

칼리 리눅스 툴

- 01 - Reconnaissance:** 정찰 (공격 대상에 대한 사전 정보 수집)
- 02 - Resource Development:** 자원 개발 (공격에 필요한 인프라 구축, 악성코드 제작 등)
- 03 - Initial Access:** 초기 접근 (네트워크나 시스템에 최초로 침투하는 단계)
- 04 - Execution:** 실행 (시스템 내부에서 악성 코드나 공격 명령을 실행)
- 05 - Persistence:** 지속성 유지 (재부팅 후에도 접근 권한을 잃지 않도록 백도어 등을 설치)
- 06 - Privilege Escalation:** 권한 상승 (일반 사용자 권한에서 관리자/루트 권한으로 격상)
- 07 - Defense Evasion:** 방어 회피 (백신, 방화벽 등 보안 솔루션의 탐지를 우회)
- 08 - Credential Access:** 자격 증명 접근 (비밀번호, 인증 토큰 등 계정 정보를 탈취)
- 09 - Discovery:** 탐색 (침투한 시스템 내부의 환경, 네트워크 구조, 자원 등을 파악)
- 10 - Lateral Movement:** 측면 이동 (또는 내부망 이동; 현재 시스템에서 네트워크 내의 다른 시스템으로 이동)
- 11 - Collection:** 수집 (목표로 하는 중요 데이터나 기밀 정보를 모음)
- 12 - Command and Control:** 명령 및 제어 (C2; 외부의 공격자 서버와 통신하며 시스템을 원격 제어)
- 13 - Exfiltration:** 유출 (수집한 데이터를 외부로 빼돌림)
- 14 - Impact:** 영향 (데이터 파괴, 랜섬웨어 암호화, 시스템 마비 등 최종적인 피해 유발)
- 15 - Forensics:** 포렌식 (디지털 증거 수집 및 침해 사고 분석 도구)
- 16 - Services and Other Tools:** 서비스 및 기타 도구 (DB 서비스, 시스템 데몬 및 기타 유틸리티)

칼리 리눅스 핵심 보안 도구 명세서 (Kali Linux Tools Summary)

1. 정보 수집 및 정찰 (Information Gathering & Reconnaissance)

공격 대상을 파악하고 네트워크 구조, 서브도메인, 이메일, 열린 포트 등 최대한 많은 배경 정보를 수집하는 단계에 사용되는 도구입니다.

- **Nmap / Zenmap**: 세계에서 가장 유명한 네트워크 스캐너로, 호스트 탐지 및 포트 스캐닝을 수행합니다. (Zenmap은 Nmap의 GUI 버전)
- **Recon-ng**: 웹 기반의 강력한 오픈소스 정보 수집(OSINT) 프레임워크입니다.
- **theHarvester**: 검색 엔진을 활용하여 이메일 계정, 서브도메인, 가상 호스트 등을 수집합니다.
- **Amass / Subfinder / Assetfinder**: 대상 도메인의 서브도메인 및 인프라 자산을 광범위하게 탐색합니다.
- **Netdiscover / ARP-scan**: 로컬 네트워크에서 활성화된 호스트의 IP 및 MAC 주소를 탐색합니다.
- **BloodHound**: Active Directory(AD) 환경에서 권한 관계를 시각화하여 공격 경로를 찾아냅니다.

2. 취약점 분석 및 웹 해킹 (Vulnerability Analysis & Web Applications)

웹 사이트, 데이터베이스, 서버 시스템에 존재하는 보안 약점(취약점)을 스캔하고 분석합니다.

- **SQLmap**: 데이터베이스에 대한 SQL 인젝션(SQLi) 취약점을 자동으로 탐지하고 공격하는 강력한 툴입니다.
- **Burp Suite**: 웹 애플리케이션 보안 테스트를 위한 통합 인터셉트 프록시 플랫폼입니다.
- **Nikto**: 웹 서버를 스캔하여 오래된 파일, 잘못된 설정 등 수천 가지 잠재적 취약점을 찾아냅니다.
- **WPScan**: 워드프레스(WordPress)로 제작된 웹사이트의 플러그인, 테마 취약점을 진단합니다.
- **Gobuster / Dirb / Dirsearch / ffuf**: 웹 서버의 숨겨진 디렉토리나 파일을 찾아내는 브루트포싱 도구입니다.

3. 익스플로잇 및 공격 프레임워크 (Exploitation Tools)

발견된 취약점을 실제로 공격하여 시스템의 제어권을 획득하는 프레임워크들입니다.

- **Metasploit-Framework (MSF):** 전 세계 보안 전문가들이 사용하는 방대한 익스플로잇 코드 라이브러리와 페이로드 생성 기능을 갖춘 통합 프레임워크입니다.
- **BeEF (Browser Exploitation Framework):** 웹 브라우저의 취약점(주로 XSS)을 타겟으로 하여 클라이언트 측을 공격하고 제어합니다.
- **Armitage:** Metasploit을 시각적이고 직관적으로 사용할 수 있게 해주는 그래픽(GUI) 기반 사이버 공격 관리 도구입니다.

4. 패스워드 크래킹 (Password Attacks)

탈취한 비밀번호 해시를 해독하거나, 원격 서비스에 무차별 대입 공격(Brute-force)을 수행합니다.

- **Hashcat:** GPU의 연산 능력을 활용하여 해시된 암호를 초고속으로 복구(크래킹)하는 최고 수준의 도구입니다.
- **John the Ripper (John):** 다양한 포맷의 패스워드 해시를 지원하는 오프라인 크래커입니다.
- **Hydra / Medusa:** SSH, FTP, HTTP 등 다양한 온라인 네트워크 로그인 프로토콜을 대상으로 빠른 브루트포스 공격을 수행합니다.
- **Crunch:** 사용자 정의 규칙에 따라 대규모 패스워드 사전(Wordlist)을 생성합니다.

5. 스니핑 및 스푸핑 (Sniffing & Spoofing)

네트워크 트래픽을 도청(Sniffing)하거나, 자신의 신원을 위장(Spoofing)하여 데이터를 가로칩니다.

- **Wireshark / Tshark:** 네트워크 상에서 캡처된 패킷을 깊이 있게 분석할 수 있는 세계 표준 패킷 분석기입니다.
- **Responder:** 내부 네트워크에서 NBT-NS, LLMNR 등의 요청을 가로채어 Windows 자격 증명(해시)을 수집하는 스푸핑 도구입니다.
- **Ettercap / Bettercap:** 중간자 공격(MitM)을 수행하여 네트워크 트래픽을 실시간으로 조작하고 가로칩니다.
- **Macchanger:** 네트워크 인터페이스의 MAC 주소를 임의로 변경하여 추적을 회피합니다.

6. 무선 네트워크 공격 (Wireless Attacks)

Wi-Fi, Bluetooth 등 무선 네트워크 환경의 보안 약점을 점검합니다.

- **Aircrack-ng:** 802.11 무선 네트워크의 WEP, WPA, WPA2 암호를 크래킹하고 패킷을 분석하는 툴킷입니다.
- **Wifite:** Aircrack-ng, Reaver 등의 도구를 자동화하여 손쉽게 무선 네트워크를 해킹할 수 있게 돕는 스크립트입니다.
- **Kismet:** 무선 네트워크 트래픽을 수동적으로 수집(Passive Sniffing)하는 침입 탐지 및 탐색 도구입니다.
- **Reaver:** WPS(Wi-Fi Protected Setup) 핀 번호의 취약점을 이용한 브루트포스 공격 도구입니다.

7. 포스트 익스플로잇 및 측면 이동 (Post Exploitation & Lateral Movement)

시스템 침투 성공 후 권한을 상승시키고, 내부망의 다른 시스템으로 감염을 확산시키는 도구입니다.

- **Mimikatz**: Windows 메모리(LSASS)에 접근하여 평문 비밀번호, 해시, Kerberos 티켓 등을 추출하는 강력한 도구입니다.
- **Impacket**: SMB, WMI, MSRPC 등 다양한 네트워크 프로토콜을 파이썬으로 구현한 스크립트 모음으로, NTLM 릴레이 및 횡적 이동(Lateral Movement)에 필수적입니다.
- **CrackMapExec / NetExec**: Active Directory 환경 내에서 대규모 네트워크의 자격 증명을 검증하고 횡적 이동을 자동화합니다.
- **Evil-WinRM**: Windows의 원격 관리 프로토콜(WinRM)을 활용하여 해킹용 원격 셸을 제공합니다.

8. 리버스 엔지니어링 및 디지털 포렌식 (Reverse Engineering & Forensics)

악성코드를 분석하거나 사고 발생 후 침해 흔적(디지털 증거)을 수집하고 복원합니다.

- **Autopsy**: 직관적인 인터페이스를 제공하는 디지털 포렌식 플랫폼으로 디스크 이미지를 분석합니다.
- **Binwalk**: 펌웨어 이미지나 컴파일된 바이너리 파일에 내장된 숨겨진 파일과 코드를 추출합니다.
- **Ghidra**: 미국 NSA에서 개발하여 오픈소스로 공개한 고급 소프트웨어 리버스 엔지니어링(SRE) 프레임워크입니다.
- **Bulk-extractor**: 디스크 이미지에서 손상되거나 삭제된 이메일 주소, 신용카드 번호 등을 빠르게 추출합니다.

분류 (Category)	목적 및 설명	주요 도구 (Key Tools)
정보 수집 및 정찰 (Reconnaissance)	공격 대상 파악, 네트워크 구조, 서브도메인, 열린 포트 등 사전 정보 수집	Nmap (네트워크 스캔) Recon-ng (OSINT 프레임워크) theHarvester (이메일/도메인 수집) BloodHound (AD 권한 시각화)
취약점 분석 및 웹 해킹 (Vulnerability & Web)	웹 사이트, DB, 서버에 존재하는 보안 약점 스캔 및 분석	SQLmap (SQL 인젝션 자동화) Burp Suite (웹 프록시 테스트) Nikto (웹 서버 취약점 스캔) WPScan (워드프레스 진단)
익스플로잇 및 공격 (Exploitation)	발견된 취약점을 실제로 공격하여 시스템의 제어권 획득	Metasploit-Framework (통합 공격) BeEF (브라우저 취약점 공격) Armitage (GUI 기반 공격 관리)

패스워드 크래킹 (Password Attacks)	탈취한 암호 해시 해독 및 원격 서비스 무차별 대입(Brute-force) 공격	Hashcat (초고속 GPU 해시 크래킹) John the Ripper (오프라인 크래커) Hydra (온라인 로그인 브루트포스) Crunch (패스워드 사전 생성)
스니핑 및 스푸핑 (Sniffing & Spoofing)	네트워크 트래픽 도청 및 신원 위장을 통한 중간자 공격(MitM)	Wireshark (네트워크 패킷 분석) Responder (내부망 자격 증명 탈취) Ettercap (네트워크 트래픽 조작) Macchanger (MAC 주소 위조)
무선 네트워크 공격 (Wireless Attacks)	Wi-Fi 등 무선 네트워크 환경의 패킷 분석 및 암호 크래킹	Aircrack-ng (Wi-Fi 패킷 분석/크래킹) Wifite (무선 해킹 자동화 스크립트) Kismet (무선 트래픽 수집/탐지) Reaver (WPS 핀 브루트포스)

포스트 익스플로잇 (Post Exploitation)	침투 성공 후 권한 상승, 자격 증명 추출 및 내부망 횡적 이동	Mimikatz (Windows 메모리 정보 추출) Impacket (네트워크 프로토콜 스크립트) CrackMapExec (AD 환경 횡적 이동) Evil-WinRM (원격 셸 제어)
리버스 엔지니어링 & 포렌식 (Reverse Eng. & Forensics)	악성코드 분석 및 사고 발생 후 침해 흔적(디지털 증거) 수집/복원	Autopsy (디스크 이미지 포렌식) Binwalk (펌웨어/바이너리 파일 추출) Ghidra (소프트웨어 역공학 프레임워크) Bulk-extractor (민감 데이터 추출)

주 정 통

Web Application(웹)

코드 인젝션

SQL 인젝션

디렉터리 인덱싱

에러 페이지 적용 미흡

정보 누출

크로스사이트 스크립팅

크로스사이트 요청 위조(CSRF)

SSRF(Server Side Request Forgery)

약한 비밀번호 정책

불충분한 인증 절차

불충분한 권한 검증

취약한 비밀번호 복구 절차

프로세스 검증 누락

악성 파일 업로드

파일 다운로드

불충분한 세션 관리

데이터 평문 전송

쿠키 변조

관리자 페이지 노출

자동화 공격

불필요한 Method 악용

진단 항목	관련 해킹 기술 명세 (Attack Techniques)
코드 인젝션	OS Command Injection (운영체제 명령어 삽입) RCE (Remote Code Execution, 원격 코드 실행) SSTI (Server-Side Template Injection)
SQL 인젝션	SQLi (SQL Injection) - 세부 기법: Error-based SQLi, Blind SQLi, Union-based SQLi
디렉터리 인텍싱	Forced Browsing (강제 브라우징 - 경로 직접 입력)
에러 페이지 적용 미흡	Error-based Information Disclosure (에러 기반 정보 노출)
정보 누출	Directory Brute-forcing (디렉터리 무차별 대입) Google Dorking (구글 해킹을 통한 정보 수집)
크로스사이트 스크립팅	XSS (Cross-Site Scripting) - 세부 기법: Stored XSS, Reflected XSS, DOM-based XSS
크로스사이트 요청 위조	CSRF (Cross-Site Request Forgery)

SSRF	SSRF (Server-Side Request Forgery)
약한 비밀번호 정책	Brute-force Attack (무차별 대입 공격) Dictionary Attack (사전 대입 공격)
불충분한 인증 절차	Authentication Bypass (인증 우회) Parameter Tampering (파라미터 변조)
불충분한 권한 검증	IDOR (Insecure Direct Object Reference, 안전하지 않은 직접 객체 참조) Privilege Escalation (수직적/수평적 권한 상승)
취약한 비밀번호 복구 절차	Token Hijacking (인증 토큰 탈취) OTP Bypass (인증 번호 우회)
프로세스 검증 누락	Business Logic Exploitation (비즈니스 로직 악용) Race Condition (경쟁 조건 공격)
악성 파일 업로드	Web Shell Upload (웹셸 업로드) File Extension Bypass (확장자 우회) Null Byte Injection (널 바이트 삽입)

파일 다운로드	Path Traversal / Directory Traversal (경로 탐색 공격) LFI (Local File Inclusion, 로컬 파일 포함)
불충분한 세션 관리	Session Hijacking (세션 하이재킹 - 세션 탈취) Session Fixation (세션 고정 공격)
데이터 평문 전송	Packet Sniffing (패킷 스니핑 - 네트워크 도청) MITM (Man-in-the-Middle, 중간자 공격)
쿠키 변조	Cookie Spoofing (쿠키 스푸핑 - 타인 위장) Cookie Tampering (쿠키 변조)
관리자 페이지 노출	Forced Browsing (강제 브라우징) Admin Page Brute-forcing (관리자 경로 무차별 대입)
자동화 공격	Credential Stuffing (크리덴셜 스텀핑 - 유출된 계정 대입) Application Layer DoS (L7 서비스 거부 공격)
불필요한 Method 악용	XST (Cross-Site Tracing - TRACE 메서드 악용) WebDAV Exploitation (PUT, DELETE 메서드 악용)

해킹 기술 명세

1. 랜드 어택 (Land Attack)

- 개념: 출발지(Source) IP 주소와 목적지(Destination) IP 주소를 공격 대상 서버의 IP로 똑같이 위조하여 패킷을 보내는 공격입니다.
- 원리: 타겟 서버가 패킷을 받고 응답(SYN-ACK)을 보내려고 목적지 주소를 확인해보니 자기 자신입니다. 결국 자기 자신에게 무한히 응답을 보내는 루프(Loop)에 빠지게 됩니다.
- 결과: 시스템의 자원(CPU, 메모리)이 급격히 고갈되어 정상적인 서비스를 할 수 없게 됩니다. (일종의 DoS 공격 기법입니다.)

2. 스니핑 (Sniffing)

- 개념: 네트워크 상에서 흘러다니는 데이터 패킷을 '엿듣는(도청)' 행위입니다.
- 원리: 전화선에 몰래 도청 장치를 다는 것과 같습니다. 데이터가 암호화되지 않은 평문(HTTP, FTP 등)으로 전송될 경우, 공격자가 패킷 분석 도구를 통해 그 안의 내용을 그대로 읽어냅니다.
- 결과: 사용자의 아이디, 비밀번호, 이메일 내용 등 민감한 정보가 유출됩니다. (데이터를 조작하지는 않으므로 수동적 공격에 속합니다.)

3. 스푸핑 (Spoofing)

- 개념: 영어 단어 'Spoof(속이다)'에서 유래한 것으로, 자신의 신원을 신뢰할 수 있는 다른 사람이나 시스템인 것처럼 '위장'하는 공격입니다.
- 원리 및 종류:
 - IP 스푸핑: 자신의 IP 주소를 신뢰받는 IP로 위장하여 보안 시스템(방화벽 등)을 우회합니다.
 - ARP 스푸핑: MAC 주소를 속여, 통신 패킷이 원래 가야 할 라우터가 아닌 공격자에게 오도록 흐름을 바꿉니다.
 - DNS 스푸핑: 도메인 주소를 속여, 사용자가 정상적인 네이버나 구글을 입력해도 공격자가 만든 가짜(피싱) 사이트로 접속하게 만듭니다.
- 결과: 시스템에 불법적으로 접근하거나, 통신 내용을 가로채기 위한(스니핑을 위한) 사전 작업으로 널리 쓰이는 능동적 공격입니다.

4. 숄더 서핑 (Shoulder Surfing / 오버 더 숄더)

- 개념: 컴퓨터 기술을 사용하는 것이 아니라, 사람의 심리나 행동의 취약점을 노리는 사회공학적 기법(Social Engineering)입니다. 문자 그대로 피해자의 어깨 너머로 화면이나 키보드를 몰래 훑쳐보는 행위입니다.
- 예시: 카페나 도서관에서 다른 사람의 노트북 화면(사내 기밀문서 등)을 몰래 보거나, ATM 기기나 도어락에서 비밀번호를 누를 때 뒤에서 손가락 위치를 훑쳐보는 것을 말합니다.
- 대응: 노트북에 시선 차단 필름(보안 필름)을 붙이거나, 비밀번호 입력 시 손으로 가리는 습관이 필요합니다.

5. DoS (Denial of Service, 서비스 거부 공격)

- 개념: 특정 서버나 네트워크에 정상적인 처리 용량을 초과하는 대량의 악성 데이터(트래픽)를 보내어 시스템을 마비시키는 공격입니다.
- 원리: 단일 공격자(1대의 PC)가 타겟 시스템의 자원(네트워크 대역폭, CPU 등)을 고갈시킵니다.
- 결과: 서버가 다운되거나 너무 느려져서, 실제 정상적인 사용자들이 해당 웹사이트나 서비스에 접속할 수 없게(거부당하게) 됩니다.

6. DDoS (Distributed Denial of Service, 분산 서비스 거부 공격)

- 개념: DoS 공격의 진화형으로, 수많은 PC가 동시에 한 곳을 공격하는 방식입니다. (Distributed = 분산된)
- 원리: 공격자가 사전에 악성코드에 감염시켜 놓은 수천~수십만 대의 좀비 PC(봇넷)를 원격으로 조종하여, 일제히 타겟 서버에 트래픽 폭탄을 쏟아붓습니다.
- 결과: 1대의 PC가 공격하는 DoS와 달리 공격 근원지(IP)가 전 세계에 분산되어 있어 차단하기가 매우 어렵고, 피해 규모도 압도적으로 큼니다.

1. 네트워크를 마비시키는 공격 (DoS/DDoS의 세부 기법)

서버를 다운시키기 위해 트래픽을 보내는 방식도 아주 다양합니다.

- SYN 플러딩 (SYN Flooding): TCP 통신의 약점을 노린 공격입니다. 연결을 요청(SYN)만 하고 마무리(ACK)를 하지 않아, 서버가 응답을 기다리며 메모리를 낭비하다가 결국 뻘어버리게 만듭니다.
- 스머프 공격 (Smurf Attack): 출발지 IP를 피해자의 IP로 위조(스푸핑)한 뒤, 네트워크 상의 모든 PC에게 브로드캐스트(전체 방송)로 핑(Ping)을 보냅니다. 그럼 모든 PC가 일제히 피해자에게 응답을 보내어 네트워크가 마비됩니다.
- 죽음의 핑 (Ping of Death): 네트워크에서 허용하는 최대 패킷 크기보다 훨씬 큰 핑(Ping)을 잘게 쪼개서 보냅니다. 피해자 서버가 이를 다시 조립하는 과정에서 과부하가 걸려 다운됩니다.
- 티어드롭 (Teardrop Attack): 데이터를 쪼개서(단편화) 보낼 때, 조립 순서(오프셋)를 고의로 겹치거나 어긋나게 조작하여 보냅니다. 수신 측 시스템이 퍼즐을 맞추지 못하고 오류를 일으키며 멈춰버립니다.

2. 사람의 심리를 낚는 '사회공학적 공격'

시스템의 취약점이 아닌, 사람의 실수나 무지를 노리는 가장 성공률 높은 기법들입니다.

- 피싱 (**Phishing**): 금융기관이나 공공기관을 사칭한 가짜 이메일/웹사이트로 유도하여 개인정보를 빼냅니다.
 - 스피어 피싱 (**Spear Phishing**): 불특정 다수가 아닌, 특정 타겟(개인/기업)을 꼭 집어서 맞춤형으로 보내는 정교한 피싱입니다.
- 파밍 (**Pharming**): 피싱보다 한 단계 진화한 수법입니다. 사용자가 브라우저에 정확한 주소(예: www.naver.com)를 입력해도, 해커가 DNS(도메인 주소록)를 조작해두어 가짜 사이트로 강제 접속되게 만듭니다.
- 워터링 홀 (**Watering Hole**): 사자가 물웅덩이(Watering Hole)에 숨어서 먹잇감을 기다리듯, 타겟 조직의 직원들이 자주 방문하는 웹사이트를 미리 해킹해 두고 접속할 때 악성코드에 감염되게 만드는 함정입니다.

3. 시스템 메모리와 운영체제를 노리는 '시스템 해킹'

프로그램이 실행되는 가장 낮은 단계를 노리는 치명적인 공격입니다.

- 버퍼 오버플로우 (**Buffer Overflow**): 해킹의 꽃이라고도 불립니다. 프로그램이 할당받은 메모리 공간(버퍼)보다 훨씬 더 많은 양의 데이터를 밀어 넣어서, 넘쳐흐른 데이터가 정상적인 코드를 덮어쓰고 해커가 원하는 악성 명령을 실행하게 만드는 기법입니다.
- 레이스 컨디션 (**Race Condition / 경쟁 조건**): 두 개의 프로세스가 하나의 자원을 놓고 서로 먼저 쓰려고 경쟁하는 타이밍의 허점을 노립니다. 시스템이 파일을 검사하는 시점과 실제로 사용하는 시점의 아주 짧은 찰나에, 정상 파일을 악성 파일로 바꿔치기하는 수법입니다.
- 루트킷 (**Rootkit**): 시스템의 최고 권한(Root/Admin)을 획득한 후, 해커가 자신의 존재를 숨기고 백도어(뒷문)를 유지하기 위해 설치하는 은폐용 악성코드 패키지입니다. 백신으로도 탐지가 매우 어렵습니다.

4. 트렌디하고 고도화된 현대 공격

뉴스에 자주 오르내리는 강력한 해킹 위협들입니다.

- APT (**Advanced Persistent Threat / 지능형 지속 위협**): 특정 조직이나 국가를 타겟으로 삼고, 수단과 방법을 가리지 않으며 장기간에 걸쳐 끈질기게 은밀하게 침투를 시도하는 대규모 공격 캠페인입니다.
- 제로데이 공격 (**Zero-Day Attack**): 소프트웨어의 취약점이 발견되었으나, 개발사가 이를 수정하는 보안 패치(업데이트)를 발표하기도 전(0일 차)에 그 틈을 타서 번개처럼 공격하는 가장 막기 힘든 기법입니다.
- 랜섬웨어 (**Ransomware**): 사용자의 시스템이나 중요 파일들을 암호화하여 열어볼 수 없게 만든 뒤, 이를 풀어주는 대가로 돈(Ransom, 암호화폐)을 요구하는 악질적인 기법입니다.

어떤 분야의 해킹(웹, 네트워크, 시스템 등)을 주로 공부하고 계신가요? 특정 분야를 알려주시면 그쪽 위주로 더 깊게 파고들어 설명해 드릴 수 있습니다.

1. 템플릿 및 구조적 취약점 (Advanced Server-Side)

- **SSTI (Server-Side Template Injection, 서버 사이드 템플릿 인젝션)**
 - 개념: 웹 프레임워크가 화면을 그릴 때 사용하는 '템플릿 엔진(예: Python의 Jinja2, Java의 Thymeleaf, Node.js의 Pug 등)'에 악성 코드를 주입하는 기법입니다.
 - 시나리오 활용: 공격자가 프로필 이름이나 게시물 제목에 `{{7*7}}` 같은 템플릿 문법을 넣었을 때 화면에 49로 출력된다면 SSTI 취약점이 있는 것입니다. 이를 악용해 서버의 시스템 명령어를 직접 실행(RCE)하여 최고 권한을 탈취하는 강력한 시나리오를 짤 수 있습니다.
- **XXE (XML External Entity Injection, XML 외부 엔티티 삽입)**
 - 개념: 웹 서버가 사용자가 업로드한 XML 데이터를 파싱(해석)할 때, 외부 자원을 참조하도록 하는 `<!ENTITY>` 구문을 악용하는 공격입니다.
 - 시나리오 활용: 주로 파일 업로드(SVG 이미지, DOCX 문서 등도 내부적으로 XML 구조임)나 API 통신에서 발생합니다. 이를 통해 서버의 로컬 파일(/etc/passwd 등)을 읽어내거나, 내부망의 다른 서버를 공격(SSRF로 연계)하는 데 사용됩니다.
- **Insecure Deserialization (안전하지 않은 역직렬화)**
 - 개념: 객체(Object)를 전송 가능한 형태(문자열, 바이트 등)로 바꾸는 것을 '직렬화', 반대로 되돌리는 것을 '역직렬화'라고 합니다. 공격자가 조작된 직렬화 데이터를 서버로 보내, 서버가 이를 역직렬화하는 과정에서 악성 코드가 실행되게 만듭니다.
 - 시나리오 활용: 자바(Java), 파이썬(Pickle), PHP 환경에서 주로 발생하며, 쿠키 값이나 API 페이로드를 변조하여 서버를 완전히 장악(RCE)하는 치명적인 엔딩으로 자주 쓰입니다.

2. 최신 인증 및 API 취약점 (Modern Auth & API)

- **JWT (JSON Web Token) 변조 공격**
 - 개념: 현대 웹에서 로그인 상태를 유지하기 위해 가장 많이 쓰는 JWT 토큰의 구조적 취약점이나 검증 로직의 허점을 노립니다.
 - 시나리오 활용: 토큰 헤더의 암호화 알고리즘(alg)을 none으로 변조하여 서명 검증을 우회하거나, 서버가 사용하는 비밀키(Secret Key)가 너무 단순할 경우 브루트포싱으로 키를 알아내어 공격자가 직접 '관리자(Admin) 토큰'을 스스로 위조해 냅니다.
- **CORS (Cross-Origin Resource Sharing) 설정 오류 악용**
 - 개념: 웹 브라우저는 보안상 다른 도메인 간의 데이터 요청을 막지만(SOP 정책), CORS 설정을 통해 예외를 둡니다. 이 예외 설정이 너무 관대하게(* 등) 되어 있을 때 발생하는 취약점입니다.
 - 시나리오 활용: 공격자가 만든 악성 사이트에 피해자가 접속하면, 피해자의 브라우저가 몰래 타겟 웹사이트로 요청을 보내 피해자의 민감한 개인정보나 API 응답 데이터를 공격자에게 빼돌립니다.

3. 구조 및 네트워크 꼼수 (Architecture Flaws)

- **HTTP Request Smuggling (HTTP 요청 스머글링)**

- 개념: 사용자와 백엔드 서버 사이에 있는 프론트엔드 서버(로드밸런서, 리버스 프록시 등)가 HTTP 요청 패킷의 끝(길이)을 서로 다르게 해석하는 차이를 악용합니다.
- 시나리오 활용: 하나의 HTTP 요청 안에 또 다른 '숨겨진 악성 요청'을 밀어 넣습니다. 정상 사용자의 요청에 악성 요청이 섞여 들어가게 만들어, 다른 사람의 세션을 가로채거나 캐시를 오염시킵니다. (난이도가 높지만 매우 세련된 시나리오에 적합합니다.)
- **WebSockets 취약점 (CSWSH)**
 - 개념: 실시간 채팅이나 주식 티커 등에 쓰이는 웹소켓 통신 과정에서 인증 토큰이나 Origin 검증이 누락되는 취약점입니다.
 - 시나리오 활용: 피해자가 공격자의 사이트에 접속했을 때, 공격자가 피해자의 권한으로 대상 웹소켓 서버에 연결을 맺고 실시간으로 민감한 데이터를 도청하거나 악성 메시지를 브로드캐스트합니다.