

Lecture 1: Security Concepts

ECE1776
David Lie

1

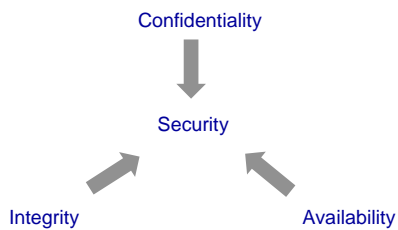
Definition of Security

- Security is a very nebulous term -- what does it mean to be secure?
 - Varying definitions, but in the end often one has to rely on intuition
 - From real life, people have an intuitive idea of what is secure, and these can for the most part be applied to computer systems
 - However, as a science, one would like a more formal definition



2

Components of Security



3

Confidentiality

"The protection of information or resources from exposure."

- There are 2 aspects of information or resources that are often important to conceal:
 1. The *content* of a piece of information or resource. Ex: most people want to keep their account passwords confidential. In many cases, content may also cover things such as configuration, cost or other attributes.
 2. The *existence* of information or resource. In this case, the mere knowledge that something exists is damaging. Ex: a company may want to keep the existence of a new product secret until it is ready for commercial introduction to the market.



4

Integrity

"The trustworthiness of information or resources."

- Integrity tells you how ready you should be to believe something. Like confidentiality, there are 2 aspects of integrity that are important:
 1. The correctness of the *contents* of a piece of information or resource. Ex: by altering the contents of a piece of e-mail, you are violating the integrity of its contents.
 2. The correctness of the *origin* of a piece of information. Ex: by altering the sender's e-mail address, you can mail an e-mail look like it's coming from someone else. Ensuring that something came from where you think it did is called *authentication*.



5

Availability

"The ability to access or use information or resources as desired"

- In real terms, a resource is available if it is accepting and responding to requests. Information is available if the service which stores that information is up and running. Availability is generally more difficult to deal with for 2 reasons:
 1. In automata theory, availability is not a finite property. Unlike confidentiality or integrity, you can't always say that at time t some object has become unavailable.
 2. Many systems deal with availability in a probabilistic fashion. Components in any system have some inherent level of unreliability and the unreliability of the system is the composition of unreliability of these components. However, availability cannot be treated probabilistically for security since an active adversary will cause unlikely events to occur more often (i.e. network flooding)



6

CIA = Security

- Any system that can be called secure provides all three (Confidentiality, Integrity & Availability) of these attributes to some degree. Remember that no system is absolutely secure, and so no system can provide all or any of the three absolutely.
- Measures:
 - Confidentiality & Integrity are often provided by cryptographic algorithms. Their strength is often measured in terms of complexity (how long will it take to break the algorithm). Ex: 256-bit keys are considered more secure than 128-bit keys.
 - Availability is very hard to measure. Traditional measures (probabilistic) measure availability in terms of percentage of time a system is accessible. A system with 99.999% (five 9's) availability is only down 0.001% of the time. Unfortunately, this doesn't apply well to measuring security.



7

Assessing Security: Threats

- A *threat* is a potential vector for a system's security to be compromised.
 - When an attacker exercises a threat, it becomes an *attack*.
 - If the attack is successful, a system's security is then *compromised*.
- Threats can come in many forms, and a good security practitioner learns to identify and assess their seriousness:
 - A system hooked up to the Internet. Network traffic that arrives from the Internet and is accepted by the system is a threat.
 - The University allows students to hook laptops up to their wireless network. The laptops pose a threat because they may transport viruses they are infected with.



8

Assessing Security: Vulnerabilities

- A *vulnerability* is a flaw in a system that has a security implication.
- Vulnerabilities are very difficult to identify (especially after any reasonable amount of testing). They are almost always serious:
 - A unchecked string copy allows an attacker to overflow a buffer and execute arbitrary code in a privileged program.
 - During configuration, a system administrator forgets to disable debug mode on a program, allowing an attacker to gain administrator privileges
 - A naïve user does not change the default password on their router from the factory default. An attacker who is experienced with the router guesses their password and gains access to their network.



9

Threats and Vulnerabilities

- Compromises occur when an attacker matches a threat (think of these as the attacker's arsenal), with a vulnerability (these are weaknesses in the system).
- In the previous slide, all vulnerabilities were accompanied by an attack the attacker used. By removing the threat, a security practitioner can prevent the attack:
 - Even though the router's password was not changed, the router is behind a firewall that prevents the attacker from connecting to the router directly. The vulnerability cannot be exploited.



10

Human Factors: User Awareness & Assurance

- Humans are actually the leading causes for computer security breaches. They are prone to making mistakes:
 1. Humans make configuration, design and implementation errors. To counter this, people try to find their own (and other's) mistakes. The amount of checking that has been done to remove errors is defined by the level of *assurance*.
 2. Humans are not all equal in terms of knowledge and education. Security knowledgeable users will create fewer vulnerabilities than unknowledgeable users. *When assessing a system, always keep in mind who created it and who is going to use it.*



11

Trust

- Trust defines how much exposure a system has to a particular interface. The more a system trusts a component, the more likely that component will be a serious threat, and the more likely that threat will find a vulnerability.
- The danger is that rather than actively assigning trust, trust in system is usually assigned via the assumptions the designer makes.



12

Introduction to Cryptographic Mechanisms

13

Reasons we need Cryptography

- Cryptography is an indispensable tool for security
 - Provides very powerful guarantees
 - Much encryption seems impossible to break (though surprisingly no concrete proof)
 - However, it's not infallible, must understand it to use it
- Cryptographic techniques can provide
 - Confidentiality/Secrecy
 - Integrity
 - Authentication
 - Nonrepudiation



14

Basic Terminology

- Plaintext or Cleartext:
 - Data that is plainly readable and understandable by anyone
- Ciphertext:
 - Data that has been processed so that only authorised principals can read it
- Encryption:
 - The process of making ciphertext into plaintext
- Decryption:
 - The opposite of encryption
- Key:
 - A value that is used with encryption or decryption to make the particular process unique
- Channel:
 - A means of communicating data between parties



15

Cryptographic Mechanisms Overview

- Ciphers:
 - Symmetric or Private-Key Ciphers
 - Asymmetric or Public-Key Ciphers
- One Way Hashes:
 - Message Authentication Codes
 - Digital Signatures



16

Symmetric Ciphers

- Examples AES (Rijndael) and DES
 - Single key used for encryption and decryption
 - Pretty fast when implemented in hardware

The diagram shows a Sender (Alice) and a Receiver (Bob) both using a shared Secret Key. Alice uses the Secret Key to encrypt Plain Text into Cipher Text, which is then sent to Bob. Bob uses the same Secret Key to decrypt the Cipher Text back into Plain Text.

- How do we securely distribute the keys?

17

Asymmetric Ciphers

- RSA and DSA are the common ones
 - Pairs of keys
 - Public key and is used to encrypt data
 - Private key and is used to decrypt data

The diagram shows a Sender (Alice) using a Public Key to encrypt Plain Text into Cipher Text, which is then sent to a Receiver (Bob) who uses a Private Key to decrypt it.

- Public-key ciphers are very slow
 - Typically use hybrid systems with both ciphers together

18

Hybrid Crypto System

The diagram shows a Sender and Receiver. The Receiver sends a public key to the Sender. The Sender randomly selects a symmetric key and uses it to encrypt a Data Payload. The Sender then uses the Receiver's public key to encrypt the symmetric key. The Sender sends the encrypted data and the encrypted key to the Receiver. The Receiver receives the encrypted code with the encrypted key and uses the symmetric key to decrypt the data.

19

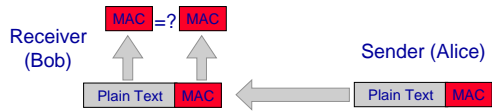
Cryptographic One-way Hashes

- Hash Function (CRC, Parity)
 - Converts a *pre-image* into a *hash value*
 - Lossy compression function
 - Hash value is shorter than pre-image
 - This means you have the problem of collisions (two pre-images with the same hash value)
- One-way hash function (SHA, MD5)
 - Difficult to reverse
 - Hard to find a pre-image that matches a given hash value
 - Collision-resistant/collision-free
 - Hard to find two pre-images that have the same hash value

20

Message Authentication Codes

- Message Authentication Codes (MAC)
 - Hash with a key that makes hash value unique
 - Attacker cannot compute hash value from pre-image
 - Sender and receiver must both know the key
 - Ensure that the message is not modified in transit



- Bob compares computed value with Alice's MAC
 - This is called *verification*



21

Digital Signatures

- Combine public-key cryptography and One-way Hashes
 - Provides Authentication, Integrity and Nonrepudiation
 - The message has not been tampered with
 - The message is from the person we think sent it
- Alice creates a hash of the message and then encrypts it with her private key
 - Bob can decrypt the hash with Alice's public key and verify it against the message
 - Since only Alice has the private key, only she could have produced the signature
 - So since the message is signed by Alice, and only Alice has the key, then she cannot deny that she signed the message
 - This is useful in digital contracts



22