Managing sensitive data across the data lifecycle

Russell Butson, David Eyers
{firstname.lastname}@otago.ac.nz
Rationale

• **Need**: Our goal is to offer managed instances of storage repositories to those researchers that are working with highly **sensitive** data (video, audio, patient records, etc.)

• **Researcher Control**: In particular, we want to facilitate PI control over the teams that work with those data

• **Progressive**: Many types of eResearch projects require collaborative sourcing, management or sharing of sensitive datasets
  – HPC projects have large pools of data to replicate and transport
  – Distributed researchers have larger numbers of smaller files
Current state

- Many of us employ ad hoc approaches when dealing with our research data.
  - Storage: External hard drives, DVD/CDs, memory sticks
  - Sharing: Multiple replications often distributed via email & post

- Very few of us use password protection or encryption

- Even if we do have secure data storage – this is often compromised once we replicate the data for analysis.
The Typical Solution

simple data storage
technology-centric

Digital Files
- Audio
- Video
- Text
- Spreadsheets
A more... Relevant Solution
researcher-centric
focus on data lifecycle
Overview of system
Importance of Input Devices

Source Identifiable | Source Unidentifiable

Data Capture | Data Storage | Data Analysis | Working Papers | Outputs

Cameras
Digital Recorders
Surveys
Interviews
Observations
Records
Overview of system
Data Storage

Data Capture
- Cameras
- Digital Recorders
- Surveys
- Interviews
- Observations
- Records

Data Storage
- Digital Files
  - Audio
  - Video
  - Text
  - Spreadsheets

Data Analysis

Working Papers

Outputs

Source Identifiable

Source Unidentifiable
Overview of system
Analysis

Source Identifiable

Data Capture
Cameras
Digital Recorders
Surveys
Interviews
Observations
Records

Data Storage
Digital Files
Audio
Video
Text
Spreadsheets

Data Analysis
Applications
Atlas
Nvivo
SPSS

Source Unidentifiable

Working Papers
Outputs
Overview of system
Distribution

Source Identifiable

Data Capture
- Cameras
- Digital Recorders
- Surveys
- Interviews
- Observations
- Records

Data Storage
- Digital Files
  - Audio
  - Video
  - Text
  - Spreadsheets

Data Analysis
- Applications
  - Atlas
  - Nvivo
  - SPSS

Working Papers
- Positions
- Papers
- Reports
- Articles
- Case Studies

Outputs
- Positions
- Papers
- Reports
- Articles
- Case Studies

Source Unidentifiable
Overview of system
Focus on Identifiable Data

Data Capture
- Cameras
- Digital Recorders
- Surveys
- Interviews
- Observations
- Records

Data Storage
- Digital Files
  - Audio
  - Video
  - Text
  - Spreadsheets

Data Analysis
- Applications
  - Atlas
  - Nvivo
  - SPSS

Working Papers
- Positions Papers
- Reports
- Articles
- Case Studies

Outputs
- Positions Papers
- Reports
- Articles
- Case Studies
Overview of system
PI - Control

Source Identifiable

Data Capture  Data Storage  Data Analysis  Working Papers  Outputs

Cameras  Digital Recorders  Surveys  Interviews  Observations  Records

A  B  C

PI

Digital Files
Audio
Video
Text
Spreadsheets

Applications
Atlas
Nvivo
SPSS

Positions Papers
Reports
Articles
Case Studies

Source Unidentifiable

= PI management of access-transfer-replication

PI = Principal Investigator/s (responsible for data management under the project ethics approval)

A-I = Various project members

= PI management of access-transfer-replication
Overview of system

Source Identifiable

Data Capture
- Cameras
- Digital Recorders
- Surveys
- Interviews
- Observations
- Records

Data Storage
- Digital Files
  - Audio
  - Video
  - Text
  - Spreadsheets

Data Analysis
- Applications
  - Atlas
  - Nvivo
  - SPSS

Working Papers
- Positions
- Papers
- Reports
- Articles
- Case Studies

Outputs
- Positions
- Papers
- Reports
- Articles
- Case Studies

Source Unidentifiable

PI = Principal Investigator/s (responsible for data management under the project ethics approval)

A-I = Various project members

= PI management of access-transfer-replication
Overview of system
researcher-centric
focus on data lifecycle

Source Identifiable

Data Capture
- Cameras
- Digital Recorders
- Surveys
- Interviews
- Observations
- Records

A

Data Storage
- Digital Files
  - Audio
  - Video
  - Text
  - Spreadsheets

B

Data Analysis
- Applications
  - Atlas
  - Nvivo
  - SPSS

C

Working Papers
- Positions Papers
  - Reports
  - Articles
  - Case Studies

D

Outputs
- Positions Papers
  - Reports
  - Articles
  - Case Studies

E

F

G

H

I

PI = Principal Investigator/s (responsible for data management under the project ethics approval)

A-I = Various project members

= PI management of access-transfer-replication
Technical requirements

• (1) Direct management of storage by researchers
  – Aligned administrative workflow with research workflow
  – Must be convenient to use and simple to understand
    • Gaining buy-in from PIs is crucial

• (2) Security of storage system is well explored
  – Needs to be able to handle sensitive patient data
  – Auditing and management are important partners to overall security

• (3) The resulting eResearch system must interoperate with the NeSI DataFabric (currently “BeSTGRID”)
iRODS is the near-term target

• Integrated Rule Oriented Data Storage (iRODS) is a storage middleware used in NZ, Australia (ARCS), USA,…
  – Gives scaffolding for local techies to instrument distributed storage
  – Simplifies naming, auditing, sharing, replicating, archiving, etc.

• Use of iRODS over SRB and alternatives flows from (3)
  – … but this complicates goal (2): assuring the security of the system

• iRODS has access control, but it is not a core concern
  – So using iRODS for storing sensitive data will pose challenges
  – Even so, likely better than current data management (mal)practice
  – Does the flexibility of iRODS impact on its security? inter-zone trust?
Integrate three layers:

- **Top**: Research users
- **Middle**: Departments adding local rules and authorisation
- **Bottom**: Core, centrally managed service

Aim: best possible separation of concerns
- Although aspects of security are moving out of the core...
Tech demonstrator for goal one

• iRODS lets group owners manage group membership
  – However sysadmins must be involved in creating groups
  – Want group creation & removal by PIs: no centralised management

• Easy to achieve a basic, hacky version
  – Changes to certain resources trigger group management
  – Not fit for integration yet, but have resources to apply over summer

• We actually want roles rather than groups
  – Richer semantics in terms of capturing actual workflow activity
  – iRODS may be able to get this technology from SRB
Research into iRODS security

• Review iRODS security from users through to data access
  – Past SRB certification processes may provide useful guidance

• iRODS policy will emerge for sensitive data
  – e.g. the rules to integrate appropriate audit, replication, overrides

• iRODS’ rules should be amenable to verification
  – Can build independent software to check and author rule sets

• Verification of the iRODS engine itself is much harder
  – Nothing can be proven to be secure, but good tools exist
  – Current implementation: no systematic protection of data access
Verification of iRODS rules

• Built an external rule parser to facilitate rule verification
  – Easier than engineering into iRODS itself
  – Provides a fresh view on the rule structure
    (at the cost of potentially behaving differently)

• Started work with iRODS version 2.5
  – It represents rules that the NZ BeSTGRID DataFabric understands
  – Rule engine has been similar for a few minor versions before, too

• iRODS version 3.0 release is now final
  – This is a relief, for reasons described on the next slide...
Despite looking formal, this grammar is only an informal summary of what iRODS is actually interpreting.

Thus, not much use for doing verification.
• Slight differences in the rule grammar produce entirely different parsing of the more complex rule structures...
  – … although there’s very little use of these complex rules in practice!
  – Parsing has “fall through” to strings

• Grammar description was found to be incomplete
  – Even fairly simple rules hit a missing path (see next slide)
  – Situation gets more complicated for ‘delayed execution’ rules
    • These require parsing the encapsulated, delayed rule
    • However the grammar does not indicate these mechanics at all
iRODS grammar examples

• An example rule highlighting a lack of syntactic beauty:

```ruby
approvalFlagB(*TagFile,*MailTo)||msiMakeQuery(COLL_NAME,META_COLL_ATTR_NAME = 'B Flag' and META_COLL_ATTR_VALUE = '1',*QueryC)##msiExecStrCondQuery(*QueryC,*GenCOut)##forEachExec(*GenCOut,msiGetValByKey(*GenCOut,COLL_NAME,*C))##msiMakeQuery(DATA_NAME,COLL_NAME = '*C',*QueryD)##msiExecStrCondQuery(*QueryD,*GenDOut)##forEachExec(*GenDOut,msiGetValByKey(*GenDOut,DATA_NAME,*D))##assign(*DataVal,*C/*D)##mDExtract(*TagFile,*DataVal)##writeLine(stdout,Metadata Extracted for *C/*D)##msiString2KeyValPair(BFlag=1,*KVPD)##msiAssociateKeyValuePairsToObj(*KVPD,*C/*D,-d)##nop##nop##nop##nop##nop
```

• Rules in core3.irb highlight incompleteness of grammar:

```ruby
?- loadirb('irods2.5/core3.irb').
Parsed: actionDef(acRegisterData,[])[ms(acGetResource,[]),ms(msiRegisterData,[])]ms(nop,[])ms(recover_msiRegisterData,[])]
Could not parse rule: acDeleteData|msiCheckPermission(delete)|msiDeleteData|recover_msiDeleteData
Could not parse rule: acDeleteData|msiCheckOwner|msiDeleteData|recover_msiDeleteData
Parsed: actionDef(acGetResource,[])ne(rescName,null)[ms(nop,[])][ms(nop,[])]
Parsed: actionDef(acGetResource,[])eq(rescName,null)[ms(msiGetDefaultResourceForData,[])][ms(nop,[])]
Parsed: actionDef(acSendMail,[])[][ms(msiSendMail,[\$ARG[0],\$ARG[1],\$ARG[2]])][ms(nop,[])]
```

22
iRODS 3.0 rules

• iRODS version 3.0 includes a new rule engine
  – Syntax is much more like the RuleGen syntax from 2.x
  – However it is more strict than the RuleGen syntax

• **Problem:** the parser in version 3.0 is still hand-coded
  – **Worse:** the parser makes heavy use of C preprocessor macro expansion including use of CONCAT to create identifiers!
  – Has explicit, hand-coded fall-back from new to old rule syntax

• **Irony:** RuleGen tools used YACC/Bison even back in 2.x
  – For easy verification of rules, iRODS should use a parser generator
  – Tools like KLEE may allow us to eke out issues externally, however
Lessons learnt so far

• Many lessons learnt so far in the project
  – Across organisations: many stakeholders, many different interests!
  – Widely varying understanding of the needs and possibilities

• Extending iRODS was easy: PI-driven group management
  – (well... once you’re in the iRODS “headspace”)

• For security hardening, the iRODS rule system (v2.x or v3.x) would benefit from some reengineering
  – (this is an understatement)
  – iRODS should use a formal grammar for rules
  – At least it should be able to output its rule parsing results
Conclusions

• Slow—but real—progress being made including sensitive data in researcher-driven, data-grid infrastructure use
  – Keeping different project stakeholders in sync is a lot of work!

• iRODS provides a practical data-grid starting point:
  – Works “out of the box” in many contexts
  – iRODS’ extensibility and openness really is a benefit
  – However security management will be a significant undertaking
  – Reengineering will likely be beneficial

• We hope to contribute our work back to iRODS
  – From there, hopefully benefit other eResearchers in Australasia
FIN
iRODS v2.5 grammar for rules

- original table from the iRODS book
- Note to self:
  - heading line is supposed to be a rule!
  - inconsistent ::= meaning alternative or not
  - weird case

<table>
<thead>
<tr>
<th>Rule</th>
<th>::= actionDef</th>
<th>condition</th>
<th>workflow-chain</th>
<th>recovery-chain</th>
</tr>
</thead>
<tbody>
<tr>
<td>actionDef</td>
<td>::= actionName</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>action</td>
<td>::= actionName</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>actionName</td>
<td>::= actionName(arg1, ..., argn)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Micro-service</td>
<td>::= msName</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>msName</td>
<td>::= alpha-numeric string</td>
<td>/* pre-defined and compiled */</td>
<td></td>
<td></td>
</tr>
<tr>
<td>condition</td>
<td>::= log-expr</td>
<td>/* can be empty */</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log-expr</td>
<td>::= 1</td>
<td>/* true */</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>::= 0</td>
<td>/* false */</td>
<td></td>
<td></td>
</tr>
<tr>
<td>expr == expr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>expr &gt; expr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>expr &lt; expr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>expr &gt;= expr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>expr &lt;= expr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>expr != expr</td>
<td></td>
<td>/* not equal */</td>
<td></td>
<td></td>
</tr>
<tr>
<td>expr like reg-expr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>expr not like reg-expr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>expr</td>
<td>::= string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>number</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$-variable</td>
<td></td>
<td>/* session variable */</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*-variable</td>
<td></td>
<td>/* state variable */</td>
<td></td>
<td></td>
</tr>
<tr>
<td>concatenation of string, $-variables and/or *-variables</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reg-expr</td>
<td>::= regular-expression-string</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>argx</td>
<td>::= expr</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>paramx</td>
<td>::= *-variable</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>number</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>workflow-chain</td>
<td>::= Micro-service</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>action</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>workflow-chain ## workflow-chain</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>recovery-chain</td>
<td>::= workflow-chain</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>