Simon Foley,
Department of Computer Science, UCC

CS4615 Computer Systems Security
Assignment2/Test1 - security policy models

Time: 30 mins.

STUDENT NAME	
STUDENT NUMBER	

A multilevel secure system offers a document indexing system with the operations:

- assign(n,p,s): give the document (file) located at path p the name n.
- view(n,s): view the contents of the document with name n.

where s is the subject requesting the operation. Note that document names are unique across the system and are in addition to the name (path) given to the file containing the document. A table is used to store the name-path relationship, for example:

Name	Path
ExamPaper	/home/store/a
Attendance	/home/store/b
LectureNotes	/home/store/c

The system is used to index staff and student documents whereby staff are permitted to access all documents. Students are permitted to access most documents, however, there are certain sensitive documents, such as exam papers, that students may not view.

1. Propose a suitable information flow/partial order policy for this system and provide sample clearances for staff and for students, and classifications for the documents named above. *(2 marks)*

A

- *The information flow policy has a set of classifications* $\{staff, student\}$ *with an ordering relation defined as* $student \leq staff$.
- *Each row/document with name n in the table has a classification security-level(n), for example,*

Name	Path	security-level(Name)
ExamPaper	/home/store/a	staff
Attendance	/home/store/b	staff
LectureNotes	/home/store/c	student

  Note that security-level(n) gives the classification of the information in the given ***row*** *of the table; the classification security-level(p) of the file, at path p on the MLS file system, containing the document may be different.*

- *Sample clearances for users. Student Alice is cleared to classification student which means she can read and write LectureNotes, but cannot read ExamPaper or Attendance.*

  Staff member Bob has clearance $staff$ *which means he can create a process (subject) running at student to read/write lecture notes and can create another process running at student to read/write exam papers and attendance data.*

$\overline{\textbf{A}}$

STUDENT NAME	
STUDENT NUMBER	

2. Sketch suitable algorithms that describe the behavior of the above operations, taking care to ensure that the BLP axioms/multilevel security is preserved. *(2 marks)*

A *Using the Bell-LaPadula axioms, let $\leq$ denote the sensitivity ordering between security levels and assume that every file with path p, and subject s, has security level, security-level(p) and security-level(s), respectively. Need to recognize that creating a document-path relationship is in itself a piece of information that is at the level of the subject s carrying out the assignment and we must therefore extend the table to include a security-level for this relationship.*

```
assign(n,p,s){
    if document with name n exists in table then return null;
    if security-level(p) ≤ security-level(s) then
         add (n,p,security-level(s)) to table;
}

view(n,s){
    retrieve entry (n,p,security-level(n)) for document with name n from table;
    if no entry found then return null;
    if security-level(n) ≤ security-level(s) then
         return p;
    else
         return null;
}
```

$\overline{\textbf{A}}$

STUDENT NAME	
STUDENT NUMBER	

3. Given that document names must be unique in the table, explain how a Trojan-Horse could use this table to establish a covert-channel and signal one bit of information about the sensitive exam paper to a student. (*1 marks*)

A *A covert channel exists because a low-level user can test for the existence of a high level name-path table entry by attempting to assign that name to another document. For example,*

(a) A Trojan-Horse executing in Bob's process (classified at $staff$) wishes to signal a bit of ($staff$) information to Alice's process classified at student.

(b) The Trojan-Horse assigns/inserts a new document with agreed name 1111 *meaning* 1*, or does no insertion meaning* 0*. For example,*

$$\textsf{assign}(1111, /home/store/somefile, processBob)$$

(c) At a later (agreed) point in time the collaborator Alice (a student process with classification student) then attempts to assign the document name 1111 *to some file.*

$$\textsf{assign}(1111, /home/store/somefile, processAlice)$$

If the insertion fails Alice deduces 1*, or* 0 *otherwise.*

The covert channel can be removed by allowing duplicate names for different security levels. In the example above, the unclassified user would be allowed insert a secret tuple with key 1111. $\overline{\textbf{A}}$

Commentary on answers

- Lack of precision in answers.

Q1 should clearly identify the policy (a set of classifications and an ordering), the clearances for users and classifications for objects.

Q1 In general, an MLS policy can be any ordering, depending on the application. Its not always $unclassified \leq secret \leq topsecret$.

Q1 Clearance of user is different to classification of subject.

Q2 The system is a document indexing system built on top of an existing MLS (file) system. It is for assigning document names to existing file paths, not for changing file path names, or for creating files.

Q2 Not sufficient to just give the BLP axioms: they need to be encoded in the algorithms for the operations assign and view.

Q2 Should not encode hard-coded tests in the operations that are based on specific security classifications such as a test: if security-level(name) equals secret, etc.

Q2 Cannot make comparisons like $n \leq securitylevel(s)$, where n is the name of the document. The flow ordering is defined over security classifications, and so comparison must be between security classifications. Ambiguity aside, you would also need to define the security level of the document.

Q2 The security level of the document needs to be stored in the table: the classification of the row corresponds to the classification of the subject that inserted the row and in principle may be different from the level of the actual document. However, if the assign operation you designed required that the security classification of the inserting subject equalled that of the document, then relying on the security class of the document is fine.

Q2 Note that, in general, a comparison $level(p) \ngtr level(s)$ is not the same as $level(p) \leq level(s)$ as it is possible that security classifications are disjoint.

Q3 In MLS a Trojan horse runs at a high-classification/has access to sensitive data. It corresponds to a user, with a subject at a high classification, running malicious/untrusted software. The intention is that the security mechanism should prevent the Trojan horse from leaking the high information it has access to down to a lower level accomplice.

Q3 The answer needs to clearly identify how the particular operation of the assign and view operations can lead to a covert channel. A general discussion of covert channels is not sufficient. Remember, the covert channel is due to the particular design of these operations (assign and view), not of the file system per-se.

Q3 Creating a file with some name is different to creating a document with some name: the file name is something that's managed by the MLS file system, the document name is something that's managed by our document indexing system (and that's where the covert channel is).