NFSRODS

Kory Draughn
korydraughn@renci.org
Software Developer, iRODS Consortium

June 25-28, 2019
iRODS User Group Meeting 2019
Utrecht, Netherlands

NFSRODS - Overview

- What
  - A new iRODS client
  - Presents iRODS as NFSv4.1
  - Allows an iRODS collection to be mountable
  - https://github.com/irods/irods_client_nfsrods
- Why
  - Provides a standard POSIX filesystem presentation to existing/legacy tools and applications
  - Provides full iRODS policy layer and enforcement
- How
  - A full nfs4j Virtual File System implementation
  - Implemented using Jargon
  - Deployed as a Docker container

NFSRODS v0.8 Release

Available today …

Provides:

- Authentication: Trusted OS User
- Authorization: Traditional Unix Permissions

NFSRODS - Initial Authentication Model

Initially built with a hard requirement on Kerberos. Why?

- We needed to distinguish users from each other.
- Kerberos provided access to the user's name which is what iRODS needed.
- NFS4J had built-in support for Kerberos.

The Good:

- It worked!
- It had built-in authentication.

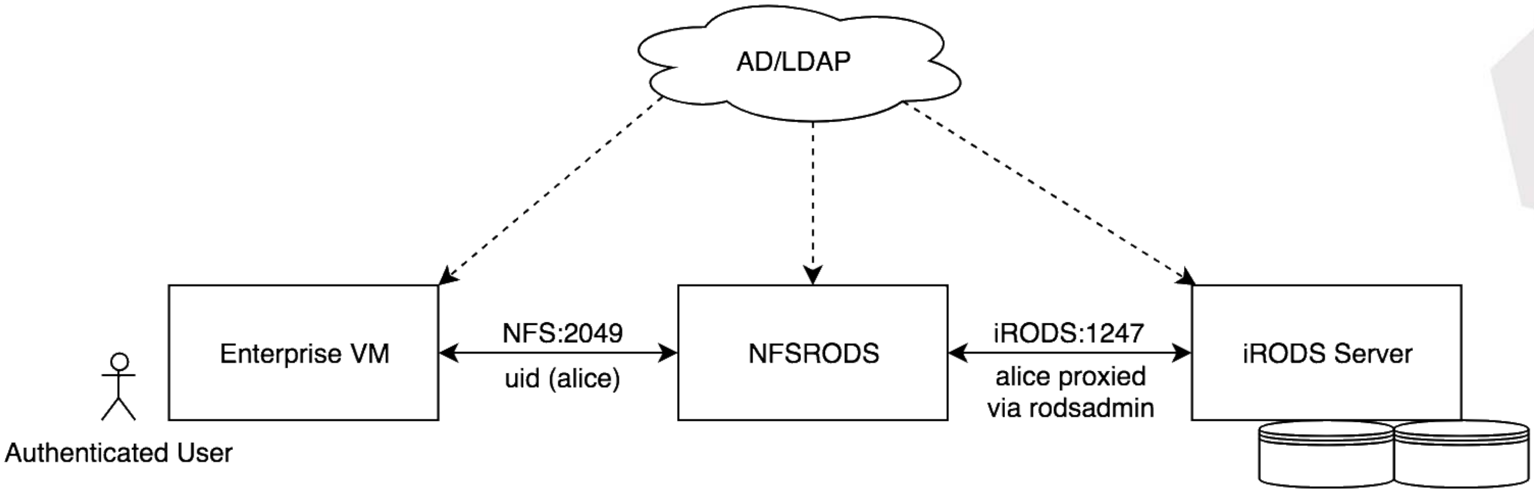
The Bad:

- It was too complex to stand up quickly.
- It required knowledge of Kerberos and all of its tools.
- It couldn't be containerized because of Kerberos/Docker issues.

NFSRODS v0.8 - Current Authentication Model

- Assumptions
  - Authenticated access is via unix user with identically named iRODS user account.
  - Authenticated unix user is traversing the mount point (VM).
  - Entries in **/etc/passwd** and **/etc/shadow** are synced (uids/gids must match) on both the machine with the mount point (VM) and the machine running NFSRODS.

- Note
  - An authenticated user with sudo/root access on VM could appear to iRODS (and, therefore, all policy) as any user.



NFSRODS v0.8 - Authorization Model

- This model maps to traditional Unix permissions
- Permission masks change in real-time depending on who is accessing the mount point.
- Groups are not (yet) supported.
- Collections are always executable, while data objects are never executable.
- iRODS users who have **own** permissions on a collection or data object are mapped into Unix-space as the owner.
- iRODS users who have **read** or **write** permissions are mapped into Unix-space via world permissions.

iRODS Permission	Collection as Directory	Data Object as File
OWN	drwx-----x	-rw-------
WRITE	d--x---rwx	-------rw-
READ	d--x---r-x	-------r--
NULL	d--x-----x	----------

NFSRODS v0.8 - Authorization Model Feedback

After early testing in an enterprise environment ...

The Good:

- Happy with deployment model (Docker)
- Happy with authentication model (Trusting the OS)
- Permissions mapping works for users

Other:

- Groups are missing
- Usage of world permissions was surprising/alarming to sysadmins

Suggestion:

- Can we have extended ACLs (getfacl, setfacl)?

NFSRODS - Deployment

1. Requirements:

- iRODS 4.2.6
- Update Collection MTime Rule Engine Plugin
- Docker

2. Build the image (if desired):

```
ubuntu$ git clone https://github.com/irods/irods_client_nfsrods

ubuntu$ cd irods_client_nfsrods

ubuntu$ docker build -t nfsrods .
```

NFSRODS - Deployment (continued)

3. NFSRODS Configuration:

```
ubuntu$ cat /home/ubuntu/nfsrods_config/server.json
{
    "nfs_server": {
        "port": 2049,
        "irods_mount_point": "/tempZone",
        "user_information_refresh_time_in_minutes": 60,
        "file_information_refresh_time_in_milliseconds": 1000
    },

    "irods_client": {
        "zone": "tempZone",
        "host": "irods-server.ugm-2019",
        "port": 1247,
        "default_resource": "demoResc"
    },

    "irods_proxy_admin_account": {
        "username": "rods",
        "password": "rods"
    }
}
```

NFSRODS - Deployment (continued)

4. Launch the NFSRODS Docker container:

```
ubuntu$ docker run -d --name nfsrods \
        -p 3000:2049 \
        -v /home/ubuntu/nfsrods_config:/nfsrods_config:ro \
        -v /etc/passwd:/etc/passwd:ro \
        -v /etc/shadow:/etc/shadow:ro \
        nfsrods:latest
```

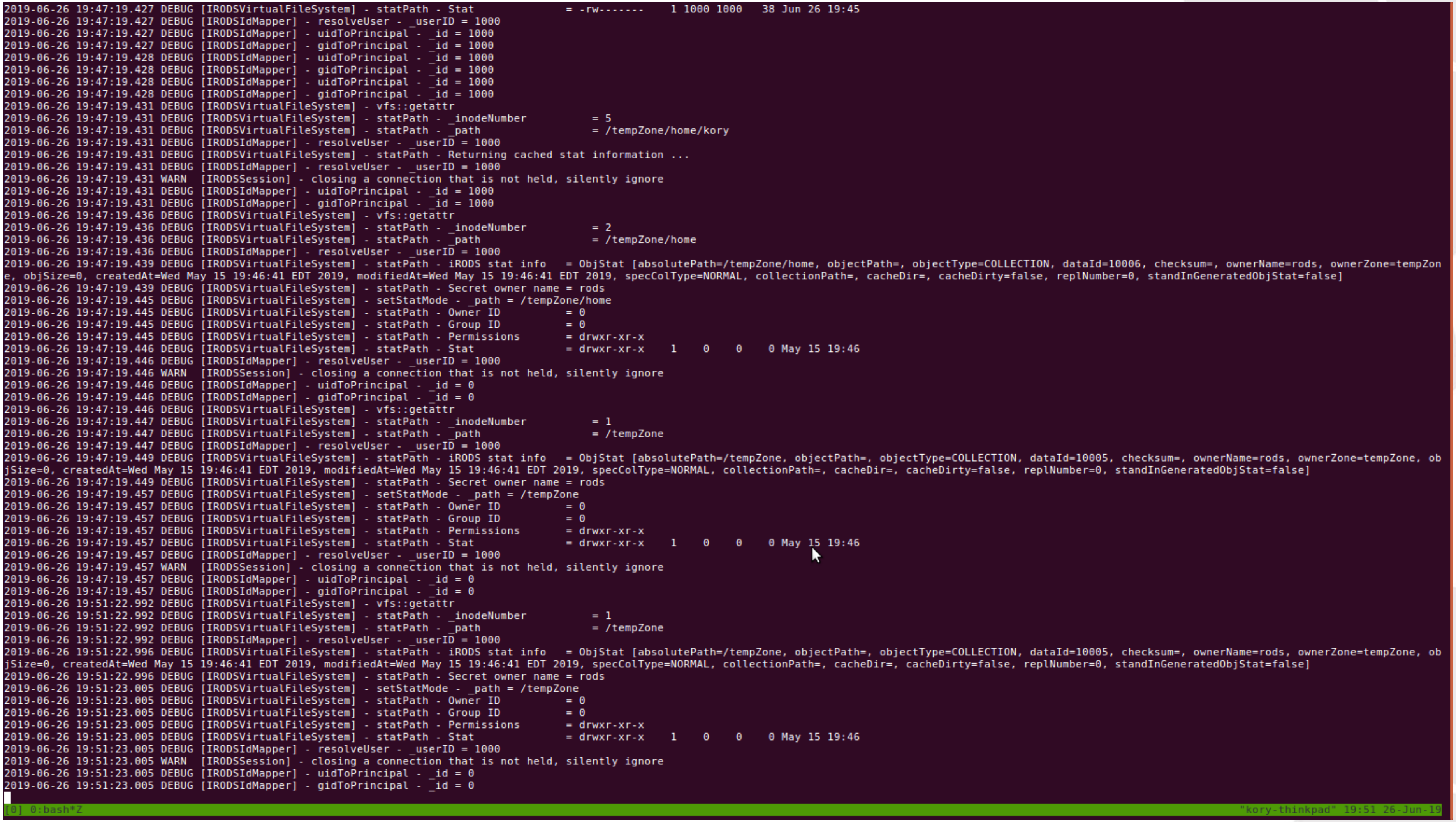
5. Create the mount point:

```
ubuntu$ sudo mkdir -p /mnt/the_nfsrods_mountpoint
ubuntu$ sudo mount -o sec=sys,port=3000 `hostname`:/ /mnt/the_nfsrods_mountpoint
```

6. Use the mount point:

```
bobby$ cd /mnt/the_nfsrods_mountpoint/home/bobby
bobby$ echo "science" > science.txt
bobby$ ls -l science.txt
-rw------- 1 bobby bobby 8 May 15 17:29 science.txt
bobby$ cat science.txt
science
```

NFSRODS v0.8 - Live Demo



```
2019-06-26 19:47:19.427 DEBUG [IRODSVirtualFileSystem] - statPath - Stat              = -rw-------    1 1000 1000   38 Jun 26 19:45
2019-06-26 19:47:19.427 DEBUG [IRODSIdMapper] - resolveUser - _userID = 1000
2019-06-26 19:47:19.427 DEBUG [IRODSIdMapper] - uidToPrincipal - _id = 1000
2019-06-26 19:47:19.427 DEBUG [IRODSIdMapper] - gidToPrincipal - _id = 1000
2019-06-26 19:47:19.428 DEBUG [IRODSIdMapper] - uidToPrincipal - _id = 1000
2019-06-26 19:47:19.428 DEBUG [IRODSIdMapper] - gidToPrincipal - _id = 1000
2019-06-26 19:47:19.428 DEBUG [IRODSIdMapper] - uidToPrincipal - _id = 1000
2019-06-26 19:47:19.428 DEBUG [IRODSIdMapper] - gidToPrincipal - _id = 1000
2019-06-26 19:47:19.431 DEBUG [IRODSVirtualFileSystem] - vfs::getattr
2019-06-26 19:47:19.431 DEBUG [IRODSVirtualFileSystem] - statPath - _inodeNumber          = 5
2019-06-26 19:47:19.431 DEBUG [IRODSVirtualFileSystem] - statPath - _path                 = /tempZone/home/kory
2019-06-26 19:47:19.431 DEBUG [IRODSIdMapper] - resolveUser - _userID = 1000
2019-06-26 19:47:19.431 DEBUG [IRODSVirtualFileSystem] - statPath - Returning cached stat information ...
2019-06-26 19:47:19.431 DEBUG [IRODSIdMapper] - resolveUser - _userID = 1000
2019-06-26 19:47:19.431 WARN  [IRODSSession] - closing a connection that is not held, silently ignore
2019-06-26 19:47:19.431 DEBUG [IRODSIdMapper] - uidToPrincipal - _id = 1000
2019-06-26 19:47:19.431 DEBUG [IRODSIdMapper] - gidToPrincipal - _id = 1000
2019-06-26 19:47:19.436 DEBUG [IRODSVirtualFileSystem] - vfs::getattr
2019-06-26 19:47:19.436 DEBUG [IRODSVirtualFileSystem] - statPath - _inodeNumber          = 2
2019-06-26 19:47:19.436 DEBUG [IRODSVirtualFileSystem] - statPath - _path                 = /tempZone/home
2019-06-26 19:47:19.436 DEBUG [IRODSIdMapper] - resolveUser - _userID = 1000
2019-06-26 19:47:19.439 DEBUG [IRODSVirtualFileSystem] - statPath - iRODS stat info   = ObjStat [absolutePath=/tempZone/home, objectPath=, objectType=COLLECTION, dataId=10006, checksum=, ownerName=rods, ownerZone=tempZon
e, objSize=0, createdAt=Wed May 15 19:46:41 EDT 2019, modifiedAt=Wed May 15 19:46:41 EDT 2019, specColType=NORMAL, collectionPath=, cacheDir=, cacheDirty=false, replNumber=0, standInGeneratedObjStat=false]
2019-06-26 19:47:19.439 DEBUG [IRODSVirtualFileSystem] - statPath - Secret owner name = rods
2019-06-26 19:47:19.445 DEBUG [IRODSVirtualFileSystem] - setStatMode - _path = /tempZone/home
2019-06-26 19:47:19.445 DEBUG [IRODSVirtualFileSystem] - statPath - Owner ID          = 0
2019-06-26 19:47:19.445 DEBUG [IRODSVirtualFileSystem] - statPath - Group ID          = 0
2019-06-26 19:47:19.445 DEBUG [IRODSVirtualFileSystem] - statPath - Permissions       = drwxr-xr-x
2019-06-26 19:47:19.446 DEBUG [IRODSVirtualFileSystem] - statPath - Stat              = drwxr-xr-x    1    0    0    0 May 15 19:46
2019-06-26 19:47:19.446 DEBUG [IRODSIdMapper] - resolveUser - _userID = 1000
2019-06-26 19:47:19.446 WARN  [IRODSSession] - closing a connection that is not held, silently ignore
2019-06-26 19:47:19.446 DEBUG [IRODSIdMapper] - uidToPrincipal - _id = 0
2019-06-26 19:47:19.446 DEBUG [IRODSIdMapper] - gidToPrincipal - _id = 0
2019-06-26 19:47:19.446 DEBUG [IRODSVirtualFileSystem] - vfs::getattr
2019-06-26 19:47:19.447 DEBUG [IRODSVirtualFileSystem] - statPath - _inodeNumber          = 1
2019-06-26 19:47:19.447 DEBUG [IRODSVirtualFileSystem] - statPath - _path                 = /tempZone
2019-06-26 19:47:19.447 DEBUG [IRODSIdMapper] - resolveUser - _userID = 1000
2019-06-26 19:47:19.449 DEBUG [IRODSVirtualFileSystem] - statPath - iRODS stat info   = ObjStat [absolutePath=/tempZone, objectPath=, objectType=COLLECTION, dataId=10005, checksum=, ownerName=rods, ownerZone=tempZone, ob
jSize=0, createdAt=Wed May 15 19:46:41 EDT 2019, modifiedAt=Wed May 15 19:46:41 EDT 2019, specColType=NORMAL, collectionPath=, cacheDir=, cacheDirty=false, replNumber=0, standInGeneratedObjStat=false]
2019-06-26 19:47:19.449 DEBUG [IRODSVirtualFileSystem] - statPath - Secret owner name = rods
2019-06-26 19:47:19.457 DEBUG [IRODSVirtualFileSystem] - setStatMode - _path = /tempZone
2019-06-26 19:47:19.457 DEBUG [IRODSVirtualFileSystem] - statPath - Owner ID          = 0
2019-06-26 19:47:19.457 DEBUG [IRODSVirtualFileSystem] - statPath - Group ID          = 0
2019-06-26 19:47:19.457 DEBUG [IRODSVirtualFileSystem] - statPath - Permissions       = drwxr-xr-x
2019-06-26 19:47:19.457 DEBUG [IRODSVirtualFileSystem] - statPath - Stat              = drwxr-xr-x    1    0    0    0 May 15 19:46
2019-06-26 19:47:19.457 DEBUG [IRODSIdMapper] - resolveUser - _userID = 1000
2019-06-26 19:47:19.457 WARN  [IRODSSession] - closing a connection that is not held, silently ignore
2019-06-26 19:47:19.457 DEBUG [IRODSIdMapper] - uidToPrincipal - _id = 0
2019-06-26 19:47:19.457 DEBUG [IRODSIdMapper] - gidToPrincipal - _id = 0
2019-06-26 19:51:22.992 DEBUG [IRODSVirtualFileSystem] - vfs::getattr
2019-06-26 19:51:22.992 DEBUG [IRODSVirtualFileSystem] - statPath - _inodeNumber          = 1
2019-06-26 19:51:22.992 DEBUG [IRODSVirtualFileSystem] - statPath - _path                 = /tempZone
2019-06-26 19:51:22.992 DEBUG [IRODSIdMapper] - resolveUser - _userID = 1000
2019-06-26 19:51:22.996 DEBUG [IRODSVirtualFileSystem] - statPath - iRODS stat info   = ObjStat [absolutePath=/tempZone, objectPath=, objectType=COLLECTION, dataId=10005, checksum=, ownerName=rods, ownerZone=tempZone, ob
jSize=0, createdAt=Wed May 15 19:46:41 EDT 2019, modifiedAt=Wed May 15 19:46:41 EDT 2019, specColType=NORMAL, collectionPath=, cacheDir=, cacheDirty=false, replNumber=0, standInGeneratedObjStat=false]
2019-06-26 19:51:22.996 DEBUG [IRODSVirtualFileSystem] - statPath - Secret owner name = rods
2019-06-26 19:51:23.005 DEBUG [IRODSVirtualFileSystem] - setStatMode - _path = /tempZone
2019-06-26 19:51:23.005 DEBUG [IRODSVirtualFileSystem] - statPath - Owner ID          = 0
2019-06-26 19:51:23.005 DEBUG [IRODSVirtualFileSystem] - statPath - Group ID          = 0
2019-06-26 19:51:23.005 DEBUG [IRODSVirtualFileSystem] - statPath - Permissions       = drwxr-xr-x
2019-06-26 19:51:23.005 DEBUG [IRODSVirtualFileSystem] - statPath - Stat              = drwxr-xr-x    1    0    0    0 May 15 19:46
2019-06-26 19:51:23.005 DEBUG [IRODSIdMapper] - resolveUser - _userID = 1000
2019-06-26 19:51:23.005 WARN  [IRODSSession] - closing a connection that is not held, silently ignore
2019-06-26 19:51:23.005 DEBUG [IRODSIdMapper] - uidToPrincipal - _id = 0
2019-06-26 19:51:23.005 DEBUG [IRODSIdMapper] - gidToPrincipal - _id = 0

[0] 0:bash*Z                                                                                                                                                              "kory-thinkpad" 19:51 26-Jun-19
```

NFSRODS - Science. Done.

GREAT!!!

Let's run all of our existing tools against NFSRODS, right?

NFSRODS - Science. Done.

Well ...

NFSRODS v0.8 - Caveats

- Speed
  - NFSRODS slower than using direct clients (e.g. iCommands)

- Caching
  - NFS caches file/directory information between all requests
  - Possible information leakage
  - Possible out-of-date information
  - Increasing consistency decreases speed

Consider passing **lookupcache=none** as an additional option to **mount**. Although NFSRODS will be less responsive, the benefit to using this is that information will be more consistent and less likely to be leaked to users with more restrictive access.

NFSRODS - Future Work

- NFS 4.1 Access Control List (ACL) support
  - Standardized
  - Could enable support for groups
  - Removes the need for world permissions
  - Provides more granular control
- Parallel Transfer
- Unit Testing
- NFStest - POSIX Filesystem Level Access Testing
- Samba/CIFS - NFSRODS provides the reference implementation for making iRODS accessible to Microsoft Windows machines

Questions?

- Thank you!

- This version (NFSv4.1) of NFSRODS was built by:
  - Kory Draughn, iRODS Consortium
  - Alek Mieczkowski, iRODS Consortium
  - Mike Conway, NIH/NIEHS
  - Jason Coposky, iRODS Consortium
  - Terrell Russell, iRODS Consortium

- Inspired by work (NFSv3) presented at UGM2016 (slides, paper):
  - Danilo Oliveira, Center for Informatics UFPE, Brazil
  - I. Fé, Center for Informatics UFPE, Brazil
  - A. Lobo Jr., Center for Informatics UFPE, Brazil
  - F. Silva, Center for Informatics UFPE, Brazil
  - G. Callou, Center for Informatics UFPE, Brazil
  - V. Alves, Center for Informatics UFPE, Brazil
  - P. Maciel, Center for Informatics UFPE, Brazil
  - Stephen Worth, EMC Corporation

- Preliminary testing and feedback provided by:
  - Bristol-Myers Squibb Company