



Operating Systems

System Security

- The Security Problem
- Authentication
- Program Threats
- System Threats
- Securing Systems
- Intrusion Detection
- Encryption

1

Definition

"Security is a process not a product"

Bruce Schneier

- protocol: a detailed plan of a scientific or medical experiment, treatment of procedure.
- A security protocol defines the rules and procedures that govern interactions within a secure system or environment. e.g.
 - presenting an id or badge at the front gate.
 - using an ATM card to get money.

2

Access Control

- The system must determine whether a request should be carried out. E.g.
 - open a door
 - get a file
- Access control methods include
 - authenticating the requester and then checking the request against an access list, e.g.
 - visitor presents id to guard, guard checks list of visitors
 - delegating trust to a security token
 - e.g. door key, ATM card

3

How do we authenticate?

- Example: use remote control to unlock garage door.
 - for any meaningful security we need encryption:
 - $\{X\}_{KT}$ means message X is encrypted with key KT
 - Message takes form:

$$T \rightarrow G: T, \{T, N\}_{KT} \quad \text{where:}$$

T is the remote control token
 G is the recipient of the message (the garage door in our example)
 N is a nonce that ensures that a message is used only once
 T (on the RHS) is the remote control identifier (e.g. a 32 bit globally unique number).
 KT may simply be the token serial number encrypted by a master key M ,
 i.e. $KT = \{T\}_M$

4

Attacks on above protocol

- how do we ensure that the nonce is unique?
 - compare with last one received \rightarrow thief needs to record more than one exchanges
 - keep a list (finite length) \rightarrow valet attack
 - require that N has some kind of relationship with $N-1$ (e.g. N is last valid code incremented by no more than 16). (Problem?)
- Most attacks do not bother with breaking the encryption, rather they attack the protocol.
- So even if we use strong encryption we need to ensure that the protocol is also secure.

5

Challenge and Response

- More sophisticated than previous example.
- Uses a two pass protocol whereby the guard generates a *challenge* and the security token provides a *response* based on the challenge.

$$G \rightarrow T: N$$

$$T \rightarrow G: \{T, N\}_K$$

- where K is a key shared between the door and the token and N is a random number generated by the door.
- Key requirement in this case is that N is not predictable.

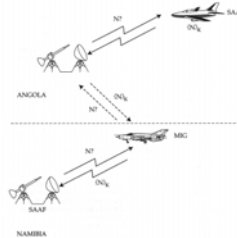
6

IFF Systems

- IFF = Identify Friend or Foe
- Used by the military to avoid shooting at friendly units.
- Unless implemented correctly IFF systems may be subjected to *man-in-the-middle* attack.

7

Man in the middle Attack



- MIGs take off as soon as SAAF bombers cross into Angolan airspace.
- SAAF anti-aircraft defenses challenge incoming MIGs.
- MIGs relay challenge to Angolan anti-aircraft defenses.
- Challenge is sent to the SAAF bombers which respond.
- Response is forwarded to MIGs which use it to cause the SAAF anti-aircraft defenses to stand down.
- *How did the protocol fail?*

8

Mutual Authentication

- Often we need to be sure that both sides in a transaction identify themselves. For example when performing banking transactions on-line both the bank needs to authenticate the users and the users need to be sure that they are talking to the correct server.
- In the IFF scenario we need to prevent our aircraft from revealing their exact positions by responding to fake challenges.

9

Other Attacks

- most attacks attempt to introduce variations in the use of the protocol. A variation must be close enough to the procedure defined by the protocol so as not to break it, but also exploit the unanticipated use to break the protocol.
- changing the message
 - off-line ATM cards
 - delayed data transfer
- manipulating the environment
 - basic assumption on ATM cash withdrawals was that only the PIN had to be secret. Output of the mag stripe reader was unencrypted and receipts often contained the full account number.

10

System Security

- Security must consider external environment of the system, and protect it from:
 - unauthorized access.
 - malicious modification or destruction
 - accidental introduction of inconsistency.
- Easier to protect against accidental than malicious misuse.

11

Authentication

- User identity most often established through passwords, can be considered a special case of either keys or capabilities.
- Passwords must be kept secret.
 - Frequent change of passwords.
 - Use of “non-guessable” passwords.
 - Log all invalid access attempts.
- Passwords may also either be encrypted or allowed to be used only once.

12

Program Threats

- Trojan Horse
 - Code segment that misuses its environment.
 - Exploits mechanisms for allowing programs written by users to be executed by other users.
- Trap Door
 - Specific user identifier or password that circumvents normal security procedures.
 - Could be included in a compiler.
- Stack and Buffer Overflow
 - Exploits a bug in a program (overflow either the stack or memory buffers.)

13

System Threats

- Worms – use spawn mechanism; standalone program
- Internet worm
 - Exploited UNIX networking features (remote access) and bugs in finger and sendmail programs.
 - Grappling hook program uploaded main worm program.
- Viruses – fragment of code embedded in a legitimate program.
 - Mainly effect microcomputer systems.
 - Downloading viral programs from public bulletin boards or exchanging floppy disks containing an infection.
 - Safe computing.
- Denial of Service
 - Overload the targeted computer preventing it from doing any useful work.

14

Threat Monitoring

- Check for suspicious patterns of activity – i.e., several incorrect password attempts may signal password guessing.
- Audit log – records the time, user, and type of all accesses to an object; useful for recovery from a violation and developing better security measures.
- Scan the system periodically for security holes; done when the computer is relatively unused.

15

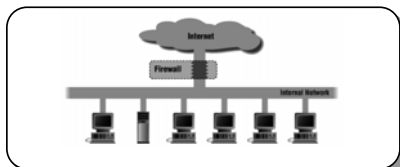
Threat Monitoring (Cont.)

- Check for:
 - Short or easy-to-guess passwords
 - Unauthorized set-uid programs
 - Unauthorized programs in system directories
 - Unexpected long-running processes
 - Improper directory protections
 - Improper protections on system data files
 - Dangerous entries in the program search path (Trojan horse)
 - Changes to system programs: monitor checksum values

16

FireWall

- A firewall is placed between trusted and untrusted hosts.
- The firewall limits network access between these two security domains.



17

Intrusion Detection

- Detect attempts to intrude into computer systems.
- Detection methods:
 - Auditing and logging.
 - Tripwire (UNIX software that checks if certain files and directories have been altered – i.e. password files)
- System call monitoring

18

Cryptographic Techniques

- What is Cryptography?
 - Algorithms reversibly converting data (plaintext) into unintelligible form (ciphertext).
 - $E(P) = C$, and then $D(C) = P$
 - only knowledge of “secret” allows recovery.
 - Kerckhoff’s doctrine (19th century).
 - Building blocks for services/protocols that provide other assurances such as privacy, integrity, authentication, non-repudiation ...
 - Keys and algorithms

19

Types of Cryptography

- Block Ciphers: we encrypt several plaintext symbols at once in a block
- Stream Ciphers: the encryption rule depends on a plaintext symbol’s position in the stream of plaintext symbols.
- Secret Key (symmetric encryption)
- Public Key (different encryption/decryption keys).
- Hash Functions.
- Digital Signatures

20

Example: Hash Functions

- Hash functions accept input of **variable** length, and return output of **fixed** length.
 - hash function denoted as $h(M)$, also called the *message digest*, or the *hash value*
- Uses:
 - passwords
 - digital signatures (cheaper to sign the hash)
 - timestamping service (timestamp the hash rather than the document) *why?*
 - Key updating ($K_{i+1} = h(K_i)$) and autokeying ($K_{i+1} = h(K_i, M_{i1}, M_{i2}, \dots)$)
- Must be careful to avoid collisions!
- Must not be easily reversible (*one way function*)

21

Other examples

- Stream ciphers: random generators
- Block ciphers: random permutations
- Attacks:
 - key recovery
 - chosen plaintext
 - related key
- Public key encryption: trapdoor one-way permutations
 - lots of scrolls - each scroll has a different name, we use the name to encrypt and give our friends the hash of the name to use it for decryption.

22

Public Key Encryption

- Give some input R , if we get two keys: KR (encrypt) and KR^{-1} (decrypt) such that:
 - Given KR , it is infeasible to compute KR^{-1} . (so its not possible to compute R either).
 - There is an encryption function such that $C = \{M\}_{KR}$
 - There is a decryption function such that $M = \{C\}_{KR^{-1}}$

23

Digital Signatures

- Like PKE, a signature scheme has a keypair generations function that given R , produces keys σR (private signing) and $V R$ (public verification), such that:
 - Given VR , its is infeasible to compute σR .
 - There is a digital signature function, the given M and key σR will produce $Sig_{\sigma R}(M)$
 - There is a signature verification function that given signature $Sig_{\sigma R}(M)$ and the public key VR , will determine whether the signature is correct.
 - Depending on the signature M may be recoverable.

24

Encryption Example - SSL

- SSL – Secure Socket Layer
- Cryptographic protocol that limits two computers to only exchange messages with each other.
- Used between web servers and browsers for secure communication (credit card numbers)
 - The server is verified with a certificate.
 - Communication between each computers uses symmetric key cryptography.

25

Computer Security Classifications

- U.S. Department of Defense outlines four divisions of computer security: A, B, C, and D.
 - D – Minimal security.
 - C – Provides discretionary protection through auditing.
 - Divided into C1 and C2. C1 identifies cooperating users with the same level of protection. C2 allows user-level access control.
 - B – All the properties of C, however each object may have unique sensitivity labels. Divided into B1, B2, and B3.
 - A – Uses formal design and verification techniques to ensure security.

26