# Nmap

**Nmap(Network Mapper)**는 네트워크 탐색과 보안 진단을 위해 사용하는 오픈소스 도구로 **Zenmap**이라는 GUI 버전도 지원한다. 외부로 오픈 된 포트를 점검하고, 서비스 버전과 취약점 정보 등 **shodan, censys**와 같이 침투 테스트 전 정보수집을 위한 용도로 사용한다.

|  |  |
| --- | --- |
| **Scan Type** | **설명** |
| **-sS/sT/sA** | TCP SYN / Connect(Open) / ACK Scan |
| **-sU** | UDP Scan |
| **-sN/sF/sX** | TCP NULL / FIN / Xmas Scan |
| **-sP** | Ping Scan |
| **-sL** | 핑 목록 스캔 |
| **-PS** | TCP SYN ping |
| **-PA** | TCP ACK ping |
| **-D** | Decoy Scan: 실제 스캐너 주소 외에 다른 주소로 위조하여 스캔 |
| **-b** | TCP FTP Bounce Scan |

표 2‑1 스캔 타입

|  |  |
| --- | --- |
| **Option** | **설명** |
| **-p** | 포트 범위 지정 |
| **-d** | Debugging |
| **-oN/oX <file>** | 결과를 일반/XML 파일 형식으로 출력 |
| **-F** | 빠른 네트워크 스캐닝 |
| **-T0 ~ -T5** | 공격속도 설정, 스캐닝 탐지 우회 목적으로 사용하며 T0 > T5 순으로 빠르다. |
| **-sV** | 서비스 버전 확인 |
| **–source-port [N]** | 출발지 포트 지정 |
| **-O** | 대상 호스트의 운영체제 정보 출력 |

표 2‑2 옵션

Nmap 실습 환경은 표 2-3과 같다.

|  |  |  |
| --- | --- | --- |
|  | **PC 환경** | **IP** |
| **공격 PC** | Kali Linux 2019.01 | 192.168.86.144 |
| **스캔 대상 PC** | Metasploitable2-linux | 192.168.86.135 |

표 2‑3 실습 환경

## **핑 스캐닝**

네트워크 조사 첫 번째 단계 중 하나는 대상 IP 범위에서 살아있는 호스트 목록을 만드는 것이다.

침투 테스터는 IP 주소를 가진 모든 시스템에 관심을 가져야 한다. 핑 스캐닝은 전달받은 IP 주소가 실제로 활성화돼 있는지 확인하기 위해 필요하다.

### 목록 스캔(-sL)

목록 스캔은 어떤 패킷도 대상 호스트에 보내지 않고 지정한 침투 네트워크 IP범위 검증에 사용한다. 스캔 시 제공받지 않은 호스트가 출력된다면 잘못된 주소일 수 있다. 포트 스캐닝, OS 탐지, 핑 스캐닝 기능은 같이 사용할 수 없다. 그림 2‑1은 구글 웹서버 CIDR/28 네트워크 범위를 스캔한 결과로서 172.217.25.4 IP를 [www.google.com](http://www.google.com) 도메인이 사용한다.

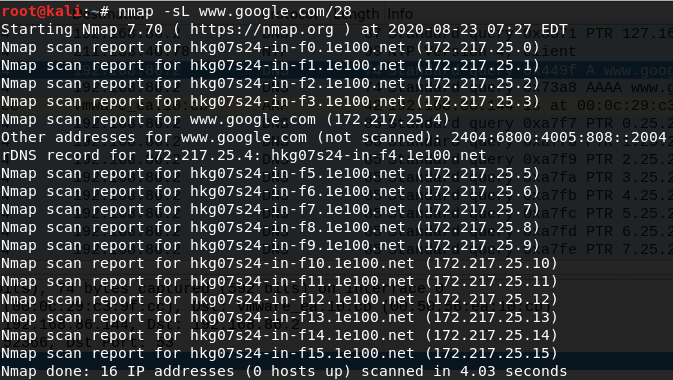


그림 2‑1 목록 스캔

그림 2‑2와 같이 목록 스캔은 호스트에 패킷을 보내는 것이 아니라 DNS에 호스트를 질의한다.

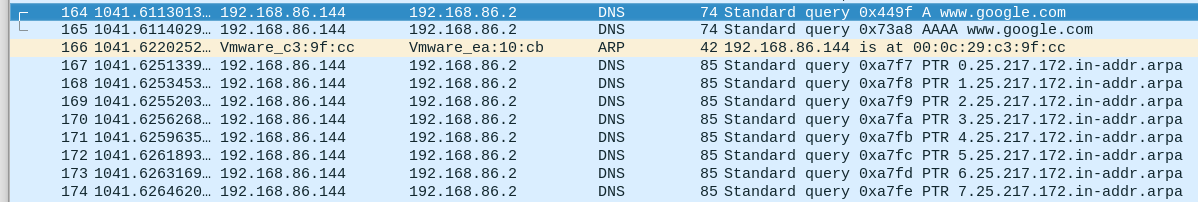


그림 2‑2 목록 스캔 패킷 분석

### Ping Scan(-sP)

Nmap이 핑 스캔만을 수행한 후 스캔에 응답해 이용 가능한 호스트만을 출력한다. 목록 스캔과 비슷한 기능이지만 속도가 더 빠르며 얼마나 많은 호스트가 살아있는지 스캔할 때 사용한다. **-sP**옵션은 ICMP 에코 요청과 TCP ACK 패킷을 80 포트에 기본으로 보낸다. 권한이 없는 유닉스 또는 **WinPcap**이 설치된 윈도우 사용자는 로우 패킷을 보낼 수 없어 **SYN** 패킷이 전달된다.

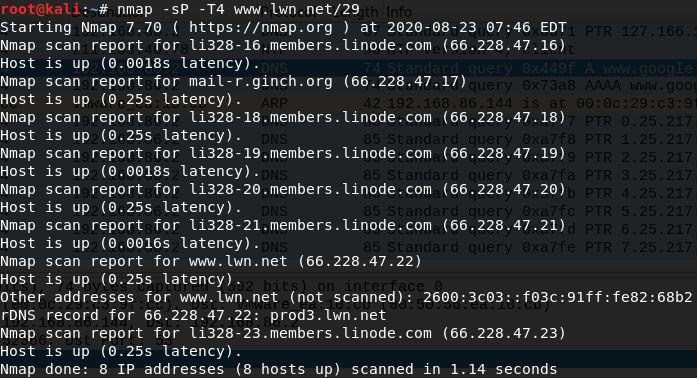


그림 2‑3 ping scan

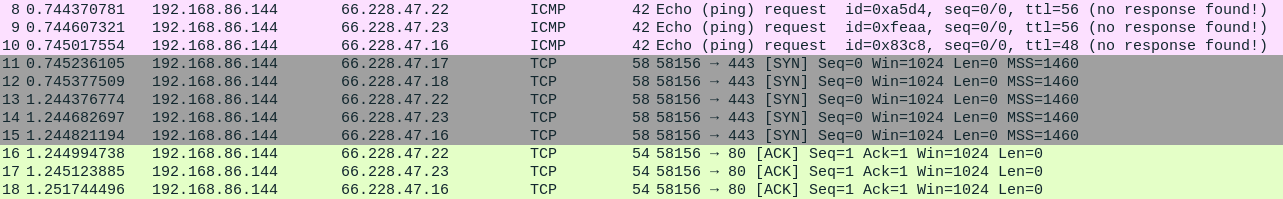


그림 2‑4 ping scan 패킷 분석

### TCP SYN ping(-PS)

**-PS** 옵션은 SYN 플래그 세트와 함께 빈 TCP 패킷을 보낸다. 기본 목적지 포트는 80이지만 다른 포트가 매개변수로 지정될 수 있다. 호스트에 SYN 플래그가 전송되면 RST 패킷이 되돌아온다.

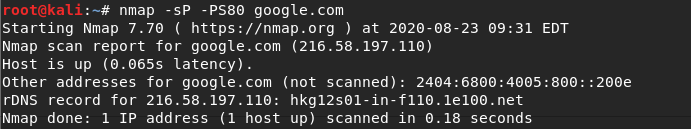


그림 2‑5 SYN ping



그림 2‑6 SYN ping 패킷 분석

### TCP ACK Ping(-PA)

**-PA** 스캔은 **-PS** 스캔과 유사하다. 차이점은 SYN 플래그 대신 ACK 플래그가 설정된다. SYN과 ACK 핑 프로브 둘 다 제공하는 이유는 방화벽을 회피하는 방법을 최대한 제공하기 위해서다. **Non-stateful** 방화벽과 라우터는 SYN 패킷을 차단하고 **stateful** 방화벽은 ACK 패킷 즉 예상치 못한 패킷을 차단한다. 따라서 SYN과 ACK 둘 다 적절하게 사용해야 한다.

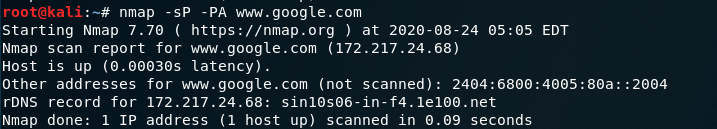


그림 2‑7 ACK ping

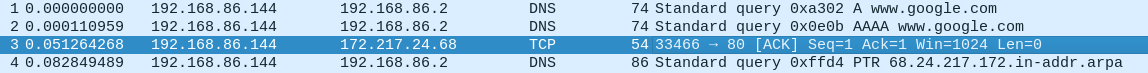


그림 2‑8 ACK ping 패킷 분석

## 포트 스캐닝

포트는 서비스와 연결된 통로이다. 관리자 페이지, 테스트 페이지 등에 불필요한 포트가 열려 있다면 공격자가 침해할 수 있는 보안 위협이 높아진다. 따라서 주기적으로 포트 스캔을 해 불필요한 포트가 열려 있는지 확인해야 한다. 예를 **들어 Shodan, censys, google hacking** 등과 같은 외부 OSINT 도구를 이용하거나 서버 대상으로 직접 Nmap과 같은 포트 스캐너를 활용해 오픈 된 포트, 취약점 정보, 서버 정보 등을 확인한다.

### TCP SYN can(-sS)

SYN 스캔은 **stealth** 스캔으로 불린다. 빠르게 동작하며 방화벽에 방해받지 않고 수천 개 포트를 스캐닝 할 수 있으며 Open, closed, filtered 상태에 관해 명확한 결과를 보여준다. SYN 패킷을

보내며 포트가 열려 있다면 서버 측은 SYN/ACK 패킷으로 응답한다. SYN/ACK 패킷은 포트가

열려 있다는 의미이기 때문에 Nmap은 RST패킷을 보내 연결을 종료한다.

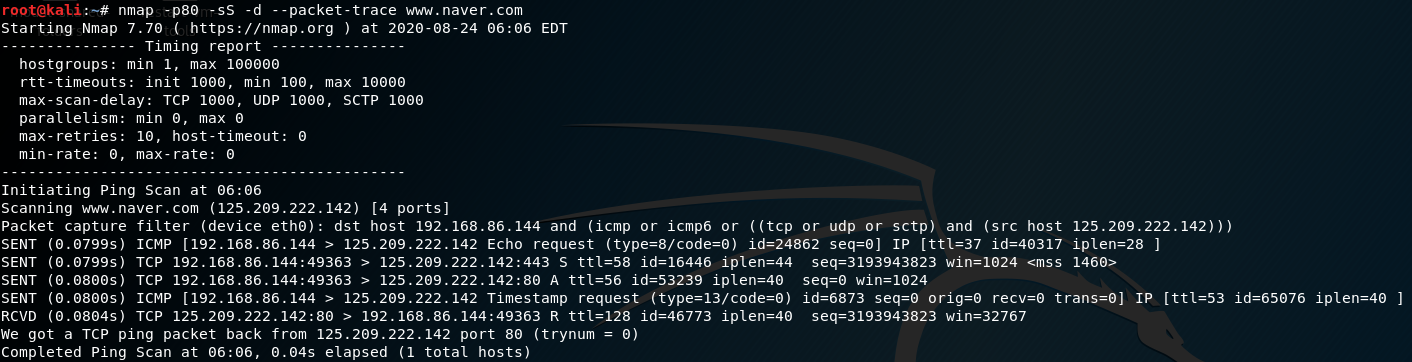


그림 2‑9 SYN scan

스캔을 시작하면 그림 2‑9와 같이 ping scan을 먼저 시도하고 그림 2‑10처럼 **stealth** 스캔을   
시도한다.

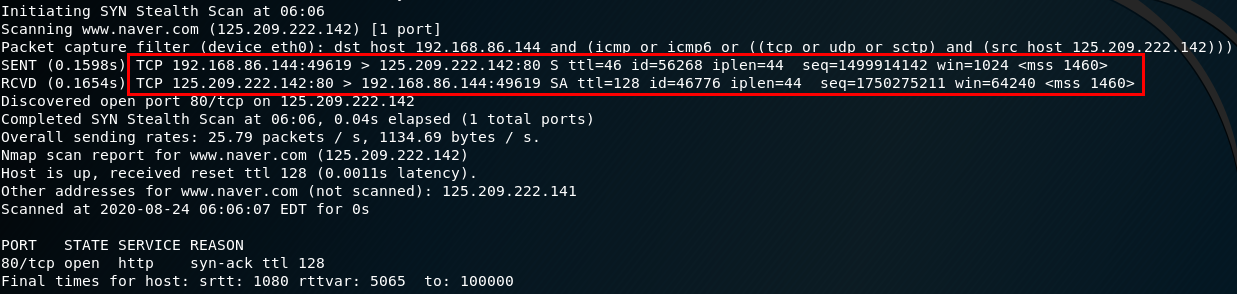


그림 2‑10 SYN scan 패킷 분석

### TCP Connect scan(-sT)

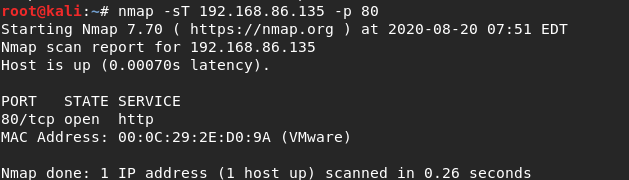


그림 2‑11 TCP Connect scan

**-sT** 스캔은 TCP CONNECT Scanning으로 connect() 함수를 사용한다. 각 포트에 TCP 3-way handshake 연결을 시도하므로 스캔 결과가 정확하지만 서버에 스캔 로그가 남아 탐지 가능성이 높다.

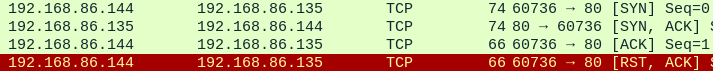


그림 2‑12 TCP Connect scan 연결 패킷 분석

포트가 열려 있다면 TCP 3-way handshake 연결 후 RST flag로 연결을 끊어 스캔을 완료한다.



그림 2‑13 TCP Connect scan closed 패킷 분석

포트가 닫혀 있다면 서버는 그림 2‑13과 같이 RST, ACK flag를 보내고 nmap은 closed를 출력한다.

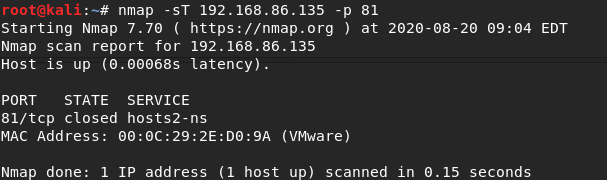


그림 2‑14 TCP Connect scan closed

### UDP scan(-sU)

DNS, SNMP, DHCP는 UDP 서비스다. 일반적으로 UDP 스캐닝이 TCP보다 더 느리고 어려워 일부 보안 관리자는 UDP 포트들을 무시한다. 하지만 탐지 가능한 UDP 서비스가 많아 공격자들의 표적 중 하나이므로 잘 관리해야 한다.

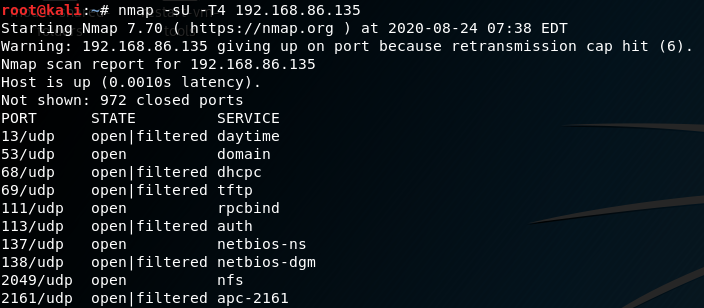


그림 2‑15 UDP scan

## Nmap NSE

**Nmap NSE**는 **Nmap Scripting Engine** 약자로 **Nmap**을 활용해 더 많은 정보를 수집하고자 할 때 사용하는 **Lua** 언어의 스크립트 도구다. NFS, SMB, RPC 등 상세한 서비스 정보를 수집할 수 있으며, 도메인 lookup, Whois 검색, 다른 네트워크 대역 서버의 백도어 설치 여부, 취약점 여부 등 많은 작업을 수행할 수 있다.

### **NSE 포맷 설명**

|  |
| --- |
| -- The Head Section --  description = [[  Show Test  ]]  categories = {"safe"}  -- The Rule Section --  portrule = function(host, port)  return port.protocol == "tcp"  and port.number == 80  and port.state == "open"  end  -- The Action Section --  action = function(host, port)  return "Hello world!"  end |

표 2‑4 NSE 포맷

* + - 1. description 필드

이 필드는 스크립트를 설명하는 필드로 어떤 스크립트인지 자세한 설명을 작성한다.   
소스코드에 영향을 주지 않으며 스크립트 실행 방법 등을 적는 것이 좋다.

* + - 1. Categories 필드

이 필드는 해당 스크립트가 어떤 카테고리에 포함돼 있는지 표시한다.

2가지 이상의 카테고리에 포함된다면 카테고리 이름을 모두 포함시키면 된다. 사용자는

이 카테고리를 참고해서 진단하기 때문에 누락하지 않아야 한다.

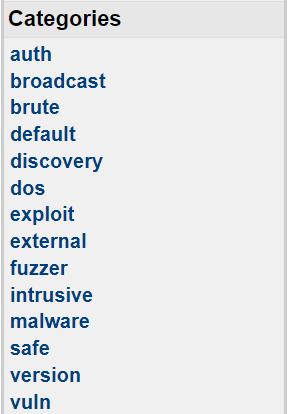


그림 2‑16 NSE categories

* + - 1. Rules 필드

이 필드는 스크립트 Action을 실행시킬지 결정한다. 메소드 응답 값은 **true, false**로 rule이 true일 경우만 실행된다. 스크립트에는 다음 나열돼 있는 메소드 중 적어도 하나가 포함돼야 한다.

|  |
| --- |
| **Prerule()**  **hostrule(host)**  **portrule(host, port)**  **postrule()** |

표 2‑5 NSE rules

**Prerule()**은 스크립트가 사전 스캐닝 단계에서 호스트가 스캔되기 전에 한 번 실행된다. **Hostrule()**과 **portrule()** 스크립트는 호스트의 각 배치가 정해진 후에 실행된다. **Postrule()** 스크립트는 모든 호스트 스캔이 완료된 후에 실행된다.

* + - 1. Action 필드

스크립트가 실행되는 부분이다. **Portrule()**이 일치됐을 때 메인 함수인 **action=function()**이 실행된다. 스크립트가 실행될 수 있는 모든 명령들이 **action**에 포함돼 있다.

### http-method.nse **분석**

웹서버에 파일 업로드, 삭제와 같은 악의적인 행위를 못하게 **HEAD, PUT, DELETE, TRACE** 등과 같이 불필요한 메소드는 비활성화 해야 한다. **http-method.nse** 스크립트는 HTTP 서버에 OPTIONS 메소드 요청을 보내 지원하는 메소드가 무엇인지 알아내, 잠재적 위험이 있는 메소드를 표시한다.

|  |
| --- |
| nmap --script=http-method.nse --script-args http-methods.retest=1 <target IP>  nmap --script=http-method.nse --script-args http-methods.url-path=/website <target IP> |

표 2‑6 http-mthod.nse 옵션

첫번째 사용법에서 **http-methods.retest** 인수는 개별적으로 사용해 HTTP 서버가 지원하는 메소드 목록을 출력한다. 와이어샤크로 패킷을 보면 **OPTIONS** 메소드를 HTTP 서버에 보내 허용돼 있는 메소드를 확인한 후 해당 메소드를 이용해 HTTP 서버에 다시 요청한다. 그림 2‑17과 같이 다시 요청한 응답 메시지에 포함돼 있는 응답 코드까지 확인해 결과로 출력한다.

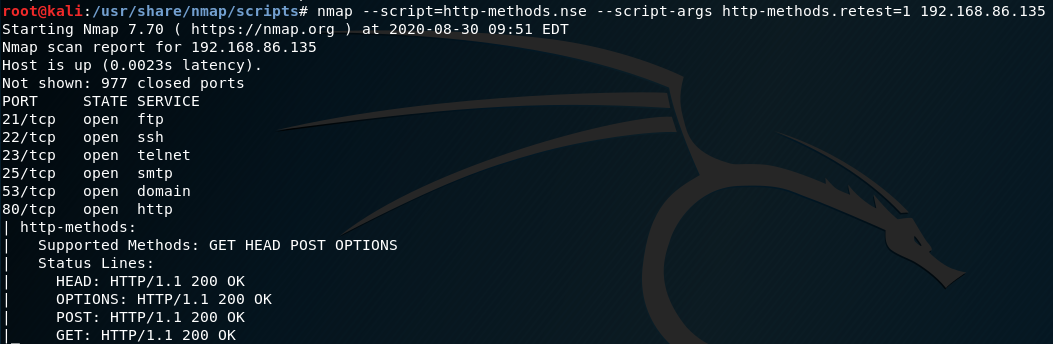


그림 2‑17 http-methods.retest 인수

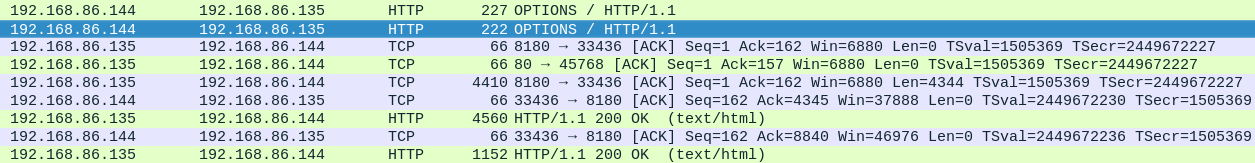


그림 2‑18 OPTIONS method



그림 2‑19 GET method

두번째 사용법에서 **http-methods.url-path=/website** 인수는 그림 2‑20과 같이 사용자가 지정한 경로에 요청한다. 요청 결과 80 포트 페이지는 **HET, HEAD, POST, OPTIONS** 메소드를 지원하고 8180 포트 페이지는 **GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS** 메소드를 지원한다.



그림 2‑20 http-methods.url-path=/website 인수

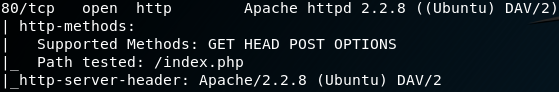


그림 2‑21 80 포트 지원 메소드

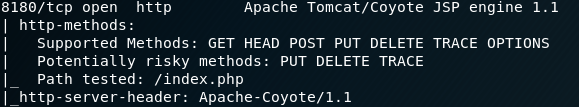


그림 2‑22 8180 지원 메소드

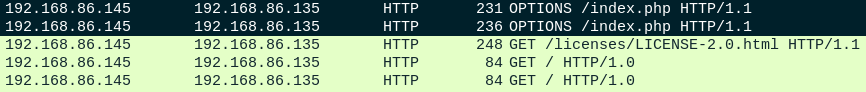


그림 2‑23 Options 메소드 패킷

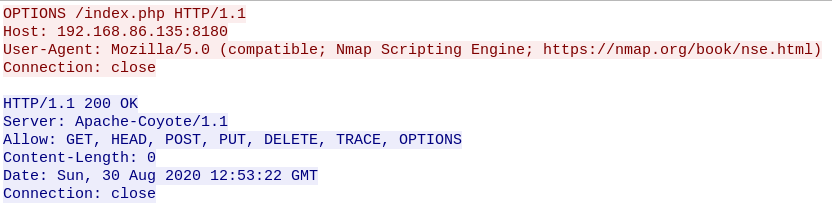


그림 2‑24 8180 지원 메소드 패킷 분석

### mysql-empty-password.nse

MySQL은 오라클, MS-SQL과 함께 시장 점유율이 가장 높은 데이터베이스 관리 시스템이다. **Mysql-empty-password.nse** 스크립트는 **mysql**의 root 또는 익명 계정 비밀번호를 설정하지 않았을 시 체크하는 스크립트다. SK infosec EQST insight 7월호 보고서에 따르면 DB 계정 무작위 대입 공격으로 인한 개인정보 유출사고가 증가하는 추세다. 개인정보 유출사고를 막으려면 DB 계정 관리는 필수이므로 빈 패스워드 또는 취약한 패스워드를 사용하는지 주기적으로 점검해야 한다.

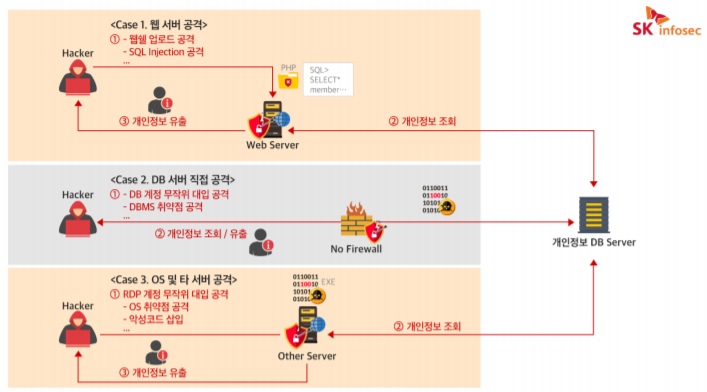


그림 2‑25 DB 공격 시나리오 흐름도

**Mysql port 3306**을 대상으로 실행 결과 anonymous와 root 계정이 빈 패스워드를 사용한다.

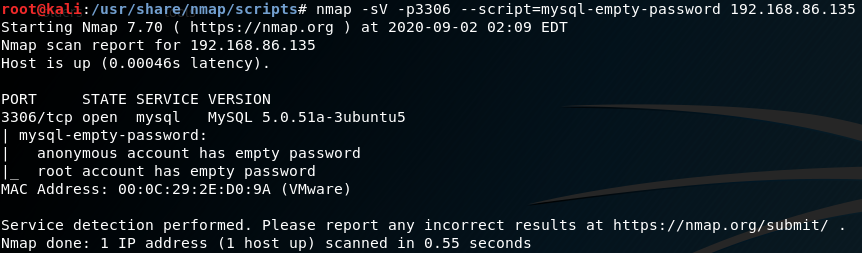


그림 2‑26 mysql-empty-password.nse

와이어샤크로 패킷을 보면 그림 과 같이 anonymous 계정으로 로그인 시도, response OK 패킷을

응답 받는다.

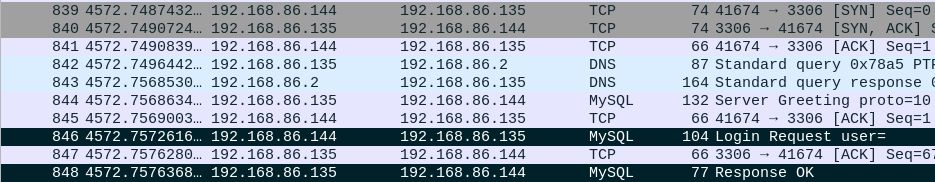


그림 2‑27 anonymous 계정 로그인 시도 패킷 분석

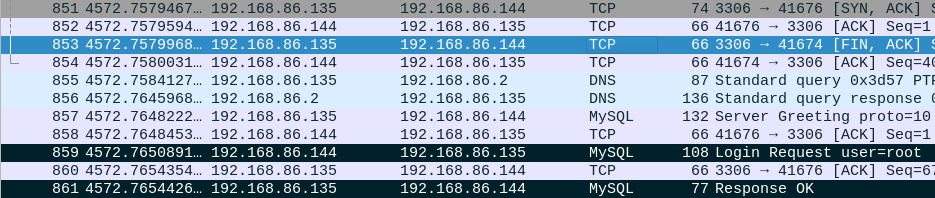


그림 2‑28 ROOT 계정 로그인 시도 패킷 분석

**Mysql-empty-password.nse** 스크립트는 anonymous와 root 계정을 대상으로만 패스워드를 체크한다. 점검 계정을 추가하려면 그림 2‑29 와 같이 users 변수에 계정 목록을 추가한다.

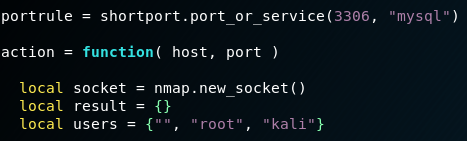


그림 2‑29 mysql-empty-password 소스코드

스크립트를 수정하거나 새로운 스크립트를 추가했다면 **nmap –script-update, nmap –script-updatedb** 명령어를 사용해 **Nmap DB**를 업데이트한다.

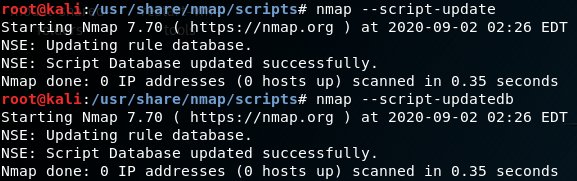


그림 2‑30 Nmap DB 업데이트

실행 결과 추가한 **kali** 계정을 출력한다.

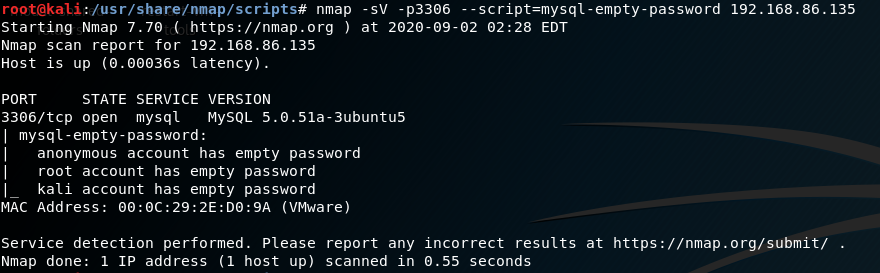


그림 2‑31 계정 추가 후 실행

### vulscan.nse

**vulscan**은 취약점 표준을 관리하는 CVE의 데이터베이스 등 8개 취약점 데이터베이스를 비교한 결과를 스캔한 정보와 결합해 출력하는 종합 취약점 관리 스크립트다.

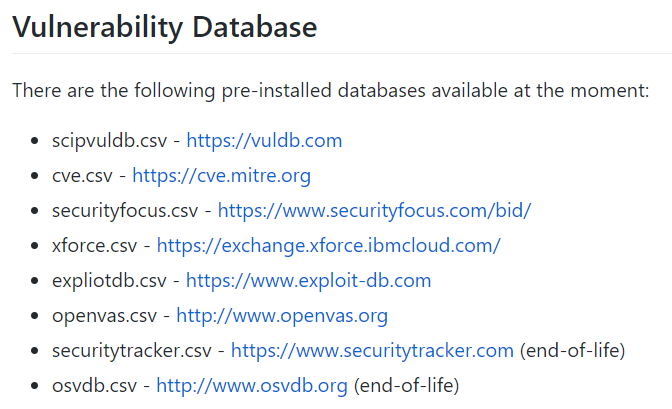


그림 2‑32 vulscan.nse DB 목록

Marc Ruef가 개발했으며, 깃허브에서 다운로드 해 사용한다.

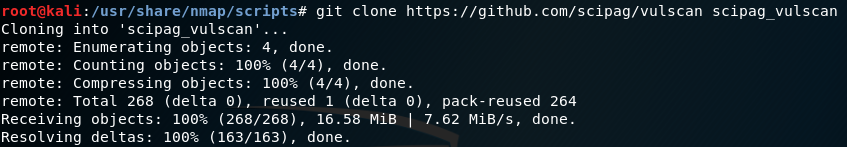


그림 2‑33 git clone

다운로드 되면 아래 명령어를 사용해 엔맵이 설치돼 있는 경로로 옮긴다.

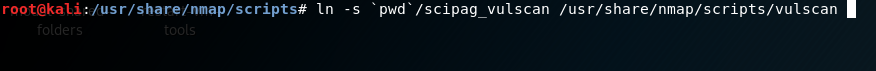


그림 2‑34 경로 이동

**Vulscan**을 실행하면 그림 2‑35와 같이 대상 서버의 취약점 번호와 간단한 설명이 출력된다.

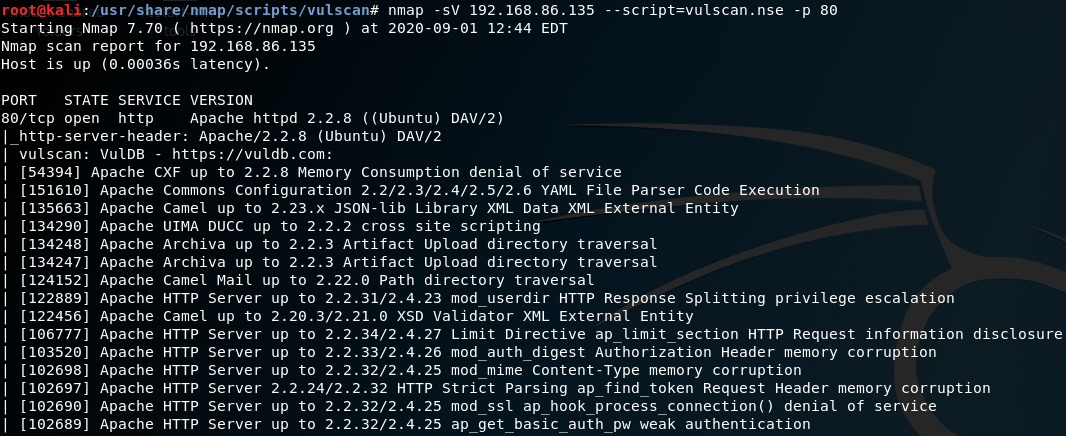


그림 2‑35 vulscan.nse

취약점 데이터베이스를 업데이트 하려면 그림 2‑36과 같이 8개 데이터베이스 CSV 파일을 다운로드 해 **vulscan** 폴더에 저장한다.

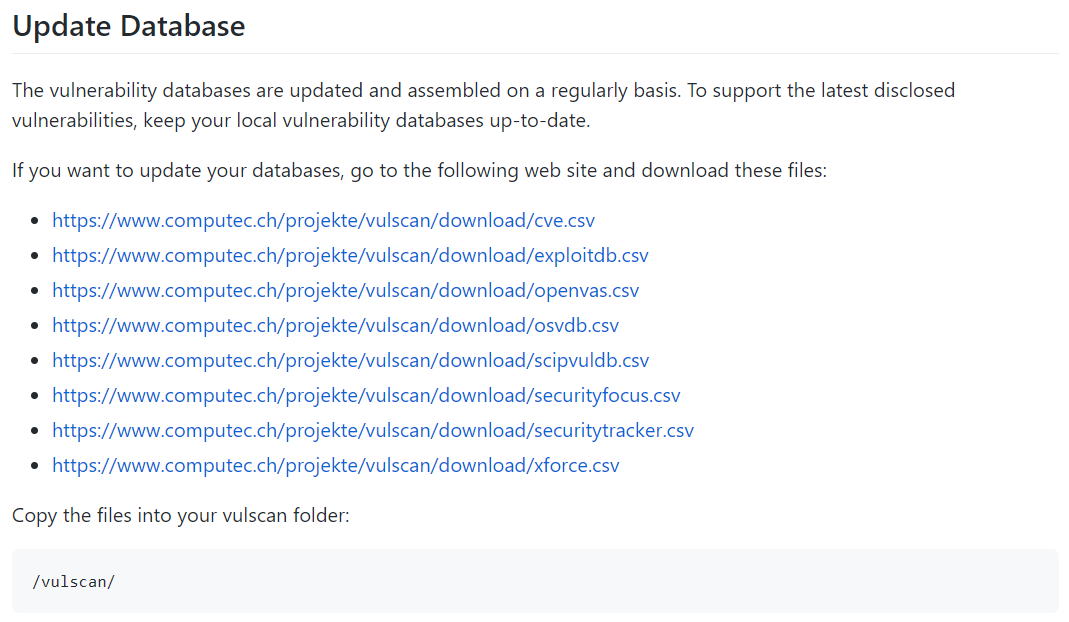


그림 2‑36 DB 업데이트 방법

# 참고 문헌

## 참조 보고서/논문/도서

|  |  |
| --- | --- |
| 보고서/논문/도서 명 | 저자 |
| Nmap NSE를 활용한 보안 취약점 진단 | 조정원, 박병욱, 이준형, 서준석 지음 |
| 엔맵 네트워크 스캐닝 | 고든 표도르 라이언 |

표 3‑1 단행본

## 참조 홈페이지

|  |
| --- |
| 참조 홈페이지 |
| <http://wildpup.cafe24.com/archives/264>  <https://hj-kwon.tistory.com/29>  <https://peemangit.tistory.com/88>  <http://www.linuxlab.co.kr/docs/00-05-2.htm>  <https://m.blog.naver.com/sol9501/70084530792> |

표 3‑2 참조 홈페이지