Reasoning about Security Properties: Safety, Liveness and beyond

Felix Gärtner

LPD, EPFL

fgaertner@lpdmail.epfl.ch

Aim of this talk

- An important result by Alpern and Schneider [1985] is often quoted as:

  Every property is the intersection of a safety and a liveness property.

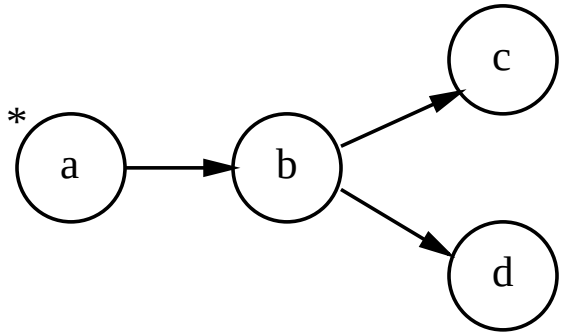
- This is not true (neither the statement nor the quote).

- Please take home from this talk:
  - There are properties which are neither safety nor liveness properties!
  - The correct quote is:

    Every property which is formalizable as a set of traces can be written as the intersection of a safety and a liveness property.

- Outline: Traces? Safety/liveness? Examples of not-trace-set properties and relation to security.

Systems and Traces



- Program Σ: State machine (C, I, T) with set of states C, set of initial states $I \subseteq C$ and transition relation $T \subseteq C \times C$.
- Program generates traces, e.g.. abc, abd.
- Semantics of program $sem(\Sigma)$: Set of all traces of Σ (interleaving semantics).

Properties

- Property P: Set of traces.
- Examples:
  - Property "never d" is modeled as the set

$$\{a, aa, aaa, b, ba, baa, cab, \ldots\}$$

  - Property "whenever a, then in the next step b (if there is a next step)":

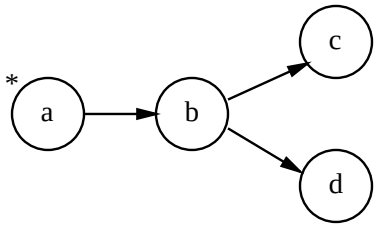
$$\{ab, a, bb, ccc, cccab, dbababb, \ldots\}$$

- Linear temporal logic provides a "syntax" for properties:
  - Example: $\Diamond a = \{a, ba, aaa, bbbbba, dcdca, \ldots\}$
  - another example: $\Box\neg d =$ "always not d" $=$ "never d"

Proving Properties

- Given program Σ and a property P.
- Σ satisfies P iff all traces of Σ are in P.
- Example: Does the following program satisfy

$$P = \{abcd, aabcd, abc, abbc, abd, abad, \ldots\}$$



- Formally: Correctness is trace subsetting ($sem(\Sigma) \subseteq P$).
- Proofs depend on type of property. . .

Safety Properties

- Safety property S ("always . . . "): Examples
  - mutual exclusion: "never two processes in the critical section at the same time"
  - partial correctness: "if the system has terminated, the postcondition holds"

- Formally: Violation occurs through an irremediably bad thing

$$\sigma \notin S \Rightarrow \exists i : \forall \beta : \sigma|_i \cdot \beta \notin S$$

  Notation: σ, β are traces, $\sigma|_i$ means prefix of σ of length i, '$\cdot$' is concatenation.

- Proof through an invariance argument.

Liveness properties

- Liveness property L ("eventually . . . "): Examples
  - Termination: "eventually a termination state is reached"
  - Availability: "every request is eventually served"
- Formally: Something "good" remains possible

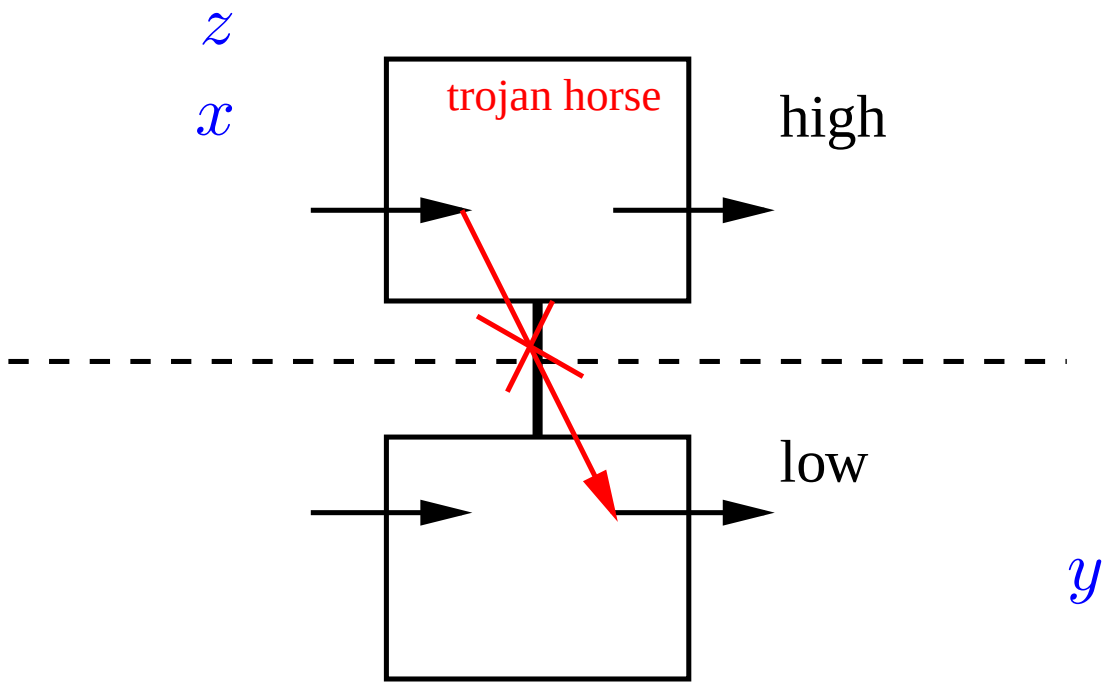
$$\forall i : \exists \beta : \sigma|_i \cdot \beta \in L$$

- Proof using a well-foundedness argument (e.g. termination function).
- Alpern and Schneider [1985]: $\forall P : \exists S, L : P = S \cap L$ (proof using topological arguments: safety properties correspond to closed sets and liveness properties correspond to dense sets)

Formalizing security properties

- Some security properties can be formalized as safety or liveness:
- Safety:
  - access control [Schneider 1998]: "bad" thing happens if intruder enters restricted area.
  - aspects of confidentiality in key establishment [Gray, III. and McLean 1995]: attacker cannot get hold of the established key by any logical or algebraic method (see also BAN logic by Burrows, Abadi, and Needham [1990])
- Liveness:
  - availability: an attacker cannot delay a response infinitely long.
- Problematic: information flow properties over covert channels. . .

Information flow properties

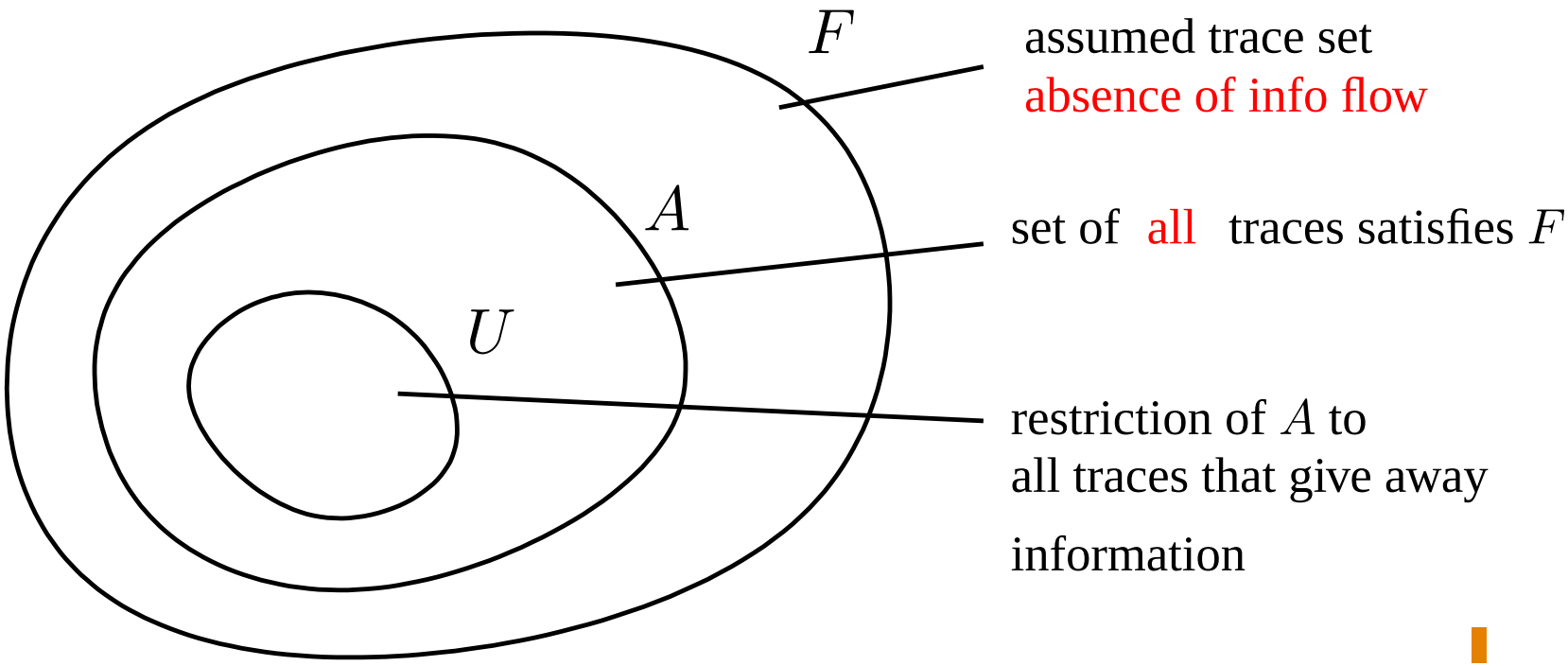


possible traces: $x\ y$ $z\ y$

- Interface definition of security.

Information flow is not a trace set

- Context: properties are trace sets and satisfaction is trace subsetting.



Information flow is not a trace set (cont.)

- Proof [McLean 1994]:
  - Assume F is a set of traces specifying absence of information flow from high to low.
  - Set A of all possible traces is "secure".
  - A implements F, i.e. $A \subseteq F$.
  - Construct set U with all traces from A where x is "switched through" to be y (information flows).
  - U refines A, i.e. $U \subseteq F$.
  - $U \subseteq F$, a contradiction!

Properties of properties

- Properties are of the form: if trace x, y is possible, then trace z, y must be possible too.
- Closure condition on a trace set:

$$\sigma \in P \Rightarrow f(\sigma) \subseteq P$$

- Not a trace set but a property of a trace set, a set of trace sets.
- Term in the literature: noninterference, possibilistic security properties.
- Term relating to Alpern/Schneider framework: higher level properties [Rushby 1994]

Structures in higher level properties

- Non-interference usually defined in the context of event systems. Attacker only sees low level events. It should not be possible to deduce "confidential" information about high level events.

- Different flavors of non-interference depending on interpretation of "confidential" (non-inference [O'Halloran 1990], perfect security property [Zakinthinos and Lee 1997], etc.).

- Structure of non-interference properties discovered by Mantel [2000a]: Inserting and deleting events in constructing the closure.

- Assume events h (high) and l (low).
  - $sem(\Sigma) = \{hl\}$: Attacker derives that h has happened.
    If this is confidential, postulate that $l \in sem(\Sigma)$.
  - $sem(\Sigma) = \{l\}$: Attacker derives that h has *not* happened.
    If this is confidential, postulate that $hl \in sem(\Sigma)$.

Intricacies of higher level properties

- Be careful with inserting events:

  Rule "IE": For every high level event $\alpha h \beta$ add $\alpha h' \beta$ for all high level events h'.

  - Assume Σ is specified as "if h_1 occurs then the next event must be h_2 (if there is a next event) at the high level".
  - Example: $sem(\Sigma) = \{h_1, h_1 h_2\}$
  - Closure adds $h_1 h_3, h_1 h_4, \ldots$, i.e., IE rules out any meaningful high level behavior.

- Verifying higher level properties is different from proving safety or liveness [Mantel 2000b; Rushby 1992].

Research questions

- Specification of a secure system consists of a safety property, a liveness property and a non-interference property.
  - Non-interference can be incompatible with safety.
- Challenge: make higher level properties as well understood as safety and liveness properties.
  - Is there a higher level analogon to the Alpern/Schneider decomposition result?
- What are the relations between higher level properties and cryptographic (complexity-theoretic) definitions of security?
- What is the difference between faults and attacks?
  - Faults are random events, attacks not.
  - Faults are known in advance, attacks not (they usually exploit unformalized parts of the system).

Remember

There is more than safety and liveness!

- Recommendation (one of my top ten favorite papers):

  "Critical system properties: Survey and Taxonomy" by John Rushby (Reliability Engineering and System Safety, 1994)

Acknowledgments

- Slides produced using pdfLaTeX and Klaus Guntermann's PPower4.

References

Alpern, B. and Schneider, F. B. 1985. Defining liveness. *Information Processing Letters 21*, 181–185.

Burrows, M., Abadi, M., and Needham, R. 1990. A logic of authentication. *ACM Transactions on Computer Systems 8*, 1 (Feb.), 18–36.

Gray, III., J. W. and McLean, J. 1995. Using temporal logic to specify and verify cryptographic protocols. In *Proceedings of the Eighth Computer Security Foundations Workshop (CSFW '95)* (June 1995), pp. 108–117. IEEE Computer Society Press.

Mantel, H. 2000a. Possibilistic definitions of security - An assembly kit. In *Proceedings of the 13th IEEE Computer Security Foundations Workshop (CSFW 2000)* (Cambridge, England, July 2000). IEEE Computer Society Press.

Mantel, H. 2000b. Unwinding Possibilistic Security Properties. In F. Cuppens, Y. Deswarte, D. Gollmann, and M. Waidner Eds., *European Symposium on Research in Computer Security (ESORICS)*, Number 1895 in Lecture Notes in Computer Science (Toulouse, France, Oct. 2000), pp. 238–254. Springer-Verlag.

McLean, J. 1994. A general theory of composition for trace sets closed under selective interleaving functions. In *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy* (Oakland, CA, 1994), pp. 79–93.

O'Halloran, C. 1990. A calculus of information flow. In *Proceedings of the European Symposium on Research in Computer Security, ESORICS 90* (Toulouse, France, Oct. 1990), pp. 147–159.

Rushby, J. 1992. Noninterference, transitivity, and channel-control security policies. Technical Report CSL-92-02 (Dec.), Computer Science Laboratory, SRI International, Menlo Park, CA.

Rushby, J. 1994. Critical system properties: Survey and taxonomy. *Reliability Engineering and System Safety 43*, 2, 189–219.

Schneider, F. B. 1998. Enforceable security policies. Technical Report TR98-1664 (Jan.), Cornell University, Department of Computer Science, Ithaca, New York.

Zakinthinos, A. and Lee, E. S. 1997. A general theory of security properties. In *Proceedings of the 18th IEEE Computer Society Symposium on Research in Security and Privacy* (1997).

Abstract

Taking the famous Alpern/Schneider result literally ("every property is the intersection of a safety property and a liveness property"), many people grow up academically in the belief that the world consists only of safety and liveness properties. This is true for many areas of computer science and especially in fault-tolerance the notions of safety and liveness have proven to be sufficient for most tasks. When it gets to reasoning about security properties, this is not true anymore. In this talk, I will survey and formalize a couple of security properties and show that the most interesting ones cannot be represented as safety or liveness properties. Finally I will motivate and sketch some of my research questions evolving from these findings.