**정보보안기사**

**5장 네트워크 보안**

**19절 네트워크 개요(최종수정일 8/18)**

**프로토콜:** 신뢰/효율/안전하게 정보를 주고받기 위해 정보의 송/수신자측 또는 네트워크 내에서 사전에 약속된 규약 또는 규범. 송신자가 수신자를 호출해 연결하는 과정, 통신회선에서의 접속방식, 전달하는 정보의 형태, 오류 제어, 송/수신자측 간의 동기방식 등에 대한 약속.

**프로토콜의 구성 요소: 구문(Syntax) – 데이터의 구조와 형식 표현순서, 의미(Semantics) – 비트의 영역별 의미, 타이밍(Timing) – 데이터를 언제 얼마나 빨리 전송할 것인지**

**물리적/논리적 프로토콜:** 물리적 프로토콜은 데이터 전송에 사용되는 전송 매체, 접속용 커넥터 및 전송 신호등 물리적 요소에 관한 규약.(RS-232C, V.21) 논리적 프로토콜은 데이터 표현 형식 단위인 프레임의 구성, 프레임 내의 각 항목의 의미와 기능, 데이터 전송 절차 등을 말한다. SNA, SDLC 등 IBM에서 만든 전용 프로토콜(동일 회사 제품끼리 이용)과 TCP/IP 등의 범용 프로토콜로 나뉜다.

**OSI 모델과 TCP/IP 프로토콜**

**OSI 모델:** ISO 표준은 **개방 시스템 상호연결**(OSI, Open System Interconnection) 모델이다. 기반 구조에 관계없이 모든 정보 처리 시스템의 통신을 가능하게 하는 프로토콜의 집합. OSI 모델은 하드웨어나 소프트웨어 기반에 관계없이 다른 시스템 간의 통신을 원활하게 하는데 목적을 둔다.

**물/데/네: 물리, 데이터 링크, 네트워크 계층 -> 물리적인 면(전기 규격, 물리적 연결, 물리 주소, 전송시간과 신뢰도 등) 처리**

**전송 계층: 위 아래 두 그룹을 연결하고 하위 계층(물/데/네)에서 전송한 내용을 상위 계층이 사용할 수 있는 형태가 되도록 보장**

**세/표/응: 세션, 표현, 응용 계층 -> 사용자 지원 계층(관련이 없는 소프트웨어 시스템 간의 상호 연동을 가능하게 함)**

|  |  |  |  |
| --- | --- | --- | --- |
| **계층** | **특징** | **데이터 종류** | **예** |
| **응용** | 각종 응용프로그램  네트워크 관리 | 메시지 | FTP, TFTP, SNMP, SMTP,  Telnet, HTTP, DNS, DHCP |
| **표현** | 네트워크 보안(암,복호화)  압축/압축해제, 포맷 변환 수행 | ASCII, Mpeg, jpg, MME |
| **세션** | 소켓 프로그램  동기화  세션 연결/관리/종료 | 전송모드 결정(반이중, 전이중), SQL, RPC |
| **전송** | 데이터 전송보장  흐름 제어  Quality Of Service(QOS) | 세그먼트 | TCP, UDP, SCTP |
| **네트워크** | 통신경로 설정, 중계기능 담당  라우팅  IPv4 & IPv6 | 패킷 | IP, ICMP, IGMP, ARP, RARP, NAT, RIP, BGP |
| **데이터 링크** | 오류제어, Frame화  매체제어(MAC)  에러검출, 에러정정, 흐름제어 | 프레임 | 이더넷, 토큰링, PPP, SLIP, 802.11(WLAN) |
| **물리** | 물리적 연결설정, 해제  전송방식, 전송매체 | 비트 스트림 | 기계적 전기적, 절차적 규칙 |

**TCP/IP 프로토콜 그룹**

현재 인터넷에서 사용하는 프로토콜 그룹. 상호작용하는 모듈로 이루어진 계층적 프로토콜이다. 각 모듈은 특정 기능을 제공한다. 4 or 5 계층 모델로 간주된다. 두 모델 모두 적용 가능하다.

응용, 표현, 세션 -> 애플리케이션, 전/네/데/물 로 나뉜다.

|  |  |
| --- | --- |
| 계층 | 설명 |
| 데이터링크 | 전송 매체에 프레임을 송수신하는 역할 |
| 네트워크 | IP -> 호스트 네트워크 주소를 관리, 패킷을 라우팅  ARP -> 네트워크 호스트들의 하드웨어 주소를 얻는데 사용  ICMP -> 패킷 전송에 관한 오류 처리를 담당 |
| 전송 | 종단간 통신 서비스 제공을 담당한다. TCP/UDP가 있다. TCP는 데이터의 확실한 전송이 필요한 경우 사용, UDP는 데이터의 정확한 전달 보장 x |
| 응용 | 응용프로그램이 네트워크에 접근 가능하도록 인터페이스 기능 수행 |

**통신단위**

물리 – 비트, 네트워크 – 패킷, 전송 – 세그먼트 || 데이터그램 || 패킷, 응용 – 메시지

**캡슐화와 역캡슐화**

발신지 – 캡슐화, 수신지 – 역캡슐화, 데이터 링크 – 캡슐화/역캡슐화 이뤄지지 않음

**주소 지정**

TCP/IP 프로토콜을 이용한 인터넷은 3개의 다른 주소를 이용한다. -> 물리(MAC), 논리(IP), 포트 주소

**논리 주소:** 물리 주소는 홉에서 홉으로 갈 때마다 변경되지만 논리 주소는 변경되지 않는다. 물리 네트워크와 독립적으로 전세계적인 통신 서비스를 위해 필요. 현재 인터넷에 연결된 호스트를 유일하게 식별할 수 있는 32비트 주소 체계를 사용한다. 인터넷상의 두 호스트는 동일한 IP 주소를 사용할 수 없다.

**포트 주소:** 인터넷의 최종 목적은 한 프로세스가 다른 프로세스와 통신할 수 있도록 하는 것이다. A가 TELNET을 이용해 C와 통신하는 동시에 FTP를 이용해 B와 통신하려면 서로 다른 프로세스를 식별하기 위한 포트가 필요하다.

**다중화와 역다중화:** TCP/IP는 몇몇 계층에서 여러 프로토콜을 사용하기 때문에 발신자측은 다중화, 수신자측은 역다중화한다.

20절. TCP/IP

**물리 계층**

**데이터와 신호:** 응용, 전송, 네트워크, 데이터 링크의 통신은 논리적, 물리 계층 통신은 물리적이다. 호스트-대-라우터, 라우터-대-라우터, 라우터-대-호스트, 교환기 등이 물리적 통신이다.

**데이터링크 계층**

**노드와 링크:** 데이터링크 계층의 통신은 노드-대-노드. 데이터 유닛은 하나의 지점으로부터 LAN과 WAN 등 많은 네트워크를 통해 전달됨. LAN과 WAN은 라우터를 통해 전달. 이것을 노드와 링크로 나타내는 게 데이터링크 계층이다.

**서비스 순서**

**1. 프레임 짜기(Framing):** 각 노드에서 다음 노드에게 데이터그램을 전송하기 전에 네트워크계층에서 전달받은 데이터그램(패킷)을 프레임으로 캡슐화한다.

**2. 흐름 제어(Flow Control):** 서로 다른 데이터링크 계층 프로토콜은 서로 다른 흐름 제어 전략을 활용한다. 전송 계층에서 흐름 제어는 매우 중요.

**3. 오류 제어(Error Control):** 전기 신호는 오류에 취약하기 때문에 프레임 또한 오류에 취약. 이에 따라 첫 번째로 오류 검출이 필요하다. 오류 검출 후에 송신자 노드에서 오류를 수정하거나 폐기하고, 재전송을 송신 노드에게 요청해야만 한다.

**4. 혼잡 제어(Congestion Control):** 프레임 혼잡으로 프레임 손실이 발생해도 어떤 광역 네트워크 이외의 대부분의 데이터링크 계층 프로토콜은 직접적으로 혼잡을 완화하기 위해 혼잡 제어를 사용하지 않는다. 일반적으로 혼잡 제어는 종단-대-종단 통신이기 때문에 네트워크 또는 전송 계층 문제로 여겨진다.

**프레이밍 – 흐름 제어 – 오류 제어 – 혼잡 제어(일반적으로는 네트워크나 전송 계층의 일)**

**네트워크 계층**

**라우팅:** 라우팅은 route의 현재 진행형으로 경로배정을 의미한다. 패킷이 송신지에서 목적지까지 갈 수 있도록 경로를 배정하는 것.

**포워딩:** 포워딩은 라우팅 프로토콜을 실행할 때, 라우터상 하나의 인터페이스에 패킷이 도착했을 때 라우터가 취하는 행동이다. 라우터는 일반적으로 포워딩 테이블 혹은 라우팅 테이블이라는 표에 의거해 행동한다.

**IPv4:** 8 \* 4 =32bit. 전체 주소 공간은 A, B, C, D, E 5개의 클래스로 구분된다(Classful addressing).

**클래스A:** 첫 비트 0, 첫 비트의 나머지 7비트가 네트워크 주소. 하위 세 바이트는 호스트 주소

**클래스B:** 처음 두 비트 10, 이후 두 번째 바이트까지 네트워크 주소. 마지막 두 바이트는 호스트 주소

**클래스C:** 처음 세 비트 110, 이후 세 번째 바이트까지 네트워크 주소. 네트워크마다 254개 호스트 수용 가능, 작은 규모의 네트워크에서 사용

**클래스D:** 처음 네 비트 1110. 이후 네트워크 주소와 호스트 주소 구분 없음. 전체 주소가 멀티캐스트용. 멀티미디어 데이터, 리얼타임 비디오 등을 보낼 때 사용

**클래스E:** 처음 네 비트 1111. 추후 사용(시험용)을 위해 예약된 주소

**주소 고갈:** 클래스 기반 주소는 주소 고갈로 인해 더이상 사용하지 않는다.

**서브네팅과 슈퍼네팅:** 주소 고갈을 완화하기 위해 등장. 네트워크부는 부족한데 호스트부는 낭비되는 문제 발생. 서브네팅 도입으로 해결. 서브네팅의 핵심은 할당된 서브넷마스크(ex.255.255.255.0)를 원래의 IP주소와 AND 연산하는 것. 네트워크ID를 서브 네트워크ID로 쪼개어 더 많은 네트워크 주소를 할당하고 호스트 부에 원래보다 적은 양의 주소를 할당한다.

**클래스 없는 주소 지정:** 클래스 기반 주소 지정에서는 서브네팅을 적용해도 고갈 문제를 해결할수 없었다. 그런데 왜 바로 IPv6로 넘어가지 않았나? -> 더 큰 주소 공간을 확보하려면 IP주소 길이의 증가로 인해 IP패킷 형식까지 변경해야 했기 때문… 단기적 관점의 해결책으로 클래스 없는 IPv4 주소지정 방식이 사용되었다. -> 클래스 권한 제거

**CIDR(Classless Inter-Domain Routing):** ISP가 클라이언트가 요구하는 양만큼만 주소를 잘라서 공급할 수 있는 장점이 있다. 따라서 라우팅 테이블의 비대화를 막고 인터넷을 여러 개의 Addressing Domain으로 나누어 라우팅 정보량을 줄여준다. 한 도메인 내에서는 도메인 내의 모든 라우팅 정보가 공유된다.

CIDR표기법(슬래시 표기법) -> byte.byte.byte.byte/prefix length -> 클래스가 없기 때문에 슬래시 뒤에 네트워크 ID의 길이를 적어준다.

**VLSM(Variable Length Subnet Mask, 가변 길이 서브넷 마스킹):** VLSM은 서브네팅을 여러 번 반복해 네트워크를 크기가 다른 여러 세브넷으로 구분한다.

문제. C class IP주소를 가진 회사에서 100개의 주소를 필요로 하는 부서와 25개의 주소를 필요로 하는 4개의 부서를 서브네팅 하는 경우의 적용(192.168.120.0)

**특수 주소: 다섯 가지 ->** this-host 주소, 제한된 브로드캐스트 주소, 루프백 주소, 사설 주소, 멀티캐스트 주소

**디스-호스트 주소: 0.0.0.0/32** -> 호스트가 IP데이터그램을 보내려 하지만 근원지 주소인 자신의 주소를 모를 때 사용

**제한된 브로드캐스트 주소: 255.255.255.255/32** -> 호스트나 라우터가 네트워크상의 모든 장치로 데이터그램을 보낼 때 사용. 네트워크상의 라우터가 이런 패킷을 차단하기 때문에 네트워크 외부로 패킷을 보낼 수는 없다.

**루프백 주소: 127.0.0.0/8** -> 이 블록 내의 패킷은 호스트를 벗어나지 않고 호스트에 남는다. 소프트웨어 테스트용. 127.0.0.1이 가장 많이 쓰인다.

**사설 주소: RFC 1918 ->** RFC1918이 정의한 사설 주소에서만 쓸 수 있는, 라우팅이 불가능한 특수 주소 집합. 공중 인터넷에 존재하지 않는 주소들. 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16이 사설 주소로 지정되어 있다.

**사설 클래스A: 24bit block, 10.0.0.0 – 10.255.255.255**

**사설 클래스B: 20bit block, 172.16.0.0 – 172.31.255.255**

**사설 클래스 C: 16bit block, 192.168.0.0 – 192.168.255.255**

**네트워크 계층 프로토콜**

**IP(인터넷 프로토콜):** 비신뢰/비연결형 데이터그램 프로토콜. 최선형 전송 서비스(Best-Effort Delivery Service)최선형 전송이란 IPv4패킷이 훼손, 손실, 틀린 순서로 도착, 지연도착, 네트워크 혼잡 등을 발생을 시킬 수 있음을 의미한다. 신뢰성이 중요하다면 IPv4는 TCP처럼 신뢰성을 보장하는 전송 계층 프로토콜과 함께 사용되어야 한다. TCP/UDP, ICMP, IGMP 데이터는 모두 IP 데이터그램을 사용해 전송된다.

**IPv4 데이터그램 암기사항**

**데이터그램이란? -> IP가 사용하는 패킷으로 TCP/UDP, ICMP, IGMP 등이 데이터그램을 통해 전송된다. IP 프로토콜은 비신뢰성, 비연결성(Connectionless), 주소 지정, 경로 설정 방식의 프로토콜이며 데이터그램은 가변 길이의 패킷으로 헤더(주소/제어필드)와 페이로드(데이터)로 이루어져 있다. 헤더는 20~60바이트 길이이며 라우팅과 전송에 필수적인 정보를 가지고 있다. TCP/IP에서는 일반적으로 헤더를 4바이트로 표현한다. 버전 -> IPv4, 헤더 길이 -> 옵션을 포함한 헤더길이, TOS(Type Of Service) -> 3비트의 우선권(Precedence) 필드와 4비트의 TOS 필드 그리고 1비트의 예약 필드, Total Length -> 전체 길이를 바이트로 표현, TTL(Time To Live) -> 패킷이 경유할 최대 홉수, 식별자(Identification) -> 호스트가 보낸 각 데이터그램을 식별, 플래그 -> 세 개의 비트로 단편화 정보 표시 이하 그림**

**IPv4 데이터그램**

**텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명**

**ARP(Address Resolution Protocol):** 자신의 IP, MAC, 수신자의 IP는 알고 있지만 수신자 측의 MAC은 모르는 상황에서 물리 계층 브로드캐스트를 통해 모든 호스트에게 패킷을 전송한다. -> ARP 메시지를 수신한 호스트 또는 라우터는 수신 IP 주소와 자신의 IP 주소를 대조한 뒤 자신의 MAC을 요구하는 경우라면 ARP 응답 메시지를 송신자에게 전송한다. -> 각 시스템에는 ARP Cache가 있고 Cache에 이 정보를 보관한다. 정보는 일정 시간 경과 후 삭제한다.

**MAC(Media Access Control):** NIC의 ROM칩에 생산자가 프로그램한 고유 물리주소. 데이터 링크 계층이 이 MAC주소를 이해하고 작업하며 네트워크 계층은 IP 주소를 이해하고 작업한다.

**ARP Cache Table:** ARP를 통해 알아낸 MAC 정보는 OS에 따라 다르지만 통산 1~2분 정도 캐시에 저장한다. arp -a 명령을 통해 ARP Cache table을 살펴보면 dynamic과 staticd으로 타입이 나뉜다. Static은 관리자에 의해 정적으로 생성된 것으로 임의로 삭제하거나 시스템 종료 전까지 지속되고 dynamic은 ARP에 의해 동적으로 설정되어 일정 시간 동안만 유지된다.

**ARP 메시지 종류:**

1. ARP 요청 메시지 -> 특정 IP주소에 대한 물리적 주소를 요구한다. 호스트는 수신자측 물리 주소를 모르기 때문에 브로드캐스트로 전송한다.
2. ARP 응답 메시지 -> 유니캐스트로 물리 주소를 알려준다. ARP 요청 메시지를 수신한 호스트 또는 라우터는 ARP 응답 메시지를 전송한다.(호스트가 라우터를 넘어 다른 네트워크에 있으면 해당 호스트를 대신해 응답 메시지를 전송한다.)

**RARP(Reverse Address Resolution Protocol):** ARP의 반전. MAC은 알고 IP를 모를 때 사용한다. RARP는 저장장치가 없는 호스트에서 주로 사용한다. 또 RARP 응답은 일반적으로 RARP 서버에서 생성한다. 호스트가 자신의 MAC 정보를 담은 RARP 요청 메시지를 만들어 브로드캐스트 한다. -> RARP 서버는 요청자의 IP 주소를 담은 RARP 응답 메시지를 만들어 요청자의 MAC 주소로 유니캐스팅한다.

**GARP(Gratuitous ARP):** 별도의 프로토콜이 아니다. Sender IP와 Target IP가 동일한 ARP 요청을 말한다.

장비가 ARP 요청 브로드캐스트를 통해 다른 장비에게 네트워크에 있는 자신의 존재를 알릴 때 사용한다. 이 패킷을 수신한 장비는 자신의 ARP Cache에 해당 정보가 있다면 이를 갱신한다. 자신의 MAC을 동일 네트워크 상의 다른 장비에게 알려 ARP Cache를 갱신하기 위해 사용한다.

**GARP 쓰임**

1. **IP충돌 감지:** 자신과 동일한 IP가 설정된 호스트가 있다면 해당 호스트로부터 ARP 응답이 오기 때문에 충돌 여부를 확인 가능. 호스트 IP를 변경하거나 재부팅 시에 GARP 패킷이 생성된다.
2. **상대방 ARP Cache 정보 갱신:** 수신 측에서 자신의 ARP Cache에 Sender IP 정보를 갱신한다. 이 부분이 취약점이 될 수 있다. 상대방을 인증하지 않고 Cache정보를 갱신해 악의적 목적의 공격자에 의해 MAC 정보가 위조, 변조될 수 있다.

**ICMPv4(Internet Control Message Protocol version 4):** 호스트가 라우터 상태나 다른 호스트, 라우터의 정보를 획득하기 위해 사용. IP의 동반 프로토콜이다.

**ICMPv4 메시지:** 오류보고와 질의 메시지. 오류보고는 라우터(목적지)나 호스트가 IP 패킷을 처리하는 도중에 탐지하는 문제를 보고한다. 첫 번째 필드인 Type이 메시지 유형을 나타낸다. 코드 필드는 특정 메시지 유형의 이유를 지정한다. 마지막 공통 필드는 검사 합 필드.

* **오류보고 메시지:** ICMP의 주된 임무 중 하나는 IP 데이터그램의 프로세싱 동안 발생하는 오류를 보고하는 것. 오류 수정은 직접 할 수 없고 상위 계층 프로토콜에 맡긴다. 데이터그램으로 알 수 있는 경로 정보는 근원지와 목적지 IP밖에 없기 때문에 오류 메시지는 언제나 최초의 근원지에 보내진다.
  + **ICMP 메시지 중요한 점**

1. **ICMP 오류 메시지는 메시지를 운반하는 데이터그램의 응답으로 생성되지 않는다.**
2. **ICMP 오류 메시지는 처음 단편이 아닌 단편 데이터그램을 위해 생성되지 않는다.**
3. **ICMP 오류 메시지는 멀티캐스트 주소를 가진 데이터그램을 위해 생성되지 않는다.**
4. **ICMP 오류 메시지는 127.0.0.0 또는 0.0.0.0과 같은 특수 주소의 데이터그램을 위해 생성되지 않는다.**

* **목적지 도달 불가(Destination Unreachable, Type 3):** 가장 널리 사용되는 오류 메시지. 데이터그램이 왜 목적지에 도달하지 못했는지에 대한 오류 메시지를 정의하기 위해 0~15까지의 다른 코드를 사용한다.
* **근원지 억제(Source Quench, Type 4):** 송신자에게 네트워크 충돌이 발생해 데이터그램이 폐기되었음을 알린다. 이 메시지를 받은 근원지는 데이터그램 송신을 천천히(또는 억제) 수행한다. ICMP가 이런 메시지를 통해 IP프로토콜에는 원래 없는 혼잡제어 메커니즘을 추가하는 것이다.
* **재지정 메시지(Redirection, Type 5):** 발신자가 잘못된 라우터를 통해 메세지를 전송할 때 사용. 라우터는 알아서 적절한 라우터로 메시지를 보내도록 재지정되지만 발신자에게 향후 디폴트 라우터로 변경하라고 알린다. 따라서 메시지 안에 디폴트 라우터 IP가 포함된다.
* **시간 경과 메시지(Time Exceeded, Type 11):** 타임아웃이 발생해 IP패킷이 폐기되었음을 알린다. 코드를 통해 타임아웃 사유를 알 수 있다.

1. **Code 0(TTL exceeded In Transit): IP패킷이 최종 목적지에 도달 전에 TTL 값이 0이 됨**
2. **Code 1(Fragment reassembly time exceeded): IP 패킷 재조합 과정에서 타임아웃이 발생해 해당 IP 데이터그램이 모두 폐기됨. IP 데이터그램의 단편이 전송과정에서 손실될 경우 재조합 실패로 발생**

* **매개변수 문제(Parameter Problem, Type 12):** 헤더에 문제가 있거나(코드 0) 어떤 옵션이 없거나 옵션의 의미를 알 수 없는 경우(코드 1)에 전송될 수 있다.

**대표적인 ICMP 메시지**

|  |  |
| --- | --- |
| **ICMP** | **메시지 기능** |
| Echo Request | 원하는 호스트와 IP 연결 됐는지 확인 |
| Echo Reply | ICMP Echo Request 응답 메시지 |
| Redirect | 데이터를 보낸 호스트에게 목적 IP주소에 좀 더 적합한 경로가 있음을 알리기 위해 라우터가 보내는 메시지 |
| Source Quench | 데이터를 보낸 호스트에게 IP 데이터그램이 라우터의 집중 현상에 의해 손실되고 있음을 알리기 위해 라우터가 보내는 메시지. 데이터를 보내는 호스트는 전송률을 낮추게 된다. Source Quench는 ICMP에서 선택적 메시지이며 대부분 구현되지 않는다. |
| Destination Unreachable | 라우터나 목적 호스트에 의해 보내지며 데이터그램이 전달되지 못했다는 것을 데이터를 보낸 호스트에게 알린다. |

**질의 메시지:** 인터넷에서 호스트나 라우터가 활성화되었는지를 알아보거나 두 장치 사이에 IP 데이터그램이 단방향 시간인지 왕복 시간인지 찾는다. 에코요청(type 8)과 에코응답(type 0) 메시지 쌍은 다른 호스트나 라우터가 활성화됐는지 여부를 테스트하기 위해 호스트나 라우터가 사용한다. 핑과 트레이스루트(traceroute) 같은 디버깅 도구에서 에코 요청과 에코 응답으로 메시지 쌍을 확인할 수 있다.

**IPv6:** IPv4의 한계(32비트, 43억개)로 등장, 32 비트 주소체계를 4배 확장한 128비트 주소 길이를 사용한다. 약 3.4 \* 10^38(2^128) 개 주소 생성 가능. 보안 문제, 라우팅 효율성 문제 해결, QoS(Quality of Service) 보장, 무선 인터넷 지원과 같은 다양한 기능을 제공한다. 이동전화, 가전제품 등에 도입 추진 중.

**IPv6 특징:**

* **확장된 주소 공간:** NAT(Network Address Translation) 불필요, 유니캐스트, 애니캐스트, 멀티캐스트 주소 형태 지원, 100억명이 1000개 단말기 사용 가능 -> 2^128
* **새로운 헤더 포맷:** IPv4에서 사용하지 않던 헤더를 지우고 확장 헤더를 도입해 선택적 사용을 가능하게 함. 패킷을 중계하는 라우터의 부하를 줄인다. 또 헤더를 고정 길이로 변경해 시스템이 헤더 길이를 예측해 빠른 처리가 가능하게 만들었다.
* **패킷 단편화(fragmentation) 필드 삭제:** 패킷을 보내기 전에 전송에 적합한 패킷크기를 알아내는 MTU(Maximum Transfer Unit) 탐색 기능 추가
* **체크섬 필드 삭제:** 데이터링크 계층에서 체크섬 계산이 행해지므로 삭제

**IPv6는 향상된 서비스를 지원한다:** IPv6 헤더에 플로우 레이블 필드를 사용해 응용프로그램에 의해 생성된 트래픽을 실시간/비실시간으로 구분할 수 있다.

**보안기능:** IPv4는 설계 시 보안 규정이 없어 별도의 보안 프로토콜(IPSec)을 필요로 한다.(Add-on)

IPv6는 프로토콜 내부에 보안 기능을 탑재할 수 있다. 인증 절차, 데이터 무결성 보호, 메시지 발신지 확인 기능을 제공한다. 또 확장 헤더를 통해 네트워크 계층의 종단간 암호화를 제공, 패킷 변조를 방지한다.

**IPv6 주소 체계:** IPv4와 달리 브로드캐스트가 없고 애니캐스트가 있는 게 특징. 유니캐스트는 단일 인터페이스(컴퓨터, 라우터)를 정의하고 특정 컴퓨터에게만 전달한다. 애니캐스트 주소는 단일 주소를 공유하는 컴퓨터 집합으로 정의된다. 애니캐스트 주소로 결정된 패킷은 가장 가까이 있는 애니캐스트 그룹의 구성원(가장 짧은 경로를 가지는)에게만 전송된다. 멀티캐스트 주소는 컴퓨터 그룹을 정의한다. 애니캐스트와 차이는 애니캐스트는 패킷의 복사본이 그룹 중 하나의 컴퓨터에만 전송되는데 반해 멀티캐스트는 그룹의 각 컴퓨터가 복사본을 수신한다는 것이다. 브로드캐스트는 멀티캐스트의 특수한 경우로 처리한다.

**IPv4에서 IPv6로 변환: 이중 스택**(동시 사용), **터널링** – 종단간 두 컴퓨터가 IPv6로 소통할 때 Ipv6 패킷이 IPv4를 사용하는 지역을 지나가면 IPv4로 캡슐화 한 뒤 빠져나올 때 역캡슐화 하는 것, **헤더 변환** – IPv6가 주류가 되고 IPv4는 소수가 된 시점에서 IPv6를 사용하는 컴퓨터가 IPv4를 이용하는 컴퓨터와 소통하기 위해 헤더를 변환시켜 통신한다.

**전송 계층**

**UDP(User Datagram Protocol):** 비연결형/비신뢰성 프로토콜. 래퍼 프로토콜 -> 단순히 애플리케이션 데이터를 메시지 포맷으로 포장해 IP에게 전달만 하기 때문. 호스트 간 통신 대신에 프로세스 간 통신을 제공하는 것 외에 어떠한 서비스도 추가하지 않는다. 최소의 오버헤드를 가진 간단한 프로토콜. UDP 패킷은 사용자 데이터그램(User Datagram)이라고도 불리며 고정된 크기의 8바이트 헤더를 가진다. 8바이트의 헤더는 2바이트 짜리 필드 4개로 만들어진다.

**헤더구조 -> 송신 포트 주소(2바이트), 도착지 포트 주소(2바이트), 총 길이(2바이트), 체크섬(2바이트)**

**UDP 서비스:**

* **프로세스-대-프로세스 통신:** IP주소와 포트 번호의 결합인 소켓 주소를 통해 프로세스-대-프로세스 통신을 한다.
* **비연결형 서비스:** 각 사용자 데이터그램은 독립적이다. 동일한 근원 프로세서에서 동일한 목적지 프로그램으로 들어가도 서로 관계없다.
* **흐름 제어:** UDP는 단순 프로토콜로 흐름 제어와 윈도우 메커니즘이 없다. 따라서 들어오는 메시지로 수신자 측에서 오버플로우가 발생할 수 있다. 흐름 제어의 결여는 UDP를 이용하는 프로세스가 스스로 제공해야 한다.
* **오류 제어:** 검사합을 제외하고 UDP에는 오류 제어가 없다. 송신자가 메시지가 손실이 되었는지 또는 중복이 되었는지 알 수 없다.
* **혼잡 제어:** 혼잡 제어도 없다. UDP는 전송 패킷이 너무 작고 산발적이어서 혼잡을 발생시키지도 않는다. 네트워크 혼잡에 관한 설명은 UDP가 요즘 음성과 영상의 실시간 스트리밍에 사용되는 시점에 맞을 수도 맞지 않을 수도 있다.
* **다중화와 역다중화:** 호스트에서 하나의 UDP가 있지만 UDP 서비스를 사용하기 원하는 다수의 프로세스가 있을 수 있다. 이 문제를 해결하기 위해 UDP는 다중화 및 역다중화 한다.

**UDP의 일반적인 응용:** FTP와 같이 대량 데이터를 전송하는 프로세스에는 사용하지 않는다. UDP는 내부 흐름 및 오류 제어 기법을 가진 프로세스에 적당하다. TFTP는 흐름 및 오류 제어를 포함하므로 UDP를 사용할 수 있다. 또한 UDP는 멀티캐스팅을 위한 전송 프로토콜이다. UDP는 SNMP와 같은 관리 프로세스에 사용된다. RIP와 같은 경로 갱신 프로토콜을 위해서도 사용된다. UDP는 실시간 응용에 사용된다.

**TCP(Transmission Control Protocol)**

**TCP 서비스의 특징: 연결형, 신뢰성, 양방향, 승인, 스트림 중심, 흐름 관리**

1. **프로세스-대-프로세스 통신:** UDP처럼 포트 번호를 사용해 프로세스 간 통신 제공
2. **스트림 전송 서비스:** 바이트의 흐름으로 데이터를 송/수신한다. 가상의 튜브에 의해 2개의 프로세스가 연결되는 것처럼 보이는 환경을 만든다.
3. **송신 및 수신 버퍼:** 송/수신 서비스가 똑같은 속도로 전송/처리하는 게 아니기 때문에 TCP 버퍼가 필요하다. 송신 버퍼와 수신 버퍼 두 가지가 있다. 두 버퍼는 흐름 및 오류 제어에 사용된다.
4. **전이중 통신(Full-duplex service):** 데이터는 동시에 양방향으로 전달될 수 있다. 이를 위해 TCP는 송신/수신 버퍼를 가지며 세그먼트는 양방향으로 이동한다.
5. **다중화와 역다중화:** UDP와 마찬가지. 차이는 TCP는 연결-지향 프로토콜이기 때문에 프로세스 쌍을 위해 연결을 설정해야 한다.
6. **연결**-**지향 서비스:** 물리적 연결이 아니라 논리적 연결을 말한다. 논리적 연결 통로를 통해 데이터를 송/수신해 전송 순서를 보장한다. 이를 순서 제어(Sequence Control)라고 한다.
7. **신뢰성 있는 서비스**: 데이터의 안전하고 확실한 도착을 확인하기 위한 확인응답 기법 사용

**세그먼트**

**TCP에서 패킷을 세그먼트라고 한다.**

**세그먼트는 20~60바이트의 헤더와 응용 프로그램으로 구성된다. 헤더는 선택 항목이 없으면 20바이트, 있으면 60바이트까지**

**세그먼트 암기사항:**

1. **근원지 포트주소:** 송신 프로세스의 포트 번호(16비트)
2. **목적지 포트주소:** 수신 프로세스의 포트 번호(16비트)
3. **순서번호:** 32비트, 세그먼트에 포함된 첫 번째 데이터 바이트에 할당된 번호
4. **확인응답 번호:** 32비트, 송신자가 수신자로부터 받기를 기대하는 바이트 번호를 정의한다. 수신자가 상대방으로부터 바이트 번호 x를 성공적으로 수신하면 x + 1이 확인응답 번호가 된다. 확인응답과 데이터는 함께 피기백이 될 수 있다.
5. **헤더 길이:** 4비트, TCP 헤더를 4바이트 단위의 개수로 나타낸 것
6. **제어:** 1비트 \* 6, 6개의 제어(플래그) 비트 정의
7. **윈도우 크기:** 16비트, 수신측에서 송신측에 보내는 Receiver Window Size로 수신 버퍼의 여유 공간 크기를 의미한다. 송신측에서 상대의 여유 공간 크기를 파악하고 흐름제어를 수행할 수 있다.
8. **검사합(checksum):** 16비트, UDP와 동일한 절차를 따른다. UDP검사합은 옵션이지만 TCP 검사합은 필수이다.
9. **긴급 지시자(Urgent Pointer):** 16비트로 긴급 플래그 값이 설정되었을 때만 유효

![텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명]()

|  |  |  |
| --- | --- | --- |
| **플래그** | **의미** | **설명** |
| **URG** | **긴급** | ‘1’ 설정 시 우선순위, 긴급 필드값 사용 |
| **ACK** | **승인** | ‘1’ 설정 시 세그먼트가 승인 포함 |
| **PSH** | **밀어넣기** | 받는 즉시 애플리케이션에 송신 |
| **RST** | **리셋** | 송신 장비 문제, 연결 초기화 |
| **SYN** | **동기화** | 순서 번호 동기화, 연결 수립 |
| **FIN** | **종료** | 송신 장비가 연결 종료 요청 |

**TCP연결:** 연결-지향 프로토콜이기 때문에 근원지와 목적지 사이에 가상 경로를 설정하고 송/수신. 연결 설정, 데이터 전송, 연결 종료의 3단계를 거친다.(Three-way Handshaking)

1. 클라이언트가 SYN 플래그를 세팅한 세그먼트를 서버에 전송한다. 세그먼트는 순서번호와 동기화를 위한 것이다. SYN세그먼트는 데이터를 운반할 수는 없지만 순서번호 하나를 소비한다.
2. 서버는 2개의 플래그 비트가 설정된 두 번째 세그먼트 SYN + ACK 세그먼트를 전송한다. 마찬가지로 데이터를 운반할 수 없으며 순서번호 하나를 소비한다.
3. 클라이언트가 세 번째 세그먼트를 전송한다. ACK 세그먼트이다. ACK 플래그와 확인응답 번호 필드를 사용해 두 번째 세그먼트를 수신한 것에 대한 확인차 보내는 것이다. 데이터를 운반하지 않으면 ACK세그먼트는 순서번호를 소비하지 않는다.

**SYN Flooding 공격:** 쓰리 웨이 핸드셰이크 과정에서 공격자가 무수한 가짜 IP와 포트를 가진 SYN 세그먼트를 날린다. -> 서버는 이를 클라이언트로 인식하고 SYN-ACK 패킷을 보내고 응답을 대기한다. 이 과정에서 서버의 리소스가 낭비되고 실제 클라이언트들이 서버와 연결을 설정하지 못하게 된다. -> 서버는 자원 부족으로 정상적인 응답을 처리하지 못하게 되고 마비되거나 느려진다. -> 모든 요청에 대해 서비스를 거부하는 DoS 공격에 속한다. -> 현재는 SYN Cookie 기법을 사용해 SYN Flooding을 감지하고 방어한다.

**데이터 전송:** 연결이 형성되면 양방향으로 데이터를 전송한다. 클라/서버가 모두 데이터와 확인 응답을 보낼 수 있다. 클라이언트가 보낸 데이터 세그먼트에는 서버 TCP가 데이터가 수신되자마자 서버 프로세스에게 데이터를 전달하는 것을 알도록 PSH(Push) 플래그가 설정되어 있다. 그러나 서버 -> 클라이언트로 전달된 세그먼트는 밀어넣기 플래그를 설정하지 않는다.

**연결 종료:** 시작은 클라이언트(능동 오픈)이지만 종료는 양쪽 모두 가능하다. 현재 대부분의 경우는 세 방향 핸드셰이킹과 절반-종료 옵션을 가진 네 방향 핸드셰이킹 두 방법을 허용한다.

* **세 방향 핸드셰이킹의 종료:** 클라이언트 TCP가 클라이언트 프로세서로부터 종료 명령을 수신한 후 FIN 플래그가 설정된 첫 FIN 세그먼트를 전송한다. -> 서버는 FIN 플래그 수신 후 상황처리를 알리고 FIN 수신을 확인하며, 동시에 다른 방향으로 연결 종료(FIN + ACK)를 위해 두 번째 세그먼트를 송신한다. -> 클라이언트가 FIN + ACK을 받고 마지막 세그먼트인 ACK 세그먼트를 전송한다.
* **네 방향 핸드셰이킹의 종료(절반-폐쇄):** TCP에서 데이터를 받고 있는 도중에 어느 한쪽이 데이터 전송을 중단하는 것을 절반 폐쇄라고 한다. 보통 클라이언트가 시작한다.
* 클라이언트가 FIN 세그먼트를 전송해 절반-폐쇄한다.
* 서버는 ACK 세그먼트를 송신해 절반-폐쇄를 받아들인다.
* 클라이언트는 서버로 데이터 전송을 멈추지만 서버는 여전히 데이터를 송신할 수 있다.
* 서버가 처리된 모든 데이터를 보냈을 때 ACK에 의해 확인 응답되는 FIN 세그먼트가 전송된다.
* 절반-폐쇄 이후에 데이터는 서버에서 클라이언트로 이동 가능하고 확인 응답은 클라이언트에서 서버로 이동할 수 있다. 하지만 클라이언트는 더 이상 서버에 데이터를 전송할 수 없다.
* **연결 재설정:** 한 종단의 TCP가 연결 요구를 거부하거나 현재 연결을 중지, 혹은 유휴 연결을 종료할 수 있다. 이런 재설정은 RST 플래그를 가지고 수행한다.

**응용 계층**

**주요 응용 계층 프로토콜**

**HTTP:** 웹페이지를 가져오기 위해 어떻게 클라이언트-서버 프로그램을 작성할지 정의한다. 클라이언트는 임시 포트를, 서버는 80포트를 사용해 클라이언트가 요청하고 서버는 응답한다.

**FTP:** 파일 전송 프로토콜. TCP/IP표준이다. 종단간 두 시스템은 서로 다른 파일 이름, 방법으로 데이터를 표현할 수 있다. 또 두 시스템은 서로 다른 디렉터리 구조를 가질 수도 있다. FTP는 이러한 모든 기능을 제공한다.

**TELNET: TER**mina**L NET**work의 약자. 최초의 원격 로그인 프로토콜. 로그인 이름과 암호가 필요하지만 평문으로 전송하기에 해킹에 취약하다. 해커는 로그인 이름과 암호를 도청해 얻을 수 있다. 이런 보안 문제 때문에 SSH(Secure Shell)을 사용한다.

**DNS:** 중앙 디렉터리 시스템이 모든 매핑을 보유하지 않고 매핑 정보를 작게 나누어 전 세계의 서로 다른 컴퓨터에 저장한다. 각 호스트는 매핑이 필요할 경우 DNS를 통해 해당 정보를 가진 가장 가까운 컴퓨터와 통신한다.

**21절. 라우팅**

**라우팅:** 송신자의 목적 경로 설정 -> 패킷이 정해진 경로를 따라 이동

**라우터:** 네트워크 계층 주소를 참조해 최선의 경로를 찾는다. 라우팅 알고리즘은 관리자가 직접 설정하는 정적 라우팅 알고리즘과 네트워크 환경 변화에 능동적으로 대처 가능한 동적 라우팅 알고리즘으로 구분된다.

|  |  |  |
| --- | --- | --- |
| 구분 | 정적 라우팅 | 동적 라우팅 |
| 라우팅 테이블 관리 | 수동 설정  네트워크 변화(라우터 추가/변경/회선 장애…) 인지 자동 인지 불가 | 자동  네트워크 변화 자동인지 후  네트워크 경로 재구성 |
| 인터페이스 | 변경 적을 때 유리 | 변경 많을 때 유리 |
| 노드 추가/변경 | 운영 요원이 직접 | 대처 용이 |
| 중간 경로 | 단일 경로에 적합 | 다중 경로에 적합 |

**라우팅 프로토콜**

네트워크 정보를 생성/갱신/교환/제어하는 프로토콜을 라우팅 프로토콜이라고 한다. Routed Protocol은 라우팅을 받는 고객이므로 라우팅 프로토콜과 다르다(TCP/IP, IPX, AppleTalk 등). 라우팅 프로토콜에는 RIP, IGRP, OSPF, EIGRP 등이 있다. 자동차 -> 라우터, 운전기사 -> 라우팅 프로토콜, 고객 -> 라우티드 프로토콜, 네비게이션 -> 라우팅 테이블

**AS(Autonomous System자율시스템):** 하나의 관리 도메인에 속한 라우터의 집합

AS의 내부에 있는 라우팅 프로토콜 -> IGF(RIP, OSPF), 외부 -> EGF(BGP)

**유니캐스트 라우팅**

**라우팅 알고리즘: 거리벡터 라우팅, 링크-상태 라우팅, 경로 벡터 라우팅**

1. **거리벡터 라우팅(DV, distance-vector):** 인접한 이웃들의 기초 정보를 이용해 작성된 최소 비용 트리이다. 자신이 모든 이웃에게 자신이 알고 있는 네트워크 정보를 끊임없이 알려준다.(불완전한 정보라도)

**거리벡터:** 거리벡터는 목적지까지의 경로를 제공하는 게 아니라 목적지까지의 최소 비용을 제공한다.

1. **링크-상태 라우팅**

**링크상태 데이터베이스(LSDB, Link-State Database):** 링크 상태 라우팅을 이용한 최소비용 트리 작성은 각 노드의 링크 상태를 완전히 표시한 맵이 필요하다. 이러한 링크의 상태집합을 LSDB; 링크 상태 데이터베이스라고 부른다. 각 노드에 모든 인터넷에 대한 정보가 포함된 LSDB는 플러딩이라는 과정을 통해 만들어진다.

1. **경로 벡터 라우팅(PV, Path-Vector):** 스페닝 트리가 출발에서 목적지까지의 경로를 결정한다. 스패닝 트리는 최소비용 트리가 아닌 고유의 규칙을 가진 트리이다. 목적지까지 하나 이상의 경로가 있다면 근원지(출발지)는 가장 최선의 규칙을 사용하는 경로를 선택할 수 있다.

**유니캐스트 라우팅 프로토콜**

**RIP: 거리벡터 알고리즘, OSPF: 링크상태 알고리즘, BGP: 경로벡터 알고리즘**

1. **RIP(Routing Information Protocol):** 가장 오래되고 널리 사용되는 내부 라우팅 프로토콜. 각 라우터는 거리 벡터 알고리즘을 사용한다. 라우팅 테이블에 자율 시스템 내의 다양한 네트워크에 도착하는 방법과 거리를 홉수로 저장한다. RIP는 여러 한계가 있지만 설정하기 쉽고 간단해서 널리 사용된다. IPv4에서 사용할 수 있는 RIP 버전1, 버전2와 IPv6에서 사용할 수 있는 RIPng가 있다. 프로토콜 기본 동작은 세 버전에서 거의 동일하지만 전송하는 메시지의 포맷이 조금씩 다르다.

* **홉 카운트:** 비용은 네트워크의 수를 의미하는 홉수로 정의된다.
* **RIP**의 경로 최대비용은 15이다. 16은 무한대로 생각한다. 이런 이유로 RIP는 AS에서 사용된다. AS는 15홉 이내의 반지름을 갖게 된다.
* **포워딩 테이블:** RIP에서 포워딩 테이블은 3개의 열을 가진다. 열의 속성은 순서대로 네트워크 주소, 패킷이 전송될 다음 라우터 주소, 홉수로 표현된 비용
* **RIP 구현:** 라우터의 RIP 소프트웨어는 RIP 메시지로 통신한다. RIP 메시지는 UDP를 사용해, RIP-1, RIP-2는 UDP 520번 포트를, RIPng는 521번 포트를 사용한다. RIP는 다른 라우팅 프로토콜처럼 3계층으로 간주되지만 애플리케이션처럼 메시지를 보낸다. 원래 버전은 RIP-1이라고 부르는데 지금의 RIP에 비해 기능이 적었다. RIP-2부터 CIDR 등의 기능을 지원하도록 수정되었다. RIPng는 IPv6와의 호환성을 위해 새로 만들어진 버전이다.
* **RIP 특징: 거리벡터 알고리즘, 벨만포드 프로토콜, 거리벡터 값은 홉카운트**
* **문제점:** 느린 수렴(Convergence), 라우팅 루프, 무한세기(Counting of infinity), 작은 무한값, 수렴 -> 모든 라우터가 네트워크의 위치에 대해 같은 정보를 가지게 되는 것. 모든 문제는 거리 벡터 알고리즘이 수렴하도록 설계된 데서 발생한다.
* **해결방안:** Triggered Update(홉수가 변경되었을 때 즉시 통보해 복구 시간을 줄인다.), Hold Down(메트릭이 무한대인 경로에 대해서는 일정 시간 동안 경로를 갱신하지 않고 전체 네트워크의 경로가 새로 경신될 때까지 기다린다.), Split Horizon(라우터 A가 B를 거쳐 C로 전송하는 경우에 B와 C 사이에 장애가 발생하면 A와 B사이에서 루프가 발생할 수 있다. 이를 해결하기 위해 B가 A에게 보낸 정보는 다시 B에게 보내지 않는다.), Route Poisoning(회선의 고장을 감지하면 즉시 해당 경로의 메트릭을 16으로 지정해 전체 네트워크에 방송함으로써 도달 불가능을 신속히 알린다.)

1. **RIPv2:** RIPv1의 예약필드를 활용해 RIPv1의 기능을 확장한 버전이다. 서브넷마스크 식별, 경로정보 인증, AS 구별, 브로드캐스트와 멀티캐스트 전송 기능이 추가되었다. 최대 홉수는 여전히 15로 제한되어 확장성 문제를 해결하지는 못했다.

|  |  |
| --- | --- |
| **항목** | **설명** |
| CIDR, VLSM 지원 | RIPv2에서는 CIDR과 VLSM(Variable-Length Subnet Making) 지원 |
| 메시지 전달 | RIPv1은 브로드캐스팅만, RIPv2는 브로드캐스팅, 멀티태스킹 지원 |
| 인증 | RIPv2에서는 메시지마다 인증 가능 |
| 호환성 | RIP-2는 RIP-1과 동일한 메시지 포맷을 사용 |

1. **IGRP(Interior Gateway Routing Protocol): RIP는** 매트릭으로 홉수만을 사용해 네트워크의 제대로 된 상태반영이 힘듦, 홉수 제한으로 대규모 네트워크에 적용하기 어려움. 하나의 매트릭 값만을 사용하는 대신 다섯 가지 네트워크 파라미터를 이용해 거리벡터를 계산하는 IGRP 제안(시스코). 네트워크 상황에 맞게 알고리즘을 변형할 수 있다.

**IGRP 매트릭 계산 요소:** 대역폭(1200bps ~ 10Gbps), 지연(1 ~ 2^24), 신뢰도(1 ~ 255), 부하(1 ~ 255), MTU

1. **EIGRP(Enhanced Interior Gateway Routing Protocol):** IGRP의 CIDR, VLSM 지원 버전. 수렴속도가 빠르다. 부분 갱신을 지원하기 때문에 적은 대역폭을 필요로 한다. VLSM을 지원하며 DUAL 알고리즘(Diffusing-Update Algorithm)을 이용해 네트워크 구성에 변화가 생겼을 때 패킷 손실없이 재라우팅 할 수 있다.
2. **OSPF(Open Shortest Path First):** RIP와 같은 인트라 도메인 라우팅 프로토콜이다. 링크 상태 라우팅 프로토콜을 기반으로 한다. OSPF는 개방형 프로토콜로 사양이 공개되어 있다.

네트워크에 변화가 생길 시 전체 네트워크에 플러딩 과정을 수행해 라우팅 정보를 신속하게 갱신하여 수렴시간과 라우팅 트래픽의 양을 줄인다. 다익스트라 알고리즘을 사용해 최적경로를 계산한다. 보안을 위해 인증을 지원하고 세 가지 IP주소 형태인 클래스 단위, 서브 단위, 클래스 비사용 주소지정(CIDR)을 지원한다. 매우 큰 AS에서 OSPF를 사용할 때는 라우터를 그룹으로 묶어 계층 토폴로지를 만들 수 있다.

**거리벡터 라우팅 프로토콜:** 인접한 이웃으로부터 망 정보 수집, 비용은 이웃 라우터와의 거리 비용을 더해서 구함, 주기적 라우팅 정보 교환, 느린 수렴시간, 모든 라우팅 테이블 값을 이웃에게 전달, 브로드캐스트

**링크상태 라우팅 프로토콜:** 모든 라우터로부터 망 정보 수집, 최단거리 알고리즘으로 모든 라우터에 대한 비용을 직접 계산, 링크 상태 변화 시만 라우팅 정보 교환, 빠른 수렴시간, 자신에게 직접 연결된 망 정보만 전달, 멀티캐스트

**OSPF의 계층 구조:** 큰 네트워크를 잘 제어하고 관리할 수 있도록 큰 AS가 계층 구조를 가질 수 있다. 연결된 라우터와 네트워크를 영역으로 묶고 영역은 논리적 백본을 이용해 연결된다. 라우터의 브로드캐스팅은 영역 내로 제한되어 영역 외부로는 라우팅 패킷이 전달되지 않는다. 영역 간의 정보전달은 오직 영역 경계 라우터를 통해서만 수행된다.

1. **BGP4(Border Gateway Protocol version 4):** 경계 게이트웨이 프로토콜 버전 4는 현재 인터넷에서 사용하는 유일한 인터 도메인 라우팅 프로토콜이다. 경로 벡터 알고리즘을 기반으로 한다. 인터넷에서 네트워크의 접근성에 대한 정보를 제공하기에 알맞다. RIP과 같이 거리벡터에 기초한 알고리즘이지만 패스 벡터 알고리즘에서는 홉수 대신 AS 번호를 매트릭으로 사용한다.

**라우터 보안**

라우터는 동일한 프로토콜을 사용하는 네트워크 계층을 서로 연결한다. 브리지 기능에 경로 배정표를 따라 다른 네트워크 또는 자신의 네트워크 내의 노드를 결정하는 기능이 추가된 기기이다. 여러 경로 중 가장 효율적인 경로를 선택해 패킷을 보낸다.

**라우터 모드: 유저모드**(테스트, 현재 상태 보기 등 제한된 명령), **Privileged모드**(모든 명령어, configuration file 조정 가능), **Configuration모드**(라우터의 구성 파일을 변경하는 경우 사용 가능), **RXBOOT모드**(라우터 패스워드를 모르거나 라우터 이미지파일(IOS)에 문제가 생긴 경우 이를 복구하기 위해 사용 == ROMMON(ROM Monitor) 모드), **Setup모드**(라우터를 처음 구매해 전원을 키거나 라우터에 구성파일이 없는 경우 라우터가 부팅하면서 자동으로 진입하는 모드)

**+TFTP 서버에 의한 라우터 구성 ->** 미리 만들어 놓은 라우터 구성 파일을 TFTP 서버에 구성했다가 새 라우터로 다운로드해 주는 방식. 이때 다운로드에 Trivial File Transfer Protocol을 사용해서 TFTP 서버라고 한다.

**라우터 모드 변환:** Router#config terminal == conf t

이외 라우터 명령어는 교재 참조, 실습

**22절. 네트워크 장비의 이해**

**랜카드, 허브, 리피터, 브리지, 라우터, 스위치의 장비별 특징, VLAN 개념을 파악한다.**

**랜카드:** PC에서 전송 요구가 발생하면 랜카드로 정보를 일정한 형태로 만들어 보내고 랜카드에서는 이 정보를 버퍼에 저장한다음 네트워크 형태에 맞춰 보낸다. PC와 랜카드를 논리적으로 묶어주는 소프트웨어를 네트워크 드라이버라고 한다.

**리피터**: 신호가 훼손되기 전에 신호를 수신해 원래의 비트 형태로 재생(Regenerate)하고 증폭한다. 그런 뒤 새롭게 생성된 신호를 전송한다. 현재 독립된 하드웨어로서의 리피터는 보기 드물어 졌는데 허브, 스위치, 라우터 등의 장비에 리피터 기능이 들어갔기 때문이다.

**허브:** 물리계층에서만 동작하는 장치이다. 네트워크 내에서 정보를 전달하는 신호는 신호 감쇠가 데이터의 무결성을 훼손하기 전까지 고정된 거리를 이동한다. 이더넷 LAN이 버스형 토폴로지를 사용하던 과거에 동축 케이블의 거리 제한을 극복하기 위해 LAN의 두 세그먼트를 연결할 때 리피터를 사용했다. 그러나 오늘날 이더넷 LAN은 스타형 토폴로지를 사용한다. 스타형 토폴로지에서 리피터는 보통 허브라고 불리는 다중 포트 장치이다. 리피터 역할을 수행하는 동시에 연결 지점으로 작용한다. 허브는 리피터와 마찬가지로 필터링 기능이 없다. 어느 포트로 프레임을 전달해야 할지 결정하는 기능도 없다. 허브나 리피터는 물리 계층 장치이다.

**충돌 도메인:** 매체를 공유하는 **LAN**안에서 여러 장비가 동시에 데이터를 보내면 충돌이 발생한다. 이와 같이 충돌이 발생할 수 있는 영역을 충돌 도메인이라고 한다.

**브리지:** 프레임 인지 능력이 있고 OSI의 데이터 링크 계층인 MAC(Media Access Control)에서 동작. 둘 이상의 네트워크를 상호 연결한다. 네트워크를 확장시키고 네트워크 통신을 격리시키기 위한 목적으로 사용한다. 충돌 도메인을 나누기 위해서도 쓴다. 네트워크를 분산 구성해 보안성을 높일 수 있다. MAC 주소를 기반으로 전송할 포트를 결정한다.

**스위치:** 리피터와 브리지를 결합한 하드웨어 브리지의 내장 회로/인지 기능과 리피터의 증폭 기능을 갖췄다. 컴퓨터와 다른 허브 및 스위치를 연결하는 다중 포트 장비이다. 한 포트에 연결된 장치는 다른 포트에 연결된 장치와 고유한 가상 사설 링크를 통해 통신한다. 트래픽 모니터링 기능도 지원한다. 모니터링 포트 또는 네트워크 트래픽을 모니터링 하는 탭 장비를 통해 패킷을 복제해서 트래픽 분석 장비로 전달할 수도 있다.

**브리지/스위치 기능:** Learning, Flooding: 목적지를 모르면 들어온 포트를 제외한 다른 모든 포트로 뿌린다. Forwarding: 해당 포트로 건네준다. Filtering: 다른 포트로 못 건너가게 막는다. Aging: 나이를 먹는다.

**모니터링 포트:** 스위치 장비의 포트이다. 스위치를 통과하는 모든 패킷의 내용을 복제해서 전달하는 기능을 수행한다. 관리 목적을 위해 스위치가 제공하는 기능. 네트워크 사용량, 응답 시간 등 장비 성능을 관리할 목적으로 사용한다. 공격자가 물리적으로 해당 스위치에 접근 가능하면 모니터링 포트를 자신의 PC와 연결해 스니핑이 가능하다. 시스코 장비에서는 SPAN(Switch Port Analyzer) 포트라고 한다.

**탭 장비:** 네트워크에 전송되는 패킷을 중단없이 모니터링하기 위한 장비. 네트워크 구간에 직접 연결한다.

**스위치의 종류:**

1. **2계층 스위치(데이터 링크):** 모든 포트에 연결된 호스트 MAC 주소를 학습해 스위칭 테이블을 생성, 갱신한다. 스위치는 이 테이블을 기초로 프레임을 전달한다.(MAC 기반)
2. **3 계층 스위치(네트워크):** 하드웨어를 사용한 라우터의 한 형태이다. 라우팅 스위치라고도 부른다. 크로스바 스위치같은 ASIC 스위칭 기술을 이용해 OSI참조모델의 네트워크 계층에서 동작한다.(IP 기반)
3. **4 계층 스위치(전송):** L4 스위치는 포트번호를 기준으로 패킷을 전송한다. 주로 네트워크의 암호화나 애플리케이션 프로토콜에 대한 패킷 필터링에 사용한다.

전송 계층 포트번호를 통해 응용 서비스(HTTP, SMTP, FTP, NFS, Telnet, rlogin)를 구분하고 L4 스위치가 관리하는 서버의 부하에 따라 세그먼트를 적절히 배분한다.

이런 기능을 SLV(Server Load Balancing)이라고 하며 SLB의 기본 구성은 가상 서버 부분과 실 서버 부분으로 나눌 수 있다. 로드 밸런서 기능 – 부하분산 기능, 상태 확인 기능, 세션 유지 기능

**L3, L4 스위치는 L2 스위치에 상위 네트워킹 기능을 추가한 것이다.** **L2 스위치는 스타형 네트워크로 더미 허브가 연결된 시스템이 늘어날수록 패킷 간 충돌 때문에 매우 낮은 속도로 동작하는 문제점을 해결하는 획기적 방안이었다.**

1. **7 계층 스위치:** 세션 계층과 응용 계층의 데이터 영역까지 분석해 응용 세션의 제어 수행이 가능하다. 일반적으로 L4스위치가 0 ~ 1023번까지의 포트를 인식하는데 비해 L7 스위치는 그 이외의 포트번호도 인식 가능하다.
2. **L4 와 L7 스위치의 차이점:** L4는 TCP/UDP 포트 정보를 분석해 해당 패킷이 현재 사용하는 서비스 종류별로 패킷을 처리한다. L7 스위치는 트래픽의 내용(이메일의 문자열, HTTP URL, FTP 파일 및 제목 등), 패턴을 분석해 패킷을 처리한다. 더 높은 수준의 Intelligence를 갖춘 스위치일수록 더 정교한 패킷의 부하 분산(로드 밸런싱) 및 QoS기능 구현이 가능하다.

**스위칭 방법에 따른 분류:** 전송로직, 스위칭 방법에 따라 Store-and-Forwar, Cur-Through, Fragment-Free 방식 등으로 구분한다.

1. **Store-and-Free:** 스위치나 브리지가 일단 들어오는 프레임을 전부 받아들인 다음 처리를 시작, 에러 발생 시 브리지나 스위치는 해당 프레임을 버리고 재전송을 요구해 에러 복구 능력이 뛰어나다.
2. **Cut**-**Through:** Store-and-Forward처럼 프레임이 다 들어올 때까지 기다리지 않고 앞에 들어오는 목적지 주소만 본 다음 바로 목적지로 전송하기에 첫 48비트만 본다. 속도가 빠르지만 프레임에서 에러를 찾아내기 어려워 에러 복구 능력이 약하다.
3. **Fragment**-Free: 위 두 방식의 장점을 결합한 방식이다. 처음 512비트를 확인해 컷스루 방식처럼 목적지로 전송한다.

**라우터:** 3계층 장비로 물리, 데이터 링크, 네트워크 계층에서 동작한다. 이기종 LAN 간의 연결, LAN을 WAN에 연결, 라우팅 기능, 에러 패킷에 대한 폐기 등의 기능을 수행한다.

라우터와 리피터, 교환기 사에어는 세 가지 큰 차이점이 있다.

1. 라우터는 각 인터페이스를 위한 물리주소와 논리주소를 가진다.
2. 라우터는 링크 계층 목적지 주소가 패킷이 도착한 인터페이스 주소와 일치하는 패킷만 처리한다.
3. 라우터는 패킷을 전달할 때 해당 패킷의 링크 계층의 주소(발신지, 목적지 모두)를 변경한다.

**라우터의 특징:** 네트워크 세그먼트 내부에서 발생하는 브로드캐스팅과 멀티캐스팅 패킷을 모두 차단해 다른 네트워크 세그먼트로 전달되는 것을 방지한다. 접근통제목록(ACL; Access Control Lists)에 기반을 둬 트래픽을 필터하고 필요시 패킷을 분할할 수 있다. 라우팅 프로토콜을 통해 경로와 네트워크에서 발생하는 변경 정보를 발견한다. 라우팅 프로토콜들을 링크가 다운되었는지, 특정 경로가 혼잡한지, 다른 경로가 더 경제적인지 등을 알려준다.

|  |  |
| --- | --- |
| **브리지** | **라우터** |
| 헤더 읽기만 가능, 변경 불가 | 각 프레임에 새 헤더 생성 |
| MAC주소에 근거, 전송 테이블 작성 | IP주소에 근거, 라우팅 테이블 작성 |
| 모든 포트에 동일한 네트워크 주소 | 포트마다 서로 다른 네트워크 주소 |
| MAC 주소 기반 트래픽 필터링 | IP주소 기반 트래픽 필터링 |
| 브로드캐스트 패킷 전달 | 브로드캐스트 지원 X |
| 브리지에게 알려지지 않은 목적지 주소를 가진 트래픽도 전달 | 라우터에게 알려지지 않은 목적지 주소를 가진 트래픽은 전달하지 않음 |

**게이트웨이:** 두 개의 서로 다른 환경을 연결하는 하드웨어에서 실행되는 소프트웨어를 말한다. 한 환경이 다른 언어를 말하는 경우 번역을 하고 상호작용을 제한하는 역할도 수행한다. 모든 계층을 포함하는 네트워크 장비, 여러 프로토콜 변환기능을 수행해 네트워크 병목 현상을 일으키는 지점이 될 수 있다.

**계층별 장비 분류표**

|  |  |  |
| --- | --- | --- |
| **계층** | **장비 종류** | **기능** |
| 응용 | 게이트웨이 | 다른 유형의 네트워크 연결  프로토콜 및 포맷 해석 |
| L7 스위치 |  |
| 전송 | L4 스위치 |  |
| 네트워크 | 라우터 | 여러 LAN을 분리하고 연결해 인터네트워크 생성,  IP주소에 기반한 필터링 |
| 데이터링크 | 스위치 | 통신 장치들 사이에 가상 사설 링크를 제공해 VLAN이 가능하게 한다,  충돌 감소, 스니핑 저지 |
| 브리지 | MAC주소에 기반한 패킷 전송과 필터링, 브로드캐스트 패킷 전송, 충돌 트래픽은 전송x |
| 물리 | 리피터 | 신호 증폭해 네트워크 확장 |
| 허브 | 집중화 장치, 단순히 노드 연결 |

**VLAN의 구성과 관리**

**VLAN은 Virtual Local Area Network이다.** 가상 근거리 네트워크는 물리적 선이 아닌 소프트웨어로 구성한 근거리 네트워크이다. 이때 LAN을 물리적 세그먼트가 아닌 논리적 세그먼트로 분할한다. 이 논리적 LAN이 VLAN이다.

**VLAN의 특징:** 데이터링크 계층에서 브로드캐스트 도메인을 나누기 위해 사용한다. 스위치는 VLAN 태그가 상이한 네트워크로의 접근을 근본적으로 차단해 보안을 유지한다. VLAN은 관리자가 서로 다른 논리적 그룹에 대해 서로 다른 보안 정책을 적용할 수 있게 한다.

**VLAN 종류:**

1. **Port 기반 VLAN:** 포트기반 VNAL은 스위치 포트를 각 VLAN에 할당하는 것으로 VLAN에 속한 포트에 연결된 호스트들 간에만 통신이 가능하다. 가장 일반적인 VLAN
2. **MAC 기반 VLAN:** 맥어드레스 VLAN은 각 호스트의 맥어드레스를 VLAN에 등록해 같은 VLAN에 속한 맥어드레스들 간에만 통신되게 하는 방법이다. VLAN은 각 호스트의 맥어드레스를 전부 등록해야 하기 때문에 많이 쓰지는 않는다.
3. **네트워크주소 기반 VLAN:** 네트워크 주소별로 VLAN을 구성해 같은 네트워크에 속한 호스트들 간에만 통신하도록 구성한 VLAN이다. Ip네트워크 VLAN을 사용한다.
4. **프로토콜기반 VLAN:** 같은 통신 프로토콜(TCP/IP, IPX/SPX, NETVIEW 등)을 가진 호스트들 간에만 통신이 가능하도록 구성된 VLAN이다.
5. **멀티캐스트 IP 기반 VLAN:** 일부 VLAN제조사는 멤버십 자격 특징으로 멀티캐스트 IP주소를 사용한다.
6. **조합:** 관리자가 소프트웨어를 설치할 때 위 VLAN 중 여러 개를 조합해 사용하기도 한다.

**VLAN의 장점:** 경비와 시간 절약, 가상 워크그룹의 생성, 보안

**23절. 무선통신 보안**

**중요: 무선 표준, 무선랜의 기술적/물리적/관리적 보안, \*기술적 보안\***

|  |  |  |
| --- | --- | --- |
| **무선 네트워크 구조** | **내용** | **사용 예제** |
| WPAN | 단거리 Ad Hoc 또는 P2P 방식 | 블루투스 페어링 |
| WLAN | 유선랜의 확장개념 or 유선랜 설치가 어려운 지역으로 네트워크 제공 | 임시 사무실과 같은 환경에서 유선랜 구축으로 인해 발생하는 불필요 비용을 줄임 |
| WMAN | 대도시와 같은 넓은 지역을 대상, 높은 전송속도 | 대학 캠퍼스같은 넓은 지역내 건물간의 무선 연결 제공 |

**무선 네트워크 보안 위협의 주요 요소**

1. **채널:** 무선 네트워크는 일반적으로 브로드캐스팅 통신을 한다. 이 경우 유선 네트워크보다 도청이나 재밍에 훨씬 취약하다.
2. **이동성:** 휴대가 간편하고 이동이 수월해 문제가 생긴다.
3. **자원:** 무선장치는 메모리와 프로세싱 자원이 제한적이어서 DoS 공격이나 악성 코드 위협에 대처하기가 쉽지 않다.
4. **접근성:** 일부 무선 장치는 원격지에 있어 물리적 공격에 매우 취약할 수 있다.

**무선랜의 특성**

1. **감쇠:** 신호는 방사형으로 퍼지기 때문에 강도가 급감한다. 오직 신호의 한 부분만 수신기에 도달한다.
2. **간섭:** 수신자가 의도된 송신자로부터의 신호를 받을 뿐 아니라 같은 주파수 대역을 사용하고 있는 다른 송신자의 신호도 받을 수 있다.
3. **다중경로 전달:** 전기적 파동은 벽이나 땅 또는 사물과 같은 장애물에 반사될 수 있다. 한 수신자는 같은 송신자로부터 하나 이상의 신호를 받을 수 있다. 그래서 수신자는 다른 위상에 있는 신호들을 받을 수 있다.
4. **오류:** 위와 같은 이유로 무선 네트워크의 오류와 오류감지는 유선보다 더 심각한 문제이다.

**무선랜 접근제어:** CSMA/CD 알고리즘은 세 이유로 무선 LAN에서 동작하지 않는다.

1. 충돌감지를 위해 호스트는 송수신을 동시에 해야 한다.(프레임의 송신과 충돌 신호의 수신) 이는 호스트가 양방향 모드로 동작해야 함을 의미한다.
2. 어떤 장애물이나 범위 문제 때문에, 또 한 지국이 다른 지국의 전송을 인지하지 못하는 숨겨진 지국 문제 때문에, 충돌이 발생해도 감지되지 않을 수 있다.
3. 기지국들 간의 거리가 멀 경우 거리로 인한 신호 감쇠는 다른 측에서 발생하는 충돌을 감지하지 못하게 할 수도 있다.
4. 위 세 문제를 극복하기 위해, 반송파감지 다중접근 충돌회피(CSMA/CA, Carrier Sense Multiple Access with Collision Avoidance)가 무선 LAN을 위해 사용된다.

**무선표준:** 컴퓨터 통신에서 2.4GHz 통신망을 이용, 표준 규약으로 802.11을 사용한다. 802.11a, 802.11b, 802.11g, 802.11n 등이 있다. 초기 11Mbps인 802.11b를 사용하는 무선랜은 점차 사라지고 54Mbps를 지원하는 802.11g와 600Mbps를 지원하는 802.11n으로 발전중

* **IEEE 802.11b:** 공공장소에서 많이 사용, 주파수 면허를 받을 필요 없는 대역인 2.4GHz대를 사용, 최대 전송속도 11Mbps
* **IEEE 802.**11a: 5GHz 주파수대 사용, 최대 통신속도 54Mbps, b와 호환되지 않지만 b와 마찬가지로 무선랜 전용대여서 블루투스의 영향을 받지 않는다.
* **IEEE 802.**11g: b와 같은 주파수대의 고속통신 무선랜. B와 호환된다. 표준 최대 24Mbps, 옵션으로 54Mbps까지 지원된다.
* **IEEE 802.**11i: WPA-1과 WPA-2 규격 포함. 암호화 방식에 따라 분류된 것으로 WPA-1은 TKIP(Temporal Key Integrity Protocol)을, WPA-2는 CCMP 암호화 방식을 사용한다. 인증방식은 사용 모드에 따라 개인과 기업용으로 구분된다. 개인은 PSK 모드로 무선 AP와 단말이 사전에 공유한 비밀키를 가지고 있는지 4-way handshake로 확인한다. 기업용은 802.1x/EAP 모드로 RADIUS인증서버를 이용해 상호인증을 수행하고 결과를 기반으로 AP접근을 허용한다.
* **IEEE 802.**11n: 여러 개의 안테나를 사용하는 다중입력 다중출력(MIMO) 직교 주파수 분할 다중화 방식을 사용, 최대 600Mbps 지원하는 무선 LAN 표준

**1997 -** 802.11(최초의 무선랜 프로토콜) -> **1999 -** 802.11b(2.4Ghz/11Mbps WEP 보안 구현), 802.11a(5GHz/54Mbps 전파 투과/회절성 떨어짐, 통신단절 심함, b와 호환 불가) -> **2003 –** 802.11g(2.4/54 b에 a의 속도를 추가한 프로토콜, b와 호환되지만 네트워크 공유시 데이터 처리 효율이 떨어짐) -> **2004 –** 802.11i(2.4/11 b표준에 보안성만 강화) -> **2009 –** 802.11n(5GHz/2.4GHz 최대 600Mbps 속도, 다중 안테나 사용 다중 입출력(MMO) 기술과 대역폭 손실 최소화)

**무선랜 주요 구성요소**

**무선AP(Access Point):** 기존 유선랜의 가장 마지막에 위치해 무선랜 접속에 관여한다.

**대규모AP**: 기업의 비즈니스를 위한 인프라로 사용, 소규모AP보다 로밍과 보안기능에 중점

**소규모AP**: 가정이나 사무실 등 작은 규모의 무선 네트워크 구축에 사용, IP공유기능, VPN 등 다양한 기능 탑재에 저렴

**무선 브리지:** 2개 이상의 무선랜 연결. 물리적으로 떨어진 2개의 무선랜에 의해 각각 위치해 동작. 2개의 브리지 사이에 전파의 전송을 방해햐는 물체가 존재하지 않아야 한다.

**다양한 무선통신 기술**

**블루투스:** 짧은 거리에 있는 하드웨어를 연결하는 무선 LAN 기술. 1998년, 노키아 IBM 인텔 등이 협업해 결성한 블루투스 스페셜 그룹에서 공동 개발한 근거리 무선 네트워크

**블루투스 구조:**

1. **피코넷(Piconet):** 피코넷 또는 작은 네트워크라고 불린다. 8개까지 지국을 가지며 하나는 주국(Primary)라고 하고 나머지는 종국(Secondary)라고 부른다.
2. **스캐터넷(Scatternet):** 피코넷이 합쳐져 스캐터 넷이 된다. 한 피코넷의 종국은 다른 피코넷의 주국이 될 수 있다.

**블루투스 특징:** 낮은 가격에 만족할 만한 서비스, 접속기기 필요없고 동시 사용 가능, 보안되지 않은 데이터 전송시 어떤 장비라도 데이터를 캡처할 수 있어 보안 위협에 노출된다.

**보안 취약점:** 블루프린팅(블루투스 공격 장치의 검색 활동), 블루스나핑(취약점을 이용, 장비의 임의 파일에 접근), 블루버깅(블루투스 장비 간 취약한 연결고리 공격, 공격 장치와 공격 대상을 연결해 대상 장치에서 임의 동작 실행), 블루재킹(블루투스를 이용해 스팸처럼 명함을 익명으로 퍼트린다.)

**무선랜 보안**

**무선랜의 물리적 취약점:** 무선AP는 외부에 노출된 형태로 위치해 다음과 같은 취약점이 발생한다.

|  |  |
| --- | --- |
| 유형 | 내용 |
| 도난 및 파손 | . |
| 구성설정 초기화 | 리셋버튼을 통한 장비 초기화 |
| 전원 차단 | . |
| LAN 차단 | 무선 AP에 연결된 케이블 절체로 장애발생 |

**무선랜의 기술적 취약점:** 도청, 서비스 거부(DoS), 불법AP(Rogue AP)

**무선 암호화 방식:**

WEP(Wired Equivalent Privacy): 전송되는 MAC 프레임을 40비트의 WEP 공유 비밀키와 임의로 선택되는 24비트의 Initialization Vactor(IV)로 조합된 총 64의 키를 이용한 RC4 스트림 암호화로 보호, 무선 클라이언트와 AP는 동일 패스워드 문장에서 4개의 고정 장기 공유키를 생성한 후 이들 중 하나를 선택해 암호 및 인증에 활용한다.

WPA/WPA2: WEP보다 권장되는 방식. 가능한 긴 길이의 비밀키 사용, 추가적 인증 서버 운영

**비인가 접근:**

1. **SSID 노출:** 무선랜을 구분하기 위한 32byte의 이름이다. 장비에서 Wi-Fi를 활성화했을 때 나오는 무선랜 목록이 SSID이다. 무선 AP에 별도의 암호화나 인증절차가 없을 경우 무선 데이터를 모니터링해 SSID를 알아내기만 해도 인가되지 않은 접속이 가능하다.
2. **MAC 주소 노출:** 접근제어를 위해 MAC주소 필터링을 적용하기도 한다. 무선랜 카드에 부여된 MAC 주소 값을 이용해 무선랜 서비스의 접속을 제한하는데 활용할 수 있다. 공격자가 사용자의 MAC주소를 도용해 쉽게 무력화할 수 있다.

**무선랜의 관리적 취약점:** 장비관리 미흡, 사용자의 보안의식 결여, 전파관리 미흡

**무선랜 인증 기술**

**SSID 설정을 통한 접속제한:** 일부 AP에서 SSID 값을 NULL로 설정해 접속을 시도하는 사용자의 연결요청 메시지에 대해 접속을 차단하는 기능을 제공한다. 이렇게 운영하는 것을 폐쇄시스템을 구성해 운영한다고 한다.

**폐쇄시스템 운영:** SSID의 브로드캐스트를 막는 것을 SSID Cloaking이라고 한다. SSID숨김으로 설정하고 폐쇄시스템을 운영하면 SSID를 모르는 사용자의 접속 시도가 줄고, SSID를 NULL로 인가되지 않은 접속 시도를 차단할 수 있다. 또 무선랜 분석도구인 NetStumbler 등을 이용한 스푸핑을 미연에 방지할 수 있다. 구현이 쉽고 운영을 위한 추가적 노력도 필요없다.

**MAC주소 인증:** 접속을 허용하는 단말기가 랜카드의 MAC주소를 사전에 등록하고 접속을 요청하는 단말기의 MAC주소가 사전에 등록한 리스트에 존재하는지 대조해 인증한다.(네트워크 카드사에서 부여한 48비트 H/W주소) 무선랜 서비스를 미리 인가된 사용자만 접근할 수 있도록 제공하는 점이 장점이지만 대규모 기관에서 이용할 경우 사용자의 MAC주소를 관리하기 위한 관리 업무가 많아진다는 단점이 있다.

**WEP 인증 메커니즘**

1. **클 -> 인증 요청 -> AP**
2. **AP -> 인증 IV 생성 후 인증 문자열 전송(Challenge) -> 클**
3. **클 -> RC4키로 인증용 IV암호화 후 암호화 문자열 전송 -> AP**
4. **AP -> 인증용 IV 암호문 복호화 확인 후 성공/실패 여부 통보 -> 클**

**취약점:** WEP를 적용하기만 하면 사용자 인증과 데이터 암호에도 모두 적용할 수 있다. 무선랜 장비에서 WEP 구현은 매우 간단하고 인증 절차가 간결해 사용자에게 많은 편의를 제공한다. 그러나 단방향 인증방식 제공으로 인한 취약점이 있다. 복제 AP에 이러한 취약점이 노출된다. 무선랜 서비스 사용자가 무선랜 서비스를 제공하는 AP에 대한 인증을 하지 않았기 때문에 이런 일이 발생한다. 또 WEP 키값을 하나의 고정된 공유키로 사용하기 때문에 외부로 키값이 유출 시 같은 키값으로 무선랜을 사용하는 모든 장비가 위험에 노출된다.

**동적 WEP 적용:** 인증 서버를 설치해 운영시 동적 WEP를 사용할 수 있다. 현재 AP가 동적 WEP를 지원하는지 여부부터 확인해야 한다. AP에서 802.1x 프로토콜을 지원하는 경우 동적 WEP를 사용할 수 있다.

|  |  |  |
| --- | --- | --- |
| 유형 | 상세 내용 | 비고 |
| Static WEP | RC4 암호화 알고리즘은 일정량 데이터를 모으면 암호화 값을 역산할 수 있는 취약점이 존재한다. | 64비트, 128비트 암호화키 지원 |
| Dynamic WEP | RC4는 그대로 사용, 키값을 주기적으로 변경.  802.1x에서 암호화를 위한 기법으로 채택,  RC4 알고리즘을 그대로 사용해 프로세싱 파워가 강화되면 암호화 데이터의 해석이 가능할 수 있음 | 키생성을 위한 시드 값 |

**EAP 인증 메커니즘: Extensible Authentication Protocol.** RFC2284에 공식 명세가 발표. 초기에 PPP(Point to Point Protocol)에서 사용하기 위해 개발되었으나 현재는 무선랜 표준인 IEEE 802.1x에서 사용자 인증 방법으로 사용한다. 802.1x 시스템은 인증자(브리지 혹은 AP), 서플리컨트(사용자 단말), 인증 서버(RADIUS 서버)로 역할을 나눈다. 목적은 서플리컨트로 하여금 인증자로부터 물리적인 포트 사용권을 획득하는 절차를 제공하는 것이다. 인증 서버는 인증 서비스를 제공한다.

EAP는 어떤 링크에도 접속이 가능한 단순한 캡슐화 개념의 프로토콜이다. 모든 링크에 적용될 수 있으며 다양한 인증 방법을 사용할 수 있도록 설계되었다. 인증 서버와 무선장비 사이에 상호 인증을 허용해 사용자가 패스워드, 토큰, OTP패스워드, 인증서, 스마트카드 또는 커버로스를 사용함으로써 유연성을 제공한다. 802.11i와 호환되는 무선 장비와 인증 서버는 다양한 옵션을 제공한다. 따라서 802.1x에 플러그인 되는 서로 다른 인증 모듈을 가진다. 802.1x는 네트워크 관리자가 서로 다른 EAP모듈을 추가할 수 있는 프레임워크를 제공한다.

**무선랜 암호화 기술**

**WEP(Wired Equivalent Privacy):** 무선랜과 연결된 AP와 무선 단말기 간에 주고받는 데이터를 약속한 공유 비밀키와 임의 IV(Initial Vector)값을 조합한 64비트 or 128비트의 키를 이용해 전송 데이터를 암호화하는 방식이다. 공유키를 이용한 사용자 인증 기능도 제공한다. 서로 같은 공유키를 갖는 사람을 정상 사용자로 인증하는 방식을 채택하고 있다.

**WEP의 데이터 암호화 절차:** 공유키와 난수 발생기를 이용해 키스트림을 생성한다 -> 생성된 키스트림과 전송하고자 하는 평군과의 XOR 연산을 통해 암호문을 생성한다. 여기에 더해 데이터의 무결성 보장을 위해 CRC-32(Cycle Redunduancy Check 중복순환검사, 송신전 계산과 수신 후 계산 대조해 손상 파악) 알고리즘으로 구성되는 ICV(Integrity Check Value)를 사용한다. ICV는 전송 도중에 발생하는 사용자 데이터를 보호해 무결성을 제공한다.

**WEP의 복호화 방식:** 송신자가 보낸 패킷에서 초기 벡터값을 추출하고 추출한 초기 벡터값과 자신이 갖고 있는 WEP키를 암호키로 난수발생기에서 난수를 생성해 키스트림을 구성한다. 구성된 키스트림과 암호문을 이용해 XOR연산을 통해 복호화 시킨다. 복호화된 평문의 데이터와 ICV를 이용해 무결성을 체크한다. 무결성 체크를 위해 CRC-32를 통해 ICV’를 구한 후 ICV’과 송신자로부터 전송받은 ICV를 비교해 전송 도중 패킷 데이터가 변경되었는지 여부를 체크한다. 이렇게 무결성 체크가 성공적으로 이뤄지면 평문의 데이터를 최종적으로 수신한다.

**WEP 암호의 취약점:** 암호 메커니즘의 취약점과 WEP 키 관리상의 취약점이 있다.

**WEP암호 메커니즘의 취약점**: WEP 키를 이용한 무선랜 암호화 통신의 보안성은 높지 않다. 통신 과정에서 IV는 무작위로 생성되어 암호화 키에 대한 복호화를 어렵게 하지만 24비트의 IV는 24비트의 짧은 길이로 인해 반복되어 사용되기 때문이다.

**WEP키 관리 취약점**: WEP는 고정키 값을 사용해 외부 유출의 위험성이 상당하다. 뿐만 아니라 WEP은 일정시간 패킷을 수집 분석하는 공격자에 의해 키 값이 크랙 될 수 있으므로 키 관리에 항상 유의해야 한다. 고정키 값을 사용하는 WEP의 키 관리에 관한 취약점을 줄이기 위해 동적 WEP을 적용해 사용할 수 있다. 하지만 동적 WEP을 사용해도 WEP 프로토콜 자체가 갖는 암호학적 취약점을 보완할 수는 없다.

**TKIP(Temporal Key Integrity Protocol):** IEEE 802.11i 작업 그룹과 Wi-Fi Alliance에 의해 개발. TKIP의 목적은 하드웨어 교체 없이 WEP의 강도를 향상시키거나 완전히 대체하는 것. WEP을 제공하는 무선랜 장비의 펌웨어 업그레이드나 소프트웨어 업그레이드를 통해 사용자 레벨의 보안을 강화시킨다. 기존 WEP의 암호화 알고리즘인 RC4를 이용하면서 RC4의 보안상 문제를 개선하기 위해 Key Mixing 함수, Dynamic WEP Key(Temporal Key), 메시지 무결성 보장을 위한 스펙을 정의한 통신규약이다.

**WEP의 취약점을 보완하는 TKIP의 보안기술:** 48비트의 확장된 초기벡터 사용, 초기벡터 값인 IV의 순차적 증가 규칙을 보완해 재생 공격으로부터 보호, 메시지 무결성 체크를 위해 WEP에서 무결성 보장을 위해 적용되었던 CRC-32 알고리즘보다 안전한 MIC(Message Integrity Check)를 사용할 수 있도록 제공한다.

**TKIP의 취약점:** 키 관리 방법의 부재, 무선 패킷 수집 분석을 통한 키 크랙공격의 가능성, WEP의 RC4 알고리즘이 가지고 있던 기본적 취약점

**CCMP(Counter Mode with CBC-MAC Protocol):** CCMP는AES 블록 암호를 사용해 데이터의 비밀성과 무결성을 보장하기 위한 규칙을 정의한다. TKIP가 기존 하드웨어를 수용하기 위한 과도기적 방법이라면 CCMP는 기존 하드웨어를 고려하지 않고 초기부터 보안성을 고려해 새롭게 설계된 프로토콜이다. 128비트의 대칭키를 사용하고 48비트의 초기벡터를 이용한다. AES가 제공하는 여러 모드 중 CCMP는 Counter 모드 기반으로 CBC MAC(Cipher Block Chaining Message Authentication Code)를 결합한 CCM을 기반으로 한다. 패킷의 데이터 영역과 IEEE 802.11 헤더의 무결성을 보장한다. 또한 CCMP가 사용하는 PN(Packet Number)은 패킷의 재연을 방지할 수 있다.

**무선랜 인증 및 암호화 복합 기술**

**WPA: Wi-Fi**에서 정의한 무선랜 보안 규격이다. 802.11i의 보안 규격을 일부 수용해 만든 표준 규격이다. 현재 사용되는 하드웨어의 변경 없이 소프트웨어 업그레이드를 통해 지원 가능하다. WPA(Wi-Fi Protected Access)의 주요 기술로 암호화 기법인 TKIP가 있다. 패킷당 키 할당, 메시지 무결성, 키값 재설정 등을 통해 WEP의 약점을 보완했다. 802.1x와 EAP를 기반으로 강력한 사용자 인증을 제공하기도 한다. WPA는 현재 많이 사용되는 RADIUS, Kerberos, 기타 다른 인증서버와 호환되며 AP에서 비밀번호를 수동으로 설정할 수 있는 사전 공유 키 방식도 제공한다.

암호화 기술 -> TKIP, 사용자 인증 및 키 관리 -> 802.1x의 Dynamic WEP, EAP 인증 사용

**WPA2:** 2세대 WPA이다. TKIP를 대체하기 위해 AES에 기반한 CCMP(Counter Mode with Cipher Block Chaining Message Authentication Code Protocol) 암호화 방식을 사용한다. IEEE 802.11i 수정안을 포함한 보안 기술이다. WPA2는 RSN(Robust Security Network)라고 부른다.

**WEP, WAP, WAP2 비교:** WEP의 보안 취약점이 드러난 이후 WPA/WPA2가 대안으로 나왔다. WPA는 와이파이 보호접속이라고도 한다. 보안 강화를 위해 MIC(Michael)라는 8바이트 메시지 무결성 코드를 추가했고 IV생성에서 새로운 배열 규칙을 적용했다. 또한 IV와 키값과의 상관 관계를 줄이기 위한 패킷 키 믹싱 기능을 추가했다.

**WPA-PSK(Wi-Fi Protected Access Pre-Shared Key):** 인증 서버가 설치되지 않은 소규모 망에서 사용하는 인증방식. 무선 AP와 단말기가 동일한 공유키(PSK)를 가지고 있는지 802.1x에 규정된 EAPoL-Key 프레임을 활용해 4way-handshake를 통해 확인하여 인증한다. WEP처럼 AP와 통신해야 할 클라이언트에 암호화키를 기본으로 등록해두고 있다. 하지만 암호화키를 이용해 128비트인 통신용 암호화키를 새로 생성하고, 이 암호화키를 10,000개 패킷마다 바꾼다.

**WAP-PSK 인증의 취약점:** 접속/인증 패스워드를 짧게 하거나 쉬운 값으로 설정할 경우 사전 공격을 통해 손쉽게 크래킹할 수 있다.

**802.1x/EAP(Extensible Authentication Protocol)보안**: WPA Enterprise 방식이다. WPA에서 사용자 인증 영역까지 보완한 방식. 유선 랜 환경에서 포트 기반 인증 표준으로 사용되는 IEEE 802.1x 표준과 함께, 다양한 인증 메커니즘을 수용할 수 있도록 IETF의 EAP인증 프로토콜을 채택했다.

**802.1x/EAP의 추가 기능:**

1. 사용자에 대한 인증 수행
2. 사용 권한을 중앙 관리
3. 인증서, 스마트카드 등 다양한 인증 제공
4. 세션별 암호화키 제공

**WEP – WPA-PSK – 802.1x/EAP의 차이:** WEP와 WPA-PSK는 아이디와 패스워드를 통한 인증, WEP 또는 WPA-PSK는 미리 양쪽에서 설정한 암호화키 사용. 802.1x/EAP는 세션(무선랜 연결) 별로 재사용이 불가능한 암호화키 사용으로 복호화 가능성을 무력화시킴

**802.1x/EAP와 RADIUS 서버를 이용한 무선랜 인증 절차**

1. 클라이언트가 AP에 접속 요청(암호화x), AP는 클라이언트와 내부 네트워크의 연결 차단
2. RADIUS 서버가 클라이언트에 인증 Challenge 전송
3. 클라이언트 Challenge에 대한 응답으로 전송받은 Challenge값, 계정, 패스워드에 대한 해시값을 구해 RADIUS서버에 전송
4. RADIUS 서버는 사용자 관리 DB정보에서 해당 계정의 패스워드 확인. 그리고 연결 생성을 위해 최초 전송한 Challenge의 해시값을 구해 클라이언트로부터 전송받은 해시값과 비교
5. 해시값이 일치하면 암호화키 생성
6. 생성한 암호화키를 클라이언트에게 전달
7. 전달받은 암호화키를 통해 암호화 통신 수행

**WAR-Driving:** 공격자가 AP를 식별해 침입하려는 의도로 무선장비를 갖고 배회하는 것. 랩톱을 사용해 WLAN이 구축된 빌딩에 차로 근접해 이루어지는 경우가 많다. 최근에는 PDA도 공격을 위해 사용된다.

**WAP(Wireless Application Protocol)**

**WAP(무선응용프로토콜):** WAP 포럼에서 개발한 통합 표준. 웹서비스에 접근하기 위해 폰과 PDA같은 장비에서 사용한다. 기존의 무선 네트워크 기술(GSM, CDMA, TDMA) 및 IP, XML, HTML, HTTP 등의 인터넷 기술과 가능한한 호환할 수 있게 설계되었다. 보안 기능도 포함한다.

**WAP규격:** WWW 프로그래밍 모델에 기반한 프로그래밍 모델, XML과 무선 마크업 언어(WML), 모바일 무선 단말기에 적합한 소형 브라우저 규격, 경량 통신의 프로토콜 스택, 무선 전화 응용(WTA)을 위한 프레임워크

**WAP 운용:** WAP 프로그래밍 모델은 클라이언트, 게이트웨이, 오리지널 서버로 이루어진다. 게이트웨이와 오리지널 서버 사이 트래픽은 HTTP를 이용한다. 무선 도메인에서는 게이트웨이가 프록시 서버로도 사용된다. 모바일 무선 장비의 부족한 연산 능력을 보완할 때도 게이트웨이 프로세서를 사용한다.

WAP장치 -> WTLS 보호영역(무선 네트워크) -> WAP 게이트웨이 -> TSL 보호영역(유선 네트워크) -> 웹 서버

**무선 마크업 언어(WML, Wireless Markup Language):** 제한된 통신 용량, 제한된 화면 크기, 제한된 사용자 입력 기능을 가진 장비에서 콘텐츠와 양식을 표현하기 위해 설계되었다. 키패드, 스타일러스 펜, 기타 모바일 장비에서 흔히 사용되는 입력에 동작하도록 설계

**무선 응용 환경(WAE, Wireless Application Environment):** 모바일 전화, PDA같은 무선 장비의 응용 프레임워크를 규격화, WAP가 지원하는 응용 프로그램과 장비 개발을 쉽게 하기 위한 도구와 형식의 집합

**WAP 프로토콜 구조:** 무선 프로토콜 스택은 응용 – 세션 – 트랜잭션 – 데이터그램 – 트랜스포트로 나뉜다.

**WLTS(무선 전송 계층 보안):** 모바일 장비와 WAP 게이트웨이 간의 보안 서비스를 제공한다. SSL(Secure Socket Layer)을 보완해 구성한 산업표준 전송 계층보안(TLS; Transport Layer Security) 프로토콜에 기반한다. 종단 – 대 – 종단 보안을 제공하기 위해 게이트웨이 – 클라이언트 사이에는 WTLS를 사용하고 게이트웨이 – 목적지 서버 사이에는 TLS를 사용한다. WAP은 WAP 게이트웨이 내에서 WTLS와 TLS 사이의 변환을 담당한다. WTLS에는 다음과 같은 기능이 있다.

1. **데이터 무결성(Data Integrity):** 메시지 인증을 통해 클라이언트와 게이트웨이 사이의 데이터가 변조되지 않았음을 보장
2. **프라이버시:** 암호화를 통해 제3자가 데이터를 읽을 수 없는 걸 보장
3. **인증:** 디지털 인증서를 통해 상호 인증 제공
4. **서비스 거부 공격에 대한 보호:** 재전송된 메시지나 확인할 수 없는 메시지를 탐지하고 제거

**WTLS 프로토콜 구조:** 두 개의 계층으로 구성된다. 다양한 상위계층 프로토콜에 보안 서비스를 제공한다. HTTP도 WTLS상에서 작동할 수 있다. 세 가지 상위 프로토콜이 정의되어 있는데, 핸드셰이크, 암호명세 변경, 경고 프로토콜이 있다.

**WAP2(종단-대-종단 보안):** 게이트웨이는 웹 서버와 보안 SSL 또는 TLS 세션을 확립하지만 정작 게이트웨이 내부에서 데이터는 암호화되지 않은 상태로 다뤄진다. 따라서 게이트웨이가 보안 위협이 될 수 있으므로 WAP 포럼은 WAP 버전2 구조 문서에서 몇 가지 종단-대-종단 보안 방법을 정의한다.

**디바이스 인증기술(기기 인증)**

**기기인증 도입 배경:** 기기 사양의 고도화, 네트워크 환경의 변화, 서비스의 다양화, 해킹 수법의 발전

**기기인증 도입의 장점:** 보안성, 경제성, 상호연동성

**기기인증 기술:**

1. **아이디/패스워드 기반 인증:**

* **SSID**
* **AP간 WEP키 이용**
* **서버 간 접근제어 시 아이디/패스워드 이용**

1. **MAC 주소값 인증**
2. **암호 프로토콜을 활용한 인증**

* **802.1x 인증:** 유무선 네트워크에 인증된 접속을 허용하는 IEEE 표준. 중앙 사용자 ID, 인증, 동적 키 관리 및 계정 지원(포트에 기반을 둬 액세스 제어, PAP, CHAP, RADIUS, PEAP, WEP 등의 프로토콜 사용)
* **802.11i 인증:** RC4 기반 WEP 기술의 보안 취약점 개선을 위해 IEEE 802.11i 제정

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **구분** | **정적 WEP Key** | **동적 WEP Key** | **WPA v1** | **WPA v2** |
| **보안키 적용 방식** | **WEP(24 비트 IV)** | **WEP(24 비트 IV)** | **TKIP(48비트 IV)** | **CCMP** |
| **암호화 알고리즘** | **RC4** | **RC4** | **RC4** | **AES** |
| **암호 비트** | **64/128** | **128** | **128** | **128** |
| **보안 레벨** | **하** | **중/상** | **상** | **최상** |

1. **Challenge/Response 인증:** OTP와 유사하게 해시값 생성해 사용자 인증

* **Challenge:** 네트워크 접속 서버나 인증 서버가 랜덤값을 생성해 클라이언트에 전송
* **반환:** 클라이언트는 수신한 challenge와 패스워드에 해시 알고리즘을 적용해 반환
* **인증 완료:** 응답을 받은 서버가 클라이언트와 동일한 해시 단계를 거쳐 클라이언트의 결과값과 값이 일치하면 인증 완료

**각 기기인증 기술별 장/단점 파악하기**

**RFID(Radio-Frequency Identification)**

무선 주파수를 이용해 ID를 식별하는 방식이다. 동작 방법에 따라 수동형과 능동형으로 나눈다.

**RFID 네트워크 공격유형:**

1. **도청:** RFID는 바코드와 달리 수 미터 이내에서도 리더기와 태그간 통신이 가능하다. 공격자가 리더기를 갖고 태그를 스캔하는 적극적 공격, 리더기와 태그 간 통신을 도청하는 수동적 공격이 있다.
2. **트래픽 분석:** 리더기와 태그의 통신 중 트래픽 분석을 통한 공격. 내용이 암호화되어 있어도 비트의 패턴으로 특정 정보가 중복되어 있음을 알 수 있다.
3. **위조:** 태그에는 데이터 항목이 존재한다. 이 데이터 항목을 위조해 리더기와 태그가 잘못된 데이터를 교환하도록 만든다.
4. **DoS 공격:** 리더기와 태그 간 질의응답 메커니즘을 이용해 공격자가 수많은 질의를 리더기와 태그에 보내면 서비스 거부를 일으킬 수 있다.

**RFID 보안 기술(암호화 없음):**

1. **킬 명령어:** 현재 EPC(Electronic Product Code)global의 표준이다. 태그에 킬 명령어를 전송하면 태그를 영구적으로 비활성화 시킬 수 있다. 한 번 비활성화된 태그의 내용은 영원히 재활용할 수 없어 활용도가 크게 떨어지게 된다. 번외로 EPC는 차세대 바코드로 비트 크기에 따라 EPC-64, 96, 256 등의 버전이 있다. EPC는 기존 바코드와 다르게 제품 하나 하나에 고유 번호가 부여되며 가격, 제조업체, 위치 정보 등이 DB에 연동되어 미래 유통 시장의 혁명을 일으킬 것으로 예측된다.
2. **Sleep 과 Wake 명령어:** Kill의 단점을 보완하기 위해 나왔다. Sleep으로 효력을 정지시키고 안전한 상황과 장소에서 다시 Wake 명령을 통해 코드를 활성화 시킬 수 있다. 다만 사용자가 키 관리/태그의 동작 관리를 일일이 신경써야 한다는 단점이 있다.
3. **Blocker 태그 기법:** 블로커 태그라고 불리는 전용 IC태그를 소비자가 소비함으로써 가까이에 있는 IC 태그의 ID를 읽을 수 없게 만드는 것
4. **Faraday Cage:** 금속 박막으로 태그를 막아 무선 신호의 전달을 방해하는 방법. 사용법이 제한되고 상품절도에 악용될 소지가 있다.
5. **Jamming:** 전파 방해 신호를 보내 불법적인 리더기의 접근을 막아내는 방식이다. 근처의 인가된 리더기도 제 구실을 못하게 해 용도가 제한된 방법이다.

**RFID 보안 기술(암호화):**

1. **Hash Lock 기법:** RFID 태깅 하드웨어는 연산 능력이나 리소스 면에서 기존 접근제어 메커니즘을 활용하기 어려운 측면이 있다. 따라서 one-way 해시 함수에 기반한 해시 락 접근 제어 정도를 사용한다.
2. **XOR(Exclusive OR)기반 원타임 패드 기법:** XOR 기반 원타임 패드 기법을 통해 XOR()과 같은 간단한 연산으로 암호화한다.

**모바일 보안**

**iOS:** 맥 OS인 OS X의 모바일 버전에서 시작. 맥 OS는 다윈 유닉스에서 파생된 것으로 iOS의 틀은 유닉스이다. 오픈소스가 아니기 때문에 보안에 대한 통제권이 애플에게 있고 애플은 이를 기반으로 통제된 보안 체계를 갖추고 있다.

**iOS 보안 체계:**

1. **안전한 부팅 절차 확보:** iOS에서 동작하는 모든 소프트웨어는 애플 암호화 로직의 서명된 방식에 의해 무결성이 확보된 후 동작한다. 이는 부팅과정에서도 동일하다.
2. **시스템 소프트웨어 개인화:** 모든 소프트웨어를 앱스토어를 통해 일괄 배포해 보안상 문제에 직접 대응할 수 있도록 하고 있다.
3. **응용 프로그램에 대한 서명:** 설치되는 모든 앱에 코드 무결성 사인을 등록하게 하고 있다. 코드 무결성 사인은 앱에 대한 해시값으로 등록된 앱의 코드 무결성 사인이 다를 경우 다른 앱을 설치하지 못하게 한다.
4. **샌드박스 활용:** 응용 프로그램이 실행될 때 일종의 가상머신이 안에서 실행되는 것처럼 원래의 운영체제와 완전히 독립되어 실행된다.

**iOS의 취약점:** 탈옥에 의해 취약점이 드러날 수 있다.

**안드로이드:** 리눅스 커널(2.6.25)을 기반으로 한 모바일 운영체제. 48개 하드웨어, 소프트웨어, 통신 회사가 만든 오픈 핸드셋 얼라이언스(OHA)에서 공개 표준을 개발하고 있다.

**안드로이드의 보안 체계:** 안드로이드는 리눅스 커널 기반 구조 + 응용 프로그램 프레임워크 기반의 브라우저를 지원하고, 자바로 작성된 소프트웨어는 달빅 가상 머신에서 실행 가능한 코드로 컴파일된다. 모든 응용 프로그램은 일반 사용자 권한으로 실행, 사용자의 데이터에 접근할 때 모든 사항을 응용프로그램 사양에 명시하고 접근 시 사용자의 동의를 받게 한다. iOS와 마찬가지로 샌드박스를 활용해 애플리케이션 접근을 통제한다.

**안드로이드 취약점:** iOS의 탈옥과 비슷한 개념으로 루팅을 할 수 있다.

**BYOD 보안 기술:**

1. **MDM(Mobile Device Management):** IT부서가 기기를 완전히 제어할 수 있도록 직원의 스마트패드와 스마트폰에 잠금, 제어, 암호화, 보안 정책 실행을 할 수 있는 기능을 제공한다. 탈옥 기기는 사용 금지되며 분실 또는 도난 시 원격 데이터 삭제 등 모바일 기기의 중요한 디바이스 관리 기능을 제공한다.
2. **컨테이너화:** 모바일 기기 내 업무용과 개인용 영역을 구분해 보안과 프라이버시 문제 해결
3. **모바일 가상화:** 하나의 기기에 개인용과 업무용 운영체제를 동시에 담아 개인과 사무 정보 분리, 평시에는 개인용 운영체제를 사용하다가 업무시 업무용 운영체제로 전환한다.
4. **MAM(Mobile Application Management):** 기기에 설치된 업무 관련 앱에만 보안 및 관리 기능 적용.
5. **NAC(Network Access Control):** 사용자 기기가 내부 네트워크 접근 전 보안 정책을 준수했는지 검사해 비정상 접근 시 네트워크 접근을 통제한다.

**24절. 네트워크 관리**

네트워크 자원을 적절히 관리해 네트워크 사업자와 사용자 간에 안정적인 네트워크 서비스를 유지하는 것을 말한다. 네트워크 기획, 운영, 유지보수 등에 필요한 정보를 네트워크를 구성하는 다양한 장비 및 전송설비로부터 수집, 분석하고 이 분석 정보를 통해 네트워크의 상태를 파악, 장애가 발생하였을 때 적절한 제어기능을 수행하고 보고하는 등의 활동.

**필요성:** 네트워크 규모가 점점 커지고 이질적인 네트워크들이 공존함으로써 전체 네트워크의 원활한 동작을 위함. ISP(Internet Service Provider) 사업자들이 인터넷 서비스에 대한 신뢰도를 높이기 위함.

**네트워크의 5대 관리 기능:** 계정 관리(인증과 과금 기능), 구성 관리(구성원간 연결 상태와 관계 정보 제공), 성능 관리(네트워크의 처리율/이용도/에러 등 측정 및 효율성 평가), 장애 관리, 보안 관리(패스워드나 암호화된 데이터 링크 유지보수, 보안 행위 기록)

**SNMP(Simple Network Management Protocol):** 관리자와 에이전트 개념 사용. 호스트인 관리자는 보통 라우터나 서버인 에이전트 집합을 제어하고 감시한다. 프로토콜은 응용 수준에서 설계. 관리 대상의 하드웨어적 특성과 하부 네트워크 기술로부터 관리 작업을 분리한다. 따라서 서로 다른 제조사가 만들 라우터에 의해 연결된 서로 다른 LAN과 WAN으로 구성되는 이질적인 네트워크에서도 사용할 수 있다.

**관리자와 에이전트:** 에이전트는 디비에 성능정보 저장. 관리자는 디비에서 값을 읽어간다. 관리자는 라우터로 하여금 특정 동작을 행하도록 할 수 있다. SNMP에서 관리 정보를 교환할 때는 폴링과 트랩을 사용한다.

**관리 구성요소:** SNMP는 관리를 위해 두 가지 프로토콜 SMI(Structure of Management Information)과 MIB(Management Information Base)를 사용한다. 인터넷에서의 관리는 SMI와 MIB의 협동 작업이다.   
**SNMP**는 관리자와 에이전트 사이에 교환되는 패킷의 형식을 정의한다. 또한 패킷에서 객체(변수)의 상태(값)을 읽고 변경한다.

**SMI의 역할:** SMI는 객체에 이름을 붙이고 객체 유형을 정의하며 객체와 값을 부호화하는 방법을 표현하기 위한 규칙을 정의한다.

**MIB의 역할:** 관리될 각 객체의 수를 결정하고, 이들을 SMI에 의해 정의된 규칙에 따라 이름 붙이고 이름이 지어진 각 객체에 유형을 연결한다.

**유사성:** SMI -> 언어 구문, MIB -> 선언, SNMP -> 프로그램 코딩

**SNMP 주요 연산:** 관리자가 에이전트에서 정의된 객체의 값을 읽고, 정의된 객체에 값을 저장한다. 에이전트가 비정상적 상황에 대한 경고 메시지를 관리자에게 보낸다.

**PDU:** SNMPv3은 여덟 유형(GetRequest, Get-NetRequest, GetBulkRequest, SetRequest, Response, Trap, InformRequest, Report)의 프로토콜 데이터 유닛(PDU)을 정의한다.

|  |  |  |  |
| --- | --- | --- | --- |
| **PDU 클래스** | **설명** | **SNMPv1** | **SNMPv2/SNMPv3** |
| Read | 폴링 사용, 관리 장비에서 관리 정보를 읽는다. | GetRequest,  GetNextRequest | GetRequest |
| Write | 관리된 장비의 관리 정보를 변경해 장비의 동작을 바꾸는 메시지 | SetRequest | SetRequest |
| Response | 기존의 요청 메시지에 대한 응답으로 보내는 메시지 | GetResponse | Response |
| Notification | 장비에서 인터럽트 방식으로 SNMP 매니저에게 정보를 보낼 때 사용 | Trap | Trapv2,InformRequest |

텍스트, 영수증, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

**커뮤니티 스트링:** 매니저가 에이전트로부터 데이터를 요청하기 위해 사용하는 패스워드이다. 접근 수준이 다른 두 메인 커뮤니티 스트링이 있다.

1. 읽기 전용 스트링: 매니저가 장치 내의 MIB에 보관되어 있는 데이터를 읽을 수 있다.(public)
2. 읽기 쓰기 스트링: 매니저가 데이터를 읽고 수정할 수 있도록 한다.(private)

대부분 커뮤니티 스트링을 변경하지 않는다. 포트 161로 연결해 장치의 상태 정보를 읽을 수 있으면 누구나 장치를 재설정할 수 있다. 또한 커뮤니티 스트링은 SNMPv1과 v2에서 평문으로 전송된다. 기본값을 변경해도 스니퍼가 쉽게 접근할 수 있다. 반면 v3는 암호화 기능을 가지므로 암호화, 메시지 무결성, 인증 보안을 제공한다. 커뮤니티 스트링은 대부분의 회사에서 기본값(public)을 쓰거나 상호, monitor, router, mrtg 등 사회공학적으로 추측 가능한 문자열을 사용하므로 사회공학 공격에 쉽게 노출되어 있다. 관성적인 스트링 대신 추측하기 어렵고 의미가 없는 문자열로 변경해야 한다.

**원격 접속 서비스**

**TELNET:** TErminaL NETwork의 약자. ISO에서 제안된 가상 터미널 서비스에 대한 표준 TCP/IP 프로토콜이다.

**원격 로그인:** 원격 장치에 위치한 응용 프로그램에 접근하고자 할 때 원격 로그인을 수행해야 한다. 이 경우 텔넷 클라이언트와 서버 프로그램을 사용한다.

**Rlogin:** 원격 로그인을 의미한다. TELNET 프로콜과 같이 원격지 접속을 지원하는 프로토콜이다. 인터넷 표준은 아니기 때문에 BSD 계열 시스템 간의 원격접속을 위해 설계되었다. 텔넷에 비해 단순하고 별도 옵션 협상기능은 제공하지 않는다.

**SSH(Secure Shell):** 텔넷과 마찬가지로 TCP를 하부 전송 프로토콜로 사용하나 더 안전하고 텔넷보다 더 많은 서비스를 제공한다. SSH-2는 초기 방식의 보안상 취약점을 수정한 버전이다. 초기 버전과 호환되지 않는다.

**컴포넌트:**

1. **SSH 전송 계층 프로토콜(SSH-TRANS):** TCP가 안전한 전송 계층 프로토콜이 아니므로 SSH는 먼저 TCP 상에 안전한 채널을 생성하는 프로토콜을 사용한다. 새로운 계층은 SSH-TRANS라고 불리는 독립 프로토콜이다.
2. **SSH 인증 프로토콜(SSH-AUTH):** 클라이언트와 서버 간에 안전한 채널이 설정되고 클라이언트에 대해 서버 인증이 이루어진 후 SSH는 서버에 대해 클라이언트를 인증하는 소프트웨어를 호출할 수 있다.
3. **SSH 연결 프로토콜(SSH-CONN):** SSH-CONN 프로토콜에 의해 제공되는 서비스 중 하나는 여러 개의 논리적 통신채널의 다중화를 수행하는 것이다.
4. **SSH 응용(SSH-Application):** 연결을 마치면 SSH는 몇 가지 응용 프로그램이 연결을 사용할 수 있도록 한다. 각 응용은 안전하게 논리적 채널을 생성할 수 있다. 원격 로그인은 이런 SSH-CONN 프로토콜을 이용하는 서비스 중 하나이다. 파일 전송 응용 또한 이런 목적으로 논리적 채널 중 하나를 사용한다.

**포트 전달(Port Forwarding):** 텔넷, SMTP 등과 같은 응용에서 SSH의 포트 포워딩 기능을 사용할 수 있다. SSH의 포트 포워딩은 다른 프로토콜에 속한 메시지가 지나가는 터널을 만든다. 이에 따라 이 방법을 SSH터널링이라고 부른다.

**SSH 암호화 통신:** 공개키 암호화 기법을 사용해 암호화 통신한다. SSH1은 접속과 인증을 위해 RSA를 이용한다. 또한 통신의 암호화를 위해 Blowfish, DES, 3DES, RC4 및 IDEA를 이용한다.

**SSH 암호화 과정:**

1. 클라이언트가 서버의 공개키를 얻어온다.
2. 클라이언트는 자신의 사설키로 데이터를 암호화하고 이를 다시 서버의 공개키로 암호화해 서버로 전송한다.
3. 서버는 클라이언트에게서 전송 받은 데이터를 자신의 사설키로 복호화 한 후 이를 다시 클라이언트의 공개키로 복호화 해 데이터를 읽는다.

**25절. 네트워크 기반 프로그램 활용**

**연결테스트(ping):** 핑은 ICMP를 이용하는 유틸리티이다. ICMP타입 중 Echo Request(Type 8) 타입을 지정해 대상 시스템으로 ICMP패킷을 전송 후 대상으로부터 Echo Reply(Type 0) 패킷을 수신함으로써 접근성을 확인하게 된다. 접속대상 시스템의 접근성을 진단하는 것 외에 네트워크 속도를 측정할 수도 있다. 도메인명을 이용해 접속대상 시스템의 IP 주소를 알아낼 때도 효과적이다.

**핑이 제공하는 정보:** 대상 시스템에 ICMP 패킷이 도달 가능한지?, ICMP 패킷이 대상 시스템을 왕복하는데 걸리는 시간, ICMP 패킷의 손실률

**리눅스 핑 명령어:** ping [-option] ipAddress | hostname

**핑 옵션:** -c -> count 지정된 숫자만큼 ICMP패킷 송수신, -I -> wait ICMP 패킷 송신 간의 대기시간. 디폴트는 1초, -s -> packetSize ICMP 송신 패킷의 데이터 크기. 디폴트는 56바이트

**윈도우 핑 결과분석:** ICMP 패킷의 길이, 공격 대상이 보내온 ICMP Echo Reply 패킷의 크기, Echo Request 패킷을 보낸 후 Reply 패킷을 받기까지 경과시간(time = 99ms), TTL -> 라우팅 정보가 잘못되어 패킷이 라우터를 무한히 돌아다니지 않도록 최초 라우터를 제외한 라우터 하나를 지날 때마다 값이 1씩 줄어든다. Packets: 영역은 차례로 Request 패킷의 개수, Reply 패킷의 개수, 손실된 패킷 개수를 보여준다. 마지막 줄은 Request 패킷을 보낸 후 Reply 패킷이 오기까지 시간에 관한 정보를 알려준다.

**스위핑(Sweeping):** ICMP를 이용한 핑은 시스템 하나를 조사하기에는 적절하지만 큰 네트워크에서 활성화 시스템을 찾는 데는 그다지 효과적이지 않다. 네트워크 전체에서 활성화 시스템을 찾는 일을 스위핑이라고 한다. 검색하고자 하는 네트워크에 브로드캐스트 ping을 보내거나 자동화 도구를 이용해 특정 범위의 네트워크에 ping을 보낸다. 툴로는 fping이 있다.

**경로추적(traceroute)**

**TTL:** Time To Live. 목적지를 찾기 위해 경유하는 라우터의 개수, 라우터를 거치면 1씩 감소하고 목적지에 도달하기 전에 0에 도달하면 폐기한다. 라우팅의 무한 루프를 막기 위해 사용한다.

**동작원리:** 중계 노드(L3/라우터)의 구간에 대한 네트워크 상태를 관리하기 위한 명령어이다. 네트워크의 라우팅 문제를 찾기 위한 목적으로 많이 사용한다.(중계 노드의 개수, IP주소, 응답시간 파악 가능), traceroutes는 패킷의 TTL을 하나씩 증가시켜 보낸다. 기본값은 1로 시작한다. 도착한 라우터에서 TTL이 감소하고 ICMP가 출발 메시지로 보내진다. 다음 2의 TTL을 갖는 UDP 패킷은 두 번째 라우터에 도달해 소멸한다. 목적지에 도달할 때까지 계속 패킷을 보내는데 이 패킷은 사용 불가능한 포트번호(33434)를 붙여서 간다. Unreachable Port라는 ICMP 메시지를 받으면 trace가 목적지에 도달했음을 알 수 있다. Traceroute는 우연히 도달하는 것을 방지하기 위해 정확히 3개의 UDP 패킷을 보낸다.

**traceroute 사용방법:** 결과에서 응답시간이 \*로 표시되는 경우 침입차단시스템의 접근통제리스트에 의해 traceroute의 UDP패킷이 보안상의 이유로 차단되었거나 실제 해당 구간에 문제가 발생한 것이다. Traceroute를 이용한 Troubleshooting은 다음과 같다. 지정한 주소가 실제 존재하는지 확인하고 없다면 패킷이 멈춘 곳을 알려준다. 수행속도가 느리면 어느 구간에서 지연이 발생하는지 확인한다. 패킷이 적당한 곳을 통해 라우팅되고 있는지 확인한다. \*모양이 생기는 곳이 있는지 확인한다.

**traceroute 명령어:** traceroute [-option] ipAddress | hostName [packet\_len]

**traceroute Option:** -m -> maxHop: 최대 TTL 지정(디폴트 30), -p -> port: UDP 포트 지정(디폴트 33434), -q -> n\_queires: TTL마다 전달하는 탐색 패킷의 횟수.(디폴트 3), -w wait\_time: 탐색 중 응답을 기다리는 시간.(디폴트 5), packet\_len: 탐색 패킷의 크기

ex) traceroute 192.168.0.100

traceroute -q 4 192.168.0.100 /\* TTL마다 4회 탐색\*/

5초 동안 응답이 없으면 \*로 표시

**결과:** hops max(TTL의 디폴트값), 다음 표기되는 byte packets는 전송되는 패킷의 크기

제일 앞에 일련번호는 TTL값을 나타내며 뒤의 IP주소는 현재 경우하는 네트워크 구간의 주소를 나타낸다. 마지막 세 개의 필드(ex) 13.149ms 6.454ms 9.124ms)는 왕복시간을 3회 출력한 것이다. 송신된 탐색 패킷의 결과가 5초 이내에 도달하지 않으면 해당 TTL의 왕복시간은 \*기호로 표기

**네트워크 인터페이스 진단(netstat)**

**netstat**명령은 Unix 시스템의 TCP/IP 프로토콜 진단 시 사용한다. LAN카드(네트워크 인터페이스)의 성능, 시스템의 라우팅 정보, 소켓 사용 정보 등 지정 옵션에 따라 네트워크 정보를 제공한다. (네트워크 연결 상태 확인, 컴퓨터의 열린 포트 확인, 네트워크 연결 상태에서 외부에서 접속해온 호스트 유추 가능)]

**netstat 명령어와 옵션:** netstat [-option] -r: 라우팅 정보 출력, -I [interval]: 네트워크 인터페이스 정보 출력, -s: 각 네트워크 프로토콜(IP, TCP, UDP, ICMP)에 대한 통계정보 출력, -a: 모든 소켓 정보 출력, -n: 네트워크 주소를 숫자로 나타냄

**netstat 필드 내용:**

|  |  |
| --- | --- |
| 항목 | 내용 |
| Proto | 응용 프로그램을 연결하기 위한 프로토콜 |
| Recv-Q | 원격 응용프로그램에서 수신해 버퍼에 저장한 데이터 크기 |
| Send-Q | 원격 응용 프로그램으로 전송하기 위해 송신 버퍼에 저장된 데이터의 크기 |
| Local Address | 연결에 사용한 로컬 호스트 IP와 응용 프로그램이 사용하는 포트 |
| Foreign Address | 로컬 호스트와 연결된 원격 호스트의 IP 주소와 응용 프로그램의 포트 |
| State | 연결과 관련된 소켓 상태 |

**3-way handshake 연결설정 과정의 state 값:**

|  |  |
| --- | --- |
| 항목 | 내용 |
| LISTEN | 서버 프로그램이 연결요청을 기다림 |
| SYN-SENT | 클라이언트 프로그램이 원격 호스트에 연결 요청 |
| SYN-RECV | 서버가 클라이언트 프로그램으로부터 연결요청을 받아 클라이언트에 응답을 했지만 아직 클라이언트의 확인 메시지를 받지 않음 |
| ESTABLISHED | 클라이언트와 서버 프로그램이 서로 연결됨 |

**4-way handshake 연결종료 과정의 state 값:**

|  |  |
| --- | --- |
| 항목 | 내용 |
| FIN-WAIT1 | 한쪽 응용 프로그램이 연결 종료 요청 |
| CLOSE-WAIT | 연결 종료 메시지 수신 후 그에 대한 확인 메시지 보냄 |
| FIN-WAIT2 | 로컬에서 종결 메시지 전송, 원격 호스트로부터 이에 대한 확인 메시지 수신했지만 원격 프로그램이 작업을 종료하지 않아 원격 호스트의 종결 메시지 기다림 |
| LAST-ACK | 원격으로부터 종결요청을 받고 로컬에서도 회선 종결에 합의, 확인 메시지 수신 시 연결 종료 |
| TIME-WAIT | 연결 종료, 마지막 종료 확인 메시지를 재전송할 경우 감안해 소켓정보 유지상태 |
| CLOSED | 연결이 완전히 종료 |

**라우팅 테이블 설정(route)**

route 명령으로 라우팅 테이블에 라우팅 경로를 추가하거나 삭제할 수 있다. 라우팅 테이블 관리는 동적 방식과 정적 방식으로 나뉘는데 시스템 관리자가 설정을 통해 관리하게 되면 정적 방식이고 라우팅 프로토콜을 이용하면 동적 방식이다. route 명령어는 정적 방식에 해당한다.

**리눅스 route 명령어:** route add [-net | -host] destination [gw gateway] [netmask mask] [[dev] interface]

route del [-net | -host] destination [gw gateway] [netmask mask] [[dev] interface]

add: 라우팅 경로를 추가한다.

del: 라우팅 경로를 삭제한다.

-net | -host: destination의 값이 네트워크라면 net, 호스트라면 host를 지정한다.

gw: 게이트웨이의 IP주소를 gateway 자리에 지정한다.

netmask: 추가하는 라우팅 경로의 넷마스크(Netmask)를 지정한다.

dev: 인터페이스 장치명을 interface 자리에 지정한다.

**예문:**

1. route add -net 203.249.7.0 netmask 255.255.255.0 gw 192.168.1.100  
   -> 새 라우팅 경로를 추가한다. 203.249.7.0의 네트워크에 해당하는 IP주소가 전달되면 192.168.1.100번의 IP주소를 갖는 게이트웨이로 라우팅되도록 정의한다.
2. route add default gw 192.168.1.100

* 디폴트 게이트웨이를 추가한다. 라우팅 테이블에 없는 목적지는 디폴트 게이트웨이인 192.168.0.1로 라우팅 되도록 정의한다.

1. route del -net 203.249.7.0 netmask 255.255.255.0 gw 192.168.1.100

* 네트워크 주소가 203.249.7.0인 목적지에 대한 라우팅 경로를 삭제한다.

destination은 목적지 IP주소 또는 네트워크 주소를 지정한다. 디폴트 게이트웨이를 지정하려면 destination을 default 키워드로 기술한다.

**네트워크 패킷/로그 분석(tcpdump)**

tcpdump는 네트워크 인터페이스를 거치는 패킷의 내용을 출력해준다. 스니핑 도구로 자신의 컴퓨터로 들어오는 모든 패킷 내용을 도청할 수 있으며 공격자의 추적 및 공격 유형을 분석하기 위해 활용할 수 있다. 유닉스 계열에서 설치, 활용하며 윈도우용으로는 windowdump가 있다.

**tcpdump 명령어:** tcpdump [옵션], -A: 패킷 내용을 ASCII로 보여준다. -c: 주어진 수의 패킷을 받은 후 종료한다., -C: 방금 받은 패킷을 저장파일로 만들기 전에 파일이 file\_size보다 큰지 체크한다. 만약 크다면 현재 저장파일을 닫고 새로 하나를 연다., -D: 패킷을 잡을 수 있는 시스템 상에 가능한 네트워크 인터페이스 목록을 출력한다. , -i: 인터페이스를 정한다. 정해지지 않았으면 tcpdump는 시스템 인터페이스 목록에서 가장 낮은 숫자를 고른다.

**예문:** tcpdump -I eth0 -> 별도 옵션이 없으면 tcpdump는 기본으로 eth0 인터페이스의 트래픽을 모니터링한다.

tcmpdump host 211.47.66.50 and port 80 소스나 목적지에 IP가 211.47.66.50이고 포트가 80인 트래픽을 모니터링 하고자 할 때 사용한다. 여러 패킷필터가 있을 경우 위와 같이 and또는 or를 사용할 수 있다.

**네트워크 인터페이스 설정(ifconfig)**

ifconfig은 네트워크 인터페이스의 설정정보를 알아보거나 IP주소 및 서브넷마스크 등의 설정을 변경할 때 사용한다. 네트워크 인터페이스를 활성화 시키거나 비활성화 시키는 기능도 있다.

**ifconfig 명령어:** ifconfig [device] [ipAddress] [netmask mask] [broadcast broadcast\_ip] [up|down]

device: IP주소를 부여할 장치

ip\_address: 인터페이스의 IP주소 지정

netmask: 해당 장치에 부여할 넷마스크를 mask 자리에 지정

broadcast: 로컬 네트워크의 브로드캐스트 주소를 boradcate\_ip 자리에 지정

up|down: 장치를 활성화(up)하거나 비활성화(down)한다.

**예시: ifconfig -a //시스템에 있는 활성화 또는 비활성화된 모든 인터페이스 출력**

**ifconfig hme0 //장치 hme0의 인터페이스 정보 출력**

**윈도우 ipconfig 유틸리티**

윈도우는 네트워크 설정이 유닉스와 다르다. 유닉스는 ifconfig 프로그램을 통해 다양한 설정 요소를 조회하고 수정할 수 있다. 그러나 윈도우는 대부분의 설정과 인자 수정이 제어판에서 이뤄진다. ifconfig과 유사한 유틸리는 ipconfig으로 옵션을 사용해 출력을 제어할 수 있지만 ifconfig보다 훨씬 간단하고 옵션의 수도 적다.

**ipconfig 옵션: (없음) ->** 호스트의 각 인터페이스 IP주소, 서브넷 마스크, 기본 게이트 웨이 출력

/all -> 없음과 유사하지만 호스트의 인터페이스에 대한 좀 더 구체적인 설정 정보 출력

/release[<어댑터>] -> 지정한 어댑터(인터페이스) 또는 모든 어댑터의 DHCP 임대 종료

/renew[<어댑터>] -> 지정한 어댑터(인터페이스) 또는 모든 어댑터의 DHCP 임대를 수동으로 갱신

/displaydns -> 호스트의 DNS 변환기 캐시의 내용을 출력

/flushdns -> 호스트의 DNS 변환기 캐시 내용을 삭제

/registerdns -> 모든 DHCP 임대를 갱신하고 호스트와 연관된 모든 DNS 이름을 다시 등록

**26절. 네트워크 기반 공격의 이해**

**네트워크를 취약하게 만드는 요인:** 1. 많은 공격지점(대규모 네트워크일수록)**,** 2. 공유, 3. 시스템의 복잡성

물리적 공격과 기술적 공격(수동/능동)으로 분류된다.

|  |  |  |
| --- | --- | --- |
| 구분 | 수동적 공격 | 능동적 공격 |
| 특징 | 직접적 피해 없음(read) | 직접적 피해 있음(write) |
| 탐지 가능성 | 어려움 | 쉬움 |
| 예시 | 스니핑, 도청(Eavesdrop) | 재전송 공격, 변조, DoS/DDoS, 세션 하이재킹 |

**네트워크 기반 공격 및 대응책**

**서비스 거부 공격(DoS):** 시스템에 과부하를 일으켜 가용성을 떨어뜨리는 공격. DoS는 단일 컴퓨터가 공격하는 형태이고 DDoS는 분산 컴퓨터(좀비PC)로 공격하는 형태이다. DoS 공격은 크게 소프트웨어 취약점을 노린 공격(poison 패킷)과 flooding 유형의 공격으로 구분할 수 있다.

다른 분류로는 파괴 공격(데이터, 시스템), 시스템 자원 고갈 공격(CPU, 메모리 부하), 네트워크 자원 고갈 공격(쓰레기 데이터로 네트워크 대역폭 고갈)이 있다.

**TCP SYN Flooding Attack:** TCP 연결 설정 중 3-way handshaking 과정에서 Half-Open연결 시도가 가능하다는 취약점을 이용한 공격이다. 공격 대상 시스템은 외부로부터 접속 요청을 더 이상 받아들일 수 없게 되어 정상적인 서비스를 제공할 수 없게 된다. 공격자가 다수의 syn 신호를 공격대상자에게 전송하면 공격대상자는 syn/ack 신호를 공격자에게 전달하게 되는데 이때 공격자가 ack 신호를 반송하지 않으면 공격대상 시스템은 일정 시간동안 신호를 기다리게 된다. 이 공격은 윈도우 시스템 뿐 아니라 인터넷에 연결된 모든 TCP 기반 서비스(웹서버, FTP 서버, 메일서버 등)에 피해를 줄 수 있다.

1. **시작 주소 Spoofing:** DoS 공격에 이용되는 패킷은 조작된 시작 주소를 특징으로 한다. 시스템의 네트워크 코드에 접근할 수 있다면 조작된 시작 주소를 가진 패킷을 만들 수 있다. 주로 raw socket interface를 통해 행해진다. 이 인터페이스는 네트워크를 테스트하거나 네트워크 프로토콜 연구 목적으로 개발되었으나 정상 작업에는 전혀 필요하지 않은 기능이다.
2. **SYN Flooding Attack 공격 과정:**
3. 공격자가 시작 주소를 spoofing한 상당한 양의 SYN 메시지를 목적지에 보낸다. 서버는 이 메시지의 정보를 저장하고 SYN+ACK로 응답하게 된다.
4. SYN을 송신하지 않은 TCP가 SYN+ACK을 수신하면 RET패킷을 회신해 연결설정 작업중단을 요청하고 RST 패킷을 수신한 서버는 연결설정 과정을 중단하고 연결 테이블의 자료구조를 해제한다.
5. 그러나 위조 IP가 너무 바쁘거나 사용하지 않는 IP인 경우 RST 패킷이 서버로 회신되지 않고 서버는 정해진 시간 동안 SYN+ACK 패킷을 재전송하며 연결 설정 시도를 반복하게 된다.
6. 따라서 SYN Flooding 공격은 RST 패킷 수신에 의한 연결설정 작접 중단을 막기 위해 사용하지 않는 IP주소로 source address spoofing하는 게 일반적이다.
7. 서버의 TCP 연결 테이블이 꽉 차게 되고 이후 들어오는 연결 요청이 거부되면 공격이 성공한 것이다.
8. **SYN Flooding 보안대책:** 방화벽, DDoS 대응장비를 이용해 동일 클라이언트(IP)의 연결(SYN) 요청에 대한 임계치(Threshold) 설정을 통해 과도한 연결 요청이 들어오는 것을 차단한다. iptables를 이용한 설정 예를 살펴보면 다음과 같다.

iptables **-**A INPUT -p TCP - -dport 80 - -syn -m connlimit - -connlimit-above 5 -j DROP: TCP 포트 80 연결 요청에 대해 동일 출발지 IP 로 동시 연결 개수가 5개 초과 시 이를 차단한다.

**+위조 IP주소 패킷 필터링(Spoofed IP Address Packet Filtering):** 위조된 IP 패킷을 필터링해 위조 SYN 패킷이 서버 방향으로 전달될 수 없게 차단하는 게 기본적 대응책이다. 공격자 시스템에 연결된 네트워크 라우터는 자신의 네트워크 내의 모든 IP 주소를 알고 있으니 패킷의 출발지 IP주소 검사를 통해 위조 IP패킷을 구별할 수 있다. 그러나 공격자 시스템이 속한 네트워크의 라우터가 위조 IP필터링 책임의 1차 책임을 지지 않는 경우가 많다.

**Syn\_Cookie:** 서버에서 클라이언트로 보내는 SYN+ACK 패킷에 임의로 생성되는 시퀀스 넘버 대신 서버에서 암호화 기술을 이용한 인증 정보가 담긴 시퀀스 넘버를 생성해 클라이언트로 보낸다. 클라이언트로부터 받은 SYN 패킷을 받아도 Syn\_Cookie만 보내고 세션을 닫아버릴 수 있어 기다릴 필요가 없다. 대기하지 않아도 되어 SYN Flooding이 통하지 않게 된다.

**Syn\_Cookie ON[리눅스]:** sysctl -w net.ipv4.tcp\_syncookies=1

구체적으로 SYN 쿠키 기법은 서버가 수신한 SYN 패킷의 시간정보, IP, 시작 순서번호, 비밀번호 등을 입력해 해시값을 쿠키로 구한다.이 해시값을 서버 시작 순서번호(ISN)로 하는 SYN+ACK 패킷을 회신하는 것이다. 그리고 이에 대한 정상적 ACK 응답 패킷을 수신하기 전까지 연결 테이블에 자료구조 설정을 미룬다. 쿠키에는 시간정보와 비밀번호 등이 포함되어 재사용/위조가 힘듦으로 서버가 송신한 SYN+ACK 패킷을 정상적으로 수신하지 않은 공격하는 확인 번호(ack no)를 추론할 수 없다. 때문에 ACK 패킷을 위조해 서버로 전송할 수도 없다. 이렇게 해서 위조 SYN패킷에 의한 연결 테이블의 자원이 소진되는 것을 막을 수 있다.

**First SYN Drop 설정:** 연결(SYN) 요청 패킷을 보내는 클라이언트가 실재하는지 확인하는 방법이다. 첫 번째 SYN을 Drop해 재요청 패킷이 오는지 확인하면 출발지 IP가 Spoofing되었는지 확인할 수 있다. 대부분의 공격툴은 SYN 요청을 생성할 뿐 Drop 시 재전송을 수행하지 않음을 이용한 방법이다.

**TCP 연결 테이블(Backlog Queue) 엔트리 선택적 삭제 기법:** 백로그 큐에는 TCP 세션을 맺기 위해 요청된 세션 정보가 저장되어 있다. 이 연결 테이블이 오버플로우될 때 일부 엔트리를 삭제해 새 SYN 패킷을 처리할 수 있게 만드는 방법이다.

**연결 테이블 확장:** 연결 테이블에 할당된 메모리를 확장해 SYN 플러딩 공격에 보다 오래 버틸 수 있게 하는 기법.

[Linux]#sysctl -w net.ipv4.tcp\_max\_syn\_bakclog = 1024

TCP Backlog Queue의 크기를 1024로 늘린다.

연결 테이블 삭제 기법은 정상적인 연결을 삭제할 수 있고 확장은 지속적인 공격에 대한 대비책이 아니다.

**SMURF Attack:**

**증폭네트워크: ICMP Echo Request** 브로드캐스트 메시지를 받아 희생자에게 다수의 증폭된 ICMP Echo Reply를 전송하는 네트워크이다.

DoS 공격 중 가장 피해가 큰 광범위 공격. 가장 인기있는 DoS 공격으로 IP 위장과 ICMP의 특징을 이용한다. 공격자가 출발지 IP를 목표 시스템으로 스푸핑하고 목적지 주소를 직접 브로드캐스트 주소(Directed Broadcast Address)로 설정한 Ping을 날린다. Ping을 수신한 네트워크 내의 모든 시스템은 Ping 응답 메시지를 스푸핑된 타겟으로 동시에 전송함으로써 공격 목표를 마비시킨다. 직접적 ICMP 브로드캐스트와 세 구성요소인 공격자, 증폭 네트워크(Amplifier(Bounce) Network), 공격대상 서버를 이용한다.

**Fraggle Attack:** ICMP대신에 UDP를 사용하는 공격.

**SMURF 공격 방법:**

1. 공격자는 IP를 공격대상 서버의 IP로 위장해 ICMP를 브로드캐스트로 다수 시스템에 전송한다.
2. ICMP 브로드캐스트를 수신한 다수 시스템은 ICMP Echo 패킷을 공격자가 아닌 공격대상 서버로 응답하게 된다.
3. 타겟은 다수의 ICMP Echo를 수신하게 되어 시스템 성능에 문제를 일으킨다.
4. 다수 시스템은 브로드캐스트에 대한 응답으로 다수의 ICMP Echo 패킷을 발생시키기 때문에 증폭 네트워크라고 불리게 된다.

**SMURF 보안대책:**

1. 라우터 명령어: (config-if)#no ip directed-broadcast)

라우터에서 다른 네트워크로부터 자신의 네트워크로 들어오는 IP directed broadcast 패킷을 막도록 설정한다.

1. 호스트는 IP broadcast address로 전송된 ICMP패킷에 대해 응답하지 않도록 시스템을 설정할 수 있다.
2. 스머프 공격 특성상 공격이 시작되면 단시간에 다수 ICMP Echo Reply 패킷을 희생자에게 전송한다. 따라서 동일한 ICMP Echo Reply 패킷이 다량으로 발생하면 해당 패킷들을 침입차단시스템을 통해 모두 차단(Drop)한다.

**LAND Attack:** Land의 사전적 의미 중 하나인 “나쁜 상태에 빠지게 하다.”에서 왔다. 패킷을 전송할 때 출발과 목적지 IP를 동일하게 세팅해 공격 대상에게 보낸다. 세팅된 IP주소값이 공격 대상의 IP주소이다. SYN Flooding처럼 동시 사용자 수를 증가시켜 CPU에 부하를 준다.

**Land Attack 보안대책:** 공격자가 보낸 패킷의 TCP를 분석해 TCP의 소스 IP, 목적 IP, 포트가 동일한지 확인한다.(Source 주소가 내부 IP인 패킷 차단), 공격자가 보내는 패킷의 횟수를 카운트 해 공격인정 시간 내에 공격인정 횟수가 많으면 LAND Attack으로 탐지한다. 라우터나 패킷 필터링 도구를 이용해 자신의 시스템 주소와 동일한 소스 주소를 가진 외부 패킷을 필터링한다. 공격정보를 네트워크, 시스템 관리자에게 알려 관리자가 침입 차단 시스템에서 소스 IP/Port와 목적지 IP/Port가 동일한 경우 차단하도록 설정한다.

**Ping of Death:** ping을 이용해 ICMP 패킷을 정상 크기(65,535 바이트)보다 아주 크게 만드는 것이다. 큰 패킷은 라우팅되어 공격 네트워크에 도달하는 동안 작은 조각(fragment)로 쪼개진다. 공격 대상은 조각화된 패킷을 처리하느라 정상적인 ping 처리보다 부하가 훨씬 많이 걸린다.

**Ping of Death 보안대책:** 패치를 통해 시스템을 최신으로 유지한다. 또 일반적인 ICMP는 분할하는 경우가 적으므로 패킷 중 분할이 일어난 패킷을 공격으로 의심해 탐지하는 방식을 사용한다.

**Teardrop Attack:** IP가 정상적으로 패킷을 전송할 때 IP 단편화가 발생한다. 수신자는 재조립을 통해 단편화된 데이터를 복구한다. 재조합 과정에서 정확한 조립을 위해 오프셋을 더하게 되어 있는데, 오프셋 값을 단편화 간에 중복되도록 고의로 수정하거나 정상적인 오프셋 값보다 큰 값을 더해 오버플로우를 일으켜 시스템 기능을 마비시키는 DoS 공격의 하나이다. 공격당한 시스템은 네트워크 연결이 끊어지거나 죽음의 푸른 화면(Blue Screen of Death)라 불리는 화면 오류를 표시하면서 중단되게 된다.

**+ Targa 공격:** 여러 DoS 공격을 실행할 수 있도록 만든 공격 툴이다. bonk, jolt, land, nestea, newtear, syndrop, teardrop, winnuke 등을 지원한다.

**+ Open Tear Attack:** 패킷을 전송할 때 데이터 없이 단편화 신호만 보내 타겟이 재조합 과정에서 오류를 발생시키도록 하는 DoS 공격이다.

**Teardrop Attack 보안대책:** teardorp은 침입방지, 방화벽을 우회할 수 있고 보인크(Boink)와 같은 다양한 변종을 가지는 공격이기 때문에 완전히 차단하기는 힘들다. 근본적인 해결을 위해서는 타겟 시스템과 운영체제가 취약점을 갖지 않도록 지속적으로 업데이트를 해야 한다.

**Inconsistent Fragmentation 공격:**

1. **Bonk:** 패킷을 프래그먼트 해 전송할 때 패킷을 조작해 타겟의 시스템 부하를 늘리는 공격이다. 첫 패킷을 1번으로 보낸 후 다음 패킷의 순서번호도 모두 1번으로 조작해 전송하는 DoS 공격이다.
2. **Boink:** Bonk를 수정한 DoS 공격도구이다. 패킷을 정상적으로 보내다가 갑자기 패킷 시퀀스 번호를 비정상적인 상태로 보내는 기술이다.

**+ Teardrop, Bonk, Boink의 공통점:** 세 공격은 모두 타겟의 오류 제어 로직을 악용해 시스템 자원을 공격하는 메커니즘이다. TCP프로토콜은 데이터 전달의 유효성이나 효율성을 위해 시퀀스 넘버 기반의 오류 제어 방식을 사용한다. TCP는 신뢰성 있는 연결을 위해 1. 패킷 순서 확인 2. 손상 패킷 유무 확인 3. 손실된 패킷의 재전송 요구 기능을 제공한다.

**봉크, 보인크 보안대책:** 봉크와 보인크 공격은 최근 시스템에 거의 통하지 않는다. 일부 응용프로그램에서는 공격이 먹힐 수 있다. 공격에 대한 대응책은 Ping of Death나 SYN Flooding의 대책과 같다.

**+ IP Fragmentation 취약점을 이용한 침입차단 시스템 우회공격:**

1. **Tiny Fragment:** IP헤더보다 작은 단편화를 만들어 방화벽을 우회해 내부 시스템에 침입하는 공격 기법이다. DoS 공격이 아닌 우회 기법이다.
2. **Fragment Overlap:** IP fragment의 offset값을 조작해 서비스 포트 필드를 중첩시켜서 재조합시킨 뒤 방화벽에서 허용하지 않는 서비스에 접근 가능하게 만드는 공격 기법이다. 위와 마찬가지로 우회 기법이다.

**DDoS 공격**

**1999~** SMURF, LAND Attack, Ping of Death, TearDrop 등은 비교적 단순한 DoS 공격이다. 이후 발전한 DDoS 공격 기법이 등장했다.

**2005~** DDoS 방어 인프라를 압도하기 위해 봇넷으로 대규모 트래픽을 유발하는 DDoS/DRDoS 공격 및 DDoS-For-Hire 등장

**현재** Smart Attack, Memcached Reflection Attack, Carpet Bombing 등 대응시스템을 우회, 압도하는 고도화된 DDoS 공격 등장, 랜섬 디도스(Ransom DDoS) 등장.

**DDoS란 다수의 서버, PC를 이용해 비정상적인 트래픽을 유발시켜 타겟의 시스템을 마비시키는 공격이다.**

공격자는 취약한 서버를 공격해 악성코드를 배포한 뒤 유포지/경유지에서 악성코드를 내려받은 장비들을 이용해 봇넷을 구축한다. 이후 공격자가 봇넷에 명령을 전달해 대규모 트래픽을 유발하거나 취약한 서버를 악용해 반사공격(DRDoS 공격)을 수행한다. 공격자의 위치와 구체적인 발원지를 파악하는 일은 거의 불가능하다.

|  |  |
| --- | --- |
| DDoS 구성 요소 | 설명 |
| 공격자, 봇 마스터 | C&C(Command&Control) 서버에 공격 명령을 전달하는 해커(== 봇 마스터) |
| 마스터, C&C 서버 | 공격자의 명령을 받는 시스템, 여러 대의 에이전트를 관리 |
| 핸들러 프로그램 | 마스터 시스템의 역할을 수행 |
| 에이전트 | 슬레이브, 좀비라고도 함, 공격대상에 직접적인 공격을 가하는 시스템으로 악성코드에 감염된 시스템이다. |
| 데몬 프로그램 | 에이전트 시스템 역할을 수행하는 프로그램 |
| 표적 | 공격 대상 |

**전통적인 DDoS 공격**

1. **트리누(Trinoo):** 1999년 6~7월 사이 퍼졌다. 미네소타 대학 사고의 주범. 솔라리스 2.x 시스템에서 발견되었으며 최소 227개 시스템이 공격에 쓰였다. 몇 개의 서버(혹은 마스터들)와 다수의 클라이언트(데몬)으로 이뤄져 UDP flood 서비스거부 공격을 유발하는 도구이다.
2. **TFN(Tribed Flood Network)공격:** 트리누와 유사한 DDoS 도구이다. 많은 소스에서 하나 혹은 여러 개의 목표 시스템에 서비스거부 공격을 수행한다. 믹스터가 개발한 트리누의 발전형태이다. TFN은 UDP Flood에 더해 TCP SYN Flood, ICMP Echo 요청 공격, SMURF(ICMP Broadcast) 공격을 할 수 있다.
3. **Stacheldraht 공격:** 독일어로 철조망이라는 뜻이다. 1999년 10월 출현. 트리누와 TFN을 참조해 제작된 도구이다. 마스터 시스템이 자동으로 업데이트되는 에이전트 데몬 사이에서 암호화 통신을 하는 기능이 추가되었다.
4. **TFN2K 공격:** TFN의 발전형태. 특정 포트가 사용되지 않는다. 암호화. 프로그램에 의해 TCP, UDP, ICMP가 복합적으로 사용되며 포트도 임의 결정된다. 지정된 TCP포트에 백도어를 실행시킬 수 있다.

**최신 DDoS 공격유형별 분류**

|  |  |  |  |
| --- | --- | --- | --- |
| 구분 | 대역폭 공격 | 자원 소진 공격 | 웹/DB 부하 공격 |
| 공격 특성 | 높은 bps | 높은 pps  높은 connection | 높은 pps  높은 connection |
| 공격 유형 | UDP Flooding,  UDP 기반 반사공격(DNS, NTP, CLDAP, SSDP…)  ICMP Flooding | TCP SYN,  ACK Flooding 등 | GET Flooding,  POST Flooding 등 |
| 피해 대상 | 동일 회선 사용하는 모든 시스템 | 대상 서버, 네트워크 장비(과부하) | 대상 웹/DB 서버 과부하 |
| 프로토콜 | UDP, ICMP, TCP | TCP | HTTP, HTTPS |
| IP 위/변조여부 | 위/변조 가능 | 위/변조 가능 | 위/변조 불가(실제 IP로 공격) |
| 비고 | 회선 대역폭 작으면 방어 어려움(일시에 대량 트래픽) | 대역폭 공격에 비해 적은 트래픽으로 서버 과부하 유발 | 정상세션을 맺을 뒤 과도한 HTTP 요청으로 웹/DB 서버 과부하 유도 |

1. **대역폭 공격(1) – UDP, ICMP Flooding**

**UDP Flooding:** 단순한 방식, 강력한 효과. 출발지 IP를 위/변조해 UDP 프로토콜로 대규모 데이터를 타겟에게 전달한다. 대상 서버에서 UDP Port를 사용하지 않아도 공격할 수 있다.

**ICMP Flooding:** 피해서버에 대량의 ICMP Request 패킷을 전달해 피해서버의 대역폭을 고갈시킨다. 반사공격과 다르게 증폭이 발생하지 않기 때문에 공격기기의 대역폭에 따라 공격규모가 결정

1. **대역폭 공격(2) – DRDoS(Distributed Reflection Denial of Service)**

분산 반사 서비스 거부 공격(DRDoS, Distributed Reflection Dos)는 기존의 DDoS보다 발전한 새 분산 서비스 거부 공격 방법이다. 별도 에이전트 설치 없이 프로토콜 구조의 취약점을 이용해 공격한다. DRDoS는 다양한 방식으로 나타낼 수 있다. 몇 가지 유형을 살펴보자.

1. TCP 3way-handshake의 취약점을 이용, 위조된 주소의 SYN요청을 반사 서버로 전달해 SYN+ACK이 공격 대상을 향하도록 하는 방법
2. 위조된 주소의 Echo Request를 반사서버로 전달해 ICMP 프로토콜의 Echo Reply가 공격대상으로 향하도록 하는 기법
3. UDP 프로토콜 서비스를 제공하는 서버를 반사서버로 이용해 응답이 타겟을 향하도록 하는 기법

UDP 프로토콜을 사용하는 DNS, NTP, SNMP, CHARGEN 등의 서비스를 이용한 DRDoS는 크게 반사와 증폭 공격 형태로 나타난다.

**UDP를 이용한 DRDoS 공격 유형**

1. **DNS 증폭 DRDoS 공격:** DNS 서버(반사서버)에 많은 레코드 정보를 요구하는 DNS. 질의타입(ANY, TXT)을 요청해 타겟의 트래픽을 증폭시킨다.
2. **NTP 증폭 DRDoS 공격:** NTP서버(반사서버)에 최근 접속한 클라이언트 목록(monlist 명령)을 요청해 트래픽 증폭
3. **SNMP 증폭 공격:** SNMP agent에 MIB같은 정보를 대량 요청(GetBulkRequest)해 공격 대상자에게 대량 트래픽 유발
4. **CHARGEN 증폭 공격:** CHARGEN 서버 접속 시 대량의 문자열을 전송해 대량 트래픽 유발

**DRDoS 공격의 위협요소(일반 DDoS 공격과 차이):**

1. 패킷의 전송 경로가 무수히 많다. 공격자가 전송한 위조 SYN 패킷은 즉시 흩어진다. 네트워크에 있는 수많은 TCP 반사 서버로 보내지게 되는 것이다.
2. 반사서버의 단계적 사용 및 확산. 반사서버를 단계적으로 늘려 공격 경로를 변경할 수 있다.
3. 위조 SYN 패킷 근원지의 IP를 랜덤으로 생성하는 전통적 DoS 공격과 다르게 공격대상 IP를 근원지 IP로 위조해 역추적하기가 쉽지 않다.

**DRDoS 공격 종류:**

1. **DNS Reflection Attack:** 공격자가 피해자 IP로 스푸핑해 네임 서버에 비정상 DNS 질의요청을 하고 네임서버는 DNS 응답값을 타겟으로 전송해 타겟의 대역폭을 고갈시킨다. 공격 트래픽을 높이기 위해 단순 DNS Query가 아닌 Zone의 모든 정보를 요청하는 ANY Type 레코드를 요청하는 특징이 있다. DNS 증폭 DRDoS의 반사서버로 DNS 서버가 악용되지 않도록 하기 위해 다음을 점검한다.
2. 공개용이 아닌 내부 사용자용 DNS(Revolving) 서버라면 서버 설정을 통해 내부 사용자 주소만 Recursive Query가 가능하도록 제한한다.
3. 서버 방화벽(Iptables) 또는 네트워크 보안 장비를 통해 특정 바이트 이상의 DNS 질의에 대한 응답을 차단하거나 동일 IP에 대해 초당 요청 개수 제한(Limit 설정)을 설정한다.
4. **NTP Reflection Attack:** 시간 동기화에 사용하는 NTP(Network Time Protocol) 서버를 반사서버로 이용한 공격이다. 스푸핑된 타겟의 IP로 NTP서버에게 가능한 많은 응답 패킷을 만들어 내도록 monlist를 요청하는 비정상 패킷을 보내 응답값을 공격 payload로 활용하는 전략이다. monlist 구버전 NTP서버에서 사용하는 명령어로 최대 600개의 접속 호스트에 대한 정보를 응답받을 수 있는 명령이다.(v2.4.7 이상에서 삭제)
5. **CLDAP Reflection Attack:** CLDAP(Connection-less Lightweight Directory Access Protocol)은 네트워크상의 디렉터리를 연결/검색/수정하기 위해 사용하는 프로토콜이다. CLDAP 서버를 반사서버로 악용하면 CLDAP Reflection Attack이 된다. 해당 서버에 비정상적인 쿼리를 보내 응답값을 공격패킷으로 활용한다.
6. **SSDP Reflection Attack:** Simple Service Discovery Protocol은 UPnP(Universal Plug and Play)장치를 탐색할 때 주로 사용하는 프로토콜이다. 네트워크 상의 다른 장치를 찾거나 알리는 역할도 수행한다. 현재 다양한 IoT기기에 널리 활용된다. SSDP 어택은 가능한 한 많은 데이터를 요청하는 search 명령을 보내 스푸핑된 서버 IP로 대규모 응답이 가게 만드는 공격이다.
7. **Memchached Reflection Attack:** Memchashed 서비스는 DB 부하감소 및 응답속도 증가를 위해 분산된 메모리에 데이터를 캐싱하는 서비스이다. 내부에서만 접근하도록 설계되었으며 Key값으로 Data를 매핑한다. 맴캐쉬드 반사 공격은 멤캐쉬드 서버의 기능을 악용해 가능한 한 많은 request명령으로 캐싱 데이터를 요청하고 이를 스푸핑된 타겟 IP로 향하게 한다.
8. **CoAP Reflection Attack:** Constrained Application Protocol은 IoT 기기들 중 저전력 컴퓨터들을 위해 만들어진 간단한 UDP 프로토콜이다. HTTP 형식과 유사하며 UDP 프로토콜의 5683번 포트를 사용한다. CoAP 반사 공격은 타겟으로 스푸핑한 출발지 IP에서 외부에 노출된 IoT 장치들을 대상으로 GET Request를 보내고, 돌아오는 응답 패킷을 타겟으로 보내 타겟의 대역폭을 고갈시킨다.

**자원 소진 공격:**

1. **SYN Flooding:** TCP 프로토콜의 3-way handshake(이하 TCP 3way)를 악용, SYN Flag만 지속적으로 전달하고 돌아오는 SYN/ACK 패킷에 응답하지 않아 피해서버의 자원을 소모하게 만든다. bps에 비해 pps가 높은 게 특징. 대역폭 공격과 다르게 대규모 트래픽을 발생시키지 않아도 서비스 접속불가를 유도할 수 있다.
2. **ACK Flooding:** 위와 유사하게 3-way handshake를 악용해 다량의 ACK Packet을 타겟에게 보내 자원을 소모하게 만든다.
3. **DNS Query Flooding:** DNS서버에 다량의 질의를 보내 정상 서비스를 할 수 없게 만든다.

**웹/DB 부하공격:**

1. **GET Flooding:** 공격자가 TCP 3way를 통해 서버와 세션을 맺고 HTTP GET 메소드 요청을 통해 웹서버의 자원을 소진시키면서 DB서버의 자원까지 소진시켜 정상적 서비스 이용을 차단한다.
2. **Slowloris Attack(Slow HTTP Header DoS):** GET Flooding에 비해 상대적으로 규모가 작다. 단시간에 폭발적으로 요청하는 공격이 아니라 장기간 지속적으로 공격을 수행한다. 정상 HTTP 헤더는 Carriage Return & Line Feed(이하 개행문자)가 두 번 나타난다. 첫 개행은 헤더의 종료, 두 번째 개행은 전체 헤더의 종료이다. 헤더의 마지막에 개행을 하나만 전송하면 서버는 헤더가 모두 도착하지 않은 것으로 간주해 연결과 대기 상태를 유지한다. 지속적인 자원 잠식이 이뤄지는 것이다.
3. **RUDY Attack(Slow HTTP POST DoS):** R-U-Dead-Yet 공격이다. POST Method를 이용하는 대표적인 Slow공격이다. POST는 전달할 데이터 크기를 Content-Length 헤더에 삽입해 보내는데 RUDY Attack은 이 크기를 매우 크게 설정한 후 서버로 전달할 데이터를 장시간동안 조금씩 분할해 전달하게 한다.
4. **Slow read Attack:** TCP 통신에서 사용하는 windows size 악용 공격. Client마다 한 번에 처리할 수 있는 패킷 크기가 다르기 때문에 해당 크기를 windows size 필드를 통해 전달하며, window size를 기준으로 패킷을 주고받게 된다. slow read 공격은 window size를 낮게 해 서버로 전달하고 해당 size를 기준으로 데이터 전송이 완료될 때까지 커넥션을 유지하게 만들어 서버의 커넥션 자원을 고갈시키는 공격이다.
5. **GET Flooding with Cache**-**Control(CC Attack):** 웹서버는 일반적으로 캐싱 서버를 운영해 접근이 활발히 이루어지는 데이터를 캐싱서버가 응답하도록 구축한다. 캐시 컨트롤을 통한 GET 플러딩은 HTTP 메시지의 캐시옵션(no-store, must-revalidate)를 사용해 캐싱서버를 무력화하고 웹서버가 직접 처리하도록 유도해 자원을 소진시킨다.
6. **동적 HTTP Request Flooding 공격**: 겟 플러딩이나 HTTP CC 공격은 지정된 웹 페이지를 지속적으로 요청하는 공격이다. 방화벽은 이 두 공격의 특징적 HTTP 요청 패턴을 구분하고 방어할 수 있다. 동적 HTTP Request Flooding은 이 차단기법을 우회해 지속적으로 요청 페이지를 변경해가며 리퀘스트를 넣는 기법이다.
7. **기타 서비스 마비 공격**
8. **해시도스(HashDoS) 공격:** 웹 서버는 클라이언트로부터 수신한HTTP 메시지의 파라미터 관리를 위해 해시테이블을 사용한다. 다량의 메시지와 조작된 파라미터는 인덱스로 사용되는 해시값에 충돌을 발생시켜 정확한 해시값을 찾기위해 모든 해시테이블을 검사하게 만든다. 이런 방식으로 웹 서버가 CPU자원을 소진하게 만든다.
9. **헐크도스(HulkDoS) 공격:** HULK(HTTP Unbearable Load King) DoS는 웹서버의 가용량(접속가능한 최대 클라이언트 수)을 모두 사용하도록 해 정상적 서비스가 불가능하도록 유도하는 GET Flooding 공격 유형이다. 타겟의 웹사이트 주소를 지속적으로 변경해 DDoS 차단정책을 우회하는 특징을 가진다.

**+ SIP Flood:** VoIP(Voice over IP) 전화연결에 많이 사용하는 프로토콜이SIP(Session Initiation Protocol)이다. SPI Flood는 하나의 INVITE request 메시지가 상당한 자원을 소모한다는 사실을 악용한다.

**DDoS 예방:**

1. **DDoS 대응 서비스 가입:** 중소기업은 한국인터넷진흥원이 제공하는 사이버대표소를 이용할 수 있다. 1Gbps 이상의 대규모 DDoS 공격에 대응하기 위한 자체 인프라 구축은 인력과 비용 측면에서 현실적인 한계가 있다. 인터넷 회선 제공업체(ISP)나 클라우드 서비스에서 제공하는 DDoS 방어서비스를 이용하는 게 좋다.
2. **백업 서버 구축:** 서버 장애에 대비해 중요 서버들은 서로 다른 회선으로 이중화 구성을 해야 한다. Master 서버의 회선에 DDoS 공격으로 인한 장애가 발생해도 다른 회선에 있는 Slave 서버가 동작해 서비스 장애를 최소화 할 수 있다.
3. **공격 대상의 최소화:** 외부로 노출된 웹서비스 외 기업 내부용 서버는 노출하지 않고 내부망으로 망 분리 조치 후 운영한다. 내부용 서버는 DDoS 공격 및 기타 사이버 공격을 예방하기 위해 기업의 내부 서버 IP 및 서비스가 외부에 열려있는지 주기적으로 스캐닝하고 Showdan, 다크 웹 노출 여부를 확인한다. 부득이하게 외부에서 내부망으로 접속이 필요할 경우 DDoS 대응 및 방어 설정이 가능한 가상사설망(VPN) 등의 별도 서비스를 이용한다.

**DDoS 방어:**

1. **자체 방어:** 자체 방어를 위해서는 충분한 회선용량과 DDoS 방어 장비가 있어야 한다. 그렇지 않다면 소규모 공격에 한해 일부 방어가 가능하다.
2. **TCP 기반 DDoS 방어:** 방화벽 및 프록시 서버와 같은 주변 장치에서 TCP Keepalive 및 최대연결을 설정해 SYN Flooding 등과 같은 공격을 방어할 수 있도록 준비한다.
3. **이상 트래픽 모니터링:** 기본 트래픽(bps, pps, 동시접속수)를 인지하고 이상트래픽 발생을 주기적으로 모니터링한다.
4. **Whitelist 방식의 방화벽 설정:** 필요 서비스에 최소의 정책만 오픈하고 모든 프로토콜과 포트를 차단(All Deny)하는 설정을 통해 기타 포트로 들어오는 공격에 대한 방어가 가능하다.(단, 서비스하는 모든 포트 정보를 반드시 알고 있어야 함)
5. **서버 분산을 통한 가용성 확보:** Load Balancer를 이용하여 서버 부하를 분산해 서버 장애를 최소화한다.
6. **DDoS 대응서비스를 통한 방어:**
7. **0단계:** 사전준비, DDoS 방어서비스를 알아보고 절차와 시간 등을 미리 파악한다.
8. **1단계:** 공격인지, 웹사이트 모니터링 시스템을 구축해 주기적으로 모니터링을 수행하고 이상정보 탐지 시 DDoS 공격에 대한 현상인지 서버 자체의 원인인지 파악한다.
9. **2단계:** 공격 정보 파악, DDoS 공격으로 판단될 경우 공격의 정보(유형, 규모, 시간)를 파악해야 한다. 공격유형은 크게 1.대역폭공격 2.자원소진공격 3.웹/DB부하 공격으로 나뉜다.
10. **3단계:** 방어서비스 적용, DDoS 공격에 의한 서버 장애 발생시 신속히 방어 서비스 프로세스를 수행한다.
11. **4단계:** 서비스 모니터링, 공격은 차단했으나 이미 점유된 세션 혹은 서버 부하로 인해 서비스 복구가 느릴 수 있다. 따라서 DDoS 공격을 차단한 후 서비스 및 장비 모니터링을 통해 이미 유입된 공격에 의한 장비상태를 확인 후 조치한다.
12. **5단계:** 사후조치, 공격이 종료된 후 공격을 받은 시스템을 다시 점검해 특이사항이 없는지 확인한다. 프로세스를 보완해 추후 발생 가능한 공격에 대비한다.

**DNS 싱크홀(DNS Sinkhole):** 조종자를 탐지하고, 감염된 PC와 조종자의 접속을 차단해 2차 피해를 예방하는 시스템이다. 감염된 PC에서 특정 주소로 연결을 할 때 실제 해당 주소로 데이터가 전송되지 않고 싱크홀 네트워크가 대신 응답해 패킷이 외부로 전달되지 않도록 처리한다. 이 방법을 DNS에 적용한 게 DNS 싱크홀이다. 한국인터넷진흥원(KISA)에서 국내 주요 ISP업체 및 가입기관과의 협력을 통해 DNS 싱크홀을 운영중이다.

**DNS 싱크홀 동작과정:**

1. KISA에서 배포한 C&C목록을 ISP등 DNS 싱크홀 적용기관의 DNS 서버에 업데이트 작업을 주기적으로 진행한다.
2. 악성 봇에 감염된 PC가 싱크홀이 적용된 DNS에 C&C 서버에 대한 질의를 요청
3. DNS는 악성 봇 PC에 싱크홀 IP주소를 반환한다.
4. 이를 통해 좀비 PC는 C&C가 아닌 싱크홀 서버로 접속해 공격자의 명령으로부터 피해를 방지할 수 있다.

**네트워크 스캐닝:** 네트워크 스캐닝이란 사이버 공격을 위한 준비활동을 말한다. 풋프린팅, 스캐닝, 목록화의 3단계 과정을 거친다.

**풋프린팅:** 공격자가 타겟의 다양한 정보를 수집하기 위해 널리 사용하는 방법이다. 사회공학 기법이 일반적으로 이용된다. 공격대상이 스스로 공개한 여러 정보를 풋 프린팅 해 공격대상의 IP블록, 사용자 이름 및 계정, 전화번호, 메일 주소 등의 정보를 수집한다.

**스캐닝:** 스캐닝은 실제 공격방법을 결정하거나 공격에 이용할 수 있는 네트워크 구조, 시스템이 제공하는 서비스 등의 정보를 얻기 위해 수행한다. 공격자는 네트워크 보안장비 사용 현황, 우회 가능 네트워크 구조, 해당 네트워크 내의 시스템 플랫폼 형태, 운영체제의 커널 버전, 제공 서비스의 종류 등의 정보를 파악한다.

**포트 스캔 분류:** 스캐닝 기법은 Sweep, Open Scan, Stealth Scan, Security Scan으로 분류할 수 있다.

**포트 스캔 종류:**

1. **Sweep:** 특정 네트워크에 속해있는 시스템의 작동 유무를 판단할 수 있는 기법이다. 이를 통해 타겟 기관에서 사용하거나 소유하고 있는 IP주소와 네트워크 범위를 알아낼 수 있다. Sweep은 요청에 의한 응답을 수행하는 클라이언트/서버 구조를 기반으로 하며 동작은 요청에 대한 응답 여부를 판단하는 방법을 사용한다. Sweep의 종류에는 사용하는 프로토콜에 따라 ICMP Sweep과 TCP Sweep, UDP Sweep의 세 가지 기법이 있다.
2. **Open 스캔:** 시스템 활성화 여부 뿐 아니라 스캔하는 포트의 서비스가 활성화되었는지 여부도 알 수 있다. 포트를 스캔해 포트가 열려있으면 해당 시스템이 활성화되어 있는 것으로 판단할 수 있을 뿐 아니라 포트가 제공하는 서비스를 예측할 수 있다. Open스캔 방법은 사용하는 프로토콜에 따라 TCP Open 스캔과 UDP Open Scan 기법으로 나눈다. 둘 다 스캔하려고 하는 목적 포트로 연결을 시도한 후 응답 형태를 보고 포트의 활성화 여부를 판단한다.
3. **TCP 스캔:** TCP Open 스캔은 연결을 완료하느냐 완료하지 않느냐에 따라 Full 스캔과 Half 스캔으로 구분한다.

**TCP Full Open 스캔(TCP Connect 스캔)**

1. 포트가 열려 있으면 대상시스템으로부터 SYN/ACK 패킷을 수신해 그에 대한 ACK 패킷을 전송함으로써 연결을 완료한다.
2. 포트가 닫혀 있으면 대상 시스템이 연결 요청을 받아들이지 못하므로 RST/ACK 패킷을 전송한다.
3. TCP Full Open 기법은 스캔하고자 하는 포트에 접속을 시도해 완전한 TCP 연결을 맺어 신뢰성 있는 결과를 얻을 수는 있으나 속도가 느려 로그를 남기므로 탐지가 가능하다는 단점이 있다.

**TCP Half Open 스캔(TCP SYN 스캔)**

1. 세션에 대한 로그가 남는 풀 오픈 스캔의 단점을 보완하기 위해 나온 기법이다. 로그를 남기지 않아 추적을 어렵게 하는 기법이다.
2. 세션을 완전히 연결하지 않고 TCP Half Connection만으로 포트 활성화 여부를 판단한다.
3. 먼저 공격자가 SYN패킷을 공격대상 서버에 보낸 후 포트가 열려있는 서버로부터 SYN/ACK 패킷을 받으면 공격자는 즉시 RST 패킷을 보내 연결을 끊는다. 연결 시도 세션에 로그가 남지 않도록 하는 것이다. 그러나 공격자의 SYN 세그먼트 전송기록은 남아 스캐닝 공격사실을 완전히 숨길 수는 없다.
4. **UDP 스캔:** UDP는 비연결 지향 전송 프로토콜이다. 송수신자 간에 연결이 없기 때문에 ICMP Port Unreachable 에러 메시지를 통해 포트 활성화 유무를 판단할 수 있다. 스캔하고자 하는 포트를 대상으로 UDP 연결을 시도할 때 ICMP Port Unreachable 에러 메시지가 수신되면 해당 포트는 비활성화 된 것으로 생각한다. 아무런 응답이 없다면 포트 활성화를 의미한다. UDP를 이용한 포트 스캔은 신뢰하기 어렵다. UDP 패킷이 네트워크를 통해 전달되는 동안 라우터나 방화벽을 통해 손실될 수 있기 때문이다.
5. **Stealth 스캔:** TCP Full Open 스캔이나 TCP Half Open 스캔처럼 handshaking 연결 기법을 사용하는 게 아니라 TCP 헤더를 조작해 특수한 패킷을 만든 후 스캔 대상 시스템에 보내 그 응답으로 포트 활성화 여부를 알아낸다. 종류에는 TCP FIN 스캔, TCP ACK 스캔, TCP 단편화 스캔 등이 있다.
6. **FIN, NULL, XMAS 스캔:** TCP FIN(Finish) 스캔은 TCP 헤더 내에 FIN 플래그를 설정해 공격대상 시스템에 메시지를 전송하면 스캔 대상 시스템의 포트가 열려 있을 경우에는 응답이 없고 포트가 닫혀있는 경우에만 RST패킷이 되돌아오게 된다.

**NULL과 XMAS 패킷**을 전송하는 경우에도 이와 같은 결과를 얻을 수 있다. NULL패킷은 TCP 헤더 내에 플래그 값을 설정하지 않고 전송하는 패킷을 말한다. XMAS 패킷은 TCP 헤더 내에 ACK, FIN, RST, SYN, URG 플래그를 모두 설정해 전송하는 패킷이다. XMAS라고 불리는 이유는 크리스마스트리처럼 제어비트를 반짝거리게 설정했다는 의미이다. 이렇게 설정해 전송한 두 패킷 역시 포트가 열려있을 때는 응답이 없고 포트가 닫혀있을 경우에만 RST패킷이 돌아온다.

1. **TCP ACK 스캔:** 포트의 오픈 여부를 판단하는 게 아니라 방화벽의 룰셋(필터링 정책)을 테스트하기 위한 스캔이다. 다음 요소를 체크한다.
2. 대상 방화벽이 상태 기반(Stateful)인지
3. 대상 포트가 방화벽에 의해 필터링 되고 있는지
4. **Decoy 스캔:** 타겟이 스캐너의 주소를 식별하기 어렵도록 스캐너 주소 외에 다양한 위조 주소로 스캔하는 방식을 말한다.

|  |  |
| --- | --- |
| **종류** | **설명** |
| **TCP Connect 스캔** | TCP세션이 수립되는지 포트 오픈 확인 |
| **TCP Half Open 스캔(SYN 스캔)** | TCP 연결요청만(SYN 패킷만 전송)을 수행, 세션 수립 되지 않음 |
| **TCP ACK 스캔** | TCP ACK 플래그를 ON으로 설정해 패킷을 송신 |
| **TCP FIN 스캔** | TCP FIN플래그를 ON으로 설정해 패킷을 송신 |
| **TCP Null 스캔** | TCP 모든 플래그를 OFF으로 설정해 패킷을 송신 |
| **Xmas 트리 스캔** | TCP 모든 플래그를 ON으로 설정해 패킷을 송신 |
| **UDP 스캔** | UDP 패킷을 송신 |

**대표적인 스캔 도구:**

1. **NMap(Network Mapper):** 모든 운영체제에서 사용 가능, 운영체제 종류 및 사용 서비스에 대한 정보를 알 수 있음. FTP 서버의 취약점을 이용한 바운스 공격 가능. 오픈소스로 인한 확장성

**Nmap사용법: 기본 사용법, 옵션 외워두기(-s(scan),p(port),v(상세 내역), d(debugging), o(output+ N(일반 파일), +X(XML파일), +G(Grepable[grep, awk로 분석하기 편한 파일], +A(이전 세개 옵션 모두 적용한 파일 3개 디렉터리 생성))))**

1. **운영체제의 탐지(OS Fingerprinting):** 운영체제마다 취약점이 다르기 때문에 운영체제와 버전을 파악하는 일을 OS 핑거프린팅이라고 한다. OS버전에서 명시적으로 정의하지는 않지만 구현 방법이 다른 기능의 차이를 통해 OS의 버전을 파악할 수 있기 때문에 핑거프린팅이라고 한다.
2. **탐지방법:** TCP 연결설정의 시작 순서번호(ISN; Initial Sequence Number) 설정 방식은 버전마다 차이가 있다. 임의 번호의 범위 조차도 OS마다 차이가 있다. IP 패킷의 TTL도 OS마다 디폴트 값이 다르며 IP패킷 서비스 유형을 표시하는 ToL(Type of Service) 필드를 구현하는 방식도 차이가 있다. 이외에도 많은 부분의 구현에서 차이를 보인다. **배너 그래빙(Banner Grabbing)**이라는 방식도 있다. 배너는 텔넷처럼 원격지 시스템에 로그인하면 뜨는 안내문같은 것이다. 배너 그래빙은 이 배너를 확인해 OS 버전을 알아보는 방식이다. **+서버 프린팅:** OS 프린팅과 유사하게 웹과 메일 서버같은 중요 서버 유형도 비슷한 방식으로 스캐닝할 수 있다.
3. **목록화(Enumeration):** 풋 프린팅과 스캐닝 작업을 통해 수집한 정보로 더 실용적인 정보를 얻는 과정이다. 수집한 정보를 실제 공격에 사용할 수 있도록 목록화한다. 풋 프린팅과 스캐닝까지가 문을 열기 위한 작업이었다면 목록화는 문을 열고 시스템에 실제 침입해서 이루어지는 작업이다. 목록화는 공유 자원 목록화, 사용자 및 그룹 목록화, 응용 프로그램 목록화로 구분된다.
4. **포트 스캔에 대한 대응책:** 포트 스캔 자체는 단순 조사이기 때문에 실제 손해가 발생하지는 않는다. 만약 시스템에 취약점이 있다면 공격을 받아 피해를 볼 우려가 있다.
5. 불필요한 패킷을 차단하도록 방화벽 설정
6. 사용하지 않는 포트는 닫아둔다.(불필요한 서비스 종료)
7. 침입방지시스템(IDS)을 사용해 포트 스캔 감지
8. 시스템 로그 검사

**위 방법을 사용해 포트 스캔에 대비할 수 있다.**

**스니핑:** 네트워크 트래픽을 도청하는 과정을 스니핑이라고 한다. TCP/IP 프로토콜을 이용한 통신에서 통신매체를 통하는 패킷이 암호화되지 않은 상태이므로 해당 패킷을 도청해 메시지 내용을 볼 수 있다.

**스니핑의 종류:**

1. **허브 환경에서 스니핑:** 허브는 들어온 패킷을 패킷이 들어온 포트를 제외한 모든 포트에 보내는 장비이다. 따라서 허브를 사용하면 의도와 관련없이 지속적으로 패킷을 받게 된다. 네트워크 드라이버와 OS 커널 수준에서 MAC주소를 보고 이를 필터링하기 때문에 일반적인 사용자는 그런 점을 알 수 없다. 하지만 시스템의 NIC를 promiscuous 모드(유닉스/리눅스)로 동작하게 하면 다른 패킷을 버리지 않고 받아볼 수 있다. 스니핑 도구를 통해 이 패킷들을 저장하고 확인하는 것이 가능하다. 윈도우는 프러미스큐어스 모드를 위해 드라이버를 따로 설치해야 한다. 프러미스큐어스 모드를 동작시키면 실제 수신 대상이 아닌 호스트에게 패킷이 전달된다. IP와 맥주소와 관련 없이 허브는 브로드캐스트한다.
2. **스위치 환경에서의 스니핑:**
3. **스위치 재밍(= MAC Address Flooding, MACOF(MAC Overflow) 공격):** 스위치의 MAC Address Table의 버퍼를 오버플로우시켜 스위치가 허브처럼 동작하게 강제하는 기법을 말한다.

**+가용성vs보안(Fail Open(가용성 win), Fail Close(보안 win))**

스위치는 Fail Open 방식을 채택한다. 문제 발생 시 Hub처럼 연결된 모든 노드에 패킷을 전송한다. 스위치 재밍은 MAC 주소를 지속적으로 변경하면서 ARP Reply 패킷을 지속적으로 전송해 스위치의 MAC Address Table을 채우는 식으로 공격한다.

1. **ARP 스푸핑:** 공격자가 특정 호스트의 MAC을 자신의 MAC으로 위장한 ARP Reply 패킷을 타겟에게 지속적으로 전송해 타겟의 ARP Cache에 특정 호스트의 MAC 정보를 자신의 MAC 정보로 변경시키는 공격. 희생자가 특정 호스트에게 보내는 패킷을 공격자가 스니핑하게 된다.
2. **ARP 리다이렉트:** 공격자가 자신이 라우터인 것처럼 MAC 주소를 위조해 ARP Reply 패킷을 해당 네트워크에 브로드캐스트한다. 이를 통해 로컬 네트워크의 모든 호스트와 라우터 사이의 트래픽을 브로드캐스팅하고 IP Forward 기능을 통해 사용자들이 눈치채지 못하게 하는 기법이다.
3. **ICMP 리다이렉트:** 3계층에서 스니핑 시스템을 네트워크에 존재하는 또 다른 라우터라고 알려서 패킷의 흐름을 바꾸는 공격이다. 네트워크는 라우터나 게이트웨이가 하나인데 하나의 라우터로 감당할 수 없을 때 라우터나 게이트웨이를 두 개 이상 운영해서 로드 밸런싱을 해야 한다. 로드밸런싱은 시스템의 라우팅 테이블에 라우팅 엔트리 하나를 더 넣거나 ICMP 리다이렉트 방법을 이용한다.
4. 호스트A에 라우터 A가 기본 라우터로 설정되어 호스트 A가 원격 호스트 B로 데이터를 보낼 때 패킷을 라우터 A로 보낸다.
5. 라우터 A는 호스트B로 보내는 패킷을 수신하고 라우팅 테이블을 검색해 자신을 이용하는 것보다 라우터B를 이용하는 것이 더 효율적이라고 판단하고 해당 패킷을 라우터 B로 보낸다.
6. 라우터 A는 호스트 B로 향하는 패킷을 호스트 A가 자신에게 다시 전달하지 않도록, 호스트 A에 ICMP리다이렉트 패킷을 보내서 호스트 A가 호스트 B로 보내는 패킷이 바로 라우터 B를 향하도록 한다.
7. 호스트 A는 라우팅 테이블에 호스트 B에 대한 값을 추가하고, 호스트 B로 보내는 패킷은 라우터 B로 전달한다.
8. **ARP Redirect와 ICMP Redirect의 차이:** ARP 리다이렉트는 희생자의 ARP 캐쉬 테이블을 변조해 스니핑하고 ICMP Redirect는 희생자의 라우팅 테이블을 변조해 스니핑한다.
9. **스위치의 SPAN(Switch Port Analyzer)/Port Mirroring 기능 이용:** 해당 기능은 스위치를 통과하는 모든 트래픽을 볼 수 있는 기능이다. 관리 용도이지만 공격자가 물리적으로 해당 포트에 접근할 수 있으면 손쉽게 스니핑할 수 있다.

**스니핑 공격의 보안대책:** 능동적 대책과 수동적 대책으로 나뉜다. 능동적으로는 스니퍼를 탐지하는 방법이다. 내부자가 스니퍼를 몰래 설치할 방법이 다양하고 무선 환경에서는 스니핑 도구 탐지가 어려운 한계점이 있다. 수동적 대책은 통신 내용을 암호화해 스니핑이 되더라도 알아볼 수 없게 만드는 것이다. 보다 근본적인 대책이다.

**스니핑 탐지:** 스니퍼는 프러미스큐어스 모드에서 동작한다. 유입 패킷에 대한 IP 주소와 MAC주소 필터링을 하지 않는다는 의미이다.

1. **ping을 이용한 방법:** 대부분의 스니퍼는 일반 TCP/IP에서 동작해 Request를 받으면 Response를 전달한다. 이를 이용해 의심스러운 호스트에 ping을 보낼 때 네트워크에 존재하지 않는 MAC주소를 보내본다. 만약 ICMP Echo Reply를 받으면 해당 호스트가 스니핑을 하고 있는 것이다. 존재하지 않는 MAC을 사용했기 때문에 스니핑을 하지 않는 호스트는 누구도 ping Request를 볼 수 없어야 하기 때문이다.
2. **ARP를 이용한 방법:** 위와 유사한 방법으로 위조된 ARP Request를 보내 ARP Response가 오면 프러미스큐어스 모드가 설정된 것이다.
3. **DNS 방법:** 스니핑 프로그램은 편의를 위해 스니핑한 시스템의 IP주소로 Inverse-DNS lookup을 수행한다. 원격과 로컬에서 모두 사용할 수 있는 방법으로 원격에서 테스트 대상 네트워크로 Ping sweep을 보내고 들어오는 Inverse-DNS lookup을 감시해 스니퍼를 탐지할 수 있다.
4. **유인(Decoy) 방법:** 가짜 계정과 패스워드를 네트워크에 계속 뿌린다. 공격자가 이 계정과 패스워드를 이용해 접속을 시도하면 접속 시스템을 탐지해 스니퍼를 탐지할 수 있다.
5. **ARP watch:** 초기 MAC주소와 IP주소의 매칭 값을 저장하고 ARP트래픽을 모니터링해 이를 변하게 하는 패킷이 탐지되면 관리자에게 메일로 알린다. 많은 공격이 위조된 ARP를 사용해 쉽게 탐지할 수 있다.

**암호화:** 수동적 대책으로 스니핑을 당하기 전에 미리 암호화하는 것을 말한다.

1. SSL: 암호화된 웹 서핑, 40비트와 128비트 두 가지 암호화 키
2. PGP, PEM, S/MIME: 이메일을 전송할 때 사용하는 암호화
3. SSH: 텔넷과 같은 서비스 암호화
4. VPN: 공용 회선을 이용한 사설 암호화 망

**네트워크 스니퍼(Sniffer):** 패킷 또는 LAN 세그먼트상을 지나는 트래픽을 분석하는 프로그램 혹은 장비, 스니퍼는 무차별(프러미스큐어스) 모드로 동작하는 네트워크 어댑터를 이용해 네트워크에 접속해야 하며 이 어댑터의 드라이버는 데이터를 캡쳐한다. 패킷 스니핑 공격은 트래픽 통계를 분석하는 패킷 분석기(와이어샤크) 및 스니핑 공격을 통해 제작된 도구(Dsniff)를 사용한다.

**스푸핑:** 공격자가 자신을 노출시키지 않고 제3자인 것처럼 MAC, IP 주소 등을 속이는 작업. 트래픽을 중간에서 가로채는 중간자 공격 또는 대량의 트래픽을 유발해 특정 시스템을 마비시키는 DoS 공격에 활용된다. IP, 이메일, 웹, DNS 스푸핑 등이 있다.

**ARP 스푸핑:** ARP(Address Resolution Protocol)스푸핑은 호스트의 주소 매칭 테이블에 위조된 MAC주소가 설정되도록 한다. ARP 프로토콜에는 사용자가 ARP Reply를 수신하면 사용자 자신이 ARP Request를 송신 했는지와 관계 없이, 또 ARP Reply에 대한 인증도 없이 사용자 시스템의 ARP 테이블을 갱신한다는 취약점이 있다. 이를 이용한 공격이 ARP 스푸핑이다. 공격자가 주기적으로 ARP 스푸핑 공격을 수행해 ARP 캐시 테이블에 공격 정보를 유지시킨다. 이처럼 ARP 캐시 테이블의 내용을 몰래 변경하는 작업을 ARP Cache Poisoning이라고 한다. 공격자가 서버와 클라이언트 사이에서 서버와 클라이언트의 ARP 테이블을 조작한다.

**ARP 보안대책:** ARP 스푸핑은 회사의 DMZ를 관리하는 사람이 이용하면 성공률이 100%이다. 보안 사고의 50% 이상은 동일 랜에 연결된 내부자의 소행이다. 기본적인 대응은 네트워크 시스템에 대한 ARP 테이블은 관리자가 정적으로 작성하는 것이다. ARP Reply를 무시해 ARP Reply에 의해 자동으로 ARP Cache Table이 변경되는 것을 막을 수 있다. 정적 ARP Table 사용법은 관리자가 직접 ARP 테이블을 작성할 수 있을 정도로 네트워크 규모가 작아야 한다. 리부팅시 static 옵션이 사라지므로 배치 파일로 만들어 리부팅 시마다 자동 실행되도록 설정해야 한다.

* arp -s <IP주소> <MAC주소>

위 형식으로 주소를 입력하면 static 옵션을 설정할 수 있다.

**IP 스푸핑:** 공격자가 신뢰관계를 가진 시스템의 IP주소로 위장해 공격 대상 서버로부터 정보를 가로챈다(1995 캐빈 미트닉이 최초록 시도) 트러스트 관계 설정은 신뢰관계에 있는 IP를 등록해 해당 IP로 접근하는 것을 허용해주는 방식이다. IP가 확인되면 별도의 인증이 필요없다. 이는 다수의 시스템을 관리해야 하는 관리자가 아이디와 패스워드를 일일이 관리해야 하는 어려움을 해결하고 접속과정에서 아이디, 패스워드가 스니핑되는 것을 방지하지만 IP스푸핑에는 매우 취약하다. IP 스푸핑 공격은 이와 같은 취약점을 이용해 신뢰 관계인 IP로 접근한 후 rlogin을 이용해 공격대상 서버에 접속한 후 백도어를 설치해 공격 경로를 확보하는 식으로 활용된다.

ex) 서버와 타겟의 네트워크에 개입해 타겟에게는 TCP SYN Flooding으로 DoS 공격을 가하고 서버에는 타겟의 IP로 위장해 해킹한다.

**IP 스푸핑 대응방안:** 외부에서 들어오는 패킷 중 출발지 IP에 내부망 IP주소를 가지고 있는 패킷을 라우터에서 패킷 필터링을 통해 막을 수 있다. IP스푸핑에 가장 좋은 대책은 트러스트를 이용하지 않는 것이다. 부득이한 경우 트러스트된 시스템의 MAC주소를 static으로 지정해야 한다. IP스푸핑은 IP프로토콜이 출발지 IP에 대한 인증을 수행하지 않는 취약점을 이용하니 인증 기능이 강화된 IPSec을 사용해 대응할 수도 있다. TCP같은 상위 프로토콜에서 비정상적인 IP 주소 간 통신을 방지하는 기능을 추가할 수도 있다.

**DNS 스푸핑:** 실제 DNS 서버보다 빠르게 DNS Response를 보내 타겟이 잘못된 IP 주소로 접속을 하도록 유도하는 공격이다. ARP 스푸핑보다 응용 범위가 넓다. 타겟을 유인해 파밍, 중간자 공격등 2차 공격을 가할 수 있다. DNS 패킷은 UDP 패킷이므로 세션이 존재하지 않는다. 먼저 도착한 패킷을 신뢰하고 이외의 패킷은 폐기하는 UDP의 특성을 이용한다. 웹 스푸핑과 비슷한 의미로 해석되기도 한다. 하지만 웹 스푸핑은 중간자공격에 사용되는 공격 방식으로 오고가는 모든 트래픽이 공격자를 거쳐가도록 하는 공겨이다. 정상적인 웹 서버에 접근하지 못하도록 하는 DoS 공격이다.

**DNS스푸핑 공격절차:**

1. 허브 환경에서는 브로드캐스팅 하므로 타겟이 DNS Query를 보낼 때 공격자가 자연스럽게 이를 확인할 수 있다. 그러나 스위치 환경에서는 타겟이 DNS 쿼리 패킷을 보내면 공격자가 서버보다 먼저 타겟에게 DNS 패킷을 전달해야 하므로 ARP 스푸핑같은 선행 작업이 필요하다.
2. 공격자는 로컬에 존재해 DNS 서버보다 물리적으로 가깝다. 이를 이용해 서버보다 먼저 DNS 패킷을 전달할 수 있다. (시간차 공격의 특성)
3. 반드시 공격 대상을 기다리고 있어야만 공격을 수행할 수 있는 건 아니다. 네트워크의 특정 URL에 대한 거짓 IP 정보를 계속 브로드캐스팅하면 해당 패킷을 받은 클라이언트는 공격자가 유인한 IP로 접속할 수도 있다.

**방어대책:** 한 번 접속한 사이트는 캐시에 도메인에 대한 IP 저장하므로 DNS Query를 보내지 않고 바로 접속 가능하다. 이 과정에서 hosts 파일을 통해 도메인 이름에 대한 IP주소를 해석하는 절차를 거치게 된다. 따라서 hosts 파일에 중요 사이트의 IP를 미리 등록해두면 이 리스트에 있는 사이트를 접속할 때는 DNS 스푸핑 공격을 예방할 수 있다.

**이메일 스푸핑:** 사회공학 기법, Bagle warm과 같이 악성코드의 메일 유포가 대표적. L7의 콘텍스트 필터링, 안티바이러스, 전자서명, 메일 발송 시 인증 방법이 있다.

**세션 하이재킹:** Blind(원격)와 Non-Blind(로컬) 하이재킹으로 나뉜다. 인증을 마치고 정상적 세션 수립이 완료된 상태에서 공격자가 세션을 가로채는 공격이다. 공격자는 별도 인증 없이 세션 통신을 계속할 수 있다. 일회용 패스워드와 Challenge/Response 기법은 이미 세션이 수립된 후이기 때문에 소용 없다. 원격에서 TCP 연결을 공격하면 TCP 시퀀스 넘버를 알 수 없기 때문에 이를 추측해서 공격해야 한다.(Blind), 로컬에서는 세션을 탐지할 수 있는 상태이기 때문에 Non-Blind이다.

**TCP 연결 하이재킹:** TCP 응용(Telnet, FTP, HTTP 등)은 3way handshaking으로 동작한다. 여기서 하이재킹은 TCP시퀀스 넘버 추론 기반의 하이재킹과 스니핑 기반의 하이재킹으로 구분된다. 스니핑은 기밀성만 침해해서 따로 명령을 내릴 순 없다. TCP 세션 하이재킹은 클라이언트와 같은 인터페이스를 통해 명령과 실행이 가능하다. 이를 위해 HUNT와 Juggernaut라는 툴을 사용한다.

**시퀀스 추론 하이재킹:** 초기 TCP 프로토콜 구현의 경우 임의 시퀀스 넘버(디폴트값)로 세션을 수립하는 경우가 많아 이를 선취해 서버에 응답하고 클라이언트에는 DoS공격을 하는 방식이 많았다. 그 외에 ISN(Initial Sequence Number)를 설정하는 임의의 규칙이 있는데 공격자가 몇 번의 연결 시도로 이 규칙을 알아내고 이를 바탕으로 다음 연결을 위한 시작 번호를 추론해 세션을 가로채는 방식 등이 사용되었다.

**시퀀스 추론 하이재킹의 문제:** 공격자가 ACK으로 응답 후 서버에 데이터를 전송할 수는 있지만 서버가 보내는 데이터를 수신할 수는 없다. 또 정상적 인증 작업은 자동으로 이뤄지지 공격 대상자가 참여하는 형태가 아니기 때문에 서버가 인증 작업 없이 신뢰하는 호스트만이 공격의 대상이 된다.

**대응책:** 제3자가 시작 순서번호 추론을 할 수 없게 만들어야 한다. 순서번호 설정을 난수화한다.

**스니핑 기반 TCP 연결 하이재킹:** 일반적으로는 스니핑을 하는 정도에 그치지만 공격자가 패킷 가공 공격(Packet Crafting Attack)까지 수행할 수 있다면 응용 수준의 완전한 양방향 통신 하이재킹이 될 수 있다. 스니핑 기반의 하이재킹은 TCP 연결의 동기화와 비동기화를 이해하는 것부터 시작한다. 동기화는 TCP가 수신된 데이터의 시퀀스 번호(SEQ)와 상대에게 수신을 확인하는 확인번호(ACK)의 차가 수신 윈도우 크기(수신 버퍼의 빈 공간 크기)보다 작은 상태이다. 서버의 초기 설정 단계에서 연결을 끊고 다른 시퀀스 넘버로 접속하거나 대량의 Null 데이터를 보내 동기화 상태를 비동기화 상태로 만들 수 있다. 비동기화 상태에서 클라이언트와 서버가 데이터 패킷을 송신하면 상호 ACK 패킷 송신을 반복하는 ACK스톰 현상이 발생하게 된다.

**비동기화를 이용한 스니핑 기반 공격 절차:**

1. 클라이언트가 SYN을 보내 세션 수립을 시작하면 공격자가 스니핑으로 세션과 시퀀스 넘버를 확인한다.
2. 공격자는 서버에 RST 패킷을 보내 서버를 Closed하고 클라이언트는 Established로 남겨 ACK를 응답하게 한다. 공격자가 ACK를 가로채 폐기하고 새 시퀀스 넘버로 SYN을 서버에 보내 동기화를 깨트린 세션을 수립한다.
3. 공격자는 정상 연결처럼 서버와 시퀀스 넘버를 교환하고 공격자와 서버 모두 Established 상태가 된다.

**비동기화에 대한 GPT의 답:**

비동기화를 악용한 방법에 대한 내용을 이해하셨다면, TCP 통신에서 시퀀스 번호와 응답 번호의 차이가 윈도우 크기보다 클 때 발생하는 문제는 "TCP Sequence Number Wraparound" 또는 "TCP Sequence Number Wrap" 문제입니다.

TCP 통신에서는 시퀀스 번호(sequence number)를 사용하여 전송되는 데이터의 순서를 지정하고 관리합니다. 시퀀스 번호는 순차적으로 증가하며, 수신 측은 이를 이용하여 데이터의 순서를 파악합니다. 그런데 시퀀스 번호는 32비트 숫자로 표현되기 때문에, 어느 순간 시퀀스 번호가 최대값에 도달하면 다시 0부터 시작하게 됩니다. 이렇게 최대값에서 다시 0으로 돌아가는 현상을 "wraparound"이라고 합니다.

만약 TCP 통신에서 시퀀스 번호와 응답 번호의 차이가 윈도우 크기보다 크다면, 이는 wraparound가 발생했음을 의미합니다. 이러한 상황에서는 데이터의 순서를 올바르게 해석하는데 문제가 발생할 수 있습니다. 수신 측은 wraparound된 시퀀스 번호를 잘못 해석하여 데이터의 순서를 엉뚱하게 파악할 수 있으며, 이로 인해 데이터의 손실이나 혼란이 발생할 수 있습니다.

이러한 문제를 방지하고 해결하기 위해 TCP는 시퀀스 번호와 응답 번호의 관리에 정확한 로직을 적용합니다. 또한 TCP 통신에서 사용되는 윈도우 크기(Window Size)도 중요한 역할을 합니다. 윈도우 크기는 수신 측에서 처리 가능한 데이터 양을 나타내며, 너무 크거나 작은 값은 효율적인 통신을 방해할 수 있습니다.

종합적으로, TCP Sequence Number Wraparound 문제는 TCP 통신의 정확성과 신뢰성을 해치는 요인 중 하나로, 올바른 방어 및 관리가 필요한 보안 취약점입니다.

**방어 대책:** 스니핑 방지와 스푸핑 방지(MAC고정) 대책을 통해 상당 부분 방지 가능하다. 또 TCP 세션 하이재킹은 공격에 몇 가지 특이점이 있어 탐지 후 대응이 가능하다. 특징은 다음과 같다.

1. 비동기화 상태 탐지: 서버와 시퀀스 넘버를 주기적으로 확인해 비동기화에 빠지면 이를 탐지해낸다.

~461p