TCP/IP 네트워크 프로토콜의 이해와 실무 적용

Understanding TCP/IP Network Protocols and Practical Applications

목차

1. 개요

현대 IT 인프라의 기반이 되는 네트워크 기술은 시스템 관리자와 보안 전문가에게 필수적인 지식입니다. 본 보고서는 TCP/IP 네트워크 교육을 통해 학습한 내용을 체계적으로 정리하고, 실무 관점에서의 이해를 심화하기 위해 작성되었습니다.

1.1 네트워크 학습의 필요성

리눅스 시스템 관리를 학습한 이후, 네트워크는 시스템 간 통신과 서비스 제공의 핵심 요소로서 반드시 이해해야 할 영역입니다. 서버가 아무리 안정적으로 구축되어 있어도 네트워크 통신이 원활하지 않으면 서비스를 제공할 수 없으며, 네트워크 계층에서 발생하는 문제는 전체 시스템의 가용성에 직접적인 영향을 미칩니다.

1.2 학습 목표 및 범위

본 보고서는 다음의 내용을 다룹니다:

- 네트워크 계층 모델(OSI, TCP/IP)의 이해
- 각 계층에서 동작하는 주요 프로토콜의 메커니즘
- 패킷 분석을 통한 프로토콜 동작 확인
- 네트워크 보안의 기본 개념
- 실무 환경에서의 적용 사례

2. 네트워크 계층 모델

2.1 계층화 모델의 필요성

네트워크 통신은 매우 복잡한 과정입니다. 이를 효율적으로 관리하고 표준화하기 위해 계층화 모델이 도입되었습니다. 계층화를 통해 각 계층은 독립적으로 동작하며, 한 계층의 변경이 다른 계층에 영향을 미치지 않습니다. 이는 네트워크 기술의 발전과 유지보수를 용이하게 만듭니다.

2.2 OSI 7 계층과 TCP/IP 4 계층 비교

OSI(Open Systems Interconnection) 모델은 이론적인 참조 모델로 7개의 계층으로 구성되어 있습니다. 반면 TCP/IP 모델은 실제 인터넷에서 사용되는 프로토콜 스택으로 4개의 계층으로 구성됩니다. 실무에서는 두 모델을 혼용하여 사용하며, OSI 모델은 네트워크 문제를 진단할 때 개념적 틀로 활용됩니다.

OSI 7계층	TCP/IP 4계층	주요 프로토콜
7. 응용 계층 (Application)	**응용 계층**	HTTP, HTTPS, DNS, FTP, SSH
6. 표현 계층 (Presentation)		SSL/TLS
5. 세션 계층 (Session)		
4. 전송 계층 (Transport)	**전송 계층**	TCP, UDP
3. 네트워크 계층 (Network)	**인터넷 계층**	IP, ICMP, ARP
2. 데이터링크 계층 (Data Link)	**네트워크 접근 계층**	Ethernet, Wi-Fi
1. 물리 계층 (Physical)		Cable, Wi-Fi

실무에서는 OSI의 응용/표현/세션 계층을 통합하여 '응용 계층'으로, 데이터링크/물리 계층을 통합하여 '네트워크 접근 계층'으로 다루는 TCP/IP 모델이 더 실용적입니다.

3. 주요 프로토콜의 동작 원리

3.1 데이터링크 계층: ARP (Address Resolution Protocol)

3.1.1 ARP 의 필요성

네트워크 통신에서 IP 주소만으로는 실제 데이터 전송이 불가능합니다. 같은 네트워크(LAN) 내에서 통신하려면 목적지의 MAC 주소를 알아야 하는데, ARP는 IP 주소를 통해 MAC 주소를 찾아주는 프로토콜입니다.

3.1.2 ARP 동작 과정

ARP 요청 (ARP Request):

- 송신자가 목적지 IP 주소를 알고 있지만 MAC 주소를 모르는 상황
- 브로드캐스트(FF:FF:FF:FF:FF:FF)로 "이 IP 주소의 MAC 주소가 뭐죠?" 질문
- 동일 네트워크의 모든 장치가 이 메시지를 받음

ARP 응답 (ARP Reply):

- 해당 IP 를 가진 장치만 유니캐스트로 응답
- "제 MAC 주소는 XX:XX:XX:XX:XX:XX 입니다" 응답
- 송신자는 이 정보를 ARP 캐시에 저장하여 재사용

실무 활용: Wireshark를 통해 ARP 패킷을 분석하면 브로드캐스트와 유니캐스트의 차이, 그리고 MAC 주소 해석 과정을 명확히 확인할 수 있습니다. 또한 arp -a 명령어로 현재 시스템의 ARP 캐시를 확인할 수 있습니다.

3.2 네트워크 계층: ICMP (Internet Control Message Protocol)

3.2.1 ICMP 의 역할

ICMP는 IP 프로토콜의 보조 역할을 하는 프로토콜로, 네트워크 상태를 진단하고 오류를 보고합니다. 데이터 전송용이 아닌 제어 및 관리용 프로토콜입니다.

3.2.2 주요 ICMP 메시지

Type	메시지	용도
Type 8	Echo Request	ping 명령 시 전송되는 메시지
Type 0	Echo Reply	ping 요청에 대한 응답
Type 3	Destination Unreachable	목적지 도달 불가 시 오류 보고

Type	메시지	용도
Type 11	Time Exceeded	TTL이 0이 되어 패킷 폐기 시 사용 (tracert)

실무 활용: ping 명령어는 네트워크 연결을 확인하는 가장 기본적인 도구이며, tracert(Windows) 또는 traceroute(Linux) 명령어는 패킷이 목적지까지 가는 경로를 확인할 때 사용합니다. ICMP Type 11 메시지를 이용해 각 라우터에서 응답을 받아 경로를 추적합니다.

3.3 전송 계층: TCP 3-Way Handshake

3.3.1 TCP 의 특징

TCP(Transmission Control Protocol)는 신뢰성 있는 데이터 전송을 보장하는 프로토콜입니다. 연결 지향적이며, 순서 보장, 흐름 제어, 혼잡 제어 등의 기능을 제공합니다. 웹 브라우징(HTTP/HTTPS), 이메일(SMTP), 파일 전송(FTP) 등 대부분의 인터넷 서비스가 TCP를 사용합니다.

3.3.2 3-Way Handshake 과정

TCP 연결을 수립하는 과정으로, 클라이언트와 서버가 서로 준비되었음을 확인합니다:

Step 1: SYN

- 클라이언트가 서버에 연결 요청 (SYN 플래그 설정)
- 초기 시퀀스 번호(ISN)를 랜덤하게 생성하여 전송
- "연결하고 싶습니다. 제 시작 번호는 X 입니다."

Step 2: SYN-ACK

- 서버가 클라이언트의 요청을 수락 (SYN + ACK 플래그)
- 자신의 ISN 도 생성하고, 클라이언트 ISN+1 을 ACK 로 전송
- "알겠습니다. 제 시작 번호는 Y 입니다."

Step 3: ACK

- 클라이언트가 서버의 응답을 확인 (ACK 플래그)
- 서버 ISN+1 을 ACK 로 전송
- "잘 받았습니다. 이제 데이터를 주고받을 수 있습니다."

이 과정을 통해 양쪽 모두 데이터 송수신 준비가 완료됩니다. Wireshark로 패킷을 캡처하면 SYN, SYN-ACK, ACK 플래그가 설정된 세 개의 패킷을 명확히 확인할 수 있습니다.

3.3.3 TCP vs UDP 비교

특성	TCP	UDP
연결 방식	연결 지향 (Connection-oriented)	비연결 (Connectionless)
신뢰성	높음 (재전송, 순서 보장)	낮음 (재전송 없음)

특성	TCP	UDP
속도	상대적으로 느림	빠름
사용 사례	HTTP, HTTPS, FTP, SSH	DNS, 스트리밍, 게임

3.4 응용 계층: HTTP/HTTPS 와 TLS

3.4.1 HTTP 프로토콜

HTTP(HyperText Transfer Protocol)는 웹에서 데이터를 주고받기 위한 프로토콜입니다. 클라이언트(브라우저)가 요청(Request)을 보내면 서버가 응답(Response)을 반환하는 구조로, 요청 메소드(GET, POST, PUT, DELETE 등)와 상태 코드(200, 404, 500 등)를 통해 통신합니다.

3.4.2 HTTPS 와 TLS

HTTPS는 HTTP에 보안 계층(TLS/SSL)을 추가한 프로토콜입니다. TLS(Transport Layer Security)는 다음을 제공합니다:

- **암호화:** 데이터를 제 3 자가 읽을 수 없도록 암호화
- **무결성:** 데이터가 전송 중 변조되지 않았음을 보장
- **인증:** 서버의 신원을 인증서를 통해 검증

3.4.3 TLS Handshake 과정

TLS 1.3 버전의 간소화된 핸드셰이크:

1. Client Hello:

- 클라이언트가 지원하는 암호화 알고리즘 목록 전송
- 랜덤 데이터와 세션 키 생성에 필요한 정보 포함

2. Server Hello + Certificate:

- 서버가 선택한 암호화 알고리즘과 인증서 전송
- 인증서에는 서버의 공개키가 포함됨

3. 암호화된 통신 시작:

- 세션 키를 생성하여 이후 모든 데이터 암호화
- 대칭키 암호화로 빠른 데이터 전송 보장

실무 활용: 웹 브라우저의 개발자 도구(F12)에서 Network 탭을 통해 HTTP 요청/응답 헤더를 확인할 수 있으며, Wireshark로 TLS 핸드셰이크 과정의 각 단계를 패킷 레벨에서 분석할 수 있습니다.

4. 네트워크 보안 기초

4.1 방화벽의 역할

방화벽은 네트워크 트래픽을 모니터링하고 제어하는 보안 시스템입니다. 미리 정의된 보안 규칙에 따라 들어오고 나가는 네트워크 트래픽을 허용하거나 차단합니다. 내부 네트워크를 외부 위협으로부터 보호하는 첫 번째 방어선 역할을 합니다.

4.2 방화벽 규칙 설정

방화벽 규칙은 다음 요소를 기반으로 트래픽을 제어합니다:

- **출발지/목적지 IP 주소:** 어디서 오고 어디로 가는 트래픽인가
- **포트 번호:** 어떤 서비스/애플리케이션을 사용하는가
- **프로토콜:** TCP 인가 UDP 인가
- **액션:** 허용(Allow), 차단(Block), 거부(Reject)

실무 예시: 웹 서버(포트 80, 443)는 외부에서 접근 가능하도록 열어두고, SSH(포트 22)는 특정 IP에서만 접근 가능하도록 제한하며, 내부 데이터베이스(포트 3306)는 외부 접근을 완전히 차단하는 식으로 규칙을 구성합니다.

4.3 IDS vs IPS

구분	IDS	IPS
명칭	Intrusion Detection System	Intrusion Prevention System
역할	침입 탐지 및 경고	침입 탐지 및 차단
동작 방식	수동적 (Passive)	능동적 (Active)
네트워크 위치	트래픽 복사본 분석	트래픽 경로상에 위치
장점	정상 트래픽에 영향 없음	실시간 위협 차단

실무에서는 IDS와 IPS를 함께 사용하는 것이 일반적입니다. IDS로 전체 트래픽을 모니터링하면서, 중요한 구간에는 IPS를 배치하여 즉각적인 대응이 가능하도록 구성합니다. Suricata와 같은 오픈소스 도구를 통해 IDS/IPS 룰을 학습하고 커스터마이징할 수 있습니다.

4.4 포트 스캔과 보안

포트 스캔은 대상 시스템에서 열려 있는 포트를 탐지하는 기술입니다. nmap 같은 도구를 사용하여 어떤 서비스가 실행 중인지 파악할 수 있습니다. 공격자는 포트 스캔을 통해 취약점을 찾아내므로, 불필요한 서비스는 중지하고 필요한 포트만 선택적으로 열어두는 것이 중요합니다.

5. 실무 적용 및 결론

5.1 네트워크 문제 진단 절차

실무에서 네트워크 문제가 발생했을 때 계층별로 접근합니다:

1단계: 물리/데이터링크 계층 확인

- 케이블이 제대로 연결되어 있는가?
- 네트워크 인터페이스가 활성화되어 있는가? (ifconfig, ip addr)

2단계: 네트워크 계층 확인

- IP 주소가 올바르게 설정되어 있는가?
- 게이트웨이와 통신이 가능한가? (ping)
- 라우팅 테이블이 올바른가? (route, ip route)

3단계: 전송 계층 확인

- 목적 포트가 열려 있는가? (netstat, ss)
- 방화벽에서 해당 포트가 차단되어 있지 않은가?

4단계: 응용 계층 확인

- 서비스가 정상적으로 실행 중인가? (systemctl status)
- 애플리케이션 로그에 오류가 있는가?

5.2 실무 도구 활용

도구	용도	주요 명령어
Wireshark	패킷 캡처 및 분석	필터: tcp.port == 80, ip.addr == 192.168.1.1
nmap	포트 스캔	nmap -sS -sV -O <target>
Packet Tracer	네트워크 시뮬레이션	라우터/스위치 설정 실습, 토폴로지 구성
Linux 명령어	네트워크 진단	ping, traceroute, netstat, ss, ip, tcpdump

5.3 결론

네트워크는 현대 IT 인프라의 혈관과 같습니다. 단순히 이론적인 지식으로 끝나는 것

이 아니라, 실제 패킷을 분석하고 프로토콜의 동작을 확인하면서 학습한 내용은 실무에서 문제 해결의 기초가 됩니다.

이번 네트워크 학습을 통해 계층 모델의 개념, 각 계층에서 동작하는 주요 프로토콜, 그리고 네트워크 보안의 기본을 이해할 수 있었습니다. 특히 Wireshark를 통한 패킷 분석 실습은 추상적인 개념을 구체적으로 이해하는 데 큰 도움이 되었습니다.

앞으로 시스템 관리자로서 네트워크 문제를 진단하고 해결할 때, 이번 학습 내용이 문제의 원인을 파악하고 적절한 해결책을 찾는 데 중요한 기반이 될 것입니다. Linux 시스템 관리와 네트워크 지식을 결합하면, 더욱 견고하고 안전한 인프라를 구축하고 운영할 수 있을 것입니다.

참고문헌

TCP/IP 네트워크 Lab 문서(교안)

TCP/IP 네트워크 보안(교안)

처음 배우는 네트워크 보안 (한빛미디어)