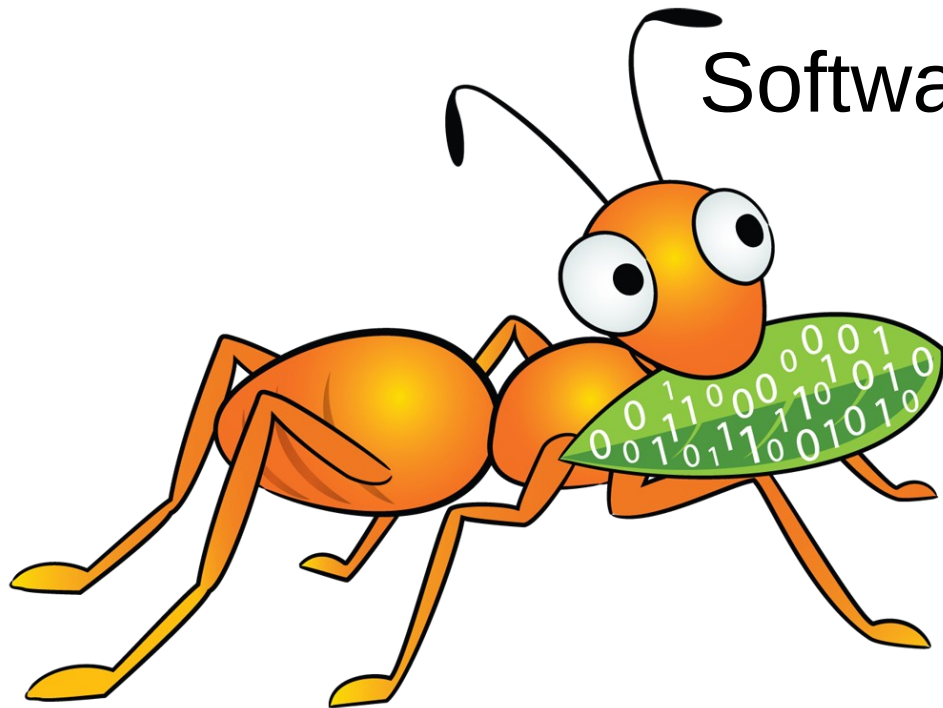


Debugging GlusterFS protocols with help from Wireshark



Niels de Vos
Software Maintenance Engineer
Global Support Services
Red Hat, Inc.

Gluster Workshop
Barcelona, November 2012

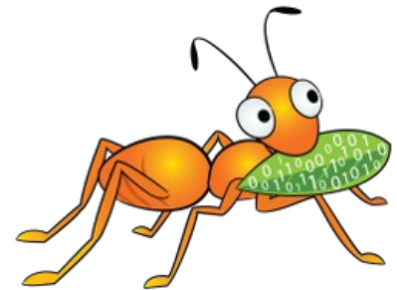
Agenda

- Some typical use-cases
- Different protocols used by Gluster
- Capturing the network traffic
- Browsing network traces with Wireshark
- Correlate the trace with the events or logs



Applications of Wireshark

- Troubleshooting (non)working environments
- Debugging during development
- Performance checking
- Auditing of network traffic



General protocol details

- All calls and replies are RPC based as defined in [RFC 5531](#) “Remote Procedure Call Protocol Specification Version 2”
 - Standard for programs, procedures and versions
 - AUTH_FLAVOR (credentials) is Gluster specific
- The encoding of parameters and structures is done in the “External Data Representation Standard” (XDR, [RFC 4506](#), 'man 3 xdr')



Overview of Gluster components

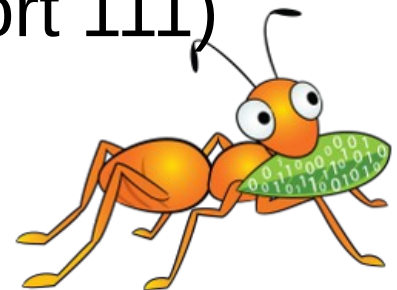
- gluster commandline
- glusterd management daemon
- glusterfsd for brick access
- glusterfs as a FUSE-client and NFS-server

glusterd is the center of everything and talks to all components, each with their own protocol.



Used ports

- Everything is TCP
- Outgoing connections use a port < 1024 by default
- glusterd listens on port tcp/24007 (rdma/24008)
- glusterfsd (brick) listens on port ≥ 24009
- The NFSv3 server (a glusterfs process) listens on ports 38465-38467
 - Registered in the standard portmapper (port 111)
- NLM (locking) uses port 38468



The Gluster CLI protocol

- Used between the client 'gluster' command and the service 'glusterd'
- Normally 'gluster' connects to 'localhost'
- Some common operations:
 - CREATE_VOLUME, START_VOLUME, GET_VOLUME, ADD_BRICK



The GlusterD Management protocol

- Used for coordination between different glusterd instances
- Part of glusterd, using port 24007
- Common operations:
 - CLUSTER_(UN)LOCK, STAGE_OP, COMMIT_OP



The Gluster DUMP protocol

- Used for discovering the available RPC programs and versions of an other glusterd processes
- Part of glusterd, using port 24007
- Only one operation:
 - DUMP



The GlusterD Friend protocol

- Interactions about peer changes
- Part of glusterd, using port 24007
- Some common operations:
 - PROBE_QUERY, ADD, UPDATE



The Gluster Portmap protocol

- Handled by glusterd on port 24007
- glusterfsd (brick) informs glusterd what port is used
- glusterfs (client) retrieves the port for a brick
- Common operations:
 - PORTBYBRICK, SIGNIN, SIGNOUT



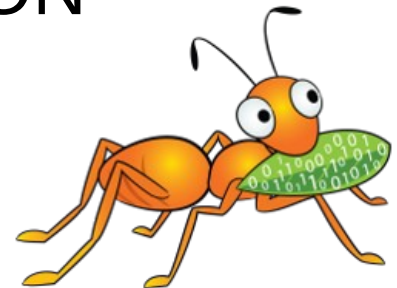
The GlusterFS Callback protocol

- Notifications to glusterd processes
- Handled by glusterd on port 24007
- Only one operation used:
 - FETCHSPEC



The GlusterFS Handshake protocol

- glusterfs (client) processes communicate with glusterfsd (brick) processes
- glusterfs (client) processes request the latest vol-file from glusterd
- Usage management and accounting
 - Client identifies itself when using a brick
- Some common operations:
 - GETSPEC, SETVOLUME, LOCK_VERSION



The GlusterFS protocol

- glusterfs (client) talks to glusterfsd (brick)
- Implements the actual filesystem procedures
- Some common operations:
 - LOOKUP, CREATE, UNLINK, OPEN, RELEASE, READ, WRITE, REaddirP



Capturing network traffic

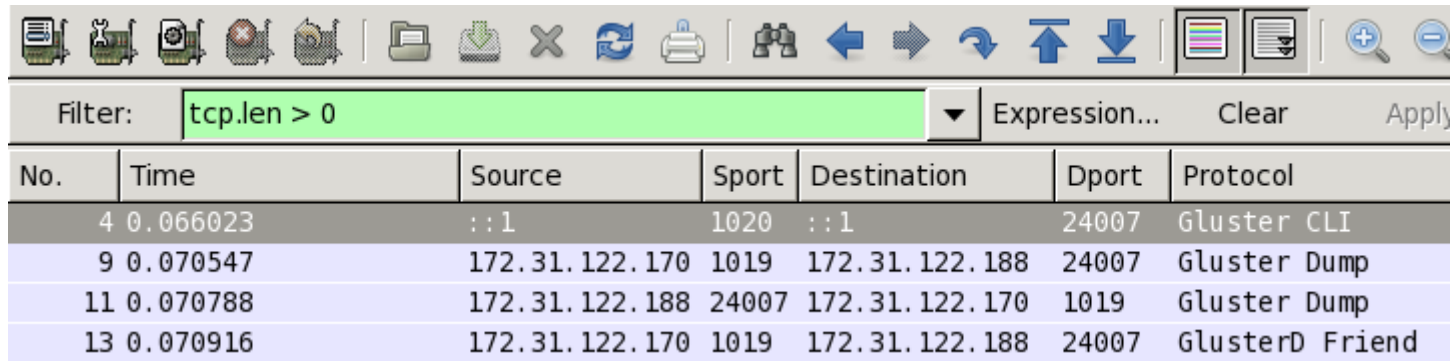
- Use tcpdump:

```
# tcpdump -s 0 -i any -w /tmp/dump.pcap \  
tcp and portrange 24007-24050
```
- When capturing on a storage server:
 - Change the portrange to match the specific bricks
 - Add a filter for the client IP-address
- Use gzip to compress the .pcap file, Wireshark knows how to read .pcap.gz files
- Use editcap to trim big captures



Browsing packets in Wireshark

- Remove all (uninteresting) empty packets
 - Filter: `tcp.len > 0`



The screenshot shows the Wireshark interface with a filter box containing `tcp.len > 0`. Below the filter is a table of captured packets. The table has columns for No., Time, Source, Sport, Destination, Dport, and Protocol.

No.	Time	Source	Sport	Destination	Dport	Protocol
4	0.066023	:::1	1020	:::1	24007	Gluster CLI
9	0.070547	172.31.122.170	1019	172.31.122.188	24007	Gluster Dump
11	0.070788	172.31.122.188	24007	172.31.122.170	1019	Gluster Dump
13	0.070916	172.31.122.170	1019	172.31.122.188	24007	GlusterD Friend

- Filter by protocol
 - Right click on the protocol in the 'Packet Details'
 - Select 'Apply as Filter'
 - Click 'Selected'



Use Case: Adobe InDesign crashes

- Issue:
Adobe InDesign crashes on occasion when editing files on a Samba share
- Error in the log:
0-data-client-1: remote operation failed: Operation not permitted
0-data-client-0: remote operation failed: Operation not permitted
0-glusterfs-fuse: 971574587: SETATTR() /dir/filename.ext => -1 (Operation not permitted)
- Diagnostics:
 - Check file permissions, owner and group
 - Compare with RPC Credentials of the SETATTR procedure

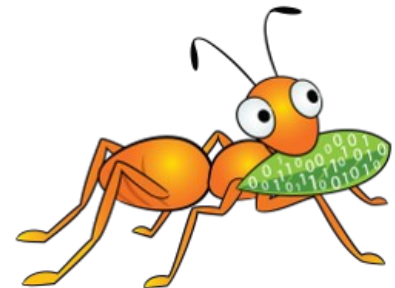


Use Case: Adobe InDesign crashes

- Result:
 - Permissions on the file are too strict
 - SETATTR should indeed be denied
- Solution:

Force permissions through the Samba configuration and open a support case at Adobe to check the return value of the `chmod(2)` call.

More details in the [related article](#) on the Red Hat Customer Portal, or on request.



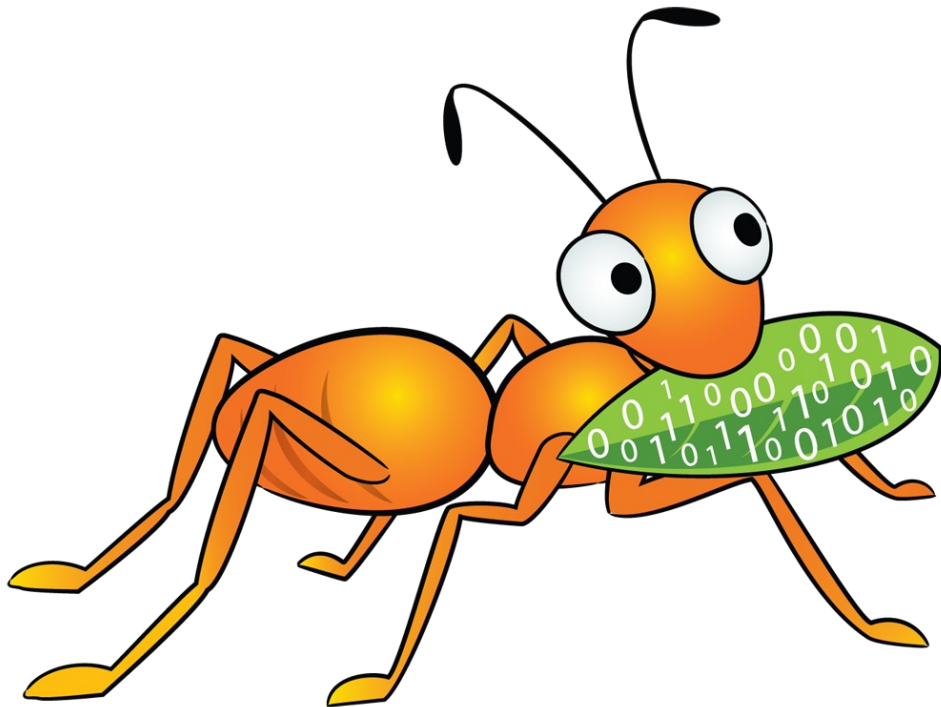
What if decoding fails?

- Wireshark tries to detect protocols by well known port numbers. If STUN, SSL, TURNCHANNEL, PCEP or others are detected, disable these through the 'Analyze' and 'Enabled Protocols' menu.
- Some lesser used procedures are not decoded yet. A few will only show 'Data (.. bytes)' and no further details.
- It may be a bug in Wireshark or a change in the protocol.



Thanks!

Questions, comments etc. are welcome.



Niels de Vos
ndevos@redhat.com
ndevos in #gluster