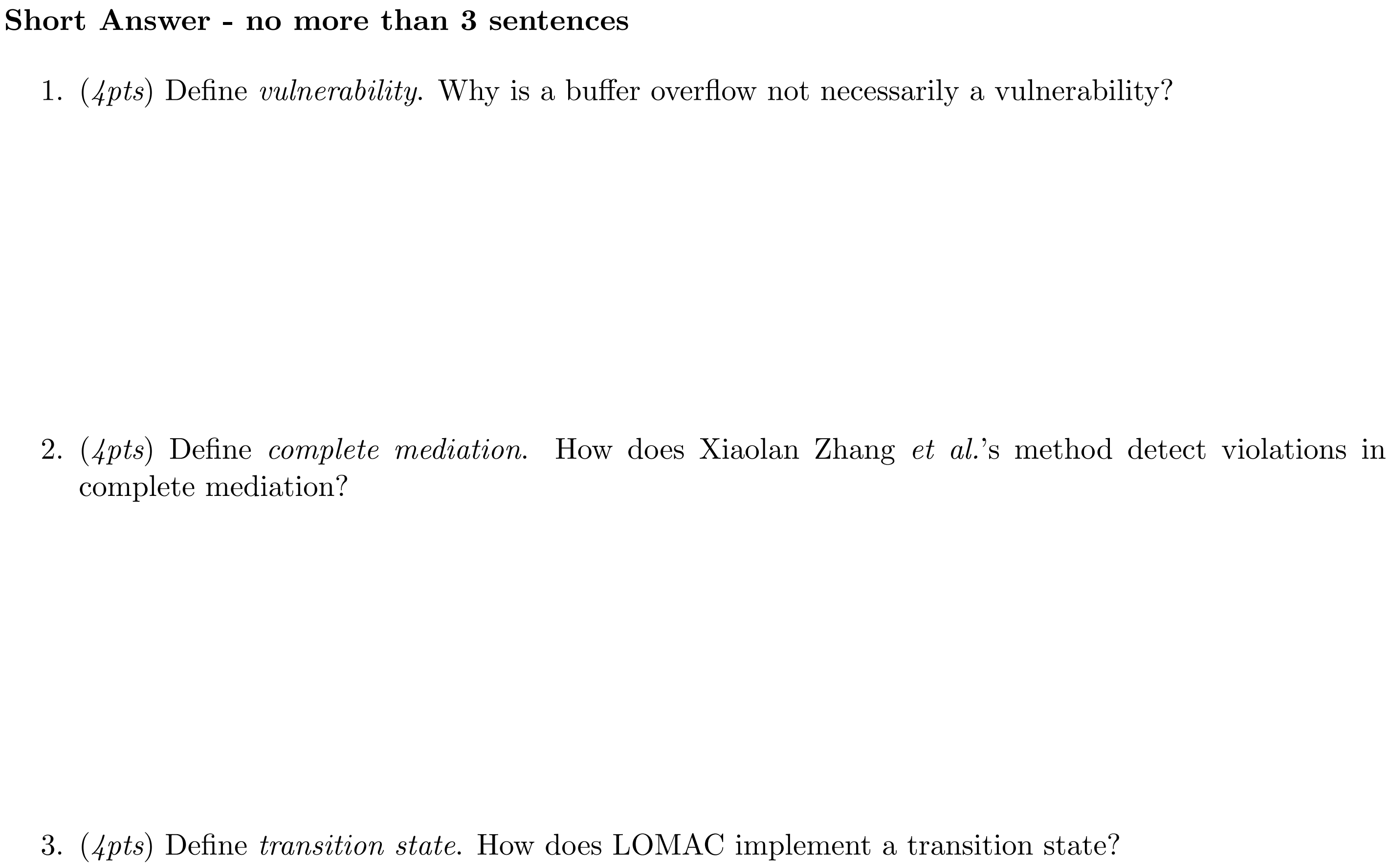
CS260 – Advanced Systems Security

Review

May 19, 2025

Short Answer



Short Answer - no more than 3 sentences

1. (*4pts*) Define *vulnerability*. Why is a buffer overflow not necessarily a vulnerability?

2. (*4pts*) Define *complete mediation*. How does Xiaolan Zhang *et al.*'s method detect violations in complete mediation?

3. (*4pts*) Define *transition state*. How does LOMAC implement a transition state?

Short Answer

Short Answer - no more than 3 sentences

1. (*4pts*) Define *vulnerability*. Why is a buffer overflow not necessarily a vulnerability?

Lookup "vulnerability" definition. Three elements.

Buffer overflow alone only implies one of those elements.

2. (*4pts*) Define *complete mediation*. How does Xiaolan Zhang *et al.*'s method detect violations in complete mediation?

Invoke reference monitor for all security-sensitive operations.

How does CQUAL paper define that in its analysis?

3. (*4pts*) Define *transition state*. How does LOMAC implement a transition state?

Transition state: About relabeling. Why might we need transitions?

LOMAC has transition state rules to change the integrity of a process
Based on the objects it accesses.

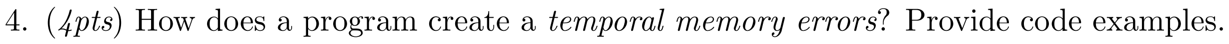
Short Answer

4. (*4pts*) How does a program create a *temporal memory errors*? Provide code examples.

5. (*4pts*) What is the purpose of the *labeling state*? That is, why is it necessary for a mandatory protection system to have a labeling state at all?

6. (*4pts*) Specify what must be verified to satisfy the reference monitor guarantee of *verification*? Explain briefly why.

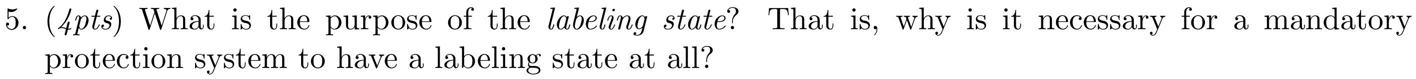
Short Answer



4. (*4pts*) How does a program create a *temporal memory errors*? Provide code examples.

Two cases UBI and UAF.

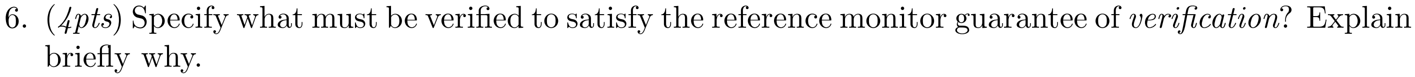
What is a use, initialization, and a free.



5. (*4pts*) What is the purpose of the *labeling state*? That is, why is it necessary for a mandatory protection system to have a labeling state at all?

Labeling state: is about assigning a label in the first place.

Why needed?

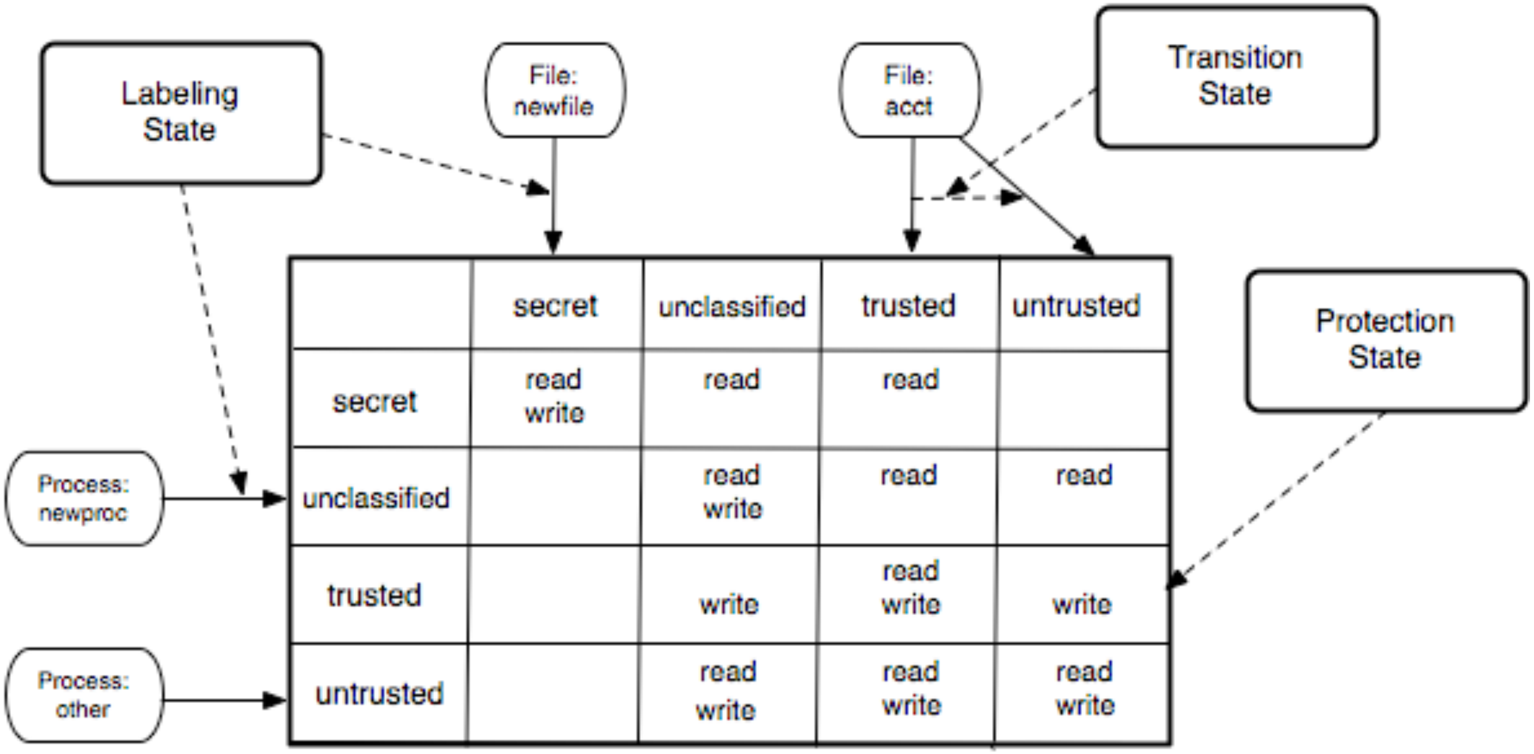


6. (*4pts*) Specify what must be verified to satisfy the reference monitor guarantee of *verification*? Explain briefly why.

Verification aims to validate the correctness of enforcement.

What are the parts of a reference monitor? How to validate?

Mandatory Protection System



	secret	unclassified	trusted	untrusted
secret	read write	read	read	
unclassified		read write	read	read
trusted		write	read write	write
untrusted		read write	read write	read write

Short Answer

7. (*4pts*) How does ASan check for violations of spatial memory safety?

8. (*4pts*) Should *least privilege* be used as a security goal? Why or why not?

Short Answer

7. (*4pts*) How does ASan check for violations of spatial memory safety?

Detect access to shadow memory in red zones.

Should have a good idea about location and identity-based defenses.

8. (*4pts*) Should *least privilege* be used as a security goal? Why or why not?

Why is least privilege good or bad for security.

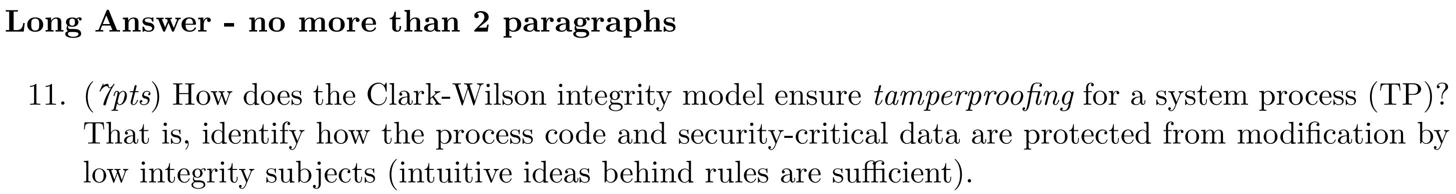
Long Answer

Long Answer - no more than 2 paragraphs

11. (*7pts*) How does the Clark-Wilson integrity model ensure *tamperproofing* for a system process (TP)? That is, identify how the process code and security-critical data are protected from modification by low integrity subjects (intuitive ideas behind rules are sufficient).

12. (*7pts*) What is a *confused deputy* attack? Detail how you would design a server to prevent confused deputy attacks in processing client requests.

Long Answer



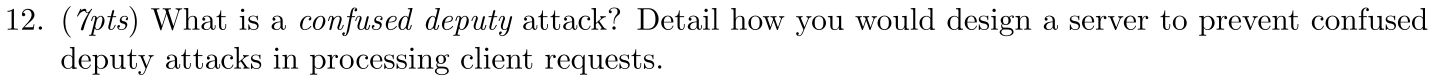
Long Answer - no more than 2 paragraphs

11. (*7pts*) How does the Clark-Wilson integrity model ensure *tamperproofing* for a system process (TP)? That is, identify how the process code and security-critical data are protected from modification by low integrity subjects (intuitive ideas behind rules are sufficient).

Enforcement and certification rules – you should have the idea

How do we know code is high integrity?

How do we know data is high integrity and stays that way when changed?



12. (*7pts*) What is a *confused deputy* attack? Detail how you would design a server to prevent confused deputy attacks in processing client requests.

Two parties involved – requestor and deputy; what is the attack?

Creative ways to answer this. What defense mechanism may help?

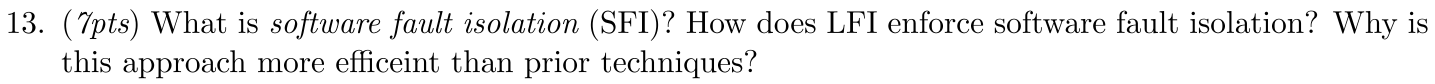
Don’t forget about other file system attacks.

Long Answer

13. (*7pts*) What is *software fault isolation* (SFI)? How does LFI enforce software fault isolation? Why is this approach more efficeint than prior techniques?

14. (*7pts*) Define *control flow integrity* (be as precise as possible). Detail (in code) an example of an attack that could circumvent fine-grained CFI.

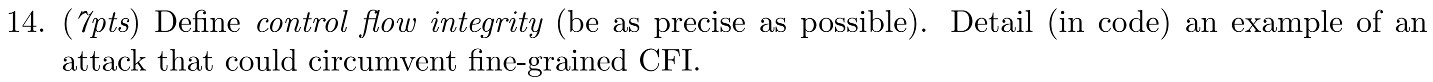
Long Answer



13. (*7pts*) What is *software fault isolation* (SFI)? How does LFI enforce software fault isolation? Why is this approach more efficeint than prior techniques?

SFI limits memory accesses to a prescribed region. E.g., masking.

What is the secret sauce of LFI for performance?



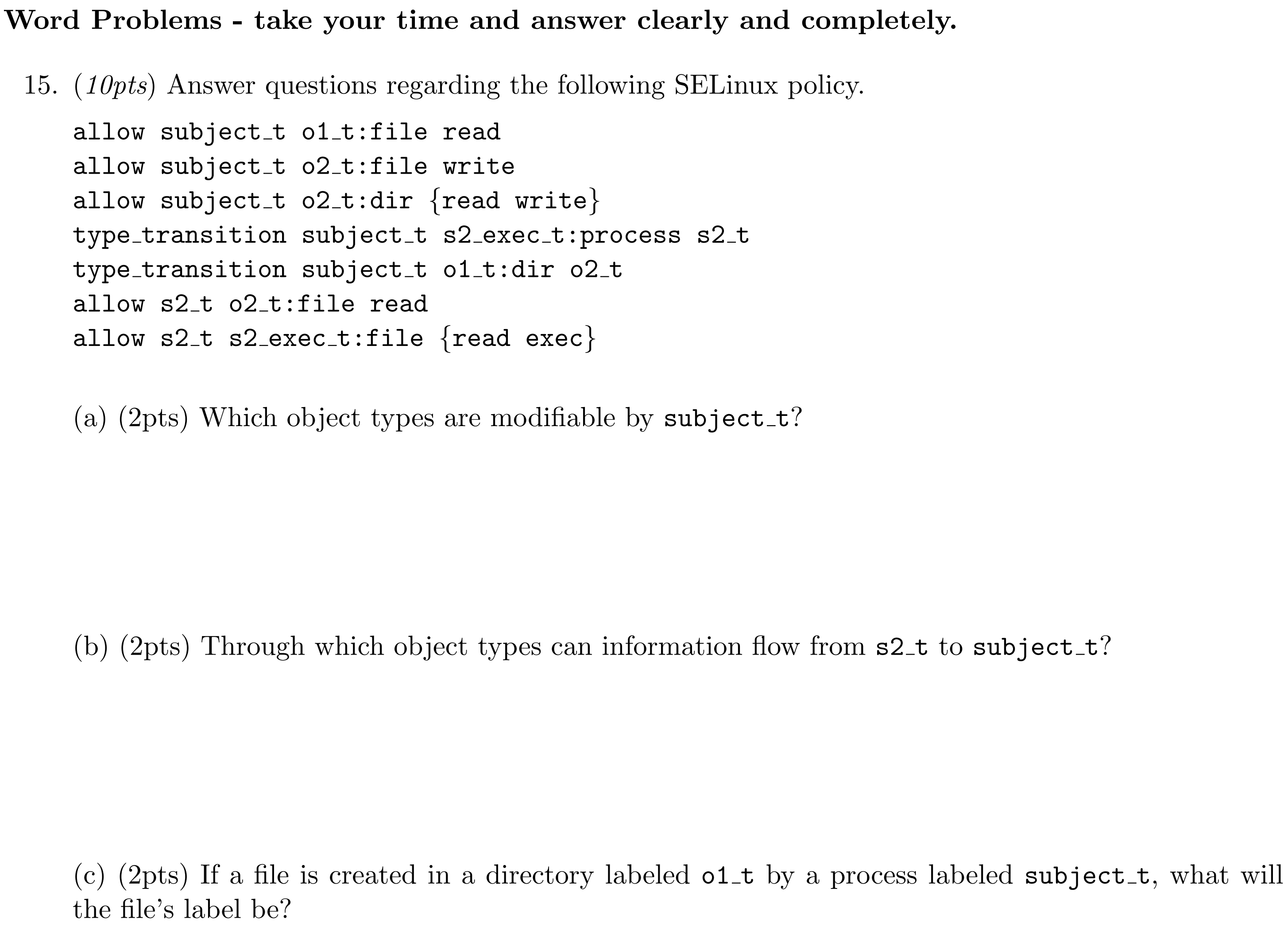
14. (*7pts*) Define *control flow integrity* (be as precise as possible). Detail (in code) an example of an attack that could circumvent fine-grained CFI.

CFI limits the set of targets of indirect control transfers.

What is the fine-grained CFI policy? Shadow stack and limited callees.

Why circumvent? Could be multiple legal targets. How?

Constructions



Word Problems - take your time and answer clearly and completely.

15. (*10pts*) Answer questions regarding the following SELinux policy.

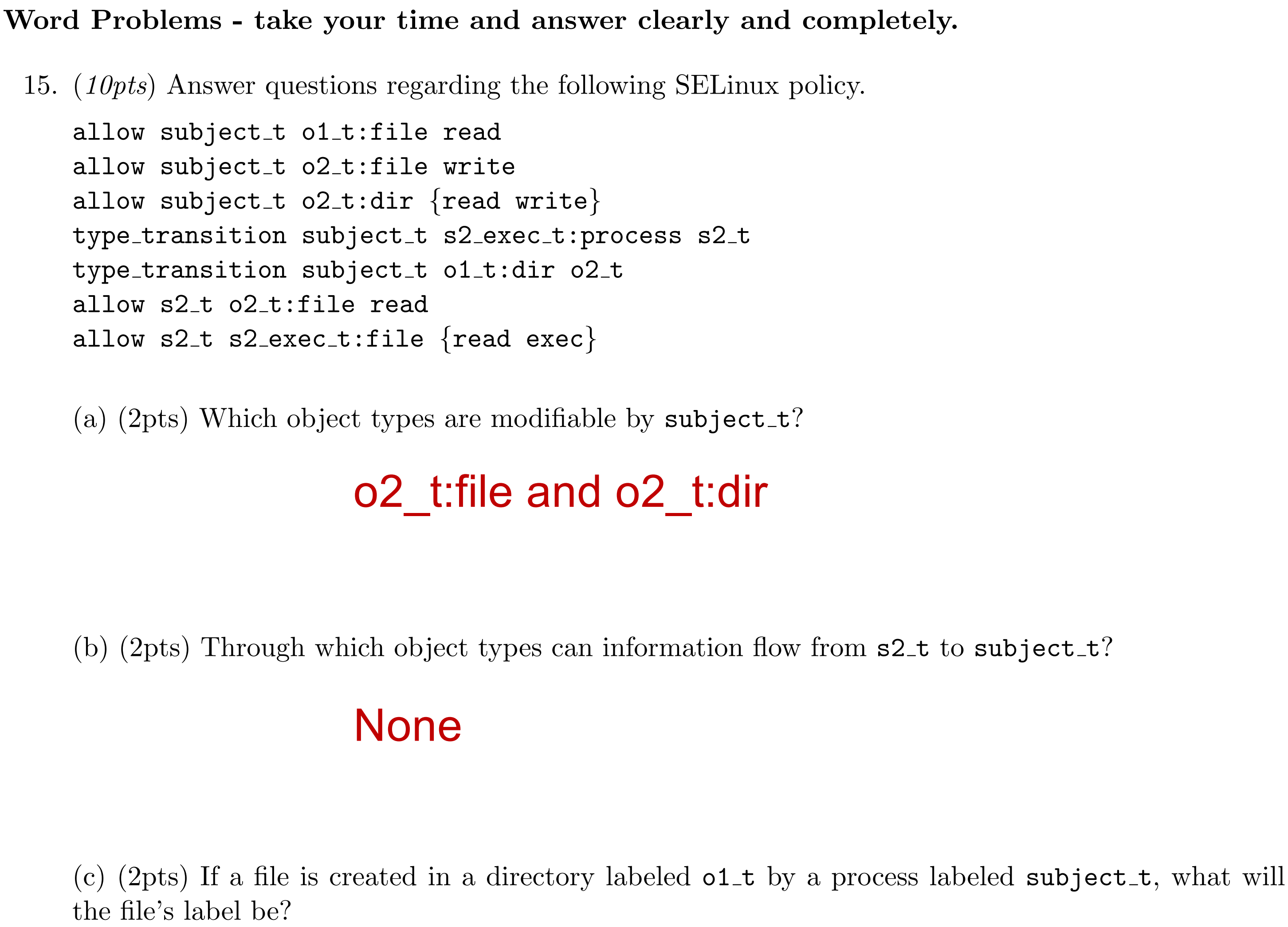
```
allow subject_t o1_t:file read
allow subject_t o2_t:file write
allow subject_t o2_t:dir {read write}
type_transition subject_t s2_exec_t:process s2_t
type_transition subject_t o1_t:dir o2_t
allow s2_t o2_t:file read
allow s2_t s2_exec_t:file {read exec}
```

(a) (2pts) Which object types are modifiable by `subject_t`?

(b) (2pts) Through which object types can information flow from `s2_t` to `subject_t`?

(c) (2pts) If a file is created in a directory labeled `o1_t` by a process labeled `subject_t`, what will the file's label be?

Constructions



Word Problems - take your time and answer clearly and completely.

15. (*10pts*) Answer questions regarding the following SELinux policy.

```
allow subject_t o1_t:file read
allow subject_t o2_t:file write
allow subject_t o2_t:dir {read write}
type_transition subject_t s2_exec_t:process s2_t
type_transition subject_t o1_t:dir o2_t
allow s2_t o2_t:file read
allow s2_t s2_exec_t:file {read exec}
```

(a) (2pts) Which object types are modifiable by `subject_t`?

o2_t:file and o2_t:dir

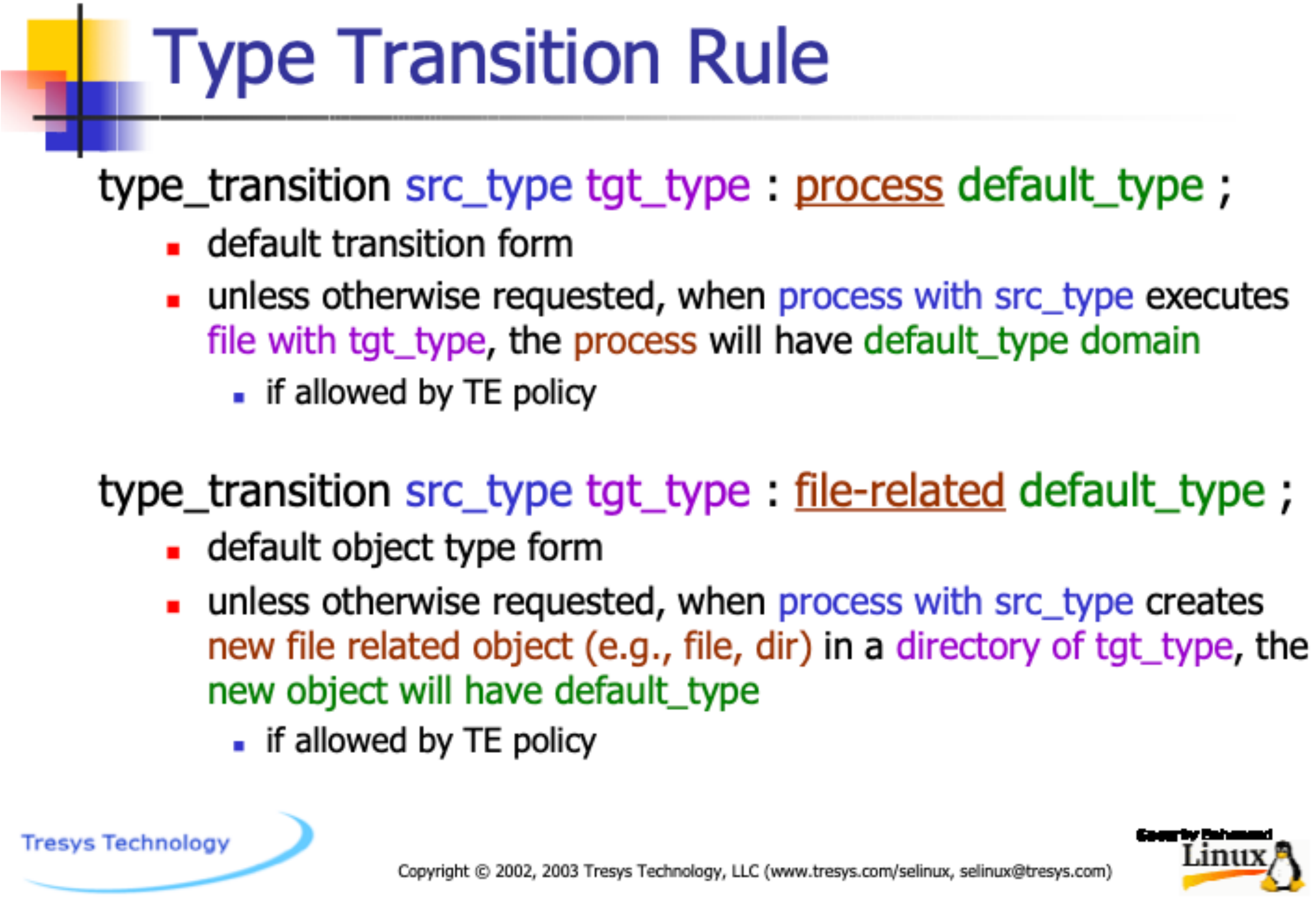
(b) (2pts) Through which object types can information flow from `s2_t` to `subject_t`?

None

(c) (2pts) If a file is created in a directory labeled `o1_t` by a process labeled `subject_t`, what will the file's label be?

o2_t (should be a directory)

Type Transitions



Type Transition Rule

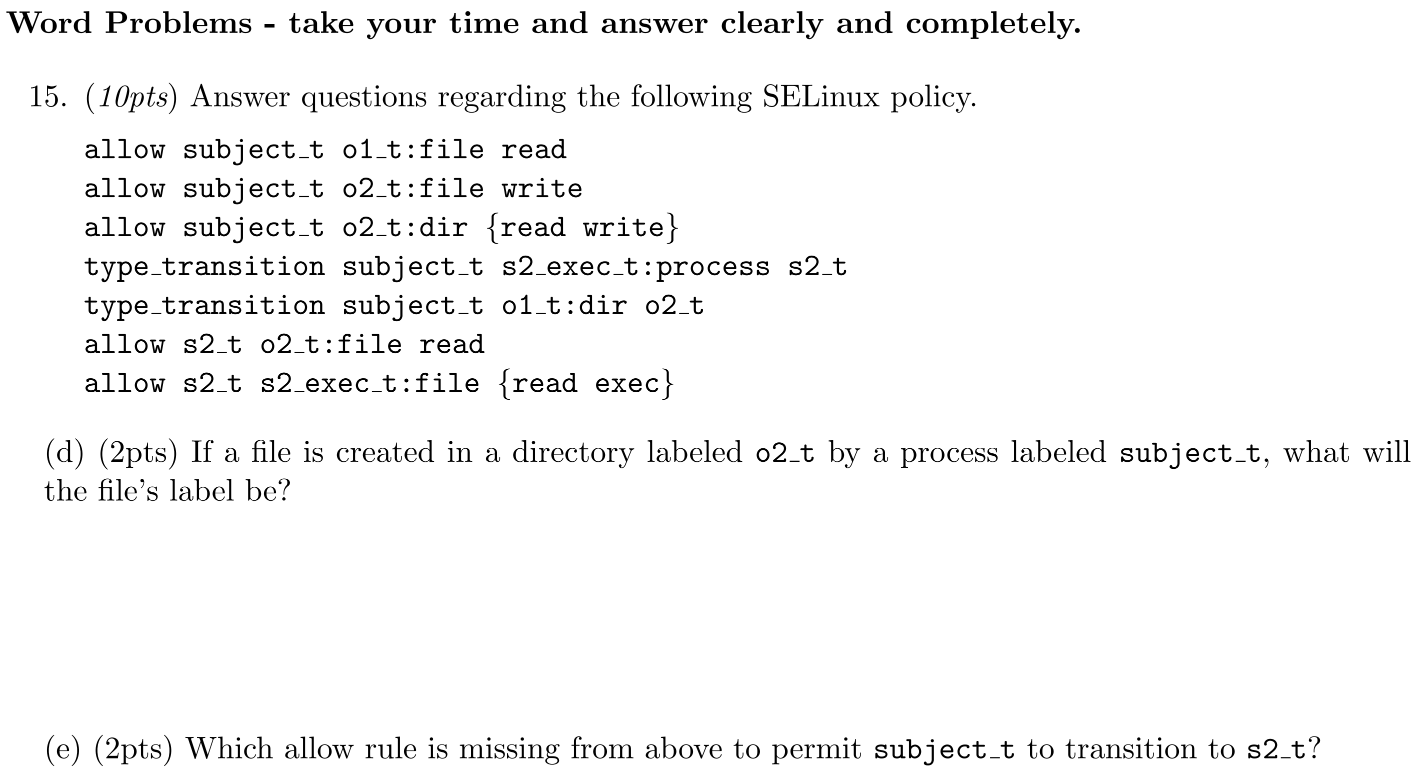
type_transition src_type tgt_type : process default_type ;

- default transition form
- unless otherwise requested, when process with src_type executes file with tgt_type, the process will have default_type domain
  - if allowed by TE policy

type_transition src_type tgt_type : file-related default_type ;

- default object type form
- unless otherwise requested, when process with src_type creates new file related object (e.g., file, dir) in a directory of tgt_type, the new object will have default_type
  - if allowed by TE policy

Constructions



Word Problems - take your time and answer clearly and completely.

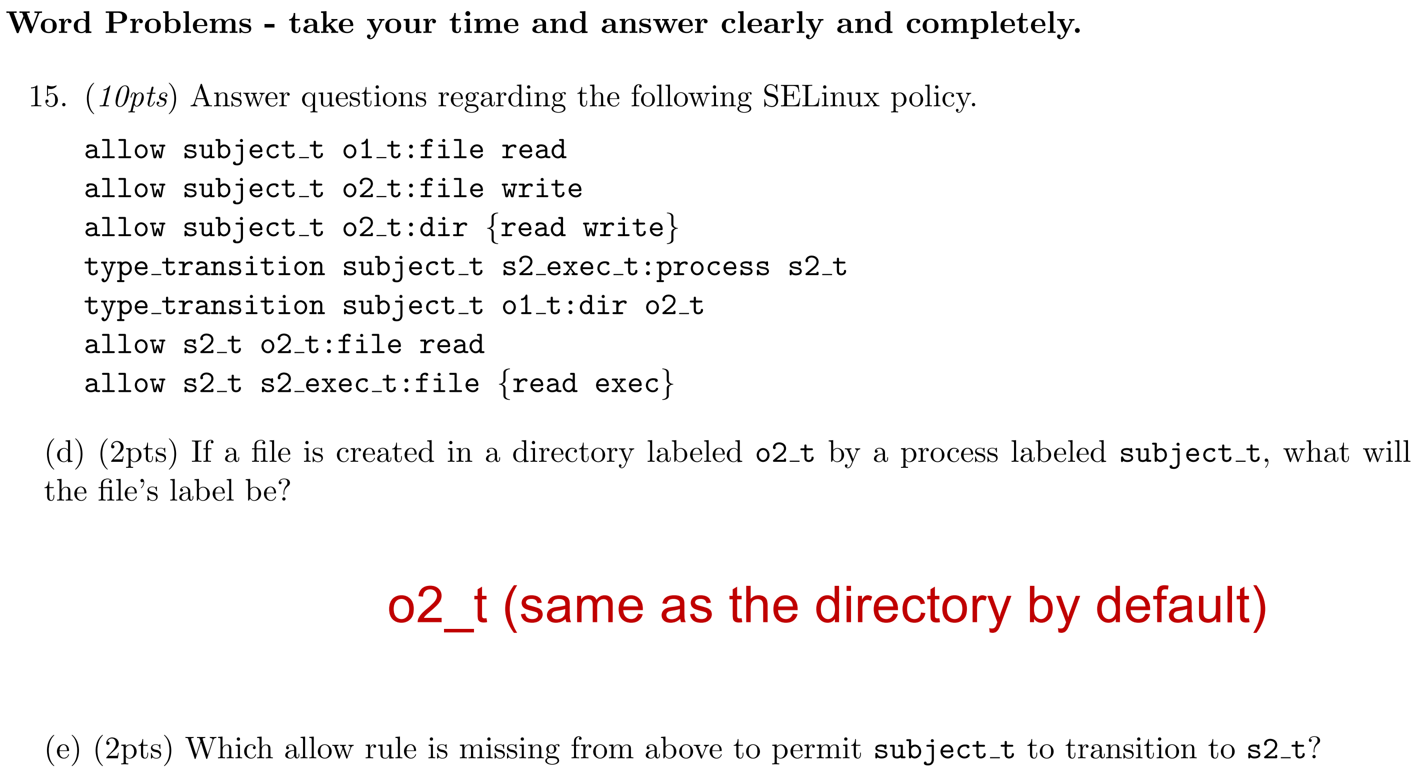
15. (*10pts*) Answer questions regarding the following SELinux policy.

```
allow subject_t o1_t:file read
allow subject_t o2_t:file write
allow subject_t o2_t:dir {read write}
type_transition subject_t s2_exec_t:process s2_t
type_transition subject_t o1_t:dir o2_t
allow s2_t o2_t:file read
allow s2_t s2_exec_t:file {read exec}
```

(d) (2pts) If a file is created in a directory labeled `o2_t` by a process labeled `subject_t`, what will the file's label be?

(e) (2pts) Which allow rule is missing from above to permit `subject_t` to transition to `s2_t`?

Constructions



Word Problems - take your time and answer clearly and completely.

15. (*10pts*) Answer questions regarding the following SELinux policy.

```
allow subject_t o1_t:file read
allow subject_t o2_t:file write
allow subject_t o2_t:dir {read write}
type_transition subject_t s2_exec_t:process s2_t
type_transition subject_t o1_t:dir o2_t
allow s2_t o2_t:file read
allow s2_t s2_exec_t:file {read exec}
```

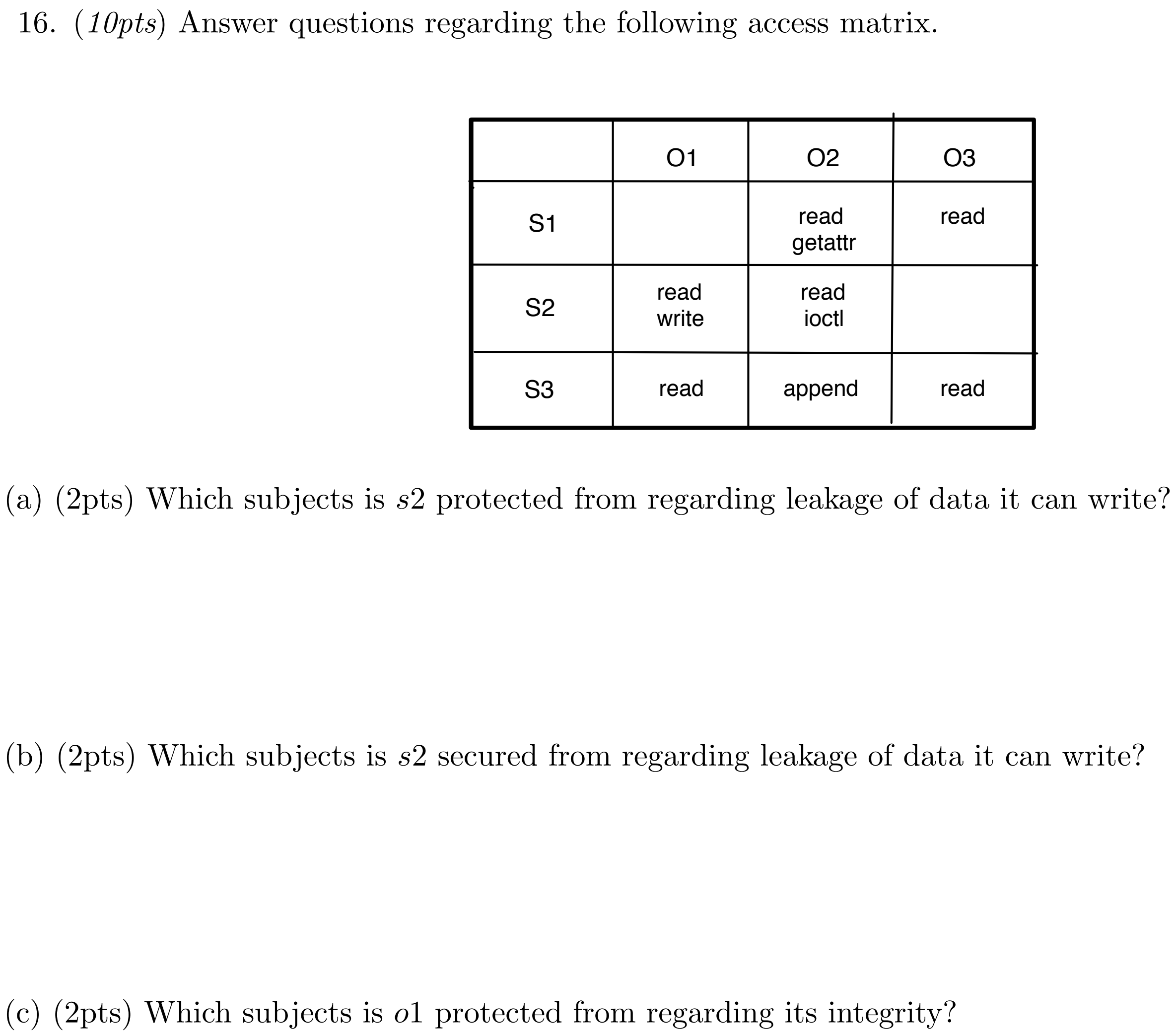
(d) (2pts) If a file is created in a directory labeled `o2_t` by a process labeled `subject_t`, what will the file's label be?

o2_t (same as the directory by default)

(e) (2pts) Which allow rule is missing from above to permit `subject_t` to transition to `s2_t`?

allow subject_t s2_exec_t:process transition

Constructions



16. (*10pts*) Answer questions regarding the following access matrix.

	O1	O2	O3
S1		read getattr	read
S2	read write	read ioctl	
S3	read	append	read

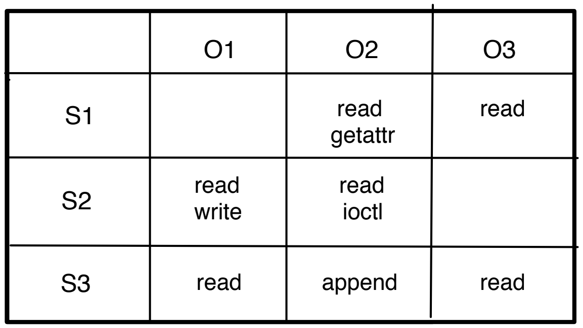
(a) (2pts) Which subjects is $s2$ protected from regarding leakage of data it can write?

(b) (2pts) Which subjects is $s2$ secured from regarding leakage of data it can write?

(c) (2pts) Which subjects is $o1$ protected from regarding its integrity?

Constructions

16. (*10pts*) Answer questions regarding the following access matrix.



	O1	O2	O3
S1		read getattr	read
S2	read write	read ioctl	
S3	read	append	read

(a) (2pts) Which subjects is $s2$ protected from regarding leakage of data it can write?

S1 – protection interprets the matrix literally – s1 can't read o1

(b) (2pts) Which subjects is $s2$ secured from regarding leakage of data it can write?

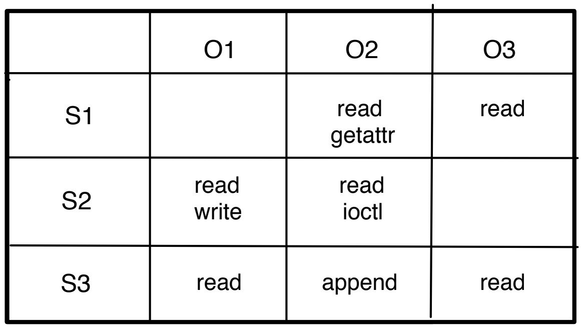
None – security considers the information flows – leak via S3

(c) (2pts) Which subjects is $o1$ protected from regarding its integrity?

S1 and S3 – based on matrix

Constructions

16. (*10pts*) Answer questions regarding the following access matrix.



	O1	O2	O3
S1		read getattr	read
S2	read write	read ioctl	
S3	read	append	read

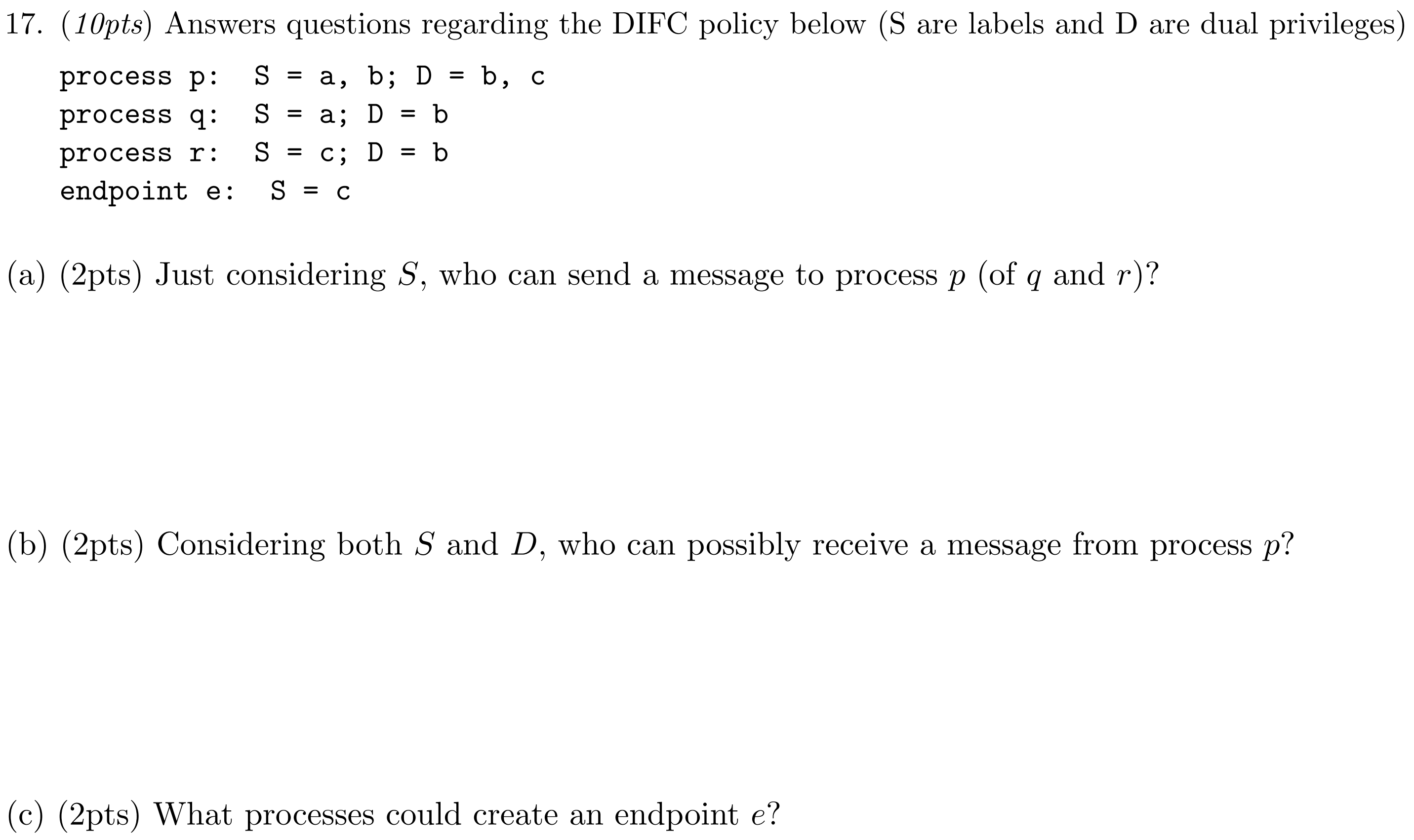
(d) (2pts) Which subjects is $o1$ secured from regarding its integrity?

Just S1 – S3 may provide data to S2 to O1

(e) (2pts) Can $s1$ *write* any object in this access matrix if we want to ensure $s2$'s integrity? Which?

No. Even a write to O3 may impact S2 indirectly via S3

Constructions



17. (*10pts*) Answers questions regarding the DIFC policy below (S are labels and D are dual privileges)

```
process p:  S = a, b; D = b, c
process q:  S = a; D = b
process r:  S = c; D = b
endpoint e:  S = c
```

(a) (2pts) Just considering S, who can send a message to process p (of q and r)?

(b) (2pts) Considering both S and D, who can possibly receive a message from process p?

(c) (2pts) What processes could create an endpoint e?

Constructions

17. (*10pts*) Answers questions regarding the DIFC policy below (S are labels and D are dual privileges)

```
process p:  S = a, b; D = b, c
process q:  S = a; D = b
process r:  S = c; D = b
endpoint e:  S = c
```

(a) (2pts) Just considering S, who can send a message to process p (of q and r)?

Just q can – S label of p is a superset of q’s S label

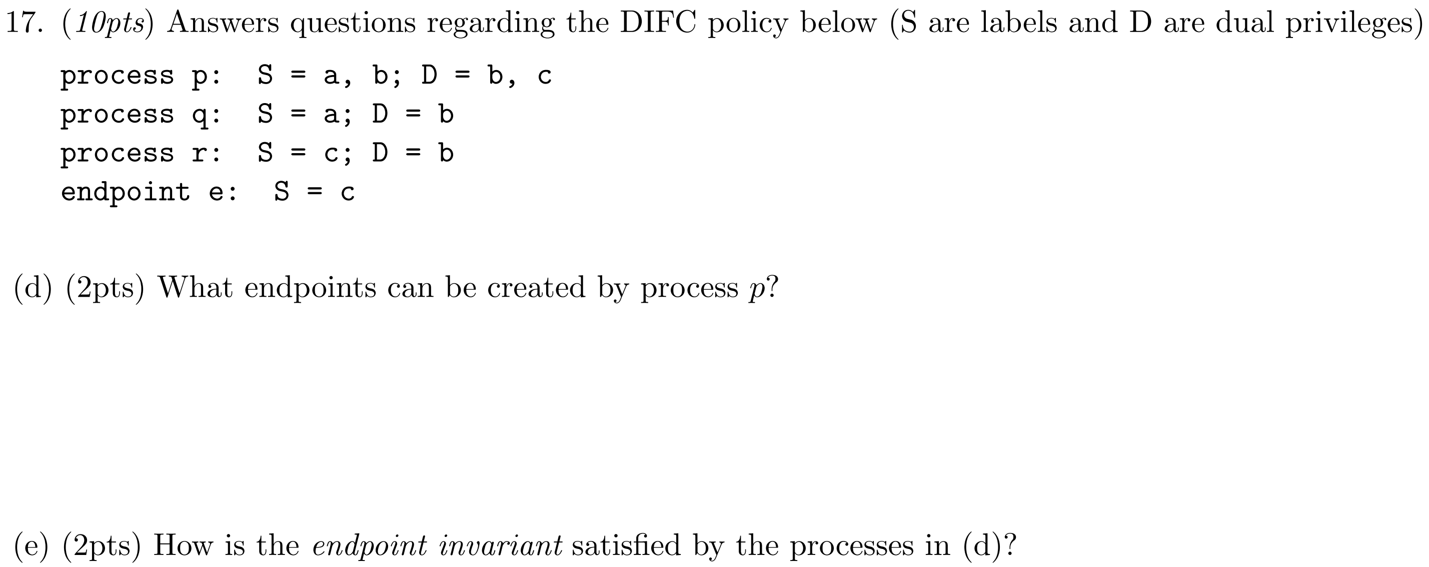
(b) (2pts) Considering both S and D, who can possibly receive a message from process p?

Only q can – p can remove the “b” label from its messages via D

(c) (2pts) What processes could create an endpoint e?

Only r can. r has that S label. While p can add c, it must keep a.

Constructions



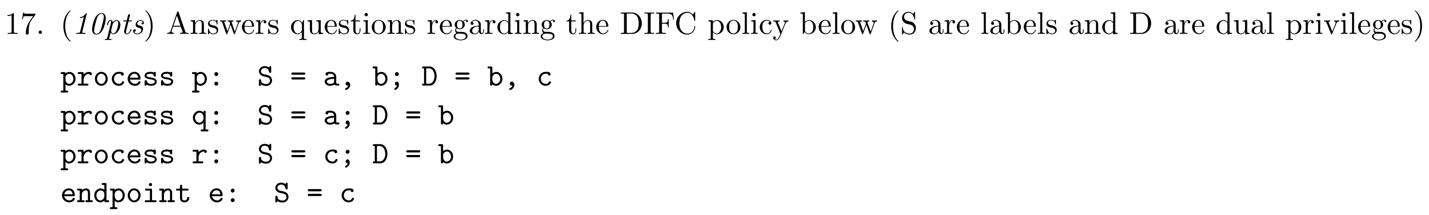
17. (*10pts*) Answers questions regarding the DIFC policy below (S are labels and D are dual privileges)

```
process p:  S = a, b; D = b, c
process q:  S = a; D = b
process r:  S = c; D = b
endpoint e:  S = c
```

(d) (2pts) What endpoints can be created by process p?

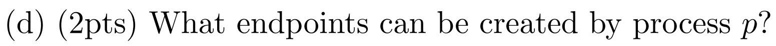
(e) (2pts) How is the *endpoint invariant* satisfied by the processes in (d)?

Constructions



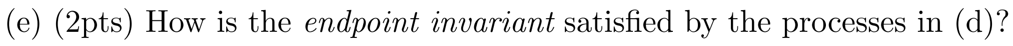
17. (*10pts*) Answers questions regarding the DIFC policy below (S are labels and D are dual privileges)

```
process p:  S = a, b; D = b, c
process q:  S = a; D = b
process r:  S = c; D = b
endpoint e:  S = c
```



(d) (2pts) What endpoints can be created by process *p*?

Process p can create endpoints for a, b, c or all combos except must have an “a” in the label



(e) (2pts) How is the *endpoint invariant* satisfied by the processes in (d)?

Write: For any tag t in Sp and t not in Se

Read: Or any tag t in Se and t not in Sp

Either case: It must be that t in Dp

Constructions

17. (*10pts*) Answers questions regarding the DIFC policy below (S are labels and D are dual privileges)

```
process p:  S = a, b; D = b, c
process q:  S = a; D = b
process r:  S = c; D = b
endpoint e:  S = c
```

(d) (2pts) What endpoints can be created by process *p*?

Process p can create endpoints for a, b, c in all combos, except must have an “a” in the label

(e) (2pts) How is the *endpoint invariant* satisfied by the processes in (d)?

Write: For any tag t in Sp and t not in Se

Read: Or any tag t in Se and t not in Sp

Either case: It must be that t in Dp

Write:
Endpoint must have “a” because “a” is not in D

Read: only “c” can be in Se and not in Sp, but it is in D

Se must include “a”

Questions