NFS INDUSTRY CONFERENCE

Rehabilitating NFS Security

Mike Eisler

Technical Director

Network Appliance, Inc.

mike@eisler.com

NFS INDUSTRY CONFERENCE

In 1984, NFS deserved its security reputation

- AUTH_SYS isn't real authentication
  - AUTH_SYS uses publicly available information (uids) to authenticate
  - Beside, it supported only 16 groups

NFS INDUSTRY CONFERENCE

NFS deserved its security reputation (continued)

- In lieu of authentication, NFS offered access control based on source client IP address
  - Access control usually enforced only at mount time partly because MOUNT and NFS are separate services listening on different ports
  - Attackers could eavesdrop for file handles, bypass MOUNT protocol, and so circumvent intended controls

NFS INDUSTRY CONFERENCE

NFS deserved its security reputation (continued)

- Initial transport was UDP
  - Thus simple-password-based authentication impractical
  - Drive-by-shootings easier with UDP: spoof a source UDP address, fake an identify, and use WRITE to corrupt a file
  - Security hardening technologies impossible/hard to use:
    - TCP-Wrappers
    - firewalls
    - SSH

NFS INDUSTRY CONFERENCE

NFS deserved its security reputation (continued)

- Protocol specifications for adjunct services (lock manager, status monitor, rquota) weren't specified to use the same access controls, transport type, authentication mechanism, etc. as the NFS/MOUNT session

NFS INDUSTRY CONFERENCE

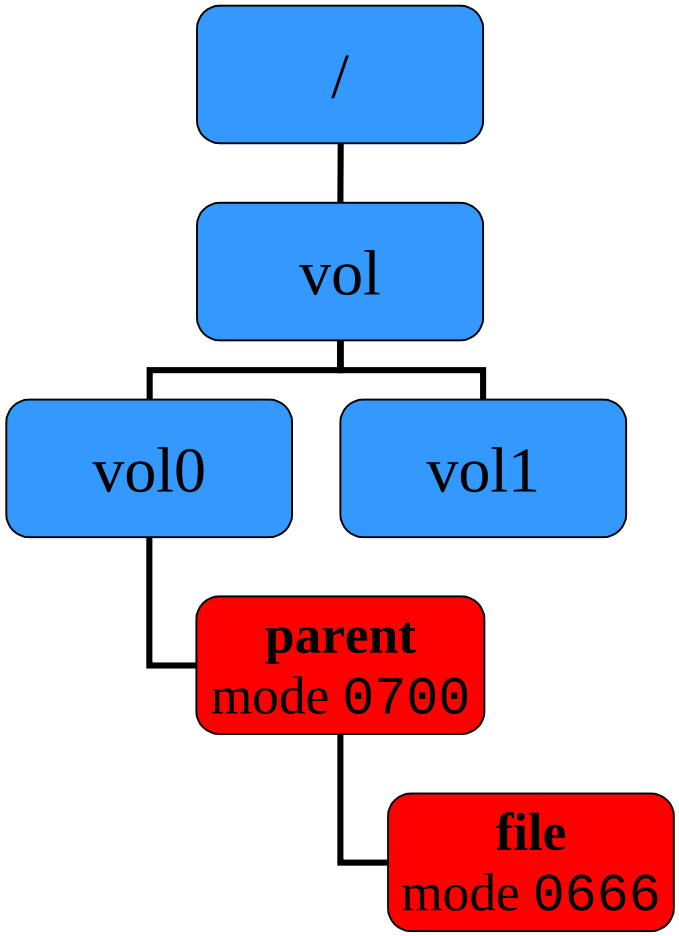
NFS deserved its security reputation (continued)

- Many early implementation errors.
  - The set of errors was captured in the SATAN tool of the early 1990s,
    - As a result this set is a non-issue among the major reference implementations and derivatives

NFS INDUSTRY CONFERENCE

NFS deserved its security reputation (continued)

- Stateless model meant no wire OPEN operation, which led to need for persistent file handles
  - Persistent file handles permit attacks to circumvents permission ancestor directories
  - leaf **`file`** is writeable by all, even though **`parent`** directory is accessible by just the owner



NFS INDUSTRY CONFERENCE

NFS deserved its security reputation (continued)

- 32 bit user and group identifiers forced enterprises to use a flat, common id namespace
  - fiefdoms within the enterprise that didn't go along couldn't share data across domains

NFS INDUSTRY CONFERENCE

NFS deserved its security reputation (continued)

- Cached data on the client represented a security hole
  - e.g. One thing NFS got right was to (by default) map super-user (**root**) to an unprivileged user (**nobody**) on the server
    - But if the data for some other user was cached, **root** on the client could read it

NFS INDUSTRY CONFERENCE

Why was NFS security created this way?

- Ease (use, deployment, implementation), cost, and performance trumped security considerations
  - NFS had to run in the kernel to perform
  - By the mid 1980s, UNIX kernels were widely divergent, creating challenges for porting the NFS reference code
  - hard stuff like security was done in user-space (MOUNT) or not at all

NFS INDUSTRY CONFERENCE

Why was NFS security created this way?

- NFS was invented during the Camelot Era of the Internet (I.e. before Morris unleashed the Worm)
- Cold-War Era translated to very restrictive and bizarre Export Control regime

NFS INDUSTRY CONFERENCE

Fixing security after the fact was hard

- 1987: AUTH_DH (AUTH_DES) was first crypto-based NFS/RPC security flavor
  - Few implementations
  - Yet ahead of its time; every NFS request and response authenticated
  - authenticated users, not NFS client nodes, to NFS servers
  - As a by product, solved too-many-groups problem of AUTH_SYS
- 1992: AUTH_KERB (Kerberos V4) shared same problems, fewer implementations
- Neither of above supported integrity or privacy
- Security experts soon scorned both for crypto weaknesses
  - A lot of development effort to produce something that was considered Dead-On-Arrival

NFS INDUSTRY CONFERENCE

Fixing security after the fact was hard (continued)

- 1993: NFSv3 introduced the ACCESS operation which solved issue of access control to cached data. But:
  - It took 8 years before we could declare NFSv3 ubiquitous
  - The ACCESS operation is still poorly implemented in NFSv3 clients

NFS INDUSTRY CONFERENCE

Fixing security after the fact was hard (continued)

- 1993: NFS/TCP implementations arrive
  - Unlike NFSv3, we still can't call it ubiquitous
  - Lots of problems in some implementations
  - Customers loathe to switch from UDP
    - We (the NFS implementers) have unwittingly addicted users to UDP

NFS INDUSTRY CONFERENCE

Fixing security after the fact was hard (continued)

- POSIX (draft) ACL standards are implemented among major UNIX-based NFS clients, but
  - none of the NFS ACL protocols interoperate

NFS INDUSTRY CONFERENCE

Fixing security after the fact was hard (continued)

- On non-Windows platforms, NFS had no real competition, hence less pressure to improve
  - While technically superior, more secure, AFS and DCE/DFS couldn't compete due to more expensive licensing terms
  - AFS and DCE/DFS didn't fail because they imposed security on the customer

NFS INDUSTRY CONFERENCE

Despite improvement, the lowest common denominator was and remains:

- NFS version 2
- UDP
- AUTH_SYS
- no-per-NFS-request access control

NFS INDUSTRY CONFERENCE

In hindsight, NFS ...

- should have been TCP only
- at mount time should have authenticated to server via per-host passwords (Kerberos would have followed)
- mounting should have been part of NFS protocol, thus binding mount authentication and authorization to subsequent NFS traffic
- In this alternate universe, NFS security would had a decent foundation that would allow incremental improvement
- In our universe, we've been forced to attack the major problems at once

NFS INDUSTRY CONFERENCE

Progress in Fixing the NFS Security Image

- In the mid-1990s several events turned the tide in the dismal story of NFS authentication
  - IETF mandated new standards to have security
  - Sun ceded change control of ONC RPC and NFS to IETF
    - Now RPC and NFS had be secure if RFCs for them were to be published
  - IETF published Generic Security Services (GSS) and Kerberos V5 standards
  - Microsoft announced that NT 6.0 (W2k) would use Kerberos V5 as it primary authentication system
- This made it inevitable that the future of NFS authentication would be Kerberos V5

NFS INDUSTRY CONFERENCE

Progress: Standardization, slow but effective

- 1997
  - IETF publishes RFC 2203: RPCSEC_GSS - RPC authentication using GSS
  - NFSv4 working group chartered with good security among goals
- 1997-1999 NFSv4 WG debates security model, resolving issue at Connectathon 1999
- 1998 U.S. government relaxes Export Controls
- 1999 First NFSv[23]/Kerberos V5 implementations ship (Hummingbird, Netmanage, Sun)
- 2000 RFC 3010, strawman NFSv4, mandates Kerberos V5
- 2002 First NFSv4/Kerberos V5 implementations ship (Network Appliance, early access Linux [U. of Michigan/CITI]).
  - NFSv[23] also supported.
- 2003 RFC 3530 published, obsoletes old NFSv4 RFC 3010
- Five implementations and counting. Better progress than previous strong authentication attempts.

NFS INDUSTRY CONFERENCE

Progress: NFS Security Features Unique to version 4

- LIPKEY/SPKM – SSL-like security model
- NT-like ACL model with some UNIX concessions
- Volatile File Handles – potential to eliminate weaknesses of persistent file handles
- All functions (mounting, locking, filing, state recovery) bound to same fixed port, which is firewall friendly
- String-based user identifiers provide hook for authorizing users from foreign domains
- NFSv4 kicks the UDP habit

NFS INDUSTRY CONFERENCE

Progress: NFSv4 - a marketing tool for NFS security

- Security, more than delegations, migration, and replication has driven interest in NFSv4
- Customers know it is an IETF standard, so it is secure
- Some of those customers are then surprised that NFSv[23] have Kerberos V5 too

NFS INDUSTRY CONFERENCE

Preventing a relapse: What could go wrong

- UDP and NFSv4: Are implementers really going stay the TCP-only course?
- When will the other UNIX clients support Kerberos V5?
- Linux is the growth engine for NFS clients, making the need for a robust Linux (2.4) NFS/Kerberos V5 client urgent.
- NFSv4 ACLs don't perfectly map to POSIX ACLs. Unwillingness to accept imperfect mappings jeopardizes client ACL support
  - Perhaps we need a user-level NFSv4 ACL editor

NFS INDUSTRY CONFERENCE

Preventing a relapse: What could go wrong (continued)

- Cross-domain authorization is in demand, but not implemented
  - NFSv4 WG is considering documents to aid implementers
- 56 bit DES for Kerberos V5 is insufficient
  - AES is the replacement, but Kerberos V5 standards for AES not done
  - Meanwhile, some NFS implementers are doing Triple DES
    - In software, Triple DES is very slow, very CPU intensive, will generate customer surprises

NFS INDUSTRY CONFERENCE

Preventing a relapse: What could go wrong (continued)

- Hardware accelerated crypto is coming, but focus is likely on IPsec, not NFS
  - Hardware accelerated IPsec will outperform software AES (and software 3DES)
  - This raises specter of NFS security being considered solid, but too slow to be useful
  - NFSv4 WG is specifying a new mechanism for leveraging IPsec integrity and privacy while using Kerberos V5 for user to server authentication

NFS INDUSTRY CONFERENCE

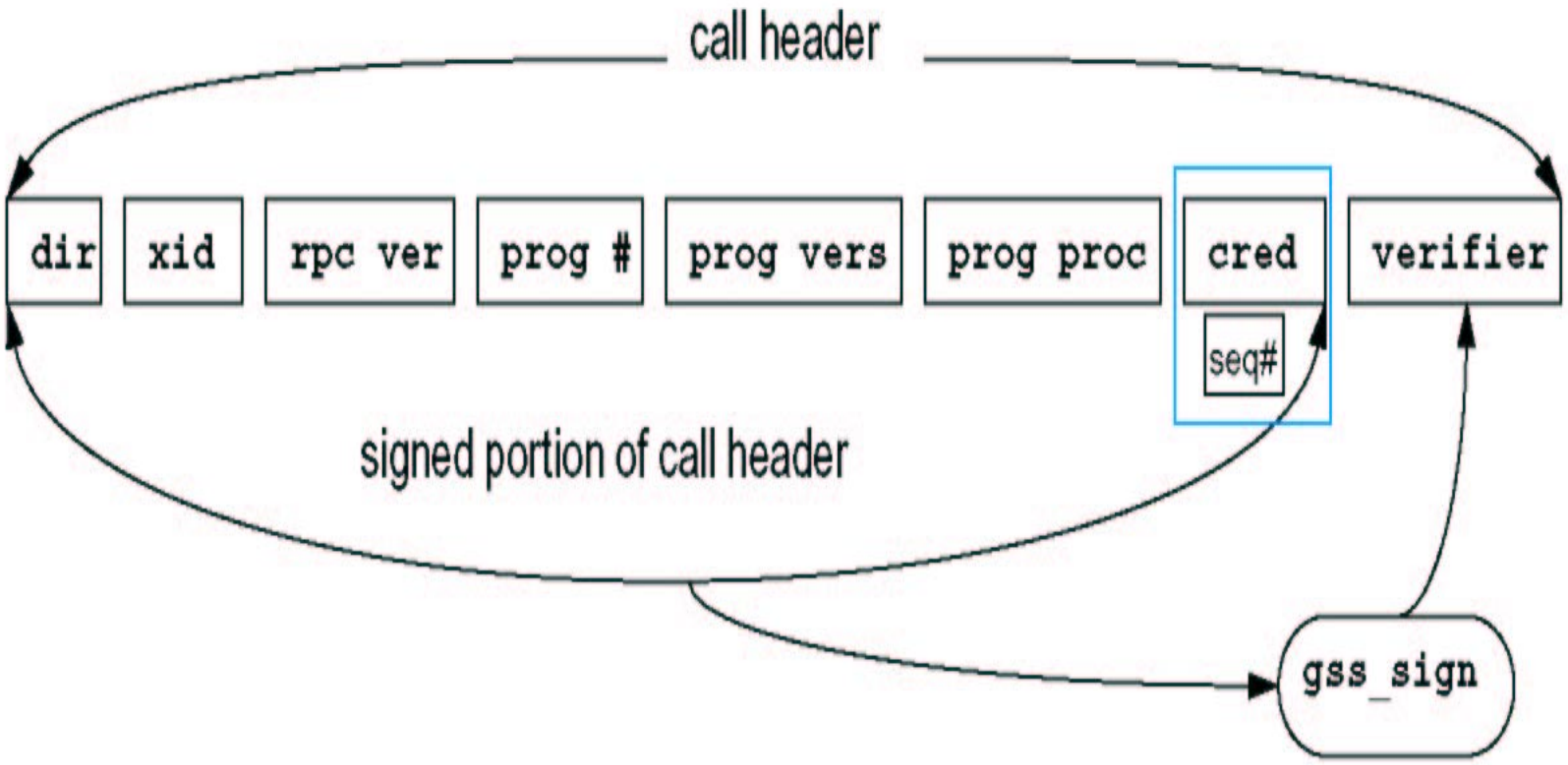
Questions?

NFS INDUSTRY CONFERENCE

Backup Slides

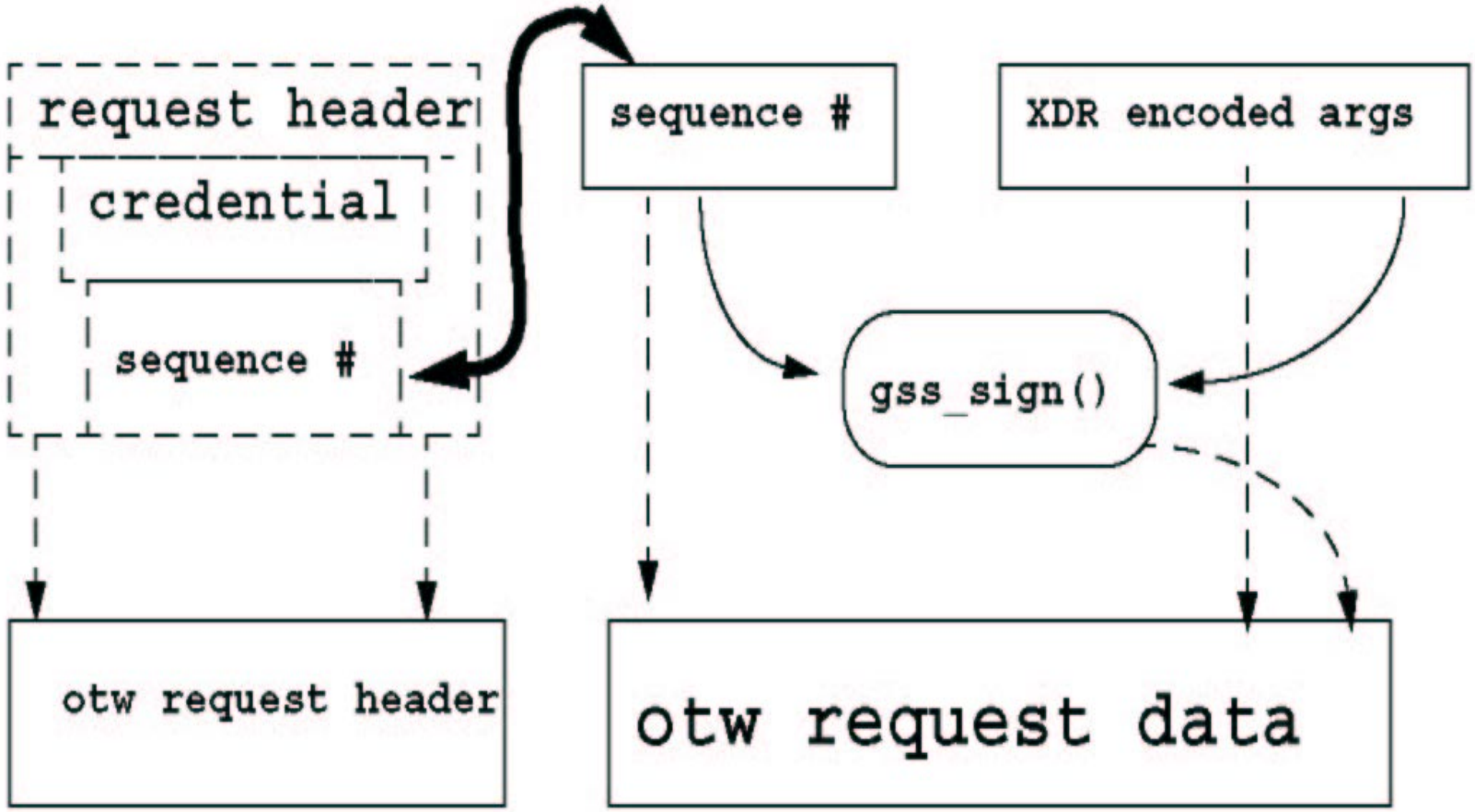
NFS INDUSTRY CONFERENCE

Overview of RPCSEC_GSS



NFS INDUSTRY CONFERENCE

Overview of RPCSEC_GSS - Integrity



NFS INDUSTRY CONFERENCE

Overview of RPCSEC_GSS - Privacy