ECE455 midterm review

Jonathan Lam

10/20/21

Contents

1 First paper

- Security researchers are focusing on the wrong problems, don't understand the true nature and risks

2 Trustworthy medical devices

- Trustworthy software: dependable (ibnlt: relaibility, safety, security, availability, and maintability) and consumer-responsive; has consumer trust and meet even unstated needs
- Specification bugs as important as implementation bugs
- Engineering stages/possible causes of error: requirements specification, design, human factors, implementation, testing, maintenance
- Need to consider security in the context of the system
- Need to use modern software engineering principles, e.g., beginning coding with security in mind
- Threat vs. vulnerability
- Specify functional requirements rather than technologies
- Enable open research

3 Diebold

- Closed source, bad software principles
- Response arrogant, believes too strongly in the reliability of people, thinks networking not attachable it not connected to the Internet
- Hardcoded keys, outdated security methods, legacy software
- Responses don't really adress many of the claims

4 Password Manager

- Bookmarklet vulnerabilities
  - Use iframes (run in iframe origin, not untrusted current origin)
  - HMAC in bookmarklet code to provide click authentication
- Web vulnerabilities
  - XSS (via sanitization, CSP), CSRF (via tokens)

    - Secrets in JS files should be avoided: JS files should not be user-customized because they can be included from any origin

- Authorization vulnerabilities
    - Insufficient authorization
    - Predictable identifier
- User interface vulnerabilities
    - iframe logins or new tab

5 PGP

- (and GPG) – overall model, secured by Web of Trust, fingerprints, or Trust on First Use
- No adoption: easy to make mistakes (UX problem); no confidence in long term keys
- Need a key client, need specialized clients for desktop and mobile; need to maintain keys
- Still need a passcode for the key
- Header is still not encrypted

6 MULTICS

- Security kernel: cannot ensure security except through mathematical proofs; since this is infeasible for larger systems, shrink to a smaller system
- Hard to find a systematic way to find and fix bugs
- Does not guarantee the security of everything in the OS, but of everything that uses its security features; i.e., a system will be secure if all secure operations go through the security kernel
- Example process: simplifying the process structure so that it makes parallel applications into sequential algorithms; base process system abstracts away parallelism and thus reduces bugs
- Tasks: move functions out of the kernel, restructure tasks in the kernel, and repartition the kernel

7 Reference monitor

- Security kernel (implementation) is based on the idea of a reference monitor (abstract idea); all entities must go through the reference monitor to get resources
- Reference monitor must fulfill the following: completeness, isolation, verifiability
- Security needs a well-defined policy
- Simple security condition: people cannot view things they are not privileged to see
- Star property: people cannot modify things of lower privilege: prevent Trojan horse
- Methods of verification: prove that intended behavior is secure w.r.t. policy model; correctness of mappings to API specifications; kernel implementation corresponds to its specification
- Considerations: kernel/userspace, hardware/software
  - Efficient hardware support for explicit processes, memory segments, execution domains, I/O mediation

8 New directions in cryptography

- Problem of key distribution; two approaches: public key cryptosystem and public key distribution systems; both eliminate secure key distribution channels
- Problem of authentication: digital signatures (with same properties as written signatures)
- Problems of privacy and authentication (note: not integrity)
- Computational vs unconditional security (one-time pads)
- Error propagation property of block ciphers: little errors scramble entire blocks
- Attack types: ciphertext only attack (common), known plaintext attack (no backwards secrecy), chosen plaintext attack (IFF)

- Threat of dispute/repudiability
- One-way functions and computational invertibility (a.o.t. mathematical invertibility)
- Trap-door functions: require a secret to invert

9 DHE explained

- DH is not secure against MITM attacks; authenticate via STS protocol
- Safe primes
- Allows dynamic negotiation of (short-term) shared keys
- DHE can be used as a PKI, but is not since RSA has a certificate authority

10 Failures of DHE in practice

- Export-grade cryptography and vulnerabilities in TLS
- Use longer keys, disable export-grade cryptography, use safe primes, don't misconfigure groups, use elliptic curves
- MITM attack attack, needs to compute shared secret before handshake times out
- Not only HTTP attacked, other DHE systems exploitable

11 Properties of cryptographic hash functions

- Properties:
  - Compression: maps to fixed length
  - Pre-image resistance (one-way)
  - Second pre-image resistance (weak collision resistance): hard to find another x that maps to same f(x)
  - Strong collision resistance: hard to find x, x' that map to same value
- Cryptographic hashes are also known as cryptographic hashes or message digests