# 룰 기반 웹 IDS 시스템을 위한 효율적인 웹 로그 전처리 기법 설계 및 구현☆

# Design and Implementation of Advanced Web Log Preprocess Algorithm for Rule based Web IDS

이 형 우*
Lee, Hyung-Woo

## 요 약

웹 기반 서비스가 다양한 형태로 제공되면서 웹 서비스 사용자 수는 꾸준히 증가하고 있다. 그러나 웹 서버에 대한 SQL Injection, Parameter Injection 및 DoS 등의 공격 등의 취약점이 발견되고 있다. 이와 같은 형태의 웹 공격에 능동적으로 대응하기 위해 현재 웹 IDS 시스템을 구축하여 룰 기반 대응 시스템을 구축하고 있으나, 웹 서버에서 생성되는 로그 정보에 대한 전처리 과정 없이 룰 기반 IDS 시스템이 구동되기 때문에 효율적인 웹 공격 대응 체계가 구축되지 못하고 있다. 이에 본 연구에서는 웹 로그 정보를 웹 IDS 기반 공격 탐지 시스템의 룰 비교 특성에 적합한 형태로 전처리하는 알고리즘을 제시하고 이를 구현하였다. 제안한 알고리즘은 웹 로그 정보에 대한 필드 단위 파싱 및 중복 문자열 처리 과정을 고속으로 수행하여 대용량의 로그 처리시 성능을 향상시켜 개선된 웹 IDS 시스템 구축이 가능하다.

## Abstract

The number of web service user is increasing steadily as web-based service is offered in various form. But, web service has a vulnerability such as SQL Injection, Parameter Injection and DoS attack. Therefore, it is required for us to develop Web IDS system and additionally to offer Rule-base intrusion detection/response mechanism against those attacks. However, existing Web IDS system didn't correspond properly on recent web attack mechanism because they didn't including suitable pre-processing procedure on huge web log data. Therfore, we propose an efficient web log pre-processing mechanism for enhancing rule based detection and improving the performance of web IDS base attack response system. Proposed algorithm provides both a field unit parsing and a duplicated string elimination procedure on web log data. And it is also possible for us to construct improved web IDS system.

☞ keywords : 웹 IDS, 웹 로그, 전처리 알고리즘, 웹 서비스

## 1. 서 론

국내의 인터넷 이용률은 꾸준히 증가 추세에 있으며, 현재 국내 인터넷 사용자 수는 약 34,570(천명)을 넘고 있다. 그리고 인터넷 사용자를 대상으로 한 조사에서 전체 사용자의 73.2%가 하루 1회 이상 인터넷을 사용하고 있는 추세이다[1].

이처럼 인터넷 사용자의 증가에 따라 국내 주요 포탈 웹 사이트의 하루 웹 로그 양은 50G 내외로 대용량의 로그가 발생하고 있다[2]. 그러나 웹의 양적인 증가와 더불어 웹 공격의 시도 및 성공도 함께 증가하고 있다. 이처럼 웹 서비스가 해커들의 공격 대상이 된 이유는 웹

* 종신회원 : 한신대학교 컴퓨터공학부 부교수
hwlee@hs.ac.kr (제1저자, 교신저자)
[2008/01/28 투고 - 2008/02/20 심사 - 2008/7/10 심사완료]
☆ 본 논문은 한신대학교 학술연구비 지원에 의해 연구되었음

서비스와 웹 어플리케이션의 빠른 증가 추세와 함께 비즈니스 및 많은 사업군이 웹 기반 서비스 방식으로 변화되어 웹 시스템에 대한 의존도가 높아졌기 때문이다. 또한 웹 서비스의 특성상 80과 443번 포트를 통해 외부로부터 유입되는 접속을 허용할 수밖에 없기 때문에 방화벽 시스템에 다른 포트가 차단하였다 하더라고 공격자는 손쉽게 웹 서버에 대한 공격을 수행할 수 있다.

이러한 웹 서비스의 취약점을 보완하기 위해 현재 웹 서비스를 대상으로한 공격 탐지 시스템(Web IDS : Web Intrusion Detection System)[3]이 제시되었다. 기존의 IDS 시스템에서는 공격을 탐지하기 위해 IP 패킷을 대상으로 룰 데이터를 이용해 공격 여부를 탐지한다[4]. 그러나 웹 서버의 공격 탐지를 위해서는 웹 서버에서 생성되는 웹 로그(Web Log) 데이터에 대한 분석을 통해 외부로부터의 불법적인 접속을 탐지하거나 이상 탐지(Anomaly Detection) 기능을 제공해야 한다. 최근 데이터 마이닝 기술을 적용하여 웹 공격에 대한 이상 탐지 및 대응 시스템 구축에 활용하는 방안에 대한 연구도 진행되고 있다[5,6]. 데이터 마이닝 기법에서의 성능 향상을 위해서는 대량의 웹 로그에 대한 효율적 전처리 기법이 제시되어야 한다.

기존의 웹 IDS 시스템은 웹 로그를 기반으로 외부로부터의 공격이나 웹 시스템 내부의 부적절한 쿼리 전송 및 이상 접속 정보를 탐지하기 위해 공격 탐지 룰(Web Attack Rule) 정보를 사용한다. 하지만 기존의 웹 IDS 시스템[3]은 대량으로 생성되는 웹 로그에 대한 별도의 전처리 과정 없이 웹 공격 탐지 룰과 비교하는 방식이므로 실시간으로 수행되는 웹 공격에 효율적으로 대처하지 못하고 있다.

이와 같은 문제점을 해결하기 위해 본 연구에서는 웹 서버에 의해 생성되는 대량의 로그 정보를 대상으로 한 룰 기반 공격 탐지의 효율성을 높이기 위해 전처리 과정을 수행하여 웹 IDS 시스템의 성능을 향상시키는 기법을 제안한다. 제안한 기법에서는 로그 정보를 각 필드별로 분할한 후 B-Tree[7] 기반의 룰 기반 탐색 과정을 수행하도록 하였다.

제안한 전처리 기법을 적용할 경우 기존의 기법보다 대량의 로그 정보를 고속으로 처리하게 되며 룰 기반 탐지 모듈과 연계시킬 수 있기 때문에 웹 서비스 공격을 보다 효과적으로 탐지 및 대처할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 웹 공격의 기법 및 웹 로그 전처리 기법을 소개하고, 웹 공격 탐지를 위한 고속화된 웹 로그 전처리 시스템의 필요성을 제시한다. 3장에서는 기존의 웹 공격 탐지 기법을 고찰하고 본 연구에서 제안하는 시스템 구조 및 전처리 알고리즘을 제시한다. 4장에서는 제안 시스템과 기존의 웹 로그 전처리 시스템과의 성능 비교 평가를 수행하고 5장에서는 결론을 제시한다.

## 2. 웹 서비스 공격 및 대응

웹 사용자의 급증과 동시에 사용자의 개인정보 유출 및 기업 홈페이지의 변조, 금융사고 등과 같은 웹 해킹 사고가 급증하고 있다. 이러한 사건들은 대부분 시스템을 해킹하는 것이 아니라 누구에게나 개방되어 있는 홈페이지를 통해서 시스템에 침투하는 것이다.

웹을 이용한 공격 기법은 꾸준히 고도화되고 있으며 공격 빈도도 급격히 증가하고 있어 이에 대한 대응 기법이 제시되어야 한다. 현재까지 공개된 웹 관련 공격 형태는 알려진 것만 무려 4,000개 이상이고 알려지지 않은 공격을 포함한다면 그 수는 헤아릴 수 없는 상황이다. 특히 최근에는 세션 하이재킹, 패킷 스니핑 및

스캔 기술을 중심으로한 네트워크 공격 기술에서 전반적으로 웹 해킹 및 공격 기술이 급증하고 있는 추세이다. 웹 서비스에 대한 공격 기법을 살펴보면 다음과 같다.

## 2.1 웹 공격 기법

웹 서비스는 버퍼 오버플로우(buffer overflow), 세션 하이제킹 및 SQL 주입/파라메터 주입 공격 등이 가능[8]하여 최근 웹 서버에 대한 DoS 공격으로 발전하고 있다. 대표적인 웹 공격 방식인 크로스 사이트 스크립트 취약점 공격, SQL 주입(SQL Injection) 공격 취약점 공격 방식에 대해 살펴보면 다음과 같다.

### 2.1.1 크로스 사이트 스크립트(XSS) 공격

XSS 공격 기법은 JavaScript, VBScript, Flash, ActiveX, XML/XSL, DHTML 등과 같이 클라이언트 측에서 실행되는 언어로 작성된 코드를 사용자 입력으로 주게 되면 이 코드가 그대로 클라이언트 측 브라우저에서 수행되는 특성을 이용해 악성 스크립트 코드를 웹 페이지, 웹 게시판, 웹 메일에 포함시켜 사용자에게 전송하게 된다[9].

예를 들면 웹 사용자가 취약한 웹서버에 접속 중일 때 공격자는 악성 스크립트를 업로드한 후 웹 사용자에게 악성 스크립트가 있는 링크를 클릭하도록 유도한다. 웹 사용자가 해당 링크를 클릭하게 되면 자신의 쿠키 등의 정보가 공격자에게 전송된다. 공격자는 수집된 정보를 이용해 피해자의 권한을 획득해 피해자의 권한으로 웹서버를 사용할 수 있게 된다. 이때 사용되는 스크립트들은 해커들에 의해 제작되고 배포되어 일반인들도 쉽게 사용할 수 있다.

### 2.1.2 SQL 주입 공격

현재 대부분의 웹 사이트들은 사용자로부터 입력받은 값을 이용해 DB 접근을 위한 SQL 쿼리(query)를 만들고 있다. 사용자 로그인 과정을 예로 들면, 사용자가 유효한 계정과 패스워드를 입력했는지 확인하기 위해 사용자 계정과 패스워드에 관한 SQL 쿼리문을 만든다. 이때 SQL 주입 공격 기법을 통해서 정상적인 SQL 쿼리를 변조할 수 있도록 조작된 사용자 이름과 패스워드를 보내 정상적인 동작을 방해할 수 있다. 이러한 비정상적인 SQL 쿼리를 이용해 다음과 같은 공격이 가능하다[10].

- 사용자 인증을 비정상적으로 통과
- 데이터베이스에 저장된 데이터를 임의로 열람
- 데이터베이스의 시스템 명령을 이용하여 시스템 조작

### 2.1.3 DoS(Denial of Service) 공격

DoS 공격은 특정 시스템에 대한 불법적인 권한을 얻는 적극적인 방법이 아니라, 네트워크와 시스템의 자원을 공격 대상으로 하는 공격 방법이다. 웹에서의 DoS 공격에 대해서는 패킷의 발신지 IP를 중심으로 특정 서버에 대한 자원의 요청 및 오류 메시지의 발생 빈도를 통해 공격을 탐지 할 수 있다[11]. 그러나 현재는 IP 스푸핑과 같은 공격으로 공격의 근원지 역추적이 힘든 실정이다. 스푸핑 방식을 통한 DoS 공격은 사용자 ID 정보를 중심으로 전처리 모듈에서 IP 스푸핑 공격 여부를 확인 할 수 있으며, 웹 서비스의 취약점을 보완하기 위해 룰 기반의 웹 IDS 시스템이 제시되었다. 하지만 기존의 룰 기반의 웹 IDS 시스템은 다음과 같은 취약점을 보이고 있어 이에 대한 대응 기술이 제시되어야 한다.

## 2.2 기존의 룰 기반의 웹 IDS 및 취약점

HTTP 프로토콜은 가장 범용적으로 사용되는 프로토콜중의 하나로 많은 지식을 필요로 하지

않으며 URL 상의 간단한 조작 및 유추 등으로도 웹 해킹이 가능하다는 특징이 있다. 따라서, 기존의 일반적 형태의 IDS 시스템만으로는 웹 서비스에 특화된 공격에 능동적으로 대응할 수 없게 되었기에 웹 IDS 시스템이 필요하게 되었다.

현재 룰 기반의 웹 IDS는 웹 서버 장치 각각의 공격 탐지를 위해 설계된 HIDS (Host-IDS)이다[12]. 이때 사용되는 룰은 아래 표 1과 같은 형식과 같다.

〔표 1〕 룰 기반 웹 IDS 시스템 필드 형식

| 형식 | 예시 |
|---|---|
| rule id | 31103 |
| if_sid | 31100 |
| url | '\|select%20\|select+\|insert%20\|%20from%20\|%20where%20\|union%20\| |
| descripton | SQL injection |
| group | attack, sql_injection |

룰 기반의 웹 IDS는 아래 그림 1과 같이 시스템 내부의 로그 파일을 분석해 정의된 룰과의 비교를 통해 공격을 탐지하는 기법이다.

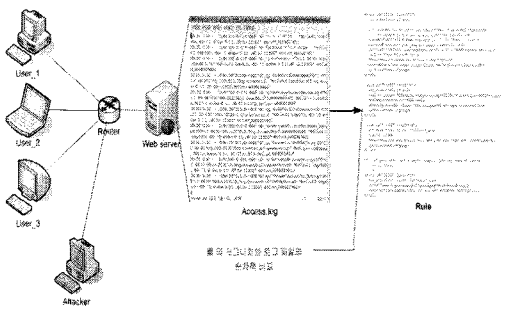


그림 2 룰 기반의 웹 IDS

기존의 전처리 기법은 다음 단계를 통해 공격을 탐지 하여 관리자에게 탐지 결과를 전달했다.

1단계: 로그 수집
2단계: 수집된 로그와 룰과의 비교
3단계: 룰 적용 결과를 관리자에게 전송

그러나 기존의 웹 IDS는 로그 파일의 전처리 과정이 없이 순차 검색을 통해 룰과의 비교를 수행하고 있다. 따라서 최악의 경우 웹 로그 생성 후 일정 시간이 경과된 이후에서야 탐색 결과를 제시하게 된다는 단점이 있다. 기존의 시스템에서는 가공되지 않은 원본 형태의 웹 로그를 대상으로 룰 기반 공격 탐지를 수행하기 때문에 만족할만한 성능을 보이지 못하고 있다. 따라서 본 논문에서는 공격 탐지를 위한 웹 로그 전처리 알고리즘의 설계 및 구현을 통해 빠른 공격 탐지로 효율적인 웹 IDS 성능을 제공하고자 한다.

웹 로그에 대한 전처리 과정을 수행할 경우 앞에서 제시한 웹 서비스의 취약점 들이 이용한 스크립트 업로드, 쿼리 전송 등 웹 서버에서의 공격 행위들에 대해 효율적으로 탐지할 수 있다. 따라서 본 논문에서는 기존의 웹 로그 전처리 기법을 고찰하고, 웹 공격 탐지에 적합한 웹 로그 전처리 기법에 대해 제안하였다.

## 3. 기존 웹 로그 전처리 기법의 취약점

### 3.1 기존 웹 로그 전처리 기법

웹 로그는 일정한 형태의 포맷으로 구성되어진다. 그 종류로는 일반적인 형태의 CLF(Common Log Format)[13] 형태와 확장된 로그 파일 형식인 ELF(Extended Log Format)로 크게 구분 지을 수 있다[14]. 위 포맷은 일반적으로 아래 [표 2]와 같은 형태로 구성되며, 각 필드별로 저장된 정보를 이용하여 웹 IDS 시스템 기반 공격탐지 및 이상탐지 기능에 적용 가능한 형태로 변형되어야 한다.

웹 로그 정보에 대해 기존 [5] 및 [6] 기법

등에서 제시된 웹 로그 전처리 기법에 대해 살펴보고 기존 기법의 문제점을 분석하여 개선된 방법에 대해 제시하고자 한다.

표 2. 로그 구성 필드

| 로그 구성 필드 | 설명 |
|---|---|
| %a | 원격의 IP 주소 |
| %b | 헤더를 포함한 전송량(byte) |
| %{var}e | 환경변수"var" |
| %f | 파일 이름 |
| %h | 원격의 호스트 |
| %{hdr}i | 서버에 들어오는(요청) 헤더 값 "hdr" |
| %{hdr}o | 응답 헤더 값 "hdr" |
| %l | 원격의 로그인 ID(지원한다면) |
| %{label}n | 다른 모듈에서 "label"구성 |
| %p | 서버의 Canonical 포트 번호 |
| %P | 자식 Process ID(PID) |
| %r | 첫 번째 요청 라인 |
| %t | 시간 포맷(CLF 포맷) |
| %{format}t | "format"으로 구성된 시간 포맷 |
| %T | 서버에 요청하는 시간(초) |
| %u | 원격지의 유저 이름(인증시) |
| %U | 요청한 URL |
| %v | 클라이언트 요청에 따른 Canonical 서버 네임 |
| %V | Use CanonicalName 설정에 따른 서버 네임 |

기존 웹 로그 전처리 기법에서는 다음 과정을 수행해 사용자 성향 및 웹 분석을 위해 전처리 과정을 수행하였다.

- 1단계 : 로그 수집 - 현재 웹 로그의 포맷은 일반적으로 가장 많이 사용되는 CLF 포멧, 확장된 로그 파일 형식의 ELF 포맷 형식으로 크게 나눌 수 있다. ELF 형식은 다시 MicroSoft사의 웹서버에서 사용하는 MS-IIS 포멧과 NCSA 계열의 웹서버에서 사용하는 NCSA 포멧 으로 나눌 수 있다.
- 2단계 : Cleaning Log[5] - 분석에 필요하지 않은 아이템을 정제하며 일반적으로 gif, jpeg, jpg, map 등을 로그 파일에서 삭제함. 데이터 용량이 일반적으로1/10에서 1/40 정도로 축소되는 효과를 가져온다. 그러나 현재의 웹 공격 성향을 분석한 결과 이미지 파일로 가장된 백도어 프로그램을 유포 시키는 등 기존의 우회 공격이 존재하고 있어서 본 논문에서는 Cleaning Log 과정을 제외하였다.
- 3단계 : User Identification[5] - 로그 파일에 기록된 사용자 정보를 확인 하여 이동경로를 정제하는 과정이다. 일반적으로 IP Address와 Browser의 환경 정보를 이용하여 전처리한다.
- 4단계 : Session Identification[6] - 사용자의 세션 초과 유무를 점검하여 정제하는 과정이다. 임계치 값을 설정하여 사용자 세션 분류 과정을 수행한다.

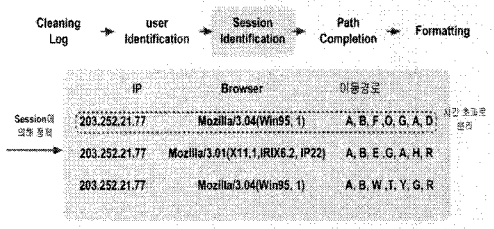


그림 3. 웹 로그 전처리 과정 - Session Identification

- 5단계 : Path Completion - 로그에 기록되지 않은 이동 경로 정보를 연결하는 과정이다. 일반적으로 back 또는 forward 버튼을 눌러 이동한 경우 경로 연결 과정을 수행한다.

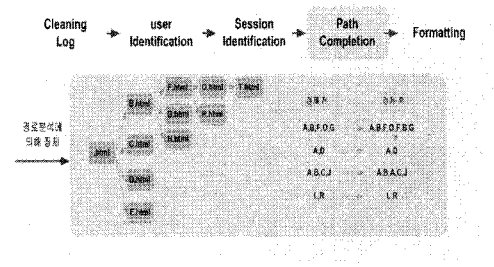


그림 4 웹 로그 전처리 과정 - Path Completion

- 6단계 : Formatting - 웹 로그 분석에 적합

한 정보의 형태로 포맷을 전환한다. 포맷 형태는 사용되는 데이터 분석 기법에 따라 결정한다.

위 각 단계는 기존의 웹 로그 전처리 기법으로 본 논문에서는 공격 탐지를 위한 새로운 전처리 알고리즘 설계를 위해 기존 전처리 기법의 취약점을 고찰한다.

### 3.2 기존 웹로그 전처리 알고리즘의 취약점

초고속 인터넷의 발달로 웹 사용자들이 급격히 증가하였고, 많은 웹 서비스들은 사용자들에게 공개되고 있다. 이런 공개된 웹 서비스에 대해 최근 해커들의 공격 역시 급격히 증가하고 있는 추세이다.

그러나 기존의 웹 로그 전처리 및 분석 기법은 웹 서비스 사용자 성향 및 이용 경로 분석을 위한 고전적 전처리 방식을 그대로 사용하고 있어 다음과 같은 문제점을 가지고 있다.

- 로그내 각 필드 정보의 문자열 특성을 배제하고 단순히 문자열 대상 순차 탐색 방식만을 사용하여 전처리 및 공격 탐지 수행
- 대용량의 웹 로그 정보에 대한 고속처리 기능이 미비
- 웹 공격 탐지를 위한 효율적인 자료구조미비 및 검색 성능 향상을 위한 템플릿 생성 모듈 미비

위 문제점들을 해결하기 위해 본 논문에서는 B-트리 기반 멀티쓰레드를 이용한 로그 분할 및 공격 탐지를 위한 중복 문자열 단위의 로그 인덱싱 테이블을 생성하는 전처리 알고리즘을 제안한다. 이를 통해 대용량 웹 로그 정보에 대한 효율적인 검색 기능을 제공하며 동시에 웹 로그에 대한 룰기반 IDS 시스템에 적용 가능하여 공격 탐지 성능을 향상시킬 수 있는 기반을 제공하고자 한다.

## 4. 제안하는 웹 로그 전처리 알고리즘

본 논문의 제안 기법의 모듈별 전체적인 흐름도는 그림 4와 같다.

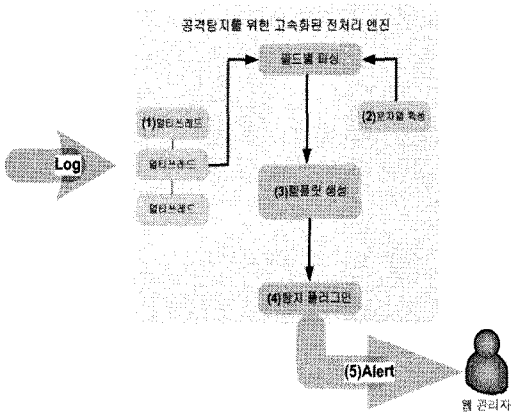


그림 5 제안 시스템의 모듈 구성도

- 1단계(멀티쓰레드): 전체 로그 파일을 멀티쓰레드를 이용해 필드 단위로 분할함

  - 로그 파일($D_{WL_x}$)내 필드($f_x$)별로 분리
    - $D_{WL_x} = \{L_1|L_2|\cdots|L_x\}$
    - $L_x = \{f_{date}, f_{time}, f_{SIP}, f_{DIP}, \cdots\}$

- 2단계(문자열 특성): 필드 단위로 분할된 로그에서 중복 문자열 인덱스 테이블을 구성
  - 필드 단위로 중복 문자를 제거하고 문자열을 키 값($k_{f_x}$)으로 변환
  - 검색 성능 향상을 위해 B-트리 기반 인텍스 테이블($I_f$)을 구성함
  - $I_{f_x} = [k_{f1}|k_{f2}|\cdots|k_{fx}]$
- 3단계(템플릿 생성): 로그 분할 및 인덱싱 테이블을 이용하여 웹 로그 정보를 축약된

정보로 변환하여 템플릿($T_{WL_x}$)을 생성

- $T_{WL_x} = \{I_{f_{1,k0}} | I_{f_{host}} | I_{f_{agt}} | \cdots | L_{f_n}\}$

• 4단계(탐지 플러그인): 인덱싱 정보 기반 웹 공격 탐지 과정을 수행 및 결과 제시

로그 정보로 변환 생성이 가능하다.

즉, 본 연구에서 제시한 기법은 원문 로그의 한 라인이 가지고 있는 각 필드의 인덱스를 통해 B-트리를 거치지 않고 해당 필드의 문자열들을 가져와 바로 하나의 로그 열로 통합하여 다시 원문 로그로 복구 시킬 수 있다.

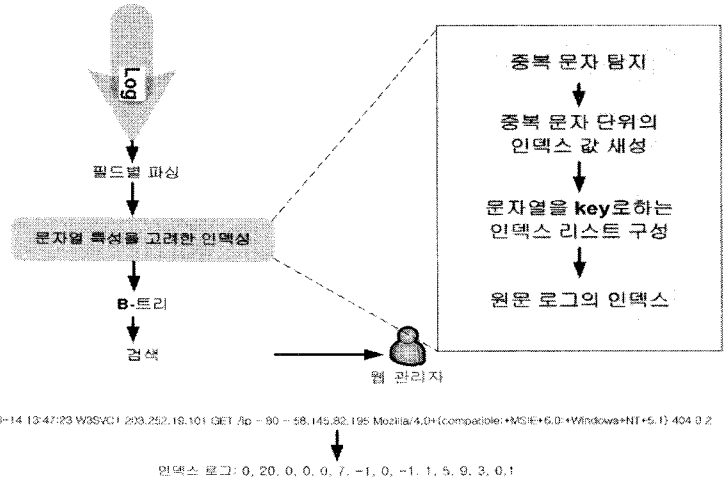


그림 6 제안하는 웹 로그 전처리 방식

웹 IDS 시스템에 접목시키기 위해서 웹 로그($D_{WL_x}$) 정보내 문자열 특성을 고려한 중복 문자열 처리 방식은 원문 웹 로그 파일의 각 필드($f_x$) 단위별로 구성된 테이블($I_{f_x}$)에 중복된 문자열을 인덱싱(indexing)하는 방식이다.

또한 본 연구에서는 그림 5와 같이 웹 로그 내 중복 문자열 인덱스 처리 과정을 수행하므로 원문 웹 로그보다 물리적인 효율성을 제공한다. 예를 들어 웹 로그의 특정 필드에 특정 문자열을 검색할 때 문자열의 검색보다는 인덱스된 키 값($k_{f_x}$)을 통해 검색하므로 효율적이다. 만일 키워드 검색 과정을 수행하고자 할 경우, 전체 로그 파일의 각 필드 정보가 이미 인덱스 정보로 저장되어 있기 때문에 SQL 주입 공격 등과 같은 웹 공격 시도에 대해 빠른 검색 기능을 제공하며, 인덱스 정보로부터 본래의 웹

각 단계별로 수행되는 과정에 대해 설명하면 다음과 같다.

## 4.1 웹 로그 필드별 분류 및 고속 처리

구체적으로 그림 6과 같이 멀티쓰레드 기법을 사용하여 원문 로그를 분할 통치하는 알고리즘 기법을 사용한다. 대량의 웹 로그 정보에 대해 멀티쓰레드 방식으로 분할하여 전처리 과정을 수행한다. 각 쓰레드에 할당되는 정보는 대용량 웹 로그내 동일 필드별 정보를 대상으로 전처리 과정을 수행하도록 하였다.

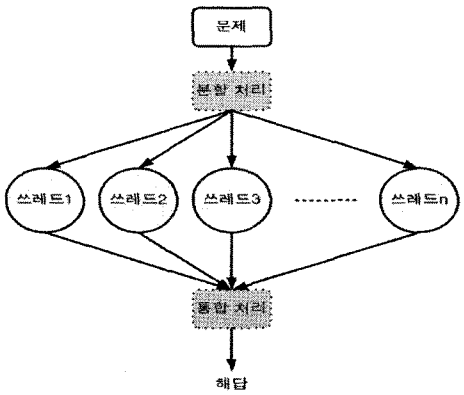


그림 7 멀티스레드 기반 고속 처리 구조

제안하는 전처리 기법은 로그 파일을 분할하여 룰 비교를 효율적으로 할 수 있게 중복 문자열을 처리한다. 이때 생성되는 템플릿은 아래와 같은 구조로 생성된다.

〔표 3〕 템플릿 생성 예시

| 인덱스 | cs-uri-query | 발생 빈도 |
|---|---|---|
| 1 | menu=business07 | 9 |
| 2 | dept_seq=17 | 3 |
| 3 | cust_no=126 | 6 |
| 4 | \|77\|800a0046\| | 9 |
| . | . | . |
| n-1 | menu=service_02 | 9 |
| n | menu=service02&m=1 | 2 |

본 논문에서 비교 대상이 될 로그 파일의 필드는 원격지 IP 주소(%a), 헤더를 포함한 전송량(%b), 첫 번째 요청 라인(%r), 요청한 URL(%U)들이다. 각 필드별로 나누어진 로그 데이터는 본 연구에서 제안하는 멀티쓰레드 기반 전처리 모듈에 의해 처리된다.

## 4.2 문자열 특성을 고려한 중복 문자열 전처리

문자열 특성을 고려한 중복 문자열 처리는 원문 로그 파일의 각 필드 단위로 구성된 테이블에 중복된 문자열을 인덱싱하는 기법이다.

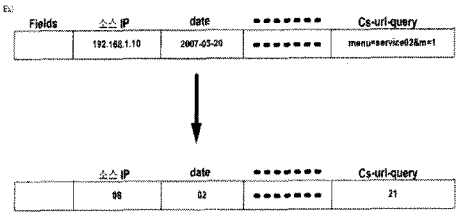


그림 8 원문 로그의 인덱싱

각 필드별로 분할된 로그 파일들은 중복 문자열 처리 과정을 수행한 결과 약 50%의 성능 향상을 가져올 수 있으며, 전체 로그 파일의 각 필드 정보들은 인덱스 정보들로 저장되어 완전한 문자열로의 변환/생성이 가능하다.

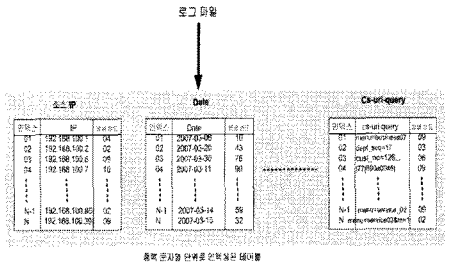


그림 9 중복 문자열 단위로 인덱싱된 필드

## 4.3 이진 탐색 알고리즘을 이용한 룰 비교

기존의 룰 기반의 웹 IDS는 access.log 파일의 순차적인 문자열 탐색을 통해 웹 공격을 탐지했다. 그러나 웹 공격의 입력 값 부재 공격은 클라이언트가 서버에게 요청하는 쿼리문에 의해 공격이 이루어 진다. 그러므로 본 논문에서 제안하는 기법은 룰 비교 시 불필요한 탐색 과정을 분할 통치법을 이용해 각각의 필드별로 분할하고 룰과의 비교를 이진 탐색 알고리즘을 이용해 기존 알고리즘을 개선한다.

크기가 $n$인 선형 리스트에서 원소들의 키

값이 주어진 키 값과 같을 확률이 $\frac{1}{n}$로 모두 같다고 할 때 임의의 위치 $k$에서 탐색키를 찾는데 $k$번 비교 연산이 필요하며, 평균 비교 횟수는 수식 1과 같다.

$$\sum_{k=1}^{n} k\frac{1}{n} = \frac{n+1}{2} \quad \cdots\cdots (1)$$

제안 기법인 이진 탐색 법은 한번 탐색시마다 탐색 원소의 개수가 반으로 줄어든다. 수식 2는 $n$이 거듭제곱 수라 할 때 최악의 실행 시간 $T(n)$이다.

$$\begin{aligned} T(n) &= T(\tfrac{n}{2}) + \Theta(1) \\ &= T\tfrac{n}{2^2} + \Theta(1) + \Theta(1) \\ &= \cdots\cdot \\ &= T(\tfrac{n}{2^{\lg n}} + \Theta(1) + \Theta(1) + \cdots + \Theta(1) \quad \cdots\cdots (2) \\ &= \Theta(\lg n) \end{aligned}$$

$n$이 거듭제곱 수가 아닌 경우의 이진 탐색법의 시간 복잡도는 $O(\lg n)$이다[7].

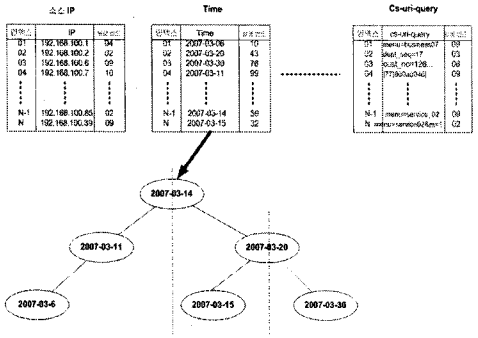


그림 10 웹 로그내 공격 탐색 과정

본 논문에서 제안한 룰 비교시 탐색 기법은 실제 비교 대상의 문자열들이 정렬되어 있어 실제 룰과의 비교는 이진 트리의 탐색에서 이루어진다. SQL 주입 공격 및 파라메터 주입 공격은 클라이언트가 요청한 url의 비교를 통해 공격 탐지가 가능하다. 그러므로 기존의 룰 중 url의 문자열을 비교하여 공격을 탐지한다.

```
- <rule id="31104" level="6">
  <if_sid>31100</if_sid>
- <!--
    Attempt to do directory transversal, simple sql injections,
    - or access to the etc or bin directory (unix).
  -->
  <url>%027|%00|%01|%7f|%2E%2E|%0A|%0D|../..|..\..|echo;|..</url>
  <url>cmd.exe|root.exe|_mem_bin|msadc|/winnt/|</url>
  <url>/x90/|default.ida|/sumthin|nsiislog.dll|chmod%|wget%|cd%|</url>
  <url>cat%|exec%|rm%20</url>
  <description>Common web attack.</description>
  <info>http://www.armbrustconsulting.com/LogEntries.html</info>
  <group>attack,</group>
</rule>
```

그림 11 웹 IDS 룰 형식 예시

## 5. 구현 결과 및 성능 평가

웹 IDS 시스템을 위한 전처리 시스템 구현 결과는 다음과 같다. 본 연구에서 사용한 IIS 로그는 1년 동안 수집한 웹 서버 로그를 이용하였다. Visual Studio 6.0을 이용하였으며 C++ 언어 기반으로 개발하였다.

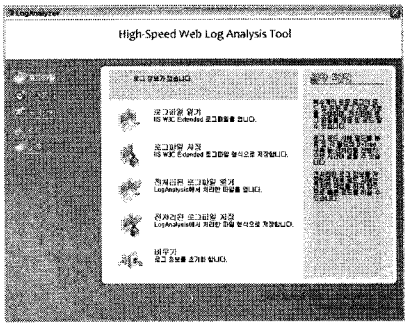


그림 11 시스템 구현 결과

구현한 시스템을 이용할 경우 웹 로그 정보에 대한 고속 세션 분류 및 필드별 검색 기능을 제공하며 SQL 주입 공격 및 파라메터 주입 (Parameter-Injection) 공격 등과 같이 웹 서버에 대한 공격을 시도할 경우 발생하는 로그 정보

에 대한 효율적인 검색 기능을 제공하였다.

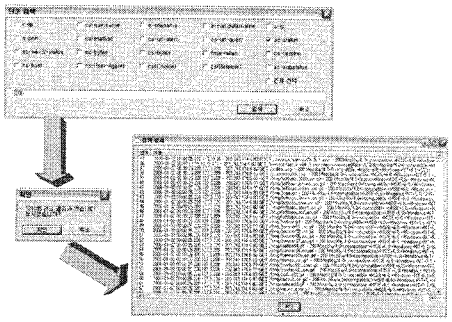

그림 12 웹 로그내 세션 분류 및 공격 검색

본 제안 기법에서는 멀티쓰레드를 이용해 MS-IIS 로그 형태로 생성된 대용량 로그 파일에 대해 필드별 고속 분할 및 전처리 과정을 수행하였다. 멀티쓰레드를 이용한 로그 파일의 분할 성능 및 효율성을 분석하면 다음 그림과 같이 대용량의 웹 로그 정보에 대해 다중 쓰레드를 사용하였기 때문에 전체적인 프로세스 부하를 줄일 수 있었으며 각 쓰레드 별 처리율은 높아지는 것을 확인할 수 있었으며, 결과적으로 웹 로그 데이터의 크기에 상관없이 일정한 처리율을 제공하였다.

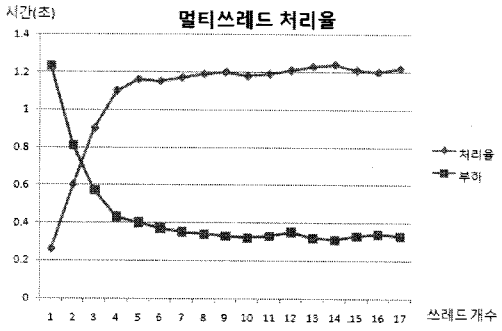


그림 13 멀티쓰레드 기반 처리율

본 연구에서는 49MB, 99MB 및 149MB 크기의 로그 파일을 대상으로 제안 기법과 기존 LogParser[15] 기반 전처리 기법에서의 로그 파일 로딩 시간을 비교하였다. 성능 비교 결과 아래 그림 14와 같이 본 연구에서 제시한 기법을 이용할 경우 149MB 로그 파일에 대해 기존 기법(30.787초)보다 개선된 결과(20.688초)를 보이고 있다.

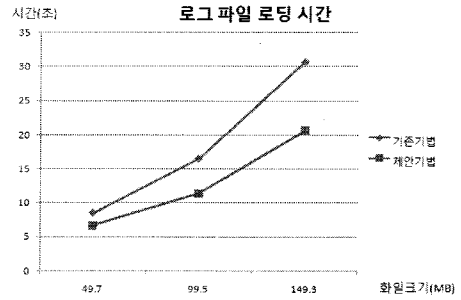


그림 14 로그 파일 로딩 시간 비교

그림 15는 앞에서 제시한 세 가지 형태의 로그 파일을 이용하여 공격 탐지에 적용하였을 경우 공격 탐지 룰 비교 시간 측정 결과이다. 성능 비교 결과 기존 전처리 기법에서는 대용량의 로그 파일인 경우 공격 탐지를 위한 프로세스 검색 시간이 상대적으로 많이 소요되어 효율이 떨어지지만 제안 기법을 적용할 경우 로그 파일에 대한 빠른 로드와 인덱싱 모듈을 통해 룰 정보에 대한 효율적인 비교/검색 과정을 수행한다는 것을 확인할 수 있었다.

본 연구에서 제시한 기법은 대용량의 웹 로그에 대해 한 번의 전처리 과정을 수행하여 B-트리 형태로 축약된 인덱싱 로그 정보를 생성하고 룰 검색 및 로그 검색을 수행하기 때문에 기존 기법 보다 전체적인 검색 시간이 줄어든다는 것을 확인할 수 있었다. 따라서 본 연구에서 제시한 기법인 경우 인덱싱 방식에 기반한 룰 검색/비교 과정을 수행하기 때문에 로그 파일의 크기/용량을 줄이면서도 효율적으로 대용량 로그 파일에 대한 빠른 검색 기능을 제공한다는 것을 확인할 수 있다.

또한 제안한 기법을 이용할 경우 대용량 웹 로그내 공격 로그 정보 등에 대해 정확하게 검색하여 세션별로 제공하는 기능이 있음을 확인

할 수 있었다.

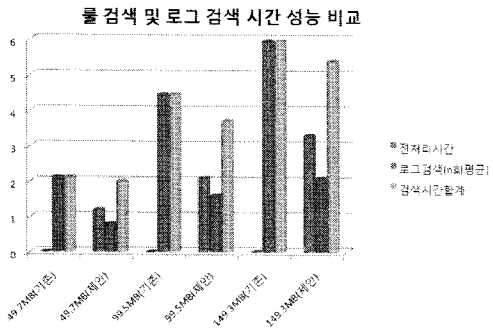


그림 15 룰 검색 및 로그 탐지 시간 비교

[표 4]는 기존의 전처리 기법과 제안 전처리 기법의 기능을 비교한 표이다. 본 연구에서 제안한 기법은 웹 IDS 시스템에 적용 가능하며, 멀티 쓰레드 기반 고속 전처리 기능을 제공하며 효율적 공격 탐지 기능을 제공하기 위해 템플릿을 생성하는 기능이 있어 기존의 방식보다 개선된 기능을 제공한다.

Microsoft사에서 개발하여 배포하고 있는 기존의 LogParser 시스템[15]인 경우 고속 인덱싱 기능을 제공하지 못하고 있으며 단순 쿼리 검색 기능만을 제공한다. 또한 로드 가능한 웹 로그 파일의 크기에도 제한이 있으며 공격 탐지 기능을 제공하지 못하고 있다. 하지만 본 연구에서 제시한 기법인 경우 필드별 분할 및 중복 문자열 처리 기능을 제공하며, B-트리에 기반하여 고속 인덱싱 구축 기능을 제공한다.

〔표 4〕 기존 기법과의 기능 비교 (○: 기능 수행, x: 기능 없음)

| 비교 항목 \ 전처리 | 기존 전처리 | 제안 전처리 |
|---|---|---|
| 공격탐지 | X | ○ |
| 로그 분할 | X | ○ |
| 멀티쓰레드 사용 | X | ○ |
| Cleaning Log | ○ | X |
| 템플릿 생성 | X | ○ |
| 다중 룰 비교 | X | ○ |

## 6. 결론

인터넷 사용의 급증으로 웹 서비스는 급격히 발전하고 있다. 그러나 웹 서비스의 양적인 성장에 비해 보안 분야에 대한 발전은 지지부진한 상태이다. 급격히 증가하고 있는 웹 서비스에 대한 해킹 사고들로 웹 보안에 대한 심각성은 충분히 인식 할 수 있다. 현재 웹 보안을 위한 많은 시스템들이 있다. 그중에 네트워크 기반의 IDS는 웹 서비스 환경에 적합한 공격 탐지 시스템이라 할 수 없다. 웹 서비스의 특성상 서버와 사용자 간의 요청과 응답 로그 분석을 통해 공격 여부를 판단 할 수 있기 때문이다.

이에 본 논문에서는 룰 기반 웹 IDS 시스템에서의 공격 탐지 효율을 높이기 위해 멀티쓰레드를 이용한 웹 로그의 분할 및 중복 문자열의 전처리를 통해 룰 비교를 위한 템플릿을 생성해 룰 비교의 효율을 향상 시켰다.

## 참 고 문 헌

[1] 정보통신부 한국인터넷진흥원 "2005년 하반기 정보화 실태조사 요약보고서", 2006년

[2] 케이티하이텔(주) "2006년 하반기 CRM 1단계 개발 제안 요청서", 2006.

[3] Giovanni vigna, William Robertson, Vishal

Kher, "A Stateful Intrusion Detection System for World-Wide Web Server", In ACSAC, pp.2-10, 2003.

[4] Robert Drum, "IDS AND IPS PLACEMENT FOR NETWORK PROTECTION", CSSIP, 26 March 2006.

[5] Zhenglu Yang, Yitong Wang, Masaru Kitsuregawa, "An Effective System for Mining Web Log", Frontiers of WWW Research and Development - APWeb 2006, LNCS, 2006.

[6] R. Cooley, B. Mobasher, and J. Srivastava, "Data preparation for mining world wide web browsing patterns," Knowledge and Information Systems, vol. 1, no. 1, pp. 5-32, 1999.

[7] 조유근 외 3명 "알고리즘" 이한 출판사 2005.

[8] Sacha Faust, "Are your web applications vulnerable?", SOAP Web Services Attack, 2007.

[9] 김기남, "WEB 2.0 에서의 보안취약성 분석 및 대응책에 관한 연구", 동국대 학위논문, 2008.

[10] Bisson R. "SQL Injection", ITNOW, Oxford Univ., Vol.47, No.2, 2005

[11] http://www.ossec.net/en/loganalysis.html

[12] http://www.ossec.net

[13] http://httpd.apache.org/docs/1.3/logs.html

[14] http://www.w3.org/TR/WD-logfile.html

[15] Mircosoft Log Parser 2.2, http://www.dsus4.net/2460597

## ◐ 저 자 소 개 ◑

**이 형 우(Hyung-Woo Lee)**

1994년 고려대학교 전산과학과 졸업(학사)
1996년 고려대학교 대학원 전산과학과 졸업(석사)
1999년 고려대학교 대학원 전산과학과 졸업(박사)
1999년~2003년 2월 천안대학교 정보통신학부 조교수
2003년~현재 한신대학교 컴퓨터공학부 부교수
관심분야 : 정보보호, 유무선 네트워크 보안, 웹 보안기술
E-mail : hwlee@hs.ac.kr