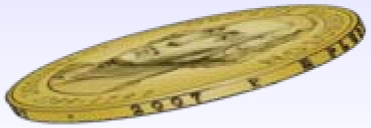
The other side of the coin:



Applications of Typing in Computer Security

Iliano Cervesato

Carnegie Mellon University - Qatar

iliano@cmu.edu

Types in Protocol Specifications

- Identify the meaning of message entities
  - Descriptive / Prescriptive
- Abstraction
  - Simplifies verification
  - … but is it valid?
    - Type violation attack

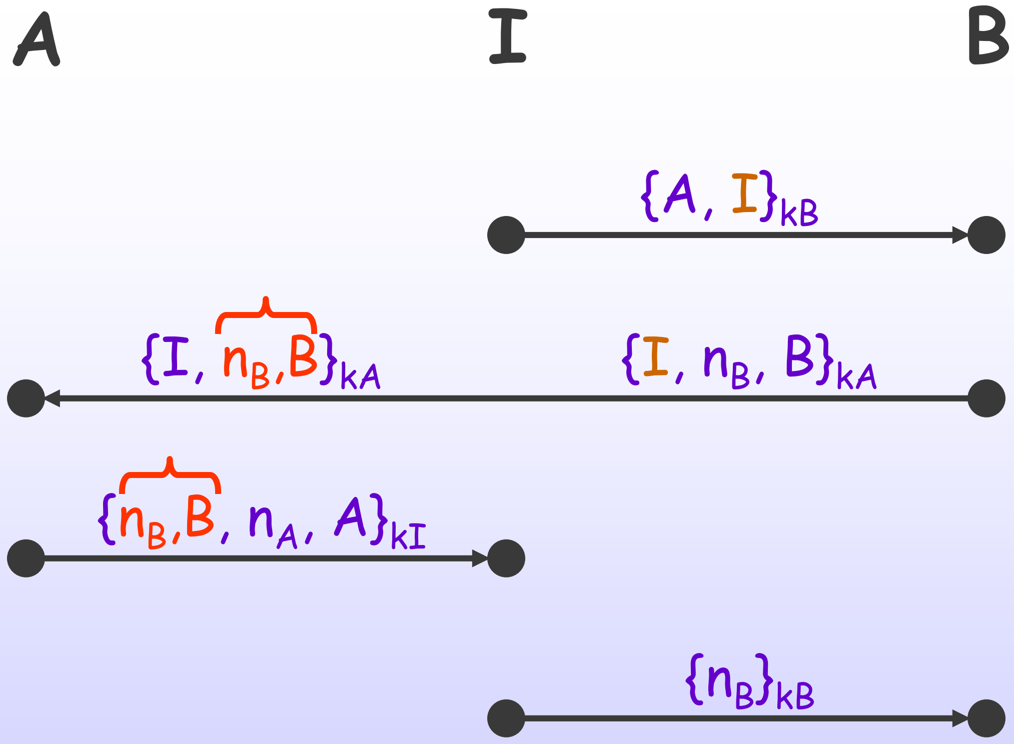
Type-Flaw Attacks

- Functionalities seen as "types"
  - Names
  - Nonces
  - Keys, …
- Violation
  - Principal misinterprets data
- Type flaw/confusion attack
  - Intruder manipulates message
  - Principal led to misuse data

Example: NSL

[Millen]

$A \rightarrow B: \{A, n_A\}_{kB}$
$B \rightarrow A: \{n_A, n_B, B\}_{kA}$
$A \rightarrow B: \{n_B\}_{kB}$



A
I
B
$\{A, I\}_{kB}$
$\{I, \overbrace{n_B, B}\}_{kA}$
$\{I, n_B, B\}_{kA}$
$\{\overbrace{n_B, B}, n_A, A\}_{kI}$
$\{n_B\}_{kB}$

Confusion 1:
name/nonce

Confusion 2:
pair/nonce

B is fooled!

"Unlikely type violation"

Advocates

Type-flaw attacks are serious threats

- Push type-free specifications
  - Catch all "normal" attacks
  - … and type-confusion attacks too
  - Types are not real!

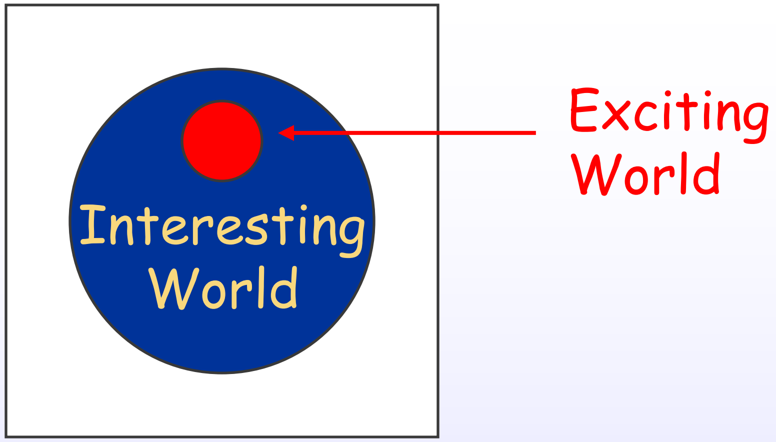
Opponents

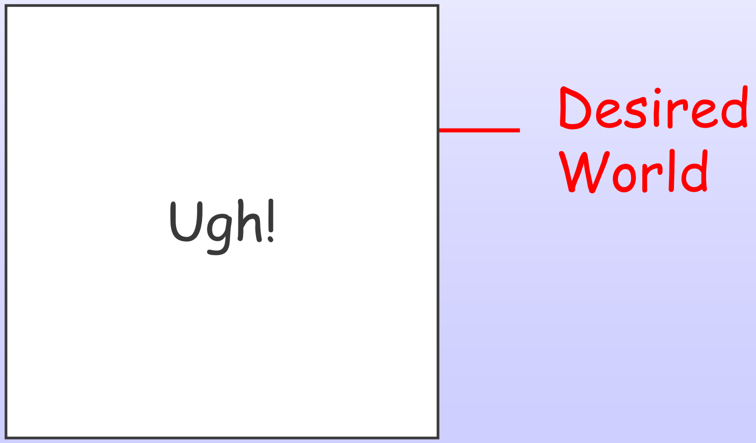
Most type-flaw attacks are unrealistic

- Push typed specification languages
  - Catch "real" attacks
  - Types guide search $\Rightarrow$ fast
  - Type-flaw attacks too low-level anyway

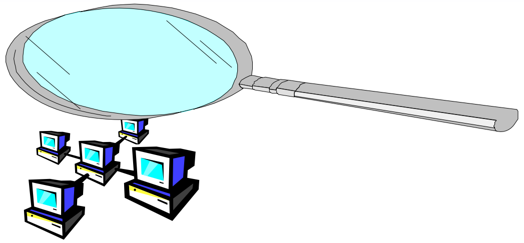
Prog. Languages vs. Security

- Types in programming languages
- Types in security





… in Reality



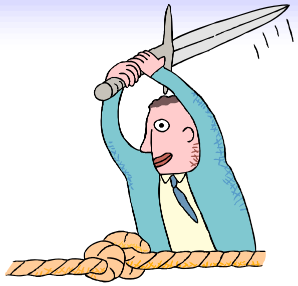
Type discriminants

- ➢ Data length
- ➢ Redundancy
- ➢ Explicit checks

- Resolve many situations …
- … but not all

"I so far found only one realistic type-flaw attack" [Meadows]

Types as Tie-Breakers



- Reconcile
  - Typed languages
  - Type violations
- User specifies confusable types
  - Flexible
  - Abstract
- Support efficient simulation

Types of Terms

- A: princ
- n: nonce
- k: shK A B
- k: pubK A
- k': privK k
- … (definable)

Types can <u>depend</u> on term

- Captures relations between objects

Subtyping

```
princ    :: msg
nonce    :: msg
pubK A   :: msg
```

- Allows atomic terms in messages
- Definable
  - Non-transmittable terms
  - Sub-hierarchies

Expressing Type Violations ?

- Impossible !

$$\frac{\boxed{\Sigma \mid\!- t : \tau}}{[S]^{R\,(\forall x:\tau.r,\rho)}{}^{A}_{\Sigma} \rightarrow [S]^{R\,([t/x]r,\rho)}{}^{A}_{\Sigma}}$$

Typing forces principal to play by the rules

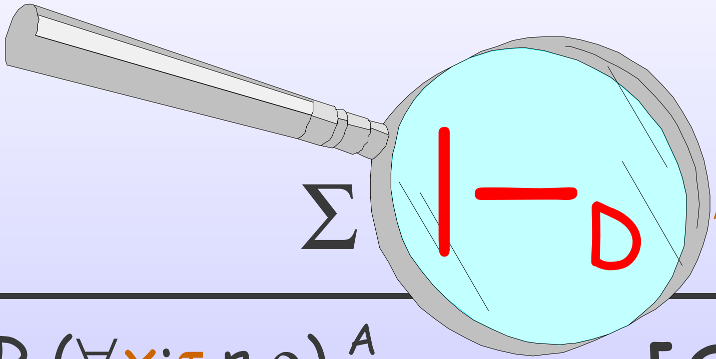
Expressing Type Violations !

Distinguish

- Static type-checking
- Dynamic type-checking

How things should be on paper

How things are in realty



$$\frac{\Sigma \vdash_D \tau}{[S]^{R\,(\forall x:\tau.r,\rho)\,A}_{\Sigma} \rightarrow [S]^{R\,([t/x]r,\rho)\,A}_{\Sigma}}$$

Subtyping Revisited

- Most rules have a rigid format

$$\frac{}{\Gamma, a{:}\tau, \Gamma' \mid\!- a : \tau}$$

- Subtyping provides hook

$$\frac{\tau' :: \tau \qquad \Gamma \mid\!- t : \tau'}{\Gamma \mid\!- t : \tau}$$

Extend subtyping with confusable types