리눅스 CCE 취약점 진단 및 시각화 시스템 개발

김한선, 우은지, 이은경, 정호심
서울여자대학교 정보보호학과

hansun33@naver.com, eunji7945@swu.ac.kr, 71126eun@swu.ac.kr, grtff@swu.ac.kr

Development of Linux CCE Vulnerability Diagnosis and Visualization System

Han-sun Kim, Eun-Ji Woo, Eun-Kyung Lee, Ho-Sim Jeong
Dept. of Information Security, Seoul-Women's University

요 약

최근 클라우드 이용자 수의 급증에 따라 클라우드 상의 리눅스 환경 보안의 중요성이 대두되고 있다. 또한 클라우드 상의 리눅스 환경의 CCE(Common Configuration Enumeration) 취약점 보안 설정은 중요성의 비해 사용자들이 놓치는 경우가 많은 실정이다. 하지만 앞서 진행된 연구에는 리눅스 서버에 대한 보안 관리 방안으로 스크립트를 통한 진단방식 만을 제공하였다. 본 논문에서는 진단 쉘 스크립트 사용 및 진단 항목별 실시간 통계 분석, 시각화를 활용해 리눅스 환경을 향한 사이버 공격을 대비한다. 이후 보안 담당자들이 클라우드 취약점을 점검하는 데 유용한 도구가 될 것으로 사료된다.

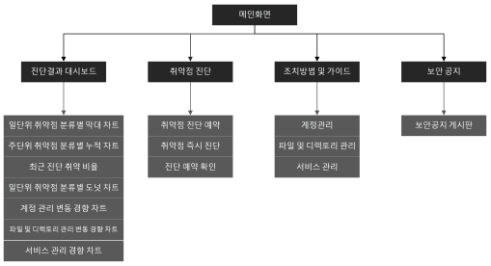
1. 서론

클라우드 이용자 수의 급증에 따라 클라우드 상의 리눅스 환경 보안의 중요성이 대두되고 있다. 클라우드 상의 리눅스 환경의 CCE(Common Configuration Enumeration) 취약점 보안 설정은 다수의 항목, 경우에 따라 다른 설정 파일 등의 이유로 중요성에 비해 사용자들이 놓치는 경우가 많다. 따라서 해당 취약점들을 자동으로 점검해주고 결과를 제공해주는 서비스가 필요하다. 더불어 다양한 환경에서의 일괄 적용이 힘들었던 앞선 perl 스크립트 사용의 한계점 극복을 위해 본 논문에서는 클라우드 상의 리눅스 취약점(cce) 진단 쉘 스크립트를 제공하였다.

또한 점검 결과를 항목별로 실시간 통계 분석함으로써 클라우드 상 리눅스 환경 사이버 공격을 주기적으로 점검 및 대응할 수 있는 서비스를 구현하였다.

2. 진단 시스템 및 결과 시각화 기획

현재 KISA 에서 발간한 CSAP CCE 취약점 가이드[3]를 살펴보면 자산 별 진단 기준 및 조치 방안을 제공하고 있다. 하지만 각 자산과 항목들을 일괄적으로 진단하고 결과를 도출하기 위해서는 별도의 시스템 구축이 요구된다. 본 논문에서는 리눅스 OS 를 진단 대상으로 하여 진단 스크립트와 MariaDB, 소켓 프로그래밍을 활용한 OS 진단 시스템을 설계하였으며, 사용자의 편의를 위하여 진단 예약, 진단 결과 시각화 기능이 구성된 웹사이트를 구축하였다. [그림 1]과 같이 기간, 항목에 따른 차트를 제작하였다. 취약점 진단 기능의 경우 서버의 DB 와 연결되어 설정한 시간, 대상, 항목에 따라 진단 스크립트가 실행되고 소켓 코드를 통해 전송된 결과 로그는 각 대시보드 형태에 따라 계산되어 시각화하도록 구성하였다.



(그림 1) 웹 메뉴 구성도

3. 진단 시스템 및 결과 시각화 구현

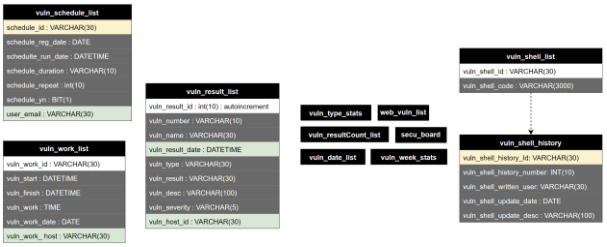
3.1. CCE 취약점 진단 과정

```
vuln_number="U14"
vuln_name="SUID, SGID, Sticky bit 설정 파일 점검"
vuln_result_date="$(date +%Y)-$(date +%m)-$(date +%d) $(date +%H):$(date +%M):$(date +%S)"
vuln_type="계정 관리"
vuln_severity="3"
vuln_result="취약하지 않음"
vuln_desc="해당 파일 또는 서비스가 없습니다."
vuln_flag1=`find  / -user root -type f \\( -perm -4000 -o -perm -2000 \\) -exec ls -lg {} \; 2>&1 >/dev/null`
vuln_host_id=$(hostname -I)

if [ -z "$vuln_flag1" ]; then
   vuln_desc="suid, sgid에 대한 설정이 부여되지 않은 파일 없음"
else
   vuln_result="취약함"
   vuln_desc="suid, sgid에 대한 설정이 부여되지 않은 파일 존재, 중요하지 않은 파일이라면 무시"
fi

echo "{\"vuln_number\":\"${vuln_number}\", \"vuln_name\":\"${vuln_name}\", \"vuln_result_dat
\"${vuln_result}\", \"vuln_desc\":\"${vuln_desc}\", \"vuln_host_id\":\"${vuln_host_id}\"}" >> res
```

(그림 2) 취약점 점검 스크립트

다음과 같이 CSAP CCE 취약점 가이드에 따라 Linux 의 계정 관리, 파일 및 디렉토리 관리, 서비스 관리, 패치 및 로그 관리에 대한 31 가지 항목의 취약점 점검 스크립트를 작성하였다. 해당 스크립트를 이용해 CCE 취약점을 진단하고, 진단 종료 후 결과는 DB 에 전송되어 웹 화면을 통해 제공된다.

3.2. DB 구조



(그림 3) ERD 설계도 일부

[그림 3]과 같이 DB 를 총 13 개의 테이블로 구성하여 클라우드 상의 리눅스에서 진단한 취약점 데이터 및 시스템 운영에 필요한 데이터를 저장했다. 저장된 취약점 진단 결과 데이터는 [그림 4]와 같이 총 7 개의 차트와 1 개의 도표로 웹 메인 화면에서 확인 가능하다. 차트는 오픈소스 라이브러리인 Chart.js 와 자바스크립트 개발 기법인 Ajax 방식을 활용해 구현했다.

(그림 4) 웹화면(대시보드, 예약 등록)

차트의 종류에는 진단 결과를 계정관리, 서비스 관리, 디렉토리 및 파일 관리 3 가지의 분류 별로 확인 가능한 막대 차트와 도넛 차트가 존재한다. 또한 주 단위로 취약점 변화 추이를 확인할 수 있는 누적 막대 차트 그리고 가장 최근 진단 취약점 개수 비율을 확인 가능한 파이 차트가 존재한다. 취약점 분류 별 변동 경향도 라인 그래프로 확인 가능하다.

3.3. 웹 화면 구성

취약점 진단 페이지로 이동하여 날짜, 진단 항목, 대상 기기, 검사 주기, 검사 횟수를 설정하고 예약 버튼을 눌러 예약할 수 있도록 했다. 따로 예약 확인 페이지가 존재하여 해당 페이지에서 예약 실행 시간, 진단 대상, 진단 항목, 예약 등록일 등을 확인할 수 있도록 구현했다.

보안 공지 메뉴는 보안 이슈에 관련한 글을 공유하는 게시판으로 사용자들에게 추가적인 정보를 제공할 수 있도록 했다. 사용자들은 보안 이슈에 대해 자유롭게 글을 작성하고 등록할 수 있도록 글 등록 버튼을 클릭하면 제목, 작성자, 내용을 입력할 수 있게 화면을 구성했다.

4. 결론

본 논문에서는 클라우드 상의 리눅스 취약점들을 자동으로 점검해주고 결과를 제공해주기 위하여 클라우드 리눅스 보안 점검(CCE) 서비스를 설계 및 구현하였다. 예약 진단 시스템으로 점검을 하고 그 점검 결과가 서버의 데이터베이스에 저장된다. 해당 데이터는 웹 페이지 상에서 통계 차트로 시각화 되어 제공된다.

이 서비스는 클라우드 상의 리눅스 취약점(CCE) 진단 쉘 스크립트를 제공하고, 진단 항목별로 실시간 통계 분석하였으며, 취약점 점검 시스템의 핵심 기능 위주의 서비스를 제공한다. 이를 통해 가격 측면과 실용성 측면에서 점검 비용이 부담스러운 영세한 사업자들에게 활용될 것으로 사료된다. 또한, 스크립트만 제공되는 오픈소스와 달리 진단시스템을 접목시켜 사용자에게 보다 많은 기능을 제공했다. 향후 클라이언트 진단 결과를 데이터 셋으로 활용하여 인공지능 기술을 접목한 사전 대비를 통해 취약점 피해를 최소화할 수 있는 리눅스 취약점 점검 서비스를 개발하고자 한다.

5. Acknowledgment

본 연구는 SW 중심대학추진사업단의 지원의 연구결과로 수행되었음 (2023)

참고문헌

[1] 이호수. "리눅스 서버 보안 취약점 개선을 위한 쉘 스크립트 구현." 국내석사학위논문 경북대학교, 2012. 대구

[2] 정길영, “리눅스 서버 보안 강화를 위한 취약점 분석 스크립트 구현”, 건국대학교 정보통신대학원, 2013 년

[3] KISA, 클라우드 취약점 점검 가이드-보안설정(CCE), 2020