크롬비트의 취약점 분석을 적용한 크롬 OS 보안 강화 방안

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

Chrome OS Security Strength On Applying Vulnerable Analysis of ChromeBit

SeulGi Lee*, HeonChang Yu**
* Micro Communication
** Department of Computer Science and Engineering, Korea University

요 약

크롬비트는 크롬 OS 를 기반으로 하여 사용자 맞춤형 클라우드 서비스를 제공하는 스틱 PC 이다. 기존의 크롬북이 이동성을 강조한 외부 인터넷 환경에 특화된 기기라고 할 때, 크롬비트는 스마트 TV 구축 등을 비롯한 내부 인터넷 환경에 특화된 기기라고 볼 수 있다. 이처럼 크롬 OS 를 실내와 실외에서 사용 가능하다는 것을 감안한다면 클라우드와 사물인터넷 등을 매개로 하는 4 차 산업 혁명에 가장 근접한 OS 임에 틀림이 없다. 그럼에도 불구하고 현재 이 운영체제의 개발 속도와 더불어 보안 취약점 분석을 위한 방안은 전혀 마련되어 있지 않다. 그래서 본 논문에서는 취약점 분석 쉘 스트립트 실행을 통한 성능 향상 방안 마련과 크롬 OS 의 보안 성능 개선 여부 확인을 살펴보고자 한다.

1. 서론

4 차 산업혁명이 도래하면서 인터넷을 이용한 다양한 기술들이 속속 선보이고 있다. 수 많은 하드웨어와 소프트웨어들이 끊임없이 혁신을 하고 있는 만큼 사용자 서비스들도 하루가 다르게 진일보 하고 있다. 이와는 반대로 또 다른 한편에서는 사이버테러를 비롯한 크래킹 기술들도 예상치 못한 방법을 통해 공격을 거듭하고 있다.

이런 상황에서 관련 정보가 충분하지 않은 크롬 OS 는 취약점 분석 스크립트를 활용해 충분한 보안 정보 분석이 필요한 상황이다. 현재 이 부분과 관련해서는 OS 를 분석할 수 있는 데이터가 거의 없기에 취약점 판단을 위한 쉘 스크립트의 실행 및 경과 분석이 시급하다. 따라서 이 문제 해결을 위해 [1], [2]의 점검항목들을 활용한 [7]의 쉘 스크립트를 사용해 취약점 분석 및 보안 강화 방안을 파악하고자 한다. [7]이 PC 및 가상화 환경에서 실험했던 것과 비교해 스틱 PC 인 크롬비트의 결과 값도 활용해 보안 성능을 높이기 위한 방안을 강구할 것이다.

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 실험 방법에 대해서 소개하고 있다. 4 장에서는 3 장에서 실험한 방법을 활용하여 실험 전후 결과값 비교와 실험 결과에 대한 평가를 하고 있다. 마지막 5 장은 논문에 대한 결론을 맺는다.

2. 관련 연구

크롬 OS 는 부트확인, 암호화, 보안 업데이트 및 안티 바이러스 프로그램을 내장하는 정책을 취한다[3]. 이 정책은 데이터 유출과 악의적인 크래킹 시도로부터 보호하는 목적을 취한다.

이호수[4]는 리눅스 서버의 취약점을 발견해 해당 OS 의 보안 성능을 개선시키기 위해서 쉘 스크립트를 작성하는 연구를 했다. 하지만 리눅스 설정 매개변수만으로는 완벽한 취약점 진단은 힘들다. 그래서 이 연구를 파악하는데 있어 추가적인 연구 목표와 다른 연구 참조를 통한 새로운 방안이 시급하다.

정길영[5]는 리눅스 서버 보안 강화를 위해 [4]보다 가독성이 높고 새롭게 갱신된 [1]을 반영한 쉘 스크립트 개발을 연구했다. 그렇지만 클라우드 OS 인 크롬 OS 에 적용하기에는 무리가 있어 쉘 스크립트의 대대적인 수정이 반드시 수반되어야 한다는 점을 명심해야 한다.

이은식[6]은 웹 서버 보안 강화 쉘 스크립트를 구현했다. [4], [5]와 대비했을 때 스크립트는 복잡하지만

가독성 높은 결과를 확인 가능하다는 점에서 더 진일보한 연구를 했다. 그러나 [1]만을 반영한 연구를 진행했다는 점에서 대조군을 포함한 더 많은 실험이 필요해 보인다.

이슬기[7]은 [1]과 [2]를 활용해 더 다양한 결과가 분석가능한 연구를 한 반면에 크롬 OS 의 최신 버전을 반영하지 못했다는 점이 한계점으로 지적된다. 그래서 최신버전을 이용한 추가적인 연구진행이 요구된다.

2.1 취약점 분석 기술점검 항목

[1]은 행정안전부에서 제공하는 가이드로서 계정 잠금 임계값 설정 등을 활용한 리눅스, 유닉스 계열 OS 보안설정 취약점 분석 점검 항목들을 안내하고 있다. 본 논문에서 사용하는 [1]의 항목은 <표 1>에 상세하게 나와 있으며 항목의 판단 기준과 시스템 설정 변수들과 대조해 양호, 취약 등으로 나누어 판별한다.

<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 배포하는 가이드이며 레드햇 엔터프라이즈 리눅스 5 를 기준으로 작성했다. [1]이 보안설정 항목만을 확인한다면, [2]는 물리적인 보안들도 항목에 같이 반영되어 있다. 본 논문은 크롬 OS 에 알맞은 항목들을 선정해 <표 2>와 같이 나열했으며 판단 기준은 [1]과 동일하게 판별한다.

<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

2.2 취약점 분석 쉘 스크립트 구현

<표 3>은 <표 1>의 항목들을 반영한 실제 쉘 스크립트에서 발췌한 내용의 일부이다. 크롬 OS 시스템 설정 값과 [1]의 판별 기준을 적용하여 양호•취약 중 알맞은 결과값을 출력한다. 특히 [4], [5], [6]과는 다르게 모든 시스템 설정 값들을 확인 할 수 있다는 것이 큰 특징이다. 주의할 점은 크롬 OS 는 기존의 리눅스와는 달리 패키지 실행과 커맨드 명령 활용에 한계가 있다는 점에서 최대한 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 항목을 반영한 실제 쉘 스크립트에서 발췌한 내용의 일부이다. 양호•취약 여부의 결과값 출력은 <표 3>과 비슷하게 크롬 OS 시스템 설정과 [2]의 판별 기준으로 결과값을 출력한다. 특히 <표 3>과는 다른 명령어를 실행함으로써 다각적인 분석을 할 수 있다는 점에서 큰 장점으로 부각된다.

<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 및 nsa_guide_check.sh를 사용해 각각 [1], [2]의 기술점검 항목들을 반영한 쉘 스크립트를 활용하여 실험을 했다. 그리고 다른 프로세스들의 영향으로부터 최소화 시키기 위해 게스트 모드로 로그인해서 진행하였다. 실험에 사용한 모델과 시스템 환경은 <표 5>와 같다.

<표 5> 실험환경 (ASUS Chromebit CS10)

CPU	Rockchip RK3288C 1.80Ghz 쿼드코어
RAM	2 GB
eMMC	16 GB
크롬 OS 버전	56.0.2924.110 9000.91.0 (Official Build) Stable-Channel Veyron_Mickey
커널 버전	Linux Version 3.14.0 (Chrome-bot@cros-beefy245-c2)

3.2 실험방법

크롬 OS 의 취약점과 보안 강화 여부를 분석하기 위해서 크롬비트 내부에 존재하는 쉘 커맨드를 사용해 moi_guide_check.sh 와 nsa_guide_check.sh 스크립트를 실행시킨다. 그 후에는 개선이 필요한 항목의 설정들을 해결하기 전과 후로 나누어 보안 성능 평점으로 변환하여 향상 여부를 판단한다. 보안 성능 평점은 [1]과 [2]를 기준으로 X 는 0 점, △는 3 점, O 는 5 점으로 배점하였으며 총점은 각각 150 점과 155 점 이다.

4. 실험 결과 및 평가

<표 6> 실험 전 행정안전부 가이드라인 보안성능 실험 결과

항목번호	보안 성능 평점	항목번호	보안 성능 평점
U-01	X (0.0)	U-16	O (5.0)
U-02	X (0.0)	U-17	O (5.0)
U-03	X (0.0)	U-18	O (5.0)
U-04	O (5.0)	U-19	X (0.0)

U-05	0 (5.0)	U-20	X (0.0)
U-06	0 (5.0)	U-21	해당없음
U-07	X (0.0)	U-22	0 (5.0)
U-08	X (0.0)	U-23	0 (5.0)
U-09	X (0.0)	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	X (0.0)	U-34	0 (5.0)
		U-35	0 (5.0)
보안 평점 총합			105.0

<표 7> 실험 전 NSA 가이드라인 보안성능 실험 결과

항목번호	보안 성능 평점	항목번호	보안 성능 평점
2.2.1.1	X (0.0)	2.3.1.5.1	X (0.0)
2.2.1.2	X (0.0)	2.3.1.7	X (0.0)
2.2.1.3.1	X (0.0)	2.3.1.9	0 (5.0)
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	△ (3.0)
2.2.4.3.1	X (0.0)	2.5.3.2.5	X (0.0)
2.2.4.4.1	해당없음	2.5.5.1	0 (5.0)
2.3.1.3	△ (3.0)	2.6.1.2.5	해당없음
2.3.1.4	0 (5.0)	2.6.1.2.6	해당없음
보안 평점 총합			51.0

<표 8> 실험 후 행정안전부 가이드라인 보안성능 실험 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	해당없음
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			150.0

<표 9> 실험 후 NSA 가이드라인 보안성능 실험 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	해당없음	2.5.5.1	0 (5.0)
2.3.1.3	0 (5.0)	2.6.1.2.5	해당없음
2.3.1.4	0 (5.0)	2.6.1.2.6	해당없음
보안 평점 총합			155.0

실험 결과 평점으로부터 실험 전•후의 보안 성능 평점에서 상당한 차이가 나는 것을 확연히 알아볼 수 있다. 그리고 이슬기[7]과 비교하였을 때 평점 총합이 비슷하다 하더라도 OS 버전에 따라서 보안이 향상된 항목이 있는 반면에 보안이 저하된 항목들도 존재했음을 확인할 수 있었다. 실험 후에는 51~105 점에서 150~155 점으로 올라간 것을 확인할 수 있다.

5. 결론 및 향후과제

본 논문에서는 크롬 비트를 사용해 크롬 OS 의 취약점 분석과 보안 성능 강화 여부를 위한 쉘 스크립트 구현 방안을 제안했다. 취약점 점검 목적의 쉘 스크립트 moi_guide_check.sh 와 nsa_guide_check.sh 를 크롬비트에서 실행시켜 크롬 OS 의 개선 여부를 확인함으로써 개선된 항목과 저하된 항목이 존재함을 확인했다. 다만 크롬 OS 의 특성상 ARM 을 비롯한 다양한 CPU 로의 이식 가능성도 높은 만큼 충분한 보안 항목 검토와 꾸준한 항목 갱신도 같이 수반되어야 한다. 그래서 이 부분에 주목하여 향후에는 CPU 플랫폼의 특성에 맞는 다양한 보안 취약점 항목과 관련하여 연구하는 것이 목표 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://enterprise.google.com/chrome/work-computers-for-smaller-businesses/
[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] 이슬기, "크롬 OS 의 보안 강화를 위한 취약점 분석 스크립트 구현", 고려대학교, 2017 (석사논문)