CSE 484 (Winter 2008)

Computer Security and Privacy

Tadayoshi Kohno

Thanks to Dan Boneh, Dieter Gollmann, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

What is Computer Security?

- Systems may fail for many reasons
- Reliability deals with accidental failures
- Usability deals with problems arising from operating mistakes made by users
- Security deals with intentional failures created by intelligent parties
  - Security is about computing in the presence of an adversary
  - But security, reliability, and usability are all related

Two key themes of this course

- How to **think** about security
  - The Security Mindset - "new" way to think about systems
  - Threat models, security goals, assets, risks, adversaries
  - Connection between security, technology, politics, ethics, ...
  - This is the most important part of the course
    - Technology changes
    - But the mindset is something you can keep forever
- **Technical aspects** of security
  - Attack techniques
  - Defenses

Technical Themes

- Vulnerabilities of computer systems
  - Software problems (buffer overflows); crypto problems; network problems (DoS, worms); people problems (usability, phishing)
- Defensive technologies
  - Cryptography
  - Authentication
  - "Defense in depth"
  - Software security (like static analysis)

What This Course is Not About

- Not a comprehensive course on computer security
  - Computer security is a broad discipline!
  - Impossible to cover everything in one quarter
    - Not much language-based security
    - Moderate discussion of crypto (crypto could take a whole year of courses!)
  - So be careful in industry or wherever you go!
- Not about all of the latest and greatest attacks
  - Read bugtraq or other online sources instead
- Not a course on ethical, legal or economic issues
  - We will touch on ethical issues, but not focus on them
- Not a course on how to "hack" or "crack" systems

What Drives the Attackers?

- Adversarial motivations:
  - Money, fame, malice, curiosity, politics....
- Fake websites, identity theft, steal money and more
- Control victim's machine, send spam, capture passwords
- Industrial espionage and international politics
- Access copy-protected movies and videos
- Attack on website, extort money
- Wreak havoc, achieve fame and glory

Challenges: What is "Security?"

- What does security mean?
  - Often the hardest part of building a secure system is figuring out what security means
  - What are the assets to protect?
  - What are the threats to those assets?
  - Who are the adversaries, and what are their resources?
  - What is the security policy?
- Perfect security does not exist!
  - Security is not a binary property
  - Security is about risk management

From Policy to Implementation

- After you've figured out what security means to your application, there are still challenges
  - How is the security policy enforced?
  - Design bugs
    - Poor use of cryptography
    - Poor sources of randomness
    - ...
  - Implementation bugs
    - Buffer overflow attacks
    - ...
  - Is the system usable?

Don't forget the users! They are a critical component!

Many Participants

- Many parties involved
  - System developers
  - Companies deploying the system
  - The end users
  - The adversaries (possibly one of the above)
- Different parties have different goals
  - System developers and companies may wish to optimize cost
  - End users may desire security, privacy, and usability
  - But the relationship between these goals is quite complex (will customers choose not to buy the product if it is not secure?)

Approaches to Security

- Prevention
  - Stop an attack
- Detection
  - Detect an ongoing or past attack
- Response
  - Respond to attacks
- The threat of a response may be enough to deter some attackers

Whole-System is Critical

- Securing a system involves a whole-system view
  - Cryptography
  - Implementation
  - People
  - Physical security
  - Everything in between
- This is because "security is only as strong as the weakest link," and security can fail in many places
  - No reason to attack the strongest part of a system if you can walk right around it.

Analyzing the Security of a System

- First thing: Summarize the system as clearly and concisely as possible
  - Critical step. If you can't summarize the system clearly and concisely, how can you analyze it's security?
- Next steps:
  - Identify the assets: What do you wish to protect?
  - Identify the adversaries and threats
  - Identify vulnerabilities: Weaknesses in the system
  - Calculate the risks
  - Evaluate controls / mitigation strategies, and iterate

Many Desirable Security Properties

- Authenticity
- Confidentiality
- Integrity
- Availability
- Accountability and non-repudiation
- Freshness
- Access control
- Privacy of collected information
- ...

Software Security

- Issues:
  - Input validation, Buffer overflows, Integer overflows, Implicit casting, XSS
  - TOCTOU
  - Poor randomness
  - Timing attacks
- Defenses:
  - Within the language (e.g., strong types)
  - Within the system/compiler (e.g., Canaries)
  - External to the language and system (e.g., static or dynamic analysis, fuzz testing)
  - General principles (e.g., small TCB, securely delete memory, least privileg)

Cryptography

- Understand different models:
  - Symmetric vs asymmetric
  - PKI
- Basic components, and examples of them:
  - Hash function (SHA-1, SHA-256, ...)
  - Block cipher (AES, DES, 3DES, ...)
  - MAC (HMAC-SHA1, ...)
  - Symmetric encryption scheme (AES in CBC or CTR mode, ...)
  - Digital signatures (e.g., RSA, DSA)
  - Asymmetric encryption scheme (e.g., RSA)
  - Key agreement (Diffie-Hellmen key exchange)

Cryptography

- Don't need to know internals of individual components.
- But definitely need to know
  - Which object to use to solve which problem
  - What bad things might happen if you use the wrong tool for the job; (e.g., why do you not want to use AES in CTR mode encryption to provide integrity?)

Authentication

- Different approaches
  - What you know
  - Who you are
  - What you have with you
- Passwords
  - Challenges to use (memorability, strength, usability, ...)
  - How server should verify password (should the server store a copy of the password, a hashed copy, ...?)
  - Types of attackers to consider (related to the above bullet)
- Biometrics
  - Advantages
  - Disadvantages

Usability

- Intimate relationship between security mechanisms and usability
  - Incentives for using system?
  - Will users be able to understand how to use the system?
  - Will new security issues arise if a security system is deployed (e.g., with password managers, some usage scenarios make users less secure)
  -

Network Security

- Problems arise in
  - Routing issues
  - DNS issues
  - Path issues
  - Bugs
- Classes of attacks
  - Denial of Service
  - Impersonation
  - Eavesdropping or content modification
- Key points:
  - Failures at interaction between layers
  - Asymmetry between attacker and defender
  - Some attacks designers never considered
  - All motivations for existing security decisions (SSL/TLS, filter certain types of packets, check inputs, etc).

Big picture again

- Defense in Depth
- Cost vs benefits (recall risk analyses)

- Example: Firewalls
  - Stateless packet filters
  - Stateful, application-aware firewalls
  - Application proxies

Web security

- Key issues
  - Browser is the new OS
  - State on client
  - Integrity (e.g., input validation, for pricing)
  - Privacy (e.g., cookies, tracking)
  - Website isolation (e.g., cross-site scripting)