Midterm Exam Review

COMPSCI210 Recitation
5 Oct 2012
Vamsi Thummala
My biased view

• A bit challenging
  – Tests your reading comprehension skills under time pressure

• Thought provoking
  – Not a brain dump test

• Opportunity for learning
  – Just an another class

• Tests your confidence
  – Not difficult to score but easy to get lost
Terminology so far ..

- Entity
  - Principal, Identity, Attribute, Label
- Component
- Context
- Connector
- Channel
- Trusted computing base
- Authentication
- Authorization
- Reference monitor
  - Subject, Object, Guard
- Process
- Thread
- Kernel
- Address space
- Files
- Pipes
- Sockets
- Binder
- Event
Systems/Abstractions

• Traditional single node
  – Unix
  – Linux

• Mobile
  – Android/Linux

• Web
  – Chromium/Linux

• Network basics
  – Client/Server
Security, an overview

We reduce it to three intertwined issues:

1. What program am I running?
   - Can this program be trusted? Who says?
   - Can I be sure that the program has not been tampered?

2. Who am I talking to?
   - Can this entity be trusted?
   - Can I be sure the communication has not been tampered?

3. Should I approve this request? $R(\text{op, subject, object})$
   - Who is the requester? (subject)
   - What program is speaking for the requester?
   - Does the subject have the required permissions?
Elements of security

• **Isolation/protection**
  – Sandboxes and boundaries prevent unchecked access.

• **Integrity**
  – Fingerprint data to detect tampering.
  – Encrypt data to prevent access or tampering.

• **Authentication**
  – Identify a peer by proof that it possesses a secret.

• **Identity and attributes**
  – Identities have credentials: names, tags, roles...

• **Authorization == access control**
  – Guard checks credentials against an access policy.
Access Control

• Triplet
  – \{\text{op}, \text{subject}, \text{object}\}
• Components run within contexts (isolated sandboxes).
• Each component/context is associated with an identity with some attributes (subject).
• Components use system calls to interact across context boundaries, or access shared objects.
• Each object has some access attributes.
• Principle of least privilege limits the damage a component can do if it “goes rogue”.
Crypto primitives

- Encrypt/Decrypt
  - Use a shared secret key (symmetric)
  - or use a keypair, one public, one private (asymmetric)

- Signing

- Secure hashing
  - useful for fingerprinting data
Cryptography for Busy People

- Standard crypto functions parameterized by **keys**.
  - Fixed-width “random” value (length matters, e.g., 256-bit)
  - Symmetric (DES: fast, requires shared key K1 = K2)
  - Asymmetric (RSA: slow, uses two keys)
- “Believed to be computationally infeasible” to break
Two Flavors of “Signature”

• A digest encrypted with a private asymmetric key is called a **digital signature**
  – “Proves” that a particular identity sent the message.
    • “Proves” the message has not been tampered.
    • “Unforgeable”
  – The sender cannot deny sending the message.
    • “non-repudiable”
  – Can be legally binding in the United States
• A digest encrypted with a shared symmetric key is called a **message authentication code (MAC)**.
  • faster, but...
Nonce

- Verifies the freshness of a message
- Eavesdropping
  - serverNonce
- Tampering
  - clientNonce