

Assurance

Assurance

Some concepts and strategies help:

- specifications and proofs,
- economy of mechanism,
- the trusted computing base (TCB),
- open design,
- maybe open source?

Specifications and proofs

Specifications and proofs of correctness sometimes help:

- in some operating systems,
- for cryptographic primitives,
- for security protocols,
- for language run-times,

:

But specifications and proofs remain:

- relatively rare,
- expensive,
- not always convincing.

The TCB

TCB: the collection of hardware, software, and set-up information on which the security of the system depends.

Also: the part of the system that has to be right.

Ideally,

- it should be defined precisely,
- it should be small and simple,
- it should be specified,
- it should be tested,
- it should be verified.

Also: the part of the system that may appear to violate its security policy.

So:

- It is easiest to put lots of dubious code in the TCB.
- The TCB gets big.
- The TCB is not trustworthy.

See Windows (with printer drivers in the kernel).

Open design

Security of a mechanism should not depend on an attacker's ignorance of the design.

Kerchoff's Principle (1883):

The security of a cryptosystem must not depend on keeping the algorithm secret.

⇒ No "security by obscurity".

Assurance (cont.)

Assurance may also concern:

- the scope of security measures,
- usability,
- process.