1.264 Lecture 16

Security basics
Case study 1: Public transport fare collection

• What is core of transit system fare collection system?
  – What are internal risks at station, bus, bus depot?
• What is the public face of the transit system?
• What systems get funds to the bank?
• What physical controls are needed?
• What controls are needed with credit cards?
• What are the risks of the Web site?
Case study 2: High tech manufacturing facility

• What electronic espionage risks are there?
• What are the Internet traffic risks?
• Does plant need to keep information at different levels of security?
• How does facility control access? Biometrics?
Definitions of system

- **System:**
  1. Product or component: protocol, smartcard, computer
  2. Collection of products, plus operating system and its communications
  3. Collection of above, plus application software
  4. Any of above, plus IT staff
  5. Any of above, plus users and management
  6. Any of above, plus customers and external users
  7. Any of above, plus environment: competitors, regulators
     - Vendors, evaluators focus on 1, 2
     - Businesses focus on 5, 6, as does Anderson, and so do we
Definitions of actors

- **Subject**: physical person: operator, principal, victim
- **Person**: physical person, company or government
- **Principal**: entity that participates in security system
  - Can be subject, person, role, communications channel or component
- **Group**: set of principals
- **Role**: function assumed by different persons in succession
- **Identity**: names of two principals that are the same person or component
Definitions of trust and secrecy

• Trusted system: one whose failure will break security policy
• Trustworthy system: one that will not fail
• Secrecy: mechanisms to limit principals who can access information
• Confidentiality: obligation to protect other person’s secrets if you know them
  Secrecy for the benefit of the organization
• Privacy: ability or right to protect your personal secrets
  Secrecy for the benefit of the individual
• Anonymity:
  Message content confidentiality
  Message source or destination confidentiality
• Authenticity: integrity plus freshness
  Participation of genuine principal, not a replay or fake
Protocol notation example

• Notation
  – $T \rightarrow G : T, \{T, N\}_{KT}$
  – Token $T$ used to enter garage $G$ (T and G are principals)
    Transmits its serial number $T$
    Then transmits its serial number $T$ and a random number used only once (nonce) $N$, encrypted with its key $KT$
    Nonce assures that message is fresh, not a replay of old message
      Can be sequential, random, or third party challenge number
  – Parking garage server:
    Reads $T$
    Looks up the corresponding key $KT$ from its database
    Deciphers $\{T, N\}_{KT}$
    Checks that the message includes $T$, and
    Checks that $N$ has not been seen before or has expected value
Exercise: challenge and response

- Vehicle anti-theft system as example
  - Vehicle key inserted into steering lock
  - Engine management unit sends random number challenge to key using short range radio
  - Key computes response by encrypting the challenge
  - Engine management unit decrypts, reads response and verifies it matches the challenge

- Exercise: write out the protocol using the notation conventions from the last slide:
  - E (engine) -> ____________________________
  - C (carkey) -> ____________________________
Solution

• E (engine) -> C : N
• C (car key) -> C : {C, N} \text{KC}
• Note the car key must send its identifier
  
  E must verify that C is valid. (More on this later)
Challenge response

- This is very common approach but has been broken repeatedly
  - Random numbers often not very random and can be grabbed or guessed by thief
  - SSL (Secure Sockets Layer) v1 was broken twice (though current SSL is very secure)
- It is also vulnerable to man-in-the-middle attacks
  - A <-> B <-> C
  - B can masquerade as C, passing A’s requests to C and sending C’s responses to A. After (fraudulent) authentication, B gains access
Basic key management example

- Alice and Bob wish to communicate
  Sam is a trusted third party (shares keys with Alice and Bob)
- Alice calls Sam, asks for key to talk with Bob
  - A -> S: A, B  \(\text{(A and B are names)}\)
- Sam sends Alice pair of certificates (ciphertexts)
  Each contains copy of key
  - First is encrypted so only Alice can read it
  - Second is encrypted so only Bob can read it
  - S -> A: \(\{A, B, K_{AB}, T\}_{K_{AS}}, \{A, B, K_{AB}, T\}_{K_{BS}}\)  \(\text{(T is time)}\)
- Alice retrieves her key, sends Bob the second certificate
  She then sends him a message that he can decrypt
  - A -> B: \(\{A, B, K_{AB}, T\}_{K_{BS}}, \{M\}_{K_{AB}}\)
Needham Schroeder

- Very similar to last slide; uses nonces instead of time stamps
- Alice calls Sam, provides nonce
  - $A \rightarrow S: A, B, N_A$
- Sam provides session key, returns nonce to prevent replay attacks, and certificate for Alice to send to Bob
  - $S \rightarrow A: \{N_A, B, K_{AB}, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$
- Alice sends certificate to Bob
  - $A \rightarrow B: \{K_{AB}, A\}_{K_{BS}}$
- Bob send challenge-response to ensure Alice is not replay
  - $B \rightarrow A: \{N_B\}_{K_{AB}}$
- Alice responds. After this, they exchange messages with $K_{AB}$
  - $A \rightarrow B: \{N_B^{-1}\}_{K_{AB}}$
Needham-Schroeder cont

• Needham-Schroeder can fail:
• Alice can wait a year between steps 2 and 3:
  Between getting the key from Sam and using it to talk to Bob
• If Charlie ever stole Alice’s key, he could impersonate Alice
  He could set up keys with many other principals (e.g. Dorothy, Freddie, Ginger, …) over a potentially long period of time
  Alice would not know of these compromised communications
  Sam would need to keep a log forever of who Alice had set up keys with to revoke them
• A variation on Needham-Schroeder is Kerberos
  Developed at MIT, used heavily in our systems
  Basis of Windows authentication also
Kerberos

- Two kinds of trusted servers:
  - Authentication server to which users log on
  - Ticket-granting server, which gives access to files and programs
    - This is more scalable than a single server
  Alice asks ticket server for access to Bob
  - A -> S: A, B
    - Server sends ticket, encrypted with A’s password, granting access to B
      - S -> A: \{T_S, L, K_{AB}, B, \{T_S, L, K_{AB}, A\}k_{BS}\}k_{AS}
      - Alice sends timestamp to resource, which confirms it’s alive
        - A -> B: \{T_S, L, K_{AB}, A\}k_{BS}, \{A, T_S\}k_{AB}
        - Bob sends timestamp incremented by one
          - B -> A: \{T_A+1\}k_{AB}
Kerberos, cont

- This fixes the Needham-Schroeder vulnerability by using timestamps instead of nonces
  - Stale keys are a problem only for their lifetime \( L \), typically measured in hours
  - However, clocks must now be synchronized
- Why don’t we use Kerberos for Internet and Web security?
  - Kerberos requires a central key server trusted by all parties
    - If it is broken, all communications are exposed
    - If it is down, no one can initiate secure connections
    - Who would such a trusted party be on the Internet?
    - It would be expensive
Smartcard banking protocols

- Used in public transport ticketing, e-cash
- Customer and transit agency share key $K$
- Customer card sends account number $C$ and transaction serial number $N_C$
  - $C \rightarrow T: \{C, N_C\}_K$
- Transit agency confirms its name $T$ and its transaction number $N_T$
  - $T \rightarrow C: \{T, N_T, C, N_C\}_K$
- Customer card sends amount and/or balance $X$
  - $T \rightarrow C: \{C, N_C, T, N_T, X\}_K$
- Redundant data sent to prevent cut-and-paste attack
Further protocols next time

Next lecture we’ll look at:
  - Encryption
  - Message authentication (against cut and paste attacks)
  - Names, or identity (who are Alice and Bob anyway?)
  - Internet security (large scale, distributed)
    - Public or asymmetric key encryption
    - Symmetric key encryption
    - Digital certificates for identity
    - Hashing for message authentication
Passwords: user issues

• The simplest security protocol is a username and password
  Often the most vulnerable piece of security
  Often used to protect other security measures
  Your browser certificate is protected by a password

• User issues
  Social engineering
    Users disclose passwords to third parties
    By accident, on purpose, or through deception
    Deception common in health care, insurance, banking
  Reliable password entry
    Users mistype passwords; password resets
  Remembering passwords
    Users write down passwords, choose weak passwords
Passwords: solutions

• There are 26 letters, 10 digits: 36 possible characters at each location in a password
  – This should be about 5 bits ($2^5 = 32$ combination)
  – Because of patterns, it’s usually only 1.5-2 bits/char
• An 8 character password is less than a 16 bit key
  – Easily broken (see the book for many attacks)
• Solutions
  – Passphrases
  – Hardware password generators
  – Biometrics (which also has problems)
Exercise

• In systems you’ve used at previous employers or educational institutions, list password attacks that you could have tried and the chances of success

• List other, related attacks you could have tried
  – Impersonating another user (borrowing or guessing or reading their username or ID card, etc.)
  – Accessing others’ personal information with your login
  – Misappropriating funds in an electronic system
  – What chances of success would you expect?
    If significant, these events probably occurred already
Access control levels

• Levels of access control
  Applications
    Typically implemented as stored passwords in data table
  Middleware (Web, XML, others)
    Highly variable, pre- or early standards these days
  Operating system (Windows, UNIX, ...)
    Groups and roles-based, not at individual level
  Hardware
    Memory management to prevent illegal access by programs
• Hardware controls are least complex and most reliable
• Application controls are most complex and least reliable
• Operating system controls have the most visible and reported bugs
  Stack overflows are the most common attack