Mandatory Access Control in Pyrrho DBMS

1. What it says in the Pyrrho manual

From December 2018 the DBMS offers a simulation of Bell-LaPadula security based on clearance and classification levels D to A: the database owner is the security administrator (see section 7). The support is quite extensive, so this section includes some sample discussion. Some aspects of the Bell-LaPadula system are found in current DBMS: essentially the idea that rows of a table can have hidden multi-level security labels that control who can access the rows (and different rows in a table can have different labels).

The access control system is based on the concept of security levels, which are conventionally labelled D to A. Level D is the default and corresponds to no access control beyond the permissions described in the above sections. In the US Department of Defense Orange Book, Levels B and C have subdivisions based on the level of auditing available: since Pyrrho always audits levels above D, its levels C and B roughly equate to levels C2 and B3. Level A requires mathematical proof, which would probably be possible, but is not further discussed here. In addition a security label can contain two lists of identifiers here called groups and references, that are visible only to the security administrator (SA), for the purpose of fine-tuning the authorisations of individual users in individual tables.

A user can be assigned a range of levels[1] called *clearance*, and tables and data records in the database can be assigned a level called *classification*. Initially all users have clearance level D (to D). As mentioned above, both clearance and classification can have lists of groups and references (see syntax below). The clearance and classification labels include the level and two sets of identifiers called here groups and references.

The database owner plays the role of security administrator SA for all objects and users of the database. The database owner has special privileges: to consult all system tables including logs, to access and modify the clearance and classification of users and tables and data records, and to specify the enforcement of these rules for tables in the database. By default, all operations on a table are enforced, but these can be limited to some combination of read/insert/update/delete.

The access rules for users other than the database owner are as follows (where the levels are ordered so that D is the lowest and A the highest). Subject to the normal SQL permissions and the enforcement policy

- A user with clearance x can access data with classification y iff x>=y
- A user with clearance x can change or create data only with classification x

In addition, the list of references in the user's clearance must include all the references mentioned in the object's classification (if any); and the list of groups in the user's clearance must include at least one of the groups mentioned in the object's classification (if any). The second bullet point above means for example that some users will be able to see objects they are not allowed to modify. If a user inserts a new record in a table where insert is subject to enforcement, the new record will have a classification with the user's minimum level, the subset of the user's groups that are in the table's classification, and all the table's references (which must be a subset of the user's references).

The database owner (as security administrator) is exempt from these access rules. The database owner rcan specify the classification label for a new table or record. By default a new row will have the same classification as the table that receives it. When called directly or indirectly by the SA, triggers and stored procedures follow the usual (definer's) rules. The SA can also determine for each table whether to apply the access rules just for some combination of read, update, insert and delete operations (by default they are applied for all operations).

[1] A range of levels as a user clearance means that the user is free to read material at a high level and trusted to create at a lower level of security (the minimum they can access), and they can update an object whose classification is in their range (its classification does not change).

The SA can use syntax for level and enforcement descriptors: (as usual [] indicate optional, {} a sequence).

```
Level = LEVEL id ['-'id] [GROUPS {id}] [REFERENCES {id}] .
```

```
Enforcement = SCOPE [READ] [UPDATE] [INSERT] [DELETE]
```

where the level id is one of the letters D to A.

The SA can add Level and Enforcement to a CREATE or ALTER for tables, specify Level in an INSERT statement or when defining columns, and use SECURITY as a pseudo column in SELECT, UPDATE and DELETE statements.

The SA can assign a clearance level to a user with the following extension to the GRANT statement:

```
GRANT Level TO user_id
```

where the user id normally requires to be enclosed in double quotes. The clearance level takes effect immediately on commit, but because of Pyrrho's approach to transaction isolation ongoing transactions will not be affected.

Where a user is unable to access some data because of classification, such data is silently excluded from any direct or indirect computation by that user. If specifically requested information is thus hidden, the requestor will be told that the objects are undefined or that the data is not found. Other exceptions raised by the operation of these rules contain only the information "access denied" (e.g. if a user has been prevented from updating something they have successfully accessed).

There are several system tables that allow the SA to monitor the operation of the above mechanisms. Actions by the SA are visible in the Log$ table and there are separate tables (Log$Clearance, Log$Classify and Log$Enforcement) that allow SQL access to details of the direct and indirect actions taken by the SA to alter clearance or classification. The current status of all clearances, classified rows, classified columns, and enforcement is available to the SA in the Sys$Clearance, Sys$Classification, Sys$ClassifiedColumnData and Sys$Enforcement table, respectively, where such status is different from the default.

2. Detailed algorithms for Mandatory access control

As usual, implementation of rules throws up unexpected complications. Tables have classifications as do the records they contain, and the interplay between them and user clearances is far from simple. The main purpose of the classification information for a table is to specify the set of groups and references that will apply to records classified above D. It can also specify a minimum clearance level for access to the table. The SA can completely specify or modify the classification of any record in the table (but for best results should use subsets of the groups and references that they have specified for the table).

I have the following for users other than the SA in my first implementation. (As usual in Pyrrho, any exception will roll back the transaction.)

Read

1. If the user does not have select privilege on any of the columns selected or select * has been specified and the user does not have select privilege for any columns, throw an informative exception (such as "User cannot select column x", or "user cannot access any columns").

2. If Select is enforced and the user's clearance level does not exceed the table's classification level, report that the table does not exist.

    Even if the table contains rows to which the user's clearance would give them access.

3. If Select is enforced by the table and the user's clearance does not allow access to a given record, skip the record.

4. If Select is enforced and any records with classification above D are accessed, an audit record is added to the database immediately, whether or not the user's transaction commits.

> This cannot be handled within the ReadConstraint mechanism since ReadConstraints only apply in explicit transaction. The context should record what audit records have already been written to avoid repetition within the same context.

Insert

Apart from actions by the SA:

1. If Insert is enforced by the table and the user does not have insert privilege or the user's clearance does not exceed the table's classification, throw an Access Denied exception.

2. If Insert is enforced by the table and the user has insert privilege, construct a record whose classification is equal to the user's clearance, and insert it.

> The new record's classification label will have the user's minimum clearance level: if this is above D, the groups will be the subset of the user's groups that are in the table classification, and the references will be the same as the table (a subset of the user's references).

3. If Insert is not enforced and the user has insert privilege, the record inserted will have level D classification.

Update

Apart from actions by the SA:

1. If the user does not have update privilege for the table, throw an Access Denied exception.

2. If Update is not enforced the record's classification will be unaffected (presumably it will be level D).

3. If Update is enforced by the table and the user's clearance does not allow access to the table, throw an Access Denied exception.

> Even if the update would access records that would match the user's clearance.

4. If Update is enforced by the table, and a record selected for update is not one to which the user has clearance or does not match the user's clearance level, throw an Access Denied exception.

> Even if the user has a higher clearance than the record's classification.

5. The updated record must have the same classification as the old.

Delete

1. If the user does not have delete permission for the table, throw and Access Denied exception.

> Even if the user has a high security clearance.

2. If Delete is enforced by the table for the table or the user's clearance does not exceed the table's classification, throw an Access Denied exception.

> Even if the delete would actually only remove records that match the user's clearance.

3. If Delete is enforced by the table and the user has delete privilege for the table, but the record to be deleted has a classification level different from the user or the clearance does not allow access to the record, throw an Access Denied exception.

Even if the delete is attempting to remove an unclassified record.

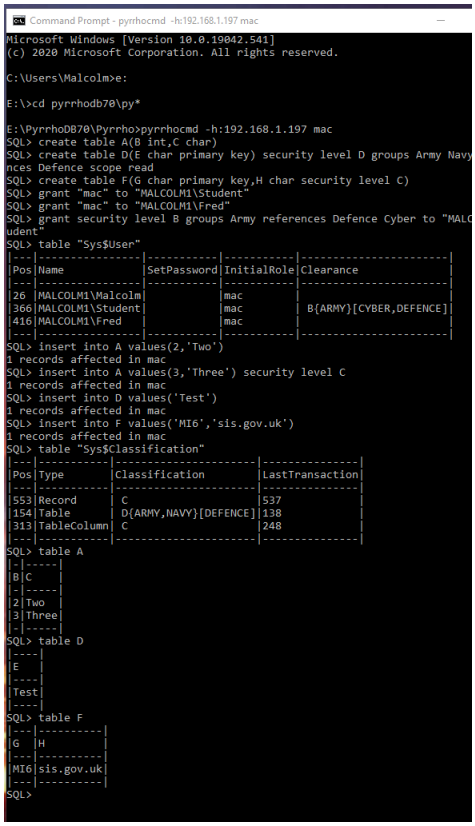
3. An example

In this example, the server is running on MALCOLM2, and the client accounts are all on the MALCOLM1 machine. Apart from Malcolm himself, there are accounts Fred and Student. We start without the database mac: on creation the server automatically adds a role mac and grants it to Malcolm, who becomes the database owner (and therefore the security administrator).

A. Logged in with MALCOLM1\Malcolm (not the server account)

1. Starting with empty database mac

```
SQL> create table A(B int,C char)
SQL> create table D(E char primary key) security level D groups Army Navy
references Defence scope read
SQL> create table F(G char primary key,H char
security level C)
```



2. Create some users with and without clearance

```
SQL> grant "mac" to "MALCOLM1\Student"
SQL> grant "mac" to "MALCOLM1\Fred"
SQL> grant security level B groups Army references
Defence Cyber to "MALCOLM1\Student"
SQL> table "Sys$User"
```

Pos	Name	SetPassword	InitialRole	Clearance
26	MALCOLM1\Malcolm		mac	
366	MALCOLM1\Student		mac	B{ARMY}[CYBER,DEFENCE]
416	MALCOLM1\Fred		mac	

3. Add some rows with and without classification

```
SQL> insert into A values(2,'Two')
1 records affected in mac
SQL> insert into A values(3,'Three') security level C
1 records affected in mac
SQL> insert into D values('Test')
1 records affected in mac
SQL> insert into F values('MI6','sis.gov.uk')
1 records affected in mac
SQL> table "Sys$Classification"
```

Pos	Type	Classification	LastTransaction
553	Record	C	537
154	Table	D{ARMY,NAVY}[DEFENCE]	138
313	TableColumn	C	248

4. Check we can see two rows in A, one row in D and two columns in F

```
SQL> table A
|-|-----|
|B|C    |
|-|-----|
|2|Two  |
|3|Three|
|-|-----|
SQL> table D
|----|
|E   |
|----|
|Test|
|----|
SQL> table F
|---|----------|
|G  |H         |
|---|----------|
|MI6|sis.gov.uk|
|---|----------|
```

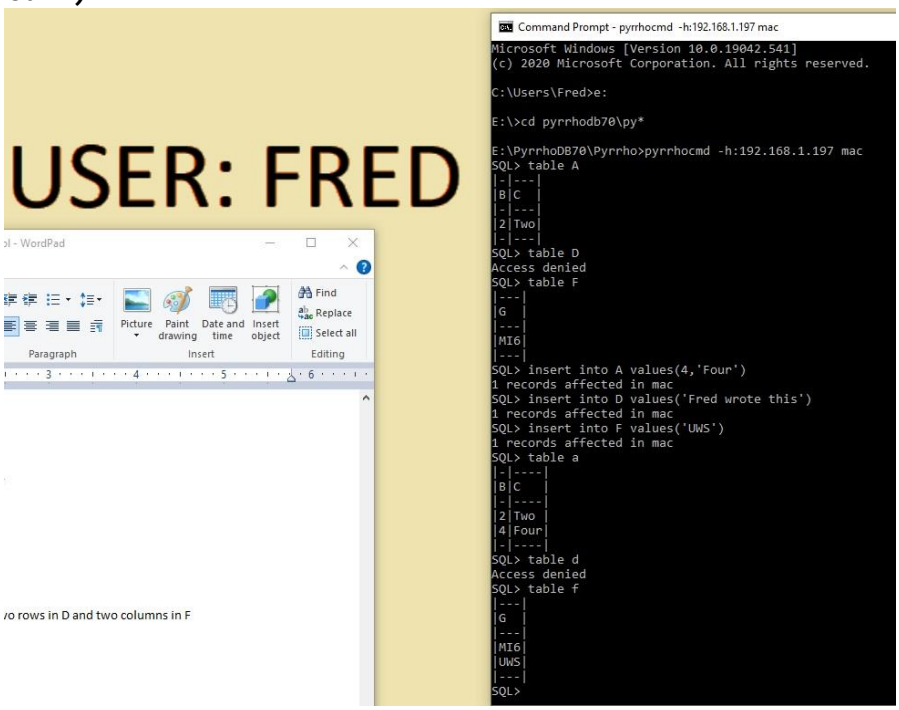
B. Logged in as Fred

5. Check we can only see one row in A, one column in F, and nothing in D

```
SQL> table A
|-|---|
|B|C  |
|-|---|
|2|Two|
|-|---|
SQL> table D
Access denied
SQL> table F
|---|
|G  |
|---|
|MI6|
|---|
```

6. Check we can add a row in A, D and F

```
SQL> insert into A values(4,'Four')
1 records affected in mac
SQL> insert into D
values('Fred wrote this')
1 records affected in mac
SQL> insert into F
values('UWS')
1 records affected in mac
SQL> table a
|-|----|
|B|C   |
|-|----|
|2|Two |
|4|Four|
|-|----|
SQL> table d
Access denied
SQL> table f
|---|
|G  |
|---|
|MI6|
```



```
|UWS|
|---|
```

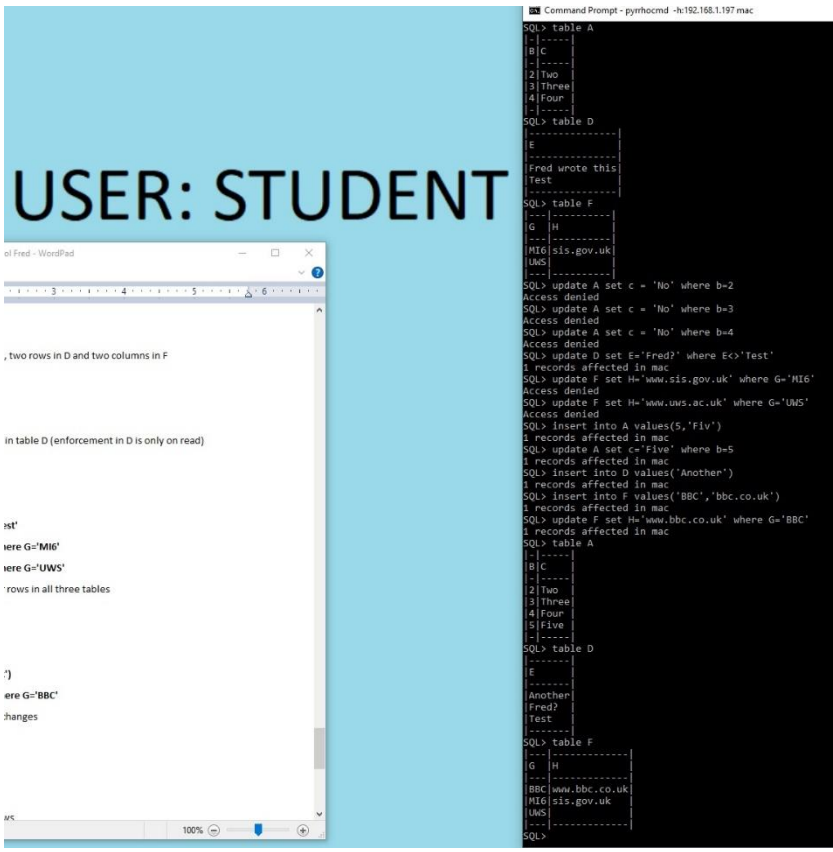
C. Logged in as Student

7. Check we can see three rows in A, two rows in D and two columns in F

```
SQL> table A
|-|-----|
|B|C    |
|-|-----|
|2|Two  |
|3|Three|
|4|Four |
|-|-----|
SQL> table D
|---------------|
|E              |
|---------------|
|Fred wrote this|
|Test           |
|---------------|
SQL> table F
|---|----------|
|G  |H         |
|---|----------|
|MI6|sis.gov.uk|
|UWS|          |
|---|----------|
```

8. Check we can only make changes in table D (enforcement in D is only on read)

```
SQL> update A set c = 'No' where b=2
Access denied
SQL> update A set c = 'No' where b=3
Access denied
SQL> update A set c = 'No' where
b=4
Access denied
SQL> update D set E='Fred?' where
E<>'Test'
1 records affected in mac
SQL> update F set
H='www.sis.gov.uk' where G='MI6'
Access denied
SQL> update F set
H='www.uws.ac.uk' where G='UWS'
Access denied
```



9. Check we can add and update our rows in all three tables

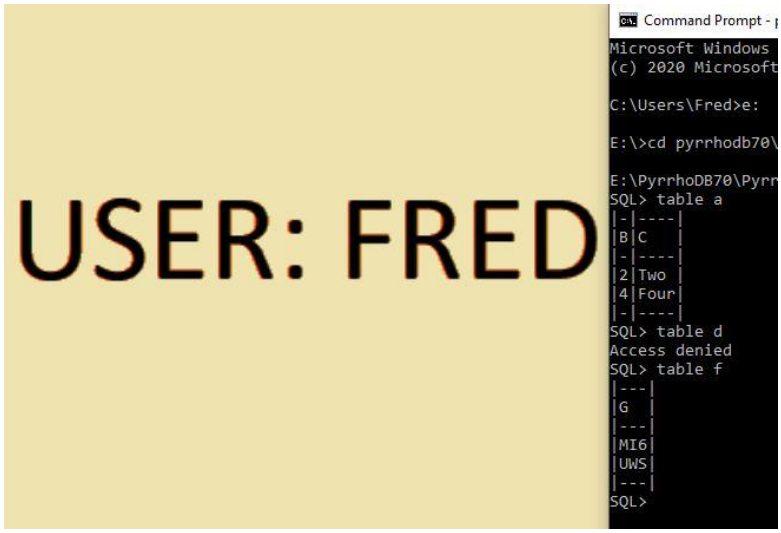
```
SQL> insert into A
values(5,'Fiv')
1 records affected in mac
SQL> update A set c='Five' where
b=5
1 records affected in mac
SQL> insert into D
values('Another')
```

```
1 records affected in mac
SQL> insert into F values('BBC','bbc.co.uk')
1 records affected in mac
SQL> update F set H='www.bbc.co.uk' where G='BBC'
1 records affected in mac
```

10. Check we can see our rows and changes

```
SQL> table A
|-|-----|
|B|C    |
|-|-----|
|2|Two  |
|3|Three|
|4|Four |
|5|Five |
|-|-----|
SQL> table D
|-------|
|E      |
|-------|
|Another|
|Fred?  |
|Test   |
|-------|
SQL> table F
|---|-------------|
|G  |H            |
|---|-------------|
|BBC|www.bbc.co.uk|
|MI6|sis.gov.uk   |
|UWS|             |
|---|-------------|
SQL>
```

D. Logged in as Fred



11. Check Fred can't see the new rows

```
SQL> table a
|-|----|
|B|C   |
|-|----|
|2|Two |
|4|Four|
|-|----|
SQL> table d
Access denied
SQL> table f
|---|
|G  |
|---|
|MI6|
|UWS|
|---|
SQL>
```

E. Logged in as database owner

12. Check all tables including the security information

```
SQL> select B,C,security from A
|-|-----|--------|
|B|C    |SECURITY|
|-|-----|--------|
|2|Two  |        |
|3|Three|        |
|4|Four | C      |
|5|Five |        |
|-|-----|--------|
SQL> select E,security from D
|-------|--------|
|E      |SECURITY|
|-------|--------|
|Another|        |
|Fred?  |        |
|Test   |        |
|-------|--------|
SQL> select G,H,security from F
|---|-------------|--------|
|G  |H            |SECURITY|
|---|-------------|--------|
|BBC|www.bbc.co.uk| B      |
|MI6|sis.gov.uk   | B      |
|UWS|             |        |
|---|-------------|--------|
SQL> select * from A where security=level c
|-|-----|
|B|C    |
|-|-----|
|3|Three|
|-|-----|
SQL> update A set security=level B where security=level C
1 records affected in mac
SQL> update F set security=level D where G='BBC'
1 records affected in mac
SQL> table "Sys$Classification"
|----|-----------|---------------------|---------------|
|Pos |Type       |Classification       |LastTransaction|
|----|-----------|---------------------|---------------|
|553 |Record     | B                   |537            |
|1022|Record     | B                   |1005           |
|154 |Table      | D{ARMY,NAVY}[DEFENCE]|138           |
|313 |TableColumn| C                   |248            |
|----|-----------|---------------------|---------------|
```

F. Logged in as Student

13. Check we can still see our row in A

```
SQL> select * from a where b=5
|-|----|
|B|C   |
|-|----|
|5|Five|
|-|----|
```

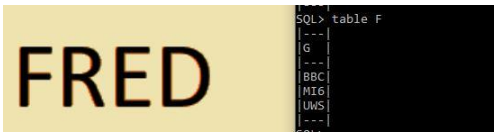
14. Check we can no longer update our rows in A or F

```
SQL> delete from A where b=5
Access denied
SQL> update F set H='bbc.com' where G='BBC'
```

```
Access denied
```

G. Logged in as Fred

15. Check we can see the row about the BBC




```
SQL> table F
|---|
|G  |
|---|
|BBC|
|MI6|
|UWS|
|---|
```

H. Logged in as database owner

16. Check that auditing has been happening

```
SQL> table "Sys$Audit"
|----|---------------|-----|-------------------|
|Pos |User           |Table|Timestamp          |
|----|---------------|-----|-------------------|
|665 |MALCOLM1\Fred  |62   |03/10/2020 10:58:52|
|684 |MALCOLM1\Fred  |62   |03/10/2020 10:59:08|
|824 |MALCOLM1\Fred  |62   |03/10/2020 10:59:21|
|849 |MALCOLM1\Student|62   |03/10/2020 11:00:28|
|868 |MALCOLM1\Student|62   |03/10/2020 11:00:40|
|893 |MALCOLM1\Student|62   |03/10/2020 11:00:40|
|918 |MALCOLM1\Student|62   |03/10/2020 11:00:40|
|986 |MALCOLM1\Student|62   |03/10/2020 11:00:52|
|1050|MALCOLM1\Student|62   |03/10/2020 11:00:52|
|1273|MALCOLM1\Student|62   |03/10/2020 11:01:02|
|1292|MALCOLM1\Fred  |62   |03/10/2020 11:01:32|
|1424|MALCOLM1\Student|62   |03/10/2020 11:02:42|
|1449|MALCOLM1\Student|62   |03/10/2020 11:02:52|
|----|---------------|-----|-------------------|
SQL> table "Sys$AuditKey"
|----|---|---|---|
|Pos |Seq|Col|Key|
|----|---|---|---|
|824 |0  |82 |4  |
|868 |0  |82 |2  |
|893 |0  |82 |3  |
|918 |0  |82 |4  |
|1050|0  |82 |5  |
|1424|0  |82 |5  |
|1449|0  |82 |5  |
|----|---|---|---|
```

17. Finally, here is the complete database log:

```
SQL> table "Log$"
|----|-------------------------------------------------------------------------|------------|-------|
|Pos |Desc                                                                     |Type        |Affects|
|----|-------------------------------------------------------------------------|------------|-------|
|26  |PUser MALCOLM1\Malcolm                                                   |PUser       |26     |
|46  |PTransaction for 3 Role=5 User=26 Time=10/03/2020 10:56:59               |PTransaction|46     |
|62  |PTable A                                                                 |PTable      |62     |
|68  |Domain INTEGER                                                           |PDomain     |68     |
|82  |PColumn3 B for 62(0)[68]                                                 |PColumn3    |82     |
|103 |Domain CHAR                                                              |PDomain     |103    |
|116 |PColumn3 C for 62(1)[103]                                                |PColumn3    |116    |
|138 |PTransaction for 5 Role=5 User=26 Time=10/03/2020 10:56:59               |PTransaction|138    |
|154 |PTable D                                                                 |PTable      |154    |
|161 |PColumn3 E for 154(0)[103]                                               |PColumn3    |161    |
|184 |PIndex D on 154(161) PrimaryKey                                          |PIndex      |184    |
|203 |Classify 154 D{ARMY,NAVY}[DEFENCE]                                       |Classify    |203    |
```

239	Enforcement [154] SCOPE read	Enforcement	239
248	PTransaction for 5 Role=5 User=26 Time=10/03/2020 10:57:00	PTransaction	248
264	PTable F	PTable	264
271	PColumn3 G for 264(0)[103]	PColumn3	271
294	PIndex F on 264(271) PrimaryKey	PIndex	294
313	PColumn3 H for 264(1)[103]	PColumn3	313
337	Classify 313 C	Classify	337
350	PTransaction for 2 Role=5 User=26 Time=10/03/2020 10:57:08	PTransaction	350
366	PUser MALCOLM1\Student	PUser	366
388	Grant UseRole on 5 to 366	Grant	388
400	PTransaction for 2 Role=5 User=26 Time=10/03/2020 10:57:08	PTransaction	400
416	PUser MALCOLM1\Fred	PUser	416
435	Grant UseRole on 5 to 416	Grant	435
447	PTransaction for 1 Role=5 User=26 Time=10/03/2020 10:57:08	PTransaction	447
463	Clearance 366 B{ARMY}[CYBER,DEFENCE]	Clearance	463
500	PTransaction for 1 Role=5 User=26 Time=10/03/2020 10:57:18	PTransaction	500
516	Record 516[62]: 82=2,116=Two	Record	516
537	PTransaction for 1 Role=5 User=26 Time=10/03/2020 10:57:18	PTransaction	537
553	Record3 553[62]: 82=3,116=Three Classification: C	Record3	553
580	PTransaction for 1 Role=5 User=26 Time=10/03/2020 10:57:18	PTransaction	580
596	Record 596[154]: 161=Test	Record	596
615	PTransaction for 1 Role=5 User=26 Time=10/03/2020 10:57:18	PTransaction	615
631	Record 631[264]: 271=MI6,313=sis.gov.uk	Record	631
665	Audit: MALCOLM1\Fred [62] 10/03/2020 10:58:52	Audit	665
684	Audit: MALCOLM1\Fred [62] 10/03/2020 10:59:08	Audit	684
703	PTransaction for 1 Role=5 User=416 Time=10/03/2020 10:59:08	PTransaction	703
720	Record 720[62]: 82=4,116=Four	Record	720
742	PTransaction for 1 Role=5 User=416 Time=10/03/2020 10:59:08	PTransaction	742
759	Record 759[154]: 161=Fred wrote this	Record	759
789	PTransaction for 1 Role=5 User=416 Time=10/03/2020 10:59:10	PTransaction	789
806	Record 806[264]: 271=UWS	Record	806
824	Audit: MALCOLM1\Fred [62] 10/03/2020 10:59:21 {82='4'}	Audit	824
849	Audit: MALCOLM1\Student [62] 10/03/2020 11:00:28	Audit	849
868	Audit: MALCOLM1\Student [62] 10/03/2020 11:00:40 {82='2'}	Audit	868
893	Audit: MALCOLM1\Student [62] 10/03/2020 11:00:40 {82='3'}	Audit	893
918	Audit: MALCOLM1\Student [62] 10/03/2020 11:00:40 {82='4'}	Audit	918
943	PTransaction for 1 Role=5 User=366 Time=10/03/2020 11:00:40	PTransaction	943
960	Update 759[154]: 161=Fred? Prev:759	Update	759
986	Audit: MALCOLM1\Student [62] 10/03/2020 11:00:52	Audit	986
1005	PTransaction for 1 Role=5 User=366 Time=10/03/2020 11:00:52	PTransaction	1005
1022	Record3 1022[62]: 82=5,116=Fiv Classification: B	Record3	1022
1050	Audit: MALCOLM1\Student [62] 10/03/2020 11:00:52 {82='5'}	Audit	1050
1075	PTransaction for 1 Role=5 User=366 Time=10/03/2020 11:00:52	PTransaction	1075
1092	Update 1022[62]: 82=5,116=Five Prev:1022	Update	1022
1120	PTransaction for 1 Role=5 User=366 Time=10/03/2020 11:00:52	PTransaction	1120
1137	Record 1137[154]: 161=Another	Record	1137
1159	PTransaction for 1 Role=5 User=366 Time=10/03/2020 11:00:52	PTransaction	1159
1176	Record3 1176[264]: 271=BBC,313=bbc.co.uk Classification: B	Record3	1176
1213	PTransaction for 1 Role=5 User=366 Time=10/03/2020 11:00:54	PTransaction	1213
1230	Update 1176[264]: 271=BBC,313=www.bbc.co.uk Prev:1176	Update	1176
1273	Audit: MALCOLM1\Student [62] 10/03/2020 11:01:02	Audit	1273
1292	Audit: MALCOLM1\Fred [62] 10/03/2020 11:01:32	Audit	1292
1311	PTransaction for 1 Role=5 User=26 Time=10/03/2020 11:02:10	PTransaction	1311
1327	Update1 553[62]: 82=3,116=Three Classification: B Prev:553	Update1	553
1359	PTransaction for 1 Role=5 User=26 Time=10/03/2020 11:02:10	PTransaction	1359
1375	Update1 1176[264]: 271=BBC,313=www.bbc.co.uk Classification: Prev:1176	Update1	1176
1424	Audit: MALCOLM1\Student [62] 10/03/2020 11:02:42 {82='5'}	Audit	1424
1449	Audit: MALCOLM1\Student [62] 10/03/2020 11:02:52 {82='5'}	Audit	1449