Vulntober: Multiple Mobile Browser Address Bar Spoofing Vulnerabilities

Oct 20, 2020 | 9 min read | Tod Beardsley (/blog/author/tod-beardsley/)

Last updated at Tue, 24 Nov 2020 16:04:20 GMT

Today, we're announcing a coordinated vulnerability disclosure publication with our longtime mobile hacker friend, Rafay Baloch. If you'd like to just jump straight to the technical details for these vulnerabilities, I invite you to read his paper here (http://www.rafaybaloch.com/2020/10/multiple-address-bar-spoofing-vulnerabilities.html). If you want to know more about why this vulnerability class matters, read on!

What we're disclosing today is a set of address bar spoofing vulnerabilities that affect a number of mobile browsers, ranging from the more common browsers, like Apple Safari (https://www.apple.com/safari/) and Opera Touch (https://apps.apple.com/us/app/opera-touch-web-browser/id1411869974), to the less common, like Bolt Browser (https://apps.apple.com/us/app/bolt-browser-and-documents/id1366502697) and RITS Browser (https://play.google.com/store/apps/details?id=acr.browser.raisebrowserfull&hl=en_US). Technically, address bar spoofing is an instance of CWE-451 (https://cwe.mitre.org/data/definitions/451.html) from the Common Weakness Enumeration, and tends to be scored around a CVSS 4.3 (https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:L/A:N&version=3.1) or so, which seems like not that big of a deal.

Mobile devices and security sigils

But here's the thing: Mobile browsers are a pretty special sort of software that end up acting as a user's multipass for all types of critical applications in their day-to-day life. Any type of malicious messing with how this application presents itself is kind of a big deal, and can have serious consequences for the user, even if the alterations are relatively minor.

Essentially, if your browser tells you that a pop up notification or a page is "from" your bank, your healthcare provider, or some other critical service you depend on, you really should have some mechanism of validating that source. In mobile browsers, that source begins and ends with the URL as shown in the address bar. The fact of the matter is, we really don't have much else to rely on.

Mobile platforms are notorious for having very little screen real estate for security indicators, and our methods for interrogating security elements are pretty restricted. If you don't believe me, just take your favorite browser on over to BadSSL.com (https://badssl.com), a great resource for seeing how your browser handles invalid certificates that are invalid for all sorts of reasons. You'll see that there's not a lot of information presented about why certs are invalid, just that they are invalid in a very binary way. And good luck trying to see the cert details of a given valid certificate; on a desktop browser, it's usually something like "click the lock, view the cert." On mobile, it's *checks notes* impossible. Yeah.

But we're not talking about cert issues, we're talking about the address bar. Because we have very few ways to actually validate the source of data on our phones, the address bar is pretty much the only bit of screen real estate that developers (angelic and devilish alike) are prohibited from monkeying with. Most browser vendors understand this, and build in controls to prevent "what's shown on the screen" and "where that came from" are inexorably linked, making it difficult to convincingly

Topics

Metasploit (800) (/blog/tag/metasploit/)

Vulnerability Management (418) (/blog/tag/vulnerability-management/)

Detection and Response (388) (/blog/tag/detection-and-response/)

Research (277) (/blog/tag/research/)

Application Security (156) (/blog/tag/application-security/)

Cloud Security (110) (/blog/tag/cloud-security/)

Popular Tags

Search Tags

Metasploit (/blog/tag/metasploit/)

Logentries (/blog/tag/logentries/)

IT Ops (/blog/tag/it-ops/)

Vulnerability Management (/blog/tag/vulnerability-management/)

Detection and Response (/blog/tag/detection-and-response/)

Metasploit Weekly Wrapup (/blog/tag/metasploit-weekly-wrapup/)

Research (/blog/tag/research/)

Automation and Orchestration (/blog/tag/automation-and-orchestration/)

Nexpose (/blog/tag/nexpose/)

Incident Detection (/blog/tag/incident-detection/)

InsightIDR (/blog/tag/insightidr/)

Exploits (/blog/tag/exploits/)

Incident Response (/blog/tag/incident-response/)

spoof the location of some text or images. (Note, there's been reporting that Google is thinking about monkeying with it again (https://arstechnica.com/gadgets/2020/06/google-is-messing-with-the-address-bar-again-new-experiment-hides-url-path/), but that's another fight for another day.)

Affected browsers

So, with all that for context, here is the surprisingly diverse set of mobile browsers, shown in the table below (note that Opera and Apple are CVE Numbering Authorities in their own right, and will be populating their own CVE identifiers for those issues).

CVE	Vendor	Browser	Version	Platform	Fixed?
CVE-2020-7363	UCWeb	UC Browser	13.0.8	Android	Fixed v13.3.2 on Oct 21, 2020
CVE-2020-7364	UCWeb	UC Browser	13.0.8	Android	Fixed v13.3.2 on Oct 21, 2020
CVE-2020-6158	Opera	Opera Mini	52.2	Android	Fixed in version 52.3 Nov. 23, 2020
CVE-2020-6157	Opera	Opera Touch	2.4.4	iOS	Fixed in version 2.4.5 released Sep 15, 2020
CVE-2020-6157	Opera	Opera Touch	2.4.4	iOS	Fixed in version 2.4.5 released Sep 15, 2020
CVE-2020-6157	Opera	Opera Touch	2.4.4	iOS	Fixed in version 2.4.5 released Sep 15, 2020
CVE-2020-7369	Yandex	Yandex Browser	20.8	Android	Automated reply, followed up Oct. 19, 2020. Fix published Oct 1 in version 20.8.4.
CVE-2020-7370	Danyil Vasilenko	Bolt Browser	1.4	iOS	Support email bounced, alerted Apple product security
CVE-2020-7371	Raise IT Solutions	RITS Browser	3.3.9	Android	Fix expected Oct. 19, 2020
CVE-2020-9987	Apple	Safari	iOS 13.6	iOS	Fix released Sept. 16, 2020

Now, some browsers are more popular than others, but even some of these relatively obscure browsers have some pretty impressive download stats—the least popular, Bolt, has over 210,000 reviews and ranks No. 47 in the App Store, and UC Browser is probably the most popular non-FOCES browser around, with over 500 million downloads from Google Play. (By the way, FOCES is an initialism I just made up to refer to the "common" browsers Chrome, Firefox, Safari, Opera, and Edge/Explorer) (and yes Opera is still common) (right?). Yandex is pretty popular, too, at over 100 million installs, and RITS is sitting at over a million. So, all together, nothing to sneeze at, installation-wise.

Exploitation details

Exploitation all comes down to, "Javascript shenanigans." By messing with the timing between page loads and when the browser gets a chance to refresh the address bar, an attacker can cause either a pop-up to appear to come from an arbitrary website or can render content in the browser window that falsely appears to come from an arbitrary website.

In all cases, the victim would have to visit a website that the attacker can post executable javascript to. Normally, this wouldn't include websites like Facebook, Reddit, Twitter, or other online forums (they do a pretty good job in protecting

Komand (/blog/tag/komand/)

Penetration Testing (/blog/tag/penetration-testing/)

Related Posts

CVE-2022-27518: Critical Fix Released for Exploited Citrix ADC, Gateway Vulnerability — READ MORE (/BLOG/POST/2022/12/13/CVE-2022-27518-CRITICAL-FIX-RELEASED-FOR-EXPLOITED-CITRIX-ADC-GATEWAY-VULNERABILITY/)

Patch Tuesday - December 2022 — READ MORE (/BLOG/POST/2022/12/13/PATCH-TUESDAY-DECEMBER-2022/)

2023 Cybersecurity Industry Predictions — READ MORE (/BLOG/POST/2022/12/08/WEBINAR-2023-CYBERSECURITY-INDUSTRY-PREDICTIONS/)

CVE-2022-4261: Rapid7 Nexpose Update Validation Issue (FIXED) — READ MORE (/BLOG/POST/2022/12/07/CVE-2022-4261-RAPID7-NEXPOSE-UPDATE-VALIDATION-ISSUE-FIXED/)

against aforementioned Javascript shenanigans), but would include a website that was set up by the attacker and sent to the victim through a phishing email, a phishing text message, or a post to a popular forum. So, for example, imagine a text message from a spoofed phone number that says, "There is an important message from your payment processor, click here" and then you click without really looking, and end up on a web page that clearly (but falsely) says it's Paypal, and hey, can you give up your password real quick?

This seems like a pretty effective attack, given that the address bar is really the only signal you have to tell "where" your browser "is." As it turns out, there are quite a few ways to get Javascript to monkey with timing, and Rafay goes over all of that in pretty deep technical depth, with proof-of-concept code galore, over at his paper (http://www.rafaybaloch.com/2020/10/multiple-address-bar-spoofing-vulnerabilities.html), so if you're the sort who's into reading clever Javascript, definitely check that out. Here's one example out of many:

```
<script>
  function spoof() {
  document.write("<h1>This is not Bing</h1>");
  document.location = "https://bing.com:1234";
    setInterval(function(){document.location="https://bing.com:1234";},9800);
  };
</script>

<p class="test"><input class="btn btn-success btn-lg" type="button" value="Ru
```

The above Javascript renders in a browser as a hyperlink on the "test" text, and when clicked, shows an in-browser rendering of the "This is not Bing" text in a window attributed to bing.com, as shown below.

Opera Mini spoofed address bar

By the way, if this all sounds familiar, an early reference for this class of vulnerability is from back in 2016, when Mozilla disclosed MFSA-2016-82 (https://www.mozilla.org/en-US/security/advisories/mfsa2016-82/), reported by—you guessed it!—a young Rafay Baloch. It will sound familiar again in just a few more weeks, since this research netted out some similar bugs in some desktop browsers as well. Those will be disclosed in a later writeup, though it should be mentioned that MacOS Safari was also affected by the same issue (and fixed in the Big Sur MacOS release from a couple days ago).

Mitigation

First off, pay attention to the browsers listed in the table above and who's actually issued a fix and who remained silent. If your favorite browser is one of the former, congratulations, you probably don't have to do anything—most people let their browser auto-update, so you should be good as of this writing if you're one of those people.

For users of the rest of the browsers listed, you might want to leave a review on the appropriate app store asking when a patch will be available, and reference either this blog post or Rafay's paper (http://www.rafaybaloch.com/2020/10/multiple-address-bar-spoofing-vulnerabilities.html). In the event you're stuck without a patch and you feel you must continue to use that browser, I would suggest that you think long and hard about clicking through on links sent to you by text or email from unknown or shady sources.

This "thinking hard" is about the only protection you really have, since on most texting and email interfaces, long press on a link might get you a preview, or it might copy it to the clipboard, or it might do something else. There's very little consistency there, and nothing on mobile is quite equivalent to the desktop practice of "mouse

over the link and see the target URL." The good news is that these aren't code execution bugs or anything like that, but any malicious uses of these vulnerabilities will be designed to trick you into doing something, so be careful out there.

For developers of mobile applications, and especially general-purpose browsers: You always need to be extra, extra careful about the parts of the interface that you are using to assert security-sensitive things to users, and you should do your best to train users to trust those areas over other, user-controlled elements. Android and iOS do a pretty good job of making sure applications can't go completely wild on the user with sandboxing and controls around other operating system level UI, so the in-application UI is where all the malicious action is at. Also, if you're looking for a code review of your mobile app, I know a guy (https://www.rapid7.com/services/security-consulting/iot-security-services/).

Disclosure details

Rafay Baloch first discovered and documented these vulnerabilities over the summer of 2020, and we coordinated with Rafay to send out a flurry of vulnerability notifications on Monday, Aug. 10, 2020. As indicated in the table of Affected Browsers up top, we experienced a range of responses to these disclosures. Both Opera and Apple were Johnny-on-the-spot when it came to handling and addressing these vulnerabilities, assigning tickets and incident IDs to these issues (Ticket number OT-2310 from the Opera team, incident number 742977112 from Apple). This isn't surprising, since both Opera and Apple are well-practiced in coordinated vulnerability disclosure (CVD), and are, in fact, CVE numbering authorities (https://cve.mitre.org/cve/cna.html) themselves.

One vendor didn't reply at all, two replied just before publication of this disclosure, and one disclosure notification bounced completely when trying to disclose to the support email listed on the app's iTunes page (for that one, we alerted Apple product security that it might be abandonware). As of this writing, we don't know whether an update is planned for those other vendors' products. If there is a fix, we'll update this blog post accordingly. (*Update October 22: UC Browser, Opera Touch, Yandex, and Safari have all shipped fixes as of today. This parenthetical will change in the future with any other response updates. Opera Mini shipped a fixed version on November 23, 2020.)*

I'm reminded of the quip from my friend Rob Graham (https://twitter.com/ErrataRob) at ErrataSec: Disclosing vulnerabilities to vendors who haven't set up a way to handle incoming vulnerabilities can feel a lot like writing "WASH ME" on a dirty car on the street (paraphrasing here). It might technically tick a "helpful advice" box, but really nobody is going to act on it, and usually, it reads kind of annoying and churlish. That said, it doesn't mean we shouldn't at least try to take a run at CVD, even if it feels kind of pointless in the moment.

In this case, I'm at least happy that nobody got all bent out of shape and sent a lawyer after me or Rafay, and in other cases, we've seen that first contact is exactly the sort of thing that can nudge a vendor over the line between "no vuln handling" to "some vuln handling," so they're in a better position for next time. This is all to say, we hope this disclosure is exactly that nudge that these other mobile browser providers need to get with the CVD program. And hey, browser vendors, if you need any help with setting that up, I know another gal (https://www.lutasecurity.com/) who's pretty good at that sort of thing. If you're providing software to tens of thousands to millions of people, having this CVD capability is pretty crucial for their safety and your continued success, so let's get on that!

As always, if you have thoughts on this, feel free to leave a comment below, shoot me a tweet (https://twitter.com/todb), or email us at research@rapid7.com (mailto:research@rapid7.com). If you want help on disclosing a vulnerability, you're

welcome to to email me at cve@rapid7.com (mailto:cve@rapid7.com), and we can help you with that too. I've known Rafay for years now, all thanks to CVD, and this sort of thing is one of the most fun and rewarding parts of my job, so don't be shy!

NEVER MISS A BLOG

Get the latest stories, expertise, and news about security today.

SUBSCRIBE

POST TAGS

Vulnerability Disclosure (/blog/tag/vulnerability-disclosure/)

Android (/blog/tag/android/)

Apple (/blog/tag/apple/)

Vulnerability Management (/blog/tag/vulnerability-management/)

SHARING IS CARING

AUTHOR

Tod Beardsley (/blog/author/tod-beardsley/)

Director of Research at Rapid7, contributing author of several Rapid7 research papers, CVE Board member, and Metasploit collaborator. https://keybase.io/todb

VIEW TOD'S POSTS

Related Posts

EMERGENT THREAT RESPON…

CVE-2022-27518: Critical Fix Released for Exploited Citrix ADC, Gateway Vulnerability

VULNERABILITY MANAGEM…

Patch Tuesday - December 2022

XDR

2023 Cybersecurity Industry Predictions

VULNERABILITY DISCLOSURE

CVE-2022-4261: Rapid7 Nexpose Update Validation Issue (FIXED)

VIEW ALL POSTS

Search all the things

BACK TO TOP

(/)

CUSTOMER SUPPORT

+1-866-390-8113 (Toll Free) (tel:1-866-390-8113)

SALES SUPPORT

+1-866-772-7437 (Toll Free) (tel:866-772-7437)

Need to report an Escalation or a Breach?

CLICK HERE (/services/incident-response-customer-escalation/)

SOLUTIONS

All Solutions (https://www.rapid7.com/solutions)

Industry Solutions (https://www.rapid7.com/solutions/industry)

Compliance Solutions (https://www.rapid7.com/solutions/compliance/)

SUPPORT & RESOURCES

Product Support (https://www.rapid7.com/for-customers)

Resource Library (https://www.rapid7.com/resources)

Our Customers (/customers/)

Contact Us

Events & Webcasts (https://www.rapid7.com/about/events-webcasts)

Training & Certification (https://www.rapid7.com/services/training-certification)

IT & Security Fundamentals (https://www.rapid7.com/fundamentals)

Vulnerability & Exploit Database (https://www.rapid7.com/db)

ABOUT US

Company (https://www.rapid7.com/about/company)

Diversity, Equity, and Inclusion (https://www.rapid7.com/about/diversity-equity-and-inclusion/)

Leadership (https://www.rapid7.com/about/leadership)

News & Press Releases (https://www.rapid7.com/about/news)

Public Policy (https://www.rapid7.com/about/public-policy)

Open Source (https://www.rapid7.com/open-source/)

Investors (https://investors.rapid7.com/)

CONNECT WITH US

Contact (https://www.rapid7.com/contact)

Blog (https://blog.rapid7.com/)

Support Login (https://support.rapid7.com/)

Careers (https://www.rapid7.com/careers)

(https://www.rapid7.com/rapid7/)

(https://www.rapid7.com/about/rapid7-cybersecurity-partner-boston-bruins/)

 Legal Terms (/legal/) | Privacy Policy (/privacy-policy/) | Export Notice (/export-notice/) | Trust (/trust/)