Lecture 33

Security
April 15, 2005

Firewalls
- Idea: separate local network from the Internet

Trusted hosts and networks
Intranet
DMZ
Demilitarized Zone: publicly accessible servers and networks

Firewall Locations in the Network
- Between internal LAN and external network
- At the gateways of sensitive sub-networks within the organizational LAN
  - payroll’s network must be protected separately within the corporate network
- On end-user machines
  - “personal firewall”
  - Microsoft’s Internet Connection Firewall (ICF) comes standard with Windows XP

Firewall Types
- Packet- or session-filtering router (filter)
- Proxy gateway
  - all incoming traffic is directed to firewall, all outgoing traffic appears to come from firewall
  - application-level: separate proxy for each application
    - different proxies for SMTP (email), HTTP, FTP, etc.
    - filtering rules are application-specific
  - circuit-level: application-independent, “transparent”
    - only generic IP traffic filtering (example: SOCKS)
- Personal firewall with application-specific rules
  - e.g., no outbound telnet connections from email client
Packet Filtering

- For each packet, firewall decides whether to allow it to proceed
  - decision must be made on per-packet basis
  - stateless; cannot examine packet’s context (TCP connection, application to which it belongs, etc.)
- To decide, use information available in the packet
  - IP source and destination addresses, ports
  - protocol identifier (TCP, UDP, ICMP, etc.)
  - TCP flags (SYN, ACK, RST, PSH, FIN)
  - ICMP message type
- Filtering rules are based on pattern-matching

Packet Filtering Examples

Stateless Filtering is Not Enough

- In TCP connections, ports with numbers less than 1024 are permanently assigned to servers
  - 20, 21 for FTP, 23 for telnet, 25 for SMTP, 80 for HTTP...
- Clients use ports numbered from 1024 to 16383
  - they must be available for clients to receive responses
- What should a firewall do if it sees, say, an incoming request to some client’s port 5612?
  - it must allow it: this could be a server’s response in a previously established connection...
  - ... OR it could be malicious traffic
  - can’t tell without keeping state for each connection

Example: Variable Port Use

Inbound SMTP

Outbound SMTP

Session Filtering

- Decision is still made separately for each packet, but in the context of a connection
  - if new connection, then check against security policy
  - if existing connection, then look it up in the table and update the table, if necessary
    - only allow incoming traffic to a high-numbered port if there is an established connection to that port
- Hard to filter stateless protocols (UDP) and ICMP
- Typical filter: deny everything that’s not allowed
  - must be careful filtering out service traffic such as ICMP

Example: Connection State Table

<table>
<thead>
<tr>
<th>Source Address</th>
<th>Source Port</th>
<th>Destination Address</th>
<th>Destination Port</th>
<th>Connection State</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.100</td>
<td>1030</td>
<td>205.19.0.29</td>
<td>80</td>
<td>Enabled</td>
</tr>
<tr>
<td>192.168.1.102</td>
<td>1031</td>
<td>206.32.42.32</td>
<td>80</td>
<td>Enabled</td>
</tr>
<tr>
<td>192.168.1.103</td>
<td>1033</td>
<td>175.203.22.23</td>
<td>85</td>
<td>Enabled</td>
</tr>
<tr>
<td>192.168.1.106</td>
<td>1033</td>
<td>175.203.32.12</td>
<td>70</td>
<td>Enabled</td>
</tr>
<tr>
<td>192.168.1.103</td>
<td>1035</td>
<td>175.203.32.12</td>
<td>70</td>
<td>Enabled</td>
</tr>
<tr>
<td>192.168.1.103</td>
<td>1033</td>
<td>175.203.32.12</td>
<td>70</td>
<td>Enabled</td>
</tr>
<tr>
<td>192.168.1.103</td>
<td>1035</td>
<td>175.203.32.12</td>
<td>70</td>
<td>Enabled</td>
</tr>
<tr>
<td>192.168.1.103</td>
<td>1035</td>
<td>175.203.32.12</td>
<td>70</td>
<td>Enabled</td>
</tr>
<tr>
<td>192.168.1.103</td>
<td>1035</td>
<td>175.203.32.12</td>
<td>70</td>
<td>Enabled</td>
</tr>
<tr>
<td>192.168.1.103</td>
<td>1035</td>
<td>175.203.32.12</td>
<td>70</td>
<td>Enabled</td>
</tr>
<tr>
<td>192.168.1.103</td>
<td>1035</td>
<td>175.203.32.12</td>
<td>70</td>
<td>Enabled</td>
</tr>
</tbody>
</table>
Example: FTP

FTP client

FTP server

Data

Command

PORT 1510

DATA CHANNEL

TCP ACK

FTP Packet Filter

The following filtering rules allow a user to FTP from any IP address to the FTP server at 172.168.10.12

access-list 100 permit tcp any gt 1023 host 172.168.10.12 eq 21
access-list 100 permit tcp any gt 1023 host 172.168.10.12 eq 20
! Allows packets from any client to the FTP control and data ports
access-list 101 permit tcp host 172.168.10.12 eq 21 any gt 1023
access-list 101 permit tcp host 172.168.10.12 eq 20 any gt 1023
! Allows the FTP server to send packets back to any IP address with TCP ports > 1023
interface Ethernet 0
access-list 100 in ! Apply the first rule to inbound traffic
access-list 101 out ! Apply the second rule to outbound traffic

Weaknesses of Packet Filters

- Do not prevent application-specific attacks
  - For example, if there is a buffer overflow in URL decoding routine, firewall will not block an attack string
- No user authentication mechanisms
  - ... except (spoofable) address-based authentication
- Firewalls don't have any upper-level functionality
- Vulnerable to TCP/IP attacks such as spoofing
  - Solution: list of addresses for each interface (packets with internal addresses shouldn't come from outside)
- Security breaches due to mis-configuration

Abnormal Fragmentation

For example, ACK bit is set in both fragments, but when reassembled, SYN bit is set (can stage SYN flooding through firewall)

Fragmentation Attack

IP Networks

- Eavesdropping
- Modification of packets in transit
- Identity spoofing (forged source IP addresses)
- Denial of service
- Many solutions are application-specific
  - TLS for Web, S/MIME for email, SSH for remote login
- IPSec aims to provide a framework of open standards for secure communications over IP
  - Protect every protocol running on top of IPv4 and IPv6
IPSec: Network Layer Security

- **Network-layer secrecy:**
  - Sending host encrypts the data in IP datagram
  - TCP and UDP segments, ICMP and SNMP messages.
- **Network-layer authentication**
  - Destination host can authenticate source IP address
  - Two principle protocols:
    - Authentication header (AH) protocol
    - Encapsulation security payload (ESP) protocol
  - For both AH and ESP, source-destination handshake:
    - Create network-layer logical channel called a service agreement (SA)
  - Each SA unidirectional.
  - Uniquely determined by:
    - Security protocol (AH or ESP)
    - Source IP address
    - 32-bit connection ID

**IPSec in Transport Mode**

- End-to-end security between two hosts
  - Typically, client to gateway (e.g., PC to remote host)
  - Requires IPSec support at each host

**IPSec in Tunnel Mode**

- Gateway-to-gateway security
  - Internal traffic behind gateways not protected
  - Only requires IPSec support at gateways

**Tunnel Mode Illustration**

IPSec protects communication on the insecure part of the network.

**Transport vs Tunnel Mode**

- **Transport mode** secures packet payload and leaves IP header unchanged
  - IP header (real host)  IPSec header  TCP/UDP header + data
- **Tunnel mode** encapsulates both IP header and payload into IPSec packets
  - IP header (gateway)  IPSec header  IP header (host)  TCP/UDP header + data

**IPSec**

**IPSec = AH + ESP + IPcomp + IKE**

- AH and ESP rely on an existing security association
  - Idea: parties must share a set of secret keys and agree on each other’s IP addresses and crypto algorithms
- Internet Key Exchange (IKE)
  - Goal: establish security association for AH and ESP
  - If IKE is broken, AH and ESP provide no protection.
ESP Protocol
- Confidentiality: encrypt packet (+ ESP trailer)
- Compute Hash Message Authentication Code (HMAC) and add to ESP header
- Encryption: 3DES, AES
- Hash: MD5, SHA

Security Association
- One-way sender-recipient relationship
- SA determines how packets are processed
  - cryptographic algorithms, keys, lifetimes, sequence numbers, mode (transport or tunnel)...
- SA is uniquely identified by SPI (Security Parameters Index)...
  - each IPSec keeps a database of SAs
  - SPI is sent with packet, tells recipient which SA to use
  - ...protocol identifier (AH or ESP)

Security Association
- More than one SA can apply to a packet!
  - e.g., end-to-end authentication (AH) and additional encryption (ESP) on the public part of the network

AH (Transport Mode)
- Provides integrity and origin authentication
- Authenticates portions of the IP header
- Anti-replay service (to counter denial of service)
- No confidentiality

AH: Authentication Header
- Next header: (TCP)
- Payload length
- Reserved
- Security parameters index (SPI)
- Sequence number
- IV: Integrity Check Value
- (HMAC of IP header, AH, TCP payload)
**ESP Packet**

- **ESP Packet**
  - Identifies security association (shared keys and algorithms)
  - Anti-replay: TCP segment (transport mode) or entire IP packet (tunnel mode)
  - Pad to block size for cipher, also hide actual payload length
  - Type of payload
  - HMAC-based Integrity Check Value (similar to AH)

**Prevention of Replay Attacks**

- When SA is established, sender initializes 32-bit counter to 0, increments by 1 for each packet
- If wraps around $2^{32}-1$, new SA must be established
- Recipient maintains a sliding 64-bit window
- If a packet with high sequence number is received, do not advance window until packet is authenticated

**IEEE 802.11 Security**

- **War-driving**: drive around Bay area, see what 802.11 networks available?
- More than 9000 accessible from public roadways
- 85% use no encryption/authentication
- Packet-sniffing and various attacks easy!

**Securing 802.11**

- **Wired Equivalent Privacy (WEP)**: a failure
- Current attempt: 802.11i

**WEP Data Encryption**

- Host/AP share 40 bit symmetric key (semi-permanent)
- Host appends 24-bit initialization vector (IV) to create 64-bit key
- 64 bit key used to generate stream of keys, $k^W$
- $k^W$ used to encrypt $i$th byte, $d_i$, in frame: $c_i = d_i \oplus k^W$
- IV and encrypted bytes, $c_i$, sent in frame

**Wired Equivalent Privacy (WEP)**

- Authentication as in protocol ap4.0
  - Host requests authentication from access point
  - Access point sends 128 bit nonce
  - Host encrypts nonce using shared symmetric key
  - Access point decrypts nonce, authenticates host
  - No key distribution mechanism
  - Authentication: knowing the shared key is enough

**802.11 WEP Encryption**

- Sender-side WEP encryption
Breaking 802.11 WEP Encryption

Security hole:
- 24-bit IV, one IV per frame, -> IV’s eventually reused
- IV transmitted in plaintext -> IV reuse detected

Attack:
- Trudy causes Alice to encrypt known plaintext \(d_1, d_2, d_3, d_4, \ldots\)
- Trudy sees: \(c_i = d_i \oplus k_i^{IV}\)
- Trudy knows \(c_i, d_i\), so can compute \(k_i^{IV}\)
- Trudy knows encrypting key sequence \(k_1^{IV}, k_2^{IV}, k_3^{IV}, \ldots\)
- Next time IV is used, Trudy can decrypt!

802.11i: Improved Security

- numerous (stronger) forms of encryption possible
- provides key distribution
- uses authentication server separate from access point

802.11i: 4 Phases of Operation

1. Discovery of security capabilities
2. STA and AS mutually authenticate, together generate Master Key (MK). AP serves as “pass through”
3. STA derives Pairwise Master Key (PMK)
4. AS derives same PMK, sends to AP
5. STA, AP use PMK to derive Temporal Key (TK) used for message encryption, integrity

802.11i

- Requires new encryption key protocols, known as Temporal Key Integrity Protocol (TKIP) and Advanced Encryption Standard (AES).
- Officially ratified by the IEEE in June of 2004, and thereby became part of the 802.11 family of wireless network specifications.
- AES requires a dedicated chip, and this may mean hardware upgrades for most existing Wi-Fi networks.
- Other features, e.g., key caching, which facilitates fast reconnection to the server for users who have temporarily gone offline.