Security

Rick Cox

UW CSE

Security

(Borrowing from Steve Gribble, and in turn, David Wagner)

- Techniques for computing in the presence of adversaries.
  - Confidentiality
  - Integrity
  - Authenticity
  - Availability/Resource allocation
  - Freshness
- What makes security hard?

All About Trust

- Do I trust this? How much do I trust it?
- Trusted Computing Base
  - The components that must be correct and integral to maintain your security.
  - Large or small TCB?
  - What is in your TCB?
  - What is in the department's TCB?
  - TCB must be designed with security in mind (e.g. Windows).
- Have to trust something (or just not use computers).
  - Have to trust keyboard?

Cryptography

- Techniques for communicating in the presence of adversaries.
- If used correctly and integrated with other mechanisms, a node in a distributed system can be assured of confidentiality, integrity, authenticity, and freshness.
- Can be used to reduce size of TCB, but can also increase it.
- Crypto does not provide security alone.
- Hard to get right. (If you get it wrong, Wagner et. al. will prove it.)
- E.g. 802.11's WEP, GSM phones, XBox...
- May create other security problems (availability).

Reuse, Reuse, Reuse

- A good software engineering principle in general.
- Especially good in security/cryptography, since it is so hard to get right; rather than reinvent, use something that has been widely studied.
- But: Principle of least common mechanism.

Principle of Least Privilege

- Grant any *principal* the least privileges necessary for it to function.
- Not UNIX root (all or nothing).

Design for Security

- Adding it later is very hard.
- See Internet
  - Despite being a DoD project, security was not an initial goal.
  - So now we have DoS attacks, spoofed addresses, hacking, ...

Security through Obscurity

Security through Obscurity

- Oldest principle (BC)
- GSM cellphones
  - Committee designed a new crypto algorithm, kept it secret.
  - Turned out very weak.
- Firewalls/network administrators
  - Depend on firewall at edge for protection.
  - *Always* has holes.
  - Conclusion: Firewall is the wrong place to protect end systems.
  - What is a firewall good for?

A positive example: OpenBSD

- First major OS to make security a top goal.
- OpenBSD applies cryptography where needed.
- Secure by default (OSX follows here).
- Audits code aggressively.
- Full disclosure.
- Has taken steps to make UNIX design more amenable to principle of least privilege.
- 5 years without a remote root exploit.