크롬 OS 의 보안 강화를 위한 취약점 분석 스크립트 구현

이슬기*, 유헌창**
*고려대학교 컴퓨터정보통신대학원
**고려대학교 대학원 컴퓨터학과
e-mail : {punky111, yuhc}@korea.ac.kr

Implementation of Vulnerable Analysis Script for Security Strength of Chrome OS

SeulGi Lee*, HeonChang Yu**
* Graduate School of Computer & Information Technology, Korea University
** Department of Computer Science and Engineering, Korea University

요 약

크롬 OS 는 온라인에 연결된 상태라면 시간과 장소의 제약 없이 인터넷에 연결해서 사용자 중심의 개인 환경을 제공하는 웹 OS 이다. 크롬 OS 의 특징이라면 구글 계정으로 접속하여 원하는 APP 설치나 다양한 구글의 온라인 서비스를 자유롭게 이용할 수 있다는 점을 꼽을 수 있다. 특히 기존의 윈도우나 리눅스와는 달리 사용자가 OS 전반에 걸친 설정을 직접 제어할 필요가 없다는 점이 획기적이다. 하지만 웹 OS 임에도 불구하고 이에 대한 보안 대비책이 전혀 마련되어 있지 않다. 따라서 이같은 보안 위협에 대비하기 위해 본 논문은 취약점 분석 쉘 스크립트 구현을 통해 취약점 탐지와 보안 강화를 통한 성능 개선 방안을 제안한다.

1. 서론

다양한 통신장비와 전자기기들이 기술 발전을 거듭해오면서 IT 기술도 나날이 진화해왔다. 그러나 이와 함께 온라인을 활용한 해킹 수법도 점차 강력해지고 있는 만큼, OS 의 보안 성능만 믿고 안심할 수는 없는 상황이 도래했다. 그로 인해 바이러스 같은 공격에서부터 시스템 취약 설정을 노린 공격들에 이르기까지 다양한 기법들이 등장했다.

크롬 OS 의 경우 현재 시스템을 탐지할 수 있는 스크립트가 전무한 상황이기에 취약점 분석을 하기 위한 쉘 스크립트의 설계 및 구현이 시급하다. 그래서 이 문제를 해결하고자 [2]의 점검항목들을 활용하여 쉘 스크립트를 구현하고자 한다. 기존의 유닉스, 리눅스 쉘 스크립트가 [1]만을 반영하였던 것에 비해 [2]까지 반영해 더 많은 항목들을 점검할 것이다. 이를 통해 실험 전후의 결과값 확인 및 보안 성능이 더 강화된 크롬 OS 사용이 가능하도록 처리하기 위한 방안을 살펴보겠다.

본 논문의 구성은 다음과 같다. 2 장에서는 크롬 OS 보안과 리눅스의 취약점 분석 쉘 스크립트와 관련해 논문 확인 및 취약점 점검항목들을 소개하고 있으며, 3 장에서는 실험 환경과 연구 방법에 대해서 소개하고 있다. 4 장에서는 3 장의 실험방법을 이용해 실험 전후 결과값 비교 및 실험 결과의 평가를 소개하고 있다. 마지막으로 5 장에서는 논문에 대한 결론을 내린다.

2. 관련 연구

크롬 OS 는 레벨 보호 메커니즘과 프로세스 샌드박싱, 커널 하드닝 같은 취약점 완화 기술을 결합한 보안 전략을 취한다[3]. 이 전략은 시스템의 그룹 장치 필터링과 자원 남용 제한 등을 제어하는 역할을 한다.

논문[4]에서는 리눅스 서버의 취약한 부분을 발견하고 이러한 문제를 개선하기 위한 목적으로 쉘 스크립트를 작성하는 연구를 진행했다. 그러나 리눅스 설정 매개변수 확인만으로 취약점을 완벽하게 진단하는데 있어서 한계점을 노출하고 있다. 논문을 읽을 때 주의가 요구된다.

논문[5]에서는 리눅스 서버 보안 강화를 위한 목적으로 쉘 스크립트 개발 연구를 진행했다. [4]와 비교했을 때 가독성이 높고 [1]의 새 버전을 적용한 것이 차이점이다. 다만 쉘 스크립트가 서버의 관점에서 작성되어 있어서 클라이언트 중심의 크롬 OS 와는 다른 관점에서 제작되었다. 따라서 논문을 탐독하면서 이 부분을 반드시 고려해야만 한다.

논문[6]에서는 웹 서버 보안 강화를 위한 목적으로 쉘 스크립트를 구현했다. [4], [5]와 비교했을 때 복잡한 스크립트를 구현하였음에도 불구하고 가독성 높은

출력물을 제공한다는 점에서 더 발전된 연구를 진행했다. 하지만 특정 명령어 패키지의 설치가 반영되지 않은 크롬 OS 에 곧바로 활용하기에는 무리가 있다.

취약점 분석 기술점검 항목

[1]은 행정안전부에서 root 계정 원격 접속 제한을 비롯한 취약점 분석 점검 항목을 이용해, OS 보안설정의 정상 적용 여부 탐지를 제공하는 가이드이다. 본 논문에서 사용하는 [1]의 항목들은 <표 1>과 같으며 해당 항목의 판단 기준과 시스템의 설정 변수들을 비교해 양호, 취약 여부를 가린다.

<표 1> 크롬 OS VM (가상메모리) 관리 설정 변수

항목번호	취약점 점검 항목
U-01	root 계정 원격 접속 제한
U-02	패스워드 복잡성 설정
U-03	계정 잠금 임계값 설정
U-04	패스워드 파일 보호
U-05	root 이외의 UID가 '0' 금지
U-06	root 계정 su 제한
U-07	패스워드 최소 길이 설정
U-08	패스워드 최대 사용 기간 설정
U-09	패스워드 최소 사용기간 설정
U-10	불필요한 계정 제거
U-11	관리자 그룹에 최소한의 계정 포함
U-12	계정이 존재하지 않는 GID 금지
U-13	동일한 UID 금지
U-14	사용자 shell 점검
U-15	Session Timeout 설정
U-16	root 홈, 패스 디렉터리 권한 및 패스 설정
U-17	파일 및 디렉터리 소유자 설정
U-18	/etc/passwd 파일 소유자 및 권한 설정
U-19	/etc/shadow 파일 소유자 및 권한 설정
U-20	/etc/hosts 파일 소유자 및 권한 설정
U-21	/etc/(x)inetd.conf 파일 소유자 및 권한 설정
U-22	/etc/syslog.conf 파일 소유자 및 권한 설정
U-23	/etc/services 파일 소유자 및 권한 설정
U-24	SUID, SGID, Sticky bit 설정 파일 점검
U-25	사용자, 시스템 시작파일 및 환경파일 소유자 및 권한 설정
U-26	world writable 파일 점검
U-27	/dev에 존재하지 않는 device 파일 점검
U-32	UMASK 설정 관리
U-33	홈디렉터리 소유자 및 권한 설정
U-34	홈 디렉터리로 지정한 디렉터리의 존재 관리
U-35	숨겨진 파일 및 디렉터리 검색 및 제거

[2]는 NSA 에서 레드햇 엔터프라이즈 리눅스 5 를 기준으로 작성한 가이드이다. [1]이 단순한 보안 설정 매개 변수의 적용만을 확인하는 반면, [2]는 이 부분과 함께 물리적인 보안까지도 같이 기술하고 있다. 본 논문에서는 크롬 OS 에 알맞은 항목들을 선별하여 <표 2>와 같이 적용하였으며 판단 기준은 [1]과 동일하다.

<표 2> 크롬 OS VM (가상메모리) 관리 설정 변수

항목번호	취약점 점검 항목
2.2.1.1	Add nodev Option to Non-Root Local Partitions
2.2.1.2	Add nodev, nosuid, and noexec Options to Removable Storage Partitions
2.2.1.3.1	Add nodev, nosuid, and noexec Options to /tmp
2.2.1.3.2	Add nodev, nosuid, and noexec Options to /dev/shm
2.2.1.4	Bind-mount /var/tmp to /tmp
2.2.3.2	Verify that All World-Writable Directories Have Sticky Bits Set
2.2.3.3	Find Unauthorized World-Writable Files
2.2.3.4	Find Unauthorized SUID/SGID System Executables
2.2.3.5	Find and Repair Unowned Files
2.2.3.6	Verify that All World-Writable Directories Have Proper Ownership
2.2.4.2	Disable Core Dumps
2.2.4.2.1	Ensure SUID Core Dumps are Disabled
2.2.4.3	Enable ExecShield
2.2.4.3.1	Ensure ExecShield is Enabled
2.2.4.4.1	Check for Processor Support on x86 Systems
2.3.1.3	Configure sudo to Improve Auditing of Root Access
2.3.1.4	Block Shell and Login Access for Non-Root System Accounts
2.3.1.5.1	Verify that No Accounts Have Empty Password Fields
2.3.1.7	Set Password Expiration Parameters
2.3.1.9	Set Accounts to Disable After Password Expiration
2.3.3.1.1	Set Password Quality Requirements, if using pam cracklib
2.3.3.1.2	Set Password Quality Requirements, if using pam passwdqc
2.3.3.2	Set Lockouts for Failed Password Attempts
2.3.3.3	Use pam deny.so to Quickly Deny Access to a Service
2.3.3.5	Upgrade Password Hashing Algorithm to SHA-512
2.3.3.6	Limit Password Reuse
2.3.4.1.2	Ensure that Root's Path Does Not Include World-Writable or Group-Writable Directories
2.3.4.4	Ensure that Users Have Sensible Umask Values
2.5.1.1	Network Parameters for Hosts Only
2.5.1.2	Network Parameters for Hosts and Routers
2.5.3.2.5	Limit Network-Transmitted Configuration
2.5.5.1	Inspect and Activate Default Rules
2.6.1.2.5	Send Logs to a Remote Host Using Reliable Transport
2.6.1.2.6	Enable rsyslog to Accept Remote Messages on Loghosts Only

취약점 분석 쉘 스크립트 구현

<표 3>은 <표 1>의 항목들을 구현한 쉘 스크립트의 일부분이다. 크롬 OS 사용 시스템의 설정 값과 [1]에서의 취약점 분석 판단 기준을 비교하여 양호•취약 여부를 판단한다. [4], [5], [6]과는 달리 실제 시스템 파일 설정 값들을 모두 확인 할 수 있다. 다만 크롬 OS 의 특성상 패키지 설치와 명령어 사용에 제약이 있어 가급적 기본 명령어만을 활용했다.

<표 3> moi_guide_check.sh 쉘 스크립트

```
#!/bin/sh
account=`whoami`
if [ "$account" != "root" ]; then
echo "Root 이외의 계정에서는 실행제한이 발생할 수 있으니 유의하시기 바랍니다."
exit
fi
...
"2")
echo"" >> moi_guide_check.log
echo"==============" >> moi_guide_check.log
echo " 1.1. root 계정 원격 접속 제한 " >> moi_guide_check.log
echo "==============" >> moi_guide_check.log
cat /etc/securetty | egrep "^tty" >> moi_guide_check.log
echo"" >> moi_guide_check.log
```

```
echo "양호: #tty가 모두 출력될 경우 (root 직접 접속 허용 및 원격 서비스 차단중)" >> moi_guide_check.log
echo "취약: tty가 1개 이상 출력될 경우 (root 직접 접속 허용 및 원격 서비스 사용중)" >> moi_guide_check.log
echo"" >> moi_guide_check.log
echo"" >> moi_guide_check.log
cat /etc/pam.d/login >> moi_guide_check.log
echo"" >> moi_guide_check.log
echo "양호: 모두 #auth로 출력시 (root 직접 접속 및 원격 서비스 차단중)" >> moi_guide_check.log
echo "취약: 모두 auth가 1개 이상 출력될 경우 (root 직접 접속을 허용 및 원격 서비스 사용중)" >> moi_guide_check.log
echo"" >> moi_guide_check.log
echo "==============" >> moi_guide_check.log
echo " 1.2. 패스워드 복잡성 설정 " >> moi_guide_check.log
echo "==============" >> moi_guide_check.log
cat /etc/shadow >> moi_guide_check.log
echo"" >> moi_guide_check.log
echo "양호: 영문,숫자,특수문자가 혼합된 8자리 이상의 패스워드가 모두 설정된 경우" >> moi_guide_check.log
echo "취약: 영문,숫자,특수문자가 혼합않된 8자 미만의 패스워드가 1개이상 설정된 경우" >> moi_guide_check.log
echo"" >> moi_guide_check.log
...
echo -n "Enter 입력시 메뉴로 다시 되돌아갑니다."
read TEMP;;
...
sh moi_guide_check.sh;;
esac
done
} MainMenuClass
```

<표 4>는 <표 2>의 항목들을 구현한 쉘 스크립트의 일부분이다. 양호•취약 여부의 판단은 <표 3>과 비슷하게 크롬 OS 사용 시스템의 설정 값과 [2]에서의 취약점 분석 판단 기준으로 비교한다. 하지만 <표 3>과는 다른 명령어를 수행하여 다양한 취약점 분석과 개선을 취할 수 있다는 장점이 있다.

<표 4> nsa_guide_check.sh 쉘 스크립트

```
#!/bin/sh
account=`whoami`
if [ "$account" != "root" ]; then
echo "Root 이외의 계정에서는 실행제한이 발생할 수 있으니 유의하시기 바랍니다."
exit
fi
...
"4")
echo"" >> nsa_guide_check.log
echo "================================" >> nsa_guide_check.log
echo " 2.2.1.1 Add nodev Option to Non-Root Local Partitions " >> nsa_guide_check.log
echo "================================" >> nsa_guide_check.log
cat /usr/share/baselayout/fstab >> nsa_guide_check.log
echo"" >> nsa_guide_check.log
echo "양호: 파일 시스템 형식이 ext2 또는 ext3이며 마운트 지점이 /가일 때 4번째 컬럼에 nodev가 입력되어 있는 경우" >> nsa_guide_check.log
echo "취약: 파일 시스템 형식이 ext2 또는 ext3이며 마운트 지점이 /가일 때 4번째 컬럼에 nodev가 입력되어 있지 않은 경우" >> nsa_guide_check.log
echo"" >> nsa_guide_check.log
echo "================================" >> nsa_guide_check.log
echo " 2.2.1.2 Add nodev, nosuid, and noexec Options to Removable Storage Partitions " >> nsa_guide_check.log
echo "================================" >> nsa_guide_check.log
cat /usr/share/baselayout/fstab >> nsa_guide_check.log
echo"" >> nsa_guide_check.log
echo "양호: (DVD 드라이브 같은) 탈착과 부착이 가능한 장치의 마운트 지점 4번째 컬럼에 nodev, nosuid, noexec가 입력되어 있는 경우" >> nsa_guide_check.log
echo "취약: (DVD 드라이브 같은) 탈착과 부착이 가능한 장치의 마운트 지점 4번째 컬럼에 nodev, nosuid, noexec가 입력되어 있지 않은 경우" >> nsa_guide_check.log
echo"" >> nsa_guide_check.log
...
echo -n "Enter 입력시 메뉴로 다시 되돌아갑니다."
read TEMP;;
...
sh moi_guide_check.sh;;
esac
done
} MainMenuClass
```

3. 실험

3.1 실험 환경

본 논문은 moi_guide_check.sh 와 nsa_guide_check.sh 에 각각 [1]과 [2]의 점검항목들을 반영한 쉘 스크립트를 이용해 실험을 진행했으며, 게스트 모드로 로그인해서 영향을 줄 수 있는 요인들을 최소화 시켰다. 실험에 사용한 모델과 시스템 환경은 <표 5>와 같다. 괄호 안은 오라클 버추얼 박스 환경이다.

<표 5> 실험환경 (LG 전자 15ND530-PX7SK)
(16GB 로 업그레이드)

CPU	인텔 i7-4702MQ 2.20Ghz 쿼드코어 (Intel i7-4702MQ 2.20Ghz 듀얼코어)
RAM	16 GB (2 GB)
HDD	500 GB (6 GB)
크롬 OS 버전	28.0.1484.1 4028.0.2013_04_20_1746 (Developer Build - hexxeh) vanilla x86-generic
커널 버전	Linux Version 3.4.0 (Chrome-bot@build107-m2)

3.2 실험방법

크롬 OS 취약점 분석 및 보안 강화를 실험하기 위해서 /bin/sh 커맨드를 통해 moi_guide_check.sh 와 nsa_guide_check.sh 쉘 스크립트를 실행한다. 그리하여 취약한 항목의 설정을 해결하기 전과 후를 보안 성능 평점으로 환산해 보안 성능의 개선 여부를 확인한다. 보안 성능 평점은 [1]과 [2]를 기준으로 X 는 0 점, △ 는 3 점, O 는 5 점으로 배점하였으며 총점은 각각 155 점과 170 점 이다.

4. 실험 결과 및 평가

<표 6> 실험 전 행정안전부 가이드라인 보안성능 평가 결과

항목번호	보안 성능 평점	항목번호	보안 성능 평점
U-01	△ (3.0)	U-16	O (5.0)
U-02	O (5.0)	U-17	△ (3.0)
U-03	X (0.0)	U-18	O (5.0)
U-04	O (5.0)	U-19	X (0.0)
U-05	O (5.0)	U-20	X (0.0)
U-06	O (5.0)	U-21	X (0.0)
U-07	X (0.0)	U-22	O (5.0)
U-08	O (5.0)	U-23	O (5.0)
U-09	X (0.0)	U-24	X (0.0)

U-10	0 (5.0)	U-25	0 (5.0)
U-11	0 (5.0)	U-26	0 (5.0)
U-12	0 (5.0)	U-27	0 (5.0)
U-13	0 (5.0)	U-32	0 (5.0)
U-14	X (0.0)	U-33	0 (5.0)
U-15	X (0.0)	U-34	0 (5.0)
		U-35	0 (5.0)
보안 평점 총합			106.0

<표 7> 실험 전 NSA 가이드라인 보안성능 평가 결과

항목번호	보안 성능 평점	항목번호	보안 성능 평점
2.2.1.1	X (0.0)	2.3.1.5.1	0 (5.0)
2.2.1.2	X (0.0)	2.3.1.7	△ (3.0)
2.2.1.3.1	X (0.0)	2.3.1.9	X (0.0)
2.2.1.3.2	X (0.0)	2.3.3.1.1	X (0.0)
2.2.1.4	X (0.0)	2.3.3.1.2	X (0.0)
2.2.3.2	0 (5.0)	2.3.3.2	X (0.0)
2.2.3.3	0 (5.0)	2.3.3.3	X (0.0)
2.2.3.4	0 (5.0)	2.3.3.5	X (0.0)
2.2.3.5	0 (5.0)	2.3.3.6	X (0.0)
2.2.3.6	0 (5.0)	2.3.4.1.2	0 (5.0)
2.2.4.2	X (0.0)	2.3.4.4	X (0.0)
2.2.4.2.1	X (0.0)	2.5.1.1	X (0.0)
2.2.4.3	X (0.0)	2.5.1.2	X (0.0)
2.2.4.3.1	X (0.0)	2.5.3.2.5	X (0.0)
2.2.4.4.1	X (0.0)	2.5.5.1	0 (5.0)
2.3.1.3	△ (3.0)	2.6.1.2.5	X (0.0)
2.3.1.4	0 (5.0)	2.6.1.2.6	X (0.0)
보안 평점 총합			51.0

<표 8> 실험 후 행정안전부 가이드라인 보안성능 평가 결과

항목번호	보안 성능 평점	항목번호	보안 성능 평점
U-01	0 (5.0)	U-16	0 (5.0)
U-02	0 (5.0)	U-17	0 (5.0)
U-03	0 (5.0)	U-18	0 (5.0)
U-04	0 (5.0)	U-19	0 (5.0)
U-05	0 (5.0)	U-20	0 (5.0)
U-06	0 (5.0)	U-21	0 (5.0)
U-07	0 (5.0)	U-22	0 (5.0)
U-08	0 (5.0)	U-23	0 (5.0)
U-09	0 (5.0)	U-24	0 (5.0)
U-10	0 (5.0)	U-25	0 (5.0)
U-11	0 (5.0)	U-26	0 (5.0)
U-12	0 (5.0)	U-27	0 (5.0)
U-13	0 (5.0)	U-32	0 (5.0)
U-14	0 (5.0)	U-33	0 (5.0)
U-15	0 (5.0)	U-34	0 (5.0)
		U-35	0 (5.0)
보안 평점 총합			155.0

<표 9> 실험 후 NSA 가이드라인 보안성능 평가 결과

항목번호	보안 성능 평점	항목번호	보안 성능 평점
2.2.1.1	0 (5.0)	2.3.1.5.1	0 (5.0)
2.2.1.2	0 (5.0)	2.3.1.7	0 (5.0)
2.2.1.3.1	0 (5.0)	2.3.1.9	0 (5.0)
2.2.1.3.2	0 (5.0)	2.3.3.1.1	0 (5.0)
2.2.1.4	0 (5.0)	2.3.3.1.2	0 (5.0)
2.2.3.2	0 (5.0)	2.3.3.2	0 (5.0)
2.2.3.3	0 (5.0)	2.3.3.3	0 (5.0)
2.2.3.4	0 (5.0)	2.3.3.5	0 (5.0)
2.2.3.5	0 (5.0)	2.3.3.6	0 (5.0)
2.2.3.6	0 (5.0)	2.3.4.1.2	0 (5.0)
2.2.4.2	0 (5.0)	2.3.4.4	0 (5.0)
2.2.4.2.1	0 (5.0)	2.5.1.1	0 (5.0)
2.2.4.3	0 (5.0)	2.5.1.2	0 (5.0)
2.2.4.3.1	0 (5.0)	2.5.3.2.5	0 (5.0)
2.2.4.4.1	0 (5.0)	2.5.5.1	0 (5.0)
2.3.1.3	0 (5.0)	2.6.1.2.5	0 (5.0)
2.3.1.4	0 (5.0)	2.6.1.2.6	0 (5.0)
보안 평점 총합			170.0

실험 결과값들을 살펴보면 실험 전과 후의 보안 성능 평점 총합에서 많은 차이가 나는 것을 볼 수 있다. 평점 총합은 비슷한 양상을 띄었지만 OS 버전에 따라서 성능 평가에 약간의 차이는 있었다. 실험 후에는 105~106 점에서 150~155 점으로 올라간 것을 확인할 수 있다.

5. 결론 및 향후과제

본 논문에서는 크롬 OS 취약점 분석 및 보안 강화를 위한 쉘 스크립트 구현에 대해서 제시했다. 보안 성능 평가를 하는 쉘 스크립트인 moi_guide_check.sh 와 nsa_guide_check.sh 를 설계 및 구현하여 실행함으로써 취약점 분석과 보안 강화가 가능함을 확인했다. 그렇지만 현재로서는 크롬 OS 버전에 따라서 항목의 활용 여부가 다른 경우가 발생하여 성능 평가를 하는데 있어서 부족한 부분이 많다. 그래서 향후에는 다양한 환경에서 커널 변형 등을 통한 취약점 분석과 관련하여 연구하는 것이 향후 과제이다.

참고문헌

[1] 한국인터넷진흥원, "주요정보통신기반시설 기술적 취약점 분석 평가 방법 상세가이드", 안전행정부, 2014

[2] Operating Systems Division Unix Team of the Systems and Network Analysis Center, “Guide to the Secure Configuration of Red Hat Enterprise Linux 5”, National Security Agency, 2011

[3] https://www.chromium.org/chromium-os/chromiumos-design-docs/security-overview

[4] 이호수, "리눅스 서버 보안 취약점 개선을 위한 쉘 스크립트 구현", 경북대학교, 2012 (석사논문)

[5] 정길영, "리눅스 서버 보안 강화를 위한 취약점 분석 스크립트 구현", 건국대학교, 2013 (석사논문)

[6] 이은식, "리눅스 웹 서버 보안 강화를 위한 취약점 분석 스크립트 구현", 성균관대학교, 2014 (석사논문)

[7] https://www.chromium.org/chromium-os/developer-information-for-chrome-os-devices/samsung-arm-chromebook