

Information Flow Security

DD2460 Software Safety and Security: Part III, lecture 1

Gurvan Le Guernic



DD2460 (III, L1)
February 14th, 2012



Outline

Information Flow Security deals with *Confidentiality* and *Integrity* related security policies.

- 1 Context
- 2 Formalization
- 3 Channels, Flows and Labels
- 4 Wrap-up

Context



Context



More and more information systems (PC, smart-phone, web browser, server, ...) inhabited by applications and data belonging to different "owners".

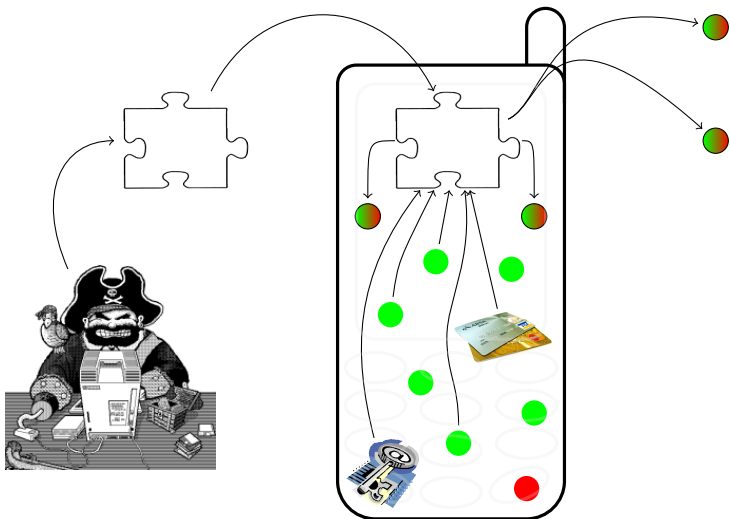
Problem: untrusted applications living in the same space as sensitive data (sometimes even manipulating them).

Same problem for every system manipulating code and/or data with different end-user access rights

- ads in websites
- cross-site scripting
- ...



What's the big deal?





Question!

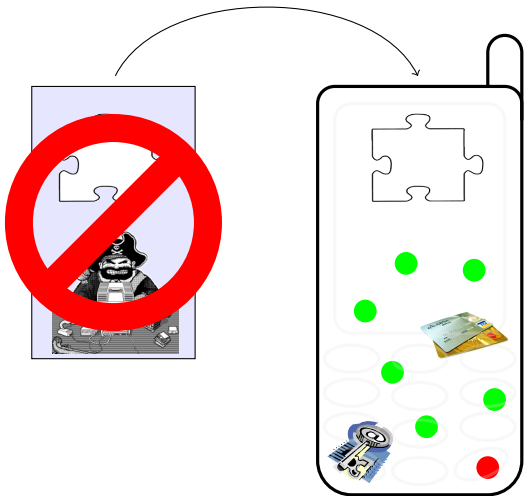
- What security policy do you want for your own connected devices (smartphone, PC, tablet, . . .) in general?

- What policy with regard to your contacts data in particular?



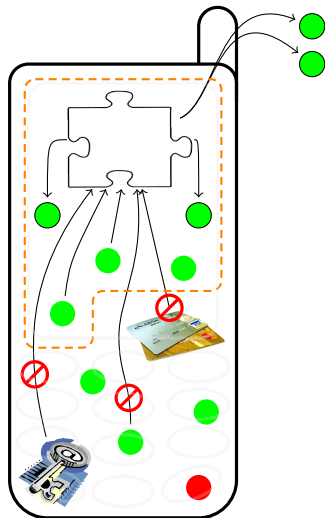
Deployed techniques: Trust

Download and/or execute only from trusted sources





Deployed Techniques: Access Control



- restrict data accessible by a software
- if it can only access public data then it can only output public data
- Allows enforcing *least privilege*

Definition 1 (Least Privilege Principle)

Every entity (process, user, program, ...) should own the least set of privileges (information and resources access right) that is necessary for its legitimate purpose.

Saltzer & Schroeder 1975

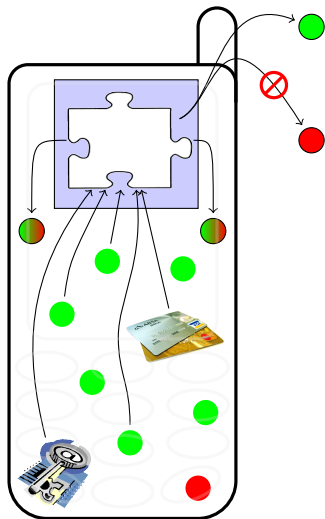


Question!

- Are your own security policies enforceable using those mechanisms (trust & access control)?



Information Flow Security: philosophy



Problem: What about all those **Android™** applications that ask for many privileges?

Philosophy:

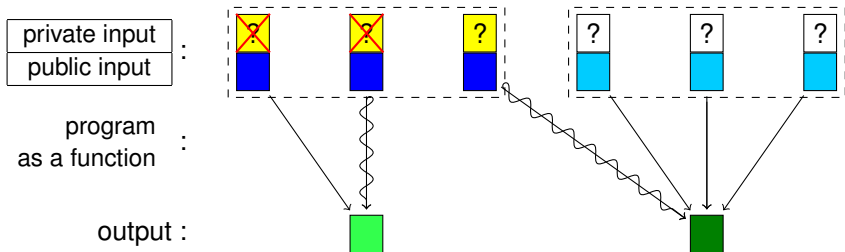
- trust and/or access control are not sufficient
- analyze/track information flows
- prevent data leaks and/or tempering



Secure Information Flows

Definition 2 (Secure Information Flows: confidentiality)

A process is said to contain only secure information flows, wrt confidentiality, if and only if an attacker is unable to deduce information about the secret (hidden) data by looking only at the publicly observable (leaked) outputs of the process.





Software Information Flow Security

Study a program to decide if its executions respects the confidentiality of secret data and the integrity of sensitive data.

For software information flow security, attacker is usually assumed to:

- know the program code
- have a partial view of/control over the execution

Noninterference:

- Cohen (77), Goguen and Meseguer (82)
- Property of a program having only *good* information flows
- Hidden/Hacked inputs do not influence Leaked/Legitimate outputs
 - No (data/control) flow from H to L

Formalization



Strong Dependency

Definition 3 (Strong Dependency)

There exists an information flow from input i to output o in a process P whenever *variety* in i is conveyed to o by the execution of P .

“information is transmitted from a source to a destination only when variety in the source can be conveyed to the destination”

E. S. Cohen, “Information Transmission in Computational Systems”, 1977

For deterministic processes, o is strongly dependent on i if and only if there exist at least two executions of P whose inputs differ only in i and whose outputs differ in o .

⇒ The process P carried over the initial variety in i to the output o .



Noninterference

- Noninterference = absence of *strong dependency* from H (hidden/hacked) inputs to L (leaked/legitimate) outputs.

Definition 4 (Noninterference)

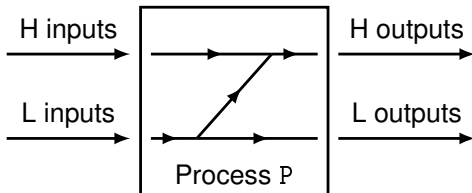
A program is said to be *noninterfering* if and only if any executions, started with the same L (leaked/legitimate) inputs, generate the same L (leaked/legitimate) outputs.



Noninterference: in picture

Allowed Information Flows

A process is said to be *noninterfering* if the values of its L (leaked/legitimate) outputs depend only on the values of its L (leaked/legitimate) inputs.





Noninterference: in Greek letters

Definition 5 (Noninterference)

A program P is *noninterfering* if and only if any two executions, started in execution environments (σ_i) having the same L (leaked/legitimate) values, generate the same L (leaked/legitimate) observations $(\mathcal{O}[\![\sigma_i \vdash P]\!])$.

$$\forall \sigma_1, \sigma_2 : \sigma_1 =_L \sigma_2 \Rightarrow \mathcal{O}[\![\sigma_1 \vdash P]\!] = \mathcal{O}[\![\sigma_2 \vdash P]\!]$$

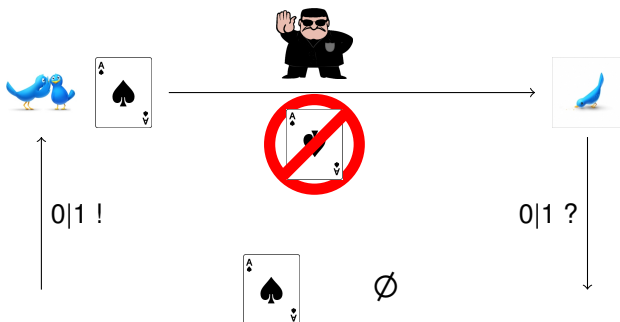
In non-deterministic case, $\mathcal{O}[\![\sigma_1 \vdash P]\!]$ can be:

- set of all possible observations
→ *possibilistic* noninterference
- mapping from all possible observations to probability
→ *probabilistic* noninterference

Channels, Flows and Labels



Card Game





Information \neq Data

Information \neq Data

A piece of data carries more information than its intrinsic value.

“the information carried by a particular message depends on the set it comes from. The information conveyed is not an intrinsic property of the individual message.”

W. R. Ashby, “An Introduction to Cybernetics”, 1956.

“Everything is fine!” does not convey the same information if it comes from:

- someone in vacations,
- someone starting a new job,
- a prisoner in a dictatorship.



Information Channels

Lampson (“A Note on the Confinement Problem”, 1973) defines 3 types of information channels:

Legitimate channels:

- use mechanisms intended for legitimate data transfer
- example: Internet communication for web browser

Storage channels:

- 2 steps transfer using data storage (not transfer) mechanisms
- goal: delaying in time and space the realization of the undesired flow
- Bell-LaPadula’s \star -property (aka “no write-down”/confinement property) aims at reducing such channels usage for access control mechanisms

Covert channels:

- use mechanisms not intended for data manipulation (transfer, computation or storage)
- encode information into visible side effects of legitimate (potentially transfer) mechanisms
- example: file locks, computation time/consumption, program counters, ... (A♠)



Information Flows

Two dimensions: direct/indirect and explicit/implicit

- direct: use legitimate channels intended for data transfer.
- indirect: use channels which are not intended for data transfer.

- explicit: created by the occurrence of a specific event.
- implicit: created by the fact that a specific event does not occur.



Some papers (particularly static techniques) use:

- direct or explicit for direct flows
- indirect or implicit for indirect flows



Information Flows: example

if b then $x := e_1$ else $y := e_2$

	Direct use legitimate channels for data transfer	Indirect use channels not intended for data transfer
Explicit created by the occurrence of an event/action	<ul style="list-style-type: none"> ● $e_1 \rightarrow x$ iff $b = \text{true}$ ● $e_2 \rightarrow y$ iff $b = \text{false}$ 	<ul style="list-style-type: none"> ● $b \rightarrow x$ iff $b = \text{true}$ ● $b \rightarrow y$ iff $b = \text{false}$
Implicit created by the absence of a specific event/action		<ul style="list-style-type: none"> ● $b \rightarrow y$ iff $b = \text{true}$ ● $b \rightarrow x$ iff $b = \text{false}$



Security Labels

Security labels:

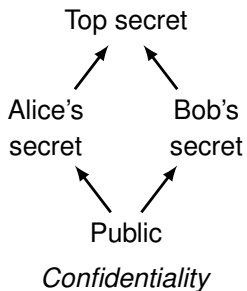
- confidentiality and/or integrity levels
- form a preorder (reflexive and transitive relation)
 - (L, \leq)

Security lattice:

- labels may form a lattice
- preorder with unique least upper-bound (aka lub or join) and greatest lower-bound (aka glb or meet) for any 2 labels
 - $(L, \leq, \sqcup, \sqcap)$
 - lub: $\forall l_1 l_2. l_i \leq (l_1 \sqcup l_2)$ and $\forall l_3. l_i \leq l_3 \leq (l_1 \sqcup l_2) \Rightarrow l_3 = (l_1 \sqcup l_2)$
 - glb: $\forall l_1 l_2. (l_1 \sqcap l_2) \leq l_i$ and $\forall l_3. (l_1 \sqcap l_2) \leq l_3 \leq l_i \Rightarrow l_3 = (l_1 \sqcap l_2)$
 - top: $\forall l. l \leq \top$
 - bottom: $\forall l. \perp \leq l$



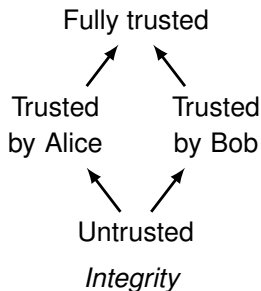
Security Labels: examples (1)



A = Alice's secret

\perp = Public

$A \sqcup \perp = A$



A = Trusted by Alice

B = Trusted by Bob

$A \sqcap B = \text{Untrusted}$



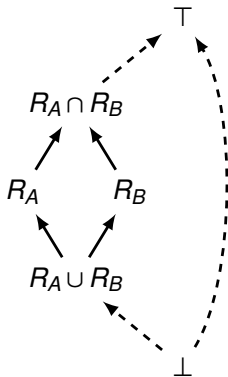
Security Labels: use

Containers (variables, files, . . .) are assigned a label:

- statically or dynamically set
- at write: verify that data's label is less secret (resp. more trusted) than container's label
- at read: consider container's label as a secrecy upper-bound (resp. integrity lower-bound) of data's label



Security Labels: examples (2)



ACL security lattice
(confidentiality)

R_A : set of allowed readers

Untainted



Tainted

Perl Security Lattice
(Integrity)



From Security Lattice to Flow Lattice

Flow lattice:

- describes allowed information flows: $x := y$ iff $l_y \leq l_x$

Flow lattice = confidentiality lattice \times inverse of integrity lattice

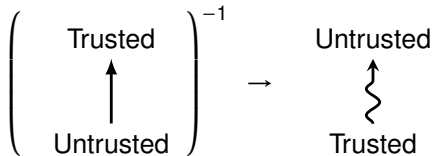
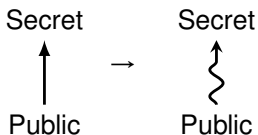


Some always talk about integrity lattice in inverse mode

- Trusted \leq Untrusted

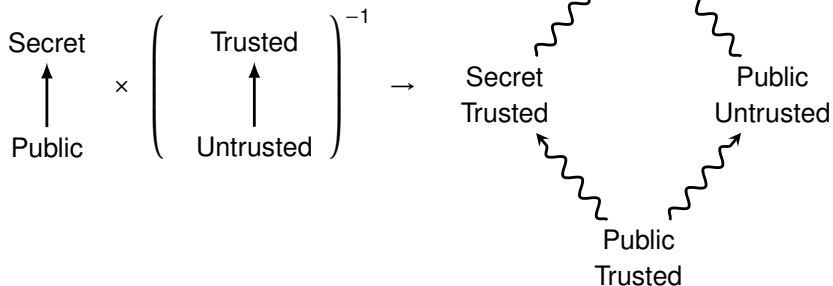


From Security Lattice to Flow Lattice





From Security Lattice to Flow Lattice




Wrap-up



6 Most Important Points

- IF policies = fine grain confidentiality and integrity policies
- Noninterference:

$$\forall \sigma_1, \sigma_2: \sigma_1 =_L \sigma_2 \Rightarrow \mathcal{O}[[\sigma_1 \vdash P]] = \mathcal{O}[[\sigma_2 \vdash P]]$$

- information \neq data
- covert channels = 
- direct/indirect explicit/implicit flows
- Security labels form a flow lattice



IF Workshop

Goal: simulate review of existing IF security techniques

Layout:

- group of 5 to 6 students study 1 paper (6 groups in total)
- presentation by randomly selected student (20-25 mn)
- additions/corrections by rest of the group (5 mn)
- audience questions (5-10 mn)

Two levels collaboration:

- at the group level: deep understanding of the paper
- at the class level: overview of all the papers



Grading

Workshop presentation is not graded per se (report is)

- E:
 - give a decent presentation (or at least additions/corrections session)
 - be able to give an accurate summary of the paper at the course level

- C: (subsumes E)
 - detail specific advantages and limitations of the paper's technique

- A: (subsumes A)
 - compare with the relevant techniques presented in the other papers



Questions?

Questions?