SCAP의 분석과 OVAL 정의파일을 이용한 운영체제 취약점에 대한 분석

최민경[1], 배주영[2], 이윤권[3], 박현경[4], 김세은[5], 안효범[6]
공주대학교 정보통신공학부[1,2,3,5]
공주대학교 인공지능학과[4,6]
e-mail : cmg_125@naver.com[1], bjs338570@gmail.com[2], 2ygun@naver.com[3], gusrud5743@gmail.com[4], longlngs@naver.com[5], hbahn@kongju.ac.kr[6]

The analysis of SCAP and operating system vulnerabilities using OVAL definition

Mingyeong Choi[1], Juyoung Bae[2], Yungwon Lee[3], Hyunkyung Park[4], Se-eun Kim[5], Hyobeom Ahn[6]
[1,2,3,5]Division of Information & Communication Engineering, Kongju National University
[4,6]Division of Artificial Intelligence, Kongju National University

요 약

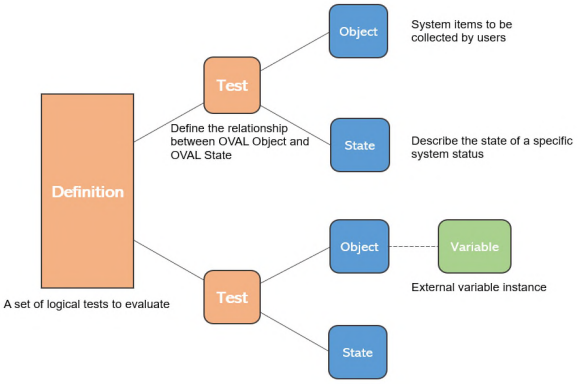
기업이나 기관에서는 시스템 관리를 위해 보안관리자들이 모니터링하며 환경 설정을 안전하게 설정하고 있는지, 취약성이 있는지 주기적으로 점검을 시행한다. 보안 취약점 점검의 자동화 및 표준화를 위해 미국의 NIST에서는 SCAP(Security Content Automation Protocol)을 개발하여 정부 기관에 배포하였다. SCAP은 보안 설정에 따른 점검항목을 OVAL(Open Vulnerability Assessment Language)로 작성하여 점검 대상 시스템의 보안 설정 정보와 비교하는 방식으로 동작한다. 따라서 OVAL로 점검항목을 작성하는 것이 보안 구성요소의 자동화 점검을 위한 핵심적인 부분이며 그 중 취약성을 진단하는데 필수적인 설정 파일을 검토해야 한다. 본 논문에서는 기업에서 많이 사용하는 레드햇(Red Hat) 리눅스의 보안 자동화 점검을 위한 OVAL Definition을 분석하고 취약점 정의를 바탕으로 운영체제 보안에 주로 사용되는 설정 파일들을 분류한다. 또한, 취약점과 연결하여 운영체제에서 발생할 수 있는 취약점에 대하여 제시한다. 본 연구를 통해 국내·외 취약성 점검 기준에 대해 OVAL 파일을 작성하는 데 도움을 줄 것으로 기대한다.

1. 서 론

현재 국가·공공기관에서는 기밀문서와 같이 중요한 정보를 효율적으로 관리하기 위해 보안 정보시스템에 많은 시간과 인력을 투자하고 있다. 많은 투자가 이루어지는 만큼 보안 취약점 분석 및 관리는 필수적인 요소가 되어 점검 자동화의 중요성도 잇달아 높아졌다. 이를 위해 미국의 NIST에서는 보안 취약점 점검의 자동화 및 표준화 프로토콜인 SCAP(Security Content Automation Protocol)을 개발하여 정부 기관에 배포하였다.[1] SCAP은 정보시스템 보안 설정 관리를 목표로 개발되었으며, 취약점 관리 및 평가, 위험성 측정, 보안 모니터링 등 업무를 자동화할 수 있도록 설계되었다. SCAP은 보안 설정에 따른 점검항목을 OVAL(Open Vulnerability Assessment Language)로 작성하여 점검 대상 시스템의 보안 설정 정보와 비교하는 방식으로 동작한다. OVAL로 점검항목을 작성하는 것이 보안 구성 요소의 자동화 점검을 위한 핵심적인 부분이며 주로 설정 파일을 중요시하게 보아야 한다. 본 논문에서는 기업에서 많이 사용하는 레드햇 엔터프라이즈 리눅스(Red Hat Enterprise Linux) 8을 기준으로 진행되며 2장에서 OVAL의 구조를 살펴보고 SCAP을 이용한 취약점 점검 방법을 간단히 소개한다. 3장에서는 OVAL Definition을 분석하고, 4장에서 운영체제 보안에 필수적인 설정 파일을 분류한 뒤 이에 따른 취약점을 살펴본다.

2. OVAL의 구조 및 취약점 점검 방법

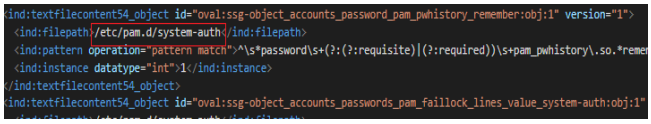
2.1 OVAL의 구조



(그림 1) OVAL Definition 구조

SCAP에서 운영체제 취약점을 구현하기 위한 핵심적인 요소는 OVAL이다. [2] OVAL은 컴퓨터 시스템의 상태를 평가하고, 취약성을 검증하는 언어로 구조와 구성 요소는 그림 1과 같다. XML 형식으로 구성되는 OVAL은 크게 Definitinon, Test, Object, State 4가지 요소로 표현할 수

있다. Definition은 평가할 논리 테스트의 집합으로 <criteria> 태그를 사용하여 테스트(Test)에서 평가할 시스템 객체(Object)를 식별하고, 해당 객체의 예상 상태(State)를 결합한다. Test는 검증 대상에 해당하는 Object와 State 정보를 포함하고 있으며 대상에 관한 서술 및 구체적인 연산을 설명한다. Object는 사용자가 수집할 item을 식별할 수 있도록 정의하며 시스템 id, version, filepath 등 다양한 자식 요소가 존재한다. State는 특정 시스템 상태를 기술하는 구체적인 값들로 구성되며 크고



(그림 2) Definition 내 filepath 예시

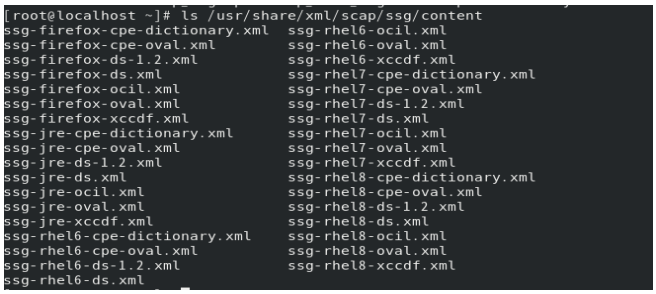
작은 연산에 대한 정보를 설명한다.

2.2 취약점 점검 방법

OVAL 취약점을 점검하기 위해선 Object의 filepath를 필수적으로 분석해야 한다. filepath는 OVAL Object에서 반드시 작성되어야 하는 자식 요소로 파일에 대한 절대 경로를 지정한다. 각 Object의 filepath를 확인하면 해당 취약점에서 어떤 설정 파일을 봐야 하는지 알 수 있다.

3. 레드햇의 OVAL Definition에 대한 분석

3.1 OVAL Definition 분석



(그림 3) SCAP 보안 컴플라이언스 파일

RHEL의 설정 파일을 분석하기 위해 OpenSCAP사의 OpenSCAP BASE 패키지를 설치할 때 기본으로 제공된 Definition인 ssg-rhel8-ds.xml을 사용했다.

해당 파일은 SCAP 보안 가이드 프로젝트에서 제공한 보안 규정 준수 파일로 /usr/share/xml/scap/ssg/content 디렉터리 내에 존재한다.

3.2 Definition에 대한 취약점 분류

(표 2) Definition 취약점 분류

Criteria	Description	Count
Account	password, root account, ID, login	48
File	Set owner and authority	102
system		
Network Service	Network and service management	90
Log Management	Patch and log management	154
Security Management	Managing overall security settings	155
System Service	Managing overall system settings	82
Device	Device settings such as usb, hard disk, etc.	50
Total		681

레드햇의 Definition에는 총 1,162개의 test_ref 항목이 존재하며 그 중 filepath를 확인할 수 있는 항목은 681항목이다. 본 논문에서 해당 definition을 계정관리, 파일 및 디렉터리 시스템, 네트워크 서비스 관리, 패치 및 로그 관리, 보안관리, 기타, 시스템 서비스, 장치 등 7개 항목으로 분류하였다. 이 기준에 대해서는 한국인터넷진흥원(KISA)에서 작성한 『기술적 취약점 분석 · 평가 방법 상세가이드』 문서의 취약점 분류를 차용하여 표 2와 같이 카테고리를 분류하였다. [3]

4. 레드햇의 OVAL Definition을 활용한 운영체제 취약점 분석

4.1 설정파일 분류

/etc/pam.d/system-auth	/etc/profile	/etc/vsftpd/vsftpd.conf	/etc/sysctl.conf	/etc/default/grub
/etc/pam.d/password-auth	/etc/httpd/conf/	/etc/firewalld/services	/boot/grub2/grub.cfg	/etc/bashrc
/etc/login.defs	/etc/shells	/etc/ssh/sshd_config	/etc/rsyslog.conf	/etc/fstab
/etc/passwd	/etc/shadow	/etc/sysconfig/network-scripts	/etc/named.conf	/dev/
/etc/group	/etc/gshadow	/etc/audit/audit.rules	/etc/yum.conf	/var/log/mount
/etc/securetty	/etc/systemd/system.conf	/usr/lib/systemd/system/auditd.service	/etc/xinetd.d/tftp	/etc/usbguard/rules.conf

(그림 4) 리눅스 설정 파일

그림 4는 Definition을 분석하여 filepath에서 수집한 파일의 목록을 나타낸다.[4] 수집한 파일 중 각 카테고리 내에서 가장 빈도수가 높은 파일을 표 3에 정리했으며 작성된 파일은 운영체제 보안에서 자주 사용되는 환경설정 파일이다.

(표 3) 주로 사용되는 RHEL 설정파일

Configuration File	Description	Counts

/etc/pam.d/system-auth	리눅스에서 사용하는 인증 모델인 PAM 모듈의 호출을 담당하는 설정파일로 라이브러리를 호출하여 인증 시 조건을 검사한다.	16
/etc/passwd	사용자가 로그인하는 데 필요한 파일로서 시스템의 계정 정보가 포함된다.	7
/etc/ssh/sshd_config	SSH 서버 설정 정보를 담은 파일	31
/etc/audit/audit.rules	시스템에서 발생된 이벤트에 관해 정보를 포함하는 로그를 생성하는 설정 파일	152
/etc/sysctl.conf	보안 설정과 관련된 시스템 커널의 속성을 읽고 설정하는 환경 설정 파일	109
/etc/default/grub	현 리눅스에서 사용하는 부트로더의 설정 파일	11
/dev	장치와 관련된 파일이 존재하는 설정 파일	5

4.2 Definition에 나타난 설정 파일과 취약점의 관계

앞서 말했듯이 표 3에 나타낸 설정 파일은 보안 규정에서 빈도수가 많은 파일이며 다른 파일보다 취약점이 많이 존재한다. 많은 취약점이 존재하기에 관련된 취약점이 발생했을 시 해당 설정 파일을 최우선으로 판별한다. 예를 들어 패치 및 로그와 관련된 취약점이 발생했을시 /etc/audit/audit.rules 파일을 우선으로 확인하면 높은 확률로 취약점에 관해 해결할 수 있을 것이다. 반대로 /etc/audit/audit.rules 파일에 대해 무수히 많은 취약점이 발견되었기 때문에 다른 설정 파일을 먼저 확인하여 취약점을 분석할 수 있을 것이다.

5. 결론

본 논문에서는 SCAP 보안 기준 표준인 OVAL Definition을 검증하고 설정 파일을 정리하여 그에 해당하는 취약점을 알아보았다. 먼저 OVAL Definition을 분석하여 사용된 설정 파일을 정리하고 취약점 기준에 따라 7개의 카테고리로 나누어 분류하였다. 분류한 파일 중 빈도수가 높은 설정 파일을 대상으로 어떠한 파일이 주로 쓰이는지 확인했고, 이러한 과정을 통해 본 논문에서 설명된 설정 파일이 OVAL 파일 작성의 기초 작업에 도움이 될 것으로 예상된다.

현재 국내 SCAP 및 취약점과 관련된 다양한 연구가 활발히 진행되고 있지 않아 필요한 자료를 수집하는 데 많은 어려움이 따른다. 따라서 SCAP 및 자동화 프로토콜의 자원에 관한 연구가 지속해서 진행된다면 정보시스템의 관리와 보안 준수에 대한 전망이 좋을 것으로 기대된다.

참 고 문 헌

[1] D. Waltermire, S. Quinn, H. Booth, K. Scarfone, and D. Prisaca, “The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.3,” NIST Special Publication 800-126. Revision 3, National Institute of Standards and Technology, February 2018, pp 5-8

[2] J. Baker, M. Hansbury, and D. Haynes, “The OVAL® Language Specification: Version 5.10 Revision 1,” The MITRE Corporation, August 2011, pp 12-21

[3] 과학기술정보통신부, 한국인터넷진흥원, “기술적 취약점 분석 · 평가 방법 상세 가이드”, 2021, pp 159-161

[4] 신동천, 김선광, “서버 보안기준 준수 평가를 위한 SCAP 적용 타당성 연구” Journal of Information Technology Applications & Management, 26(4), pp. 21-24, August 2019.