Firewall

1. Khái niệm Firewall

- **Vị trí:** Firewall được đặt giữa mạng nội bộ (premises network) và Internet.
- **Chức năng:** Tạo một lớp bảo vệ ngoại vi (outer security wall/perimeter) để:
    - Ngăn chặn các tấn công từ Internet.
    - Là điểm kiểm soát duy nhất để áp dụng chính sách bảo mật và thực hiện auditing.
- **Hình thức triển khai:** Có thể là một máy đơn lẻ hoặc nhiều máy phối hợp.

2. Mục tiêu thiết kế (Design Goals)

1. **Tất cả luồng dữ liệu** từ trong ra ngoài và ngược lại phải đi qua Firewall.
2. **Chỉ cho phép** những lưu lượng đã được xác thực theo chính sách bảo mật.
3. **Bản thân Firewall** phải miễn nhiễm với việc bị xâm nhập.

3. Các đặc điểm lọc truy cập của Firewall

- **IP Address & Protocol Values:** Lọc dựa trên địa chỉ IP nguồn/đích, số port, hướng truyền (inbound/outbound), và thông tin tầng mạng/tầng vận chuyển.
- **Application Protocol:** Lọc dựa trên giao thức ứng dụng (HTTP, FTP, SMTP...).
- **User Identity:** Kiểm soát dựa vào danh tính người dùng (thường áp dụng cho user nội bộ).
- **Network Activity:** Kiểm soát dựa vào thời gian, tần suất yêu cầu (phát hiện quét cổng, brute force...), hoặc mẫu hành vi.

4. Các loại Firewall

1. **Packet Filtering Firewall**
    - Lọc từng gói tin dựa trên tập luật:
        - IP nguồn- đích
        - Port nguồn/đích
        - Giao thức (TCP, UDP, ICMP...)
        - Interface mạng
    - Ưu điểm: Nhanh, đơn giản.
    - Nhược điểm: Không kiểm tra nội dung, dễ bị bypass nếu luật không chặt.
2. **Stateful Inspection Firewall**
    - Theo dõi trạng thái kết nối (TCP state, session table).
    - Chỉ cho phép gói tin khớp với trạng thái hợp lệ của kết nối.
    - Bảo mật hơn packet filter vì chặn được nhiều tấn công giả mạo.
3. **Application Proxy Firewall** (Application-Level Gateway)
    - Đóng vai trung gian cho giao thức ứng dụng (HTTP, FTP…).
    - Người dùng kết nối đến proxy, proxy kết nối đến server thật.
    - Có thể phân tích nội dung ứng dụng, phát hiện malware/tấn công.

    - Nhược điểm: Tốn tài nguyên, chậm hơn.
4. **Circuit-Level Proxy Firewall**
    - Tương tự Application Proxy nhưng ở mức TCP session.
    - Tạo 2 kết nối TCP riêng: giữa client – proxy và proxy – server.
    - Ví dụ: **SOCKS proxy**.
    - Không phân tích nội dung, chỉ đảm bảo kênh kết nối an toàn.

5. Các hình thức triển khai (Firewall Basing)

- **Bastion Host:** Máy chủ được cấu hình bảo mật cao, là điểm duy nhất giao tiếp giữa mạng trong và ngoài.
- **Host-Based Firewall:** Phần mềm firewall trên từng máy (Windows Defender Firewall, iptables...).
- **Personal Firewall:** Bảo vệ máy cá nhân khỏi lưu lượng không mong muốn từ mạng.

6. Vị trí & Cấu hình Firewall

- Có thể đặt ở nhiều vị trí tùy mô hình:
    - Giữa mạng LAN và Internet.
    - Giữa các vùng mạng nội bộ (Internal segmentation firewall).
    - Trong **DMZ** để bảo vệ các server công khai.
- Các cấu hình phổ biến:
    - **Single Bastion Inline**
    - **Single Bastion T**
    - **Double Bastion Inline**
    - **Double Bastion T**

7. Mở rộng thêm – Liên quan thực tế

- **Next-Generation Firewall (NGFW):**
    - Kết hợp khả năng lọc gói, kiểm tra trạng thái, phân tích ứng dụng, phát hiện xâm nhập (IPS), và sandboxing.
    - Hỗ trợ SSL/TLS inspection.
- **IPS (Intrusion Prevention System):**
    - Thường tích hợp vào firewall hoặc triển khai riêng.
    - Phát hiện và ngăn chặn tấn công dựa trên signature hoặc phân tích hành vi.
- **Zero Trust Firewall:**
    - Không mặc định tin tưởng bất kỳ kết nối nào, tất cả đều cần xác thực.

Intrusion Detection

1. Phân loại kẻ tấn công (Intruder)

- **Apprentice:** Trình độ thấp, thường dùng công cụ có sẵn.
- **Journeyman:** Có kỹ năng trung bình, hiểu rõ công cụ và có thể tùy biến.
- **Master:** Trình độ cao, tự viết hoặc tùy chỉnh kỹ thuật tấn công.

2. Hành vi kẻ tấn công (Intruder Behavior)

1. **Target Acquisition & Information Gathering**
    - Thu thập thông tin kỹ thuật và phi kỹ thuật.
    - Công cụ: `dig`, `host`, WHOIS, Nmap.
    - Ví dụ: Xem cấu trúc tổ chức trên website, dò dịch vụ mở.
2. **Initial Access**
    - Khai thác lỗ hổng mạng từ xa, brute force mật khẩu, spear-phishing.
    - Ví dụ: Lợi dụng plugin CMS lỗi để chiếm quyền.
3. **Privilege Escalation**
    - Lợi dụng lỗ hổng local để lên quyền admin.
    - Cài sniffer, thu thập mật khẩu.
4. **Information Gathering / System Exploit**
    - Tìm kiếm và trích xuất dữ liệu quan trọng.
    - Lateral movement sang hệ thống khác.
5. **Maintaining Access**
    - Cài backdoor, rootkit, thêm tài khoản ẩn.
    - Tắt hoặc chỉnh sửa antivirus, IDS.
6. **Covering Tracks**
    - Xóa log, dùng rootkit ẩn file, che dấu hành vi.

3. Khái niệm Intrusion Detection System (IDS)

- **Định nghĩa:** Hệ thống giám sát và phân tích sự kiện để phát hiện truy cập trái phép, cảnh báo theo thời gian thực hoặc gần thực.
- **Thành phần:**
    - **Sensors:** Thu thập dữ liệu.
    - **Analyzers:** Phân tích để xác định có xâm nhập.
    - **User Interface:** Hiển thị kết quả và điều khiển hệ thống.

4. Phân loại IDS

- **Host-based IDS (HIDS):** Giám sát hoạt động trên một host (system calls, file integrity, registry...).
- **Network-based IDS (NIDS):** Giám sát lưu lượng mạng tại một hoặc nhiều điểm.
- **Distributed/Hybrid IDS:** Kết hợp nhiều sensor host + network, phân tích tập trung.

5. Phương pháp phân tích

1. **Anomaly Detection**
    - So sánh hành vi hiện tại với hành vi bình thường đã học.
    - Công nghệ: thống kê, luật, machine learning (Bayesian, Markov, Neural network, Fuzzy logic, Genetic algorithms...).
2. **Signature/Heuristic Detection**
    - Dựa trên mẫu tấn công đã biết hoặc luật xác định.
    - Chỉ phát hiện được tấn công đã biết.

6. HIDS

- Monitors activity on the system in a variety of ways to detect suspicious behavior
- **Nguồn dữ liệu:**
    - System call traces.
    - Audit log.
    - File integrity checksums.
    - Registry access.
- **Loại HIDS:**
    - Anomaly HIDS.
    - Signature/Heuristic HIDS.
    - Distributed HIDS.

7. NIDS

- Giám sát lưu lượng mạng **packet-by-packet** theo thời gian thực.
- **Loại sensor:**
    - **Inline:** Nằm trên đường truyền, mọi lưu lượng đi qua.
    - **Passive:** Chỉ nghe bản sao dữ liệu (SPAN port, network tap).

8. Kỹ thuật phát hiện

- **Signature Detection:**
    - Phát hiện tấn công ở application, transport, network layer.
    - Ví dụ: SYN flood, IP spoofing, dịch vụ trái phép, vi phạm chính sách.
- **Anomaly Detection:**
    - Phát hiện DoS, quét mạng, sâu mạng (worms).

9. Honeypots

- **Mục đích:** Dẫn dụ attacker, thu thập thông tin, giữ chân kẻ tấn công.
- **Loại:**
    - **Low interaction:** Giả lập dịch vụ.
    - **High interaction:** Hệ thống thật, đầy đủ OS + dịch vụ.

10. Ví dụ: Snort

- **Packet Decoder:** Phân tách header các tầng.
- **Detection Engine:** So khớp rule, quyết định xử lý.
- **Logger:** Lưu dữ liệu.
- **Alerter:** Gửi cảnh báo qua file, socket, DB.

11. SIEM (Security Information and Event Management)

- **Mục tiêu:** Phân tích real-time cảnh báo bảo mật từ thiết bị & ứng dụng.
- **Chức năng:** Log management, event correlation, alerting, dashboard, forensic.
- **Thu thập dữ liệu:**
    - **Agent-based:** Cài agent trên máy.
    - **Agentless:** Gửi log qua Syslog, Windows Event Collector.
- **Loại cảnh báo:**
    - Lặp tấn công đăng nhập.
    - Tấn công firewall.
    - Phát hiện malware.
    - Hành vi HIPS bất thường.
- **Ưu điểm:** Nâng cao nhận thức bảo mật, hỗ trợ forensic nhanh.
- **Nhược điểm:** Dữ liệu nhiều khó xử lý, triển khai lâu, phức tạp.

Mapping nội dung trong file → các tầng mạng

1. Physical Layer (Tầng vật lý)

- Truyền tín hiệu dưới dạng bit 0/1.
- Liên quan đến **sniffing, tapping, wiretapping** → kẻ tấn công có thể nghe lén trực tiếp tín hiệu vật lý.

2. Data Link Layer (Liên kết dữ liệu)

- Có nội dung về **MAC spoofing, ARP spoofing**.
- Đề cập đến việc attacker thay đổi hoặc giả mạo **địa chỉ MAC** để chiếm quyền truy cập hoặc chuyển hướng lưu lượng.
- **Switch attacks** cũng thuộc tầng này.

3. Network Layer (Tầng mạng)

- Các tấn công:
  - **IP spoofing** (giả mạo địa chỉ IP).
  - **ICMP attacks** (ping of death, smurf attack).
  - **Routing attacks** (làm sai lệch đường đi gói tin).
- Đây là tầng chính của **IP**, nơi xảy ra **packet sniffing, fragmentation attacks**.

4. Transport Layer (Tầng vận chuyển)

- Các nội dung liên quan:
  - **TCP session hijacking**.
  - **UDP flooding**.
  - **SYN Flood (DoS attack)** → lợi dụng handshake của TCP.
- Tầng này chịu trách nhiệm cho **end-to-end communication**, vì vậy attacker hay nhắm đến TCP/UDP.

5. Application Layer (Tầng ứng dụng)

- Đề cập các giao thức ứng dụng và tấn công vào chúng:
  - **DNS poisoning / DNS spoofing**.
  - **HTTP session hijacking**.
  - **Email attacks (SMTP/POP3)**.
- Các loại malware, phishing cũng nằm ở tầng này.

Tầng	Nội dung trong file liên quan
Physical	Sniffing trực tiếp, tapping
Data Link	MAC spoofing, ARP spoofing, switch attack

Network	IP spoofing, ICMP attack, routing attack, fragmentation
Transport	TCP session hijack, SYN flood, UDP flooding
Application	DNS poisoning, HTTP hijacking, SMTP/Email attacks, malware

Network security

1. Tổng quan và nền tảng giao thức mạng *(tr. 2 – 13)*

1.1 IP Addresses (tr. 4)

- **Khái niệm**: Mỗi host có một hoặc nhiều địa chỉ IP cho mỗi network interface.
- **IPv4**:
  - 32 bit (4 byte).
  - Biểu diễn dạng **dotted-decimal**: ví dụ `128.111.48.69`.
- **Phân lớp ban đầu**:
  - **Class A**: bit đầu = `0`, netid 7 bit (1–126), hostid 24 bit → 16,777,216 host/class.
  - **Class B**: bit đầu = `10`, netid 14 bit, hostid 16 bit → 65,536 host/class.
  - **Class C**: bit đầu = `110`, netid 21 bit, hostid 8 bit → 256 host/class.
  - **Class D**: bit đầu = `1110` → địa chỉ multicast.
  - **Class E**: bit đầu = `1111` → reserved/future use.

1.2 Classless Inter-Domain Routing – CIDR (tr. 5)

Lý do: Cấp phát địa chỉ theo class gây lãng phí → IPv4 public cạn kiệt (từ 2/2011).
CIDR (giới thiệu 1993):

- Cho phép đặt ranh giới **netid/hostid** tại bất kỳ bit nào giữa bit 13 và 27.
- Ghi địa chỉ theo dạng **prefix length**: `192.168.0.0/24`.

Ưu điểm:

- Cấp phát linh hoạt, tránh lãng phí địa chỉ.
- Hỗ trợ subnetting/supernetting.

Giới hạn: Số host: min 32, max 524,288.

1.3 Special Addresses (tr. 6)

Làm địa chỉ nguồn & đích:

- Loopback: `127.x.x.x` (thường là `127.0.0.1`).

Địa chỉ nguồn:

- `netid=0, hostid=0` hoặc `netid=0, hostid=XXX`: host này trên mạng này (thường dùng khi boot).

Địa chỉ đích:

- Tất cả bit = 1: broadcast cục bộ.
- `netid` + hostid tất cả bit = 1: broadcast hướng mạng (net-directed broadcast).

Địa chỉ private (RFC 1597):

- `10.0.0.0 – 10.255.255.255`
- `172.16.0.0 – 172.31.255.255`
- `192.168.0.0 – 192.168.255.255`

Loại địa chỉ	Ví dụ	Phạm vi	Mục đích chính
Loopback	127.0.0.1	127.0.0.0/8	Test nội bộ
Nguồn netid=0	0.0.0.0	0.0.0.0/8	“Máy này” khi chưa có IP
Local broadcast	255.255.255.255	/32	Gửi tới tất cả host trong mạng cục bộ
Net-directed broadcast	192.168.1.255	netid + host all 1	Broadcast tới mạng cụ thể
Private (RFC 1597)	192.168.1.0/24	10/8, 172.16/12, 192.168/16	Mạng nội bộ, không định tuyến Internet

1.4 Internet Protocol (tr. 7)

- **“Keo dính” của Internet**: kết nối các mạng khác nhau.
- **Đặc điểm**:
  - Connectionless, unreliable, best-effort.
  - Không đảm bảo delivery, integrity, ordering, non-duplication, bandwidth.
- **Điều kiện giao tiếp**:
  - Cả hai bên phải có địa chỉ IP.
- **Phụ thuộc giao thức tầng dưới**:

    - Ví dụ: Ethernet, FDDI, RS-232.

1.5 IP Header, IP Options (tr. 9 – 11)

IP Header:

- **Kích thước thường**: 20 bytes (tối đa 60 bytes nếu có options).
- **Version**: 4 bits (IPv4 hiện tại = 4).
- **Header length**: số word 32-bit trong header.
- **TOS**: 8 bits (priority 3b, QoS 4b, 1b unused).
- **Total length**: 16 bits (tối đa 65,535 bytes).
- **Identification**: 16 bits, định danh datagram.
- **Flags & Fragment offset**: dùng cho phân mảnh.
- **TTL**: số hop tối đa.
- **Protocol**: 8 bits, xác định giao thức tầng trên (TCP, UDP...).
- **Checksum**: 16 bits, kiểm tra lỗi header.
- **Source/Destination Address**: 32 bits mỗi địa chỉ.

IP Options:

- Độ dài biến đổi.
- **Ví dụ**: Record route (ghi lại các router đi qua), Timestamp, Source routing.
- Nhận diện bởi byte đầu tiên của option.

1.6 IP Encapsulation, Direct Delivery (tr. 12 – 13)

- **Encapsulation**:
    - IP datagram được gói trong frame tầng liên kết.
    - Cấu trúc: Frame header → Frame data (IP header + IP data).
- **Direct Delivery**:
    - Khi hai host cùng mạng vật lý:
        - IP datagram được gửi trực tiếp qua tầng liên kết.
        - Địa chỉ MAC nguồn/đích lấy từ địa chỉ phần cứng NIC.
    - Ví dụ: Subnet `111.10.20.x` → gửi từ `111.10.20.121` đến `111.10.20.14`.

2. Data Link layer & phân giải địa chỉ *(tr. 14 – 19)*

2.1. Ethernet (Trang 14)

- **Mục đích:** Ethernet là giao thức tầng liên kết dữ liệu (Data Link Layer) phổ biến nhất hiện nay, dùng để truyền frame qua mạng LAN.
- **Cơ chế truy cập:** CSMA/CD (Carrier Sense Multiple Access with Collision Detection)
    - **Carrier Sense:** Lắng nghe đường truyền trước khi gửi.
    - **Multiple Access:** Nhiều thiết bị cùng dùng một môi trường truyền.
    - **Collision Detection:** Phát hiện va chạm, dừng truyền và thử lại sau.
- **Cấu trúc địa chỉ:**
    - **Địa chỉ đích:** 48 bit (6 byte), ví dụ `09:45:FA:07:22:23`

    - **Địa chỉ nguồn:** 48 bit
- **Trường Type (2 byte):**
    - `0x0800` → IPv4
    - `0x0806` → ARP
    - `0x0835` → RARP
- **Dữ liệu (Data Field):**
    - Tối thiểu: **46 byte** (nếu ít hơn sẽ thêm padding).
    - Tối đa: **1500 byte**.
- **CRC:** 4 byte cuối để kiểm tra lỗi (Cyclic Redundancy Check).

2.2. Ethernet Frame (Trang 15)

Cấu trúc:

Trường	Kích thước	Mô tả
Destination MAC	6 byte	Địa chỉ MAC đích
Source MAC	6 byte	Địa chỉ MAC nguồn
Type	2 byte	Loại payload (IP, ARP, RARP…)
Data	46–1500 byte	Payload (có thể chứa padding)
CRC	4 byte	Kiểm tra lỗi

Ví dụ:

- `0x0800` → Chứa IP datagram
- `0x0806` → Chứa ARP (28 byte)
- `0x0835` → Chứa RARP

Address Resolution Protocol – ARP (Trang 16–18)

- **Mục đích:** Chuyển đổi địa chỉ IP (tầng mạng) sang địa chỉ MAC (tầng liên kết).
- **Cách hoạt động:**
    1. **Host A** cần gửi dữ liệu đến **Host B**, biết IP B nhưng chưa biết MAC.

2. **Host A** gửi **ARP Request** dạng broadcast (`FF:FF:FF:FF:FF:FF`) hỏi "Ai có IP X.X.X.X thì cho biết MAC".
3. **Host B** trả lời **ARP Reply** unicast về A, cung cấp MAC của mình.
4. **Host A** lưu kết quả vào **ARP cache** để dùng sau.

- **Tối ưu hóa:** ARP request cũng kèm thông tin MAC và IP của người gửi để máy nhận lưu vào cache.

Cấu trúc bản tin ARP

Trường	Kích thước	Giá trị thường gặp
Hardware type (HTYPE)	2 byte	`0x0001` (Ethernet)
Protocol type (PTYPE)	2 byte	`0x0800` (IPv4)
Hardware size (HLEN)	1 byte	6
Protocol size (PLEN)	1 byte	4
Operation (OPER)	2 byte	1=Request, 2=Reply
Sender MAC	6 byte	MAC của máy gửi
Sender IP	4 byte	IP của máy gửi
Target MAC	6 byte	MAC của máy nhận (để trống trong request)
Target IP	4 byte	IP cần tìm MAC

MAC

1. Định nghĩa

- **MAC address** là **địa chỉ vật lý** gán cho mỗi thiết bị mạng (Network Interface Card – NIC) bởi nhà sản xuất.
- Dùng ở **tầng liên kết dữ liệu (Data Link Layer)** trong mô hình OSI.
- Là **duy nhất toàn cầu** (theo tiêu chuẩn IEEE) để định danh một thiết bị trong mạng LAN.

2. Đặc điểm

- **Chiều dài:** 48 bit (6 byte) đối với chuẩn phổ biến, hiển thị ở dạng 12 ký tự hex, ví dụ
  `08:00:46:07:04:A3`
  `00-1A-92-7C-4F-2B`

- **Cấu trúc:**
  - 3 byte đầu (**OUI – Organizationally Unique Identifier**): mã của hãng sản xuất card mạng (do IEEE cấp).
  - 3 byte sau: số serial duy nhất của card mạng do hãng quy định.

3. Vai trò

- **Trong mạng LAN:** MAC dùng để định tuyến gói tin ở tầng liên kết dữ liệu (Ethernet frame).
- **Địa chỉ đích MAC** quyết định thiết bị nào trong mạng sẽ nhận frame.
- **Địa chỉ nguồn MAC** giúp thiết bị nhận biết gói tin đến từ đâu.

4. Các loại địa chỉ MAC

- **Unicast MAC:** Bit đầu tiên của byte đầu tiên = 0 → dành cho một thiết bị duy nhất.
- **Multicast MAC:** Bit đầu tiên = 1 → gửi cho nhiều thiết bị (nhóm).
- **Broadcast MAC:** Tất cả bit = 1 → `FF:FF:FF:FF:FF:FF` → gửi tới tất cả thiết bị trong mạng LAN.

5. Điểm cần lưu ý

- MAC nằm cố định trong phần cứng (ROM của NIC) nhưng có thể **thay đổi tạm thời** bằng phần mềm (MAC spoofing).
- Khác với **IP address**:
  - **MAC** là định danh phần cứng, không thay đổi khi di chuyển thiết bị.
  - **IP** là định danh logic, thay đổi tùy mạng kết nối.

3. Nguy cơ trong mạng LAN & Sniffing *(tr. 20 – 33)*

3.1 Network Sniffing (tr. 22 – 23)

- Định nghĩa
  - **Sniffing** là kỹ thuật nghe lén lưu lượng mạng bằng cách thu thập gói tin di chuyển qua mạng.
  - Thường dùng để:
    - Phân tích lưu lượng (network analysis).
    - Khai thác thông tin nhạy cảm (tài khoản, mật khẩu…).
- Cách hoạt động
  - Card mạng được đưa vào **promiscuous mode** → nhận tất cả frame trên mạng, không chỉ frame gửi cho nó.
  - Trên **mạng dùng hub**: Mọi gói tin được broadcast → dễ sniff toàn bộ.

    - Trên **mạng dùng switch**: Chỉ nhận frame gửi cho mình → muốn sniff toàn bộ phải dùng kỹ thuật bypass (VD: MAC flooding, ARP spoofing).
- Tại sao nguy hiểm
    - Nhiều giao thức (FTP, POP3, HTTP, IMAP) truyền thông tin **plaintext**, dễ bị thu thập.
    - Dữ liệu mã hóa cũng có thể bị phân tích metadata (địa chỉ, kích thước, thời gian…).

3.2 Sniffing Tools (tr. 24)

- **CLI Tools**:
    - `tcpdump`: Thu thập và lọc gói tin.
    - `tcpflow`: Ghép lại luồng TCP từ gói tin.
    - `tcpreplay`: Gửi lại traffic đã thu thập.
- **GUI Tools**:
    - **Wireshark**: Hiển thị trực quan, hỗ trợ phân tích nhiều giao thức, tái dựng luồng TCP.
- **Tính năng chung**:
    - Hỗ trợ filter.
    - Lưu file PCAP để phân tích sau.
    - Phân tích header nhiều tầng (Ethernet, IP, TCP/UDP…).

3.3 TCPDump, Libpcap, Packet Structure (tr. 25 – 32)

TCPDUMP

- Giới thiệu
    - Công cụ dòng lệnh, dựa trên **libpcap**, yêu cầu quyền root để bật promiscuous mode.
    - Cho phép ghi lại hoặc phân tích trực tiếp gói tin.
- Tùy chọn quan trọng
    - `-e`: Hiển thị địa chỉ MAC nguồn/đích.
    - `-n`: Không phân giải DNS.
    - `-x`: In dữ liệu packet ở dạng hex.
    - `-i`: Chỉ định interface (VD: `-i eth0`).
    - `-r file`: Đọc packet từ file.
    - `-w file`: Ghi packet vào file.
    - `-s`: Số byte capture mỗi packet (65535 để lấy full packet).
- Bộ lọc (Filter Expression)
    - **Qualifier**:
        - `host`, `net`, `port`, `src`, `dst`, `proto` (ether, ip, arp, tcp…)
    - **Toán tử**:
        - `and`, `or`, `not`
        - So sánh: `<`, `>`, `=`, `!=`
    - **Truy cập byte trong packet**:
        - Cú pháp: `proto[offset:size]`
        - VD: `ip[0] & 0xf != 5` → lọc IP datagram có options.

Libpcap

Thư viện chuẩn để viết chương trình sniff packet.
Hàm quan trọng:

- `pcap_lookupdev()`: Tìm device bắt gói tin.
- `pcap_open_live()`: Mở interface để bắt gói.
- `pcap_open_offline()`: Mở file PCAP.
- `pcap_compile()` + `pcap_setfilter()`: Biên dịch và áp dụng filter giống tcpdump.
- `pcap_loop()`: Lặp qua các packet và xử lý bằng callback.

Packet Structure:

```
struct pcap_pkthdr {
    struct timeval ts;  // Timestamp
    bpf_u_int32 caplen; // Số byte capture
    bpf_u_int32 len;    // Độ dài thực tế của packet
};
```

3.4 Switched Environments (tr. 33)

Kỹ thuật sniff trên switch:

1. **MAC Flooding**:
    - Gửi hàng loạt frame với MAC nguồn giả.
    - Làm bảng MAC của switch đầy → chuyển sang chế độ broadcast.
2. **MAC Cloning / Duplicating**:
    - Giả mạo MAC của host mục tiêu → switch gửi frame của host đó cho attacker.

Hạn chế: Switch thông minh hoặc bảo mật tốt sẽ chặn.

4. Tấn công ARP và phòng thủ *(tr. 34 – 45)*

ARP Spoofing (tr. 34 – 42)

- **ARP Spoofing** (ARP Poisoning) là kỹ thuật gửi **gói ARP Reply giả** để đánh lừa thiết bị khác trong LAN, khiến bảng ARP cache của chúng chứa thông tin IP ↔ MAC **sai**.
- Mục tiêu: **chuyển hướng traffic** qua máy kẻ tấn công hoặc gây gián đoạn kết nối.
- Nguyên nhân
    - ARP **không xác thực** thông tin.
    - Cho phép nhận ARP Reply mà **không cần request trước** (gratuitous ARP).
    - Bảng ARP cache tự động cập nhật từ thông tin nhận được.
- Cơ chế
    1. Kẻ tấn công gửi ARP Reply giả cho **nạn nhân**:
       `"Gateway IP có MAC của tôi"`.
    2. Gửi ARP Reply giả cho **gateway**:
       `"Nạn nhân IP có MAC của tôi"`.
    3. Cả hai tin tưởng và gửi dữ liệu qua máy kẻ tấn công → **MITM**

- Hậu quả
    - **Nghe lén** (sniffing) dữ liệu.
    - **Chèn/sửa nội dung** gói tin.
    - **DoS** bằng cách chặn hoặc drop packet.
- Công cụ phổ biến
    - `arpspoof` (dsniff)
    - Ettercap, Bettercap
    - Cain & Abel (Windows)
    - Scapy (Python)

Ettercap (tr. 43) - tầng 2 và 3 của mô hình OSI.

- Dùng để **tấn công Man-in-the-Middle (MITM)** trong mạng LAN
- **Hỗ trợ ARP spoofing** để chèn mình vào giữa giao tiếp
- **Tự động forward** gói tin không gửi trực tiếp đến attacker để giữ kết nối bình thường
- **Chặn được SSH1 và SSL** (phiên bản yếu) để nghe lén nội dung
- **Thu thập mật khẩu** của nhiều giao thức khác nhau (HTTP, FTP, POP3, IMAP, Telnet, …)

ARP Defenses (tr. 44)

Nhóm giải pháp	Biện pháp cụ thể	Mô tả ngắn gọn	Ưu điểm	Nhược điểm
Cấu hình tĩnh	**Static ARP entries**	Gán cố định ánh xạ IP ↔ MAC cho thiết bị quan trọng (gateway, DNS) để bỏ qua ARP động.	Đơn giản, hiệu quả, chống giả mạo	Khó quản lý khi mạng lớn, thay đổi MAC cần cập nhật thủ công
Switch/Router	**DAI (Dynamic ARP Inspection)**, **Port Security**, **VLAN segmentation**	Kiểm tra tính hợp lệ ARP dựa trên DHCP Snooping, giới hạn MAC/port, chia VLAN giảm broadcast domain.	Ngăn chặn tấn công ngay ở tầng 2	Yêu cầu thiết bị mạng cao cấp, cấu hình phức tạp
Mã hóa	**HTTPS**, **VPN**	Mã hóa dữ liệu để ngăn đọc/sửa khi bị MITM, bảo vệ kênh truyền.	Bảo vệ nội dung, an toàn cho giao dịch	Không ngăn được ARP spoofing, chỉ bảo vệ dữ liệu
Phát hiện	**IDS/IPS**, **ARPwatch**, **XArp**	Giám sát ARP, phát hiện thay đổi ánh xạ IP/MAC, gửi cảnh báo khi nghi ngờ.	Cảnh báo sớm, hỗ trợ điều tra	Không dừng tấn công ngay, cần phản ứng thủ công

Host/OS	**ARP cache validation**, **firewall rules**	Kiểm tra tính hợp lệ ARP cache, lọc ARP từ MAC/IP không tin cậy bằng firewall.	Chủ động bảo vệ từng host	Mất công cấu hình, dễ sai sót
Nâng cao	**802.1X**, **Private VLANs**	Xác thực thiết bị trước khi truy cập mạng, cô lập cổng mạng trong VLAN riêng.	Bảo mật mạnh, hạn chế MITM	Triển khai phức tạp, cần hạ tầng hỗ trợ

5. Phát hiện Sniffer & kiểm soát truy cập mạng *(tr. 46 – 49)*

Detecting Sniffers (tr. 46 – 48)

- Suspicious ARP activity
  - ARP cache poisoning gây nhiều gói ARP bất thường → dễ bị phát hiện.
  - Dùng công cụ như **arpwatch**, **XArp** để giám sát và cảnh báo các tấn công ARP.
- Suspicious DNS lookups
  - Sniffer thường cố resolve hostname từ IP → có thể quan sát thấy truy vấn DNS lạ.
    Trap method: Gửi traffic tới một IP giả không tồn tại trong mạng, nếu thấy có DNS query cho IP này → nghi ngờ có sniffer.
- Latency analysis
  - Giả định: NIC ở chế độ promiscuous phải xử lý **mọi packet**, dẫn tới độ trễ tăng.
  - Cách làm:
    - Đo thời gian phản hồi (ping) host A.
    - Tạo lượng traffic rất lớn tới các host khác.
    - Đo lại ping host A → nếu độ trễ tăng bất thường → nghi ngờ host đang sniff.
- Linux behavior
  - Khi ở promiscuous mode, một số kernel sẽ nhận packet có:
    - **Sai địa chỉ MAC**, nhưng
    - **Đúng địa chỉ IP**
  - Kỹ thuật kiểm tra:
    - Gửi ICMP request tới host với **MAC sai** nhưng **IP đúng**.
    - Nếu host trả lời ICMP reply → có khả năng đang sniff.
- AntiSniff (viết từ năm 2000):
  - Thực hiện một số kỹ thuật trên.
  - Sử dụng TCP SYN và bắt tay TCP để kiểm tra.

Controlling Network Access – IEEE 802.1X (tr. 49)

IEEE 802.1X – Chuẩn **Port-Based Network Access Control** kiểm soát truy cập mạng dựa trên xác thực.

- **Thành phần**:
    1. **Supplicant** (client) – thiết bị muốn truy cập.
    2. **Authenticator** (switch/AP) – chặn hoặc mở cổng mạng.
    3. **Authentication Server** (RADIUS) – xác thực thông tin.
- **Quy trình**:
    1. Client kết nối → port bị khóa (chỉ cho EAPOL).
    2. Trao đổi EAP qua Authenticator đến RADIUS.
    3. Nếu xác thực thành công → port mở; thất bại → port vẫn khóa.
- **Ưu điểm**: Chặn truy cập trái phép ở tầng 2, tích hợp RADIUS, áp dụng LAN/WLAN.
- **Nhược điểm**: Yêu cầu thiết bị hỗ trợ, cấu hình phức tạp.

6. IP Spoofing & Hijacking *(tr. 50 – 66)*

IP Spoofing, mục đích, công cụ Libnet, Scapy (tr. 50 – 53)

- **Khái niệm**
    - Giả mạo địa chỉ IP nguồn trong gói tin để mạo danh thiết bị khác hoặc che giấu danh tính.
    - Hoạt động ở tầng 3 (Network Layer) của mô hình OSI.
- **Mục đích sử dụng**
    - Ẩn danh, tránh truy vết.
    - Bypass ACL/Firewall.
    - Tấn công DoS/DDoS (trực tiếp hoặc phản xạ).
    - Mạo danh nguồn tin cậy.
- **Cơ chế hoạt động**
    - Tạo gói tin với IP nguồn giả.
    - Gửi đến mục tiêu qua mạng.
    - Phản hồi sẽ gửi tới IP giả → kẻ tấn công chỉ nhận được nếu kiểm soát đường truyền hoặc thay đổi định tuyến.
- **Hạn chế**
    - Không MITM trực tiếp như ARP spoofing.
    - Khó dùng với TCP handshake (3-way handshake).
    - Dễ hơn với UDP, ICMP.
- **Ứng dụng thường gặp**
    - Smurf Attack: ICMP broadcast → trả lời về nạn nhân.
    - DNS Amplification: DNS query giả IP → server trả lời dữ liệu lớn tới nạn nhân.
    - SYN Flood: gửi TCP SYN giả → làm cạn tài nguyên server.
- **Phòng chống**
    - Ingress/Egress Filtering (RFC 2827, BCP 38).
    - Packet Filtering.
    - Authentication mạnh (TLS, IPsec).
    - IDS/IPS phát hiện lưu lượng bất thường.
- **So sánh nhanh với ARP Spoofing**
    - IP spoofing: tầng 3, khó MITM, phạm vi LAN & Internet, giả IP nguồn.
    - ARP spoofing: tầng 2, dễ MITM, phạm vi LAN, giả MAC tương ứng với IP.

Routing, Source Routing Attacks (tr. 54 – 58)

Types of Routing

1. **Hop-by-hop routing**
    - Đường đi được quyết định dần ở mỗi gateway tham gia quá trình chuyển tiếp.
2. **Source routing**
    - Host gửi datagram **tự xác định sẵn đường đi** trước khi gửi, dùng tùy chọn *IP source routing option*.

Attacks Using Source Routing

- **IP Source Routing option** cho phép chỉ định đường đi cụ thể, bỏ qua cơ chế định tuyến thông thường.
- Kẻ tấn công có thể:
    - Ép traffic đi qua **thiết bị mình kiểm soát** → sniffing hoặc **Man-in-the-Middle (MITM)**.
    - Nếu dùng reverse route cho phản hồi, có thể **mạo danh host khác** có quan hệ tin cậy với host đích.
- Hầu hết router hiện đại **vô hiệu hóa** tùy chọn này vì lý do bảo mật.

Routing Table & Mechanism (tr. 58-59)

Khi nhận một datagram, hệ thống thực hiện:

1. **Tìm khớp địa chỉ host** (đích chính xác).
2. Nếu không có → **tìm khớp địa chỉ mạng**.
3. Nếu vẫn không có → **dùng route mặc định** (default route).
4. Nếu không tìm thấy tuyến phù hợp → trả lại lỗi *host unreachable* hoặc *network unreachable* (bởi kernel hoặc gateway qua ICMP).

Cách thiết lập routing table:

- **Tĩnh**: cấu hình khi khởi động hoặc qua lệnh `route`.
- **Động**: thông qua giao thức định tuyến (RIP, OSPF, BGP).

7. Phân mảnh IP và tấn công (tr 62-69)

Fragmentation cơ chế, ví dụ (tr 62-65)

Cơ chế phân mảnh IP

- Khi IP datagram lớn hơn **MTU (Maximum Transmission Unit)** của liên kết → cần phân mảnh (fragmentation).
- Mỗi **fragment** có:
    - **Identification**: giống nhau giữa các mảnh → dùng để tái lắp ghép.
    - **Fragment Offset**: vị trí của mảnh trong datagram gốc (tính theo đơn vị 8 byte).
    - **More Fragments (MF) flag**: đặt 1 nếu còn mảnh sau; mảnh cuối MF=0.
- **Reassembly** (tái lắp ghép) chỉ xảy ra ở đích, không diễn ra tại các router trung gian.

Ví dụ phân mảnh

- Datagram gốc: 4000 byte, MTU: 1500 byte → dữ liệu tối đa mỗi mảnh: 1480 byte (20 byte header IP).
- Số mảnh:
    1. Mảnh 1: Offset 0, Length 1480, MF=1
    2. Mảnh 2: Offset 1480, Length 1480, MF=1
    3. Mảnh 3: Offset 2960, Length 1040, MF=0

Ping of Death (tr 66-67)

- **Mục tiêu**: Gửi một gói **ICMP Echo Request** (ping) có kích thước > 65,535 byte (vượt giới hạn IP).
- Cách làm:
    - Gửi nhiều mảnh nhỏ (fragments) → khi tái lắp ghép tại đích, kích thước vượt quá giới hạn.
    - Hệ thống cũ (Windows 95, Linux 2.x,…) bị tràn bộ đệm (**buffer overflow**) → crash, reboot.
- **Hiện trạng**:
    - Các hệ điều hành hiện đại đã vá lỗi, bỏ qua hoặc drop gói fragment bất hợp lệ.

Fragmentation Evasion & DoS (tr 68 - 69)

- Fragmentation Evasion
    - Kẻ tấn công cố tình phân mảnh payload để tránh bị IDS/IPS phát hiện:
        - Payload độc hại được chia thành nhiều mảnh nhỏ, IDS khó ghép lại chính xác.
        - Dùng **overlapping fragments** (mảnh chồng lấn) → IDS tái lắp ghép khác cách host đích làm.
    - Kỹ thuật thường dùng trong **IDS evasion**.
- Fragmentation DoS
    - Gửi rất nhiều mảnh IP **nhưng không gửi đầy đủ** để tái lắp ghép:
        - Host phải giữ các mảnh trong bộ nhớ chờ mảnh còn lại.
        - Làm cạn tài nguyên bộ nhớ → DoS.
    - Kỹ thuật này gọi là **Teardrop Attack** nếu dùng các offset chồng lấn để gây lỗi tái lắp ghép.

8. ICMP và tấn công *(tr. 70 – 88)*

ICMP tổng quan, thông điệp (tr. 70 – 73)

- Khái niệm
    - **ICMP** là giao thức thuộc tầng mạng, dùng để **trao đổi các thông điệp điều khiển hoặc thông báo lỗi** về quá trình chuyển giao gói tin IP.
    - ICMP **không** dùng để truyền dữ liệu ứng dụng, mà để gửi thông tin phản hồi giúp chẩn đoán hoặc điều chỉnh đường truyền.
    - Các thông điệp ICMP được **đóng gói bên trong IP datagram**.
- Phân loại
    - Request / Reply
        - **Echo Request / Echo Reply**: kiểm tra khả năng kết nối (lệnh `ping`).

        - **Address Mask Request/Reply**: máy trạm không đĩa lấy subnet mask khi khởi động.
        - **Timestamp Request/Reply**: đồng bộ thời gian giữa các hệ thống.
    - Error message
        - **Destination Unreachable**: đích không thể liên lạc được (mạng, host, port, protocol...).
        - **Time Exceeded**: TTL giảm về 0 hoặc quá thời gian ghép mảnh IP.
        - **Redirect**: thông báo cho host biết đường đi tốt hơn tới đích.
        - **Source Quench**: báo tắc nghẽn, yêu cầu gửi chậm lại (giờ hầu như không dùng).
        - **Parameter Problem**: báo lỗi trường thông tin trong header IP.
- Cấu trúc
    - **Type**: xác định loại thông điệp (ví dụ: 0 = Echo Reply, 8 = Echo Request).
    - **Code**: chi tiết phụ của loại thông điệp.
    - **Checksum**: kiểm tra lỗi.
- Công cụ sử dụng ICMP:
    - **Ping**: gửi Echo Request, nhận Echo Reply để đo độ trễ và kiểm tra kết nối.
    - **Traceroute**: gửi gói tin với TTL tăng dần, nhận ICMP Time Exceeded từ các router trung gian để xác định đường đi.

ICMP Echo Attacks, Smurf (tr.76– 77)

ICMP Echo Scan / Ping Sweep

1. Cơ chế hoạt động

- Gửi hàng loạt gói **ICMP Echo Request** (Type 8) tới một dải IP trong mạng.
- Máy chủ hoặc thiết bị nào **đang bật** và cho phép ICMP sẽ trả về **ICMP Echo Reply** (Type 0).
- Kẻ tấn công dựa vào phản hồi này để **liệt kê danh sách host đang hoạt động**.

2. Mục tiêu

- Thu thập thông tin ban đầu trong giai đoạn **reconnaissance** (trinh sát).
- Tìm ra các IP **sống** để tiếp tục quét dịch vụ, khai thác lỗ hổng.

3. Kịch bản tấn công

Hacker chạy lệnh: nmap -sn 192.168.1.0/24

1. Nmap gửi ICMP Echo Request tới toàn bộ IP từ 192.168.1.1 đến 192.168.1.254.
2. IP nào trả lời Echo Reply → đánh dấu “host up”.

4. Phòng chống

- Chặn hoặc giới hạn ICMP Echo tại tường lửa.
- Dùng IDS/IPS (Snort, Suricata) để phát hiện ping sweep.
- Giới hạn ICMP rate-limit trên router.

Smurf Attack

1. Cơ chế hoạt động

- Kẻ tấn công gửi **ICMP Echo Request** tới **địa chỉ broadcast** (ví dụ: 192.168.1.255) của một mạng đích.
- **IP nguồn bị giả mạo** thành địa chỉ của nạn nhân.
- Mọi máy trong mạng nhận được Echo Request sẽ trả Echo Reply **về nạn nhân**.
- Nếu mạng có nhiều máy, lưu lượng trả về sẽ **khuếch đại** gấp nhiều lần so với lưu lượng gửi đi.

2. Mục tiêu

- Gây **tắc nghẽn băng thông** của nạn nhân.
- Làm nạn nhân bị **DoS** (không thể sử dụng dịch vụ mạng).

3. Kịch bản tấn công

1. Hacker gửi 1 gói Echo Request 100 bytes tới broadcast của mạng có 200 máy.
2. Mỗi máy trả lại 100 bytes cho nạn nhân → nạn nhân nhận **200 × 100 bytes** cùng lúc.
3. Lặp lại liên tục → mạng của nạn nhân bị quá tải.

4. Phòng chống

- Tắt **IP-directed broadcast** trên router (`no ip directed-broadcast`).
- Dùng ACL chặn ICMP broadcast.
- Giới hạn ICMP rate-limit.

ICMP Redirect & tấn công (tr. 78 – 83)

ICMP Redirect Attack

1. Cơ chế hoạt động

- ICMP Redirect (Type 5) được router hợp lệ gửi để báo cho host biết có **đường đi tốt hơn** tới đích.
- Kẻ tấn công giả mạo router, gửi ICMP Redirect tới host, chỉ định **gateway giả** (chính là máy của hacker).
- Host cập nhật bảng định tuyến và gửi lưu lượng qua hacker.
- Hacker thực hiện **Man-in-the-Middle (MITM)** hoặc chặn luôn lưu lượng (**DoS**).

2. Mục tiêu

- Đánh cắp dữ liệu (MITM).
- Chuyển hướng kết nối qua máy tấn công.
- Ngắt kết nối của host tới dịch vụ hợp lệ.

3. Kịch bản tấn công

1. Host muốn truy cập 10.0.0.5 → gửi gói qua router 10.0.0.1.
2. Hacker gửi ICMP Redirect giả, nói rằng "10.0.0.5 tốt nhất đi qua 10.0.0.99".
3. Host tin và cập nhật route → gửi toàn bộ lưu lượng qua 10.0.0.99 (máy của hacker).

4. Phòng chống

- Tắt chấp nhận ICMP Redirect trừ khi thật sự cần (`net.ipv4.conf.all.accept_redirects = 0`).
- Dùng static route thay vì dynamic khi cần an toàn.
- IDS phát hiện gói ICMP Redirect bất thường.

Destination Unreachable & tấn công (tr. 84 – 85)

ICMP Destination Unreachable giả

1. Cơ chế hoạt động

- ICMP Destination Unreachable (Type 3) được gửi khi một router hoặc host không thể chuyển gói tin đến đích.
- Kẻ tấn công giả mạo router, gửi ICMP này tới host → host nghĩ rằng kết nối bị lỗi và ngừng gửi dữ liệu.
- Có thể dùng các code khác nhau:
    - Code 1: Host Unreachable
    - Code 3: Port Unreachable
    - Code 4: Fragmentation Needed

2. Mục tiêu

- Cắt đứt kết nối TCP/UDP.
- Gây gián đoạn dịch vụ (DoS).
- Ép ứng dụng nghĩ rằng server đã down.

3. Kịch bản tấn công

1. Client đang tải file từ server.
2. Hacker gửi liên tục ICMP Destination Unreachable → client ngắt kết nối.
3. Lặp lại với nhiều client → dịch vụ bị gián đoạn.

4. Phòng chống

- Chặn ICMP Type 3 từ nguồn không tin cậy.
- Giới hạn số lượng gói ICMP lỗi được xử lý trong một thời gian nhất định.
- IDS phát hiện ICMP lỗi bất thường.

Time Exceeded, Traceroute (tr. 86 – 88)

ICMP Time Exceeded

- Type 11, báo TTL gói IP giảm về 0 (Code 0) hoặc quá thời gian ghép mảnh IP (Code 1).
- Router/host gửi lại cho nguồn để báo lỗi.

Traceroute

- Gửi gói với TTL tăng dần (1, 2, 3…), mỗi hop hết TTL trả ICMP Time Exceeded → biết địa chỉ từng router
- Khi tới đích sẽ nhận ICMP Echo Reply hoặc Port Unreachable.
- Dùng để xác định đường đi và độ trễ, nhưng cũng bị lợi dụng để dò sơ đồ mạng.

9. UDP và tấn công *(tr. 89 – 97)*

1. UDP Tổng quan

- **User Datagram Protocol**: giao thức tầng transport, **không kết nối** (connectionless), không đảm bảo giao hàng, không kiểm soát luồng.
- **Ưu điểm**: đơn giản, độ trễ thấp, phù hợp ứng dụng real-time (VoIP, video streaming, DNS, SNMP…).
- **Nhược điểm**: không xác nhận, không đảm bảo thứ tự → dễ bị giả mạo.

2. Cấu trúc UDP Header

- **Source Port** (16 bit) – cổng nguồn.
- **Destination Port** (16 bit) – cổng đích.
- **Length** (16 bit) – độ dài header + data.
- **Checksum** (16 bit) – kiểm tra lỗi (có thể = 0).

3. Encapsulation

- UDP **đóng gói trong IP datagram**:
    - IP Header → UDP Header → Data.
- Không thiết lập kết nối trước như TCP → gói có thể đến bất kỳ lúc nào.

4. Các tấn công dựa trên UDP

a. UDP Spoofing

- Giả mạo **địa chỉ IP nguồn** trong gói UDP.
- Vì UDP không bắt tay (3-way handshake) → khó xác thực nguồn gốc.
- Mục tiêu:
    - Ẩn danh kẻ tấn công.
    - Khuếch đại lưu lượng (UDP Amplification).

Phòng chống: lọc ingress/egress (BCP 38), firewall kiểm tra tính hợp lệ.

b. UDP Hijacking

- Chiếm quyền điều khiển phiên UDP đang chạy bằng cách **chèn gói** có IP nguồn/port đúng.
- Do UDP không có cơ chế xác thực, không theo dõi trạng thái kết nối → dễ cấy dữ liệu giả.

Phòng chống: dùng mã hóa/xác thực (IPsec, DTLS), random port.

c. UDP Portscan

- Gửi UDP packet tới dải cổng đích:
    - Nếu nhận **ICMP Port Unreachable** → cổng đóng.
    - Nếu không nhận phản hồi → có thể cổng mở hoặc bị firewall chặn.
- Dùng để liệt kê dịch vụ chạy trên UDP (DNS:53, SNMP:161…).

Phòng chống: chặn ICMP trả lời, IDS phát hiện scan bất thường.

d. UDP-based DoS

- **Flood Attack**: gửi lượng lớn gói UDP tới nạn nhân → tiêu tốn tài nguyên xử lý.
- **Amplification Attack**: lợi dụng dịch vụ UDP phản hồi lớn hơn nhiều lần so với request (NTP, DNS, Memcached).
    - Ví dụ: DNS Amplification – request nhỏ (~60 bytes) → trả lời vài KB.
    - Giả mạo IP nguồn là nạn nhân để dồn lưu lượng về nạn nhân.

Phòng chống: rate-limit UDP, tắt dịch vụ không cần thiết, cấu hình chống khuếch đại.

10. TCP và tấn công *(tr. 98 – 157)*

TCP tổng quan, cấu trúc, Seq/Ack, Window, Flags (tr. 103 – 108)

Đặc điểm

- **Transmission Control Protocol**: giao thức tầng **Transport**.
- **Connection-oriented**: yêu cầu thiết lập kết nối trước khi truyền dữ liệu.
- **Reliable**: đảm bảo dữ liệu đến đích đầy đủ, đúng thứ tự.
- **Full-duplex**: truyền song song hai chiều.
- **Byte-stream**: dữ liệu được gửi như một dòng byte, TCP phân mảnh khi cần.

Cấu trúc TCP Header

Trường	Kích thước	Ý nghĩa
Source Port	16 bit	Cổng nguồn
Destination Port	16 bit	Cổng đích
Sequence Number	32 bit	Byte đầu tiên của dữ liệu trong gói
Acknowledgment Number	32 bit	Byte mong đợi tiếp theo
Data Offset	4 bit	Độ dài header
Reserved	6 bit	Dành cho tương lai
Flags	6 bit	URG, ACK, PSH, RST, SYN, FIN
Window	16 bit	Số byte còn có thể nhận

Checksum	16 bit	Kiểm tra lỗi
Urgent Pointer	16 bit	Chỉ vị trí dữ liệu khẩn
Options + Padding	biến đổi	Các tuỳ chọn (MSS, timestamp, SACK…)

Flags:

- **SYN**: bắt đầu kết nối.
- **ACK**: xác nhận dữ liệu hoặc điều khiển.
- **FIN**: kết thúc kết nối một chiều.
- **RST**: reset kết nối.
- **PSH**: đẩy dữ liệu lên ứng dụng ngay.
- **URG**: đánh dấu dữ liệu khẩn cấp.

Virtual Circuit: Setup, Handshake, Data Exchange, Shutdown (tr. 109 – 118)

Thiết lập kết nối – 3-Way Handshake

1. **SYN**: Client → Server, Seq = x.
2. **SYN+ACK**: Server → Client, Seq = y, Ack = x+1.
3. **ACK**: Client → Server, Ack = y+1.

→ Sau bước này, cả hai đã đồng bộ Seq/Ack và kết nối được mở.

Trao đổi dữ liệu

- TCP dùng **Sliding Window** để điều khiển luồng.
- Mỗi gói dữ liệu kèm Seq và Ack để đảm bảo thứ tự.
- Nếu gói bị mất → Retransmission.

Đóng kết nối – 4 bước

1. Bên A gửi **FIN** (Seq = u) → yêu cầu đóng.
2. Bên B gửi **ACK** (Ack = u+1).
3. Bên B gửi **FIN** (Seq = v).
4. Bên A gửi **ACK** (Ack = v+1) → kết thúc.

Lưu ý: TCP có trạng thái **TIME_WAIT** để tránh xung đột Seq khi kết nối cũ chưa hết vòng đời.

Port scanning: connect, SYN, FIN, Idle (tr. 119 – 130)

a. Connect Scan

- Hoàn tất handshake TCP để xác định cổng mở.
- **Ưu**: chính xác, dễ thực hiện.
- **Nhược**: bị ghi log rõ ràng.

b. SYN Scan (Half-open)

- Gửi SYN → nhận SYN+ACK (mở) hoặc RST (đóng).
- Không gửi ACK để hoàn tất handshake → giảm log.
- Công cụ: `nmap -sS`.

c. FIN Scan

- Gửi FIN tới cổng:
  - Nếu nhận RST → cổng đóng.
  - Nếu không phản hồi → có thể cổng mở (trên hệ thống không tuân RFC 793).

d. Idle Scan

- Dùng một host thứ ba (zombie) để gửi gói, dựa vào **IP ID field** để suy ra trạng thái cổng → ẩn hoàn toàn IP kẻ tấn công.

OS Fingerprinting (tr. 131)

Quan sát phản ứng của host với gói TCP/IP bất thường:

TTL mặc định (Time-To-Live)

- Ví dụ: Windows thường TTL = 128, Linux TTL = 64, Cisco = 255.
- Cho thấy host thuộc họ nào.

Window Size mặc định (TCP Window Size)

- Giá trị mặc định khác nhau giữa OS (Linux ~5840, Windows 8192/65535, v.v).
- Gợi ý phiên bản / kernel.

Cách xử lý TCP Flags lạ

- Gửi gói với flag không hợp lệ (FIN+URG+PSH hoặc NULL).
- Một số OS trả RST, một số im lặng → đặc trưng riêng.

MSS (Maximum Segment Size) & Timestamp

- MSS mặc định, TSopt (TCP Timestamps) cũng khác nhau theo OS.
- Ví dụ: Linux kernel mới có timestamp đặc trưng.

Công cụ: `nmap -O`.

TCP Spoofing (tr. 132 – 135)

- **Giả mạo IP nguồn** và **dự đoán số Sequence** để thiết lập kết nối giả.
- Khó hơn UDP spoofing vì phải đồng bộ Seq/Ack.
- Kịch bản:
  1. Chặn hoặc đoán gói từ server.
  2. Dùng IP giả mạo để gửi SYN.

3. Dự đoán SYN+ACK Seq → gửi ACK giả để hoàn tất kết nối.

Phòng chống: dùng xác thực ở tầng ứng dụng, lọc ingress/egress.

TCP Hijacking, ACK Storm, Variants (tr. 141 – 147)

TCP Hijacking – tổng quan

- **Ý tưởng:** Kẻ tấn công chen vào một phiên TCP **đang tồn tại**, bằng cách **chèn gói dữ liệu** với **Sequence number (Seq)** và **Acknowledgment number (Ack)** hợp lệ.
- **Mục tiêu:**
  - **Tiêm lệnh** (command injection) → chiếm quyền điều khiển terminal/session (ví dụ telnet, rsh cũ).
  - **Chiếm quyền** của user mà không cần re-authenticate.

Điều kiện để hijack

1. Biết hoặc đoán được **IP source & destination**.
2. Biết **port** đang dùng.
3. Biết hoặc đoán được **Sequence number** hợp lệ trong phiên.

Các biến thể chính

1. Blind Hijacking

- **Đặc điểm:** Attacker **không sniff được traffic** (không thấy gói tin thực sự).
- **Cách làm:** Dự đoán (guess) sequence number → chèn lệnh vào phiên.
- **Khó khăn:**
  - Rất khó để chèn được đúng Seq, vì TCP Seq tăng dần theo bytes gửi.
  - Nếu đoán sai → gây **ACK Storm** (dưới).
- **Ứng dụng:** Dùng để tiêm command một chiều (one-shot), ví dụ gửi lệnh `echo hacker >> /etc/passwd`.

2. Non-blind Hijacking

- **Đặc điểm:** Attacker **có khả năng sniff traffic** (ví dụ trong LAN với ARP spoofing).
- **Cách làm:**
  - Quan sát Seq/Ack từ gói thật.
  - Gửi gói giả mạo với Seq/Ack hợp lệ → chen ngang, cướp session.
- **Hiệu quả:** Chính xác hơn, dễ thành công, attacker có thể **toàn quyền điều khiển session**.

Hiện tượng ACK Storm

- **Xảy ra khi:** Một gói chèn vào **"out of sync"** (sai sequence number).
- **Kết quả:**
  - Cả client và server thấy số Seq/Ack bất thường.
  - Mỗi bên gửi **ACK** để cố gắng đồng bộ lại.
  - Vòng lặp liên tục → **bão ACK** → tiêu tốn băng thông, CPU.
  - Có thể dẫn đến **Denial of Service (DoS)**

SYN flooding & SYN Cookies (tr. 148 – 152)

SYN Flooding:

- Gửi hàng loạt SYN nhưng không hoàn tất handshake → server giữ nhiều kết nối half-open.
- Cạn kiệt bảng kết nối → từ chối dịch vụ hợp lệ.

SYN Cookies:

- Không lưu state cho đến khi nhận ACK hợp lệ.
- Mã hóa thông tin kết nối vào Sequence Number trong SYN+ACK.

State Attacks (tr. 157)

Lợi dụng việc TCP duy trì **state** của kết nối:

- Mở nhiều kết nối idle.
- Giữ kết nối lâu (Slowloris style).

Mục tiêu: tiêu tốn bộ nhớ, CPU, slot kết nối.

11. IPv6 và IPSec *(tr. 153 – 167)*

IPv6 tổng quan, header (tr. 158 – 159)

IPv6 Tổng quan (tr. 158–159)

- Đặc điểm
  - Địa chỉ dài **128 bit** → ~3.4 × 10^{38} địa chỉ.
  - Biểu diễn dạng hex, ngăn cách bằng dấu ":" (vd: `2001:0db8::1`).
  - Hỗ trợ **autoconfiguration** (SLAAC), **multicast** tích hợp.
  - Loại bỏ broadcast → thay bằng multicast và anycast.
  - Tích hợp **IPSec** bắt buộc hỗ trợ.
- Ưu điểm so với IPv4
  - Không cần NAT.
  - Header đơn giản, cố định 40 byte → xử lý nhanh hơn.
  - Tích hợp bảo mật và QoS tốt hơn.
  - Không phụ thuộc ARP, dùng **Neighbor Discovery Protocol (NDP)**.

IPv6 Header

- **Version**: 6.
- **Traffic Class (TC)**: QoS.
- **Flow Label**: định danh luồng.
- **Payload Length**: độ dài dữ liệu sau header.
- **Next Header**: giao thức tiếp theo (TCP, UDP, hoặc header mở rộng).
- **Hop Limit**: tương tự TTL.
- **Source/Destination Address**: 128 bit.

- Không phụ thuộc ARP, dùng **Neighbor Discovery Protocol (NDP)**.

IPSec: khái niệm, AH, ESP, Modes, IKE (tr. 160 – 166)

Khái niệm

- Bộ giao thức bảo mật tầng mạng, bảo vệ dữ liệu IP.
- Hỗ trợ **bảo mật IP cả IPv4 và IPv6**.
- Dịch vụ:
    - **Xác thực** (Authentication)
    - **Toàn vẹn** (Integrity)
    - **Bí mật** (Confidentiality)
    - **Chống phát lại** (Anti-replay)

3.0 Security Association (SA)

- **Định nghĩa:** Một tập hợp các tham số bảo mật được hai bên thống nhất trước khi truyền thông.
- **Nội dung SA bao gồm:**
    - Thuật toán dùng để **xác thực**.
    - Thuật toán dùng để **mã hóa bulk encryption**.
    - **Thời gian hiệu lực (validity)** của SA.
- **Thực thi:**
    - Mỗi gói IP mang một tham chiếu tới SA liên quan thông qua **Security Parameter Index (SPI)**.
    - Giúp thiết bị nhận biết phải áp dụng thuật toán bảo mật nào cho gói đó

3.1 Authentication Header (AH)

- Bảo vệ **toàn vẹn và xác thực** gói tin (IP header + payload).
- Không mã hóa dữ liệu
- Cung cấp **anti-replay** bằng Sequence Number.

Cấu trúc cơ bản:

Next Header | Payload Len | Reserved | SPI | Seq Number | Auth Data

3.2 Encapsulating Security Payload (ESP)

- Bảo mật nội dung bằng **mã hóa** + xác thực tùy chọn.
- Bảo vệ payload, không bảo vệ toàn bộ IP header.
- Hỗ trợ **confidentiality** và **integrity**.

Cấu trúc cơ bản:

SPI | Seq Number | Payload Data | Padding | Pad Len | Next Header | Auth Data

3.3 Modes

- **Transport Mode**:

    - Chỉ bảo vệ payload.
    - Dùng end-to-end giữa hai host.
- **Tunnel Mode**:
    - Bao gói toàn bộ gói IP gốc bên trong gói IP mới.
    - Dùng giữa các gateway hoặc VPN.

3.4 Internet Key Exchange (IKE)

- Giao thức thương lượng và thiết lập Security Association (SA).
- **Phase 1**: thiết lập kênh bảo mật (ISAKMP SA).
- **Phase 2**: thương lượng các SA cho AH/ESP.
- Dùng thuật toán Diffie–Hellman, xác thực bằng PSK, chữ ký số hoặc chứng chỉ.

IPv6 Security Risks (tr. 167)

- **Chuyển tiếp tự động** (auto tunneling) → có thể bị lạm dụng để vượt tường lửa IPv4.
- **Neighbor Discovery (NDP) spoofing**: tương tự ARP spoofing nhưng cho IPv6.
- **Extension Headers abuse**: lợi dụng chuỗi header mở rộng để né IDS/IPS.
- **Dual-stack attacks**: khai thác khi hệ thống chạy cả IPv4 và IPv6.
- Thiếu cấu hình bảo mật → nhiều dịch vụ IPv6 mở mà quản trị viên không biết.

12. Mạng không dây 802.11 và cơ chế bảo mật *(tr. 168 – 231)*

Wireless Networks tổng quan (tr. 168 – 171)

- Đặc điểm
    - Chuẩn **IEEE 802.11** cho mạng WLAN (Wi-Fi).
    - Hoạt động ở **tầng vật lý** và **tầng MAC** của mô hình OSI.
    - Sử dụng sóng vô tuyến (2.4 GHz, 5 GHz, 6 GHz) → linh hoạt nhưng dễ bị nghe lén.
- Thành phần
    - **Access Point (AP)**: kết nối WLAN với mạng có dây.
    - **Station (STA)**: thiết bị client (laptop, smartphone…).
    - **BSS (Basic Service Set)**: 1 AP + các STA.
    - **ESS (Extended Service Set)**: nhiều BSS liên kết.
- Ưu/nhược điểm
    - **Ưu**: tiện lợi, triển khai nhanh, hỗ trợ roaming.
    - **Nhược**: vùng phủ sóng giới hạn, nhiễu, và dễ bị tấn công qua sóng.

802.11 Data Link Layer (tr. 172 – 176)

- **LLC (Logical Link Control):** dùng chung với LAN 802 khác (Ethernet). Địa chỉ 48-bit.
- **MAC (Media Access Control):**
    - Ethernet: **CSMA/CD** (Carrier Sense + Collision Detection).
    - WLAN: **CSMA/CA** (Collision Avoidance) → do không thể “nghe va chạm” trên sóng.

        - Vấn đề near/far: tín hiệu mạnh có thể lấn át tín hiệu yếu → không phát hiện được collision.

CSMA/CA Cơ chế:

1. STA chờ kênh rảnh.
2. Đợi thêm thời gian random (backoff).
3. Gửi gói tin.
4. Nhận **ACK** từ phía nhận.
5. Nếu không có ACK → coi như mất gói → gửi lại sau random backoff.

- **Tăng độ tin cậy:**
    - **RTS/CTS (Ready To Send / Clear To Send).**
    - **CRC checksums.**

Modes of Operation

- **Infrastructure mode:**
    - AP đóng vai trò trung gian, STA kết nối qua AP.
- **Ad-hoc mode:**
    - STA kết nối trực tiếp, không cần AP.
    - Nếu muốn gửi dữ liệu ngoài vùng phủ sóng → một STA khác phải relay.

Loại frame, Association, Discovery (tr. 172 – 176)

- Loại frame 802.11
    1. **Management frames**: quản lý kết nối (Beacon, Probe, Authentication, Association…).
    2. **Control frames**: điều khiển truy cập (RTS, CTS, ACK).
    3. **Data frames**: mang dữ liệu người dùng.
- Discovery
    - **Passive**: AP phát Beacon định kỳ, STA quét và tìm AP.
    - **Active**: STA gửi Probe Request, AP trả Probe Response.
- Association
    - **Authentication**: Xác thực ban đầu (open system hoặc shared key).
    - **Association**: STA gửi yêu cầu kết nối, AP chấp nhận và cấp tham số kết nối.

WEP (Wired Equivalent Privacy) (tr. 177 – 197)

- **Cơ chế**
    - Dùng RC4 để mã hóa.
    - Khóa bí mật chia sẻ tĩnh + IV (24 bit).
    - CRC-32 để kiểm tra toàn vẹn.
- **Nhược điểm**
    - IV quá ngắn (24 bit) → lặp lại nhanh.
    - RC4 key scheduling yếu → có thể khôi phục khóa từ đủ số lượng gói.
    - CRC-32 không chống được thay đổi dữ liệu có chủ ý.
    - Khóa tĩnh → đổi khóa thủ công.
- **Tấn công**
    - **Passive cracking**: thu đủ gói tin có IV lặp → giải khóa bằng Aircrack-ng.

    - **Replay attack**: phát lại gói cũ để tạo nhiều gói mới.
    - **Bit-flipping**: thay đổi dữ liệu nhưng vẫn giữ CRC đúng.
- **Phòng chống**
    - Ngưng dùng WEP, thay bằng WPA2/WPA3.
    - Chặn client dùng WEP qua cấu hình AP.

RSN, WPA/WPA2, TKIP, CCMP, Key Management, EAP (tr. 198 – 222)

- RSN (Robust Security Network)
    - Khung bảo mật trong 802.11i.
    - Hỗ trợ WPA/WPA2/WPA3.
- WPA (Wi-Fi Protected Access)
    - Dùng TKIP thay WEP, mã hóa RC4 nhưng với khóa động.
    - IV dài 48 bit, MIC (Message Integrity Code) chống sửa đổi.
- WPA2
    - Sử dụng **AES-CCMP** thay TKIP.
    - **CCMP**:
        - AES trong Counter Mode để mã hóa.
        - CBC-MAC để kiểm tra toàn vẹn.
        - Khóa động từ 4-Way Handshake.
- Key Management
    - **4-Way Handshake**: tạo PTK (Pairwise Transient Key) từ PMK (Pairwise Master Key).
    - MIC bảo vệ handshake.
- EAP Methods
    - **EAP-TLS**: chứng chỉ số hai chiều.
    - **PEAP**, **EAP-TTLS**: kênh TLS, xác thực qua username/password.
    - **EAP-FAST**: dùng PAC (Protected Access Credential).

WPA2 Attacks, KRACK (tr. 224 – 228)

- **WPA2 Attacks**
    - **Handshake capture + offline dictionary attack**:
        - Thu gói handshake.
        - Thử nhiều passphrase cho đến khi tìm được PMK đúng.
    - **PMKID attack**:
        - Trích xuất PMKID từ AP để brute-force offline.
- **KRACK (Key Reinstallation Attack)**
    - Lợi dụng lỗ hổng trong **4-Way Handshake**:
        - Kẻ tấn công phát lại message 3 của handshake.
        - STA cài lại khóa nonce → reset bộ đếm, tái sử dụng khóa → giải mã gói tin.
    - Ảnh hưởng WPA2-PSK và WPA2-Enterprise.
- **Phòng chống**:
    - Vá firmware AP và driver client.
    - Dùng WPA3 hoặc triển khai bảo mật nâng cao.

\