File List를 이용한 Web Scan 공격탐지

김진묵* · 백영호*· 유황빈*
*광운대학교 컴퓨터과학과

Web Scan Attack Detection based on File List

Jin-mook Kim* · yung-ho Back · Hwang-bin Ryou*
*Department of Computer Science, Kwangwoon University
E-mail : jmkim@netlab.kw.ac.kr, yhback@netlab.kw.ac.kr, ryou@kw.ac.kr

요 약

웹 서비스는 매우 가깝고 편리한 인터넷 서비스중의 하나이다. 이러한 웹 서비스의 이용이 증가함으로써 웹을 이용한 공격과 취약점 분석으로 위한 위험성이 급격하게 증가하고 있다. 이는 웹 서비스가 가지고 있는 공개성 때문이다.

이에 본 논문에는 웹 공격과정에서 필요한 정보를 얻거나 어플리케이션의 취약점을 찾는 웹 스캔 공격을 탐지하기 위한 방법으로 웹 서버의 파일 리스트를 이용하는 방법을 제안고자 한다. 시스템의 설계와 구현을 위해 사용한 감사 데이터는Snort에서 일차적으로 탐지된 것을 제외한 웹 서버의 접근 로그를 사용한다. 생성된 감사 데이터와 파일 리스트를 비교하여 사용자 요청의 존재여부로 공격을 탐지하도록 설계하였다. 이와 같은 방법은 제안시스템의 실험을 통하여 웹 스캔 공격의 탐지에 효과가 있는 것으로 밝혀졌다.

키워드

침입 탐지, 해킹, 스캔 공격, 웹 서비스

I. 서 론

인터넷은 현대인의 생활과 매우 밀접한 하나의 구성 요소로 자리잡고 있다. 사람들은 인터넷을 통하여 생활 정보를 교환하고 학자들은 정보를 수집하여 그들의 연구 대상 혹은 도구로써 인터넷을 사용하고 있다. 이는 브라우저 기반의 어플리케이션이 구현, 지원, 유지 보수가 쉽기 때문이다. 하지만 이러한 현실은 보안측면에서 볼 때 매우 심각한 문제점을 드러내고 있다.

기존 네트워크 오용에 대한 대응책으로 여러 보 안 메커니즘이 제안되었고, 방화벽과 같은 접근통제 솔루션과 공격 시그니처 기반의 침입 탐지시스템[1]을 활용하여 외부로부터의 접근을 통제함으로써 침입에 대한 피해는 막을 수 있게 되었다.

하지만, 최근 해킹 동향을 보면 웹 공격도구의 다양화로 해킹의 용이성이 높고, 이에 따라 O/S 기반의 공격보다는 다양한 형태의 취약점이 존재하는 웹을 대상으로 하는 공격이 주를 이루고 있다.[8] 웹이 주요 공격대상이 되는 이유는 크게 3가지로 분류할 수 있다.[4]

첫째, 구조가 복잡하기 때문이다. 웹 어플리케이션의 구조는 그림 1 과 같이 Web Client(Browser), Web Interface, Web Server, Web Application (CGI, PHP, Perl, C/C++, Shell Script), JSP, ASP), Database와 같은 요소들로 계층적으로 이루어지므로, 공격이 발생하는 지점과 각 위치에서의 취약 요소들이 상이하다. 따라서 웹 기반의 공격에 대한 탐지기법은 보호시스템이 제공하고 있는 웹 서비스의 특징과 어플리케이션에 많이 좌우되고, 이에 제대로 대응할 수 있는 방법이 존재해야 하므로 기존의 시그니처 기반의 패턴 매칭을 주로 하는 침입 탐지기법은 효과적으로 대응하기 어렵다.

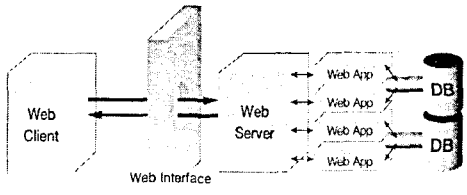

그림 1. 웹 어플리케이션의 구조

둘째, 웹 서비스 자체가 공개적이라는 특징이다. 웹 서비스는 누구나 외부에서 사용할 수 있도록 제공되어야 하므로 접근통제의 메커니즘을 적용할 수 없거나, 제한적으로 사용할 수밖에 없다. 따라서 그 효과를 제대로 발휘할 수 없고, 최근에는 이와 같은 서비스의 취약점을 이용한 비정상적이거나 악의적인 공격 행위들이 많이 발생하고 있다.

마지막으로, 어플리케이션의 통합 가속화다. 웹은 기본적으로 다른 연동 어플리케이션과의 상호작용을 할 수 있는 통로역할을 하고 클라이언트로부터의 요청을 어플리케이션 서버 프로그램에 전달하고 수행결과를 다시 돌려주는 매개체 역할을 하고 있다. 따라서, 각 어플리케이션의 취약점들을 이용 한 많은 공격형태가 나타나고 있다.

오늘날 어플리케이션에서 가장 위험한 보안상의 결함은 웜이나 바이러스, 알려져 있는 어플리케이션 서버의 취약성도 아닌 어플리케이션 자체의 취약성 이다. 이에 따라 "개방형 웹 어플리케이션 보안 프로젝트(OWASP : Open Web Application Security Project)"라는 이름의 그룹을 결성하고, 취약성을 종합해"10대 어플리케이션 취약성"을 발표했다.

본 논문에서는 웹 공격을 감행하기 위한 사전 행동인 Web Scan Attack을 Web Server의 Root Directory의 File List를 이용해 Web Scan Attack 탐지기법을 제안한다.

본 논문의 구성은 다음과 같다. 2 장에서는 기존보안 솔루션의 간단한 설명과 웹 공격 시 기존 보안솔루션의 문제점, 그리고 관련연구에 대해 기술한다. 3 장은 본 논문에서 제안하고자 하는 시스템에 대해 기술한다. 마지막으로 4 장에서는 본 논문에서 제안한 시스템의 실험에 대해 기술하고 마지막으로 5장에서는 결론에 대해 기술하는 것으로 본 논문을 맺는다.

II. 관련연구

2.1 기존의 보안 솔루션

전통적인 시스템 해킹은 침입 차단시스템과 침입 탐지시스템(IDS)을 이용해 충분히 방어가 가능하다. 그러나 비즈니스 환경의 변화로 고전적인 방어수단은 그 빛을 잃고 있다.

아직까지 침입 탐지기법 중 웹 서비스에 초점을 둔 연구는 많지 않았고, 단지 일반적인 침입 탐지기법에 웹 공격에 관련된 시그니처를 사용하였다. 대표적인 네트워크 기반 IDS인 SNORT[2]는 웹 공격에 대한 1000 여 개의 시그니처를 가지고 있다. 그러나 시그니처는 약간의 변형이라도 가해질 경우에 이를 탐지하지 못한다. 이와 반대로 침입 탐지시스템이 잘못된 경고를 나타내도록 알려진 시그니처대로 패킷을 생성하여 전송할 수도 있다.[3] 이처럼 시스템이 잘못된 경고 메시지를 과다하게 생성함으로써 시스템이 무력화될 수도 있고 이 과정에서 실제 공격 행위들이 감춰질 수도 있을 것이다.

또 다른 보안 솔루션인 침입 차단시스템은 외부로부터의 불법적인 접근이나 해킹의 공격으로부터 내부의 네트워크나 시스템을 방어해 주는 보안 솔루션이다. 그러나 오늘날의 비즈니스 환경은 정상행위와 공격행위의 구분이 힘들어지고 있다. 또한 웹을 중심으로 IT 환경이 통합 되면서 침입 차단시스템이 웹 서비스 자체를 차단할 수가 없다. 따라서 침입 차단시스템만으로는 웹 해킹에 대한 적절한 대안이 되지 못한다.

2.2 10대 어플리케이션 취약성

OWASP은 각 기관들이 웹 애플리케이션 및 웹 서비스의 보안을 이해하고 향상시키는 것을 돕기 위한 단체이다. 이 단체에서는 해커들이 웹 어플리케이션 개발과정에서 발생하는 전형적인 취약점들에 관심을 갖고 있음에 주목하여 표 1 과 같이 웹 어플리케이션의 취약점을 발표하였다.[6]

2.3 SAD(Session Anomaly Detection)

SAD는 웹 페이지를 방문하는 사용자들이 요청하는 페이지들 간에는 일정한 순서가 있다고 가정 한다. 이 연구에서는 접근통제가 상대적으로 미약한 웹 서비스의 보호를 목표로, 기존의 침입 탐지 기법들이 탐지하기 어려운 변형 혹은 새로운 형태의 공격에 대응할 수 있는 웹 사용자 세션 기반의 이상 탐지기법을 제안하였다.

그림 2 는 SAD의 구조를 나타내고 있다. 웹 서버의 접근로그(access log)는 사용자의 정상행위 프로파일과 감사 데이터로 사용된다. 정상행위 프로파일은 사용자의 세션을 분리하고 세션에서 접근하는 일련의 요청 페이지 순서를 프로파일로 완성한다. 비정상 행위 탐지는 현재의 웹 페이지 접근 순서가 과거 사이트의 접근 순서목록과 비교해 볼 때 확률적으로 얼마나 유사한지를 이상점수(anomaly score)로 정량화하여 비정상적인 요청을 탐지하는 베이지언 추정기법을 이용하여 비정상행위를 탐지한다.[11]

표 1. 웹 어플리케이션의 10대 취약점

취 약 점 설 명
A1. 입력 값 검증 부재
Browser에서 보내는 요청에 대한 검사를 진행하지 않아서 생기는 문제점
A2. 취약한 접근 통제
적절한 접근제어가 이루어 지지 않는 문제
A3. 취약한 인증 및 세션 관리
계정 인증서와 웹 통신 시 인증을 위한 세션 토큰이 적절하게 보호 받지 않는 문제
A4. 크로스 사이트 스크립팅 취약점
웹 어플리케이션의 특성상 Browser에서 실행될 수 있는 악성코드 전달
A5. 버퍼 오버플로우
Web application component가 제대로 처리되지 않아 정상적인 제어를 못하고 원치 않는 작업을 실행하게 하는 방법
A6. 삽입 취약점
Parameter값에 악의적인 값을 넣게 되면 원치 않는 결과가 나타남
A7. 부 적절한 에러처리
공격자가 웹 애플리케이션이 처리하지 못하는 에러가 발생하도록 유도
A8. 취약한 정보 저장 방식
암호화를 사용함에 있어 적절하게 사용하지 못하여 생기는 문제
A9. 서비스 방해 공격
웹 애플리케이션의 자원 고갈
A10. 부적절한 환경 설정
기존에 알려진 프로그램들을 사용할 때 기본 설정을 그대로 사용하는 경우 발생하는 문제점

III. 제안시스템

본 논문에서 제안하는 기법은 "공격자가 웹 서버를 공격하기 위해 웹 서버의 정보나 웹 어플리케이션의 취약점을 Scan한다." 는 것을 가정하고 있다. 공격자는 기존 보안솔루션을 우회할 수 있는 자동화된 스캐닝 도구를 사용해 목표 웹 서버의 어플리케이션에 대한 취약점을 발견하고 이를 이용하여 목표 시스템을 점령한다. 공격자는 다음

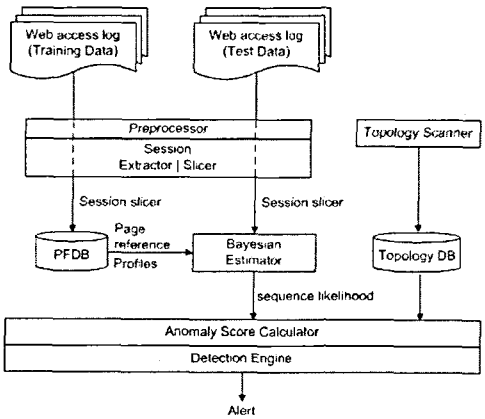

그림 2. SAD 구조

공격 단계로 목표 시스템을 통해 제 2 의 공격(FTP Server, Mail Server, Database)을 감행한다.

본 논문에서는 웹 해킹을 위한 전제조건 인 Web Scan Attack을 탐지하는 방법으로 웹 서버 Root Directory의 File List를 이용해 탐지하는 방법을 제안한다. 제안하는 연구 방법은 웹 서버의 Root Directory에 존재하는 파일목록들을 리스트화한다. 그리고 웹 서버의 접근기록(access log)으로부터 감사 데이터를 생성하여 File List와 비교해 존재하지 않는 페이지를 요청했을 경우 이를 공격으로 간주한다.

제안 시스템의 구조를 살펴보면 그림 3 과 같이 2개의 세부 모듈인 전 처리 모듈과 탐지 모듈로 구성되어 있다.

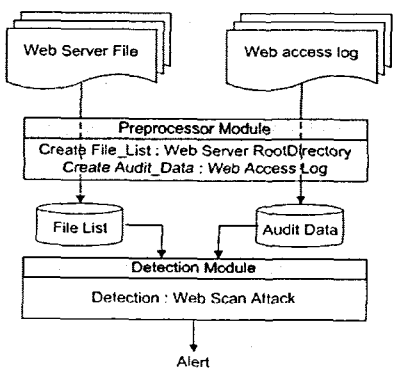

그림 3. 제안시스템 전체구성도

3.1 전 처리 모듈

전 처리 모듈에서는 File List와 Audit Data를 생성한다. 이때 웹 서버의 Root Directory와 Access Log에는 그림, 동영상 파일 등의 멀티미디어 데이터가 포함되어 있다. 하지만 본 논문에서는 멀티미디어 데이터가 논문의 목적과 개연성

이 없기 때문에 이를 제외한다.

그림 4 와 그림 5 는 File List와 Audit Data를 생성하는 구조에 대해 나타내고 있다. File List생성 모듈은 웹 서버의 Root Directory에 존재하는 파일을 리스트로 만드는 역할을 한다.

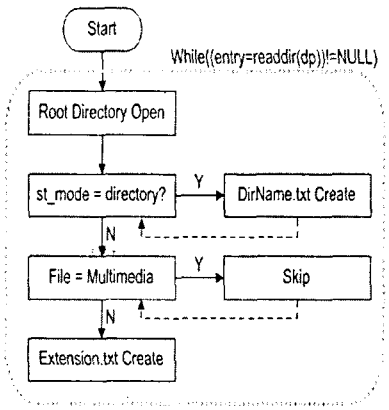

그림 4. File List 생성모듈 구조

Audit Data는 웹 서버의 접근 로그를 이용하여 생성한다. 접근 로그는 아파치 웹 서버의 combined 형식으로 남는다.

이 형식은 Host IP, Request Time, Method, Request Page, Protocol, User Agent Information 과 같은 많은 엘리먼트들로 구성된다. 하지만 본 논문에서는 감사 데이터의 엘리먼트로 Host IP, Request Time, Request Page 만을 이용하고자 한다.

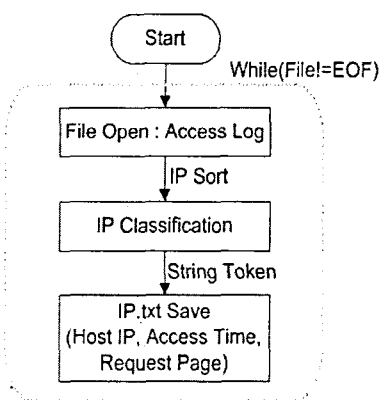

그림 5. Audit Data 생성모듈 구조

3.2 탐지모듈

탐지모듈에서는 File List와 Audit Data를 비교하여Web Scan Attack을 탐지하는데 두 번의 감사를 수행한다.

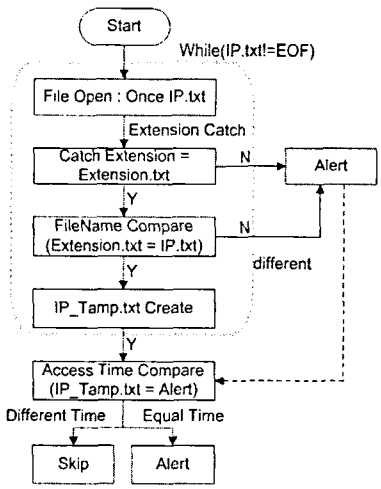

그림 6. 탐지모듈 구조

첫 번째 감사는 사용자의 요청이 존재하지 않을 때에 대한 감사이고, 두 번째 감사는 사용자의 요청이 존재할 때에 대한 감사이다.

한 개의 Audit Data에서 웹 서버 사용자의 요청이 File List와 비교 했을 시 File List에 존재하지 않는 페이지를 요청했을 경우 공격으로 간주한다. 하지만 존재하는 페이지 요청일 경우에는 감사를 다시 한번 수행하기 위해 임시 파일을 생성한다. 생성한 임시 파일의 Host IP, Access Time과 공격으로 간주된 Host IP, Access Time를 비교하여 동일 IP에서 동일 시간대에 요청이 발생한 경우를 공격으로 간주한다.

IV. 시스템 구현 및 실험결과

4.1 시스템 구현 및 실험 환경

제안하고자 하는 시스템은 Linux Kernel ver 2.4.20.8 smp RedHat 9.0, gcc V3.2.2의 컴파일러와 Apache 2.0.52 웹 서버를 사용한다. 그리고 실험 환경은 구현 환경에Snort 2.0.2, ACID(Anomaly Consol for Intursion Detection) 0.9.6b23, PHP 4.3.3, MySQL 5.0.0-alpha, Whisker 1.4, Arirang 1.6 을 사용한다.

4.2 실험 데이터

공격 데이터는 DARPA 데이터와 같은 정형화된 공격 데이터를 생성하기가 쉽지 않기 때문에 기존의 웹 침입탐지시스템에 관한 연구에서와 유사하게 운용중인 웹 서버와 임의의 공격데이터를 생성하여 사용하였다.

가) 웹 서버

웹 서버는 현재 매우 다양한 종류가 존재하고 있으나 세계적으로 70% 의 이용률을 나타내는

Apache web server를 사용한다. 그리고 웹 서버 접근 로그는 서버 운용할 때 기록되는 로그를 기반으로 하였다.

나) 공격 데이터

공격 데이터는 웹 스캔 도구인 Whisker 1.4, Arirang 1.6을 사용하였다.

4.3 실험 결과 및 분석

실험은 Apache 웹 서버에 대해 웹 스캔 공격 도구를 이용하여 공격을 수행하고 탐지 여부를 확인하였다. 탐지는 snort의 탐지와 제안 시스템의 탐지를 비교하였다. 실험 데이터는 웹 서버가 보유하고 있는 2731개의 파일에서 멀티미디어 파일을 제외한 13개의 확장자 별 파일 리스트를 사용하였고, 47개의 IP로부터 요청된 감사 데이터를 사용한다. 웹 스캔 공격 도구인 Whisker와 Arirang을 사용하여 웹 서버를 공격할 때 snort가 탐지하는 비율과 제안 시스템이 탐지하는 비율을 비교한다.

4.3.1 전 처리 모듈의 실험 결과

가) 파일 리스트 생성 결과

웹 서버에 존재하는 파일 리스트를 확장자 기준으로 생성한 예제들을 살펴보면 대표적으로 html, htm, php 확장자를 갖는 파일들을 포함하고 있음을 알 수 있다.

나) 감사 데이터 생성 결과

웹 서버가 접근 로그를 요청한 IP별 감사 데이터를 생성한 결과를 아래 그림 7 과 같이 나타내고 있다.

4.3.2 탐지 결과

탐지 결과는 snort에서의 웹 스캔 공격에 대한 탐지율과 제안 시스템에서 탐지율을 비교하였다. 추가적으로 관련 연구인 "SAD : 베이지언 추정을 이용한 웹 서비스 공격 탐지"의 탐지율과도 함께 비교한다.

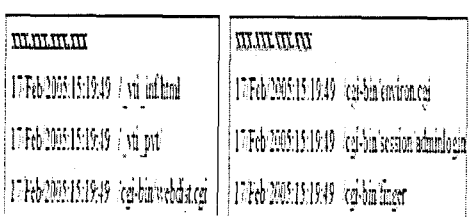

그림 7. 감사 데이터 생성 결과

가) snort 의 탐지 결과

snort는 웹 스캔 공격의 탐지 결과를 DB에 저장하고, 이를 기반으로 ACID에서 웹 문서로 나타내 준다.

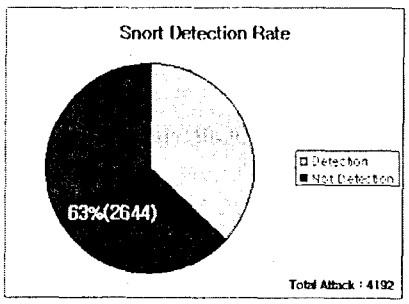

그림 8. snort 탐지율

그림 8 은 웹 스캔 도구를 사용하여 웹 서버를 공격하였을 때 snort에서의 탐지율을 나타내고 있다. 실질적인 웹 스캔 공격 패킷 총 4192개 중 1548(37%)개를 탐지한 결과를 나타내고 있다.

웹 스캔 공격 도구는 기존에 알려진 형태의 패턴과는 다르게 여러 옵션을 이용하여 변형된 형태의 요청을 수행하였다, 그러므로 단순히 요청 URL의 텍스트 비교를 수행하는 침입탐지시스템에서는 이들에 대한 시그니처를 가지고 있지 않기 때문에 탐지율이 낮을 수밖에 없다.

나) 제안 시스템의 탐지 결과

제안 시스템에서는 snort에서 탐지된 것을 제외한 나머지 접근 로그(일반 사용자의 접근, 웹 스캔 공격 포함)를 감사 데이터로 사용하였다. 이에 대한 탐지 결과를 그림 9 과 같이 나타내고 있다.

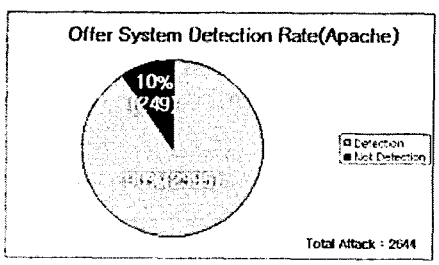

그림 9. 제안시스템의 탐지율

총 접근 로그 수 11625개 중에 snort에서 탐지된 것을 제외하면 10077개의 로그 수를 갖게 된다. 이 중에서 실질적인 웹 스캔 공격은 2644개이고 제안 시스템에서 탐지한 로그 수는 2395(90%)개로 나타났다.

다) 탐지율 비교

제안 시스템의 탐지율과 "SAD : 베이지언 추정을 이용한 웹 서비스 공격 탐지"의 탐지율, 그리고 snort의 탐지율을 비교해 보면, SAD 연구는

웹 서버의 접근 로그로부터 사용자의 세션을 분리하고 이 세션에서 접근하는 일련의 요청 페이지 순서를 프로파일로 완성하게 된다. 추가적으로 특정 순서의 페이지 요청이 있을 때 프로파일과 유사도를 이상점수로 정량화하여 비정상적인 요청을 탐지하는 방법이다.

실험을 위해 약 40개의 사용자 계정을 가진 실험실 웹 서버에게 웹 스캔 공격 도구를 이용하였고, 이에 대한 탐지 여부를 측정하였다.

실험 데이터는 5427개의 IP 주소로부터 13415개의 세션으로 구성된다. 공격 데이터는 웹 스캔 도구인 Whisker를 활용한 4367개의 세션을 사용하였고, snort의 탐지율과 SAD의 탐지율을 비교하였다.

표 2. snort와 SAD의 탐지율

구분	Whisker	탐지율
Snort(1.8.7)	1.4	36%
SAD	1.4	91%

표 2 에서 보는 바와 같이 snort의 탐지율은 36%정도의 탐지를 보이는 반면 SAD는 91%의 탐지율을 나타내고 있다.

표 3. snort와 제안시스템의 탐지율

구분	Whisker	탐지율
Snort(2.0.2)	1.4	37%
제안 시스템	1.4	94%

표 3 은 제안 시스템과 Snort의 탐지율을 비교한 것이다. 제안 시스템의 탐지율은 총 11625개의 접근 로그를 감사하였는데 이 중 실질적인 웹 스캔 공격은 4192개이고, 탐지된 수는 3943(94%)로 나타났다.

표 4. SAD와 제안시스템의 탐지율

구분	Snort		스캔 도구	탐지율
	버전	시그니처		
SAD	1.8.7	516	whisker 1.4	91%
제안 시스템	2.0.2	875	whisker 1.4	94%

표 4는 제안 시스템과 SAD의 탐지율을 비교하여 나타내고 있다. SAD에서는 1.8.7 버전의 snort를 사용하여 웹 공격 관련 시그니처를 516개를 사용하여 실험을 하였고, 스캔 도구는 제안 시스템과 동등한 것을 사용하였다. 제안 시스템에서는 2.0.2버전의 snort를 사용하여 웹 공격 관련 시그니처를 875개를 사용하여 실험 하였다. 이러한 실험을 통해서 탐지율은 각각 SAD는 91%를 제안 시스템에서는 94%의 탐지율을 보인다.

제안 시스템의 탐지율은 웹 스캔 공격 도구를 사용하여 탐지하였다. 그런데 웹 스캔 공격 도구가 사용하는 침입 탐지시스템의 회피 공격법을 탐지할 때 구별하지 못하는 공격이 있음을 발견하였다. 구별하지 못하는 공격에 대해서는 좀더 세밀한 분석이 필요할 것으로 예상된다.

V. 결론

본 논문에서는 웹 해킹을 위한 전제조건인 Web Scan Attack을 탐지하는 방법으로 웹 서버 Root Directory의 File List를 이용해 탐지하는 방법을 제안하였다. 제안한 시스템은 Web Server Root Directory의 파일들을 리스트로 생성하여 Web Server 접근 로그와 비교한다. 요청된 페이지가 존재하지 않을 경우 공격으로 간주하고, 존재 할 경우에는 또 한번의 감사를 수행하여 공격 여부를 판별하도록 하였다.

본 논문에서 제시한 파일 존재 여부를 이용해 침입 여부를 판별하는 방법은 Scan Attack의 로그 내용과 사용자의 정상 요청 내용이 유사한 경우가 발생할 가능성이 있다. 그리고 정상 사용자의 입력오류로 인해 잘못된 요청이 발생할 경우도 있다. 향후 이러한 문제점을 해결하기 위해 노력할 것이며, 정상 사용자의 성향이나 웹 페이지들의 연관성을 분석하여 제안시스템에 접목하면 좀 더 효과적인 결과를 나타낼 것으로 예상된다.

참고문헌

[1] Dorothy E. Denning. An intrusion-detection model. IEEE Transactions on Software Engineering, 13(2): 222~232, February 1987.

[2] M. Roesch. Snort lightweight intrusion detection for networks. In Proceedings of USENIX LISA' 99, 1999.

[3] William Yurcik Samuel Patton and David Dos. An achillesi hell in signature-based ids: Squealing false positives in snort. In RAID 2001, 2001.

[4] J.S. Seo. H.S. Kim. S.H. Cho and S.D. Cha. "Web Server Attack Categorization based on Root Causes and Their Locations."Internation Conference on Information Technology, April. 2004

[5] M. Almgren. H. Debar, and M. Dacier. "A Lightweight Tool for Detecting Web Server Attacks." Proceedings of NDSS 2000.

pp.157~170, Feb. 2000.
[6] OWASP, "OWASP Top Ten Most Critical Web Application Vulnerabilities", http://www.owasp.org,
[7] STG Security, http://stgsecurity.com
[8] CERT/CC, "CERT/CC Overview Incident and Vulnerability Trends : Module 2 Internet Security Overview", http://www.cert.org.
[9] CERT/CC, "CERT/CC Overview Incident and Vulnerability Trends : Module 5 Types of Intruder Attacks", http://www.cert.org.
[10] Snort-The Open Source Network IDS, http://www. snort.org
[11] 조상현. 김한성. 이병희. 차성덕. "SAD : Web Session Anomaly Detection based on Bayesian Estimation", 한국정보보호학회 논문지, 2003.4.
[12] 정혜진. 이명선. "IDSTa : Host based IDS through TCP Stream Analysis", 한국정보과학회 논문지, 2003.
[13] 이병희. 조상현. 차상덕. "Real-Time Visualization of WebUsage Patterns and Anomalous Sessions", 한국정보보호학회 논문지, 2004.