Network Security in Practice

Kai Shen

Practices of Network Security
- Key distribution and certification
- Access control: firewalls
- Attacks and counter measures
- Security protocol case studies

Key Distribution and Certification

Symmetric key distribution problem:
- How do Alice and Bob establish shared secret key over network without Trudy’s knowledge?

Public key distribution problem:
- When Alice obtains Bob’s public key (from website, email, diskette), how does she know it is Bob’s public key, not Trudy’s?

Secret Key Distribution: Key Distribution Center (KDC)

- KDC: server shares different secret key with each registered user (many users).
- Alice, Bob know own symmetric keys, $K_{A,KDC}$, $K_{B,KDC}$, for communicating with KDC.
**Key Distribution using KDC**

*Q:* How does KDC allow Bob, Alice to determine shared symmetric secret key to communicate with each other?

- **KDC** generates $K_{A,KDC}(A,B)$
- Bob knows to use $R_1$ to communicate with Alice
- Alice and Bob communicate: using $R_1$ as session key for shared symmetric encryption

---

**Security Vulnerability with Public Key Distribution**

A case example for public key-based authentication.

- $R$ sends me your public key
- $K^*_A(K_A(R)) = R$
- and knows only Alice could have the private key, that encrypted $R$ such that $K^*_A(K_A(R)) = R$

What if Bob doesn't know Alice's public key ahead of time?

---

**Security vulnerability when public keys are not well known**

**Man (woman) in the middle attack:** Trudy poses as Alice (to Bob) and as Bob (to Alice)

- I am Alice
- I am Alice
- $R$ sends me your public key
- Bob computes $K^*_A(K_A(R)) = R$
- and knows only Alice could have the private key, that encrypted $R$ such that $K^*_A(K_A(R)) = R$

---

**Public Key Distribution: Certification Authorities**

- **Certification authority (CA):** trustable by everyone; every one knows its public key.
- $E$ (person, router) registers its public key with CA.
- $E$ provides "proof of identity" to CA.
- CA creates certificate binding $E$ to its public key.
- certificate is CA-signed document saying "$E$'s public key is ..."

Bob's public key $K^*_B$

- certificate for Bob's public key, signed by CA

---

*Encryption:*

- $m = K^*_A(K^*_A(m))$
- sends $m$ to Alice
- encrypted with Alice's public key

- $K^*_A(K_A(m))$
- Trudy gets $m$
- sends $m$ to Alice
- encrypted with Alice's public key
Certification Authorities (cont.)

When Alice wants to verify Bob's public key:
- gets Bob's certificate (Bob or elsewhere).
- apply CA's public key to Bob's certificate, verify Bob's public key.

Key Certification Methods
- Public key certificate signed by a certification authority
- Peer certification:
  - if A knows B personally, they can verify each other's public keys using offline means and sign them;
- Certificate chain leading to a certificate authority
  - CA signs A's public key certificate
  - A signs B's public key certificate
  - B signs C's public key certificate

Access Control: Firewalls
- isolates organization's internal network from the public Internet through filtering, allowing some data to pass, blocking others.
Network-layer Packet Filtering

- Firewall is built into the edge router connected to the Internet
- Router filters packet-by-packet, decision to forward/drop packet based on:
  - source IP address, destination IP address
  - TCP/UDP source and destination port numbers
  - TCP SYN and ACK bits

Policies in Network-layer Packet Filtering

- **Example 1:** blocking all incoming TCP datagrams with dest port = 80
  - No external clients can access internal Web servers.
- **Example 2:** blocking all TCP datagrams with source or dest port = 23, except for those with source or dest IP = 128.151.67.155 (a particular internal machine)
  - All incoming and outgoing telnet connections have to go through a telnet gateway.
- **Example 3:** blocking all incoming TCP datagrams with ACK bit set to 0
  - Prevents external clients from initiating TCP connections with internal clients, but allows internal clients to connect to outside.

More on Network-layer Packet Filtering

- **Advantage:**
  - transparent to network applications
  - incurring little extra overhead/latency
- **Limitation:**
  - relying only on IP/TCP/UDP header info
    - not flexible enough, e.g., firewall can know the IP of the source, but not the “user”

Application-layer Gateways

- **Access control according to application-layer information.**
- **Example:** allow selected internal users to telnet outside.

1. Router filter blocks all telnet connections not originating from gateway
   - require all telnet users to telnet through gateway.
2. For authorized users, gateway sets up telnet connection to dest host.
Practices of Network Security

- Key distribution and certification
- Access control: firewalls
  - network-layer firewall
  - application-layer firewall
- Attacks and countermeasures
- Security protocol case studies

Network Security Threat: Mapping

- Before attacking: “scout the area” – find out what services are implemented on network
- Try to determine what host addresses are valid on the network
- Port-scanning: try to establish TCP connection to each port in sequence (see what happens)

Countermeasures:
- Record traffic entering network
- Look for suspicious activity (e.g., IP addresses, ports being scanned sequentially)

Network Security Threat: Packet Sniffing

- Promiscuous NIC reads all packets passing by a broadcast media (e.g. shared-link Ethernet)
- Can read all unencrypted data (e.g. passwords)

Countermeasures:
- Checks periodically if host interface in promiscuous mode.
- One host per segment of broadcast media (switched Ethernet)
- Encrypt all packets.

Network Security Threat: IP Spoofing

- with root privilege, one can generate “raw” IP packets with any value into IP source address field
- receiver can’t tell if source is spoofed
- e.g.: T pretends to be B

Countermeasures:
- authentication
- ingress filtering – routers should not forward outgoing packets with invalid source addresses
Network Security Threat:
Cross-Site Scripting
- Cross-site scripting:
  - duped to run script unintended by the original site
  - most significant vulnerability for web applications today
- Examples:
  - attacker supplies attack string (including HTML tag and JavaScript code) as msg to msg board FOOBAR; a user who trusts FOOBAR views msgs and his/her browser would run attack JavaScript
  - search engine FOOBAR displays the input search keywords in the return page; attacker prepares a search query with attack string; a user who trusts FOOBAR clicks the search
  - attacker embeds attack strings in machine names

Countermeasures?
- Careful input checking

Disclaimer
- Parts of the lecture slides contain original work of James Kurose, Larry Peterson, and Keith Ross. The slides are intended for the sole purpose of instruction of computer networks at the University of Rochester. All copyrighted materials belong to their original owner(s).