Introduction to Computer Security
Formal Security Models

Pavel Laskov
Wilhelm Schickard Institute for Computer Science
Security instruments learned so far

- Symmetric and asymmetric cryptography
  - confidentiality, integrity, non-repudiation
- Cryptographic hash functions
  - integrity, non-repudiation
- Identity management and authentication
  - authentication
- Access control
  - accountability, integrity
Why do security systems fail?

- Systems are complex. Security of single components does not necessarily imply security of the whole.
- Implementations are buggy. Even minor logical weaknesses can significantly undermine security.
- Users may compromise security by inappropriate use, e.g. weak passwords or falling prey to social engineering attacks.
Why do security systems fail?

- Systems are complex. Security of single components does not necessarily imply security of the whole.
- Implementations are buggy. Even minor logical weaknesses can significantly undermine security.
- Users may compromise security by inappropriate use, e.g. weak passwords or falling prey to social engineering attacks.

Can one prove that a system is secure?
Objectives of formal security modeling

- Facilitate the design of security systems based on imprecise specifications.
- Enable automatic verification of relevant properties.
- Demonstrate to regulatory bodies that a system implementation satisfies the design criteria.
Military security: sensitivity levels

- USA: top secret, secret, classified, unclassified.
- Germany:
  - **STRENG GEHEIM (str. geh)**: die Kenntnisnahme durch Unbefugte kann den Bestand oder lebenswichtige Interessen der Bundesrepublik Deutschland oder eines ihrer Länder gefährden.
  - **GEHEIM (geh.)**: die Kenntnisnahme durch Unbefugte kann die Sicherheit der Bundesrepublik Deutschland oder eines ihrer Länder gefährden oder ihren Interessen schweren Schaden zufügen.
  - **VS-VERTRAULICH (VS-Vertr.)**: die Kenntnisnahme durch Unbefugte kann für die Interessen der Bundesrepublik Deutschland oder eines ihrer Länder schädlich sein.
  - **VS-NUR FÜR DEN DIENSTGEBRAUCH (VS-NfD)**: die Kenntnisnahme durch Unbefugte kann für die Interessen der Bundesrepublik Deutschland oder eines ihrer Länder nachteilig sein.
Security clearance

- Quantification of trust in personnel with respect to handling of different levels of classified information.
- Corresponds to certain screening procedures and investigations.
- Connected to certain legal responsibilities and punitive actions.
Compartmentalization

- Fine-grain classification according to job-related "need-to-know"
- Horizontal division of security clearance levels into specific compartments with a narrow scope.
Implications of automation for security

- Less trust into intermediate tools: can we e.g. ensure that a text editor in which a document was created was not trojanized?
- Tampering with a digital document is much easier than tampering with a physically stored document.
- Difficulty of authentication: less reliance on physical authentication.
- Covert information channels.
Key security models

- Finite state machines
  - Bell-La Padula model: access control only
  - Biba model: additional integrity verification

- Information flow models
  - Chinese wall model: identification of conflicts of interest
  - Identification of covert channels

- Access matrix models
  - Policy manager: separation of access control into a separate process
  - Take-grant model: graph-theoretical interpretation of an access matrix
Bell-La Padula (BLP) model

- States describe system elements and access rights.
- Security policies are defined in terms of security levels and transitions between them.
- Objects $o \in O$
- Subjects $s \in S$
- Access rights $a(s, o) \in A$
  - execute (neither observe nor alter)
  - read (observe but not alter)
  - append (alter but not observe)
  - write (both observe and alter)
- Ownership attribute $x \in \{0, 1\}$.
- A tuple $b = (s, o, a, x)$ characterizes a current access relationship between $s$ and $o$. 

**Access control matrix**
Each element is assigned an integer-valued classification \( (C) \) and a set-valued category \( (K) \) attribute.

A security level is a pair \( (C, K) \).

A s.l. \( (C_1, K_1) \) dominates \( \propto \) a s.l. \( (C_2, K_2) \) if and only if

\[ C_1 \geq C_2 \text{ and } K_1 \supseteq K_2 \]

Example:
BLP security level functions

BLP defines the following three security level functions:

- $f_S(s_i)$: a (maximum) security level of a subject $s_i$,
- $f_O(o_j)$: a security level of an object $o_j$,
- $f_C(s_i)$: a current security level of a subject $s_i$ (if the latter operates at a lower security level).

A state $\nu$ of a BLP is a tuple $(B, M, F_S, F_O, F_C)$ that characterizes all current access relationships $B$, a matrix $M$ of all possible access relationships, and all security level functions.
For any \((s, o, a)\) such that \(a = \text{“observe”}\),

\[ f_S(s) \propto f_O(o). \]

This relationship is known as “no-read-up”: a subject cannot observe (read or write) an object for which is has insufficient clearance.
“Star” security property of BLP

For any pair \((s, o_1, a_1)\) and \((s, o_2, a_2)\) such that \(a_1 = \text{“alter”}\) and \(a_2 = \text{“observe”}\),

\[
f_O(o_1) \propto f_O(o_2).
\]

This relationship is known as “no-write-down”: a subject cannot use the knowledge from observing more restricted objects while altering less restricted objects.
For a tuple \((s_i, o_j, a, x)\), if \(s_i\) is an owner of \(o_j\), i.e. \(x = 1\), he can pass \(a\) to \(s_k\), provided that \(a \in M_{kj}\).

This relationship is known as “discretionary” security, as it allows access relationships to be passed between objects provided this is allowed by an access control matrix.
Consider the following service hierarchy:

General Z (Top Secret, \{crypto\})

Colonel X (Secret, \{nuclear, crypto\})

Major Y (Secret, \{crypto\})

General Z is substituted during his vacation by a colonel X. Major Y must complete a report \( R \) according to an instruction set \( I \). Permissions on these documents are set as follows:

\[
\]

\[
\]
Security level functions are set as follows:

\[ f_S(X) = (S, \{N, C\}) \]
\[ f_S(Y) = (S, \{C\}) \]
\[ f_S(Z) = (TS, \{C\}) \]
\[ f_O(I) = (S, \{C\}) \]
\[ f_O(R) = (S, \{C\}) \]

Q: Are the security properties satisfied?
BLP model example: SSP

We have to verify that:

\[(s, o, a) : a = 'R' \Rightarrow f_S(s) \propto f_O(o)\]

For example:

\[(X, I, 'RW') : f_S(X) = (S, \{N, C\})\]

\[f_O(I) = (S, \{C\}) \quad \text{OK}\]

\[(Y, I, 'R') : f_S(Y) = (S, \{C\})\]

\[f_O(I) = (S, \{C\}) \quad \text{OK}\]
BLP model example: *SP

We have to verify that:

\[((s, o_1, a_1), (s, o_2, a_2)) \text{ s.t. } a_1 = 'W', a_2 = 'R' \Rightarrow f_O(o_1) \propto f_O(o_2)\]

For example:

\[(Y, R, 'RW'), (Y, I, 'R') : f_O(R) = (S, \{C\})\]
\[f_O(I) = (S, \{C\}) \quad \text{OK}\]
Consider an extended service hierarchy below:

Q: Can \( X \) reuse an instruction set \( I \) for \( V \)?
add $V: 'R'$ to $I$’s ACL...

$$(V, I, 'R'): f_S(V) = (S, \{N\})$$
$$f_O(I) = (S, \{C\}) \quad !!!$$
add $V : 'R'$ to $I$’s ACL...

$$(V, I, 'R') : f_S(V) = (S, \{N\})$$

$$f_O(I) = (S, \{C\}) \quad !!!$$

change $f_O(I)$ to $(S, \{N, C\})$...

$$(Y, R, 'RW'), (Y, I, 'R') : f_O(R) = S, \{C\})$$

$$f_O(I) = (S, \{N, C\}) \quad !!!$$
BLP model example: correct action
BLP model example: correct action

- clone $I$ into $I'$
- set $f_O(I') = (S, \{N\})$
- set $ACL(I') = (X: 'RW', V: 'R')$
Transition functions in BLP

- **Altering current access**
  - get access (add \((s, o, a, x)\) to \(B\))
  - release access (remove \((s, o, a, x)\) from \(B\))

- **Altering level functions**
  - change object level \(f_O(o)\)
  - change current subject level \(f_C(o)\)

- **Altering access permissions:**
  - give access permission (add \(a\) to \(M\))
  - rescind access permissions (remove \(a\) from \(M\))

- **Altering the data hierarchy**
  - create an object
  - delete an object
The basic security theorem of BLP

- A state \( b, M, f \) is called **secure** if it satisfies all three security properties of BLP.

- A transition from \( v_1 = (b_1, M_1, f_1) \) to \( v_2 = (b_2, M_2, f_2) \) is secure if both \( v_1 \) and \( v_2 \) are secure.

- Necessary and sufficient conditions for secure transitions vary for different security properties. For example, a transition \( (b_1, M_1, f_1) \rightarrow (b_2, M_2, f_2) \) satisfies the simple security property if and only if:
  - each \((s, o, a) \in b_2 \setminus b_1\) satisfies \( f_2 \), and
  - \((s, o, a)\) does not satisfy \( f_2 \) implies that \((s, o, a) \notin b_2\).

- **Basic Security Theorem**: given a secure initial state, every secure transition brings a system into a secure state on any input.
Formal security models allow one to formally verify security properties of computer systems.

The Bell-La Padula (BLP) model uses the finite state machine to verify access control properties inspired by military security.

BLP was realized in a real operating systems (MULTICS) which, however, suffered from insufficient usability and high maintenance workload.