

수업 내용 정리 보고서

작성일: 2026년 7월 30일

챕터 구성

1. ARP 스푸핑의 원리와 공격 환경 구축
2. Ettercap을 이용한 MITM 공격 및 통신 중계 실습
3. 플러그인 및 Etterfilter를 활용한 트래픽 변조 심화
4. 네트워크 공격 유형 분류 및 보안 운영 점검

ARP 스푸핑의 원리와 공격 환경 구축

네트워크 해킹의 기초 단계 중 하나인 **ARP 스푸핑(ARP Spoofing)**은 동일한 네트워크 인터페이스(NI, Network Interface) 내에 존재하는 호스트들 사이에서 발생하는 취약점을 이용합니다. 공격자가 네트워크 내의 장치들에게 자신의 MAC 주소를 게이트웨이나 다른 피해자의 MAC 주소인 것처럼 속여, 트래픽이 공격자의 PC를 거쳐 가도록 유도하는 기법입니다. 이를 통해 공격자는 중간자 공격(MITM, Man-in-the-Middle)을 수행하며 데이터를 가로채거나 변조할 수 있습니다.

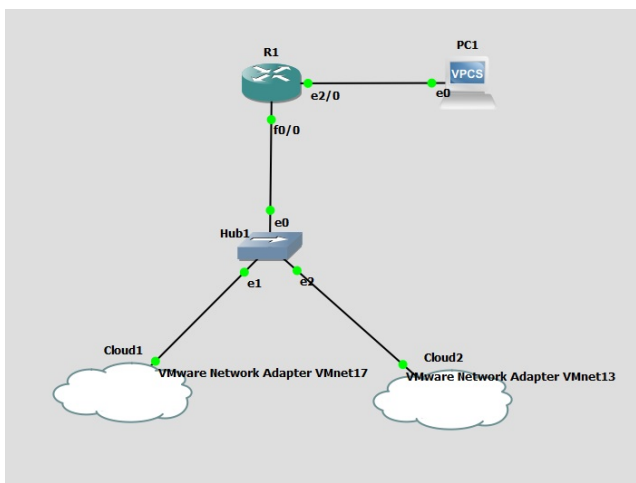
1. 공격 환경 및 네트워크 구성

성공적인 ARP 스푸핑 실습을 위해서는 공격자와 피해자, 그리고 게이트웨이가 동일한 네트워크 대역 내에 구성되어야 합니다. 이번 실습에서는 가상 환경을 활용하여 R1(라우터), PC1(VPCS), 그리고 Hub1을 중심으로 한 네트워크 토폴로지를 구축하였습니다. Hub1은 VMware의 가상 네트워크 인터페이스(VMnet17, VMnet13)를 통해 외부 클라우드 환경과 연결되어 있습니다.

네트워크의 물리적/논리적 연결 관계는 다음과 같습니다.

- **R1 (Router):** e2/0 포트는 PC1과 연결되며, f0/0 포트는 Hub1과 연결됩니다.
- **PC1:** e0 포트를 통해 R1과 통신합니다.
- **Hub1:** e0 포트로 R1과 연결되며, e1/e2 포트를 통해 각각 VMnet17(Cloud1) 및 VMnet13(Cloud2)과 연결됩니다.

R1, PC1, Hub1, Cloud1(VMnet17), Cloud2(VMnet13) 연결 구조



R1, PC1, Hub1 및 VMware 가상 네트워크 인터페이스가 연결된 네트워크 구성도

위 구성도와 같이 장치들이 적절히 연결된 상태에서 공격자는 네트워크 흐름을 파악하고 스푸핑을 시작할 준비를 합니다.

2. 공격자(Kali Linux) 설정: IP 포워딩(IP Forwarding)

ARP 스푸핑을 통해 트래픽을 중간에서 가로챌 때, 공격자가 단순히 패킷을 받기만 하고 전달하지 않으면 피해자와 게이트웨이 사이의 통신이 끊어지게 됩니다. 통신 중단을 방지하고 트래픽을 자연스럽게 중계(Routing)하기 위해서는 공격자 PC(Kali)에서 **IP 포워딩** 기능을 반드시 활성화해야 합니다.

설정 명령어:

```
sysctl -w net.ipv4.ip_forward=1
```

이 명령어를 실행하면 공격자의 커널이 수신한 패킷을 목적지 주소에 따라 다시 전달해주므로, 피해자는 네트워크 장애를 느끼지 못한 채 공격자에게 데이터를 전달하게 됩니다.

3. Ettercap을 이용한 ARP 스푸핑 수행

Ettercap은 강력한 네트워크 해킹 툴로, 사용 환경에 따라 세 가지 모드를 지원합니다.

- **GUI 모드 (-G):** 그래픽 사용자 인터페이스를 통한 직관적인 조작
- **TUI 모드 (-T):** 텍스트 기반 사용자 인터페이스
- **CLI 모드 (-C):** 명령줄 인터페이스를 통한 자동화 및 빠른 실행

실제 공격 시에는 다음과 같은 명령어를 사용하여 양방향 스푸핑(MITM)을 수행합니다.

```
ettercap -T -i [인터페이스] -M arp:[옵션] [게이트웨이_IP]//[피해자_IP]//
```

여기서 **옵션**은 매우 중요합니다. **remote**를 사용하면 게이트웨이와 피해자 양쪽 모두를 속여 양방향 통신을 장악하는 MITM 공격이 가능하며, **oneway**를 사용하면 한쪽 방향으로만 스푸핑을 시도합니다.

주요 공격 플러그인(Plugin) 활용

Ettercap은 단순 스누핑을 넘어 다양한 기능을 수행하는 플러그인을 제공합니다. -P [플러그인_이름] 옵션으로 실행할 수 있습니다.

- **dns_spoof**: 타겟이 특정 도메인에 접속할 때, 공격자가 지정한 가짜 IP로 응답을 변조하여 피싱 사이트 등으로 유도합니다.
- **dos_attack**: ARP 캐시를 지속적으로 채워 네트워크 성능을 저하시키거나 통신을 방해합니다.
- **remote_browser**: 타겟이 방문하는 웹페이지를 공격자의 브라우저에서 실시간으로 렌더링하여 보여줍니다.
- **repoison_arrow / scan_poisoner**: 네트워크 내의 ARP 스누핑 상태를 모니터링하거나, 방어 기제를 우회하기 위해 주기적으로 패킷을 재발송합니다.

4. 실시간 트래픽 조작: Etterfilter (.ef 파일)

단순히 트래픽을 가로채는 것을 넘어, 패킷 내부의 데이터를 실시간으로 변조하려면 **Etterfilter**를 사용해야 합니다. 이는 사용자 정의 필터 파일을 컴파일하여 적용하는 방식입니다.

1. **필터 소스 작성**: script.txt와 같은 파일에 변조 규칙을 작성합니다.
2. **컴파일**: etterfilter script.txt -o script.ef 명령어로 컴파일된 필터 파일을 생성합니다.
3. **실행**: ettercap -T -i eth0 -F script.ef -M arp:remote ...와 같이 필터를 적용하여 실행합니다.

[필터 스크립트 예시 분석]

```
if (ip.proto == TCP && tcp.dst == 80) {
    if (search(DATA.data, "user")) {
        replace("user", "admin");
        tcp_checksum_fix();
    }
}
```

위 스크립트는 HTTP(Port 80) 트래픽 중 데이터 영역에 "user"라는 문자열이 포함되어 있다면 이를 "admin"으로 치환합니다. 이때 가장 중요한 점은 **tcp_checksum_fix()**를 호출하는 것입니다. 패킷의 내용을 변경하면 데이터의 무결성 값이 달라지므로, 체크섬을 재계산해주지 않으면 수신 측에서 변조된 패킷을 오류로 판단하고 폐기하기 때문입니다.

5. 공격 결과 검증 및 확인

공격이 성공적으로 수행되면, 공격자는 중간에서 통신 중계와 동시에 패킷을 모니터링할 수 있습니다. 예를 들어, 피해자(2.1.1.1)와 게이트웨이(2.1.1.2) 사이의 ICMP(Ping) 통신을 캡처했을 때, 두 호스트 간에 정상적으로 *Echo Request*와 *Echo Reply*가 오가는 것을 확인할 수 있습니다.

다음은 실제 통신 로그를 통해 확인한 패킷 흐름의 모습입니다.

ICMP Echo (ping) request/reply 및 IP 2.1.1.1, 2.1.1.2 통신 내역

번호	시간	원래 IP	변조된 IP	원래 프로토콜	변조된 프로토콜	원래 데이터	변조된 데이터
624	234.952380	2.1.1.1	2.1.1.2	ICMP	74 Echo (ping) reply	id=0x0001, seq=7/1792, ttl=255	
625	235.949443	2.1.1.2	2.1.1.1	ICMP	74 Echo (ping) request	id=0x0001, seq=8/2048, ttl=128	
626	235.964296	2.1.1.1	2.1.1.2	ICMP	74 Echo (ping) reply	id=0x0001, seq=8/2048, ttl=255	
627	236.959127	2.1.1.2	2.1.1.1	ICMP	74 Echo (ping) request	id=0x0001, seq=9/2304, ttl=128	
628	236.974382	2.1.1.1	2.1.1.2	ICMP	74 Echo (ping) reply	id=0x0001, seq=9/2304, ttl=255	
629	237.334776	2.1.1.3	8.8.8.8	DNS	81 Standard query 0x41e4 0.0.debian.pool.ntp.org		

2.1.1.1과 2.1.1.2 IP 주소 간에 교환되는 ICMP Echo Request 및 Reply 패킷 로그

로그를 보면 2.1.1.1에서 2.1.1.2로 요청이 가고, 다시 2.1.1.2에서 2.1.1.1로 응답이 돌아오는 과정이 반복됩니다. 이는 공격자가 IP 포위딩을 적절히 설정하여, ARP 스누핑 상태에서도 통신이 끊기지 않고 정상적으로 중계되고 있음을 증명합니다.

학습 정리 및 점검 사항

구분	핵심 체크리스트	주의 사항
사전 준비	공격자와 타겟이 동일 NI 내에 있는가?	서로 다른 서브넷에서는 ARP 스누핑이 동작하지 않음
공격 설정	ip_forward가 1로 설정되었는가?	설정하지 않으면 DoS(서비스 거부) 공격이 됨
Ettercap 실행	remote 옵션을 사용하여 양방향을 잡았는가?	oneway는 단방향 스니핑에만 국한됨
데이터 변조	필터 적용 후 tcp_checksum_fix()를 수행했는가?	체크섬 미수정 시 변조된 패킷은 목적지에서 드랍됨

Ettercap을 이용한 MITM 공격 및 통신 중계 실습

네트워크 보안 실습의 핵심 중 하나인 **MITM(Man-In-The-Middle, 중간자 공격)**은 공격자가 통신 중인 두 대상 사이에 끼어들어 트래픽을 도청하거나 변조하는 공격 방식입니다. 이번 실습에서는 **ARP Spoofing(ARP 스누핑)** 기법을 활용하여 공격자가 네트워크의 중심에서 데이터를 가로채고, 이를 다시 전달하는 과정을 학습합니다.

1. 네트워크 해킹의 전제 조건 및 환경 구성

ARP 스누핑을 통한 네트워크 해킹이 성공하기 위해서는 공격자와 피해자가 동일한 **NI(Network Interface, 네트워크 인터페이스)** 상에 존재해야 합니다. 즉, 동일한 로컬 네트워크 영역(L2 계층) 내에 있어야만 ARP 패킷을 조작하여 대상의 MAC 주소 테이블을 오염시킬 수 있습니다.

실습을 위한 네트워크 구성은 라우터(R1), 허브(Hub1), 그리고 가상 환경(VMware)의 네트워크 어댑터를 통해 구축됩니다. 공격자는 이 구성 요소들 사이의 트래픽이 흐르는 통로에 위치하게 됩니다.

먼저, 실습 환경의 물리적/논리적 연결 구조를 확인합니다. R1 라우터의 e2/0 포트와 PC1의 e0 포트가 연결되어 있으며, Hub1을 중심으로 Cloud1(VMnet17)과 Cloud2(VMnet13)가 연결된 구조를 가집니다.

위 구성도에서 공격자는 피해자(PC1 또는 라우터와 통신하는 다른 호스트)와 게이트웨이 사이의 트래픽을 탈취할 수 있는 위치에 배치됩니다.

2. ARP Spoofing과 IP Forwarding의 중요성

ARP 스푸핑을 통해 공격자가 중간에 끼어들게 되면, 피해자는 공격자를 게이트웨이로 인식하고, 게이트웨이는 공격자를 피해자로 인식하게 됩니다. 이때 공격자가 단순히 패킷을 가로채기만 하고 전달하지 않으면, 피해자의 통신은 즉시 중단(DoS 상태)됩니다.

따라서 성공적인 MITM 공격을 위해서는 공격자(Kali Linux 등)가 받은 패킷을 목적으로 다시 보내주는 **중계(Routing)** 작업이 필수적입니다. 이를 위해 공격자 PC에서는 다음과 같이 IP 포워딩 기능을 활성화해야 합니다.

```
# 공격자 PC에서 IP 포워딩 활성화
sysctl -w net.ipv4.ip_forward=1
```

이 설정이 완료되어야만 공격자가 트래픽을 가로채서 분석(도청)한 뒤, 원래 목적으로 자연스럽게 넘겨주어 피해자가 통신 장애를 느끼지 못하게 할 수 있습니다.

3. Ettercap을 이용한 실전 공격 수행

Ettercap은 ARP 스푸핑을 포함한 다양한 네트워크 공격을 수행할 수 있는 강력한 도구입니다. 실행 모드는 다음과 같이 세 가지로 나뉩니다.

- **-G (GUI)**: 그래픽 사용자 인터페이스 모드
- **-T (TUI)**: 텍스트 기반 사용자 인터페이스 모드
- **-C (CLI)**: 명령줄 인터페이스 모드

실습에서는 TUI 모드를 사용하여 다음과 같은 명령어로 ARP 스푸핑을 수행합니다. 명령의 핵심은 공격 옵션인 **remote**를 사용하는 것입니다.

```
# ettercap -T -i [인터페이스] -M arp:[옵션] [게이트웨이_IP] [피해자_IP]
ettercap -T -i eth0 -M arp:remote /2.1.1.1// /2.1.1.2//
```

여기서 **remote** 옵션은 양방향 스푸핑을 의미하며, 게이트웨이와 피해자 사이의 통신을 모두 공격자 쪽으로 유도하여 완벽한 MITM 환경을 구축합니다. 반대로 **oneway** 옵션을 사용하면 단방향 스푸핑이 이루어집니다.

4. 공격 성공 검증 및 트래픽 확인

공격이 성공적으로 수행되었다면, 공격자는 중간에서 두 대상 간의 패킷을 실시간으로 볼 수 있어야 합니다. 통신이 끊기지 않고 정상적으로 중계되고 있는지 확인하기 위해 **ping** 명령어를 통한 ICMP 트래픽을 모니터링합니다.

터미널의 패킷 캡처 로그를 통해 2.1.1.1과 2.1.1.2 사이에서 ICMP Echo Request(요청)와 Echo Reply(응답)가 정상적으로 오가고 있음을 확인할 수 있습니다.

로그를 분석하면 **id=0x0001, seq=7/1792** 등의 시퀀스 번호와 함께 **ttl** 값이 확인됩니다. 이는 공격자가 트래픽을 성공적으로 중계하고 있으며, 두 호스트 간의 통신이 중단되지 않고 유지되고 있음을 증명하는 결과입니다.

5. 심화 공격: 플러그인 및 필터링을 통한 데이터 변조

단순한 도청을 넘어, Ettercap은 다양한 플러그인과 사용자 정의 필터를 통해 데이터를 실시간으로 조작할 수 있습니다.

주요 공격 플러그인 (-P)

- **dns_spoof**: 타겟이 특정 도메인에 접속할 때, 공격자가 지정한 가짜 IP로 응답하여 피싱 사이트로 유도합니다.
- **dos_attack**: ARP 캐시를 강제로 채워 네트워크 성능을 저하시킵니다.
- **remote_browser**: 타겟이 방문하는 웹페이지를 공격자의 브라우저에서 실시간으로 렌더링하여 보여줍니다.
- **repoison_arrow / scan_poisoner**: 네트워크 내의 ARP 스푸핑 상태를 모니터링하거나 방어 기제를 우회하기 위해 주기적으로 패킷을 재발송합니다.

실시간 트래픽 조작 (Ettercap Filter)

사용자는 **.ef** 확장자를 가진 컴파일된 필터 파일을 사용하여 패킷의 내용을 실시간으로 검사하고 변조할 수 있습니다. 예를 들어, HTTP 패킷 내의 특정 문자열을 바꾸거나 이미지를 교체하는 것이 가능합니다.

필터 적용 절차:

1. 필터 소스 파일 작성 (예: **script.txt**)
2. **etterfilter script.txt -o script.ef** 명령으로 컴파일
3. **ettercap -T -i eth0 -F script.ef -M arp:remote /2.1.1.1// /2.1.1.2//**로 실행

필터 소스 예시 (HTML Injection):

```
if (ip.proto == TCP && tcp.dst == 80) {
    if (search(DATA.data, "user")) {
        replace("user", "admin");
        tcp_checksum_fix(); // 패킷 크기 변경에 따른 체크섬 재계산 필수
    }
}
```

위와 같이 `replace()` 함수를 통해 데이터를 변조한 후에는 반드시 `tcp_checksum_fix()`를 호출하여 TCP 체크섬을 재계산해야 패킷이 깨지지 않고 정상적으로 목적지에 전달됩니다.

6. 실습 요약 및 점검 사항

MITM 공격 실습을 마친 후, 다음 항목들을 통해 학습 내용을 점검하십시오.

점검 항목	확인 내용	실패 시 조치
IP Forwarding	<code>net.ipv4.ip_forward=1</code> 설정 여부	통신 끊김 발생 시 설정 재확인
ARP 옵션	<code>remote</code> (양방향) vs <code>oneway</code> (단방향)	양방향 통신 확인을 위해 <code>remote</code> 사용
인터페이스	공격 도구가 사용하는 NIC(예: <code>eth0</code>) 일치 여부	<code>ifconfig</code> 또는 <code>ip addr</code> 로 확인
필터 컴파일	<code>.ef</code> 파일 생성 및 <code>-F</code> 옵션 사용	<code>etterfilter</code> 명령 수행 여부 확인
체크섬 수정	데이터 변조 시 <code>tcp_checksum_fix()</code> 포함 여부	패킷 유실 발생 시 필터 코드 점검

주의사항: ARP 스누핑은 동일 네트워크 내에서 매우 강력한 공격이므로, 반드시 허가된 격리된 가상 환경(VMware 등)에서만 실행해야 합니다. 또한, `driftnet`과 같은 도구를 병행하면 탈취된 트래픽 내의 이미지를 실시간으로 캡처하여 시각적인 피해 규모를 확인할 수 있습니다.

플러그인 및 Etterfilter를 활용한 트래픽 변조 심화

단순히 네트워크 트래픽을 훔쳐보는 스니핑(Sniffing) 단계를 넘어, 공격자가 중간에서 데이터를 조작하거나 사용자를 가짜 사이트로 유도하는 공격을 수행하기 위해서는 **MITM(Man-in-the-Middle, 중간자 공격)** 환경을 완벽하게 구축하고 트래픽을 제어할 수 있는 능력이 필요합니다. 이번 장에서는 Ettercap의 플러그인 기능과 사용자 정의 필터인 Etterfilter를 활용하여 실시간으로 패킷 내용을 변조하는 심화 기술을 학습합니다.

1. 공격 환경 구축 및 통신 중계의 원리

효과적인 ARP 스누핑 공격을 위해서는 공격자의 PC(Kali Linux)가 단순히 패킷을 가로채는 것에 그치지 않고, 가로챈 패킷을 원래 목적지로 전달해 주는 **라우팅(Routing)** 역할을 수행해야 합니다. 만약 공격자가 패킷을 가로챈 뒤 전달하지 않으면, 피해자와 게이트웨이 사이의 통신이 끊겨 공격 사실이 즉각적으로 발각되기 때문입니다.

이를 위해 공격자는 반드시 다음과 같은 설정을 선행해야 합니다.

- **IP 포워딩 활성화:** Linux 커널에서 패킷을 전달할 수 있도록 설정합니다.

```
sysctl -w net.ipv4.ip_forward=1
```

- **ARP 스누핑 모드 선택:**
 - `remote`: 게이트웨이와 피해자 양방향으로 모두 스누핑하여 완벽한 MITM 환경을 구축합니다.
 - `oneway`: 한 방향으로만 스누핑을 수행합니다.

실습 환경에서는 다음과 같은 가상 네트워크 토폴로지를 통해 공격을 수행할 수 있는 구조를 구성하였습니다.

위 구성도와 같이 Hub1을 중심으로 다양한 네트워크 인터페이스(VMnet13, VMnet17 등)가 연결된 환경에서는 동일한 네트워크 인터페이스(NI) 내에서만 ARP 스누핑이 가능하다는 점을 유의해야 합니다.

2. Ettercap 주요 플러그인(Plugin) 활용

Ettercap은 특정 목적을 가진 공격을 자동화하거나 강화하기 위해 다양한 플러그인을 제공합니다. 명령어를 실행할 때 `-P [플러그인_이름]` 옵션을 사용하여 활성화할 수 있습니다.

플러그인 명칭	주요 기능 및 공격 원리	공격 시나리오
<code>dns_spoof</code>	타겟이 도메인 접속 시 공격자가 지정한 가짜 IP로 응답을 변조	피싱(Phishing) 사이트로 유도하여 계정 정보 탈취
<code>dos_attack</code>	ARP 캐시를 지속적으로 채워 네트워크 마비를 유도	서비스 거부 공격(DoS)
<code>remote_browser</code>	타겟이 방문하는 웹페이지를 공격자의 브라우저에서 실시간 렌더링	피해자의 웹 서핑 내용 실시간 모니터링
<code>repoison_arrow / scan_poisoner</code>	ARP 스누핑 상태를 모니터링하거나 방어 기제 우회를 위해 패킷 재발송	지속적인 스누핑 상태 유지 및 방어 우회

ARP 스누핑이 정상적으로 수행되어 공격자가 중간에서 패킷을 가로채고 중계하고 있다면, 실제 네트워크 상에서는 다음과 같이 타겟(2.1.1.1)과 게이트웨이(2.1.1.2) 사이의 ICMP 트래픽이 정상적으로 오가는 것처럼 보여야 합니다.

위 로그에서 확인할 수 있듯이, Echo (ping) request와 reply가 서로 교차하며 데이터가 흐르고 있다면, 공격자가 ip_forward 설정을 통해 통신을 성공적으로 중계하고 있음을 의미합니다.

3. Etterfilter를 이용한 실시간 트래픽 변조

단순히 트래픽을 보는 것을 넘어, 패킷 내부의 데이터(HTTP 헤더, HTML 내용 등)를 실시간으로 수정하려면 **Etterfilter**를 사용해야 합니다. 이는 사용자가 정의한 규칙에 따라 패킷을 검사하고 변조하는 고급 기술입니다.

[Etterfilter 작업 절차]

1. **필터 소스 작성**: .txt 파일에 변조 규칙을 작성합니다.
2. **컴파일**: etterfilter 명령어를 사용하여 실행 가능한 .ef 파일로 만듭니다.

```
etterfilter script.txt -o script.ef
```

3. **실행**: Ettercap 실행 시 -F 옵션으로 컴파일된 파일을 적용합니다.

```
ettercap -T -i eth0 -F script.ef -M arp:remote /2.1.1.1// /2.1.1.2//
```

[실시간 변조 코드 예시 및 해석]

아래는 HTTP 트래픽을 대상으로 특정 문자열을 치환하는 필터 스크립트의 예시입니다.

```
# 특정 패킷의 문자열을 변조하는 예시

if (ip.proto == TCP && tcp.dst == 80) {
    # 1. HTTP 요청 패킷 내 'user'를 'admin'으로 변조
    if (search(DATA.data, "user")) {
        msg("[-] Found target keyword 'user', replacing with 'admin'...\n");
        replace("user", "admin");
        tcp_checksum_fix(); # 중요: 패킷 크기 변경에 따른 체크섬 재계산
    }
}

if (ip.proto == TCP && tcp.src == 80) {
    # 2. 서버 응답(HTML) 내 페이지 제목 변조
    if (search(DATA.data, "<title>Old Title</title>")) {
        msg("[+] Replacing web page title...\n");
        replace("<title>Old Title</title>", "<title>Hacked by Attacker</title>");
        tcp_checksum_fix();
    }
}
```

⚠ 실무 주의사항: tcp_checksum_fix()의 중요성

패킷 내의 문자열을 replace 함수로 치환하면 데이터의 길이가 변할 수 있습니다. 이때 TCP 체크섬(Checksum)을 재계산하여 수정해주지 않으면, 수신 측(피해자 또는 서버)에서 패킷의 무결성이 깨진 것으로 판단하여 해당 패킷을 즉시 폐기합니다. 따라서 변조 후에는 **반드시 tcp_checksum_fix()**를 호출해야 합니다.

4. 학습 요약 및 점검 리스트

트래픽 변조 실습을 마친 후, 다음 항목들을 통해 공격 및 방어 메커니즘을 이해했는지 점검하십시오.

- **공격 환경 체크**: ip_forward가 설정되지 않았을 때 피해자의 통신이 끊기는 현상을 이해하였는가?
- **스누핑 모드 구분**: remote와 oneway의 차이점을 알고 상황에 맞게 선택할 수 있는가?
- **플러그인 활용**: dns_spoof를 이용한 피싱 공격의 원리를 설명할 수 있는가?
- **필터링 메커니즘**: Etterfilter의 컴파일(Compile) → 실행(Execute) 과정을 숙지하였는가?
- **데이터 무결성**: 패킷 변조 시 tcp_checksum_fix()를 누락했을 때 발생하는 결과(패킷 드랍)를 예측할 수 있는가?

네트워크 공격 유형 분류 및 보안 운영 점검

네트워크 해킹의 기초 단계에서는 동일한 네트워크 인터페이스(NI, Network Interface) 내에서 발생하는 통신 흐름을 가로채는 기술을 이해하는 것이 매우 중요하다. 공격자는 주로 **ARP 스누핑(ARP Spoofing)**을 사용하여 대상 호스트와 게이트웨이 사이의 관계를 조작하며, 이를 통해 중간자 공격(MITM, Man-in-the-Middle) 환경을 구축한다.

1. 네트워크 공격 환경 구성

효과적인 ARP 스푸핑 실습을 위해서는 공격자와 피해자, 그리고 게이트웨이가 논리적으로 연결된 환경이 필요하다. 실습 환경에서는 라우터(R1), PC1(VPCS), 그리고 Hub1을 중심으로 다양한 가상 네트워크 인터페이스(VMnet17, VMnet13)가 구성되어 통신 경로를 형성한다.

공격 환경의 토폴로지를 살펴보면, R1의 인터페이스(e2/0, f0/0)와 Hub1의 포트(e0, e1, e2)가 각각 클라우드 환경 및 가상 머신 인터페이스와 연결되어 네트워크를 구성하고 있음을 확인할 수 있다.

이러한 구성에서 공격자는 동일한 네트워크 세그먼트에 위치하여 허브나 스위치를 통해 흐르는 패킷을 가로챌 준비를 한다.

2. ARP 스푸핑 및 MITM 메커니즘

ARP 스푸핑은 공격자가 자신의 MAC 주소를 게이트웨이나 피해자의 IP 주소와 매핑된 것처럼 속여, 네트워크 상의 트래픽이 공격자의 PC를 거쳐 가도록 만드는 공격이다. 이때 단순히 패킷을 가로채기만 하면 통신이 끊겨 공격 사실이 쉽게 발각될 수 있다.

따라서 공격자(Kali Linux 등)는 반드시 **IP 포워딩(IP Forwarding)** 설정을 활성화해야 한다. 이는 공격자가 받은 패킷을 원래 목적지로 다시 전달해 주는 '중계(Routing)' 역할을 수행하게 하여, 사용자가 통신 장애를 느끼지 못하도록 하기 위함이다.

- **설정 명령어:** `sysctl -w net.ipv4.ip_forward=1`

공격 도구인 **Ettercap**을 사용하여 양방향 스푸핑(Remote Spoofer)을 수행하면, 피해자와 게이트웨이 사이의 모든 트래픽을 완벽하게 통제할 수 있다. 다음은 공격 성공 시 발생하는 통신 흐름의 예시이다.

실제 패킷 캡처 결과(ICMP 트래픽)를 분석해 보면, 2.1.1.1과 2.1.1.2 사이에서 Echo Request와 Reply가 정상적으로 오가고 있는 것처럼 보이지만, 실제로는 모든 패킷이 공격자의 시스템을 거쳐 중계되고 있음을 알 수 있다.

위 로그에서 확인되는 Echo (ping) request와 reply 패킷의 흐름은 스푸핑이 성공적으로 수행되어 네트워크 연결이 유지되고 있음을 증명한다. 만약 IP 포워딩이 설정되지 않았다면, 이러한 왕복 통신은 불가능했을 것이다.

3. Ettercap 주요 기능 및 확장 공격

Ettercap은 단순한 스푸핑을 넘어 다양한 플러그인(Plugin)을 통해 공격의 목적을 달성한다. 공격자는 상황에 따라 다음과 같은 플러그인을 활용할 수 있다.

- **dns_spoof:** 타겟이 특정 도메인(예: google.com)에 접속하려 할 때, 공격자가 미리 지정한 가짜 IP를 응답하여 피싱 사이트로 유도한다.
- **dos_attack:** ARP 캐시를 지속적으로 오염시켜 네트워크를 마비시킨다.
- **remote_browser:** 타겟이 방문하는 웹페이지를 공격자의 브라우저에서 실시간으로 렌더링하여 모니터링한다.
- **repoison_arrow / scan_poisoner:** 네트워크의 방어 기제를 우회하기 위해 주기적으로 ARP 패킷을 재발송하여 스푸핑 상태를 유지한다.

또한, **driftnet**과 같은 도구를 병행하면 네트워크 상의 이미지 트래픽을 탈취하여 피해자의 활동을 시각적으로 감시할 수도 있다.

4. 실시간 트래픽 조작: Etterfilter (.ef)

공격자는 단순한 도청을 넘어, 지나가는 패킷의 내용을 실시간으로 변조(Manipulation)할 수 있다. 이를 위해 **Etterfilter**를 사용하여 사용자 정의 필터 파일을 작성하고 컴파일한다.

[실습 프로세스]

1. 필터 소스 작성 (예: script.txt)
2. 컴파일: `etterfilter script.txt -o script.ef`
3. 실행: `ettercap -T -i [인터페이스] -F script.ef -M arp:remote [게이트웨이] [피해자]`

[필터 스크립트 분석 예시]

```
if (ip.proto == TCP && tcp.dst == 80) {
    # HTTP 요청에서 'user'라는 문자열을 찾아 'admin'으로 변조
    if (search(DATA.data, "user")) {
        replace("user", "admin");
        # 패킷 내용 변경에 따른 체크섬 재계산 필수
        tcp_checksum_fix();
    }
}

if (ip.proto == TCP && tcp.src == 80) {
    # 서버의 응답 메시지 중 HTML 타이틀을 변조 (HTML Injection)
    if (search(DATA.data, "<title>Old Title</title>")) {
        replace("<title>Old Title</title>", "<title>Hacked by Attacker</title>");
        tcp_checksum_fix();
    }
}
```

위 스크립트에서 가장 중요한 점은 `tcp_checksum_fix()` 함수이다. 패킷 내부의 데이터(문자열 등)를 수정하면 패킷의 전체 크기나 내용이 변하므로, 이를 반영하여 TCP 체크섬을 다시 계산해주지 않으면 수신 측에서 변조된 패킷을 오류로 간주하여 폐기하기 때문이다.

5. 사회공학적 공격의 구분: 피싱 vs 파밍

네트워크 공격을 통해 확보한 권한을 바탕으로 사용자를 기만하는 기술은 크게 두 가지로 나뉜다.

구분	핵심 기법	설명
피싱 (Phishing)	유사 도메인 활용	공격자가 만든 가짜 웹사이트로 유도하여 개인정보를 탈취한다.
파밍 (Pharming)	DNS/호스트 변조	사용자가 올바른 주소를 입력해도 DNS 변조 등을 통해 자동으로 가짜 사이트로 접속되게 만든다.

6. 보안 운영 및 점검 체크리스트

네트워크 관리자 및 보안 운영자는 다음과 같은 항목을 정기적으로 점검하여 ARP 스푸핑 및 MITM 공격을 방어해야 한다.

- **ARP 테이블 모니터링:** 동일한 MAC 주소가 서로 다른 IP 주소에 할당되어 있는지, 혹은 짧은 시간 내에 MAC 주소가 빈번하게 변경되는지 확인한다.
- **정적 ARP(Static ARP) 설정:** 매우 중요한 서버나 게이트웨이의 경우, ARP 테이블에 MAC 주소를 정적으로 등록하여 변조를 방지한다.
- **IP Forwarding 상태 점검:** 일반적인 클라이언트 PC에서 ip_forward 설정이 활성화되어 있다면 공격자의 좀비 PC 여부를 의심해야 한다.
- **HTTPS 및 암호화 통신 강제:** 데이터 내용 변조(Etterfilter 공격)를 막기 위해 모든 주요 통신은 SSL/TLS 기반의 암호화 프로토콜을 사용해야 한다.