WebKnight 공개 웹 방화벽

IIS 웹서버 보안

최윤미
unia1012@is119.jnu.ac.kr

Abstract

WebKnight는 IIS 웹서버를 위한 오픈 소스 웹 방화벽이다.

서버관리자들이 서버를 운영함에 있어서 방화벽을 운영함으로써 쉽고 편리하게 대처할 수 있다는 장점이 있다.

WebKnight를 현재 운영 중인 서버에 설치하고 적용하는 방법과 어떠한 공격들을 막을 수 있으며, 얼마나 효과적인지를 조사하고 기술할 것이다.

Content

1. WebKnight 개요

1.1 소개

WebKnight는 AQTRONIX사(http://www.aqtronix.com/)에서 개발한 IIS 웹서버에 설치할 수 있는 공개용 웹 방화벽이다. WebKnight는 ISAPI 필터 형태로 동작하며, IIS 서버 앞단에 위치하여 웹서버로 전달되기 이전에 IIS 웹서버로 들어온 모든 웹 요청에 대해 웹서버 관리자가 설정한 필터 룰에 따라 검증을 하고 SQL Injection 공격 등 특정 웹 요청을 사전에 차단함으로써 웹서버를 안전하게 지켜준다. 이러한 룰은 정기적인 업데이트가 필요한 공격 패턴 DB에 의존하지 않고 SQL Injection, 디렉토리 traversal, 문자 인코딩 공격 등과 같이 각 공격의 특징적인 키워드를 이용한 보안필터 사용으로 패턴 업데이트를 최소화하고 있다. 이러한 방법은 알려진 공격 뿐만 아니라 알려지지 않은 공격으로부터도 웹서버를 보호할 수 있다.

또한, WebKnight는 ISAPI 필터이기 때문에 다른 방화벽이나 IDS에 비해 웹서버와 밀접하게 동작할 수 있어 많은 이점이 있다. MS의 URLScan과 마찬가지로 ISAPI 필터로써 inetinfo.exe 안에서 동작하므로 오버헤드가 심하지 않다. 해킹당한 한 웹사이트에 WebKnight를 적용하여 테스트한 결과 안정적인 웹서버 운영으로 인해 웹서버 속도가 오히려 빨라진 것을 느낄 수 있었다. 하지만 다량의 웹 트래픽이 발생되는 사이트에서는 사전에 충분한 검증을 거친 후에 적용할 필요는 있다.

1.2 주요 특징

• 낮은 보유 비용(Total Cost of Ownership)
WebKnight는 윈도우즈 인스톨러 패키지와 원격 설치 스크립트로 설치 가능해 사내에서 쉽게 WebKnight를 채택할 수 있다. 또한 WebKnight 설정을 바꾸기 위해 그래픽 사용자 인턴페이스를 제공한다.

• 운영 중 업데이트 가능
일부 설정의 변경을 제외하고 대부분의 설정 변경은 웹서버의 재가동을 요구하지 않아, 웹 사용자들에 대한 어떠한 서비스 장애 없이 설정을 변경할 수 있다. 성능상의 이유로 매 1분마다 이러한 변경을 탐지하여 적용한다.

• SSL 보호(protection)
다른 전통적인 방화벽과는 달리 WebKnight는 ISAPI 형태로 IIS의 일부로 동작하므로 HTTPS 상의 암호화된 세션들도 모니터링 및 차단할 수 있다.

• Logging
기본적으로 차단된 모든 요청에 대해 로그를 남기고, 로깅 전용 모드로 운영할 경우 추가적으로 모든 허용된 요청에 대해서도 로그를 남길 수 있다. 로깅 전용 모드는 공격을 차단하지는 않고 로그 파일에서 공격 사실을 조사하는데 도움을 줄 수 있다.

• HTTP Error Logging
WebKnight는 웹서버로부터 HTTP 에러들을 로그할 수 있도록 설정할 수 있다. 이 방법으로 '404 Not Found'와 같은 일반적인 에러나 ' 500 Server Error'와 같이 보다 심각한 로그들도 기록할 수 있다. 에러 로그를 이용하여 공격을 탐지하거나 깨진 링크를 발견하거나 잘못된 설정도 쉽게 발견할 수도 있다.

• 웹기반 애플리케이션과의 호환성
WebKnight는 Frontpage Extensions, WebDAV, Flash, Cold Fusion, Outlook Web Access, SharePoint 등과도 호환이 잘 이루어진다.

New in WebKnight 2.0

• 향상된 Scanning
Scanning 엔진이 향상되었고 확장되었다. 임의의 공격이나 데이터를 위한 참조 헤더와 User Agent 헤더를 스캔하는 것이 가능하다.

• 인증 Scanning
인증 Scanning은 계정의 무작위 대입 공격이나 시스템 계정의 서비스 거부 공격을 스캔하는 것을 허용했다. 또한 취약한 패스워드를 스캔할 수 있다.

• 연결 Control/Monitoring
특정 IP주소 또는 범위로부터 들어오는 트래픽을 차단하거나 Monitoring 할 수 있다. 또한 중요한 파일에 대한 접근을 감시하거나 하나의 IP주소에서 들어오는 요청의 수를 제한할 수 있다.

• Robots 차단

거대한 Robot 데이터베이스를 만들어 특정 타입에 Robot을 차단하거나 허가 가능하다. 또한 악의적인 Robot을 위해 bot trap을 세우고 공격적인 Robot을 차단할 수 있다.

• Hot Linking 차단

이미지나 파일 다운로드와 같이 특정 타입의 파일을 Hot Link 또는 Direct Link 하는 것이 차단될 수 있다.

2. WebKnight 설치 및 제거

2.1 WebKnight 설치

① 아래 URL에서 WebKnight 2.0(2006.12.24 릴리즈)을 다운로드 받는다.
http://www.aqtronix.com/downloads/WebKnight/2006.12.24/WebKnight.zip

② 다운로드 받은 WebKnight.zip 파일을 압축 푼 후 그 안의 Setup 폴더로 이동한다.

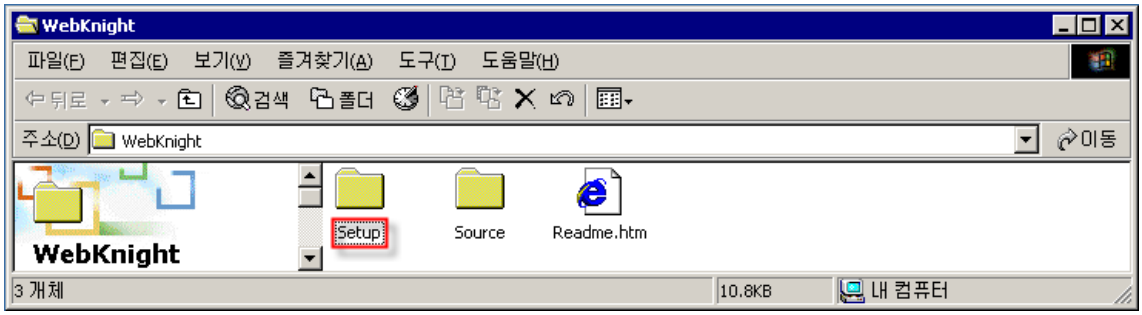

그림 1 압축이 풀려 생성된 WebKnight 폴더

③ install.vbs을 더블클릭하면 스크립트를 이용하여 설치가 가능하다. 여기서는 Windows Installer를 사용하기 위하여 WebKnight.msi 파일을 더블클릭한다.

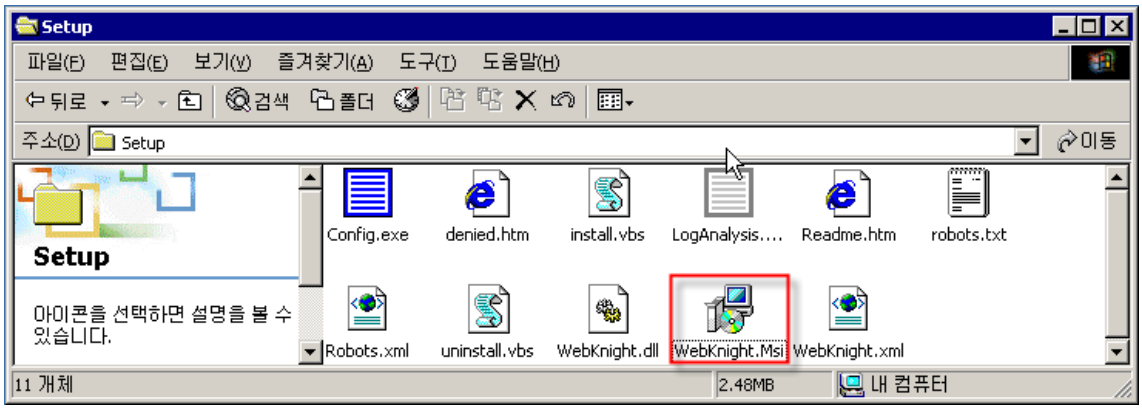

그림 2 WebKnight 설치

④ 다음 그림은 WebKnight가 Windows Installer에 의해 설치되는 화면이다.

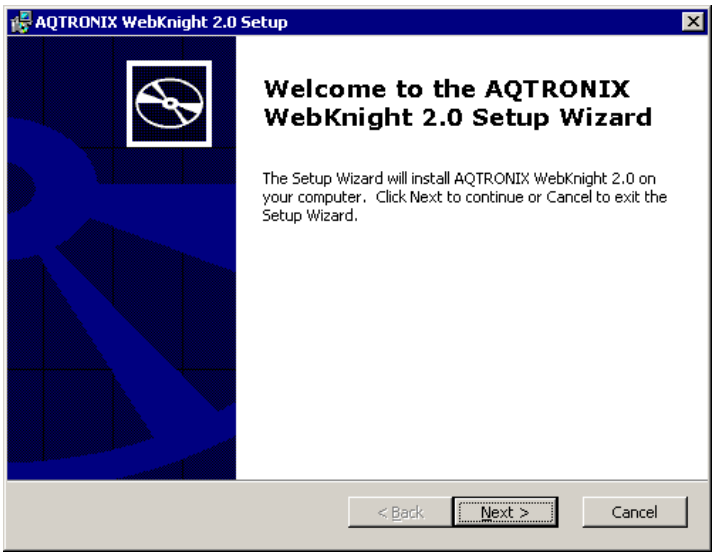

그림 3 Windows Installer가 실행된 화면

⑤ 라이센스 동의 후 일반적으로 “Typical”을 선택하여 설치한다.

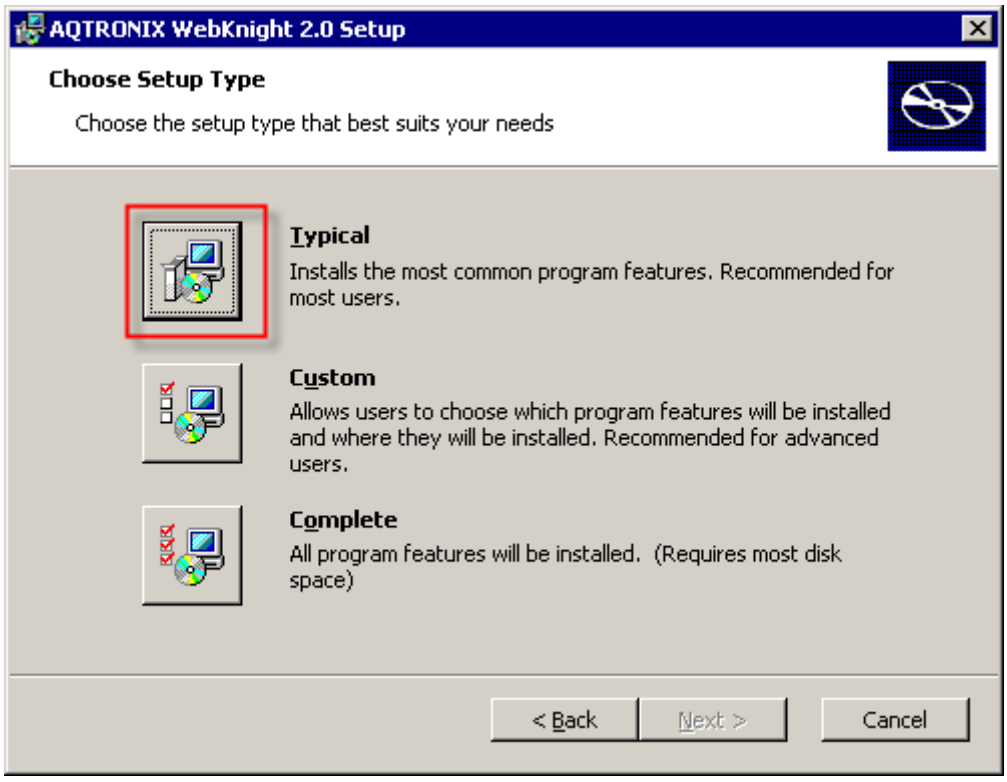

그림 4 “Typical” 설치

⑥ 설치가 완료되면 아래와 같은 메시지가 나타난다.

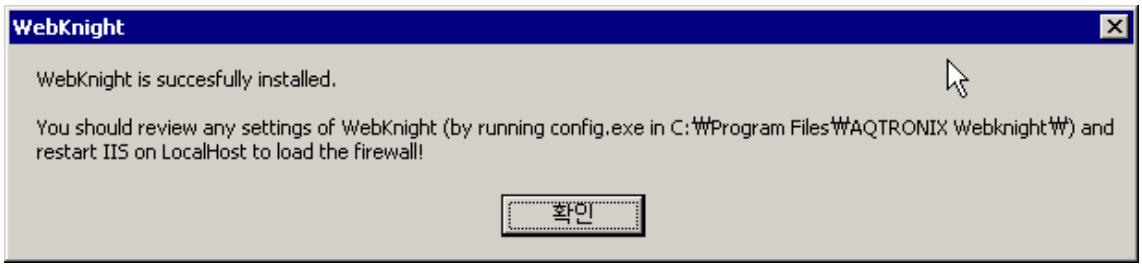

그림 5 설치 완료

⑦ 기본적으로 설치 경로는 C:₩Program Files₩AQTRONIX₩WebKnight 이다.

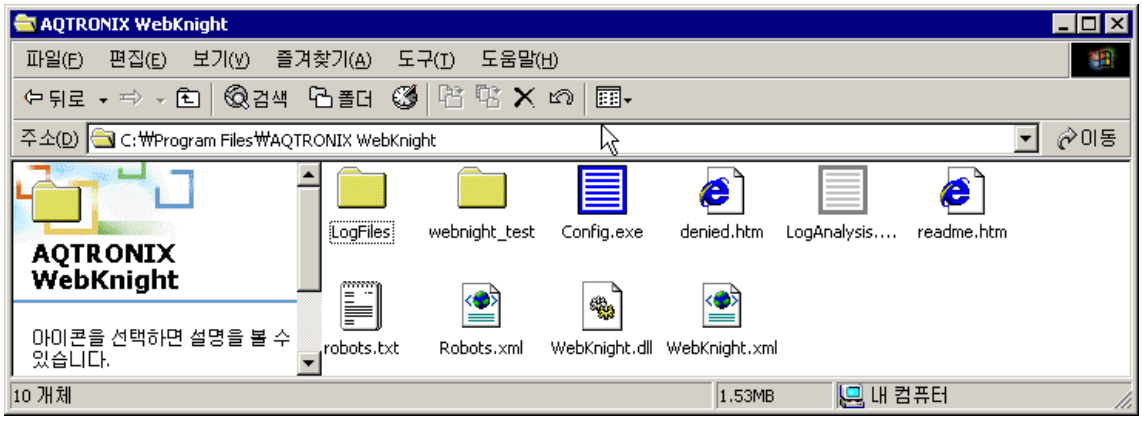

그림 6 WebKnight가 설치된 폴더

⑧ IIS 서버를 재시작한다. 정상적으로 설치가 완료되면 아래와 같이 웹사이트 등록 정보의 “ISAPI 필터”에 WebKnight 필터가 정상적으로 적용이 된 것을 확인할 수 있다.

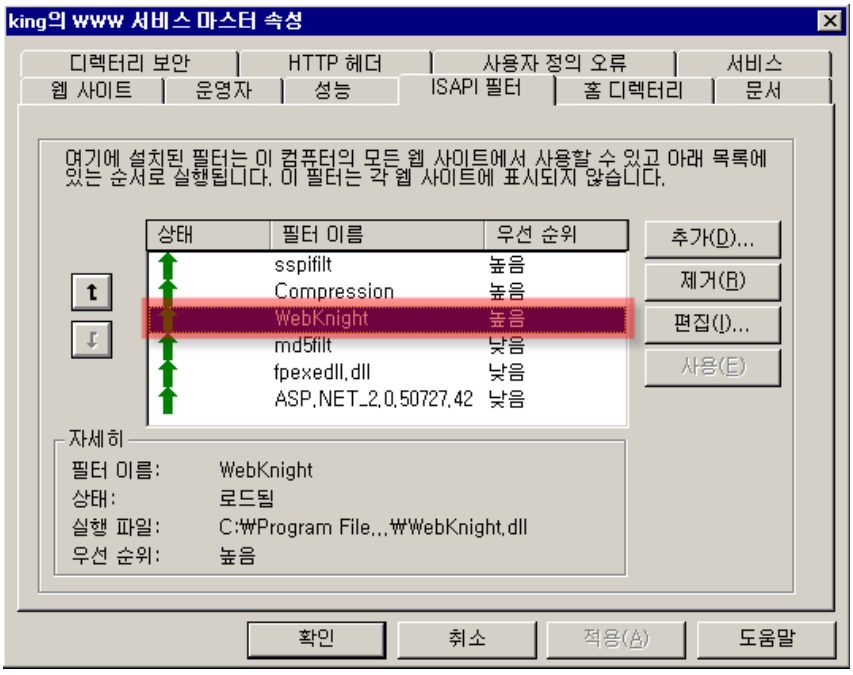

그림 7 WebKnight 필터 확인

■ 글로벌 필터로 수동 설치

① 압축 해제 후 생성되는 Setup 폴더를 C:₩Program Files₩AQTRONIX WebKnight와 같은 서버 내의 로컬 폴더를 생성하고 여기에 복사한다.

② 인터넷 정보 서비스를 연다.

③ 서버 이름(사이트 이름이 아님)에서 우측 마우스를 클릭하여 "등록정보"를 선택한다.

④ 마스터 속성 리스트에서 "WWW 서비스"를 선택하고, "편집" 버튼을 누른다.

⑤ "ISAPI 필터" 탭을 선택하고 "추가" 버튼을 클릭한다.

⑥ "필터 등록 정보"가 나타나면 필터 이름과 실행 파일 경로를 입력한다.
(예를들어, 필터 이름 : WebKnight, 실행 파일 경로 : C:₩Program Files₩AQTRONIX WebKnight₩WebKnight.dll)

⑦ "OK" 버튼을 누르고 대화상자를 빠져 나간다.

⑧ IIS를 재시작한다.

■ 사이트 필터로 수동 설치

① 압축 해제 후 생성되는 Setup 폴더를 C:₩Program Files₩AQTRONIX WebKnight₩W3SVC1 과 같은 서버내의 로컬 폴더를 생성하여 여기에 복사한다.(단, 각 WebKnight 설치를 위한 unique한 폴더를 가져야 한다.)

② 인터넷 정보 서비스를 연다.

③ 사이트 이름(서버 이름이 아님)에서 우측 마우스를 클릭하여 "등록정보"를 선택한다.

④ "ISAPI 필터" 탭을 선택하고 "추가" 버튼을 클릭한다.

⑤ "필터 등록 정보"가 나타나면 필터 이름과 실행 파일 경로를 입력한다.
(예를들어, 필터 이름 : WebKnight, 실행 파일 경로 : C:₩Program Files₩AQTRONIX WebKnight₩W3SVC1₩WebKnight.dll)

⑥ "OK" 버튼을 누르고 대화상자를 빠져 나간다.

⑦ Setup 폴더 아래의 config.exe 파일을 실행해서 "Global Filter Capabilities" 섹션에서 "Is Installed As Global Filter"의 체크를 해제한다.

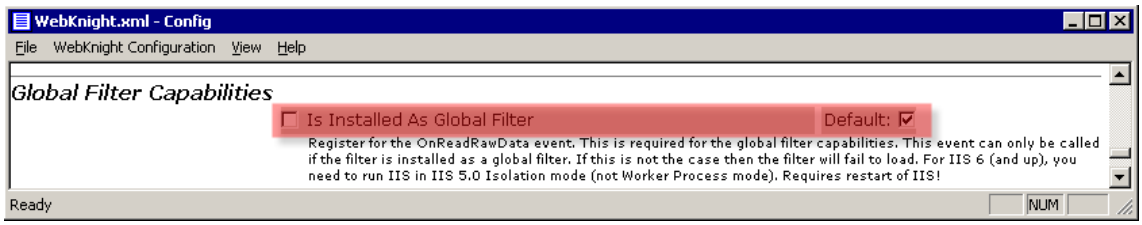

그림 8 사이트 필터 설치 설정

⑧ IIS를 재시작한다.

2.2 WebKnight 제거

① WebKnight를 제거하기 전에 IIS 서버를 중지시킨다.

② WebKnight를 제거하기 위해 WebKnight.msi 파일을 더블클릭한다.(uninstall.vbs을 더블클릭하면 스크립트를 이용하여 제거가 가능하다.)

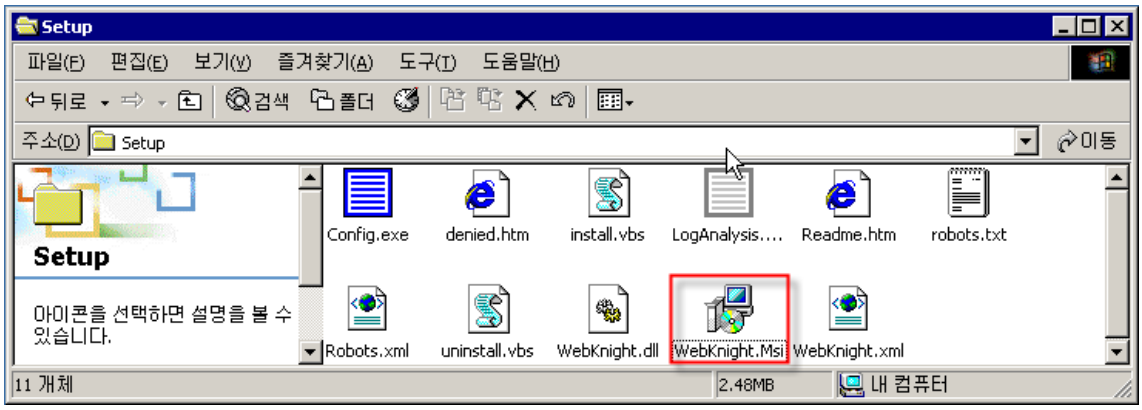

그림 9 WebKnight 제거

③ “Remove” 를 클릭한다.

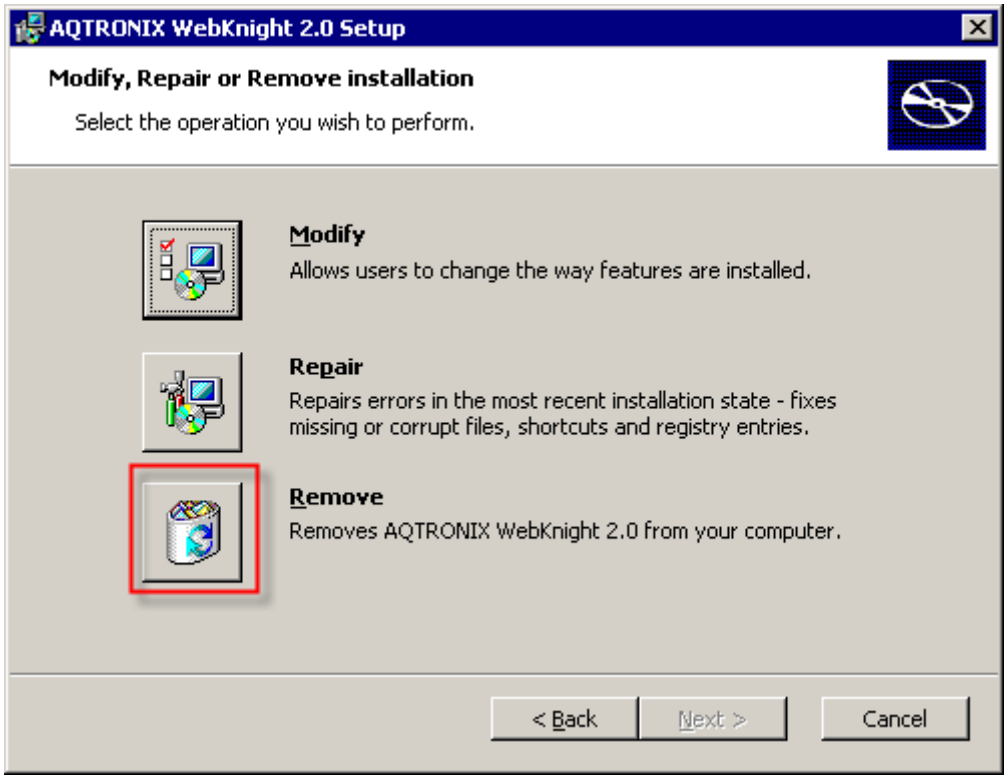

그림 10 “Remove” 선택

④ 제거가 완료되면 아래와 같은 화면이 나타난다.

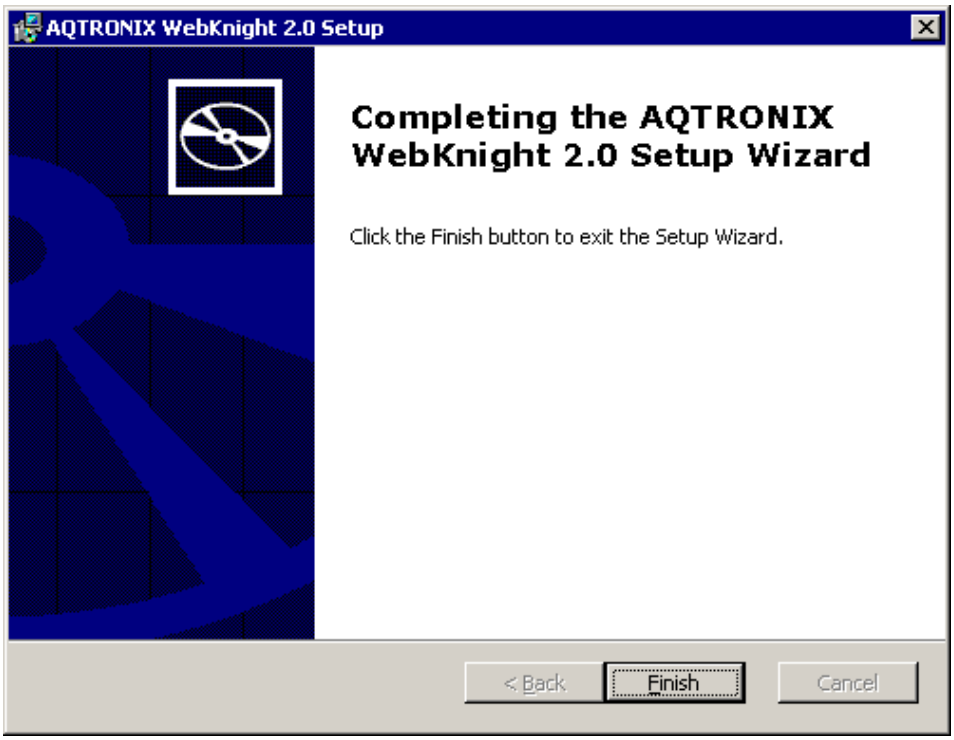

그림 11 WebKnight 제거 완료

⑤ IIS 서버를 재시작한다.

■ 수동 제거 - 인터넷 서비스 등록정보의 ISAPI필터에서 WebKnight를 선택하고 제거한다.

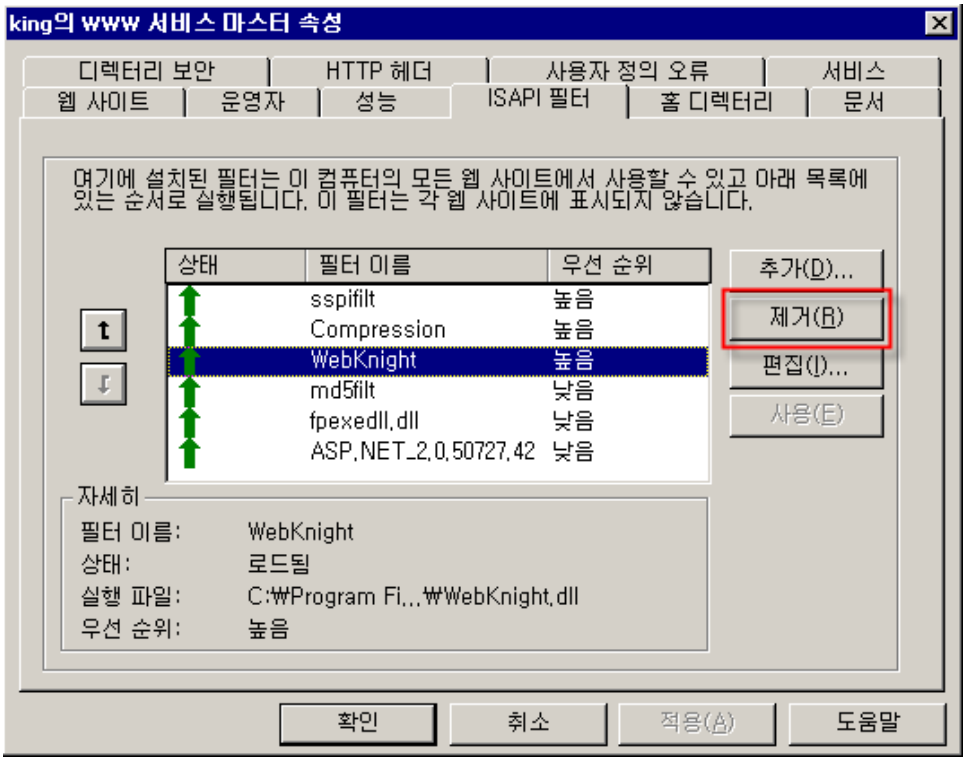

그림 12 WebKnight 수동 제거

3. WebKnight 설정

3.1 초기 설정

WebKnight는 SQL Injection 공격차단, 허용하지 않는 파일 또는 확장자에 대한 접속 차단 등 웹 공격에 대해 대단히 다양한 차단기능을 제공해 주고 있다. 또한 기본적으로 이러한 차단기능이 설정되어 설치와 동시에 적용이 되는데 이 차단기능이 정상적인 웹 접속을 차단할 수도 있다. 따라서 설치이후 자신의 웹사이트 환경에 맞게 적절하게 설정하는 과정을 반드시 거쳐야 한다. 실제 설치보다는 초기 설정에 많은 노력과 시간을 들여야만 한다. 설정과정을 통해 오히려 웹 공격의 다양한 패턴을 익힐 수 있는 기회도 될 수 있을 것이다.

먼저, WebKnight 설치 이후 해당 웹사이트에 방문해서 정상적으로 웹 요청 및 응답이 이루어지는지 확인을 하고, 접속이 차단될 경우 WebKnight의 로그를 참조하여 어떠한 룰에 의해 요청이 차단되었는지 이 룰을 수정하여야 한다.

디폴트 설치 시 로그파일의 위치와 설정프로그램은 다음과 같다.

○ 로그파일 : C:₩Program Files₩AQTRONIX WebKnight₩Log Files₩YYMMDD.log
○ 설정프로그램 : C:₩Program Files₩AQTRONIX WebKnight₩config.exe

WebKnight 설치 후 웹 접속시 다음과 같은 경고 화면이 뜰 수 있다.

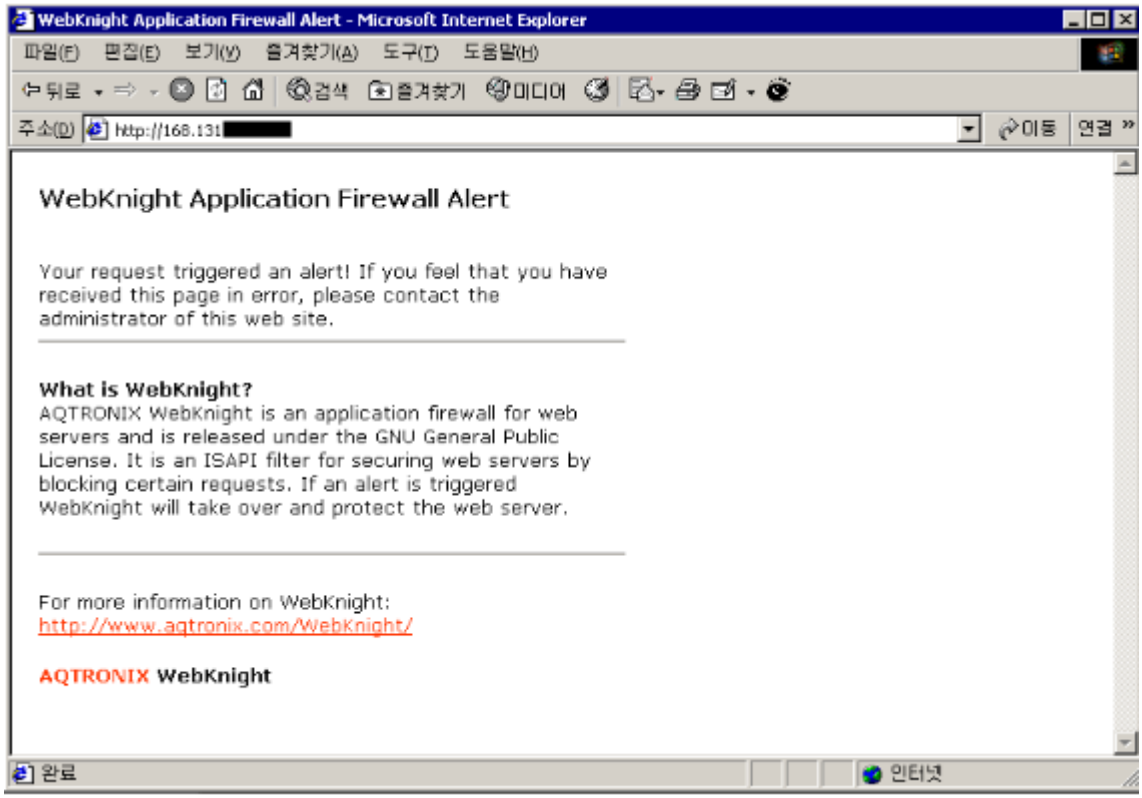

그림 13 WebKnight 경고 화면

이 화면은 WebKnight에서 필터 룰에 의해 차단을 시킨 후 웹 접속자에게 보내는 기본 경고화면이다. 정상적인 웹 요청을 했는데도 불구하고 이와 같이 차단된다면 로그파일을 열어 “BLOCKED” 메시지를 확인하고 어느 룰에서 차단되었는지 찾아 설정파일에서 이를 해제하여

야 한다. 디폴트 설치의 경우 WebKnight의 로그파일은 설치 후 IIS 웹서버러를 재가동하게 되면 “C:₩Program Files₩AQTRONIX WebKnight₩LogFiles” 폴도가 생성되고 그 하위에 일자별로 로그파일이 생성된다.

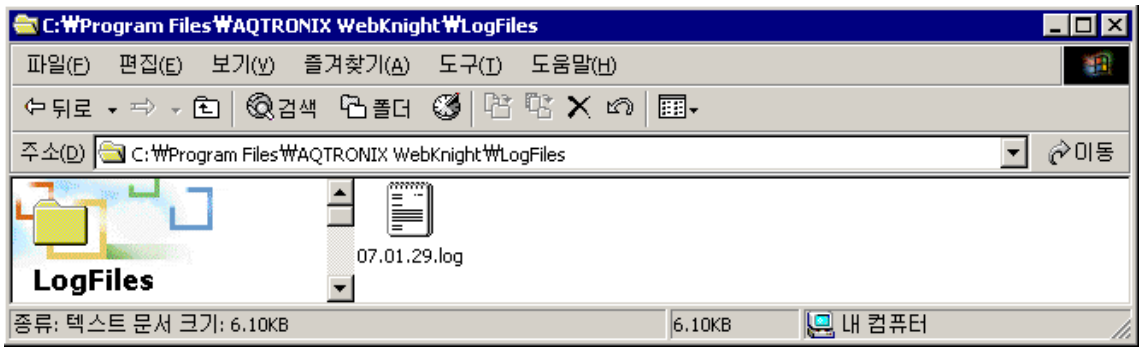

그림 14 WebKnight 로그 파일

기본적인 로그파일의 각 필드는 다음과 같다.

Time ; Site Instance ; Event ; Client IP ; Username ; Additional info about request(event specific)

3.2 룰 설정

WebKnight가 설치된 폴더에 Config.exe 파일을 실행시키고 WebKnight.xml을 선택하여 다양한 필터링 룰을 설정할 수 있다.

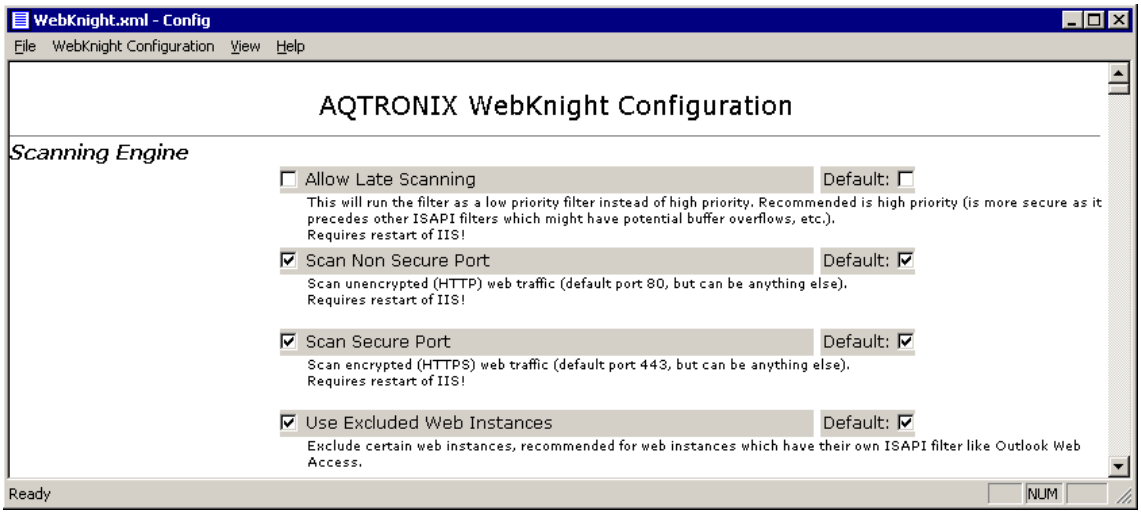

그림 15 WebKnight Config.exe 파일 실행 화면

○ **Scanning Engine** : 암호화 포트(HTTPS), 비암호화 포트(HTTP)에 대한 모니터링 기능 설정

• Allow Late Scanning - 높은 우선권 대신에 낮은 우선권 필터처럼 WebKnight가 동작하게 함. 높은 우선권 추천(잠재적 버퍼 오버플로우를 가진 이전의 다른 ISAPI 필터들과 달리 더 안전하다.)

• Scan Non Secure Port - 비암호화(HTTP) 웹 트래픽을 스캔한다.(디폴트 80번 포트, 다른 포트일 수 있음) IIS 웹서버 재가동이 필요하다.

• Scan Secure Port - 암호화(HTTPS) 웹 트래픽을 스캔한다.(디폴트 443 포트, 다른 포트일 수 있음) IIS 웹서버 재가동이 필요하다.

• Use Excluded Web Instances - 신뢰할 수 있는 웹 인스턴스들을 제외한다. Outlook Web Access와 같이 그들 자신의 ISAPI 필터를 가지는 웹 인스턴스들의 경우 추천한다.

>> Excluded Web Instances - WebKnight의 스캔으로부터 제외된 웹 인스턴스 목록이다. 그 예로 Outlook WebAccess 웹 사이트가 있다.(인스턴스 100 사이트에서 시작함)

○ **Incident Response Handing** : 공격 발생시 WebKnight가 어떻게 행동할지를 결정하며, 기본적으로 경고화면인 denied.htm으로 redirect하고 웹 요청을 차단하지만, 차단하지 않고 로그만 남기게 할 수도 있음

• Response Directly - 공격이 발견되면, 클라이언트에게 표준 메시지를 즉시 응답을 보낸다. 이때 보내진 메시지는 (WebKnight 폴더의)denied.htm 파일의 내용이다.

• Response Redirect - 공격이 발견되면, 클라이언트를 정해진 URL로 redirect 시킨다.(URL은 'Response Redirect URL' 에 명시됨)

>> Response Redirect URL - 만약 'Response Directly' 가 체크 해제되고, 'Response Redirect' 가 체크 상태이면 클라이언트를 이곳에 입력한 주소로 redirect 시킨다. 입력한 내용은 절대 경로 또는 상대 경로가 될 수 있다.

• Use Response Status - 공격이 탐지될 때마다 클라이언트로 보내질 HTTP 응답 상태로서

'Response Status'의 값을 사용한다. 이것은 입력된 URL에 대한 클라이언트의 redirect를 사용하지 않을 때 작동한다.

>> Response Status - 이것은 공격이 탐지될 때 클라이언트로 보내질 '31337 No Hacking' 또는 '404 Object Not Found'와 같은 HTTP 응답 상태이다.

• Response Drop Connection - 공격이 탐지할 때마다 요청 후 응답을 기다리는 클라이언트와의 연결을 중단한다.

• Response Log Only - 만약 공격이 발견되면 웹서버는 요청을 차단하지 않고, 오로지 로그만 남기고 요청을 실행, 처리한다.

○ Logging : 로깅 여부, 로그 시간대, 로그 항목(클라이언트 IP, 사용자 명 등) 등을 설정

• Enabled - 로깅을 활성화 또는 비활성화를 선택한다. 방화벽 재시작 요구.

>> Log Directory - 로그 파일이 위치할 폴더를 설정한다. 방화벽 재시작 요구.

• Use GMT - 로그 날짜와 시간을 GMT로 설정한다. 방화벽 재시작 요구.

• Per Process Logging - 웹서버 하나의 프로세스에 하나의 로그 파일을 만든다. 하나 이상의 프로세스를 동시에 호스트 필터 가능한 웹서버를 위한 것이다. 방화벽 재시작 요구.

>> Log Retention - 로그 파일을 보관하는 기간. 방화벽 재시작 요구.

• Log Client IP - 클라이언트의 IP 주소를 기록한다.

• Log User Name - 로긴한 클라이언트의 유저네임을 기록한다.

• Log Allowed - 차단된 응답들의 로그를 남기는 것에 추가로 허가된 응답에 대한 로그를 남길 수 있다. 이것은 시스템의 과부하를 일으키므로 추천하지 않는다.

• Log HTTP VIA - 만약 클라이언트가 하나 이상의 프록시를 사용할 경우 원래 요청이 어디에서 왔는지 단서를 가진 'Via:' 헤더를 기록한다. Note: 모든 사용된 프록시에 대한 로그를 남길 수는 없다. (헤더를 제거하거나 가지고 있지 않은 프록시의 경우)

Log the 'Via:' header to have a clue where the original request came from (if the client uses 1 or more proxies). Note: you will not be able to log all used proxies (certain proxies don't have or remove this header)!

• Log HTTP X FORWARDED FOR - 'X_Forwarded_For:' 헤더를 기록한다. 어떤 프록시들(NetCache와 같은)은 요청의 출발지 IP주소를 가리키는 요청에 이 헤더를 추가한다.

• Log User Agent - 클라이언트 user agent를 기록한다. 이것은 소프트웨어/툴이 공격을 행하는데 이용된 것을 가리킬 수 있다. 그러나 이것은 악용을 알리기 위한 필수적인 정보가 아니다.

• Log HTTP Client Errors - '404 Not Found'와 같은 HTTP 클라이언트 에러를 기록한다. 이 에러들은 '4'로 시작한다.

• Log HTTP Server Errors - '501 Not Implemented'와 같은 HTTP 서버 에러를 기록한다. 이 에러들은 '5'로 시작한다.

○ Connection : 특정 IP 주소에 대한 모니터링, 접근 거부, 요청 카운트 제한 또는 요청 카운트 증가 시간을 제한

• Use Monitored IP Addresses - 'Monitored IP Addresses'에 명시된 IP주소의 요청 기록에 의하여 트래픽을 감시한다.

>> Monitored IP Addresses - 이것은 감시받을 IP주소와 IP 범위이다. 범위를 나타내기 위해 '*' ('1.*.*.*') 와 CIRD 표기법('1.0.0.0/24')을 사용할 수 있다.

• Use Denied IP Addresses - 'Denied IP Addresses'에 명시된 IP주소로 부터의 접근을 차단하고 그 요청을 기록한다.

>> Denied IP Addresses - 이것은 차단된 IP주소와 IP 범위이다. 범위를 나타내기 위해

‘*’ (‘1.*.*.*’) 와 CIRD 표기법(‘1.0.0.0/24’)을 사용할 수 있다.
• Use Connection Requests Limit - IP주소가 요청할 수 있는 요구의 수를 제한한다.
>> Connection Requests Limit Max Count - 어떤 상당량의 시간 안에 만들어 질 수 있는 요구의 수.
>> Connection Requests Limit Max Time - 요청은 입력된 시간(분)안에 카운트된다.

○ Authentication : 공백이나 사용자 이름과 같은 패스워드, 또는 자주 사용되는 패스워드 입력을 차단하고 최대 인증 횟수나 최대 시간을 설정하여 Brute Force 공격을 차단
• Scan Authentication Excluded Web Instances - 또한 이 이벤트에 제한된 웹 인스턴스를 스캔한다. 제한된 웹 인스턴스는 인증 시도가 없으면 방화벽으로부터 스캔되지 않는다.
• Deny Blank Passwords - 패스워드 입력 없는 경우 인증 시도를 차단한다.
• Deny Same Password As Username - 사용자 이름과 같은 패스워드 입력의 경우 인증 시도를 차단한다.
• Use Denied Default Passwords - ‘Denied Default Passwords’에서 기본적이고 자주 사용된 패스워드를 가지고 인증 시도를 차단한다.
>> Denied Default Passwords - 이것은 인증에 사용될 수 없는 기본적은 패스워드들이다.
• Deny System Accounts - 시스템의 영향을 미칠 수 있는 중요한 계정(IUSR_SERVERNAME, IWAM_SERVERNAME, SYSTEM, NETWORK SERVICE, TsInternetUser...)에 대한 인증 시도를 차단한다.
• Use Deny Account Brute Force Attack - 무작위 대입 공격과 서비스 거부 공격을 차단한다. 정해진 시간 안에 인증 시도를 카운트함으로써 공격을 탐지한다.
>> Deny Account Brute Force Attack Max Count - 한 IP주소에 대한 정해진 시간 안에 허락된 인증 최대 횟수
>> Deny Account Brute Force Attack Max Time - 입력된 시간(분)동안 인증 시도의 수를 카운트한다.
• Use Allowed Accounts - ‘Allowed Accounts’에 계정에 대한 인증 시도를 모두 허가한다.
>> Allowed Accounts - 이것은 인증이 허가된 계정 목록이다.
• Use Denied Accounts - ‘Denied Accounts’에 계정에 대한 인증 시도를 차단한다.
>> Denied Accounts - 이것은 명백하게 인증이 차단된 계정 목록이다.
• Scan Account All Events - 모든 다른 ISAPI 이벤트에 사용되는 계정들은 스캔될 수 있고, 인증이 허가되지 않은 계정의 요구는 차단이 가능하다.

○ Request Limits : 컨텐츠 길이, URL 길이, 쿼리스트링 길이 등을 제한
• Limit Content Length - ‘Max Content Length’에 명시된 값에 따라 요청의 헤더에 Content-Length의 값을 제한한다. 이러한 방법은 요청된 서버에서 보내진 바이트의 수를 제한 할 수 있다.
• Limit URL - ‘Max URL’에 명시된 값에 따라 URL의 길이(URL의 ‘?’ 전까지의 길이)를 제한한다. 긴 URL의 경우 공격으로 의심된다. 운영 체제가 수락하는 길이보다 더 긴 URL은 사용해서는 안된다.
• Limit Querystring - ‘Max Querystring’에 명시된 값에 따라 querystring의 길이(URL에서 ‘?’ 이후의 길이)를 제한한다.
• Limit HTTP Version - 웹서버에 대한 모든 요청은 HTTP 버전(‘HTTP/1.1’)이 명시되어 있다. ‘Max HTTP Version’에 명시된 값에 의해 이것의 길이를 제한할 수 있다.
>> Max Content Length - 웹서버로 보내지는 요청의 헤더에 Content-Length의 최대 값

>> Max URL - URL의 최대 길이
>> Max Querystring - querystring의 최대 길이
>> Max HTTP Version - HTTP 버전 문자열의 최대 길이
• Use Allowed HTTP Versions - 'Allowed HTTP Versions'에 명시된 HTTP 버전만을 허가한다.
>> Alowed HTTP Versions - 이것은 허가된 HTTP 버전 목록이다.
• Use Max Headers - 추가된 헤더의 길이를 제한한다. 'Max Headers'에 커스텀 헤더와 각 헤더를 위한 최대 길이를 명시할 수 있다.
>> Max Headers - 이것은 길이가 특정한 값으로 제한된 헤더이다.

○ **URL Scanning** : URL Encoding 공격 차단, 상위 패스(..) 차단, URL 백슬래쉬(₩) 차단, URL 인코딩(%) 차단, 특정 URL 스트링 차단 등 URL 입력 모니터링 및 차단
• RFC Compliant Url - 만약 URL이 RFC를 따르면 체크해라. 만약 RFC를 따르지 않는다면 차단될 것이다.
• RFC Compliant HTTP Url - 만약 HTTP URL이 RFC를 따르면 체크해라. 이것은 HTTP URL(오로지 절대 경로)에 인증을 차단할 것이다.
• Use Url Raw Scan - 웹서버가 URL을 디코드하기 전에 URL을 스캔하기 위해서는 기본적인 스캐닝이나 로우 스캐닝을 사용한다.
• Deny Url Encoding Exploits - URL에서 인코딩 공격을 허락하지 않는다.
• Deny Url Parent Path - 요청된 URL에서 상위 경로('..')로 이동하려는 시도를 차단한다.
• Deny Url Trailing Dot In Dir - URL의 디렉토리 이름 안에 './' 가 포함된 모든 요청을 차단한다.
• Deny Url Backslash - URL에 '₩' 가 포함되면 차단한다.
• Deny Url Alternate Stream - URL에 ';' 가 포함된 모든 요청을 차단한다.
• Deny Url Escaping - 디코딩을 한 후 URL에 '%' 가 포함되면 허락하지 않는다.
• Deny Url Running Multiple CGI - URL에 '&' 가 포함되면 허락하지 않는다. 이것은 다중 CGI 응용 프로그램에서 실행될 수 있다.
>> Deny Url Characters - 입력한 문자들을 차단한다. 만약 URL에 입력한 문자들이 포함되면 요청을 차단한다.
• Deny Url HighBitShellcode - 하이 비트 쉘 코드(ascii > 127)를 허락하지 않는다. 이것은 웹 사이트에서 US-ASCII만을 사용하도록 제한하고 US-ASCII에 속하지 않는 문자들을 차단한다. 영문 웹 사이트가 아닌 곳은 추천하지 않는다. 이것은 또한 URL에서 Unicode/UTF-8 그리고 MBCS를 차단한다.
• Use Denied Url Sequences - 'URL Denied Sequences'에 명시된 시퀀스들이 URL에 하나 이상 포함되면 요청을 차단한다.
• Use Allowed Url Starts - 설정된 URL로 시작하는 요청만 허가한다. 이것은 'URL Allowed Starts'에 명시된다.
>> URL Denied Sequences - 이것은 URL에 허가되지 않은 시퀀스 목록이다.
>> URL Allowed starts - 이것은 URL이 시작할 수 있는 허가된 시퀀스 문자 목록이다.

○ **Mapped Path** : 경로에 상위 패스, 백슬래쉬(₩) 등 차단 및 로컬 파일시스템의 허용하는 경로 정의
• Deny Parent Path - 경로에 상위 경로('..')로 이동하려는 시도를 차단한다.
• Deny Backspace - 경로에 '₩' 가 포함되면 차단한다.

• Deny Carriage Return - 경로에 carriage return 문자가 포함되면 차단한다.
• Deny New Line - 경로에 new line 문자가 포함되면 차단한다.
• Deny Escaping - 경로에 '%'가 포함되면 차단한다.
• Deny Dot In Path - 경로에 ./'가 포함되면 차단한다.
>> Deny Characters - 만약 경로에 이 문자들이 하나 이상 나타나면 요청을 차단한다.
• Use Allowed Paths - 오로지 'Allowed Paths'에 명시된 경로로 시작되는 경로만을 허가한다.
>> Allowed Paths - 이것은 시작할 수 있는 경로 목록이다.

○ **Requested File** : 차단시킬 파일 목록과 차단 · 허용할 파일 확장자 정의
• Use Filename Raw Scan - 웹서버가 URL을 디코드하기 전에 기본적인 스캐닝 또는 로우 스캐닝을 이용하여 요청된 파일을 스캔할 수 있다.
>> Deny Filename Characters - 만약 파일명이 이 문자중 하나를 포함할 경우 요청을 차단한다.
• Deny Default Document - 기본적인 문서 요청을 차단한다. 클라이언트는 디렉토리가 아닌 오직 특정 파일만을 요청할 수 있다.
• Use Denied Files - 접근이 가능하거나 사용할 수 있는 'Denied files'에 명시된 파일명/CGI 응용 프로그램은 차단된다.
>> Denied Files - 이것은 허가되지 않은 파일명/CGI 응용 프로그램 목록이다.
• Use Monitored Files - 'Monitored Files'에 명시된 파일의 접근을 모니터링한다.
>> Monitored Files - 이것은 'Use Monitered Files'에서 사용되는 모니터링 될 파일명 목록이다.
• Use Allowed Extensions(확장자) - 'Allowed Extensions'에 명시된 파일 확장자만을 요청 허가한다.
>> Allowed Extensions - 이것은 허가된 요청 파일의 확장자 목록이다.
• Use Denied Extensions - 'Denied Extensions'에 명시된 파일 확장자 요청을 차단한다.
>> Denied Extensions - 이것은 차단된 요청 파일의 확장자 목록이다.

○ **Robots** : 검색 엔진의 robots 차단 설정
• Allow Bots Robots File - 서버에 robots.txt파일이 있을 경우 파일의 룰에 맞춰 정보 수집을 허락하고 파일이 없을 경우에도 정보를 수집하는 robot이 파일에 접근했다고 생각하게 만든다.
• Deny Bots All - 모든 봇의 요청을 차단한다.
• Deny Bots Bad - 악의적인 봇의 요청을 차단한다. 이것은 robots.txt 파일에 봇 URL 트랩을 추가한다.(이 설치에서 robots.txt 샘플을 찾을 수 있다.)
>> Deny Bots BotTraps - robots.txt에 봇 URL 트랩을 추가한다.
• Use Deny Bots Aggressive - 봇이 일정 시간 안에 설정된 수 이상의 요청을 하는 경우 차단한다.
>> Deny Bots Aggressive Max Count - 봇을 차단할 요청의 수
>> Deny Bots Aggressive Max Time - 입력된 시간 내에 요청 수를 카운트한다.
>> Deny Bots Timeout - 봇을 차단할 타임아웃.
• Block Bots Data Mining Commercial - 상업적인 데이터마이닝 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.
• Block Bots Data mining Public - 비 상업적이거나 공공의 데이터마이닝 봇을 차단한다.

Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Download Managers - 다운로드 매니저를 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Email Harvesting - email을 수집하는 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Guestbook Spammers - 방명록 스팸 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Hack Tools - 해킹 도구를 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Image Downloaders - 이미지 다운로더 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Indexing - 색인 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots monitoring - 모니터링 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Offline Browsers - 오프라인 브라우저 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Other Bad - 다른 악의적인 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Trademark - 저작권/트레이드마크 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Validation Tools - 인증 도구 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Link Checking - URL 검사 유틸리티 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Browsers - 브라우저 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Media Players - 미디어 플레이어 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Proxies - 프록시 서버 를 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Adware - 애드웨어 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Browser Extensions - 브라우저 확장자 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Spyware - 스파이웨어 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Editing - web/html을 편집하는 소프트웨어 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Device - 장치 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots News Feed - news feed 유틸리티 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Search Engines - 검색엔진 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

• Block Bots Filtering Software - 소프트웨어를 필터링하는 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.
• Blcok Bots Software Component - 소프트웨어 컴포넌트 봇을 차단한다. Robots.xml에서 정의된 잘 알려진 사용자 에이전스와 IP주소를 찾아서 이를 수행한다.

○ **Headers** : 서버 헤더 정보 변경, 특정 헤더 차단 등 설정
• Remove Server Header - 클라이언트로 보내는 웹서버로부터의 모든 응답에 포함된 'Server:' 헤더를 제거한다. 서버 헤더는 해커나 웜으로부터 시스템의 취약점을 탐색하거나 서버가 위험에 빠질 수 있는 민감한 정보가 있다.
• Change Server Header - 서버의 헤더를 제거하는 대신 그것을 교체한다. 이 방법은 서버 헤더에 다른 상업적인 웹서버 정보를 입력함으로써 해커나 웜을 바보로 만들 수 있다. Note: 'Remove Server Header'가 이것보다 우선권을 갖는다. 만일 서버 헤더를 바꾸길 원한다면 'Remove Server Header'를 활성화해서는 안된다.
>> Server Header - 서버 헤더가 클라이언트로 보내는 모든 응답이다. 이것은 소프트웨어/버전의 형식을 갖는다.
• Deny Cookie SQL Injection - 'Cookie:' 헤더의 SQL 인젝션을 차단한다. 만일 웹사이트가 DB를 사용하고 DB에 관계된 정보를 저장하기위해 쿠키를 사용한다면 유용할 것이다.
• Deny Cookie Encoding Exploit - 'Cookie:' 헤더의 쿠키 공격을 허가하지 않는다.
• RFC Compliant Host Header - 만일 HTTP 1.1. 요청이 'Host:' 헤더를 포함하지 않으면 차단한다.
• Use Denied Headers - 만약 'Denied Headers'에 기입된 헤더중의 하나라면 차단한다.
>> Denied Headers - 웹서버로의 요청에 허가되지 않는 헤더들을 기입.
• Use Allowed Content Types - 요청에 Content-Type 헤더의 체크가 활성화되고 만약 Content-Type가 'Allowed Content Types' 리스트에 있지 않다면 요청을 차단한다.
>> Allowed Content Types - 요청의 허가된 Content-Types 이다. 만약 예를 들어 당신이 모든 Multipart 타입이 가능하길 원한다면 간단하게 'multipart/'를 추가한다. 이 방법은 'multipart/form-data', 'multipart/mixed',...을 가능하게 하는데 효과적이다. 빈 줄은 Content-Type이 없는 것 또한 허가된다는 의미이다.
• Deny Header SQL Injection - 웹서버로 보내진 헤더에 SQL 인젝션을 허가하지 않는다.
• Deny Header Encoding Exploits - 웹서버로 보내진 헤더에 인코딩 공격을 허가하지 않는다.
• Deny Header Directory Traversal - 웹서버로 보내진 헤더에 상위 경로로의 이동을 허가하지 않는다. '..' 이나 '/', 'W'를 차단한다.
• Deny Header High Bit Shellcode - 하이 비트 쉘코드를 차단한다. 이것은 웹 사이트에서 US-ASCII만을 사용하도록 제한하고 US-ASCII에 속하지 않는 문자들을 차단한다. 영문 웹 사이트가 아닌 곳은 추천하지 않는다.
• Use Denied Header Sequences - 만약 'Denied Header Sequences'에 기입된 시퀀스 문자중 어느 것이라도 헤더에서 나타나면 요청을 차단한다.
>> Denied Header Sequences - 헤더에서 허가되지 않은 시퀀스 문자 목록이다.

○ **Referrer** : Referrer 헤더 스캔, 바로 연결하거나 다운로드 가능한 파일 확장자 제한
• Use Referrer Scanning - URL 참조를 스캔한다. 이것은 이 섹션에서 다른 체크를 허가하는 것을 가능하게 한다.
• Referrer URL RFC Compliant - URL 참조는 RFC를 따른다.
• Referrer URL RFC HTTP Compliant - URL 참조는 (인증이 아닌)HTTP RFC를 따른다.

• Deny Referrer Encoding Exploits - URL 참조에 인코딩 공격을 차단한다.
• Deny Referrer Hot Linking - 임의의 도메인으로부터 임의의 파일을 바로 링크되는 것을 차단한다.
>> Referrer Hot Linking File Extensions - 이 파일 확장자들이 바로 링크되는 것을 차단한다.
• Use Referrer Hot Linking Allow Domains - 특정 도메인만 바로 링크되는 것을 허가한다.
>> Referrer Hot Linking Allow Domains - 바로 링크되는 것이 허가된 도메인이나 IP주소이다. 이 목록에 자신의 도메인을 추가하는 것이 필요하다.
• Use Referrer Hot Linking Deny Domains - 특정 도메인이 바로 링크되는 것을 차단한다.
>> Referrer Hot Linking Deny Domains - 바로 링크되는 것이 차단된 도메인이나 IP주소이다.
• Referrer Hot Linking Use Host Header - 호스트 헤더 도메인이 바로 링크되는 것을 허가한다. 이것은 허가된 목록에 도메인 이름을 추가할 필요 없이 그것을 참조하기 위해 로컬 웹사이트를 허가 한다.
• Referrer Hot Linking Deny Blank Referrer - 보호받아야 할 파일 확장자에 대해 참조가 없는 요청을 차단한다.
>> Deny Referrer Characters - URL 참조에 폼에 입력된 문자들을 차단한다.
• Den Referrer High Bit Shellcode - 참조 URL에 하이 비트 쉘코드를 차단한다. URL 참조에 ASCII>127인 문자를 차단하고 당신의 사이트에 링크된 US-ASCII 웹사이트가 아닌 사이트를 차단한다.
• Use Deny Referrer Sequences - URL 참조에서 특정 시퀀스 문자를 차단한다.
>> Deny Referrer Sequences - URL 참조에서 폼에 입력된 시퀀스 문자를 차단한다.

○ **User Agent** : User Agent 헤더를 스캔
• Deny User Agent Empty - 만약 User Agent가 비어있거나 나타나지 않으면 요청을 차단한다.
• Deny User Agent Non RFC - 만약 User Agent가 RFC를 따르지 않으면 요청을 차단한다.
>> Require User Agent Character - 만약 User Agent가 이 문자들 중 적어도 하나를 포함하지 않으면 요청을 차단한다.
• Use Denied user Agents - 만약 User Agent가 'Denied User Agents' 목록에 있으면 요청을 차단한다.
>> Denied User Agents - 이것은 요청이 거부된 User Agents 목록이다.
• Use Denied User Agent Sequences - 만약 User Agent가 'Denied user Agent Sequences'에 입력된 시퀀스 문자를 포함하면 요청이 차단된다.
>> Denied User Agent Sequences - 이것은 거부된 User Agents 시퀀스 목록이다.

○ **Methods** : 허용 또는 차단할 Method를 결정(예 : GET, HEAD, POST은 허용하고 DELETE, PUT 등은 차단)
• Use Allowed Verbs - 오직 'Allowed Verbs'에 입력된 요청 메소드 만을 허가한다.
>> Allowed Verbs - 허가된 요청 메소드.
• Use Denied Verbs - 만약 요청 메소드가 'Denied Verbs'에 입력된 것들 중 하나이면 요청은 차단된다.
>> Denied Vervs - 차단된 요청 메소드.

○ **Querystring** : 특정 query 스트링(xp_cmdshell, cmd.exe 등) 차단, query 스트링에서 SQL Injection 차단 등 설정

• Use Querystring Raw Scan - 디폴트 스캐닝이나 로우 스캐닝은 웹서버에서 URL을 디코드하기 전에 querystring을 스캔할 수 있다.

• Deny Querystring SQL Injection - querystring에 SQL 인젝션을 허가하지 않는다.

• Deny Querystring Encoding Exploits - querystring에 인코딩 공격을 허가하지 않는다.

• Deny Querystring Directory Traversal - querystring에서 디렉토리 이동을 허가하지 않는다. 이것은 '..' 또는 '/' , 'W'를 차단할 것이다.

• Deny Querystring High Bit Shellcode - 하이 비트 쉘코드를 허가하지 않는다. 이것은 웹 사이트를 US-ASCII만으로 제한하고 이 문자가 아닌 문자들을 차단한다. 영문 웹사이트가 아닌 곳은 추천하지 않는다.

• Use Denied Qerystring Sequences - 만약 'Denied Querstring Sequences'에 기입된 것 중 하나이면 차단한다.

>> Denied Querystring Sequences - 허가되지 않는 querystring 문자 목록이다.

○ **Global Filter Capabilities** : 글로벌 필터 적용 여부, 특정 헤더 스트링(xp_cmdshell, cmd.exe 등) 차단 등 결정

• Is Installed As Global Filter - 이것은 글로벌 필터 기능을 위해 요구된다. 이 이벤트는 오로지 필터가 글로벌 필터로서 인스톨될 때 요청될 수 있다. 만약 WebKnight를 설치할 때 그 케이스가 아닐 경우 로드하는 것은 실패할 것이다. IIS 재시작 필요.

• Deny Header SQL Injection - 웹서버에 보내진 헤더 안에 SQL 인젝션을 허가하지 않는다.

• Deny Header Encoding Exploits - 웹서버로 보내진 헤더 안에 인코딩 공격을 허가하지 않는다.

• Deny Header Directory Traversal - 웹서버에 보내진 헤더 안에 디렉토리 이동을 허가하지 않는다. 이것은 '..'나 '/' 또는 'W'를 차단한다.

• Deny Header High Bit Shellcode - 하이 비트 쉘코드를 허가하지 않는다. 이것은 웹 사이트가 US-ASCII만을 사용하도록 제한하고 그것이 아닌 문자들을 차단한다. 영문 웹 사이트가 아닌 곳은 추천하지 않는다.

• Use Denied Header Sequences - 헤더 안이 'Denied Header Sequences'에 기입되지 않은 문자 중 하나이면 요청을 차단한다.

>> Deny Postdata SQL Injection - 웹서버로 보내진 POST 데이터 안에 SQL 인젝션을 허가하지 않는다.

○ **SQL Injection** : SQL Injection 공격에 이용되는 키워드 정의(' ; select insert xp_ 등)

>> SQL Injection Keywords - SQL 인젝션 스캐닝을 위한 SQL 키워드들. 만약 둘 또는 그 이상이 발견되면 경고가 나타나고 요청이 차단된다.

○ **Web Applications** : WebDAV, IISADMPWD 등 웹 애플리케이션의 허용유무 결정

• Allow File Uploads - 서버에서 HTTP POST 명령을 사용하는 파일 업로드를 허가한다.

• Allow Unicode - URL과 서버로 보내진 다른 데이터에서 유니코드를 인코딩하는 것을 허가한다.

• 그 밖에 WebDAV, IISADMPWD 등 웹 애플리케이션의 허용 유무를 결정하고 그 밖에 기타 사항이 기술되어 있다.

4. 모의 공격 및 확인

4.1 SQL Injection

SQL Injection은 악의적인 SQL 명령어를 인자로 삽입함으로써, 사용자가 웹 애플리케이션에 악의적인 Query를 DB로 전달하도록 조작할 수 있다. 이는 자주 발견되는 문제이고 매우 위험한 공격 방법 중에 하나이다. 이 공격을 통하여 악의적인 사용자는 DB에 저장된 정보를 획득, 조작, 파괴할 수 있다. 결국 해당 취약점을 통해 단순 공격에서부터 시스템을 완전히 장악하거나 파괴하는 결과에 이르기까지 다양한 영향을 끼칠 수 있다.

이제 WebKnight의 SQL Injection을 차단하는 룰을 설정하여 공격이 차단됨을 보이겠다.

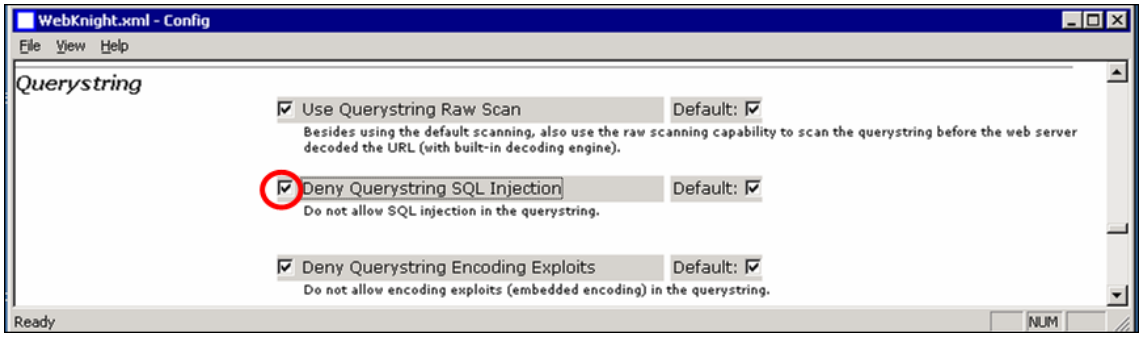

그림 16 SQL Injection 공격 차단 설정

그림 16과 같이 'Deny Querystring SQL Injection'을 체크하여 룰을 설정하였다.

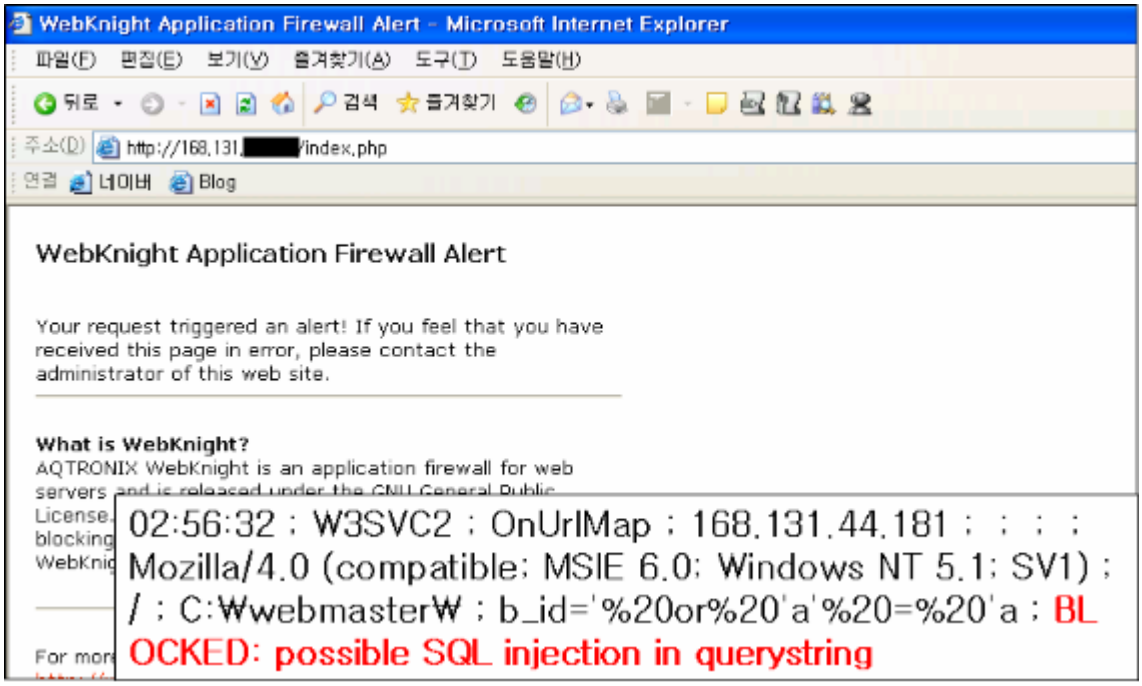

그림 17 SQL Injection 공격 차단 확인

그림 17을 보면 공격이 차단되어 WebKnight에서 설정한 페이지가 화면에 나타나고 공격 차단 로그가 로그파일에 남아있는 것을 확인할 수 있다.

4.2 Directory Traversal

Directory Traversal 취약점은 자료실에 올라간 파일이나 웹에서 파일을 다룰 때 파일명을 적절하게 체크하지 않아 공격자가 '../' 와 같은 파일명 앞에 상위 디렉토리로 이동하는 문자를 입력해 '../../../../../etc/passwd' 와 같이 시스템의 중요한 파일을 다운로드할 수 있는 취약점이다.

이제 WebKnight를 이용하여 Directory Traversal을 차단하는 룰을 설정하여 공격이 차단됨을 확인하겠다.

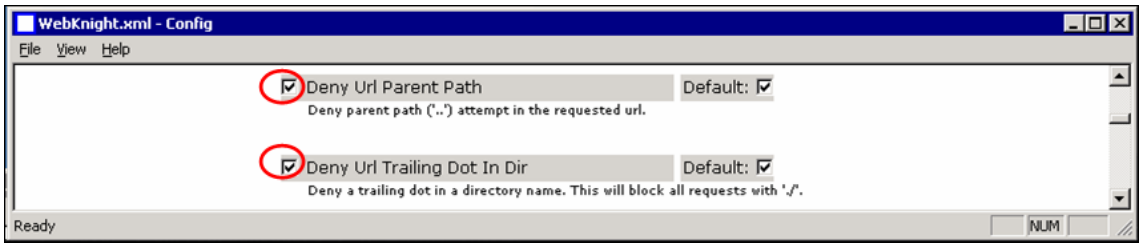

그림 18 Directory Traversal 공격 차단 설정

그림 18과 같이 'Deny URL Parent Path' 와 'Deny URL Trailing Dot In Dir' 을 체크하여 룰을 설정하였다.

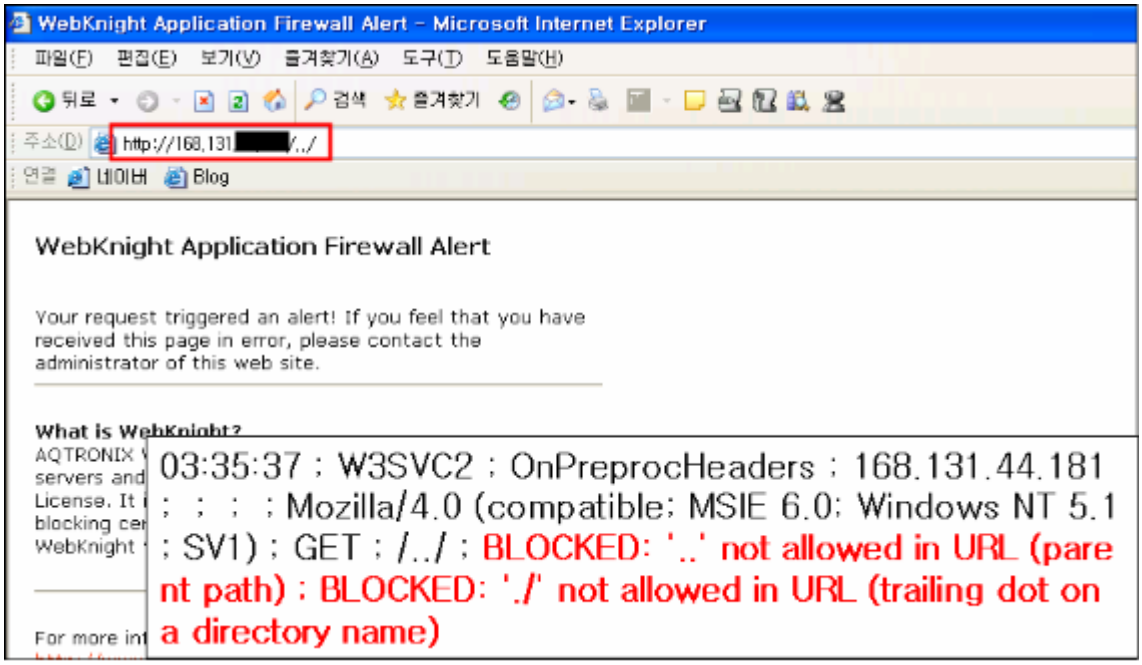

그림 19 Directory Traversal 공격 차단 확인

그림 19를 보면 공격이 차단되어 WebKnight에서 설정한 페이지가 화면에 나타나고 공격 차단 로그가 로그파일에 남아있는 것을 확인할 수 있다.

5. 참고 문서

WebKnight를 이용한 SQL Injection 공격 차단 - [자료: 한국정보보호진흥원(KISA)]