS context handle and list of successful assertions

RPCSEC_GSS_CREATE

- Client and server associate the assertion payload and child handle with the parent context
  - Child handle uses parent GSS security context
- Client uses child handle in NFS requests that require the associated assertion
  - Assert client process MAC subject label
  - Assert NFSv4.2 SSC structured privilege
- Client uses parent handle when no assertion required

Basic RPCSEC_GSS_CREATE Exchange



GSSv3 Prototype: New Version

- Negotiate GSSv3, and fall back to GSSv1 if GSSv3 is not supported
- Required changes to libtirpc which does the negotiation and rpc.gssd which passes the negotiated version to the kernel
- Client and server kernel changes to parse the GSS version

GSSv3 Prototype: New Reply Verifier

- Uses the exact same data as in the call verifier
  - except change the direction from ‘Call’ to ‘Reply’
- Required changes to libtirpc to verify the RPCSEC_GSS_INIT reply
- Client and server kernel changes to create and validate the reply verifier.

Mandatory Access Control (MAC)

- MAC bases access decisions with a subject label and a label on the object the subject wishes to access, which are both input to a policy engine.
- For NFS:
  - Subject label is from the client NFS request process
  - Object label is on a file
- LNFS stores the object label on the server
  - Client retrieves for client guest MAC
- GSSv3 asserts the subject label on the server

GSSv3 Prototype: Label Assertion

- A user makes the first NFS request
  - Causes a GSSv3 (parent) context to be created
- If using GSSv3 and SELinux is enabled set up full mode LNFS
  - Create a child context to enable full mode MAC
- After parent context creation and prior to the sending the NFS request, send an RPCSEC_GSS_CREATE with a label payload

GSSv3 Prototype: Label Assertion

- Export an SELinux routine to create a string version of the client NFS request thread's subject label for the label assertion payload
  - numeric subject label to string representation
- Client creates an RPCSEC_GSS_CREATE NFS NULL RPC with the label payload
  - All done in kernel, no upcall
- Prototype does not query server for supported Label Format Specifiers (RPCSEC_GSS_LIST)

GSSv3 Prototype: Label Assertion

- Server receives the RPCSEC_GSS_CREATE call
- Export another SELinux routine which creates a subject label from the label assertion payload
  - String form to SELinux numeric representation
- Server creates a child handle, and stores it with the label associating it with the parent context, then sends the reply

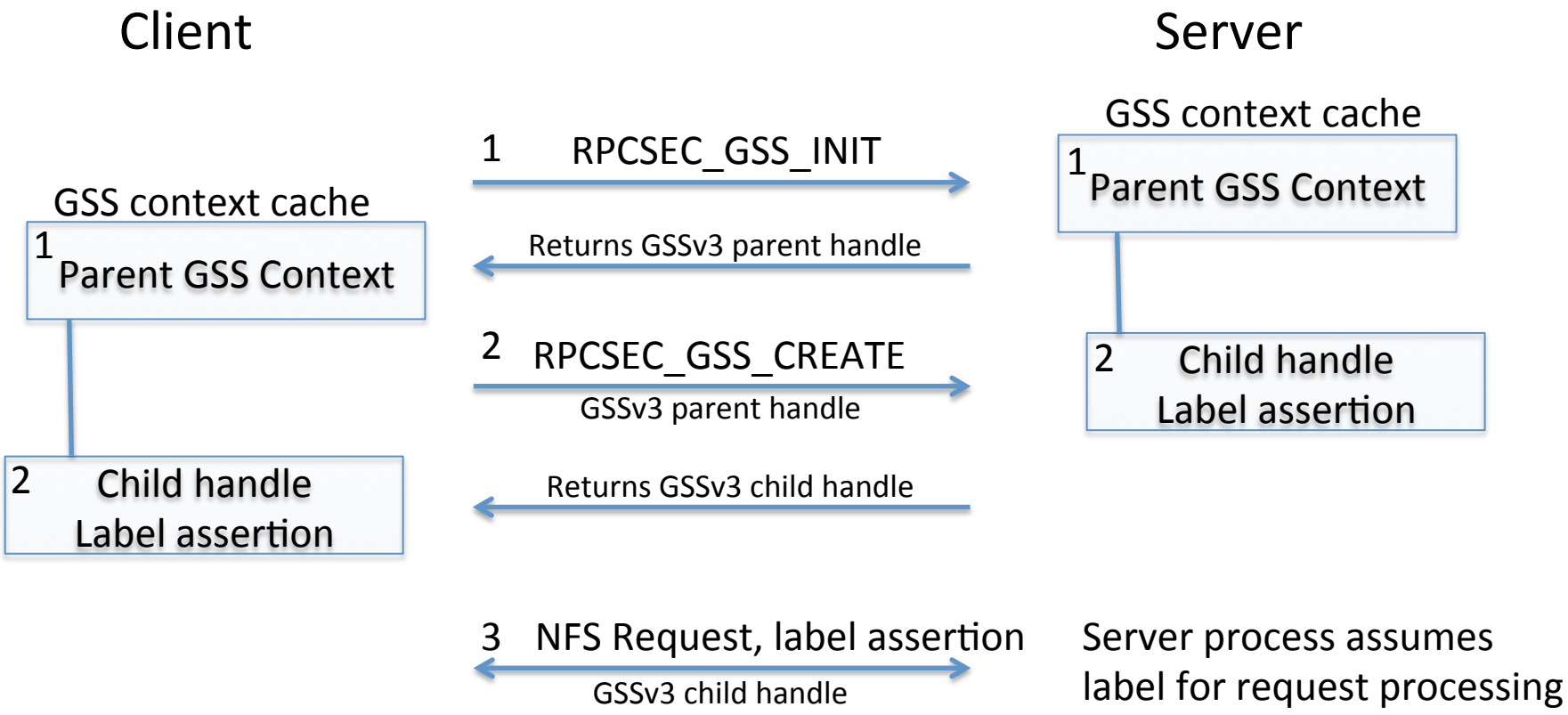
GSSv3 Prototype: Label Assertion

- Client receives the RPCSEC_GSS_CREATE reply
  - Stores child handle with the label associating it with the parent context
- NFS request is then sent using the new GSSv3 child handle to assert full mode MAC labels
  - Which NFS requests need to assert the label(s)?
    - Prototype uses the child handle for all subsequent requests where the request thread subject label is equal to the label assertion child handle payload

GSSv3 Prototype: Label Assertion

- Server uses child handle to look up parent context and assertion info
- Just as the server thread assumes the UID of the requestor while processing the request, it also assumes the assertion subject label
- Server SELinux can use thread subject labels and file LNFS label to enforce full mode MAC.

RPCSEC_GSS_CREATE Label Assertion



GSSv3 Prototype: Label Assertion Issues

- Will the SELinux implementation allow for thread labels to be used this way?
  - What are the security issues?
- How does the client determine which context handle to use?
  - Parent versus child
  - One child versus other child (multiple MAC labels?)
- Interoperability with other label systems such as FreeBSD

GSSv3 Prototype: Not complete

- Still an early prototype
- Have not asserted thread subject at server
- Error paths

Thank you