Website-DRM: Devtools blocken als Chromium-Feature

Dario Bartussek

November 16, 2024

Was ist Devtools Detection?

https://dbartussek.github.io/devtools_detection/

Was ist Devtools Detection?

https://dbartussek.github.io/devtools_detection/

What gave your devtools away?

The stack method (Chrome)

```
const e = new Error();
Object.defineProperty(e, 'stack', {
    get() {
        report();
    }
});
console.log(e);
```

The size method (Chrome/Firefox, devtools are not undocked (Also detects if your browser has 2000s levels of toolbars!))

```
function checkSize() {
    const pixelRatio = window.devicePixelRatio || 1;
    if (
        (window.outerWidth  - window.innerWidth)  * pixelRatio > 200 ||
        (window.outerHeight - window.innerHeight) * pixelRatio > 300
    ) {
        report();
    }
}
```

The Error.message method (Chrome/Firefox)

```
console.log(Object.defineProperties(new Error, {
    message: {
        get() {
            report()
        },
        toString: {},
    },
}));
```

Was ist Devtools Detection?

https://dbartussek.github.io/devtools_detection/

What gave your devtools away?

The stack method (Chrome)

```
const e = new Error();
Object.defineProperty(e, 'stack', {
    get() {
        report();
    }
});
console.log(e);
```

The size method (Chrome/Firefox, devtools are not undocked (Also detects if your browser has 2000s levels of toolbars!))

```
function checkSize() {
    const pixelRatio = window.devicePixelRatio || 1;
    if (
        (window.outerWidth  - window.innerWidth)  * pixelRatio > 200 ||
        (window.outerHeight - window.innerHeight) * pixelRatio > 300
    ) {
        report();
    }
}
```

The Error.message method (Chrome/Firefox)

```
console.log(Object.defineProperties(new Error, {
    message: {
        get() {
            report()
        },
        toString: {},
    },
}));
```

Wozu wird Devtools Detection verwendet?

Wozu wird Devtools Detection verwendet?

1. DRM

Wozu wird Devtools Detection verwendet?

1. DRM
2. Malware

Wozu wird Devtools Detection verwendet?

1. DRM
2. Malware

Am beliebtesten (bisher) auf fragwürdigen Streamingseiten als möglicher 2 for 1 Deal!

Wozu wird Devtools Detection verwendet?

1. DRM
2. Malware

Am beliebtesten (bisher) auf fragwürdigen Streamingseiten als möglicher 2 for 1 Deal!

=> gegen Nutzersicherheit und offenes Internet

Wie funktioniert (diese) Devtools Detection?

Seit 2022 in Firefox und seit Mitte 2024 in Chrome:

Wie funktioniert (diese) Devtools Detection?

Seit 2022 in Firefox und seit Mitte 2024 in Chrome:

```
const leaker = new Error();

Object.defineProperties(leaker, {
    message: {
        get() {
            REPORT()
        },
        toString: {},
    },
});

console.log(leaker);
```

Was kann getan werden?

Hier und jetzt:

Was kann getan werden?

Hier und jetzt: Monkey Patching: console.log überschreiben

Was kann getan werden?

Hier und jetzt: Monkey Patching: console.log überschreiben

Verlässliche, langfristige Lösung: Browserentwickler müssen das Problem beheben

Firefox

Immerhin ist der Bug noch offen ¯_(ツ)_/¯

Firefox

Immerhin ist der Bug noch offen ¯_(ツ)_/¯

Open Bug 1789258 Opened 2 years ago Updated 17 days ago

Dev tools detection with console.log via Error.message override

... und ein Mozilla-Angesteller hat letzes Jahr mal auf ne Frage zu Devtools-Erkennung geantwortet

Honza

Employee

04-24-2023 03:34 AM

04-24-2023 03:34 AM

Thank you for the link! We are actively looking into this (and fixed several things in the past).

Chrome

Positive Aussage im Dezember 2016:

ph...@chromium.org <ph...@chromium.org> #9 Dec 19, 2016 06:26PM

That's fair. I agree sites should always be able to be inspected to discourage this sort of malicious behavior. Rather than make this class of console issues not execute client code though I'd be in favor of turning enhanced/more descriptive console logging into an option instead (this seems to be at the root of this set of vulnerabilities DOM element properties, Function.toString, etc). I think it provides significant user value over the more generic `function ()` logging in FF, but this way the user is in control of whether they accept the tradeoff.

Chrome

... gefolgt von einem mäßigen Kompromiss, 2 Jahre später

pf...@chromium.org <pf...@chromium.org> #17 Dec 14, 2018 01:04AM

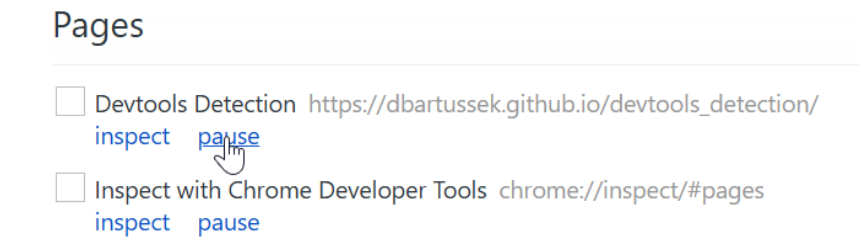
Marked as fixed.

The patch above concludes what we think should be done here. Note that the console.log detection was fixed previously. There is now a way to pause the page before it has a chance to detect devtools. For that you should:

- open chrome://inspect
- select 'Pages'
- click the 'pause' link next to the page you would like to freeze.

Chrome

..... der nicht mal mehr funktioniert



What gave your devtools away?

The stack method (Chrome)

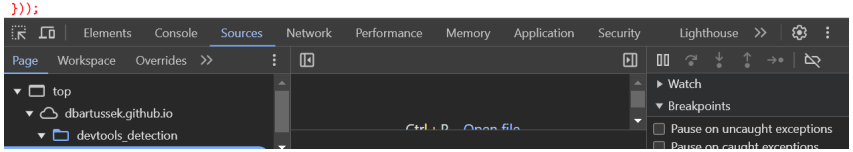
```
const e = new Error();
Object.defineProperty(e, 'stack', {
    get() {
        report();
    }
});
console.log(e);
```

The size method (Chrome/Firefox, devtools are not undocked (Also detects if your browser has 2000s levels of toolbars!))

```
function checkSize() {
    const pixelRatio = window.devicePixelRatio || 1;
    if (
        (window.outerWidth  - window.innerWidth)  * pixelRatio > 200 ||
        (window.outerHeight - window.innerHeight) * pixelRatio > 300
    ) {
        report();
    }
}
```

The Error.message method (Chrome/Firefox)

```
console.log(Object.defineProperties(new Error, {
    message: {
        get() {
            report()
        },
        toString: {},
    },
}));
```



Chrome

Seitdem: Devtools zu erkennen ist Intended Behavior!

Chromium > Platform > DevTools 40092943 Won't fix (Obsolete)

pa...@chromium.org <pa...@chromium.org> #2 Nov 6, 2018 12:20AM

Assigned to dg...@chromium.org.

+DevTools OWNERS.

Is it now/has it ever been a goal to stop web origins from knowing when DevTools is open? If so, we should fix this; if not, we should document that.

Is there any chance of closing any of these 5 side-channels, or any others? It'd sure be nice if we could.

[Monorail components: Platform>DevTools]

dg...@chromium.org <dg...@chromium.org> #3 Nov 6, 2018 12:29AM

Hiding DevTools presence has never been a goal. We try to not affect JS execution by DevTools, but we are calling getters/ToString sometimes for better ergonomics. It's essential to call JS while debugging.

Server-side detection with image urls from console (as in TEST 1) or source mapping urls is possible too.

Where do you think we should document this?

pa...@chromium.org <pa...@chromium.org> #4 Nov 6, 2018 01:15AM

1 possible place to document it would be the Chrome Security FAQ (https://chromium.googlesource.com/chromium/src/+/master/docs/security/faq.md). Before we do that, though, let's have a think in email.

ke...@chromium.org <ke...@chromium.org> #5 Nov 9, 2018 04:21PM

palmer@: Is there any action to be taken here?

I don't think this needs to be called a security bug, since AFAICT this doesn't create any specific risk for users.

pa...@chromium.org <pa...@chromium.org> #6 Nov 9, 2018 09:15PM

Status: Won't Fix (Obsolete)

According to the Dev Tools team, it is not a goal to prevent web origins from knowing if DT is active. So, I'll mark this WontFix because working as intended.

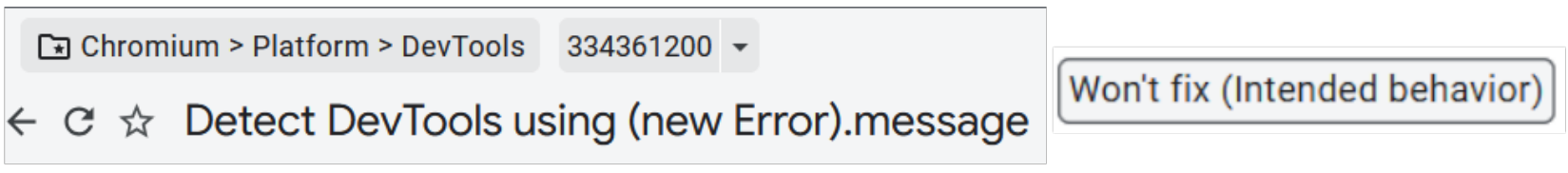
Chromium > Platform > DevTools 327245340 Won't fix (Infeasible)

bm...@google.com <bm...@google.com> #7 Mar 20, 2024 08:12PM

Status: Won't Fix (Infeasible)

Thanks for the report. This is a known issue with no good solution (at the moment). It's specifically related to `Error` objects and their reporting to DevTools front-end. We might come back to this at some later point, but for now this is considered Infeasible.

Chrome



bm...@google.com <bm...@google.com> #4 May 21, 2024 11:49AM

Status: Won't Fix (Intended Behavior)

This was intentionally changed as part of fixing issue 327467399 (https://crrev.com/c/5378709).