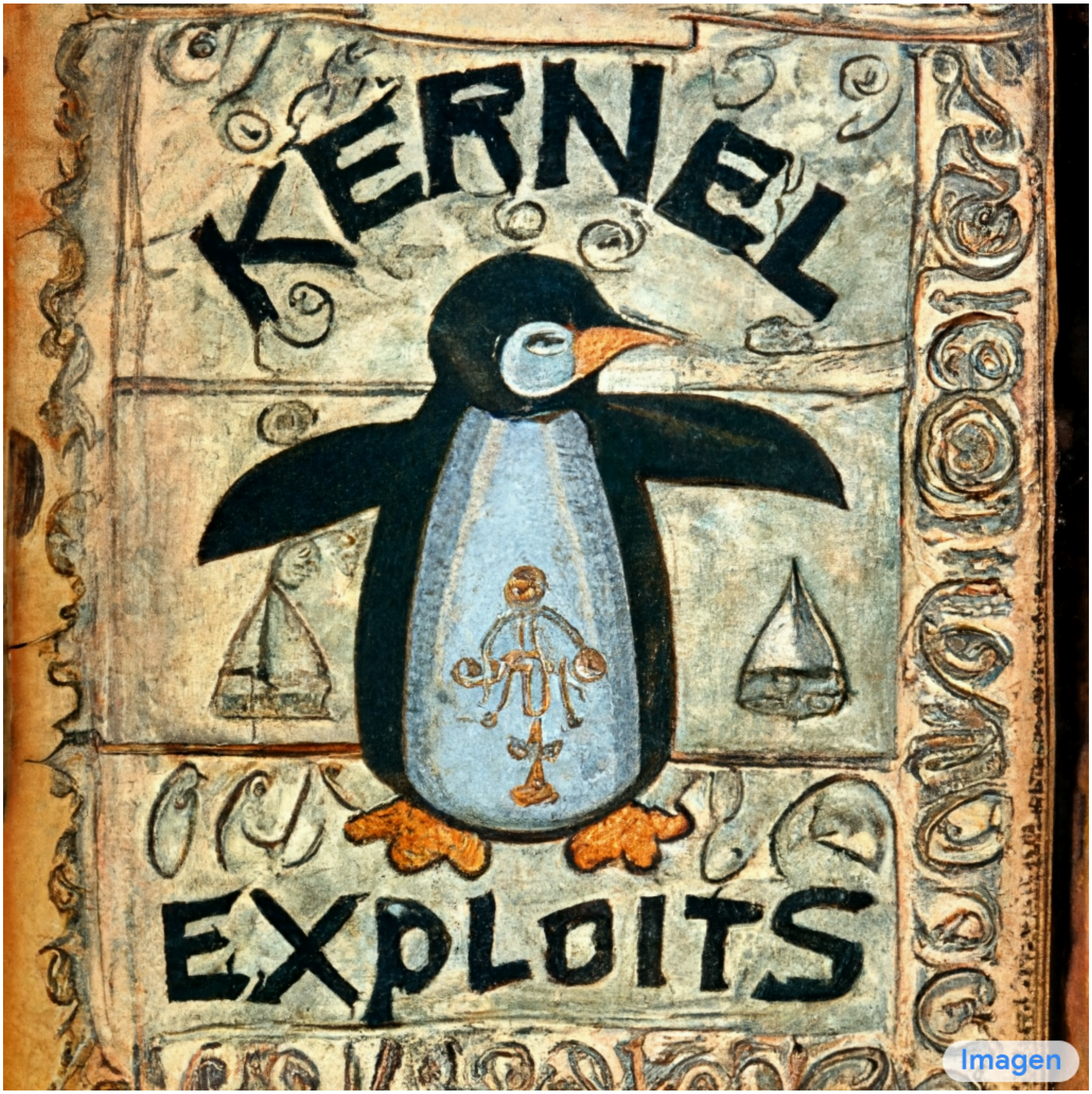
Kernel Exploits Recipes Notebook

This document contains a summary of the exploits we received in kCTF VRP. If you wish to send comments please contact us on discord here. You can download an illustrated (outdated) version of this document here.



CVE-2021-4154

Finder	Affected Versions	Fixed Versions
Syzbot	5.1	5.14, 5.13.4, 5.12.19, 5.10.52, 5.4.134

Cause	Patches
Type confusion	● https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=3b0462726e7ef281c35a7a4ae33e93ee2bc9975b

Consequence	PoCs
Temporal Memory Safety Violation	● https://syzkaller.appspot.com/bug?id=1bef50bdd9622a1969608d1090b2b4a588d0c6ac

Exploits

kctf-2022-exp1

Exploiter	Ingredients
Zhenpeng Lin	● CAP_SYS_ADMIN (vuln) ● [FREE][ELASTIC] with [GROOM][CROSSCACHE] ○ vuln obj was filp file pointer ○ attacking object was msgseg ○ victim object was pipe_buffer

Timeline

- Kernel patch - July 14 2021
- Exploited - December 14 2021
  - Kernel: 5.4.120
- Cluster updated - January 28 2022
  - GKE: 1.21.6-gke.1500

Directions

- [FREE][ELASTIC]
  - Convert to UAF using [GROOM][CROSSCACHE]
- [UAF][READ/WRITE]
  - Gain code execution with [UAF][WRITE] and [WRITE][PTR][FUNC]
    - Use [ROP][HEAP]
      - Leak heap address using [FREELIST][EMPTY] and [UAF][READ]

            - Find text pointer using [UAF][READ]
        - Execute [ROP][SELFPRIVESC]
        - Return via [ROP][USERSPACE]

kctf-2022-exp2

Exploiter

Bing-Jhong and Ramdhan from Starlabs

Timeline

- Kernel patch - July 14 2021
- Exploited - December 24 2021
    - Kernel: 5.4.120
- Cluster updated - January 28 2022
    - GKE: 1.21.6-gke.1500

Ingredients

- CAP_SYS_ADMIN (vuln)
- [FREE][ELASTIC] with [GROOM][CROSSCACHE]
    - vuln obj was filp file pointer
    - attacking object was msgseg
    - victim object was pipe_buffer

Directions

- [FREE][ELASTIC]
    - Convert to UAF using [GROOM][CROSSCACHE]
- [UAF][READ/WRITE]
    - Gain code execution with [UAF][WRITE] and [WRITE][PTR][FUNC]
        - Use [ROP][HEAP]
            - Leak heap address using [UAF][READ]
            - Find text pointer using [UAF][READ]
        - Execute [ROP][CHILDPRIVESC]
        - Return via [ROP][USERSPACE]

CVE-2021-22600

Finder	Affected Versions	Fixed Versions
Syzbot	5.6, 5.5.14, 5.4.29, 4.19.114, 4.14.175	5.16, 5.15.11, 5.10.88, 5.4.168, 4.19.222, 4.14.259
Cause	**Patches**	

Type confusion	● https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=ec6af094ea28f0f2dda1a6a33b14cd57e36a9755
Consequence	**PoCs**
Temporal Memory Safety Violation	● https://syzkaller.appspot.com/bug?id=8b2fd4b920d0bb1e6d9c839a1da0a6b5f5c1b118

Exploits

kctf-2022-exp3

Exploiter	**Ingredients**
Bing-Jhong and Ramdhan from Starlabs	● CAP_NET_RAW (vuln) ● [FREE][ELASTIC] with [GROOM][CROSSCACHE] ○ vuln obj was pg_vec ○ attacking object was msg_msg/msgseg ○ victim object was pipe_buffer
Timeline	
● Kernel patch - December 15 2021 ● Exploited - January 5 2022 ○ Kernel: 5.4.120 ● Cluster updated - April 26 2022 ○ GKE: 1.21.10-gke.2000	

Directions

- [FREE][ELASTIC]
  - Convert to UAF using [GROOM][CROSSCACHE]
- [UAF][READ/WRITE]
  - Gain code execution with [UAF][WRITE] and [WRITE][PTR][FUNC]
    - Use [ROP][HEAP]
      - Leak heap address using [UAF][READ]
      - Find text pointer using [UAF][READ]
    - Execute [ROP][CHILDPRIVESC]
    - Return via [ROP][USERSPACE]

CVE-2022-0185

Finder	**Affected Versions**	**Fixed Versions**
Jamie Hill-Daniel William Liu	5.1	5.17, 5.16.2, 5.15.16, 5.10.93, 5.4.173
Cause	**Patches**	
Integer Overflow	• https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=722d94847de29310e8aa03fcbdb41fc92c521756	
Consequence	**PoCs**	
Spatial Memory Safety Violation	• https://syzkaller.appspot.com/bug?id=53c05996968fc87df17de205b461f4f96d5b5907	

Exploits

kctf-2022-exp4

Exploiter	**Ingredients**
Jamie Hill-Daniel William Liu	• CAP_SYS_ADMIN (vuln) • Convert [OOB][WRITE] with [GROOM][SAMESIZE] into [OOB][READ] ○ vuln object was fs_context ○ victim object was msg_msg (m_ts) • Convert [OOB][WRITE] with [GROOM][SAMESIZE] into [FREE][INVALID] ○ vuln object was fs_context ○ attacking object was msg_msg ○ victim object was msg_msg • Convert [UAF][WRITE] with [GROOM][SAMESIZE] into [ROP] ○ attacking object was msg_msg ○ victim object was pipe_buffer
Timeline	
• Kernel patch - January 18 2022 • Exploited - January 18 2022 ○ Kernel: 5.10.68 • Cluster updated - April 17 2022 ○ GKE: 1.22.8-gke.200	

Directions

- [OOB][WRITE]
  - Obtain [OOB][READ] using [OOB][WRITE] with [WRITE][LEN]
    - Prepare the heap with [GROOM][SAMESIZE]
  - Obtain [FREE][INVALID] using [OOB][WRITE] and then [WRITE][PTR][FREE]
    - Prepare the heap with [GROOM][SAMESIZE]

- [FREE][INVALID]
    - Gain code execution with [UAF][WRITE] and then [WRITE][PTR][FUNC]
        - Prepare the heap with [GROOM][SAMESIZE]
        - Use [ROP][HEAP]
            - Leak heap address using [OOB][READ]
            - Find text pointer using [OOB][READ]
        - Execute [ROP][SELFPRIVESC]
        - Return via [ROP][USERSPACE]

kctf-2022-exp5

Exploiter

Bing-Jhong and Ramdhan from Starlabs

Timeline

- Kernel patch - January 18 2022
- Exploited - January 6 2022
    - Kernel: 5.4.144
- Cluster updated - April 26 2022
    - GKE: 1.21.10-gke.2000

Ingredients

- CAP_SYS_ADMIN (vuln)
- [OOB][WRITE] with [GROOM][CROSSCACHE]
    - vuln object was fs_context
    - attacking object was msg_msg
    - victim object was pipe_buffer

Directions

- [OOB][WRITE]
    - Prepare the heap with [GROOM][CROSSCACHE]
    - Gain new arbitrary write from [OOB][WRITE] and then [WRITE][LEN] and then an arbitrary read using [OOB][READ]
- [OOB][READ/WRITE]
    - Gain code execution with [OOB][WRITE] and then [WRITE][PTR][FUNC]
        - Use [ROP][HEAP]
            - Leak heap address using [OOB][READ]
            - Find text pointer using [OOB][READ]
        - Execute [ROP][SELFPRIVESC]
        - Return via [ROP][USERSPACE]

CVE-2022-27666

Finder	Affected Versions	Fixed Versions
slipper from pangu team valis	4.11	5.17, 5.16.15, 5.15.29, 5.10.108, 4.19.237, 4.14.274

Cause	Patches
Missing Bounds Check	● https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=ebe48d368e97d007bfeb76fcb065d6cfc4c96645

Consequence	PoCs
Spatial Memory Safety Violation	● https://syzkaller.appspot.com/bug?id=517fa734b92b7db404c409b924cf5c997640e324 ● https://syzkaller.appspot.com/bug?id=57375340ab81a369df5da5eb16cfcd4aef9dfb9d

Exploits

kctf-2022-exp6

Exploiter	Ingredients
slipper from pangu team	● CAP_NET_ADMIN (vuln) ● [OOB][WRITE] with [GROOM][BUDDY] ○ vuln object was esp frag buffer ○ attacking object was xattr ○ victim object was xfrm_policy
Timeline	
● Kernel patch - March 7 2022 ● Exploited - February 12 2022 ○ Kernel: 5.4.120 ● Cluster updated - April 17 2022 ○ GKE: 1.22.8-gke.200	

Directions

- [OOB][WRITE]
  - Prepare the heap with [GROOM][BUDDY]
  - Convert to UAF using [OOB][WRITE] and then [WRITE][REF].
- [UAF][READ/WRITE]
  - Gain code execution with [OOB][WRITE] and then [WRITE][PTR][FUNC]
    - Use [ROP][HEAP]

            - Leak heap address using [UAF][READ]
            - Find a text pointer using [UAF][READ]
        - Execute [ROP][SELFPRIVESC]
        - Return via [ROP][USERSPACE]

kctf-2022-exp7

Exploiter

valis

Timeline

- Kernel patch - March 7 2022
- Exploited - March 15 2022
    - Kernel: 5.10.90
- Cluster updated - April 17 2022
    - GKE: 1.22.8-gke.200

Ingredients

- CAP_NET_ADMIN (vuln)
- [OOB][WRITE] with [GROOM][BUDDY]
    - vuln object was esp frag buffer
    - attacking object was xattr
    - victim object was socket

Directions

- [OOB][WRITE]
    - Prepare the heap with [GROOM][BUDDY]
    - Convert limited [OOB][WRITE] to unlimited [OOB][WRITE] using [WRITE][LEN]
- [OOB][READ/WRITE]
    - Gain code execution with unlimited [OOB][WRITE] and then [WRITE][PTR][FUNC]
        - Use [ROP][PTREGS]
            - Find text pointer using [OOB][READ]
        - Use [ROP][HEAP]
            - Leak heap address using [OOB][READ]
        - Execute [ROP][SELFPRIVESC]
        - Return via [ROP][USERSPACE]

kctf-2022-exp12

Exploiter

d3v17

Timeline

Ingredients

- CAP_NET_RAW (vuln)
- [OOB][WRITE] with [GROOM][BUDDY]
    - vulnerable object is esp frag buffer

● Kernel patch - March 7 2022 ● Exploited - April 28 2022 ○ Kernel: 5.10.90 ● Cluster updated - April 17 2022 ○ GKE: [1.22.8-gke.200]()	○ attacking objects are poll_list, user_key_payload and packet_fanout ○ victim object is pipe_buffer
Directions	
● [OOB][WRITE] ○ Prepare the heap with [GROOM][BUDDY] ○ Convert limited [OOB][WRITE] to unlimited [OOB][READ] using [WRITE][LEN] ● [OOB][WRITE] ○ Convert [OOB][WRITE] to [FREE][INVALID] ● [FREE][INVALID] ○ Gain code execution with [UAF][WRITE] and then [WRITE][PTR][FUNC] ■ Use [ROP][HEAP] ● Find text pointer using [OOB][READ] ● Leak heap address using [OOB][READ] ■ Execute [ROP][SELFPRIVESC] ■ Return via [ROP][USERSPACE]	

CVE-2022-1055

Finder	**Affected Versions**	**Fixed Versions**
Syzbot	5.1	5.17, 5.16.6, 5.15.20, 5.10.97, 5.4.177
Cause	**Patches**	
Race Condition	● https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=04c2a47ffb13c29778e2a14e414ad4cb5a5db4b5	
Consequence	**PoCs**	
Temporal Memory Safety Violation	● https://syzkaller.appspot.com/bug?id=2212474c958978ab86525fe6832ac8102c309ffc	
Exploits		

kctf-2022-exp8

Exploiter

valis

Timeline

- Kernel patch - January 31 2022
- Exploited - March 16 2022
    - Kernel: 5.4.144
- Cluster updated - April 17 2022
    - GKE: 1.22.8-gke.200

Ingredients

- CAP_SYS_ADMIN (vuln)
- [UAF][WRITE] with [GROOM][SAMESIZE]
    - vuln object was tcf_chain
    - attacking object was simple_xattr
    - victim object was tcf_chain
- [FREE][ELASTIC] with [GROOM][SAMESIZE]
    - vuln object was sock
    - attacking object was simple_xattr
    - victim object is sock

Directions

- [UAF][WRITE]
    - Convert [UAF][WRITE] to [FREE] by using [WRITE][PTR][STRUCT] to do [WRITE][PTR][FUNC] through [WRITE][STATIC] and call [ROP][GADGET][ULP] pointing to a function that calls free
        - Prepare the heap with [GROOM][SAMESIZE]
        - Limits: [WRITE][PTR][FUNC] can't be used for ROP directly because there's no known controlled memory address to put the stack on and the context is RCU (so [ROP][PTREGS] isn't possible)
- [FREE][ELASTIC]
    - Gain code execution with [UAF][WRITE] and [WRITE][PTR][FUNC]
        - Prepare the heap with [GROOM][SAMESIZE]
        - Use [ROP][PTREGS]
            - Bruteforce text pointer (1/512 chance)
        - Use [ROP][HEAP]
            - Get heap address from previous ROP chain
        - Execute [ROP][SELFPRIVESC]
        - Return via [ROP][USERSPACE]

CVE-2022-29582

Finder	Affected Versions	Fixed Versions
Jayden Rivers David Bouman	5.5	5.18, 5.17.3, 5.16.20, 5.15.34, 5.10.111

Cause	Patches
Race Condition	● https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=e677edbcabee849bfdd43f1602bccbecf736a646

Consequence	PoCs
Temporal Memory Safety Violation	● n/a

Exploits

kctf-2022-exp9

Exploiter

Jayden Rivers
David Bouman

Timeline

- Kernel patch - April 8 2022
- Exploited - April 12 2022
  - Kernel: 5.10.90
- Cluster updated - June 29 2022
  - GKE: [1.24.1-gke.1400]

Ingredients

- No capabilities needed
- [FREE][ELASTIC] with [GROOM][CROSSCACHE]
  - vulnerable object is filp
  - attacking object is msgseg
  - victim object is tls_context

Directions

- [FREE][ELASTIC]
  - Convert to UAF using [GROOM][CROSSCACHE]
- [UAF][WRITE]
  - Gain code execution with [UAF][WRITE] and [WRITE][PTR][FUNC]
    - Use [ROP][HEAP]
      - Leak heap address using [UAF][READ]
      - Find text pointer using [UAF][READ]
    - Execute [ROP][SELFPRIVESC]
    - Return via [ROP][USERSPACE]

CVE-2022-1116

Finder	Affected Versions	Fixed Versions
Bing-Jhong from Starlabs	5.4.24	5.4.189

Cause	Patches
Integer Overflow	● https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=1a623d361ffe5cecd4244a02f449528416360038

Consequence	PoCs
Temporal Memory Safety Violation	● n/a

Exploits

kctf-2022-exp10

Exploiter	**Ingredients**
Bing-Jhong from Starlabs	● No capabilities needed ● [FREE][ELASTIC] with [GROOM][CROSSCACHE] ○ vulnerable object is fs_struct ○ attacking object is msgseg ○ victim object is pipe_buffer
Timeline	
● Kernel patch - April 14 2022 ● Exploited - April 20 2022 ○ Kernel: 5.4.170 ● Cluster updated - June 29 2022 ○ GKE: 1.21.12-gke.1500	

Directions

- [FREE][ELASTIC]
  - Convert to UAF using [GROOM][CROSSCACHE]
- [UAF][WRITE]
  - Gain code execution with [UAF][WRITE] and [WRITE][PTR][FUNC]
    - Use [ROP][HEAP]
      - Leak heap address using [UAF][READ]
      - Find text pointer using [UAF][READ]

- Execute [ROP][SELFPRIVESC]
- Return via [ROP][CHILDSPACE]

CVE-2022-29581

Finder	Affected Versions	Fixed Versions
Syzbot	4.14	5.18, 5.17.5, 5.15.36, 5.10.113, 5.4.191, 4.19.241, 4.14.278

Cause	Patches
Faulty Reference Count	• https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=3db09e762dc79584a69c10d74a6b98f89a9979f8
Consequence	**PoCs**
Temporal Memory Safety Violation	• https://syzkaller.appspot.com/bug?id=0ca897284a4e1bbc149ad96f15917e8b31a85d70

Exploits

kctf-2022-exp11

Exploiter	Ingredients
Kyle Zeng	• CAP_NET_RAW, CAP_SYS_ADMIN (vuln) • [FREE][ELASTIC] with [GROOM][CROSSCACHE] ○ vulnerable object is net ○ attacking object is msg_msg ○ victim object is net
Timeline	
• Kernel patch - April 13 2022 • Exploited - December 27 2021 ○ Kernel: 5.4.120 • Cluster updated - June 29 2022 ○ GKE: 1.21.12-gke.1500	
Directions	

- [FREE][ELASTIC]
  - Prepare the heap using [GROOM][CROSSCACHE]
  - Gain code execution with [UAF][WRITE] and [WRITE][PTR][FUNC]
    - Use [ROP][HEAP]
      - Leak heap address using [UAF][READ]
      - Find text pointer using [UAF][READ] against cpu_entry_area
    - Execute [ROP][SELFPRIVESC]
    - Return via [ROP][USERSPACE]

CVE-2022-1786

Finder	Affected Versions	Fixed Versions
Kyle Zeng	5.10	5.10.117, 5.12

Cause	Patches
Type Confusion	● https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=29f077d070519a88a793fbc70f1e6484dc6d9e35

Consequence	PoCs
Temporal Memory Safety Violation	● https://www.openwall.com/lists/oss-security/2022/05/28/1

Exploits

kctf-2022-exp13

Exploiter	**Ingredients**
Kyle Zeng	● No capabilities needed ● [FREE][INVALID] with [GROOM][SIZE] ○ vulnerable object is io_identity ○ attacking object is msgseg ○ victim object is timerfd_ctx
Timeline	
● Kernel patch - May 16 2022 ● Exploited - Apr 29 2022 ○ Kernel: 5.10.90	

- Cluster updated - June 29 2022
    - GKE: 1.24.1-gke.1400

Directions

- [FREE][INVALID]
    - Limits: Small overlap
    - Prepare the heap using [GROOM][SAMESIZE]
    - Leak heap address using [UAF][READ]
    - Leak freelist using [UAF][READ] and [FREELIST][EMPTY]
    - Create a new [FREE][INVALID] (with a better overlap) by adding a new slot to the freelist using [UAF][WRITE]
- [FREE][INVALID]
    - Gain code execution with [UAF][WRITE] and [WRITE][PTR][FUNC]
        - Use [ROP][HEAP]
            - Leak heap address using [UAF][READ]
            - Find text pointer using [UAF][READ]
        - Use ROP for [ROP][GADGET][BINFMT] to get [WRITE][PTR][FUNC] but on on task context
            - Limits: ROP is limited because of RCU context
    - Gain code execution (in task context) with [WRITE][PTR][FUNC]
        - Use [ROP][PTREGS]
            - call copy_from_user + pop rsp
        - Use [ROP][HEAP]
        - Execute [ROP][SELFPRIVESC]
        - Return via [ROP][TELEFORK]

CVE-2022-2327

<table>
<tr><th>Finder</th><th>Affected Versions</th><th>Fixed Versions</th></tr>
<tr><td>Bing-Jhong Billy Jheng</td><td>5.10</td><td>5.10.125</td></tr>
<tr><th>Cause</th><th colspan="2">Patches</th></tr>
<tr><td>Use After Free and Faulty Reference Count</td><td colspan="2">• https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?h=linux-5.10.y&id=df3f3bb5059d20ef094d6b2f0256c4bf4127a859</td></tr>
</table>

Consequence	PoCs
Temporal Memory Safety Violation	• n/a

Exploits

kctf-2022-exp14

Exploiter	Ingredients
Bing-Jhong Billy Jheng	• Requires userns for allocating a nsproxy • [FREE][ELASTIC] with [GROOM][CROSSCACHE] ○ vulnerable object is nsproxy ○ attacking object is msg_msg ○ victim object is dentry->d_op (for exec) and pipe_buffer (for leak)

Timeline

- Kernel patch - June 22 2022
- Exploited - June 22 2022
  - Kernel: 5.10.107
- Cluster updated - August 1 2022
  - GKE: 1.24.2-gke.1900

Directions

- [FREE][ELASTIC]
  - Convert to UAF using [GROOM][CROSSCACHE]
- [UAF][READ/WRITE]
  - Gain code execution with [UAF][WRITE] and [WRITE][PTR][FUNC]
    - Use [ROP][HEAP]
      - Leak heap address using [UAF][READ]
      - Find text pointer using [UAF][READ]
    - Execute [ROP][EXEC]
    - Return via [ROP][SLEEP]

CVE-2022-20409

Finder	Affected Versions	Fixed Versions
Zhenpeng Lin	5.10	5.10.134

Cause	Patches
Faulty Reference Count	● https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=2ee0cab11f6626071f8a64c7792406dabdd94c8d
Consequence	**PoCs**
Temporal Memory Safety Violation	● n/a

Exploits

kctf-2022-exp15	
Exploiter	**Ingredients**
Zhenpeng Lin	● [FREE][DIRTY] with [GROOM][CROSSCACHE] ○ vulnerable object is io_identity ○ target object is cred
Timeline	
● Kernel patch - June 29 2022 ● Exploited - August 5 2022 ○ Kernel: 5.10.107 ● Cluster updated - ○ GKE: xxx	
Directions	
● [FREE][DIRTY] with [GROOM][CROSSCACHE] ○ To stabilize the heap use [GROOM][CROSSCACHE] with iov (so the second free triggers on iov, and then the legitimate free of iov is used for the double free on the target object) ○ To allocate the target object use setuid ○ Spray the privileged \`cred\` by invoking usermode helper by trying to create IRDA sockets, which are attempted to be autoloaded by loading a kernel module, which allocates a usermode helper which allocates a privileged cred struct.	

DRAFT

CVE-2022-2588

Finder	Affected Versions	Fixed Versions
Zhenpeng Lin		

Cause	Patches
	●

Consequence	PoCs
	●

Exploits

kctf-2022-exp17

Exploiter	Ingredients
Zhenpeng Lin	● [FREE][ELASTIC] with [GROOM][CROSSCACHE] ○ vuln obj was ○ attacking object was msgseg ○ victim object was pipe_buffer
Timeline	
● Kernel patch - ● Exploited - ○ Kernel: ● Cluster updated - ○ GKE:	

Directions

- [FREE][ELASTIC]
  - Convert to UAF using [GROOM][CROSSCACHE]
- [UAF][READ/WRITE]
  - Gain code execution with [UAF][WRITE] and [WRITE][PTR][FUNC]
    - Use [ROP][HEAP]
      - Leak heap address using [FREELIST][EMPTY] and [UAF][READ]
      - Find text pointer using [UAF][READ]

        - Execute [ROP][SELFPRIVESC]
        - Return via [ROP][USERSPACE]

<table>
<tr><th colspan="2">kctf-2022-exp19</th></tr>
<tr><td><b>Exploiter</b></td><td><b>Ingredients</b></td></tr>
<tr><td>Zhenpeng Lin</td><td rowspan="3">●</td></tr>
<tr><td><b>Timeline</b></td></tr>
<tr><td>● Kernel patch -
● Exploited -
○ Kernel:
● Cluster updated -
○ GKE:</td></tr>
<tr><td colspan="2"><b>Directions</b></td></tr>
<tr><td colspan="2">●</td></tr>
</table>

Definitions

[WRITE]

This primitive allows you to write into a new object. Usually just a field (in the case of UAF) or just the beginning (in the case of OOB).

- [STATIC] write to a static address, which sits at a known address (still bound to KASLR)
- [REF] If the new object is corrupted with a different reference count, follow the [UAF] or [FREE] recipes.
- [LEN] If the new object is corrupted with a different length, follow the OOB recipe.
- [PTR] If the new object is corrupted with a different pointer, depends on what that pointer is used for:
    - [FREE] If it can be used to trigger a **free** on an arbitrary pointer, then this requires identifying a pointer to a known object and then following the UaF recipe.
    - [READ] If it can be used to change a pointer to an object/string/number that can be **leaked** then it requires to have an address of something interesting to leak, and then follow the instructions of UaF Read. Obtaining an address of something

    interesting to leak requires either an infoleak, or a static address with interesting data on it (like cpu_entry_area in order to bypass KASLR).
    - [WRITE] If it can be used to change a pointer to an object/string/number that is written to, then it could be used as a write primitive on an arbitrary address (**write-what-where**).
    - [FUNC] If it can be used to change a **function pointer** then one could try to just point it to a stack pivot and follow [ROP] recipe.
    - [STRUCT] If it can be used to change a pointer to an object that has other objects inside that are then called/read/written to. This might require having an object at a known location which is controlled. Primitive is the same as UaF Write PTR (but may provide more flexibility).

[UAF]

Use After Free. Find a new object that can be allocated in the same slot and has data (either other pointers or length) at the same offsets that the UaF lets you control. Follow [GROOM] recipe.

- [WRITE]
    - Follow a [WRITE] recipe.
- [READ] Use it to leak secrets. In a data-only attack, for example, this could be used to leak information mapped in kernel memory from other processes or just text/function pointers to bypass KASLR.

[FREE]

- [ELASTIC] Exploit a double-free by allocating an object of an arbitrary size that gives read/write/execute primitives.
    - Trigger a free on the vulnerable object.
    - Allocate an elastic object around the same spot of the vulnerable object. Elastic object provides more control over content (like msg_msg). Follow [GROOM] recipe.
    - Use the vulnerability to modify memory across objects
        - Use the vulnerability to free the elastic object so you can put another object on the same slot.
        - Allocate a victim object.
        - Use the elastic object to read and possibly modify data in the victim object. Follow [UAF] recipe.
- [DIRTY] Exploit a double free to replace a structure that stores privileges (like cred), with a more privileged one
    - Trigger free on the vulnerable object.
    - Allocate a target object (an object that will have its privileges escalated like cred, file, inode, etc).
    - Use the vulnerability to free the target object so the slot becomes available again.
    - Do an operation that forces the allocation of a more privileged target object so it reuses the recently vacated space.

- [INVALID]
    - Exploit invalid free that adds an entry to the freelist that is not properly aligned.

[OOB]

- [WRITE] Find a new object that can be allocated on a contiguous slot (cross-slab, same size or using the buddy allocator - see [GROOM] recipe) and has something that can be overwritten safely (that is, the data on the previous fields in the struct won't break and panic the kernel), then follow a [WRITE] recipe.
- [READ] Find a new object that can be allocated after the vulnerable object, and read its contents (for example, function pointers to bypass KASLR, or linked lists, to find a heap object that can be controlled by the attacker).

[GROOM]

Heap Grooming / Heap Feng Shui

- [BUDDY] For situations where there is an overflow on a "linear" memory mapping on the buddy allocator, one needs to allocate memory of specific sizes to ensure the right pages are reused.
- [CROSSCACHE] By attacking the slab allocator, one can free all objects in a slab to force the pages to be freed and reallocated to another slab.
- [SAMESIZE] Control which objects are put on the heap by freeing and then allocating another object of the same size. Could use elastic objects (like msgmsg) or constant-size objects.

[FREELIST]

Attacking freelist hardening mitigations

- [EMPTY] Free all values, which results in secret xor 0 which is equal to secret. Reallocating the value results in secret xor address xor 0 which leaks address if secret is known (eg, by reading it when empty).

[ROP]

- [GADGET]
    - [ULP] Register a TCP ULP so that it frees socket objects on demand (or just get rip control on task context)
    - [BINFMT] Register a binfmt listener that gives rip control (useful to move from RCU to task context)
- Stack
    - [HEAP]
        - Put the chain in the heap (requires a way to leak the heap address).
        - Jump to the gadget that pivots the stack to the address.
    - [PTREGS]
        - Put a short (80 bytes) ROP chain on the registers which are then pushed

to the Kernel stack.

        - Jump to the gadget that shifts the stack and jumps to it.
    - [STATIC]
        - Put a ROP chain on a static variable (see [WRITE][STATIC]). Requires building a JOP chain that sets RSP to the right value.
- Execute
    - [SELFPRIVESC] Change task privileges of current process
    - [CHILDPRIVESC] Change task privileges of child process
    - [EXEC] Execute arbitrary command (eg call_usermodehelper_exec)
    - [WRITE] Write arbitrary content in arbitrary location
- Exit
    - [USERSPACE] Return to userspace
    - [TELEFORK] Fork (&sleep)
    - [SLEEP] Sleep