Distributed Systems - Security and PKI

P-O Östberg

2007-09-14

1. Security
   - Analysis
   - Attacks
2. Cryptography
   - Encryption Algorithms
   - Public Key Infrastructures
   - Encryption Systems
3. Security Systems

Security

- Freedom from danger, risk, etc.; safety.
- Something that secures or makes safe; protection; defense.
- Precautions taken to guard against crime, attack, sabotage, espionage, etc.
- An assurance; guarantee.
- *Archaic.* overconfidence; cockiness.

(definitions from *dictionary.com*)

Policy vs Mechanism

- A security policy is a statement of what is, and what is not, allowed
- A security mechanism is a method, tool, or procedure for enforcing a security policy

Security Policies

- Defines "secure" for a system or a set of systems
- Contains (security) purpose and goal
- Foundation for security mechanism selections
- Establishes who is responsible for what
- Defines what is allowed and what is not
- Motivates why the policy looks like it does

System Properties

- Authentication
- Authorization
- Confidentiality
- Integrity
- Availability
- Non-repudiation

System Threats

- Snooping (disclosure)
  - Confidentiality
- Modification, alteration (deception, usurpation)
  - Integrity
- Masquerading, spoofing (deception, usurpation)
  - Integrity
- Repudiation (deception)
  - Integrity, non-repudiation
- Denial of receipt (deception)
  - Integrity, availability
- Delay (usurpation, deception)
  - Availability
- Denial of service
  - Availability

Security Goals

- Prevention (may hinder system availability)
- Detection (does not prevent system compromization)
- Recovery
  - stop attack
  - assess damage
  - repair damage (complex)
  - retaliate (?)

Cost-Benefit Analysis

- Weighing security mechanism costs against expected utility
- Adding security usually more expensive than including them from the start

Risk Analysis

- Required level of protection is a function of the probability of attack and the expected damage of an attack
- Risk is a function of environment
- Risks (via environments) change over time
- Remote risks still exist
- Analysis paralysis
- Legal issues
- Psychological factors (airport security)

Threat Analysis

- Level of protection
  - Physical security
  - Psychological security (awareness, knowledge etc)
  - Virtual security
- Trust models
- System life cycles (do we need to protect old systems?)
- Detection
- Reaction

Threat Analysis

- Protect against the greatest risks
- Value of protected items?
- Loss expectancy (immediate and annual)
- Attack trees
- Failure Modes and Effect Analysis (FMEA)
  - Bottom up
  - How does component failure affect the system?
- Combine as a matrix

Attack types

- Passive: hard to detect & hard to prevent
  - surveillance
  - publication of message contents
  - traffic analysis
- Active: easier to detect, hard to prevent (alt detect and recover)
  - masquerade
  - replay
  - alteration of message content
  - denial of service

Practical Authentication

- Biometric methods
  - face scanners, fingerprint readers, retinal & iris scanners, voice recognition etc
- Pass tokens
  - signatures, keys, magnetic cards, smart cards etc
- Human guards
  - id cards
- Password-based systems
- Hybrids

Methods can be attacked

Password attacks

- Snooping (looking over the shoulder, keyloggers)
- Eavesdropping
- Trojans (false login prompts)
- Information leakage (logs)
- Stealing the password database
- Online guessing
- Offline guessing

Attackers

- Non-secured resources
- Careless / untrained users / sysadmins
- Thieving users
- Users assisting external attacks
- Socially engineered attacks

(Active) Attack Methodology

1. Identify target
2. Gather information
3. Analyze information / locate vulnerabilities
4. Gain access
5. Execute attack
6. Erase attack traces

Usually enough to stop one of the above

Cryptography

- Mathematical tools for enabling trust
- Based on fundamental assumptions
  - algorithms are safe (there are no shortcuts)
  - parameter space searches for keys takes a long time
  - techniques used as intended
- Message: data
- Algorithm: the encryption method
- Key: encryption key, parameter to encryption algorithm
- Cipher text: the encrypted message

One-Way Encryption

- Messages encrypted using secret keys
- Messages can not be decrypted
- Cipher texts (to a high probability) uniquely mapped to message content
- Cipher texts used instead of messages in situations where messages must be kept secret (e.g., passwords)
- Closely related to hashcodes and Message Authentication Codes (MACs)

Symmetric Encryption

- Commonly referred to as *private key encryption*
- Messages encrypted and decrypted using the same key
- Anyone with access to the key can decrypt the message
- Fast
- Suffers from *the key distribution problem*

Encryption modes

- Electronic Code Book (ECB)
  - each block encrypted by itself
- Cipher Block Chaining (CBC)
  - each block XORed the previous encrypted block
  - Initialization Vector (IV) used for the first block
- Output Feedback (OF)
  - repeated encryption of the IV yields keystream

Encryption modes

- Counter Encryption (CE)
  - key x = (IV + i) key
  - similar to OF
  - less costly and less sensitive to birthday attacks
- Cipher Feedback (CF)
  - self-synchronizing stream ciphers
  - one of the output bits is added to the next message bit
- Message Authentication Code (MAC)
  - use CBC, save only the last ciphertext block = MAC
  - provides integrity and authenticity

Challenge-Response

1. clients says "hi, I'm xxx"
2. server says "hi xxx, here's a challenge"
3. client encrypts challenge (using a cryptographic checksum of the secret key and the challenge) and sends it
4. server encrypts challenge and compares results

challenge contains

- a suitably large random block
- an timestamp
- client identity

Asymmetric Encryption

- Commonly referred to as *public key encryption*
- Messages encrypted using key pairs (public & private)
- One key used for encryption, the other for decryption
- Public key distributed as much as possible
- Private key kept secret
- Versatile and more secure than symmetric algorithms
- Slow

Asymmetric Encryption

- Encrypt message using public key = encryption
- Encrypt message using private key = signature
- Encrypt message using receivers public key and senders private key = message both encrypted and signed
- As long as the keys can be trusted
  - messages can be kept secret (only receiver can decrypt)
  - senders and receivers can be authenticated
  - message content can be trusted
  - (acknowledged) messages cannot be reputed

Certificates

- Certificate = signed tuple of public key & identity
- Certificates can be self-signed or signed by others
- Self-signed certificates can be used for encryption (but suffer from *the key distribution problem*)
- Certificates signed by trusted parties can be used for encryption, authentication and message integrity checks

Public Key Infrastructures (PKI)

- Virtual infrastructures consisting of clients, servers and Certificate Authorities (CA)
- CAs are trusted third parties which provide signed certificates (i.e., signs public keys)
- CA certificates are distributed in browsers and similar tools (trusted and considered known by all)
- Since CA public keys are known, (signed) certificates can be validated offline (without connecting to the CA)
- Secure connections are established between parties using certificates and encryption algorithms
- Network traffic *tunneled* through encrypted channels

Secure Socket Layer (SSL)

- A protocol for establishing secure connections using certificates and cryptography algorithms
- Transport Level Security (TLS) = SSL v3.0 (almost)
- Clients use server certificate to authenticate server
- Servers use client certificate to authenticate client (optional)
- Once identities have been established, encryption keys are exchanged and symmetric encryption algorithms are used
- SSL clients uses *keystores* to manage certificates and keys

Secure Socket Layer (SSL)

Bruce Schneir, Secrets and Lies (page 239):
"As it is used, with the average user not bothering to verify the certificates and no revocation mechanism, SSL is simply a (very slow) Diffie-Hellman key-exchange method. Digital certificates provide no actual security for electronic commerce: it's a complete sham."

Keystores

- An encrypted database used to store keys and certificates
- Usually stored in a single file called `.keystore`
- Applications must provide database decryption key (username & password) to access keystore content
- Keystores only containing public keys and certificates are commonly referred to as *truststores*
- Keystores can be shared between SSL applications (usually only done for truststores)

Encryption System Security

- An encryption system is deemed secure if
  - cost of breaking cipher larger than data value
  - time required to break cipher longer than data lifetime
- Algorithm strength scales with key size
  - encryption cost increases linearly with key size
  - attack cost increases exponentially with key size
- Faster computers increase encryption strengths (as long as we use algorithms balanced to available computing power)

Encryption System Strengths

- Choice of algorithm and key length determines encryption strength (in practice: good enough and long enough is strong enough)
- Key quality issues
  - uniqueness (random number generators flawed)
  - availability (not all prime numbers known)
  - password entropy
  - dictionary attacks
  - how do we protect keys / passwords?
- Algorithm quality
  - use tried and tested algorithms

Choosing Encryption Algorithms

- Choose one matched to the currently available computational power
- Classes of algorithms available
  - asymmetric for signatures, PKIs etc
  - symmetric for data
  - stream ciphers for streaming applications
- Within class: no absolute answer
- Use standardized algorithms
- Do **not** rely on security through obscurity

Encryption System Problems

- How are keys generated?
- How are secret keys exchanged?
- How are public keys distributed?
- How do we trust security mechanisms of other organizations?
- How do we ensure that mechanisms are used correctly?

Security Systems

- A security policy is a statement that partitions the states of a system into a set of authorized (secure) states and a set of unauthorized (insecure) states
- A secure system is a system that starts in an authorized and cannot enter an unauthorized state
- A breach of security occurs when a system enters an unauthorized state.
- A security mechanism is an entity or procedure that enforces some part of a security policy.
- A security model is a model that represents a particular policy or set of policies.

Why Do Security Systems Fail?

- Vulnerabilities arise (over time)
- Security mechanisms applied in the wrong context
- Security mechanisms applied in the wrong way
- Lacking user knowledge
- System complexity inhibits correct use

Security Implementation Problems

- Security rarely adds functionality / benefits
- Security often adds complexity
- Human factors often compromises mechanisms
- Security resources are rarely sufficiently planned for
- Security is a process, not a product

Security Methodology

- Partition the system into parts
- Secure the weakest links first
- Channel the system
- Implement security in layers
- Fail securely
- Hide internal mechanisms
- Keep it simple
- Involve and activate the users
- Test, test and test again
- Question everything

Security Process

1. Threat
2. Policy
3. Specification
4. Design
5. Implementation
6. GOTO 1-5

Security Actions

- Protection
  - prepare for the inevitable
- Detection
  - eternal vigilance
  - watch the watchers
- Reaction
  - react: doing something better than doing nothing
  - recover
  - counterattack / retaliate
  - analyze, diagnose, implement - learn

Summary

- Systems will always be insecure to some degree
- It is not possible to create a distributed system which is both usable and completely secure at the same time (?)
- Know your tools
- Assess your risks
- Choose the right level of protection
- Monitor and update systems
- Manage risks

Next Time

- Time and global states