

12. Protection/Security Interface

- 12.1 Security Threats
 - Types of Damage
 - Vulnerable Resources
 - Types of Attacks
- 12.2 Functions of a Protection System
- 12.3 User Authentication
 - Approaches to Authentication
 - Passwords
- 12.4 Secure Communication
 - Principles of Cryptography
 - Secret-Key Cryptosystems
 - Public-Key Cryptosystems

ICS 143

1

Security threats

- types of damage
 - information disclosure
 - information destruction
 - unauthorized use of services
 - denial of service
- vulnerable resources
 - hardware (CPU, memory, devices)
 - software (files, processes, VM)

ICS 143

2

Types of attacks

- from within
 - direct access as a valid process
 - browsing for information (main memory, disks)
 - leaking of information
 - indirect access via agent (perpetrator not present during attack)
 - Trojan horse
 - trap door (bypass authentication)

ICS 143

3

Types of attacks

- from outside
 - via legitimate channels
 - viruses
 - worms
 - remote execution
 - via illegitimate channels
 - wire tapping (passive or active)
 - searching of waste

ICS 143

4

Types of attacks

- viruses
 - designed to replicate themselves
 - removable storage media, email, file transfer
 - intended to cause damage
 - need a host program
 - attach to and modify host
 - execute as part of host
 - virus detection
 - check program length (virus can hide or compress program)
 - check for virus “signature” (viruses use encryption)

ICS 143

5

Types of attacks

- worms
 - intended to cause damage
 - exploit some system weakness to replicate
 - no host needed
- Example: Morris worm
 - 3 separate attacks:
 - rsh: spawn process on remote machine (trusted machines)
 - sendmail: in debug mode, may mail itself and start
 - finger: buffer overflow not checked (Figure 12-2)

ICS 143

6

Types of attacks

- remote execution
 - upload and start code on remote machine
 - mobile agent: may migrate among machines
 - unlike worm, relies on legitimate servers for migration
 - protection
 - interpret code -- safe but slow
 - sandboxing -- limit scope and capabilities

ICS 143

7

Types of attacks

- masquerading
 - impersonate process, user, service
 - used from outside:
 - steal password, login as “legitimate” user
 - break communication line, assume session
 - used from within:
 - impersonate login shell, steal password
- trail and error
 - e.g., try to guess password (from outside) or by examining password files (from within)

ICS 143

8

Functions of a protection system

- external safeguards
 - guard physical access (locks, badges, cameras)
- verification of user identity
- access control
 - can S perform f on R
- information flow control
 - can S get information contained in R (indirectly)
- communication safeguards
 - protect public/vulnerable lines: cryptography
- threat monitoring

ICS 143

9

User authentication

- approaches:
 - knowledge of some information
 - password, dialog
 - possession of some artifact
 - machine-readable cards (ATM)
 - combine with knowledge (PIN)
 - physical characteristics of person
 - fingerprint
 - hand geometry
 - face geometry
 - retina or iris scan
 - voice print
 - signature dynamics

ICS 143

10

User authentication

- problem with biometrics: uncertainty in recognition
 - system generates number $0 \leq n \leq 1$
 - bimodal distribution:
 - Figure 12-3
 - threshold must be chosen to minimize both
 - false alarms
 - acceptance of imposter

ICS 143

11

User authentication

- passwords
 - protect stored password files from access
 - prevent trial and error (guessing)
- protecting password files
 - maintain unencrypted; rely on access control
 - encrypt using one-way function H ;
 - keep only $H(\text{pw})$ with user name
 - at login, compute $H(\text{pw}')$ and compare with $H(\text{pw})$

ICS 143

12

User authentication

- preventing passwords guessing
 - system-generated
 - difficult to memorize
 - system-validated
 - accept only passwords that obey specifications (length, mix of letters/digits, upper/lower case)
 - employ password-cracking programs to reject easy-to-guess passwords
 - time-limited
 - expiration date or number of uses

ICS 143

13

User authentication

- one-time passwords
 - smart card
 - use secret function; apply to challenge n generated by system; e.g. $f(n)=3*n/2$
 - use one-way function to generate series of one-time passwords from one password pw
 - $H(H(pw))$ $H(pw)$ pw
 - intruder can derive $H(H(pw))$ from $H(pw)$ but not $H(pw)$ from $H(H(pw))$ because H^{-1} is unknown

ICS 143

14

User authentication

- system-extended passwords
 - for each pw , generate random number slt (called “salt”)
 - store: user name, slt , $H(slt,pw)$

Figure 12-4

 - testing if a string s is a valid password of any user:
 - w/o salting: check for $H(s)$
 - w/ salting: check for every $H(slt,s)$
 - salting does not reduce guessing of password of a specific user

ICS 143

15

Secure communication

- principles of cryptography

$$C = E(P,K)$$

$$P = D(C,K) = D(E(P,K),K)$$

- goals:
 - secrecy (message content not revealed)
 - integrity (message not modified)
 - authenticity of creator (prove that S created message, regardless of who sent it)
 - authenticity of sender (prove that S sent message)

ICS 143

16

Secure communication

- secret-key cryptosystems
 - symmetric: both S and R have common secret key

Figure 12-4

ICS 143

17

Secure communication

- secret-key cryptosystems
 - enforcing secrecy
 - only R can decrypt
 - enforcing integrity
 - intruder cannot produce valid message
 - enforcing authenticity of creator
 - not possible, S can deny
 - enforcing authenticity of sender
 - must prevent replay: nonce or timestamp

ICS 143

18

Secure communication

- use nonce N to prevent replay of message

$$\begin{array}{ccc} S & \xrightarrow{\quad} & R \\ & \leftarrow N & \\ & C=E(\{P,N\},K) \rightarrow & \end{array}$$

- capturing either message does not help; both are different every time

- use timestamp T to prevent replay

$$\begin{array}{ccc} S & \xrightarrow{\quad} & R \\ & C=E(\{P,T\},K) \rightarrow & \end{array}$$

- limits possible replay to a chosen time interval

ICS 143

19

Secure communication

- key distribution and authentication
 - both S and R must have the same key K
 - trusted server approach:
 - each process has a secret key to communicate with KDC
 - at runtime, process A may request session key to communicate with process B

$$\begin{array}{ccccc} \text{KDC} & \xrightarrow{\quad} & A & \xrightarrow{\quad} & B \\ & \leftarrow A,B & & & \\ & E(\{K_{AB},B,\text{tk}\},K_A) \rightarrow & & & \text{tk} \rightarrow \end{array}$$

$$\text{tk} = E(\{K_{AB},A\},K_B)$$

ICS 143

20

Secure communication

- public-key cryptosystems
 - asymmetric: different key for encryption and decryption
 - one cannot be derived from the other
 - one is made public, the other is kept secret

Figure 12-7

ICS 143

21

Secure communication

- public-key cryptosystems

$$C = E(E(P, K_S^{\text{priv}}), K_R^{\text{publ}})$$

- enforcing secrecy
 - only R can decrypt message using K_R^{priv}
- enforcing integrity
 - intruder cannot produce valid message without K_S^{priv}
- enforcing authenticity of creator
 - same as integrity: only S knows K_S^{priv}
- enforcing authenticity of sender
 - use nonce or timestamp to prevent replay

ICS 143

22

Secure communication

- Example: RSA

$$C = E(P) = P^e \text{ mod } n$$

$$P = D(C) = C^d \text{ mod } n$$

- e, n: public encryption key
- d, n: secret decryption key; d cannot be derived from e

ICS 143

23

Secure communication

- Example: RSA

- choose large prime numbers p and q; compute $n=p*q$

- Ex: p=5, q=7, n=35

- choose d as large prime number with no common factors with $(p-1)*(q-1)$

- Ex: $(5-1)*(7-1)=24$, d=5 or 7 or 11 (choose 11)

- choose e such that $e*d \text{ mod } (p-1)*(q-1) = 1$

- Ex: $e*11 \text{ mod } 24 = 1$; e = 11 or 35 or 59 or 83 ...

$$C = E(P) = P^{59} \text{ mod } 35$$

$$P = D(C) = C^{11} \text{ mod } 35$$

ICS 143

24

Secure communication

- public key distribution and authentication
 - making key public is easy, but need to authenticate:
 - when a process A uses a public key K, prove that K is A's public key
 - trusted server approach
- $$\begin{array}{c} \text{KDC} \qquad \qquad \qquad A \\ \leftarrow A, B \\ E(\{B, K_B^{\text{publ}}\}, K_{\text{KDC}}^{\text{priv}}) \rightarrow \end{array}$$
- KDC provides B's public key K_B^{publ}
 - $K_{\text{KDC}}^{\text{priv}}$ guarantees authenticity (KDC sent it)

ICS 143

25

Secure communication

- digital signatures
 - a plaintext document M is to be signed
 - generate digest: $d = H(M)$
 - H is a one-way function
 - H minimizes collisions (d is different for every M)
- Figure 12-8
- decryption authenticates sender; it proves sender send d
 - $d=d'$ authenticates M; d cannot be linked with any other document but M; i.e., sender signed M

ICS 143

26
