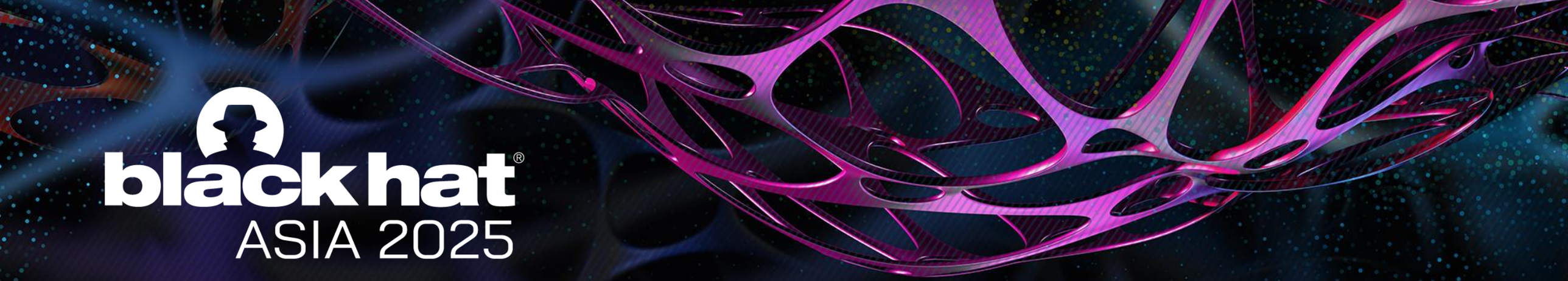


KernJC: Automated Vulnerable Environment Generation for Linux Kernel Vulnerabilities

Speakers: Bonan Ruan, Jiahao Liu

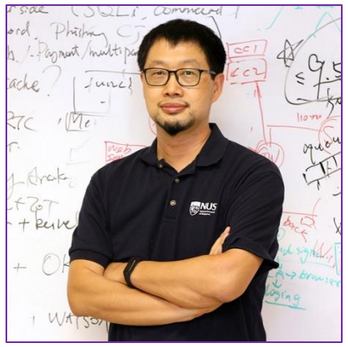
Contributors: Chuqi Zhang, Zhenkai Liang

ABOUT US

Bonan Ruan, Ph.D. Student, NUS
- Ex-NSFOCUS Security Researcher
- GitHub: @brant-ruan
- Homepage: *profile.wohin.me*
- E-mail: r-bonan@comp.nus.edu.sg

Jiahao Liu, Ph.D. Student, NUS
- GitHub: @ljiahao
- Homepage: *ljiahao.github.io*
- E-mail: jiahao99@comp.nus.edu.sg

Chuqi Zhang, Ph.D. Student, NUS
- GitHub: @Icegrave0391
- Homepage: *chuqiz.notion.site*
- E-mail: chuqiz@comp.nus.edu.sg

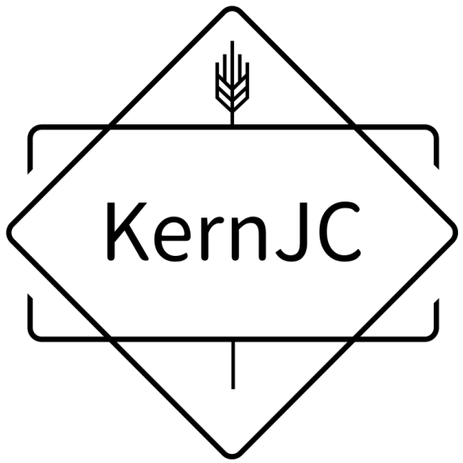


Zhenkai Liang, Assoc Prof, NUS
- Homepage: *comp.nus.edu.sg/~liangzk*
- E-mail: liangzk@comp.nus.edu.sg

nus-curiosity.github.io

ABOUT KERNJC

github.com/NUS-Curiosity/KernJC



ENDLESS KERNEL VULNERABILITIES!

Public kCTF VRP / kernelCTF responses

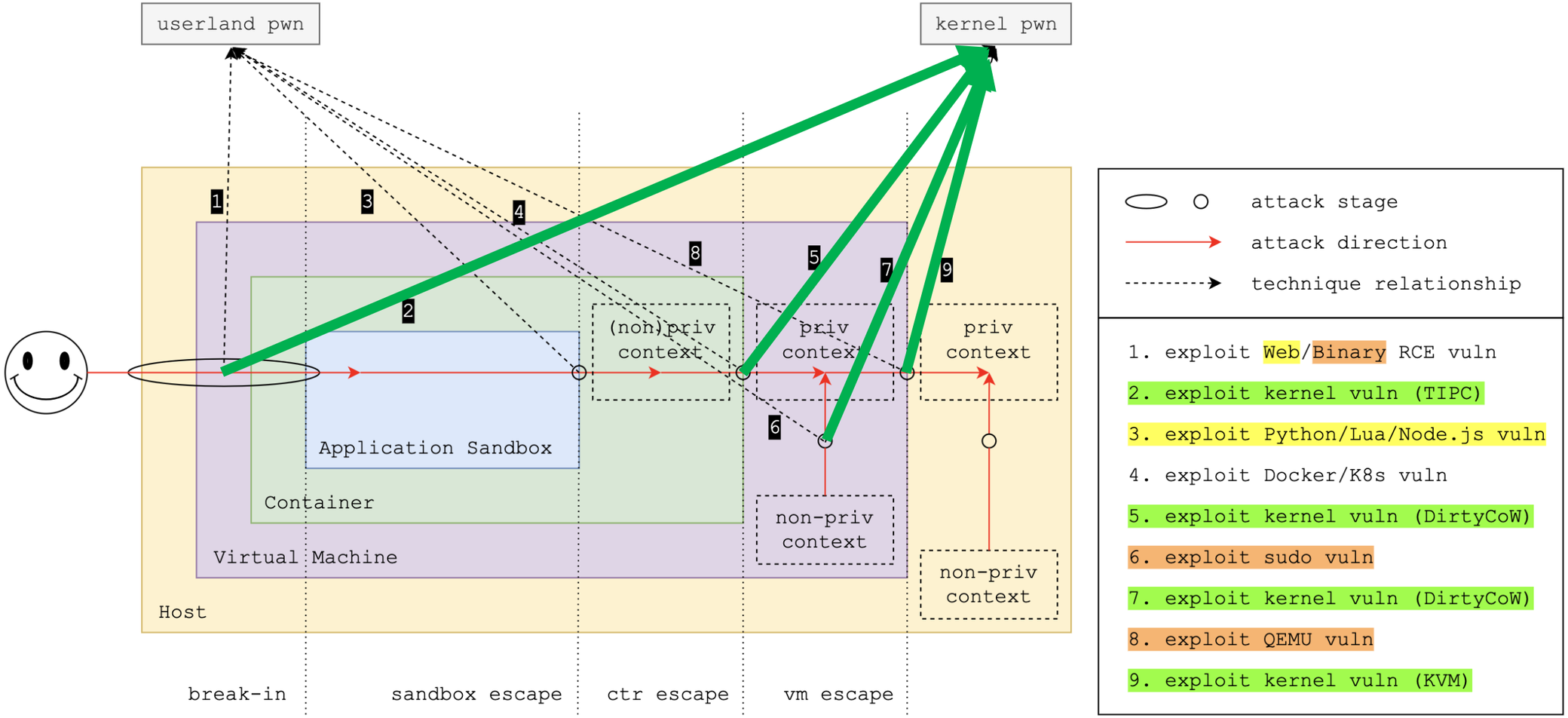
kernelCTF | kCTF VRP

ID	Flag submission time	Flags	0-day / 1-day	LTS slot	COS slot		Patch commit title	CVE
exp265	2025-03-18T12:25:41.453Z	kernelCTF{v1:lts-6.6.80:1742300713}	0-day	(dupe)				
exp264	2025-03-10T10:02:08.477Z	kernelCTF{v1:mitigation-v3b-6.1.55:1741600741}	0-day					
exp263	2025-03-10T05:24:57.276Z	kernelCTF{v1:lts-6.6.80:1741584289}	0-day	(dupe)				
exp262	2025-03-10T05:23:23.919Z	kernelCTF{v1:lts-6.6.80:1741584196}	0-day	(dupe)				
exp261	2025-03-10T05:21:36.795Z	kernelCTF{v1:lts-6.6.80:1741583786}	0-day	(dupe)				
exp260	2025-03-10T05:11:38.272Z	kernelCTF{v1:lts-6.6.80:1741582061}	0-day	(dupe)				
exp259	2025-03-10T05:06:39.671Z	kernelCTF{v1:lts-6.6.80:1741582061}	0-day	(dupe)				
exp258	2025-03-10T04:55:57.284Z	kernelCTF{v1:lts-6.6.80:1741582061}	0-day	(dupe)				
exp257	2025-03-10T04:49:16.310Z	kernelCTF{v1:lts-6.6.80:1741582061}	0-day	(dupe)				
exp256	2025-03-08T00:29:07.498Z	kernelCTF{v1:cos-109-17800.436.33:1740495359}	1-day		(dupe)		09cdafb08fba004cb12ec	netfilter: allow exp not to be removed in nf_ct_find_expectation	CVE-2023-52927
exp255	2025-03-07T12:00:28.657Z	kernelCTF{v1:lts-6.6.80:1741348804}	0-day	(dupe)					
exp254	2025-03-07T12:00:30.295Z	kernelCTF{v1:lts-6.6.80:1741348816}	0-day	(dupe)					
exp253	2025-03-07T12:00:27.042Z	kernelCTF{v1:lts-6.6.80:1741348809}	0-day	(dupe)					
exp252	2025-03-07T12:00:12.484Z	kernelCTF{v1:lts-6.6.80:1741348802}	0-day	(dupe)					
exp251	2025-03-07T12:00:10.974Z	kernelCTF{v1:lts-6.6.80:1741348802}	0-day	lts-6.6.80					
exp270	2025-03-03T13:40:05.909Z	kernelCTF{v1:cos-105-17412.535.55:1741008705}	0-day		cos-105-17412.535.5				
exp269	2025-02-27T05:35:59.629Z	kernelCTF{v1:cos-105-17412.535.55:1740634203}	0-day		(revoked in favor of e				
		kernelCTF{v1:cos-109-17800.436.33:1740550876}							
exp268	2025-02-26T06:26:50.565Z	kernelCTF{v1:mitigation-v3b-6.1.55:1740551007}	0-day		(dupe)		120b3d0ec9c79568fceb9c921e		
exp267	2025-02-25T15:22:04.926Z	kernelCTF{v1:cos-109-17800.436.33:1740495359}	1-day		cos-109-17800.436.3				
exp266	2025-02-22T00:52:00.277Z	kernelCTF{v1:lts-6.6.77:1740185407}	0-day	(dupe)					
exp265	2025-02-21T13:12:27.854Z	kernelCTF{v1:cos-105-17412.535.55:1740143473}	1-day		(vuln dupe of exp248		e49af10427459c2de35f71	netem: Update sch->q.qlen before qdisc_tree_reduce_backlog()	CVE-2025-21703
exp264	2025-02-21T13:11:28.581Z	kernelCTF{v1:cos-109-17800.436.33:1740143455}	1-day		(vuln dupe of exp248		e49af10427459c2de35f71	netem: Update sch->q.qlen before qdisc_tree_reduce_backlog()	CVE-2025-21703
exp263	2025-02-21T12:00:47.718Z	kernelCTF{v1:lts-6.6.77:1740139205}	0-day	(dupe)					
exp262	2025-02-21T12:00:37.246Z	kernelCTF{v1:lts-6.6.77:1740139203}	0-day	(dupe)					
exp261	2025-02-21T12:00:27.723Z	kernelCTF{v1:lts-6.6.77:1740139204}	0-day	lts-6.6.77			120b3d0ec9c79568fceb9c921e		
exp260	2025-02-13T16:38:31.110Z	kernelCTF{v1:cos-105-17412.535.33:1739464192}	1-day		(vuln dupe of exp224		ec2b392fb5544f250fb6f	net: sched: Disallow replacing of child qdisc from one parent to another	CVE-2025-21700
exp259	2025-02-13T07:31:08.024Z	kernelCTF{v1:cos-105-17412.535.34:1739431609}	1-day		(vuln dupe of exp237		4e1404315c245ae56322a	vsock: Keep the binding until socket destruction	CVE-2025-21756
exp258	2025-02-11T09:28:26.298Z	kernelCTF{v1:mitigation-v3b-6.1.55:1739268941}	1-day				271e018d5d15d998b629d	pfifo_tail_enqueue: Drop new packet when sch->limit == 0	CVE-2025-21702
exp257	2025-02-10T07:08:14.970Z	kernelCTF{v2:lts-6.6.75:io_uring:1739151644}	0-day	(io_uring LTS promotic			1ac43365c20142e85603	io_uring/kbuf: reallocate buf lists on upgrade	CVE-2025-21836
exp256	2025-02-10T02:20:23.829Z	kernelCTF{v2:lts-6.6.75:io_uring:1739151644}	0-day	(revoked in favor of ex					
exp255	2025-02-08T04:10:28.252Z	kernelCTF{v2:lts-6.6.66:io_uring:1738987489}	1-day	(io_uring LTS promotic			c27fc6a783c8a77c756dd5461b15e465020d075	io_uring/rw: split io_read() into a helper	CVE-2023-52926
exp254	2025-02-07T12:42:08.288Z	kernelCTF{v1:cos-109-17800.436.14:1738932066}	1-day		(dupe)				
exp253	2025-02-07T12:03:24.982Z	kernelCTF{v1:cos-109-17800.436.14:1738929777}	0-day		(dupe)				
exp252	2025-02-07T12:01:58.784Z	kernelCTF{v1:cos-109-17800.436.14:1738929657}	1-day		cos-109-17800.436.1		1677fd260af5a305316a1f	net: avoid race between device unregistration and ethnl ops	CVE-2025-21701
exp251	2025-02-07T12:01:38.535Z	kernelCTF{v1:lts-6.6.75:1738929615}	0-day	(dupe)					
exp250	2025-02-07T12:00:56.011Z	kernelCTF{v1:lts-6.6.75:1738929604}	0-day	(dupe)					
exp249	2025-02-07T12:00:39.948Z	kernelCTF{v1:lts-6.6.75:1738929605}	1-day	lts-6.6.75			4e1404315c245ae56322a	vsock: Keep the binding until socket destruction	CVE-2025-21756
exp248	2025-01-24T12:23:22.338Z	kernelCTF{v1:cos-109-17800.372.99:1737720991}	0-day		(dupe, but eligible be		e49af10427459c2de35f71	netem: Update sch->q.qlen before qdisc_tree_reduce_backlog()	CVE-2025-21703
exp247	2025-01-24T12:02:56.585Z	kernelCTF{v1:lts-6.6.71:1737720127}	0-day	(dupe)					
exp246	2025-01-24T12:02:44.060Z	kernelCTF{v1:cos-109-17800.372.99:1737720140}	1-day		(dupe)				
exp245	2025-01-24T12:02:03.079Z	kernelCTF{v1:cos-109-17800.372.99:1737720030}	0-day		cos-109-17800.372.9				
exp244	2025-01-24T12:01:30.626Z	kernelCTF{v1:lts-6.6.71:1737720004}	0-day	(dupe)					
exp243	2025-01-24T12:01:20.769Z	kernelCTF{v1:lts-6.6.71:1737720008}	1-day	(dupe)					
exp242	2025-01-24T12:01:20.206Z	kernelCTF{v1:lts-6.6.71:1737720005}	1-day	(dupe)			4e1404315c245ae56322a	vsock: Keep the binding until socket destruction	CVE-2025-21756
exp241	2025-01-24T12:01:16.371Z	kernelCTF{v1:lts-6.6.71:1737720004}	0-day	lts-6.6.71			1677fd260af5a305316a1f	net: avoid race between device unregistration and ethnl ops	CVE-2025-21701
exp240	2025-01-24T11:48:41.659Z	kernelCTF{v1:mitigation-v3b-6.1.55:1737719308}	0-day						
	2025-01-24T06:11:58.028Z	invalid flag (signature error)	0-day						
exp239	2025-01-17T12:01:41.244Z	kernelCTF{v1:lts-6.6.69:1737115282}	0-day	(dupe)					
exp238	2025-01-17T11:36:45.979Z	kernelCTF{v1:lts-6.6.69:1737113789}	0-day	(dupe)					
exp237	2025-01-17T08:54:30.881Z	kernelCTF{v1:cos-105-17412.495.75:1737102254}	1-day		cos-105-17412.495.75	https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=fcdd2242c0231032fc84e1404315c245ae56322a		vsock: Keep the binding until socket destruction	CVE-2025-21756
exp236	2025-01-17T05:42:43.856Z	kernelCTF{v2:cos-109-17800.372.84:io_uring:1737	0-day		(io_uring LTS promotion	https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/io_uring?h=linux-6.6.y&id=a94592ec30ff67dc36c424327f1e0a9ceeeb9bd3		io_uring: fix io_req_prep_async with provided buffers	
exp235	2025-01-17T01:28:11.087Z	kernelCTF{v2:lts-6.6.69:io_uring:1737076911}	0-day	(io_uring LTS promotic		https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/io_uring?h=linux-6.6.y&id=a94592ec30ff67dc36c424327f1e0a9ceeeb9bd3		io_uring: fix io_req_prep_async with provided buffers	
exp234	2025-01-10T12:01:45.109Z	kernelCTF{v1:cos-109-17800.372.84:1736510422}	0-day		cos-109-17800.372.84	https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=647cef20e649c576dff271e018d5d15d998b629d		pfifo_tail_enqueue: Drop new packet when sch->limit == 0	CVE-2025-21702
exp233	2025-01-10T12:01:00.219Z	kernelCTF{v1:lts-6.6.69:1736510410}	0-day	(dupe)					

Source: Google kernelCTF

(https://docs.google.com/spreadsheets/d/e/2PACX-1vS1REdTA29OJftst8xN5B5x8ilUcxuK6bXdzF8G1UXCmRtoNsoQ9MbebdRdFnj6qZ0Yd7LwQfvYC2oF/pubhtml#)

IMPACT OF KERNEL VULNERABILITIES



Source: Bonan's blog post

(https://blog.wohin.me/posts/thoughts-on-vuln-research-2)

REPRODUCTION!

Severity Assessment

Application Scenarios

Detection & Mitigation

Defense Evaluation

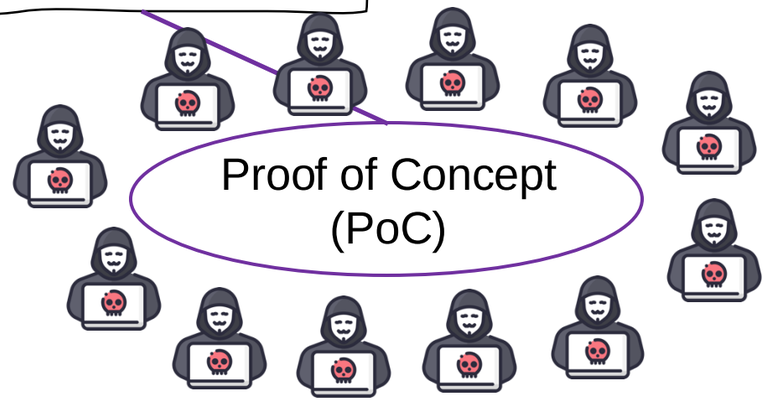
Vulnerable Environment

Brain

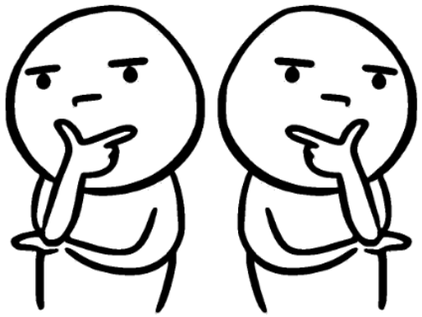
Patient

What Do We Need for Reproduction?

Luck



Proof of Concept (PoC)



DON’T TAKE SUCCESSFUL ENV FOR GRANTED!

Hello, when building the test environment, I followed the steps above to compile the kernel... It kept getting stuck... During the test, I didn't find any 'NFQUEUE' rule in the target...

At the time, I selected many configs, and it's possible that some configs were not included. First, check if it's an issue with the compilation options...

EXAMPLE: CVE-2021-22555

Description:
A heap out-of-bounds write affecting Linux since v2.6.19-rc1 was discovered in net/netfilter/x_tables.c. This allows an attacker to gain privileges or cause a DoS (via heap memory corruption) through user name space.

Report Date: 2021-04-06
Affected Product: Linux Kernel
CVSS: 7.8 (High)
CWE: CWE-787 (Out-of-bounds Write)
Impact: Privilege Escalation
Exploit: Public

Vulnerable Version Ranges in NVD Database:

[v2.6.19, v4.4.267)	[v4.5, v4.9.267)	
[v4.10, v4.14.231)	[v4.15, v4.19.188)	
[v4.20, v5.4.133)	[v5.5, v5.10.31)	[v5.11, v5.12)

EXAMPLE: CVE-2021-22555

Code Snippet (v5.11.22)

```
void xt_compat_target_from_user(struct xt_entry_target
*t, void **dstptr, unsigned int *size) {
    // ... omitted ...
        target->compat_from_user(t->data, ct->data);
    else
        memcpy(t->data, ct->data, tsize - sizeof(*ct));

    tsize += off;
    t->u.user.target_size = tsize;
```

Patch Snippet

```
@@ -1126,9 +1123,6 @@ void xt_compat_target_from_user(struct
xt_entry_target *t, void **dstptr,
        target->compat_from_user(t->data, ct->data);
    else
        memcpy(t->data, ct->data, tsize - sizeof(*ct));
-   pad = XT_ALIGN(target->targetsize) - target->targetsize;
-   if (pad > 0)
-       memset(t->data + target->targetsize, 0, pad);

    tsize += off;
    t->u.user.target_size = tsize;
```

NVD Version Ranges

```
[v2.6.19, v4.4.267)
[v4.5, v4.9.267)
[v4.10, v4.14.231)
[v4.15, v4.19.188)
[v4.20, v5.4.133)
[v5.5, v5.10.31)
[v5.11, v5.12)
```

v5.11.22 seems to be vulnerable

but already patched!

UM...

You can’t wake a person who is pretending to be asleep.

You can’t trigger a vulnerability which has been patched.

EXAMPLE: CVE-2021-22555

Patch Snippet

```
diff --git a/net/netfilter/x_tables.c ...
index 6bd31a7a27fc58..92e9d4ebc5e8d7 100644
--- a/net/netfilter/x_tables.c
+++ b/net/netfilter/x_tables.c
@@ -1126,9 +1123,6 @@ void xt_compat_target_from_user(...
         target->compat_from_user(t->data, ct->data);
     else
         memcpy(t->data, ct->data, tsize - sizeof(*ct));
-    pad = XT_ALIGN(target->targetsize) - target->targetsize;
-    if (pad > 0)
-        memset(t->data + target->targetsize, 0, pad);
...
```

Related Makefiles

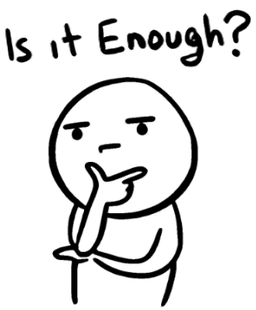
```
obj-$(CONFIG_NETFILTER)            += netfilter/
obj-$(CONFIG_NETFILTER_XTABLES) += x_tables.o
```

Vulnerable Code Snippet

```
#ifdef CONFIG_COMPAT
...
void xt_compat_target_from_user(...
    ...
        target->compat_from_user(t->data, ct->data);
    else
        memcpy(t->data, ct->data, tsize - sizeof(*ct));
    pad = XT_ALIGN(target->targetsize) - target->targetsize;
    if (pad > 0)
        memset(t->data + target->targetsize, 0, pad);
...
```

Temporary Results

```
CONFIG_COMPAT
CONFIG_NETFILTER_XTABLES
CONFIG_NETFILTER
```



EXAMPLE: CVE-2021-22555

Temporary Results

```
CONFIG_COMPAT
CONFIG_NETFILTER_XTABLES
CONFIG_NETFILTER
```

Related Kconfig Files

```
[net/netfilter/Kconfig]
menu "Core Netfilter Configuration"
    depends on NET && INET && NETFILTER
... omitted ...
config NETFILTER_XTABLES
```

```
[net/Kconfig]
if NET
config INET
... omitted ...
menuconfig NETFILTER
```

Heuristic Analysis Result of Configs

```
CONFIG_COMPAT          CONFIG_NETFILTER_XTABLES      CONFIG_NETFILTER
CONFIG_NET             CONFIG_NETFILTER_FAMILY_ARP   CONFIG_NETFILTER_ADVANCED
CONFIG_INET            CONFIG_IP_NF_IPTABLES         CONFIG_NLATTR
CONFIG_IPV6            CONFIG_IP_NF_ARPTABLES        CONFIG_GENERIC_NET_UTILS
CONFIG_BPF             CONFIG_IP6_NF_IPTABLES
```

EXAMPLE: CVE-2021-22555

Heuristic Analysis Result of Configs

```
CONFIG_COMPAT          CONFIG_NETFILTER_XTABLES      CONFIG_NETFILTER
CONFIG_NET             CONFIG_NETFILTER_FAMILY_ARP   CONFIG_NETFILTER_ADVANCED
CONFIG_INET            CONFIG_IP_NF_IPTABLES     CONFIG_NLATTR
CONFIG_IPV6            CONFIG_IP_NF_ARPTABLES    CONFIG_GENERIC_NET_UTILS
CONFIG_BPF             CONFIG_IP6_NF_IPTABLES
```

```
CONFIG_NETFILTER_XT_TARGET_NFQUEUE
```

PoC Snippet

```
data.match.u.user.match_size = (sizeof(data.match) + sizeof(data.pad));
strcpy(data.match.u.user.name, "icmp6");
data.match.u.user.revision = 0;
data.target.u.user.target_size = sizeof(data.target);
strcpy(data.target.u.user.name, "NFQUEUE");
data.target.u.user.revision = 1;
```

UM...

Version

You can’t trigger a vulnerability which has been patched.

You can’t trigger a vulnerability which doesn’t exist or is inaccessible.

Config

Bingo!

Patch

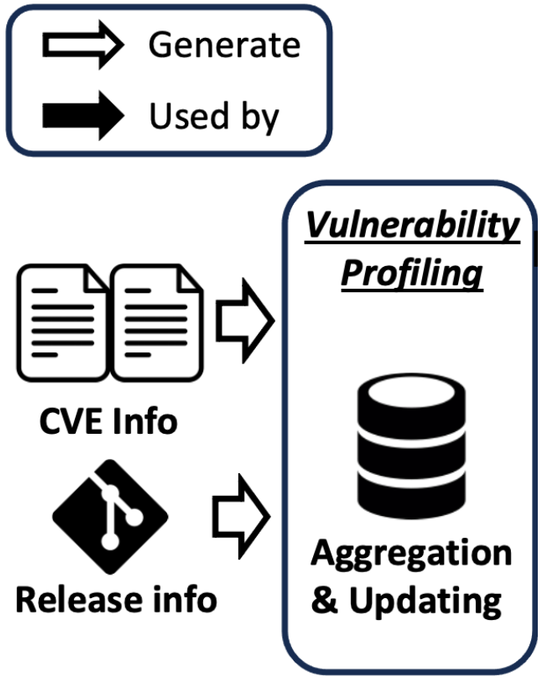
The presence of patch implies the absence of vulnerability.

Kernel configs can be regarded as a graph.

Kconfig and Kbuild mechanisms work in tandem to tailor the kernel.

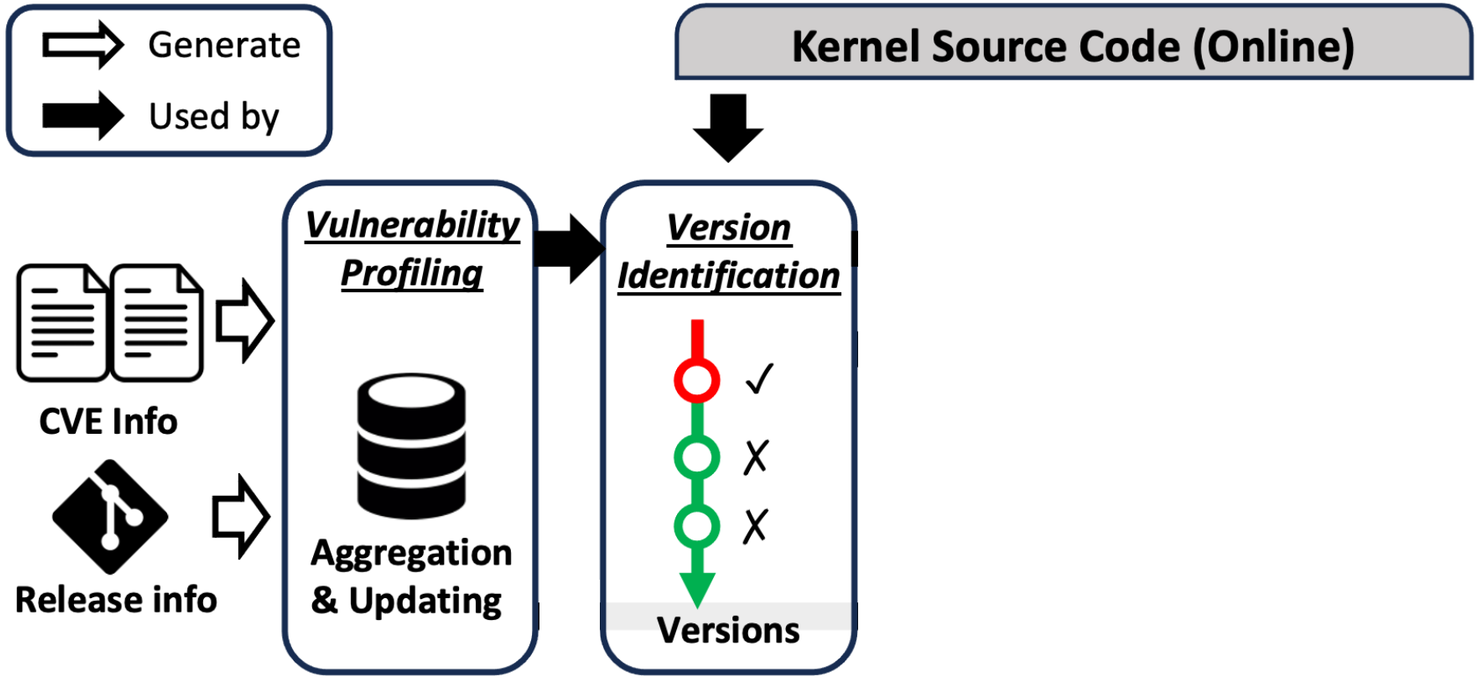
Graph

OVERVIEW OF KERNJC



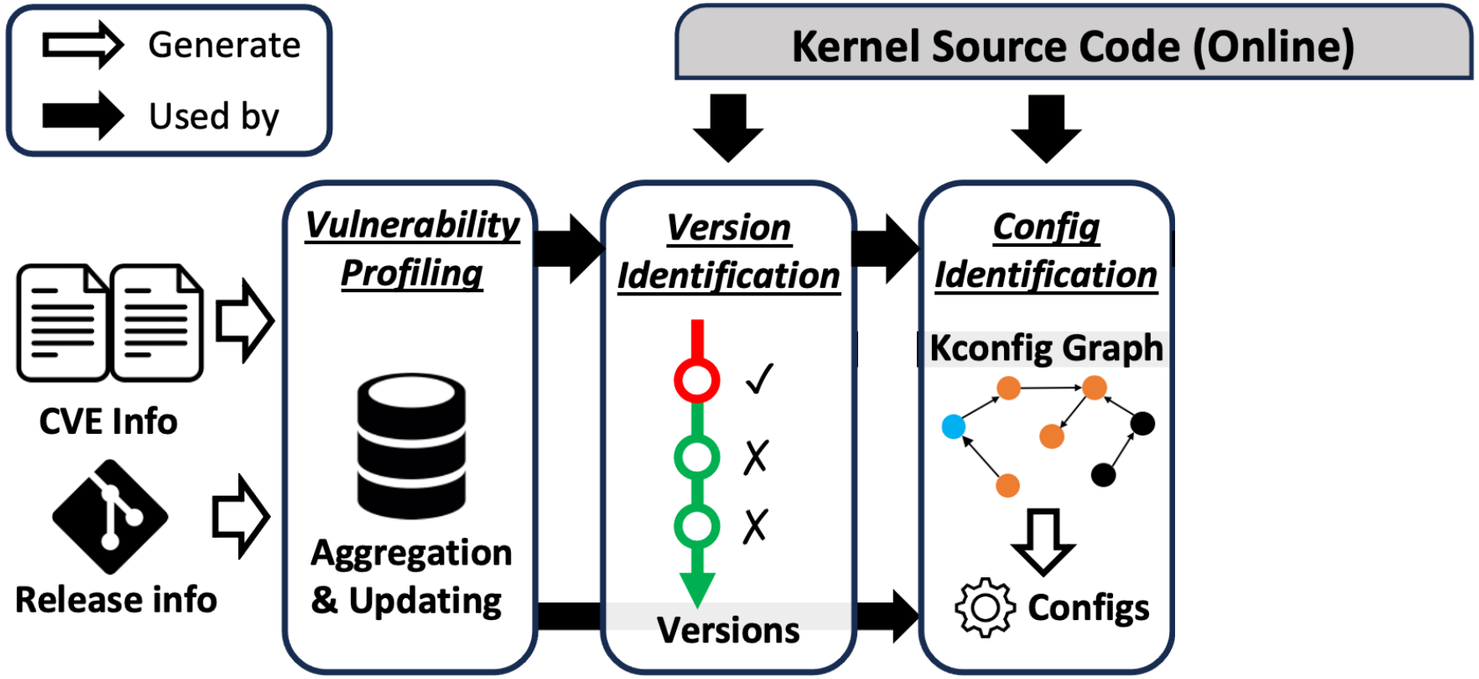
- **Vulnerability Profiling:** Collect vulnerability information for later usage.

OVERVIEW OF KERNJC



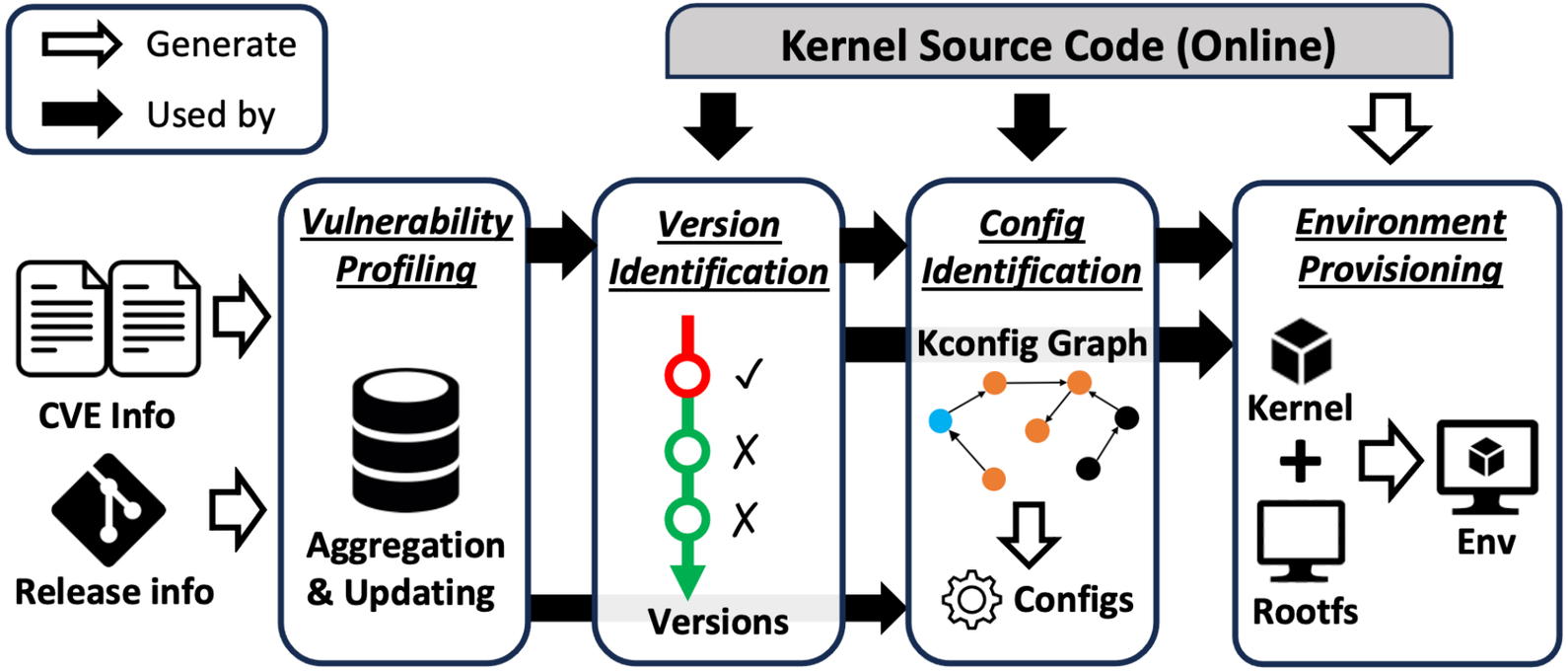
- **Vulnerability Profiling:** Collect vulnerability information for later usage.
- **Version Identification:** Perform patch operation to detect patch presence.

OVERVIEW OF KERNJC



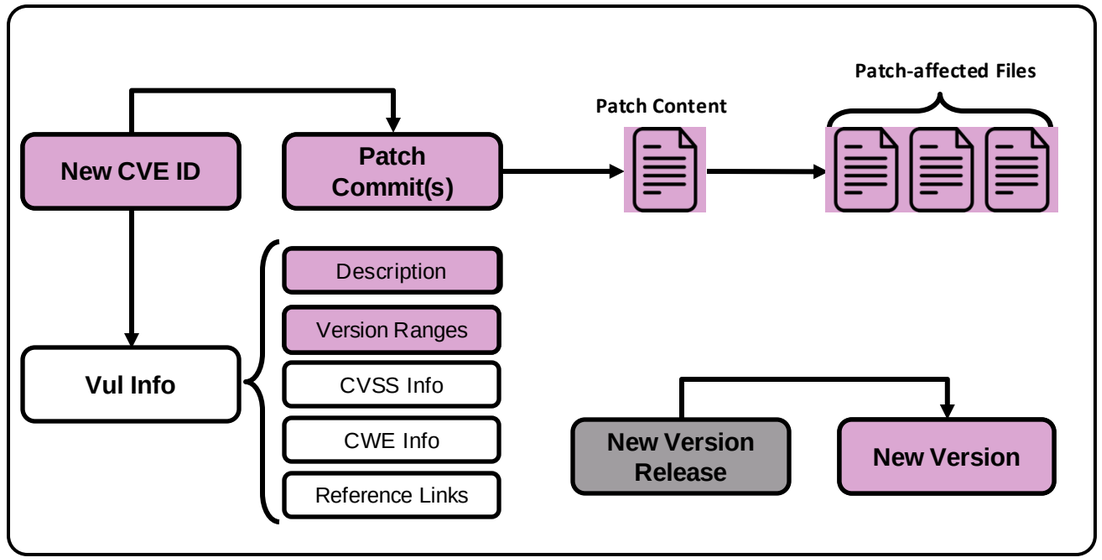
- **Vulnerability Profiling:** Collect vulnerability information for later usage.
- **Version Identification:** Perform patch operation to detect patch presence.
- **Config Identification:** Build Kconfig graph and mine reachable configs.

OVERVIEW OF KERNJC



- **Vulnerability Profiling:** Collect vulnerability information for later usage.
- **Version Identification:** Perform patch operation to detect patch presence.
- **Config Identification:** Build Kconfig graph and mine reachable configs.
- **Environment Provisioning:** Build the kernel and provision the virtual machine.

VULNERABILITY PROFILING



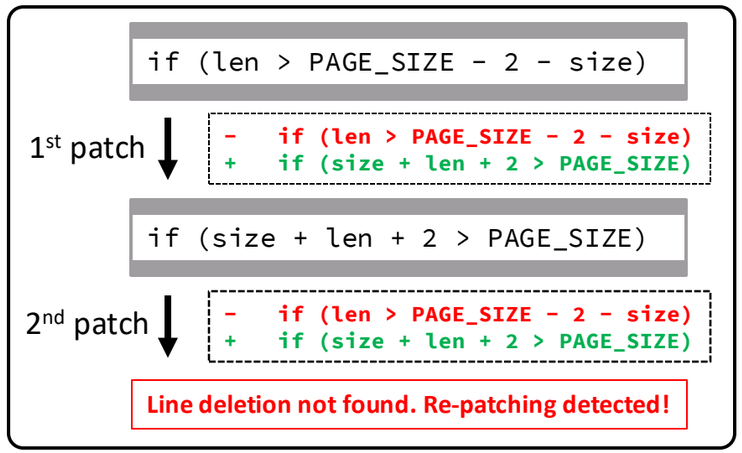

```
cve: CVE-2022-0847
patch:
- 9d2231c5d74e13b2a0546fee6737ee4446017903

diff --git a/lib/iov_iter.c b/lib/iov_iter.c
index b0e0acdf96c15e..6dd5330f7a9957 100644
--- a/lib/iov_iter.c
+++ b/lib/iov_iter.c
@@ -414,6 +414,7 @@ static size_t
copy_page_to_iter_pipe(struct page *page,
size_t offset, size_t by
         return 0;
     buf->ops = &page_cache_pipe_buf_ops;
+    buf->flags = 0;
...

file: lib/iov_iter.c
```

A flaw was found in the way the "flags" member of the new pipe buffer structure was lacking proper initialization in copy_page_to_iter_pipe and push_pipe functions in the Linux kernel and could thus contain stale values. An unprivileged local user could use this flaw to write to pages in the page cache backed by read only files and as such escalate their privileges on the system.

```
[v5.8, v5.10.102)
[v5.15, v5.15.25)
[v5.16, v5.16.11)
...
```

```
    v6.9.1
    v6.9.2
    v6.9.3
+   v6.9.4
+   v6.9.5
+   v6.9.6
```

VUL VERSION IDENTIFICATION



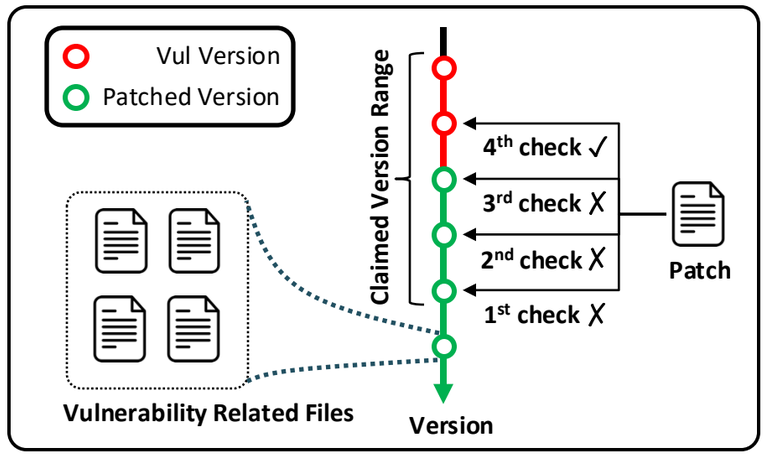
Re-patching Operation

1. Apply the patch on vulnerable file

>>> The vulnerability is patched successfully

2. Apply the patch once again

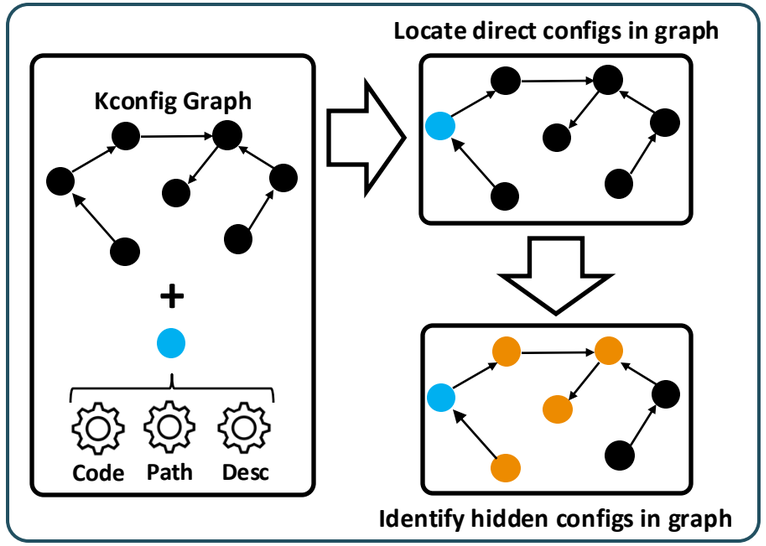
>>> Fail to locate the vulnerable site

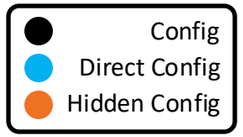


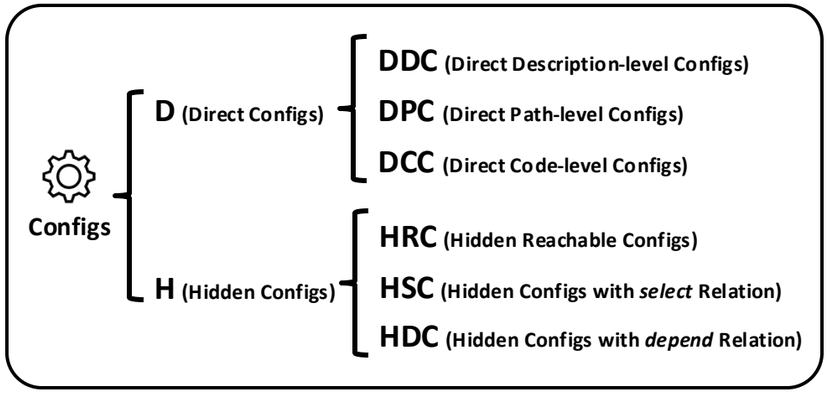
Identification Process

1. Locate the latest vulnerable version *v* claimed by NVD
2. Start from *v* and move downwards along the kernel version list
   * Apply the patch on vulnerability related files of each version
   * Stop when no re-patching occurs

VUL CONFIG IDENTIFICATION







Identification Process

1. Build the Kconfig graph for target kernel
2. Gather direct configs ($D = DDC \cup DPC \cup DCC$)
3. For each config *c* in *D*
   * Locate *c* in the Kconfig graph
   * Discover hidden configs for *c* ($H_c = HRC \cup HSC \cup HDC$)
4. Collect all hidden configs

VUL CONFIG IDENTIFICATION

CVE-2017-18344

The timer_create syscall implementation in kernel/time/posix-timers.c in the Linux kernel before 4.14.8 doesn't properly validate the sigevent->sigev_notify field, which leads to out-of-bounds access in the show_timer function (called when /proc/$PID/timers is read). This allows userspace applications to read arbitrary kernel memory (on a kernel built with **CONFIG_POSIX_TIMERS** and **CONFIG_CHECKPOINT_RESTORE**).

CVE-2021-22555

```
#ifdef CONFIG_COMPAT
...
void xt_compat_target_from_user(...
    ...
        target->compat_from_user(t->data, ct->data);
    else
        memcpy(t->data, ct->data, tsize - sizeof(*ct));
    pad = XT_ALIGN(target->targetsize) - target->targetsize;
    if (pad > 0)
        memset(t->data + target->targetsize, 0, pad);
...
```

CVE-2021-22555

```
diff --git a/net/netfilter/x_tables.c b/net/netfilter/x_tables.c
index 6bd31a7a27fc58..92e9d4ebc5e8d7 100644
--- a/net/netfilter/x_tables.c
+++ b/net/netfilter/x_tables.c
```

```
net/Makefile:           obj-$(CONFIG_NETFILTER)         += netfilter/
net/netfilter/Makefile: obj-$(CONFIG_NETFILTER_XTABLES) += x_tables.o
```

Direct Config Examples

1. Description-level configs from CVE description of CVE-2017-18344
2. Path-level configs from patches for CVE-2021-22555
3. Code-level configs from vulnerable source code of CVE-2021-22555

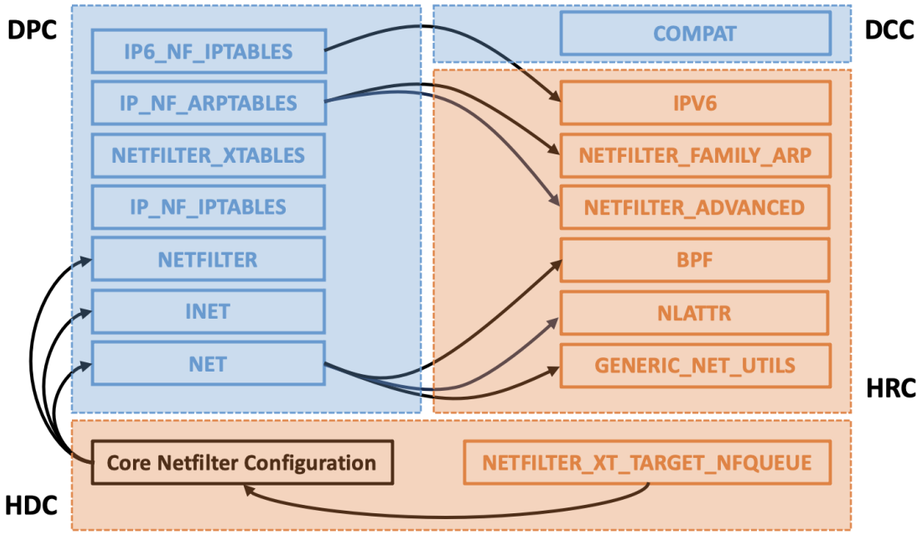
VUL CONFIG IDENTIFICATION

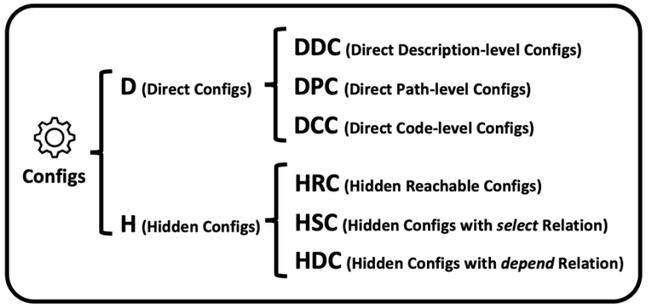
Manual Identification

```
CONFIG_COMPAT          CONFIG_NETFILTER_XTABLES      CONFIG_NETFILTER
CONFIG_NET             CONFIG_NETFILTER_FAMILY_ARP   CONFIG_NETFILTER_ADVANCED
CONFIG_INET            CONFIG_IP_NF_IPTABLES         CONFIG_NLATTR
CONFIG_IPV6            CONFIG_IP_NF_ARPTABLES        CONFIG_GENERIC_NET_UTILS
CONFIG_BPF             CONFIG_IP6_NF_IPTABLES
```

```
CONFIG_NETFILTER_XT_TARGET_NFQUEUE
```

KernJC's Identification





DOCKER-LIKE INTERACTION!

```
$ ./kjc -h
usage: kjc [-

KernJC - A L

optional arg
  -h, --help
  -v, --vers

subcommands:
  {update,bu
    update
    build
    start
    stop
    attach
    exec
    cp
    logs
    rm
    ps
    enter
    info
    query
```

```
$ ./kjc build CVE-2016-10150
[*] Removing potential
[+] Auto-selected kern
[*] Initializ
[*] Downloadi
    100%|
[*] Decompres
[*] Building
[*] Applying
[*] Loading c
[*] Generatin
[*] Finding k
[*] Building
[+] Built kcf
[+] Found 37
[*] Loading c
[!] Vuln conf
[*] Merging c
[+] Applied custom con
... kernel compilation
[+] Built kernel source code
[*] Preparing rootfs (
[+] Env a30ebfa6f5747f
```

```
$ ./kjc start --enable-kvm a3
[*] Starting env a3
[+] Started env a30ebfa6
```

```
$ ./kjc
+------------------+-------------
| ID
+-------
| a30ebf
+-------
$ ./kjc
{'create
 'cve': 'CVE-2016-10150',
 'ip': N
 'kernel
 'pid':
 'port':
 'status
```

```
$ ./kjc exec a3 /home/user/poc
Warning: Permanently added '[localhost]:10000' (ECDSA) to the list of known hosts.
```

```
$ cd db/pocs/cve-2016-10150/; gcc -o poc poc.c -static; cd -
~/pjts/KernJC
$ ./kjc cp db/pocs/cve-2016-10150/poc a3:/home/user/
Warning: Permanently adde
poc
```

```
$ ./kjc logs -f a3
[  OK  ] Reached target
         Starting Update
[  OK  ] Finished Update

Debian GNU/Linux 11 kern
... output omitted ...
```

```
$ ./kjc logs -f a3
... output omitted ...
[  408.497181] ================================================================
[  408.498170] BUG: KASAN: use-after-free in kvm_vm_ioctl+0x1150/0x1340 at addr ffff8800
[  408.498170] Read of size 8 by task poc/2983
[  408.498170] CPU: 1 PID: 2983 Comm: poc Tainted: G    B           4.8.12 #1
[  408.498170] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.10.2-1ubunt
[  408.498170]  0000000000000097 ffff88006118faf0 ffffffff81bfe5a2 ffff88006cc018c0
[  408.498170]  ffff88006b8c9a20 ffff88006b8c9a60 ffffffff83a46400 ffff88006118fb18
[  408.498170]  ffffffff815c8cbc ffff88006118fba8 ffff88006b8c9a20 ffff88006cc018c0
[  408.498170] Call Trace:
... output omitted ...
```

```
$ ./kjc attach a3
... output omitted ...
user@kernjc:~$ su # password: neo
Password:
root@kernjc:/home/user#
Adding user `user' to gr
Adding user user to grou
Done.
```

```
$ ./kjc rm --force a3
[+] Env a30ebfa6f5747fa9 removed
```

iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help

```
(venv) ➜  KernJC git:(main) ✗ ./kjc build CVE-2021-22555
[*] Building environment for CVE-2021-22555
```

EXPERIMENTAL RESULTS

Reproduction Performance

RwKC: Reproducibility with KernJC-identified Configs
RwDC: Reproducibility with Default Configs FPV: False Positive Version claims in NVD

CVE	RwKC?	RwDC?	FPV?
2016-10150	✓	✗	✗
2016-4557	✓	✗	✗
2016-6187	✓	✗	✗
2017-16995	✓	✗	✗
2017-18344	✓	✗	✗
2017-2636	✓	✗	✗
2017-6704	✓	✗	✗
2017-8824	✓	✗	✗

CVE	RwKC?	RwDC?	FPV?
2018-12233	✓	✗	✗
2018-5333	✓	✗	✗
2018-6555	✓	✗	✗
2019-6974	✓	✗	✗
2020-14381	✓	✓	✓
2020-16119	✓	✗	✗
2020-25656	✓	✓	✓
2020-25669	✓	✗	✗
2022-34918	✓	✗	✗

CVE	RwKC?	RwDC?	FPV?
2020-27194	✓	✗	✗
2020-27830	✓	✗	✗
2020-28941	✓	✗	✗
2020-8835	✓	✗	✗
2021-22555	✓	✗	✓
2021-26708	✓	✗	✗
2021-27365	✓	✗	✗
2021-34866	✓	✗	✗
2023-32233	✓	✗	✗

CVE	RwKC?	RwDC?	FPV?
2021-3490	✓	✓	✗
2021-3573	✓	✗	✓
2021-42008	✓	✗	✗
2021-43267	✓	✗	✗
2022-0995	✓	✗	✗
2022-1015	✓	✗	✗
2022-25636	✓	✗	✗
2022-32250	✓	✗	✗

- KernJC successfully builds reproduction environments for all 66 vulnerabilities.
- 4 of 66 are detected to have incorrect (FP) version claims in NVD.
- 32 of 66 need non-default configs identified by KernJC to be activated.

EXPERIMENTAL RESULTS

Vulnerability Config Identification Statistics

CVE	Subsystem	DDC	DPC	DCC	HRC	HSC	HDC	CVE	Subsystem	DDC	DPC	DCC	HRC	HSC	HDC
CVE-2016-10150	KVM	0	1	0	39	0	4	CVE-2020-28941	Accessibility	0	2	0	19	0	0
CVE-2016-4557	eBPF	0	1	0	0	2	0	CVE-2020-8835	eBPF	0	1	0	0	2	1
CVE-2016-6187	AppArmor	0	1	0	14	0	2	CVE-2021-22555	Netfilter	0	7	1	10	3	406
CVE-2017-16995	eBPF	0	1	0	0	2	0	CVE-2021-26708	VSOCK	0	1	0	4	0	6
CVE-2017-18344	Time	2	0	0	3	0	3	CVE-2021-27365	SCSI	0	2	0	22	8	0
CVE-2017-2636	TTY	0	1	0	17	0	0	CVE-2021-34866	eBPF	0	1	0	0	2	3
CVE-2017-6074	DCCP	0	1	0	9	0	0	CVE-2021-3490	eBPF	0	1	0	0	2	2
CVE-2017-8824	DCCP	0	1	0	9	0	0	CVE-2021-3573	Bluetooth	0	1	0	32	0	45
CVE-2018-12233	JFS	0	1	0	4	0	4	CVE-2021-42008	NET	0	2	0	18	0	14
CVE-2018-5333	RDS	0	1	0	9	0	3	CVE-2021-43267	TIPC	0	1	0	5	0	4
CVE-2018-6555	IRDA	0	2	1	7	0	37	CVE-2022-0995	WQ	0	1	1	0	0	1
CVE-2019-6974	KVM	0	1	0	42	0	4	CVE-2022-1015	Netfilter	0	1	0	4	0	241
CVE-2020-16119	DCCP	0	1	0	5	0	0	CVE-2022-25636	Netfilter	0	4	0	19	2	241
CVE-2020-25669	Input	0	3	0	3	37	3	CVE-2022-32250	Netfilter	0	1	0	4	0	238
CVE-2020-27194	eBPF	0	1	0	0	2	1	CVE-2022-34918	Netfilter	0	1	0	4	0	238
CVE-2020-27830	Accessibility	0	2	0	19	0	0	CVE-2023-32233	Netfilter	0	2	0	5	0	317

EXPERIMENTAL RESULTS

Vulnerabilities with FP Version Range Claims in NVD (TOP 10)

We identify 128 vulnerabilities with incorrect version claims in NVD.

The aggregate count of incorrect (FP) versions is 3,042.

Averaging 24 incorrect versions per identified vulnerability.

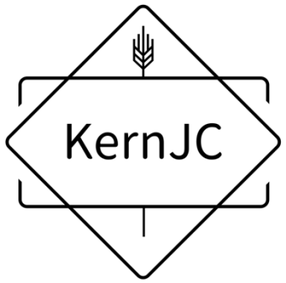
CVE	CVSS	FP Version Range	Vulnerable Version	FP Count
CVE-2017-1000407	7.4	v4.14.6 – v4.14.325	v4.14.5	320
CVE-2017-18216	5.5	v4.14.57 – v4.14.325	v4.14.56	269
CVE-2017-18224	4.7	v4.14.57 – v4.14.325	v4.14.56	269
CVE-2020-35508	4.5	v5.9.7 – v5.11.22	v5.9.6	229
CVE-2021-4002	4.4	v5.15.5 – v5.15.132	v5.15.4	128
CVE-2021-4090	7.1	v5.15.5 – v5.15.132	v5.15.4	128
CVE-2022-0264	5.5	v5.15.11 – v5.15.132	v5.15.10	122
CVE-2021-4155	5.5	v5.15.14 – v5.15.132	v5.15.13	119
CVE-2016-10906	7.0	v4.4.191 – v4.4.302	v4.4.190	112
CVE-2015-4170	4.7	v3.12.7 – v3.13.3	v3.12.6	72

KERNJC

Source: ChatGPT

KernJC = Kernel *JiaoChang*

JiaoChang, in ancient China, referred to a site dedicated to military training and competition.



Jiao Chang
/dʒaʊ tʃɑːŋ/

https://github.com/NUS-CURIOSITY/KernJC