ISSN 2383-6318(Print) / ISSN 2383-6326(Online)
KIISE Transactions on Computing Practices, Vol. 24, No. 4, pp. 190-197, 2018. 4
https://doi.org/10.5626/KTCP.2018.24.4.190

리눅스 아파치 웹 서버 실시간 로그 분석을 통한 공격 탐지 프로그램 개발

(Implementation of Linux Apache Web Server Attack Detection Program through Real-time Log Analysis)

박 재 연 † (Jaeyeon Park)　이 송 연 † (Songyeon Lee)　이 하 은 † (Haeun Lee)　이 종 우 †† (Jongwoo Lee)

요 약 단순 검색, 원격 채팅부터 IoT, 클라우드 컴퓨팅 등 다양한 분야에 널리 사용되고 있는 웹 서버는 높은 접근성 때문에 다른 시스템보다 더 높은 보안성이 필요하고, 이에 맞춰 다양한 연구가 이루어지고 있다. 그런데도 예상치 못하는 곳에서 발생하는 취약점으로 인해 개인정보 유출, 서버 마비와 같은 중대한 피해들이 지속해서 발생한다. 특히 ISEC 2017에서 제기된 보안 전문가들의 의견에 따르면, IoT 기기의 취약성, 보안이 취약한 오픈소스의 사용, 보안 기초 부족 등의 이유로 보안 사고가 증가하고 있다. 본 논문에서는 웹 서버로 들어오는 공격을 웹 서버 로그 차원에서 실시간으로 탐지하여 개인적, 사회적 피해를 최소화하는 것을 목표로 하였다. 이를 위해 리눅스 환경에서 아파치 웹 서버 접근 로그를 실시간으로 분석하여 DDoS, SQL Injection, RFI, Webshell 업로드 공격을 탐지하고, 이를 관리자에게 통지해주는 GUI 기반 로그 탐지/뷰어를 개발하였다. 이는 위의 공격들을 매우 높은 확률로 탐지하며, 빠르게 들어오는 로그의 누락 또한 최소화하여 사용자를 웹 서버 공격으로부터 안전하게 지켜준다.

키워드: 리눅스, 웹 서버 접근로그, 실시간, DDoS, SQL Injection, RFI, Webshell 업로드, GTK +3, 통계

Abstract With wide usage in various fields, such as simple search, remote chat, IoT and cloud computing, the web requires a higher security than other systems due to its high accessibility. Therefore, various studies have been conducted accordingly to find ways to secure the web. Vulnerabilities that occur unexpectedly in the web cause serious damages, such as personal information leakage and server down. In particular, security experts from ISEC 2017 have reported that security accidents are increasing due to the weaknesses in IoT devices, usage of open sources with weak security and lack of security bases. In this paper, we aimed to minimize the individual and social damages by using a real-time web server log detector to locate web server attacks. In order to achieve this goal, we developed a GUI-based log detection/viewer. The viewer detects DDoS, SQL Injection, RFI and Webshell upload attacks through real-time analyzing of the Apache web server access log in a Linux environment, which notifies the administrator of the attack afterwards. It detects the above mentioned attacks with very high probabilities of occurring, and minimizes the loss of fast incoming logs, thus safeguarding users from Web server attacks.

Keywords: linux, access log, real time, DDoS, SQL injection, RFI, webshell upload, GTK + 3, analytics

† 비 회 원 : 숙명여자대학교 ICT융합공학부
jaey1774@naver.com
jb3690@naver.com
leehaeunhi@naver.com
†† 종신회원 : 숙명여자대학교 ICT융합공학부/ICT융합연구소 교수
(Sookmyung Women's Univ.)
bigrain@sm.ac.kr
(Corresponding author임)

논문접수 : 2017년 12월 14일
(Received 14 December 2017)
논문수정 : 2018년 2월 10일
(Revised 10 February 2018)
심사완료 : 2018년 2월 12일
(Accepted 12 February 2018)

정보과학회 컴퓨팅의 실제 논문지 제24권 제4호(2018. 4)

1. 서 론

일반적인 웹 사이트로부터 IoT 인터넷 서비스까지 다양한 서비스가 개발되면서 웹 보안의 중요성이 대두되고 있다. 특히 국가와 기업 차원에서 중요한 정보의 유출을 막기 위해 다양한 방법을 고안하고 있으나 보안 기초 지식 부족 및 관리 소홀과 같은 이유로 여전히 위험에 노출되어 있다[1]. 이를 해결하기 위해 본 논문에서는 웹 서버로 들어오는 접속 로그들을 실시간으로 분석하여 서버 관리자에게 현 서버 상태를 알려주는 프로그램을 개발하였고, 프로그램 명칭을 'Web Attack Defender (이하 WAD)'로 하였다.

이 프로그램을 구현하기에 앞서, 본 논문에서는 웹 서버 접근 로그를 분석하는 기존 리눅스용 프로그램을 조사한 결과 다음과 같은 한계점들을 발견하였다.

첫째, 대부분 네트워크 드라이버 수준에서 패킷을 분석하고 있어 서버에 과부하를 준다.

둘째, 단순히 로그를 보여줄 뿐, 현재 어떤 공격이 이루어지고 있는지에 대한 판단은 해주지 않는다.

셋째, 실시간으로 보여주는 기능이 부족하다.

이와 같은 분석을 통해, WAD가 지향해야 할 방향을 다음과 같이 세 가지로 설정하였다.

첫째, 장치 드라이버가 아닌 애플리케이션 계층에서 동작한다.

둘째, 로그를 파싱하여 특정 패턴이 감지될 경우 공격으로 인식하여 알려준다.

셋째, 모든 과정을 실시간으로 처리하고 보여준다.

이러한 세 가지 방향을 중심으로 WAD의 주요 기능을 구상하였다. WAD의 핵심 기능은 아파치 웹 서버 접근 로그 실시간 분석과 공격 탐지이다. WAD가 탐지 가능한 공격에는 DDoS, SQL injection, RFI, Webshell upload 공격이 포함된다. 또한, WAD는 사용자에게 친숙한 UI를 제공하기 위해 GTK +3을 사용하여 시각적으로 직관적인 정보를 전달한다. 이와 더불어 IP 접근 횟수, 각 공격별 탐지 횟수 등도 보여준다.

2. 관련 연구

2.1 goAccess

goAccess는 유닉스/리눅스 시스템에서 브라우저를 통해 터미널에서 실행되는 오픈소스 실시간 웹 로그 분석기 및 대화식 뷰어[2]로, 그림 1은 그 실행화면이다. 비주얼 웹 서버 보고서를 즉각적으로 필요로 하는 시스템 관리자를 위해 빠르고 유용한 HTTP 통계를 제공한다. 하지만, WAD와 달리 로그 뷰어 기능을 제공하지 않으며, GUI를 지원하지 않기 때문에 실행 방법이 복잡하다.

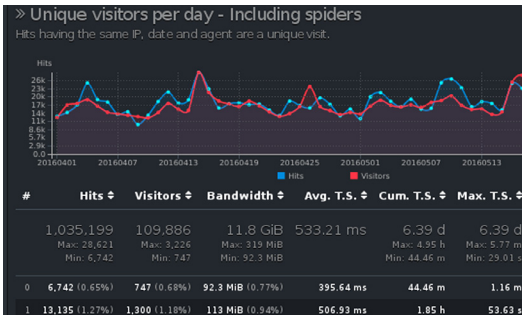

그림 1 goAccess 실행 모습

Fig. The screen of goAccess

2.2 Visitors

Visitors는 리눅스, 윈도우 및 기타 유닉스 계열 운영체제를 위한 매우 빠른 웹 로그 분석기이다[3]. 웹 서버 로그 파일을 입력으로 받아서 그림 2의 화면과 같이 다양한 보고서 형태로 통계를 출력한다. 별도의 설치를 요구하지 않고, 스트림 모드를 통한 실시간 통계를 내며, html 형태의 리포트를 제공하는 프리웨어 프로그램이다. 하지만 실행 시 읽은 로그만 분석하므로 로그의 내용이 실시간으로 반영되지 않는다.

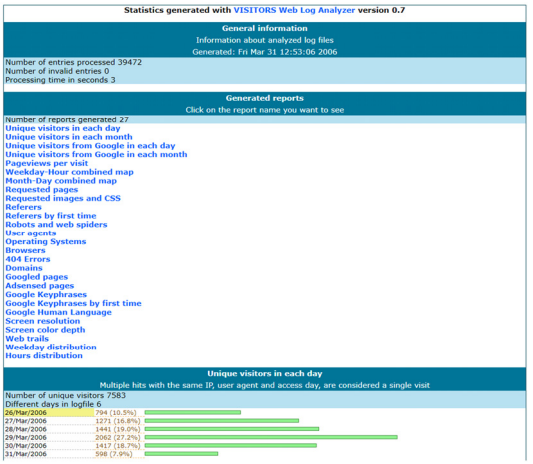

그림 2 Visitors 인터페이스

Fig. 2 The report screen of Visitors

2.3 Webalizer

Webalizer는 빠르고 무료로 제공되는 웹 서버 로그 파일 분석 프로그램이다[4]. 표준 웹 브라우저로 보기 위해 매우 상세하고 쉽게 구성할 수 있는 HTML 형식의 사용 보고서를 그림 3의 형태로 생성한다. 하지만 실시간으로 로그를 분석하지 않으며, 웹 공격을 탐지해 주지도 않는다.

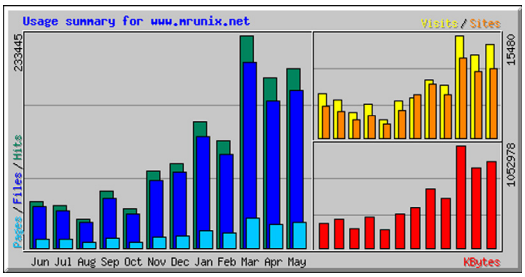

그림 3 Webalizer 인터페이스
Fig. 3 The report screen of Webalizer

2.4 탐지 가능 공격

2.4.1 DDoS

단체표준 TTAK.KO-06.0358 내 정의에 따르면 서비스 거부(Denial of Service) 공격은 시스템을 악의적으로 공격해 해당 시스템의 자원을 부족하게 하여 원래 의도된 용도로 사용하지 못하게 하는 공격이다[5]. 특정 서버에게 수많은 접속 시도를 만들어 다른 사용자가 정상적으로 서비스를 이용하지 못하게 하거나, 서버의 TCP 연결을 바닥내는 등의 공격이 이 범위에 포함된다.

분산 서비스 거부 공격(Distributed DoS)은 여러 대의 공격자를 분산적으로 배치해 동시에 서비스 거부 공격을 하는 방법으로 단체표준 TTAK.KO-12.0262에 정의되어 있다[6]. 이는 IAB(인터넷아키텍쳐 위원회, Internet Architecture Board)[7]의 정당한 인터넷 사용 정책에 반하는 것으로 여겨지며, 거의 모든 인터넷 서비스 공급자가 사용자들에게 허용할 수 있는 사용 정책도 위반한다. 또한, 개별 국가의 법률에도 저촉된다.

2.4.2 SQL 삽입(injection)

단체표준 TTAK.KO-12.0002/R3에 따르면 SQL 삽입은 데이터베이스 내 SQL 명령문의 구문 오류를 이용하여 발생하는 공격 기법이다[8]. 2017년 3월 숙박업체 예약 애플리케이션 '여기어때'를 해킹해 이용자 99만 명의 숙박 예약 정보와 회원 정보, 가맹점 정보 등 총 341만 건을 빼내 이를 미끼로 '여기어때' 측에 금품을 요구하며 협박한 사건[9]과 2015년 9월 국내 최대 휴대폰 커뮤니티 '뽐뿌'의 홈페이지가 SQL 삽입 침입으로 190만 건의 회원 ID, 비밀번호, 생년월일, 닉네임, 가입일, 회원 점수 등을 유출 당한 사건이 대표적인 사례이다[10].

SQL 삽입의 동작 원리는 다음과 같다. 웹 페이지의 ID 찾기 폼에서 자신의 닉네임을 입력하면 SQL 구문 내에 닉네임이 삽입되고 해당 질의가 실행된다. 이때, 해커가 악의적으로 "' OR '1"이라는 구문을 삽입할 경우, SQL 구문은 "select id, name, email from member where nick='' OR '1'"로 변형된다. 이 경우 WHERE 절의 모든 값이 참으로 판단하기 때문에 그림 4와 같이 다른 사용자의 정보가 출력될 위험이 있다.

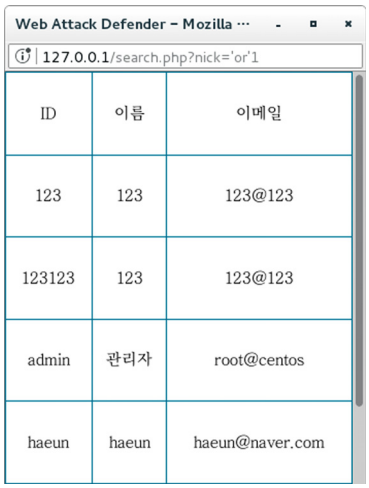

그림 4 비정상적인 회원 정보 출력 화면
Fig. 4 Abnormal member information output screen

2.4.3 RFI/LFI

RFI/LFI는 Remote File Inclusion/Local File Inclusion의 약자로, 스크립트 일부를 다른 파일에서 참조할 때 공격자가 지정한 외부 서버의 URL로부터 읽어오도록 만들어 임의의 스크립트가 실행되도록 하는 공격이다[11]. 외부에 설치된 악의적인 코드를 웹 서버에 전달하는 기법이다. 2013년 2월 식·음료 및 약품 생산 전문 기업 '일화'의 웹사이트에 특정 웹 사이트에서 파일을 내려 받아 그 파일이 사용자 정보 탈취 및 또 다른 기능의 악성 파일을 내려 받게 하는 다운로더 역할을 하는 악성 링크가 삽입된 사건이 RFI 공격의 대표적인 예이다[12].

이러한 공격은 PHP 내의 Include 함수를 악용한 것으로, 웹 페이지 내에 웹 서버 내의 파일 혹은 외부 URL을 의도적으로 포함한다. 그림 5와 같이 answer 파라미터에 경로명을 입력할 경우 그림 6과 같이 일반 사용자에게 보이면 안 되는 웹 서버에 등록된 사용자의 정보가 표시된다.

2.4.4 웹쉘 업로드(WebShell Upload)

웹쉘 업로드는 웹 서버에 명령을 실행하여 권리자 권한을 획득해 행하는 공격 방법이다[13]. 웹 애플리케이션의 첨부파일에 대한 부적절한 신뢰와 불충분한 점검으로 인해 악의적인 원격 공격 코드가 웹 서버로 전송, 실행되는 방법으로 권리자 권한을 획득한 후 웹 페이지의 소스 코드 열람은 물론 서버 내 자료 유출, 비밀문 프로그램 설치 등 다양한 공격이 가능하다. 그림 7은 웹쉘이 실제 실행되는 화면으로, 웹 서버 내의 파일 목록을 보거나 명령어 실행이 가능하다. 2012년 'EBS'가 웹쉘 업로드 공격을 받은 사례가 있다[14].

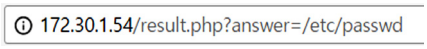

그림 5 잘못된 파라미터 입력
Fig. 5 Incorrect parameters

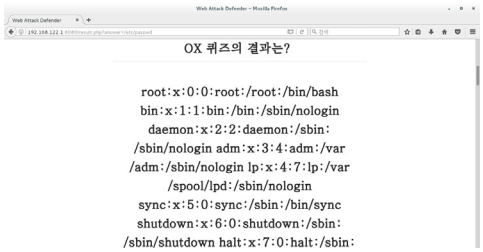

그림 6 잘못된 파라미터로 인한 출력
Fig. 6 Output due to incorrect parameters

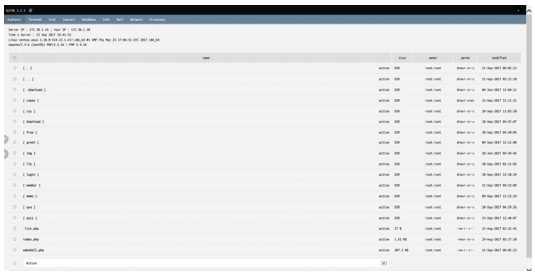

그림 7 웹쉘 실행 화면
Fig. 7 Webshell launch screen

3. 시스템 설계

3.1 시스템 목표

WAD는 크게 '실시간', '공격 탐지'라는 큰 주제를 잡고 다음과 같이 세부적인 설계 목표를 설립하였다.

첫째, 접근 로그를 실시간으로 뷰어에 불러온다. 실제 로그 파일에 쌓이는 동시에 보여야 하므로 별도의 DB를 구축하지 않는다.

둘째, 각 공격에서 드러나는 패턴을 판별하여 뷰어를 보는 동안 관리자가 빠르게 확인할 수 있도록 한다.

셋째, 공격 탐지로 인해 생성된 데이터들을 향후 응용할 수 있도록 가공한다. 해당 서버에 특정 IP의 접속횟수와 전체 IP의 접속 횟수를 계산하고, 어떤 종류의 공격이 감지되었는지 알아냄으로써 일종의 블랙리스트를 작성할 수 있도록 한다.

3.2 시스템 구성

WAD의 시스템 구조는 그림 8과 같다.

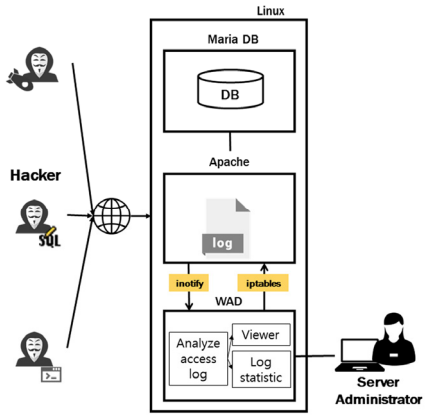

그림 8 WAD의 시스템 구조도
Fig. 8 The structure of WAD

4. 시스템 세부 구현

4.1 개발 환경

WAD의 개발환경은 다음과 같다. Linux CentOS 7.0 환경에서 C언어를 기반으로 구현을 진행하였으며, GUI는 내장된 GTK + 3[15]을 이용하여 구현하였다.

4.2 시스템 기능 구현

4.2.1 공격 탐지 기능

4.2.1.1 DDoS 탐지 구현

일반적으로 DDoS는 패킷을 이용하여 탐지하고 하드웨어 차원에서 차단하지만, WAD 프로그램은 애플리케이션 계층에서 DDoS 공격을 탐지한다. '패킷 카운팅을 이용한 DoS/DDoS 공격 탐지 알고리즘 및 이를 이용한 시스템'[16]에서는 패킷량을 이용한 DDoS 탐지 알고리즘을 제안하였는데, 본 연구는 패킷 카운팅 대신 접근 로그량을 이용하여 DDoS를 탐지한다.

WAD의 DDoS 탐지 알고리즘은 그림 9와 같다. 메인 함수에서 DDoS를 탐지하는 스레드를 생성한다. 이 스레드는 단위 시간(본 연구에서는 5초로 정하였다)마다 DDoS를 검사한다. 스레드에서는 평균 로그량과 단위 시간당 접근 로그량의 중간값을 계산한다. '평균 로그량과 단위 시간당 접근 로그량의 차'의 절댓값이 허용 임

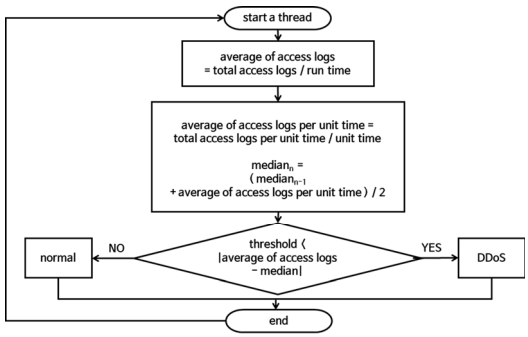

그림 9 DDoS 탐지 알고리즘
Fig. 9 The algorithm of DDoS detection

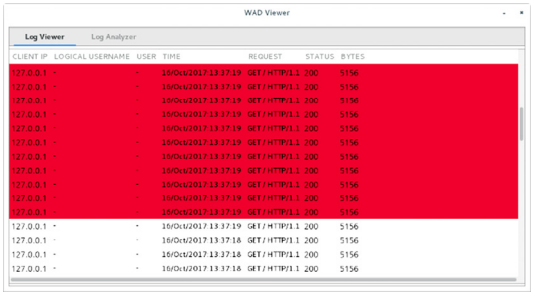

그림 10 DDoS 탐지 화면
Fig. 10 The viewer of detection DDoS

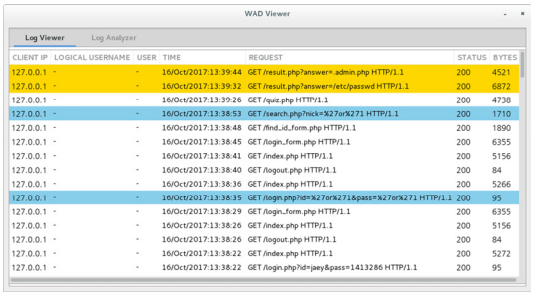

그림 12 SQL injection, RFI 탐지 화면
Fig. 12 The viewer of detection SQL injection, RFI

계치보다 크면 DDoS로 판별한다. 이때 허용 임계치는 평균 로그량과 중간값을 이용하여 구하는데 일정 기간 학습된 결과이다.

WAD 프로그램은 그림 10) 같이, DDoS 공격을 탐지하였을 경우 로그를 빨간색으로 출력한다.

4.2.1.2 SQL 삽입/RFI,LFI 공격 탐지 구현

SQL 삽입과 RFI/LFI 공격은 GET 파라미터로 공격 문자열이 전달될 때 감지가 가능하다. 따라서 본 논문에서는 각각의 공격에 해당하는 문자열을 '공격 탐지 문자열'로 지정하여 공격을 탐지하였다.

WAD에서 이러한 공격을 탐지하는 상세한 알고리즘은 다음의 그림 11과 같다. 메인 함수에서 접근 로그 파일에 변화가 생기면 로그를 한 줄씩 읽어 들이는 스레드를 생성한다. 이때 읽은 로그의 리퀘스트, 즉 사용자가 웹페이지 상에서 어떤 작업을 수행하였는지 파싱하여 GET 파라미터 내에 공격 탐지 문자열이 있는지 확인한다.

SQL 삽입 취약점 탐색을 위해 사용되는 문자열을 응용하여 해당 공격을 탐지하였다. 가장 대표적인 문자열인 "' or '1"을 기반으로, "'"와 공백을 공격 탐지 문자열로 지정하였다. 아파치 접근 로그는 해당 문자를 ASCII Code로 변환하여 기록하는 경우가 존재하므로, 각각에 해당하는 "%27"과 "%20"을 공격 탐지 문자열에 추가하였다.

RFI의 경우 공격자의 웹 서버가 지정되어 의도하지 않은 PHP 코드가 삽입되는 경우가 있으므로, "http"가 GET 파라미터로 전달되었을 경우를 공격으로 탐지한다. 또한, 웹 서버 내의 디렉터리를 탐색하여 정보를 탈취하거나, 파일 확장자를 무효화시키기 위한 문자가 사용되므로, "/"와 "%00" 역시 공격에 해당한다고 판단한다.

WAD는 SQL 삽입 탐지 시 그림 12와 같이 하늘색, RFI 탐지 시 노란색으로 출력한다.

4.2.1.3 웹쉘 업로드 탐지 구현

웹쉘을 웹 서버에 업로드하고 공격한 후, 로그를 분석하면, 전송 바이트 수가 급격히 증가한다. 이 점을 이용하여 이전 로그 행과 현재 로그 행의 전송 바이트 차이가 클 때, 웹쉘이 실행되고 있다고 판단한다.

상세 알고리즘은 그림 13과 같다. 메인 함수에서 접근 로그 파일에 변화가 생기면 로그를 한 줄씩 읽어 들이는 스레드를 생성한다. 이때 읽은 로그를 파싱하여 웹쉘 업로드 공격을 판별한다. 현재 전송 바이트와 직전 전송 바이트의 차의 절댓값이 10000보다 클 경우 웹쉘 업로드 공격으로 판별하고 그렇지 않을 경우 정상 접근으로 판별한다.

WAD 프로그램은 웹쉘 업로드 공격을 탐지할 경우 그림 14와 같이 로그를 분홍색으로 출력한다.

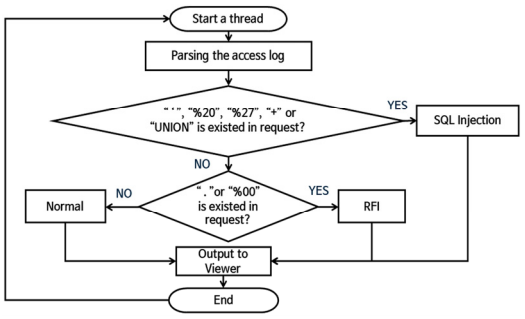

그림 11 SQL 삽입/RFI 탐지 알고리즘
Fig. 11 The algorithm of detection SQL Injection and RFI

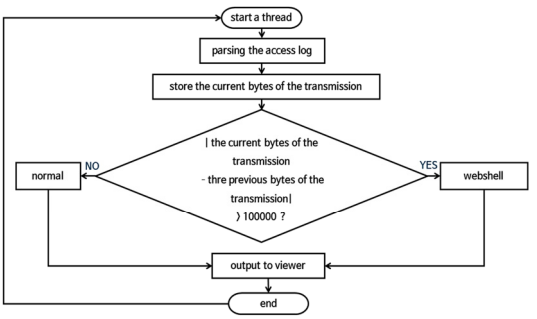

그림 13 웹쉘 업로드 탐지 알고리즘
Fig. 13 The algorithm of detection Webshell Upload

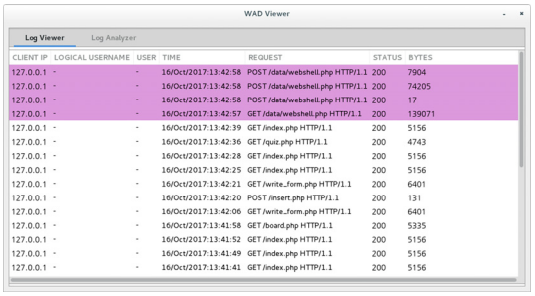

그림 14 웹쉘 업로드 탐지 화면

Fig. 14 The viewer of detection WebShell Upload

4.2.2 로그 통계

WAD의 Log Analyzer 탭을 선택하면 그림 15와 같이 로그 통계를 볼 수 있다. 각 IP가 몇 번 접속 하였는지를 그래프로 확인할 수 있다. DDoS, SQL 삽입, RFI, 웹쉘 업로드 공격이 몇 번 탐지되었는지 알 수 있다. 또 총 몇 번의 로그가 들어왔는지도 알 수 있다.

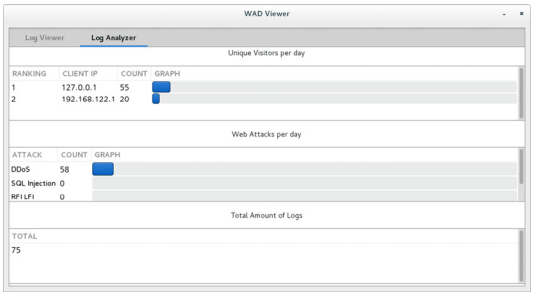

그림 15 로그 통계 출력 화면

Fig. 15 Log Statistics Output Screen

4.2.3 IP 차단

WAD 프로그램은 그림 16과 같이, 뷰어 상에서 더블 클릭을 할 경우 해당 IP를 차단할 수 있다.

로그를 더블 클릭하였을 때 해당 IP가 이미 차단되었

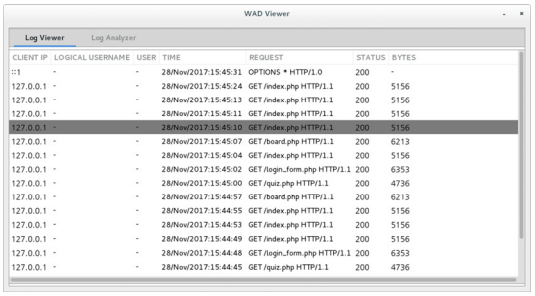

그림 16 차단된 IP 주소

Fig. 16 Blocked IP address

```
if(check_block_ip(name)   == 1) {
    sprintf(buf,   "iptables -D INPUT -s %s -j DROP", name);
    system(buf);

    gtk_list_store_set(store,   &iter_selected,
                       COLOR, "White",-1);
}
else{
    sprintf(buf, "iptables -A   INPUT -s %s -j DROP", name);
    system(buf);

    gtk_list_store_set(store, &iter_selected,
                       COLOR,"Grey",-1);
}
```

그림 17 IP 차단 과정

Fig. 17: IP blocking procedure

는지 확인한다. 차단되지 않은 ip일 경우 “iptables -A INPUT -s 해당IP -j DROP” 시스템 명령을 통해 IP를 차단하고 로그를 회색으로 바꾼다. 이미 차단된 IP일 경우 “iptables -D INPUT -s 해당IP -j DROP” 시스템 명령을 통해 IP 차단을 해제하고 그림 17과 같이 로그를 흰색으로 바꾼다.

5. 성능 평가

WAD의 성능 평가를 위해 현재 리눅스 상에서 사용되고 있는 유사한 프로그램과 비교하고자 하였으나, 타 웹 서버 공격 탐지 프로그램은 패킷 분석을 통해 이루어지므로 WAD와 비교하기에 부적합하다고 판단하여 자체적으로 성능 평가를 진행하였다.

5.1 DDoS

서버에 과부하를 일으키는 다량의 접속을 하여 다른 사용자의 서비스 이용을 막는 DDoS 공격 발생 도구를 제작해서 성능 평가를 진행하였다. 그림 18은 공격 대상이 되는 홈페이지에 0.8초의 간격을 두고 30번 접속하게 설정한 모습이다.

그림 18 DDoS 공격 도구

Fig. 18 DDoS attack program

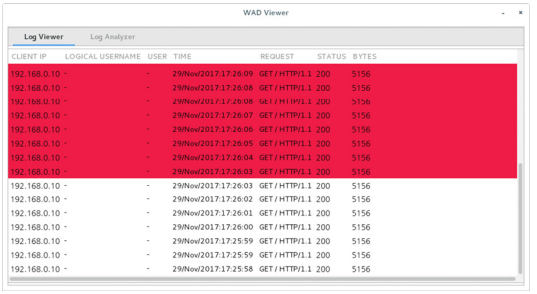

그림 19 DDoS 탐지 화면

Fig. 19 DDoS detection viewer screen

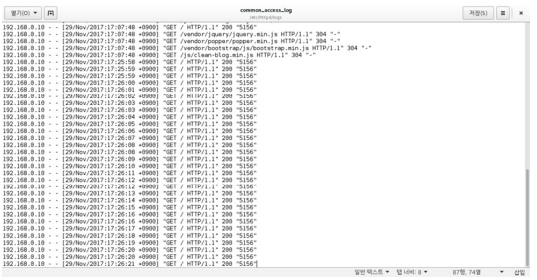

그림 20 접속 로그

Fig. 20 Common access log

뷰어에 나타난 그림 19와 실제 common_access_log에 기록된 로그 그림 20과 비교했을 때, 30번 접속한 기록이 모두 뷰어에 나타나 있다는 것을 확인할 수 있으며, DDoS 공격이 들어왔다고 인식한 시점은 최초 접속 이후 약 5초 후인 8번째 접속이다.

5.2 SQL 삽입

SQL 삽입 공격은 잘못된 sql문을 입력하는 순간 감지해야 하므로 시간 측정보다 탐지 여부에 중심을 두고 결과를 확인한다.

그림 21과 같이 아이디와 패스워드에 sql을 무력화시키는 'or'1을 입력한 경우, WAD의 뷰어에서 REQUEST에 "GET/login.php?id=%27or%271&pass=%27%271 HTTP/1.1"이라는 구문이 들어온 것을 볼 수 있다. WAD에 내장된 SQL 삽입 공격 알고리즘에 따라 %27을 탐지

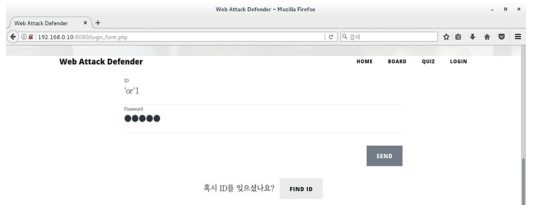

그림 21 비정상적 sql 삽입 시도

Fig. 21 Abnormal sql injection attempts

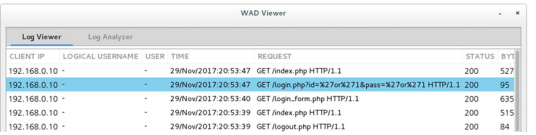

그림 22 sql 삽입 탐지 화면

Fig. 22 SQL injection detect viewer screen

하여 그림 22와 같이 파란색으로 바꿔 공격이 이루어졌다고 알려준다.

5.3 RFI/LFI

RFI/LFI 공격의 경우 탐지 원리가 SQL 삽입 공격과 유사하므로 비정상적인 php 구문이 들어갔는지 확인한다.

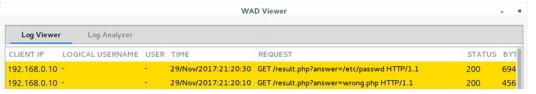

그림 23 비정상적인 php 구문 삽입 탐지

Fig. 23 Abnormal php syntax insertion detection

프로그램에 넣었던 알고리즘에 따르면 .php와 같이 확장자를 탐지하기 위한 "."과 리눅스 프로그램상에서 우회기법을 사용하기 위한 "%00"라는 특수문자를 탐지하도록 했다. 이에 따라 .이 들어간 리퀘스트 "/request.php?answer=/etc/passwd"를 파싱하여 RFI 공격이라고 인식, 그림 23과 같이 화면에 노란 줄로 출력하였다.

5.4 웹쉘 업로드

웹쉘 업로드는 서버에 올라오는 파일의 크기가 기준보다 크면 공격이라고 인식하도록 하였다. 그림 24에서처럼 REQUEST에 'webshell.php'라는 구문이 들어갔을 때 탐지하도록 알고리즘을 만들지 않은 이유는 공격자가 이 이외의 다른 이름으로 웹쉘을 올렸을 시 탐지하지 못할 수 있기 때문이다. 따라서 웹쉘 특성상 큰 용량을 가지고 있기 때문에 bytes 수로 공격을 판별하도록 하였다.

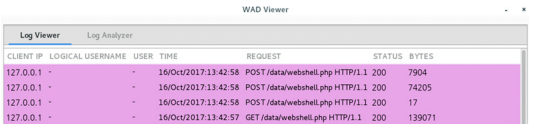

그림 24 웹쉘 탐지 화면

Fig. 24 Webshell Upload Detect Viewer

6. 결론 및 향후 연구

본 논문에서는 웹 서버로 들어오는 공격을 탐지하여 개인적, 사회적 피해를 최소화하는 것을 목표로 하였다.

WAD 프로그램은 리눅스 CentOS 환경에서 DDoS,

SQL injection, RFI, Webshell upload 공격이 이루어졌을 때 접근 로그에서 나타나는 패턴을 찾아내는 알고리즘을 구현하였다. 위의 알고리즘에 따라 현재까지 알려진 공격들의 패턴을 분석하여 웹 서버를 공격자로부터 높은 확률로 막을 수 있다.

WAD 프로그램에서는 이러한 분석을 토대로 웹 서버 관리자를 위한 접근 로그 뷰어, 접근 로그 통계 및 IP 차단 기능을 제공한다.

향후 연구에서는 위에서 제시한 4개의 공격 외에도 추가적인 공격에 대한 패턴을 찾아내어 웹 서버 보안을 강화하고자 한다. 특히 계속해서 바뀌는 공격자들의 침입 방법에 대해 지속적인 연구가 이루어져야 하고, 소수의 사례를 기반으로 공격 패턴을 설정한 것이므로 이에 대한 보완점을 추가할 예정이다.

References

[1] K. Kim, (2017, Sep.). "5 reasons why security incidents do not decrease," [Online], Available: http://www.boannews.com/media/view.asp?idx=56923&kind=

[2] goAccess, [Online]. Available: https://goaccess.io/

[3] Visitors, [Online]. Available: http://www.hping.org/visitors/

[4] WebAlizer, [Online]. Available: http://www.webalizer.org/

[5] *Requirement for User Authentication of Mobile Content or Application*, TTAK.KO-06.0358, 2013.

[6] *Security Requirements for the Anti-DDoS Appliances*, TTAK.KO-12.0262, 2014.

[7] IAB. (1989, Jan.) *Ethics and the Internet* [Online]. Available: http://www.faqs.org/rfcs/rfc1087.html

[8] *Information Security Terminology*, TTAK.KO-12.0002/R3, 2013

[9] J. Hwang, (2017, Jun.), "'yeogieottae', hacking group was caught, 0.99 million people 3.41 million information outflows," [Online], Available: http://www.nextdaily.co.kr/news/article.html?id=20170602800004

[10] H. Maeng and H. Park, (2015, Sep.). "ppomppu, 1.9 million member managed by only three people. Community site hack vulnerability," [Online], Available: http://news1.kr/articles/?2419964

[11] S. Ueno and H. Yang, Vulnerability diagnosis start guide for web secure manager, 2017, p. 96. (in Korean)

[12] T. Kim, (2013, Feb.). "ilhwa website, Abuse by malicious code infections!," [Online], Available: http://www.boannews.com/media/view.asp?idx=34742&kind=1

[13] webshell attack, [Online]. Available: http://terms.tta.or.kr/dictionary/dictionaryView.do?word_seq=060063-2

[14] T. Kim, (2012, May). "EBS hacking, Estimated Webshell using homepage bulletin vulnerability," [Online], Available: http://www.boannews.com/media/view.asp?idx=31349

[15] GTK+3 API, [Online]. Available: https://developer.gnome.org/gtk3/stable/gtk.html

[16] T. Kim et al., "DoS/DDoS attacks Detection Algorithm and System using Packet Counting," *Journal of the Korea Society for Simulation*, Vol. 19, No. 4, pp. 151-159, 2010. (in Korean)

박 재 연

2014년~2018년 숙명여자대학교 ICT융합공학부 IT공학전공 공학사. 관심분야는 리눅스 서버 보안, 시스템 소프트웨어

이 송 연

2014년~2018년 숙명여자대학교 ICT융합공학부 IT공학전공 공학사. 관심분야는 리눅스 서버 보안, 시스템 소프트웨어

이 하 은

2014년~2018년 숙명여자대학교 ICT융합공학부 IT공학전공 공학사. 관심분야는 리눅스 서버 보안, 시스템 소프트웨어

이 종 우

1990년 서울대학교 컴퓨터공학과(학사) 1992년 서울대학교 컴퓨터공학과 대학원(석사). 1996년 서울대학교 컴퓨터공학과 대학원(박사). 1996년~1998년 현대전자 정보시스템사업본부 과장. 1999년~1999년 현대정보기술 책임연구원. 1999년~2002년 한림대학교 정보통신공학부 조교수. 2002년~2003년 광운대학교 컴퓨터공학부 조교수. 2003년~2004년 아이닉스 소프트(주) 개발이사. 2004년~ 숙명여자대학교 멀티미디어과학과 교수. 2008년 뉴욕주립대 스토니브룩 Research Scholar. 2012년~2013년 숙명여자대학교 지식정보처장 2012년~2018년 NAVER 주식회사 사외이사. 2014년~ 한국정보과학회 컴퓨팅의실제 논문지 편집위원장. 2015년 뉴욕시립대 John Jay College Visiting Scholar. 관심분야는 Mobile System Software, Storage Systems, Computational Finance, Cluster Computing, Parallel and Distributed Operating Systems, and Embedded System Software