SE-4920: Lecture 14
Real-time communication security

- Reading
  - Chapter 16
- Today's Outcomes
  - Define "real-time communication security"
  - Discuss problems unique to real-time communication security and some solutions
  - Define "perfect forward secrecy," explain why it is desirable, and show one way that it can be attained

1

Introduction

- Real-time protocol
  - Interaction to authenticate and establish a session key
  - As opposed to preparing a message for later decryption (*e.g.*, email, tape backup)

2

Which layer?

- Recall the OSI model
  - 1 physical
  - 2 datalink (*e.g.*, Ethernet MAC)
  - 3 network (*e.g.*, IP)
  - 4 transport (*e.g.*, TCP)
  - …
- Layer 3/4 typically in OS
- Two main choices are
  - "at layer 4" (right above layer 4)
    - SSL, SSH
    - No change to OS, but change applications
    - Lost data (closed connection) problem: injecting duplicate TCP sequence numbers
  - "at layer 3" (right above layer 3)
    - Change OS so that everyone using TCP gets security
    - IPsec
      - Current IPsec implementations only pass IP information, not user information, to application
      - IPsec can authenticate an IP address, though
      - Still need to authenticate user, perhaps using password, but at least that's encrypted on the network
- Each one has weaknesses – limitations of "bolting on" security

3

Session key establishment

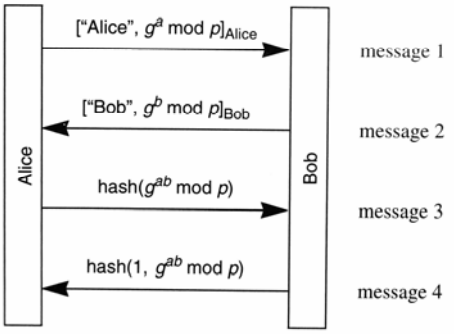
- Session hijacking
  - Vulnerability if cryptographic protection ends after the initial handshake
  - *E.g.*, forging IP packets from an address where legitimate user proved their presence
    - using higher TCP sequence numbers so legitimate traffic is ignored
- Solution: use a session key for confidentiality, integrity, and to avoiding hijacking
  - Sequence numbers used to prevent replay within a session
  - Both sides should contribute to key
    - Random if *either* is random

4

Perfect forward secrecy (PFS)

- PFS – impossible for eavesdropper to decrypt conversation even with
  - Entire encrypted conversation
  - Long-term secrets captured from both parties
- The trick: generate a temporary session key and forget it ASAP
  - Session key cannot be derivable from long-term secrets
  - May use multiple keys in a long conversation
    - Compromise of running system only compromises recent part of conversation
- Non-PFS protocols
  - Just encrypting the data with public keys
  - Kerberos (key in ticket, encrypted with long term secret)
  - Alice chooses key, sends to Bob encrypted with Bob's public key
- PFS protocols usually have "escrow-foilage"
  - If keys are escrowed, the conversation is still secure
  - Key escrow still allows impersonation
  - Escrow systems often require separate keys for encryption and signing
    - Only escrow the encryption key (enable nonrepudiation)

5



Protocol 16-2. Diffie-Hellman for perfect forward secrecy using signature keys

6

Denial-of-service/clogging protections: Cookies

- DoS: *E.g.*, SYN flood
  - Normal exchange is SYN, SYN/ACK, ACK
    - SYN – synchronization packet with initial sequence for sender (x)
    - ACK – acknowledging SYN – record x and reply x+1
  - Between 2 & 3, server is waiting for connection to be fully opened
    - Resources need to maintain this state
  - In attack, client(s) never send final ACK, but send a lot of SYNs
- Cookies (not web browser cookies)
  - Eliminate the need for state on server until connection is opened
  - Replace SYN/ACK with a message corresponding to source, but unpredictable to source (*e.g.*, c = hash(source IP, secret)
    - Retain no state
  - Client must return the cookie to complete the connection
    - Connection valid if cookie matches hash(source IP, secret)

7

Endpoint identifier hiding

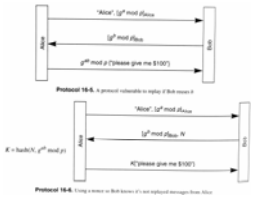
- Begin with anonymous Diffie-Hellman exchange
- Passive attacker cannot see endpoint names
- Active attacker can learn at least 1 identity
  - Fails at signature phase
- If Bob's public (encryption) key is known beforehand and used
  - Secure from active attack

Protocol 16-4. Identity hiding

8

Live partner reassurance

- Ensure key is changed each time
  - Avoid replay by attacker
- Could choose different D-H exponent
  - Somewhat expensive
- Instead choose a different nonce each time that contributes to the key
  - Nonce – number used once, usually pseudorandom



Protocol 16-5. A protocol vulnerable to replay if Bob reuses b

Protocol 16-6. Using a nonce so Bob knows it's not replayed messages from Alice

9

Session resumption

- PK authentication is expensive
  - May need to present certificates, sign D-H quantities, etc.
- May want to only do this once per day, month, etc. per client
- Give client a quantity similar to a Kerberos ticket to present later
  - Bob has a private secret S
    - Change once per day, month, etc.
  - Bob authenticates Alice
  - Bob sends $\{S_{AB}=\text{hash}(\text{“Alice”},S)\}e_A$
  - If Alice later proves knowledge of S_{AB}, authentication steps can be bypassed
- Used in Lotus Notes

10