경량 로그 분석 소프트웨어 개발

이시헌[1], 김지윤[1], 김서현[1], 황지온[1*]
[1]협성대학교 소프트웨어공학과
*e-mail : zhwang@uhs.ac.kr

A Lightweight Software Solution for Log Analysis

Siheon Lee[1], KIM JIYun[1], Kim Seo Hyeon[1] and Hwang Zion[1*]
[1]Hyupsung University, Department of Software Engineering

Abstract

As cyber threats become more advanced and AI technology develops in parallel, the establishment of AI-powered proactive attack detection systems is increasing. However, the adoption of such systems requires excessively large computing resources and costs. Meanwhile, many GUI-based log analysis software either cannot guarantee real-time analysis or fail to provide backup functions for web logs. In addition, the installation process is often complex, and the interface is frequently not user-friendly. Therefore, this study aims to develop a "lightweight, general-purpose log interpretation tool that can be easily installed and used." This refers to a web-based log analysis tool that consumes minimal computing resources, can be operated at low cost, guarantees real-time analysis, provides secure backup functions for web logs, has a simple installation process, and offers a user-friendly interface. Specifically, the goal is achieved by constructing a system based on Docker containers. This software provides an interface for users implemented with Next.js and React, and is designed to be connected to the host system through a reverse proxy. The service communicates internally with an API prepared in an isolated PHP-Apache container. The API provides the core functions of this system (detecting web attacks through log analysis, backing up logs, generating AI-based security reports from logs, etc.). The results of this study may be applied at low cost by small businesses or individuals, thereby minimizing potential security incidents at both the personal and societal levels. In the future, the set of regular expressions for detecting attack signs will be improved, and studies on log analysis using language models such as BERT will be incorporated to enhance detection accuracy.

I. 서론

고도화되는 사이버 위협과 AI 기술 발달이 맞물리며, AI를 활용한 능동 공격 감지 시스템 구축이 늘어나고 있다. 하지만 이러한 시스템의 도입은 너무 큰 컴퓨팅 자원과 비용을 요구한다. 한편, 많은 GUI 기반 로그 분석 소프트웨어는 실시간 분석을 보장하지 못하거나, 웹 로그에 대한 백업 기능을 제공하지 못한다. 또한 설치 과정이 복잡하고 인터페이스가 사용자 친화적이지 않은 경우가 많다.

본 연구는, '쉽게 설치하여 사용 가능한 경량화 된 범용 로그 해석 도구' 개발을 목적으로 한다. 이는,

적은 컴퓨팅 자원을 소비하며, 적은 비용으로 운용할 수 있고, 실시간 분석을 보장하며, 웹 로그에 대한 안전한 백업 기능을 제공하며, 설치 과정이 간단하고, 인터페이스가 사용자 친화적인 웹 기반의 로그 분석 도구를 말한다.

II. 로그 분석 소프트웨어 현황

네트워크 차원에서 패킷을 분석하는 소프트웨어와 하드웨어는 비용 효율적이지 않다. 본 연구에서 추구하는 시간과 비용 효율성을 충족하기 위해서는 7계층에서 동작하는 로그 분석 소프트웨어를 구현해야 한다. 그러므로 기존 로그 분석 소프트웨어 또한 7계층에서 동작하는 것 만을 선별하여 제시한다.

한편, 본 연구의 목표에 '적은 비용'이 포함되므로 오픈소스 소프트웨어가 아닌 경우는 따로 소개하지 않는다.

2.1 ModSecurity

ModSecurity는 OWASP[1] Core Rule Set(CRS)을 적용하여 웹 트래픽을 실제로 검사하는 웹 어플리케이션 방화벽(WAF)이다.

이는 정적 웹서버와 결합하여 작동하며, 대표적으로 다음과 같은 유형의 공격을 방어한다.[1]

- Injection Attacks
- Cross-Site Scripting, XSS
- File Inclusion
- Protocol-level Attacks
- Session Management
- Information Leakage
- Automated Threats

WAF가 활성화된 정적 웹서버는, 저장된 Rule Set을 통해 모든 요청이 정상인지 검사한다. 만약 접근이 공격으로 판단되면, 403 Forbidden 같은 오류를 반환하게 된다.

2.2 Web Attack Defender (WAD)

웹 서버의 로그를 실시간 분석하는 소프트웨어 개발 연구가 있다.[2]

해당 연구에서 개발한 WAD는, 애플리케이션 계층에서 동작하며, 실시간으로 로그를 검사하여 공격을 감지한다.

2.3. Wazuh

Wazuh는 대표적인 XDR[2] 과 HIDS[3] 기능을 제공하는 오픈소스 솔루션이다.[3]

데이터를 수집하여 공격을 감지하며, 감지 결과를 시각화 하여 유려한 인터페이스로 제공한다. 구체적으로는, 미리 준비된 규칙에 기반하여 이벤트를 파싱하고, 침해 또는 정책 위반 여부를 감지하여, 그 결과를 시각화 하여 제공한다.

III. 설계 및 구현

3.1. 컨테이너 기반 로그 복제 및 전처리

전체 시스템의 요구사항을 충족하기 위하여, 다수의 컨테이너를 구성한다. 구체적으로는 다음과 같다.

container name	docker image name
nextjs	Node:current
php-apache	php:8.1-apache
ollama	ollama/ollama
mysql	mysql:8.0

표 1. 컨테이너 이미지 목록

위의 컨테이너는 다음과 같은 구조로 계층화 되어 하나처럼 동작한다.

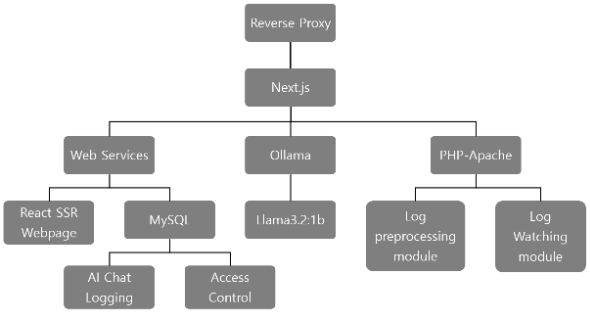

그림 1. 컨테이너 계층 구조.

nextjs 컨테이너만이 외부로 노출되어 있으며, 사용자를 위한 웹 페이지를 SSR[4]로 구성한다. 이때 php-apache 컨테이너에는 nextjs 컨테이너가 사용 가능한 REST API들이 준비되어 있다. nextjs 컨테이너는 해당 API들을 호출하여 사용자를 위한 모든 연산 결과를 제공받는다. 즉, nextjs 컨테이너는 php-apache 컨테이너에 격리된 로그에 직접 접근할

[1] OWASP: Open Worldwide Application Security Project

[2] XDR: Extended Detection and Response

[3] HIDS: Host-based Intrusion Detection System

[4] SSR: Server Side Rendering

수 없으며, 로그를 사용한 모든 연산을 직접 수행할 수 없다.

한편, 호스트 시스템의 로그는 일반적으로 잘 알려진 위치에 저장되기에 언제나 변조 위험에 처해 있다. 그러므로 별도의 독립된 공간에 로그를 복제하여 안전하게 보관할 필요가 있다. 또한 많은 정적 서버 소프트웨어는 주기적으로 기존 로그를 압축하고, 새 로그 파일을 생성한다. 이로 인하여 전체 로그를 대상으로 한 키워드 검색과 공격 탐지의 복잡성이 증가한다.

이에 본 연구는 사용자가 입력한 로그 경로를 감시하는 백그라운드 프로세스를 준비하고, 로그 파일의 크기 증가가 일어날 때 마다, 추가된 로그를 격리된 컨테이너(php-apahce)로 복사하여 보존한다. 이때 php-apache 내부에서는 모든 로그를 하나의 단일 파일로 병합하며, 로그를 계속 누적하며, nextjs를 제외한 모든 접근을 차단한다.

ollama는 로컬 LLM을 동작시키는 도구이다. 해당 도구는 미리 준비된 docker image를 제공한다. 내부에서 Llama3.2:1b 모델을 동작 시키며, nextjs의 접근만을 허용한다.

mysql 또한 docker image가 제공된다. 후술할 '로그를 컨텍스트로 갖는 Chat bot' 기능 제공을 위한 RDBMS로서, nextjs의 접근만을 허용한다.

3.2. 파일구조

항목 3.1. 에서 설명한 컨테이너 구조는 다음과 같은 docker-compose.yml 파일을 통해 자동적으로 설치된다.

```
services:
  nextjs:
    image: node:current
    container_name: nextjs
    working_dir: /app
    volumes:
      - ./frontend:/app
    ports:
      - "127.0.0.1:8445:3000"
    command: >
      /bin/sh -c "apt-get update &&
      apt-get install -y python3 make g++ &&
      npm install &&
      npm run dev"
    environment:
      NODE_ENV: development
    networks:
      - loganalyzer_net

  php-apache:
    image: php:8.1-apache
    container_name: php-apache
    volumes:
      - ./backend:/var/www/html
      - /var/log/apache2:/webLogs:ro
      - ./readLog:/readLog
      - ./apache:/etc/apache2/sites-available
    command: >
      /bin/sh -c "apt-get update &&
      apt-get install -y curl procps libsqlite3-dev &&
      docker-php-ext-install pdo_sqlite &&
      mkdir -p /var/www/html/LOG &&
      php /readLog/init.php &
      apache2-foreground"
    expose:
      - "80"
    depends_on:
      - nextjs
      - ollama
    networks:
      - loganalyzer_net

  ollama:
    image: ollama/ollama
    container_name: ollama
    restart: unless-stopped
    ports:
      - "127.0.0.1:11434:11434"
    volumes:
      - ollama_data:/root/.ollama
    entrypoint: ["/bin/sh", "-c"]
    command:
      - |
        ollama serve &
        sleep 2
        ollama pull llama3.2:1b
        wait
    networks:
      - loganalyzer_net

  mysql:
    image: mysql:8.0
    container_name: mysql
    restart: unless-stopped
    env_file:
      - .env
    environment:
      MYSQL_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}
      MYSQL_DATABASE: ${MYSQL_DATABASE}
      MYSQL_USER: ${MYSQL_USER}
      MYSQL_PASSWORD: ${MYSQL_PASSWORD}
    volumes:
      - mysql_data:/var/lib/mysql
      - ./mysql/init:/docker-entrypoint-initdb.d
    expose:
      - "3306"
    networks:
      - loganalyzer_net

networks:
  loganalyzer_net:
    driver: bridge

volumes:
  ollama_data:
    driver: local
  mysql_data:
    driver: local
```

그림 2. docker-compose.yml

위의 docker-compose.yml 파일을 실행할 수 있는 준비된 소스코드가 GitHub에 게시되어 있으며, 해당 소스코드의 파일구조는 다음과 같다.

```
LogAnalyzer
├── apache
├── backend
│   ├── LOG
│   └── alertDisplayPage
├── frontend
│   ├── app
│   ├── node_modules
│   └── public
├── mysql
│   └── init
├── readLog
└── scheduler
```

그림 3. docker-compose.yml

backend 경로는 php-apache 컨테이너로 로드되는 정적 파일이 저장되며, frontend 경로는 nextjs

컨테이너에서 동작하는 React와 Next.js 기반의 SSR을 위한 정적 파일이 저장된다.

docker-compoes.yml 파일이 실행되는 즉시 readLog 경로의 php 소스코드가 CLI로 실행되며, 호스트 시스템의 로그 용량 변화를 주기적으로 감지하여 php-apache 컨테이너의 LOG 경로에 병합된다. (/readLog/init.php)

한편, /backend/alertDisplayPage 경로에는 실시간 로그 검사 결과를 출력하는 웹페이지가 존재하며, nextjs 컨테이너의 프록시를 통해 서비스된다.

3.3. 정규식 기반 공격 감지

항목II에서 제시한 공격 감지 시스템의 경우, 공통된 단점을 가진다. 오픈소스로 제공되는 많은 로그 분석 도구는 사용자의 높은 역량과 전문성을 요구한다. 충분한 성능을 위해서 Rule Set의 조정과 보안 정책 수립이 필수적이기 때문이다. 즉, 사용자의 역량 요구와 비용은 트레이드-오프 관계에 있다는 것이다. 이에 본 연구는 공격 감지를 위한 시그니처 목록을 최대한 쉽게 관리할 수 있도록 설계하였다.

정규식 세트를 단일 CSV파일로 php-apache 컨테이너 내부에 보관하며, 사용자는 해당 파일을 수정하는 것으로 모든 로그를 검사할 수 있다.

해당 CSV 파일에 준비된 정규식은 Attack Type, Attack Details, Search Regex 의 세 속성으로 정의되어 있으며, Attack Type를 기반으로, 사용자에게 어떤 유형의 공격이 의심되는지를 출력한다. 현재 정규식 세트는 다음 유형의 공격에 대응한다.

- SQL injection
- XSS (Cross-site sctipting)
- 디렉토리 탐색
- 명령어 삽입
- 파일 포함
- 웹 취약점 스캐너 시그니처

감지 결과는 다음과 같이 출력된다.

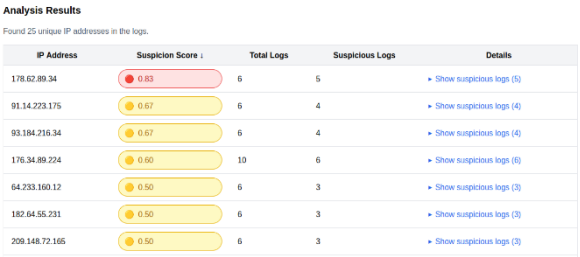

그림 4. 정규식 기반 공격 감지 결과

IP로 그룹화 된 로그에 대하여, 정규식 감지 결과

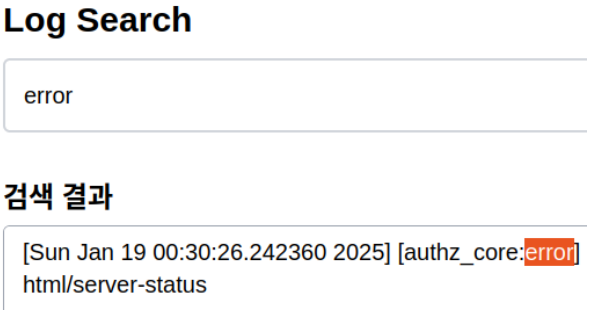

그림 7. 검색 페이지

의심스러운 로그로 분류된 비율이 80% 이상인 경우 붉은 색으로, 30% 이상인 경우 노란색으로, 그 이하인 경우 녹색으로 표시한다.

이때 개별 IP에 대하여, 어떤 로그가 의심으로 분류되었고, 왜 의심스러운지 또한 확인 가능하다.

그림 5. 의심 로그 분류 상세

3.4. Local LLM을 사용한 보안 보고서 작성

전체 로그를 LLM에 입력하여 보안 보고서 작성을 요청할 수 있다. netxjs 컨테이너에서 php-apache 에 로그를 요청하고, 전달받은 로그를 ollama 컨테이너로 전송하여 보안보고서를 얻어낸다.

nextjs 컨테이너에서는 php-apahce에 전체 로그를 요청하거나, 상태코드별로 나뉘어 전처리 된 로그를 요청하여 보안 보고서를 출력한다.

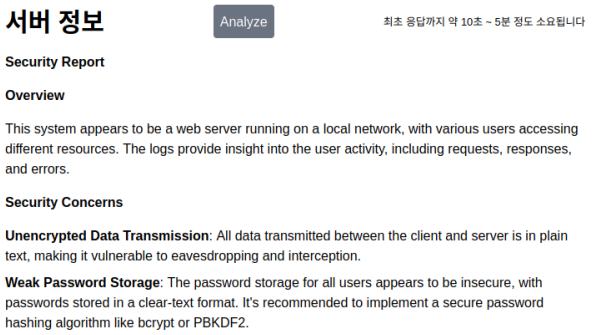

그림 6. 전체 로그를 통한 보안 보고서

3.5. 검색과 AI Chat bot

nextjs에서 php-apache와 ollama에 준비된 API를 호출하여, 전체 로그를 대상으로 하는 단순 검색 기능과 AI 검색 기능을 지원한다.

3.6. 실시간 경고 페이지

/readLog 경로의 '호스트 시스템의 로그를 수집하는 프로세스'가 새로운 로그를 감지할 때 마다, /backend/alertDisplayPage 내부에 존재하는 API로 로그를 복제 및 전송하여, 정규식 감지를 수행하고 그 결과를 SQLite3 DB에 저장한다.

해당 경로에 존재하는 웹페이지가 DB의 정보를 읽어서 신규 로그에 대한 공격 여부를 출력한다. (공격이 의심되는 경우 붉은 박스로 감지된 패턴 내용을 출력한다.)

그림 8. 실시간 경고 페이지

Ⅳ. 결론 및 향후 연구 방향

경량의 로컬 LLM과 정규식에 기반한 로그 분석을 통해, 웹 공격을 능동적으로 감지하고 시각화 하여 제시하는 소프트웨어 개발이 완료되었다.

소규모 사업체 혹은 개인이 저비용으로 적용하여 사용 가능한 경량 로그 분석 툴의 배포를 통하여, 개인과 사회 차원에서 발생 가능한 보안사고를 최소화할 수 있을 것이다.

공격 징후 탐지를 위한 다양한 추가 연구를 진행할 예정이다. 정규식 세트의 정확도를 개선하고, 추후 BERT 등의 언어모델을 통한 로그 분석 연구[4]를 수용하여 로그 분류의 정확도를 개선할 예정이다. 한편, 다른 논문[5]에서 제안하는 DDoS 감지 체계를 추가할 예정이다.

로그의 규모가 늘어나기 때문에 발생하는, 보안 보고서와 AI Chat bot의 할루시네이션 문제를 해결하는 연구를 진행할 예정이다. 로그를 작은 용량으로 분해하여 여러 번의 LLM 입력을 수행하거나, 특정 기준으로 필터링 된 일부 로그를 사용하여 입력 컨텍스트 크기를 줄이는 등의 방법을 적용해 볼 예정이다.

참고문헌

[1] https://docs.varnish-software.com/varnish-waf/owasp-crs/

[2] 박재연 외, "리눅스 아파치 웹 서버 실시간 로그 분석을 통한 공격 탐지 프로그램 개발", 정보과학회 컴퓨팅의 실제 논문지, Vol.24, no.4, pp. 190-197, 2018.

[3] https://wazuh.com/

[4] 조영복 외, "웹 방화벽 로그 분석을 통한 공격 분류: AutoML, CNN, RNN, ALBERT", Journal of The Korea Institute of Information Security & Cryptology, Vol.34, No.4, Aug, 2024.

[5] 이석우 외, "웹 모니터링 기반 암호화 웹트래픽 공격 탐지 시스템", 한국정보통신학회논문지, Vol.25, No.3, pp. 449-455, Mar, 2021.