

P2600 subjects, objects, etc. revisited
Brian Smithson, Ricoh Corp
7/18/06

For discussion at Rochester P2600 meeting

Introduction

I have taken another look at how we began to define and use Subjects, Objects, Operations, Interfaces, Users, and Attributes. My purpose was to try to add some rigor to the process of defining the operations of the TOE (and perhaps also to threats and objectives) in order to help make concise SFRs.

The first thing I found was that I don't think we correctly defined our Subjects. Most of what we had considered to be Subjects are, in my opinion, Interfaces. Subjects are supposed to be *active* entities in the TOE, and I think that things like network interfaces and operator panels are passive entities through which Users bind to Subjects.

I tried to define Operations as either (1) execution of some function or (2) permission/denial of access to some kind of data.

The Objects that I defined are mostly what we have defined as TOE assets, except I also needed to include assets in transit (data communications).

Users are pretty much as we defined before, except that I found I needed to add a "fax sender" attribute.

Take a look at the work in progress that follows. It's not complete and I can't say it's anywhere close to perfect, but I'd like to get some discussion and feedback going to see if it's worth continuing along this vein.

Definitions

Subjects

"A *subject* is an active entity in the TOE: subjects perform operations and actions in the TOE. A typical implementation of a subject is a kernel process. Subjects are distinct from active entities outside of the TOE (users)".

APP (*these are the basic HCD applications, plus an undefined applet*)

PRINT

SCAN

FAX

COPY

APPLET

CONFIG (*these are operations to configure/maintain/manage the HCD*)

AUTH
DEV
NET
SEC
AB
LOGS
FW

NET (*this subject represents the network handler, not the network itself*)

Interfaces

IF interface
NET
FAX
LOCAL
MAINT
EXTDEV
OP (*operator panel*)

Objects

“An *object* is a passive entity in the TOE: they are the entities that subjects perform operations on. Typical implementations of objects are a file, a queue, and a database.”

UDD (*user document data*)
HARD
SOFT
TEMP
UFD (*user function data*)
ID (*user identification*)
AB (*address book data*)
MD (*management data*)
DEV (*device configuration*)
SEC (*security configuration*)
NET (*network configuration*)
AUTH (*authentication data*)
LOG (*job, device, event logs – do we need to distinguish these?*)
RES (*resources – I didn't find a use for this yet*)
PROG (*programs*)
FW (*firmware*)
APPLET (*applet*)
COMMS (*communications*)
IN (*established inbound*)
OUT (*established outbound*)

Operations

“An *operation* is a specific type of action of a subject to an object. An example of an operation is ‘modify’. Actions that bring an object into being (e.g. the operation ‘create’) are also considered to be an operation, though strictly speaking, the object does not yet exist.

```
FCN  (functions)
      PRINT
          START
          RELEASE  (release the hardcopy to the user)
          END
      SCAN
          START
          SEND
          END
      FAXOUT
          START
          SEND
          END
      FAXIN
          START
          RELEASE
          END
      COPY
          START
          RELEASE
          END
ACC  (access)
      ALLOW  (applies to COMMS)
      READ   (applies to UFD or MD)
      WRITE  (applies to UFD or MD)
      LOAD   (applies to FW)
      EXEC   (applies to APPLET)
```

User attributes

“A *user* is any active entity outside of the TOE. It is important to realize that the CC does not limit users to the traditional concept of human users. In the CC, a user may be a human or a machine.”

I have only defined a human-oriented user, one who interacts with the TOE to use or maintain it, with attributes listed below that indicate the user’s role. However, I wonder if we’ll need some other kind of user, such as an SNMP client.

```
NORM
DEVADMIN
SECADMIN  (we don’t currently define security administrator separately)
```

NETADMIN
AUDITOR
CUSTENG (*I didn't make a use case for this one yet*)
FAXSENDER

Use cases

Here are some sample use cases that show how these entities fit together in normal TOE operation. Note that if no User or Interface is specified, then the Subject is invoking the Operation without requiring User action or initiation.

Use case	User binding	Interface	Subject	Operation	Object
1. Print a doc over network, with job release			NET	ACC.ALLOW	COMMS.IN
	NORM	NET	APP.PRINT	FCN.PRINT.START	UDD.SOFT
	NORM	OP	APP.PRINT	FCN.PRINT.RELEASE	UDD.HARD
			APP.PRINT	FCN.PRINT.END	UDD.TEMP
2. Copy a doc, no job release	NORM	OP	APP.COPY	FCN.COPY.START	UDD.HARD
			APP.COPY	FCN.COPY.RELEASE	UDD.HARD
			APP.COPY	FCN.COPY.END	UDD.TEMP
3. Fax send from local interface	NORM	LOCAL	APP.FAX	FCN.FAXOUT.START	UDD.SOFT
			APP.FAX	FCN.FAXOUT.SEND	UDD.SOFT
			APP.FAX	FCN.FAXOUT.END	UDD.TEMP
4. Fax send from op panel with address book	NORM	OP	APP.FAX	FCN.FAXOUT.START	UDD.HARD
	NORM	OP	APP.FAX	ACC.READ	UFD.AB
			APP.FAX	FCN.FAXOUT.SEND	UDD.SOFT
			APP.FAX	FCN.FAXOUT.END	UDD.TEMP
5. Fax receive with job release	FAXSENDER	FAX	APP.FAX	FCN.FAXIN.START	UDD.SOFT
	DEVADMIN	OP	APP.FAX	FCN.FAXIN.RELEASE	UDD.HARD
			APP.FAX	FCN.FAXIN.END	UDD.TEMP
6. Scan to email from op panel with addr book	NORM	OP	APP.SCAN	FCN.SCAN.START	UDD.HARD
	NORM	OP	APP.SCAN	ACC.READ	UFD.AB
			NET	ACC.ALLOW	COMMS.OUT
			APP.SCAN	FCN.SCAN.SEND	UDD.SOFT
			APP.SCAN	FCN.SCAN.END	UDD.TEMP
7. Add a user record from remote mgmt tool			NET	ACC.ALLOW	COMMS.IN
	SECADMIN	NET	CONFIG.AUTH	ACC.WRITE	UFD.ID
	SECADMIN	NET	CONFIG.AUTH	ACC.WRITE	MD.AUTH
8. Configure IP address from op panel	NETADMIN	OP	CONFIG.NET	ACC.WRITE	MD.NET
9. Configure port filtering from op panel	SECADMIN	OP	CONFIG.SEC	ACC.WRITE	MD.NET
10. Download and then delete log data over net			NET	ACC.ALLOW	COMMS.IN
	AUDITOR	NET	CONFIG.LOGS	ACC.READ	MD.LOG

	AUDITOR	NET	CONFIG.LOGS	ACC.WRITE	MD.LOG
11. Install new firmware from net			NET	ACC.ALLOW	COMMS.IN
	SECADMIN	NET	CONFIG.FW	ACC.LOAD	PROG.FW
12. User operation that downloads an applet			NET	ACC.ALLOW	COMMS.IN
	NORM	NET	APP.APPLET	ACC.LOAD	PROG.APPLET
			NET	ACC.ALLOW	COMMS.OUT
			APP.APPLET	ACC.EXEC	PROG.APPLET

For example, in use case 1, the following operations occur:

1. The NET network handler ALLOWS an incoming communication session.
2. A user binds as a NORMAL user through the NET network interface to the PRINT application to START printing a user document input in SOFTcopy form.
3. The user then needs to bind as a NORMAL user again at the Operator panel to the PRINT application to RELEASE the document output in HARDcopy form.
4. The PRINT application then ends the print function by cleaning up TEMP data.

For another example, in use case 6, these operations occur:

1. The user binds as NORMAL through the operator panel to SCAN a HARDcopy document.
2. The user (still bound as NORMAL through the operator panel) now accesses the AddressBook to choose a destination.
3. The NET handler ALLOWS the establishment of an outbound connection.
4. The SCAN application then sends the SOFT copy of the document (somewhere – could be email, could be to a file server)
5. The SCAN application then cleans up TEMP files.

As a final example, in use case 10:

1. The NET handler ALLOWS an incoming session to be established.
2. A user binds as AUDITOR to the CONFIG.LOGS application, and reads the contents of the stored logs.
3. The user, still bound as AUDITOR to the CONFIG.LOGS application, deletes the stored logs.

SFR examples

The next step is to try to write some CCv3 SFRs using these entities. Before writing them out in full form, I propose to make a table of some kind that lists the applicable entities for each SFR. I think that will help make a clearer picture of whether or not the entity definitions are correct and sufficient.