Address Bar Spoofing in Modern Web Browsers: Taxonomy, Exploitation, and Mitigation

Renua Hiwa Ismael

Jaza Mahmood Abdullah

Table of Contents

[Abstract 3](#_Toc190265037)

[Contribution 3](#_Toc190265038)

[Conclusion 3](#_Toc190265039)

[1. Introduction 4](#_Toc190265040)

[1.1 Problem Statement 4](#_Toc190265041)

[1.2 Research Objectives 4](#_Toc190265042)

[1.3 Motivation 4](#_Toc190265043)

[1.4 Thesis Structure 5](#_Toc190265044)

[2. Background and Literature Review 5](#_Toc190265045)

[2.1 Browser Architecture 5](#_Toc190265046)

[2.2 Address Bar Spoofing 6](#_Toc190265047)

[2.3 Prior Work 8](#_Toc190265048)

[2.3.1 Yorick Koster Address Bar Spoof in IE 6 Using Dialog Box 8](#_Toc190265049)

[2.3.2 Luka Treiber in Chrome 14 HTTPS Address Spoof with Delayed Response 8](#_Toc190265050)

[Bibliography 10](#_Toc190265051)

# Abstract

Address bar spoofing is a technique which makes a malicious URL appear like a legitimate one on the address bar. [1] A user might visit an innocent looking website like example.com and using a vulnerability inside the browser will manipulate the address bar to show as bank.com, which will trick the user into inserting his credentials or sharing any sensitive information with the site which eventually will be sent to the attacker. In this research we will conduct a full investigation on the current state of browsers address bar security by looking into previous findings and applying them to different browsers on different operation systems. In the process we will make browsers more secure and help billions of users not getting scammed by attackers leveraging these vulnerabilities. At the current state of writing, we have helped secure 10+ web browsers and sent more than 50 reports with many new novel techniques and in the process, we have earned $60,000+ in bug bounty awards from the vendors.

## Contribution

This area of research is very narrow and there wasn’t any previous academic research on the topic, we created a GitHub repo [2] of all the findings and documenting each attack vector along with identifying many new techniques that have never been discussed that can be applied to a wide range of browsers. Also, the process of classing the bugs into categories that can be used to understand more about the vulnerability. We will showcase the proper mitigation for the vulnerabilities for current browsers or designing a new browser that can be used as best practice.

## Conclusion

While modern browsers have put a lot of efforts into mitigating these vulnerabilities but still there is some techniques that are overlooked and missed the tests, by participating in bug bounty programs and responsible disclosure policies we can pin point these weaknesses and help them secure the web browsers. With sharing the research publicly and the GitHub repo [2] many vendors can mitigate the bugs by going through each classification case and test its browser which we have provided POC code for each one.

# 1. Introduction

## 1.1 Problem Statement

Phishing is a social engineering technique that, through the use of various methodologies, aims to influence the target of the attack to reveal personal information, such as an email address, username, password, or financial information. This information is then used by the attacker to the detriment of the victim [3] . Today, phishing is considered one of the most pressing cybersecurity threats for all internet users, regardless of their technical understanding and how cautious they are. These attacks are getting more sophisticated by the day and can cause severe losses to the victims [4] . Most of these attacks are made through URL Masking also known as URL cloaking, is a technique used to hide the true URL of a website behind a different URL. This technique is often used to make it appear like a website is hosted on a different domain or to simplify a long and complicated URL [5] . But what if an attacker can also spoof the address bar and leverages the browser vulnerability to show a trusted origin? The current state of this issue shows that the attack is not easy to

discover, since the URL matches the legitimate address [6]. As reported by FBI [7], billions

of dollars have been lost due to spear phishing attacks in recent years.

## 1.2 Research Objectives

Investigate existing and novel address bar spoofing techniques by looking into Chromium Issue Tracker [8], Bugzilla database [9] and HackerOne publicly disclosed reports [10]. Document each bug and categorize them into multiple predefined classes. Investigate each modern web browser (Chrome, Firefox, Safari, Edge, Opera…) state of security and number of patched vulnerabilities and how they handled each case then looking into bypassing the patches or retesting across different versions. New features interduces new vulnerabilities and a complex system like a web browser is very prone to a bug comes back to live after a successful patch. Then propose a solid framework for mitigating these vulnerabilities for current web browsers or any new browser that are been built from scratch or on top of an engine like Chromium, also making a list of best practices to make a near-perfect secure web browser address bar that is not vulnerable to all previous attacks and those that are discovered during this research.

## 1.3 Motivation

A phishing campaign is a scam created by cybercriminals to steal financial resources or sensitive data from victims using manipulative emails or other fraudulent digital assets. [11] In a report published by Kaspersky in 2022 states that 48.63% of all emails around the world spam emails. [12] Phishing attacks cost large organizations $15 million annually, or more than $1,500 per employee. [13] Phishing attacks continue to be a significant threat in the cybersecurity landscape, targeting individuals, businesses, and even national security. These attacks have evolved in sophistication and variety, exploiting human vulnerabilities and leveraging new technologies. [14] By eliminating URL based phishing attacks and browser address bar spoofing we will minimize the attack surface and reducing these campaign attacks in a significant amount

## 1.4 Thesis Structure

This thesis is structured to provide a clear and systematic analysis of address bar spoofing vulnerabilities. The introduction presents an overview of address bar spoofing, outlining the research objectives, motivation, and scope of the study. The background and literature review explores browser architecture, previous research on spoofing techniques, and identifies gaps in existing studies. The methodology details the research approach, tools, testing environment, and ethical considerations for responsible disclosure. The vulnerability taxonomy and case studies categorize different spoofing techniques and present selected real-world vulnerabilities with proof-of-concept demonstrations. The findings and analysis summarize key discoveries, statistical trends, and vendor responses to disclosed vulnerabilities. The mitigation strategies section proposes technical solutions, developer best practices, and user awareness strategies. The discussion examines the broader implications of address bar spoofing, challenges in browser security, and areas for future research. Finally, the conclusion recaps the key findings and contributions, emphasizing the importance of addressing UI-level security threats. The references and appendices provide citations, detailed PoC code (where permitted), a full list of vulnerabilities, and vendor communication logs.

# 2. Background and Literature Review

## 2.1 Browser Architecture

Navigation is the first step in loading a web page. It occurs whenever a user requests a page by entering a URL into the address bar, clicking a link, submitting a form, as well as other actions. [15] After requesting the URL, the browser will make a DNS query to the given address, Domain Name System (DNS) is a data store that stores the mapping from a domain name (such as google.com) to its IP address (142.250.185.78). You could compare DNS to an address book or a telephone book [16]. Once the IP address is known, the browser sets up a connection to the server via a TCP three-way handshake connection over an IP based network. Similarly, a four-way handshake is used to terminate the connection. [17] Once the browser establishes a connection to the server, it follows the communication rules of the HTTP(s) protocol. The browser initiates this process by sending an HTTP request to retrieve the page's content. This request includes a request line, headers (which provide metadata about the request), and a body. The request line contains details that help the server understand what the client—your browser—intends to do. [18] After getting the response the browser will start rendering process and display the contents based on the engine. In figure 1 summarizes the process.

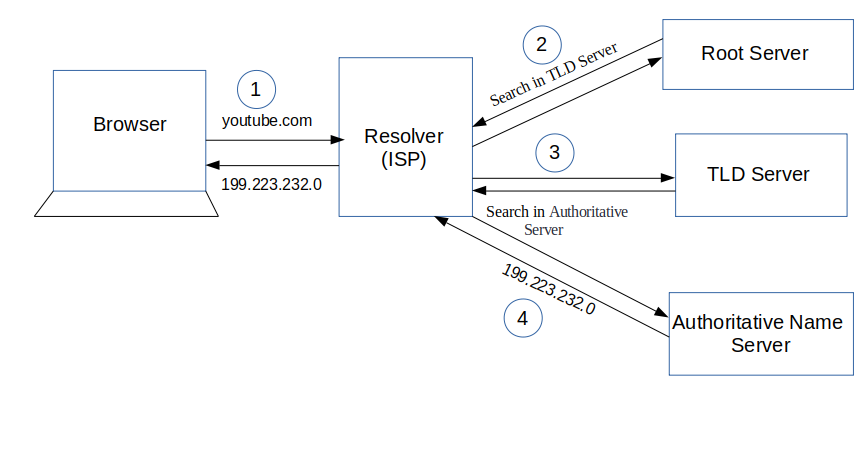


Figure , What happens when you click on a URL in your browser [19]

The value of the address is a complex process and varies across different browsers and operation systems, when a navigation happens from a site to another the value changes only when the response is success and the address is reachable, but if the user types the address inside the URL bar and presses enter the value will be what the user has typed and stays the same in both cases of success response or unreachable address. Browsers use URL scheme which is a complex syntax process as seen in Figure 2.

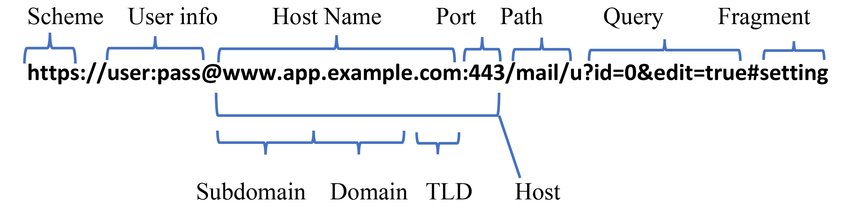


Figure , Typical URL syntax [20]

With this complexity it’s up to the browser vendor to decide what to display inside the URL bar, for example Chrome only shows (Host Name, Port, Path, Query, Fragment), Safari only shows (Host Name) and Opera shows everything except User info. When conducting tests and fuzzing the address bar we have to do separately for each browser and note down the results because these differences.

## 2.2 Address Bar Spoofing

Address bar spoofing is a security vulnerability in web browsers where an attacker manipulates the browser’s address bar to display a fake or misleading URL, making users believe they are visiting a legitimate website when they are actually on a malicious one. This can be achieved through various techniques, such as JavaScript injections, exploiting browser rendering flaws, abusing Unicode characters, or using overlapping elements to mask the real URL. Attackers use this tactic for phishing, fraud, or malware distribution, tricking users into entering sensitive information like login credentials or financial details. [1] [21] When an attacker exploits this vulnerability it will make an identical website to the spoofed website and tricks the users into inserting any info that is interest to the attacker. In Figure 3 a normal user can’t figure out which of the address bars are legit and trust the website as both of the URLs are the same with same spelling and URL syntax structure, this what makes this vulnerability very much effective and scary.

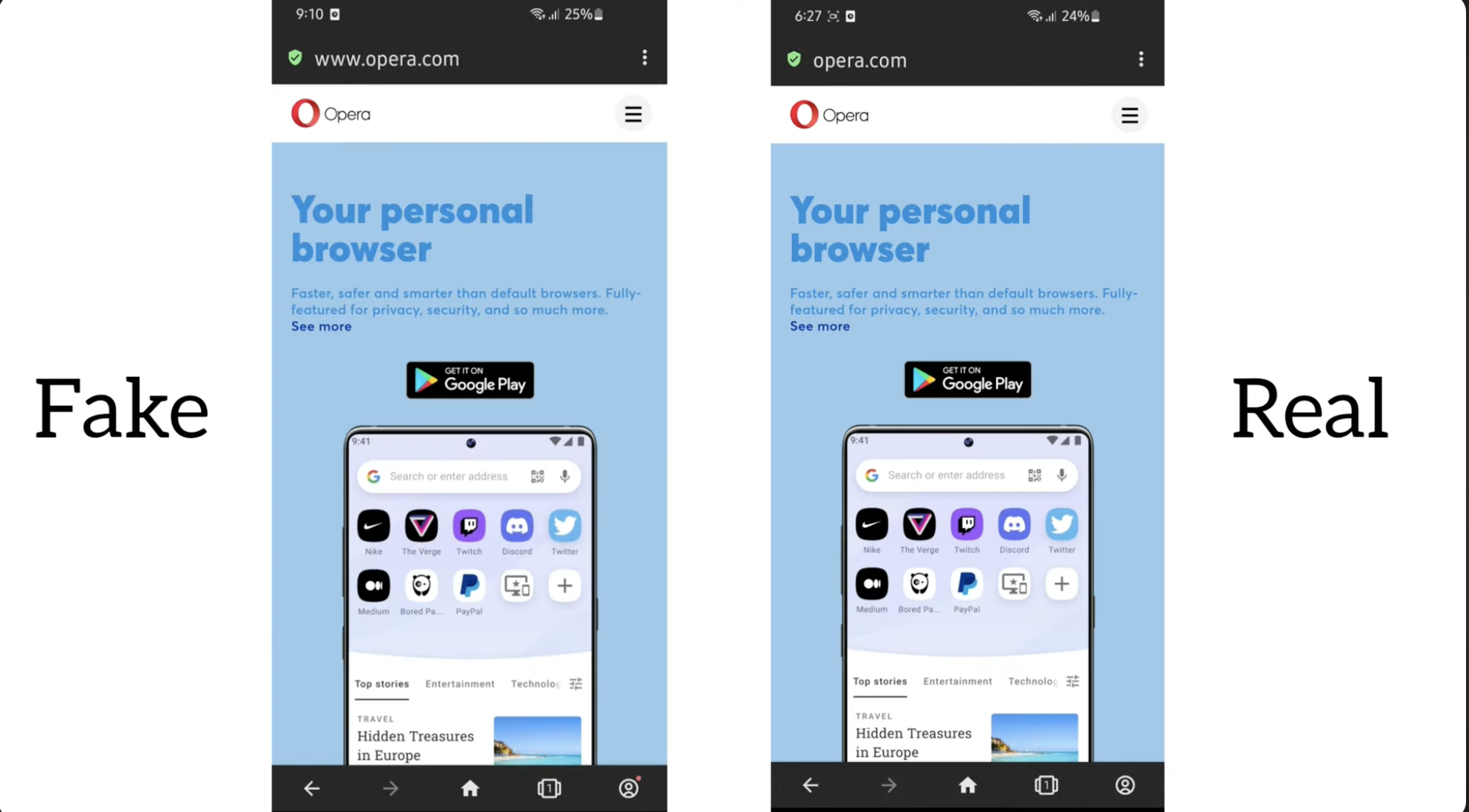


Figure , Fake opera website (left), legitimate opera website (right)

The Chromium Docs Guidelines for URL Display [22] goes through multiple attack scenarios and best practices on how to display a URL inside the browser that uses Chromium engine. One simple example of address bar spoof would be Eliding URLs attack which an attacker uses a long subdomain to hide the main domain part and trick the users into thinking they’re on a legitimate bank site, see Figure 4.



Figure , Eliding URLs attack using long subdomain

In this scenario the attacker is owner of evil.com and has create multiple levels of subdomains which according to RFC 1035 [23] the limit of subdomain levels is 255 octets and each with a maximum length of 63 characters; this means the attacker can create enough subdomains to hide the main evil.com domain. For a bigger display size, it has to use many more subdomains but in a small display like a mobile phone using three subdomains would do the job. Here the browser’s security mechanism should prevent this kind of attack by always highlighting the main domain and not the subdomains of it.

The goal of a successful address bar spoof is displaying another website/URL inside the URL bar that is not under control by the attacker, any ability to modify, hide or freeze the URL without a navigation to the legitimate site is a vulnerability and has to be patched. By address bar spoof we don’t mean misleading domain names and deceptive looking sites, for example an attacker purchases google-login.com and tricks users into sending credentials. [22] It’s an address bar spoof it the user visits evil.com and the browser shows google.com/login inside the URL bar, this way it’s not a misleading domain and an accepted vulnerability.

## 2.3 Prior Work

Address bar spoofing has been an area of concern in web security for years, yet limited academic research has explored its technical depth. This section reviews previous studies, industry reports, and real-world case studies that highlight the evolution of address bar spoofing attacks and the countermeasures implemented by browser vendors.

### 2.3.1 Yorick Koster Address Bar Spoof in IE 6 Using Dialog Box

In 2004, Yorick Koster [24] identified an address bar spoofing vulnerability in Internet Explorer 6 SP1 that allowed attackers to disguise a webpage’s true origin. By exploiting JavaScript dialog boxes and character encoding quirks, an attacker could load a malicious page while keeping the original website’s URL in the address bar.

**var** w = window.open('javascript:alert(\'Hello World!\');', '\_blank');

**setTimeout**('w.location.href="http://www.google.com"', 100);

Inside a vulnerable version of Internet Explorer using the code snippet will open a new window showing an alert box then after 100 milliseconds the window will redirect to Google and the browser will start rendering the website but the address bar is will not change because of the alert box, now the address bar value is attacker’s site and the contents are rendered from google.com. This attack itself is not very useful as for a phishing attack we want the address bar to be google.com and contents from the attacker so you need to find a vulnerable site which has an alert box and chain it with this vulnerability.

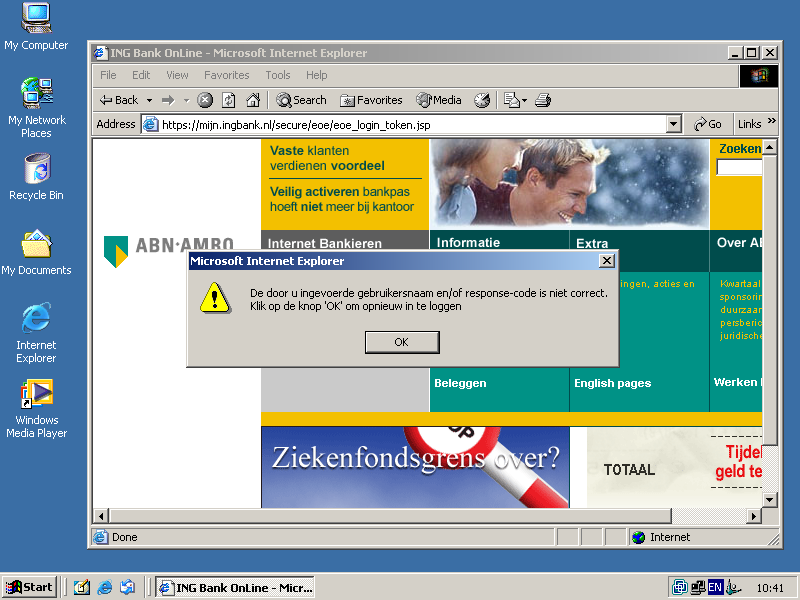


Figure , address bar spoofing of ING Bank [24]

### 2.3.2 HTTPS Address Spoof with Delayed Response by Luka Treiber in Chrome 14

Back in 2010 Chromium launched its bug bounty program [25] we see multiple bugs reported to Chrome and one of them is Luka Treiber in 2012 reported a high severity address bar spoof report in Chrome [26]. Inside the report Luka demonstrates 3 ways an attacker can spoof address bar on the browser, the specific Chrome version has a vulnerability in handling view-source: URLs that when a website redirects to another origin the address bar will get updated before start of rendering and displaying contents. In Figure 6 the diagram shows how this vulnerability worked and what caused the spoof.

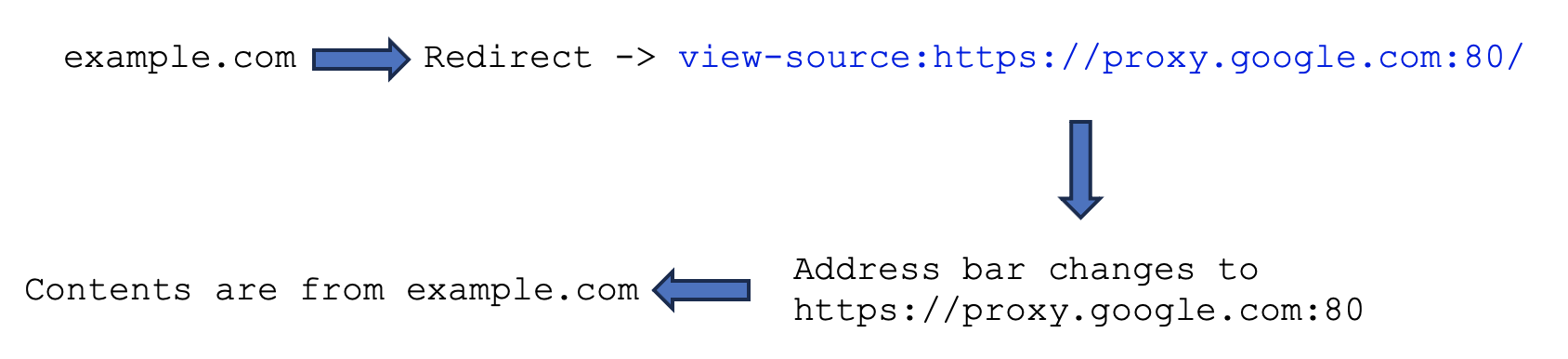


Figure , Address bar spoof diagram using Luka's bug

For the POC Luka created a fake Google login page when the form is filled it will be sent to the attacker’s site and it can be used to account takeover as see in in Figure 7.

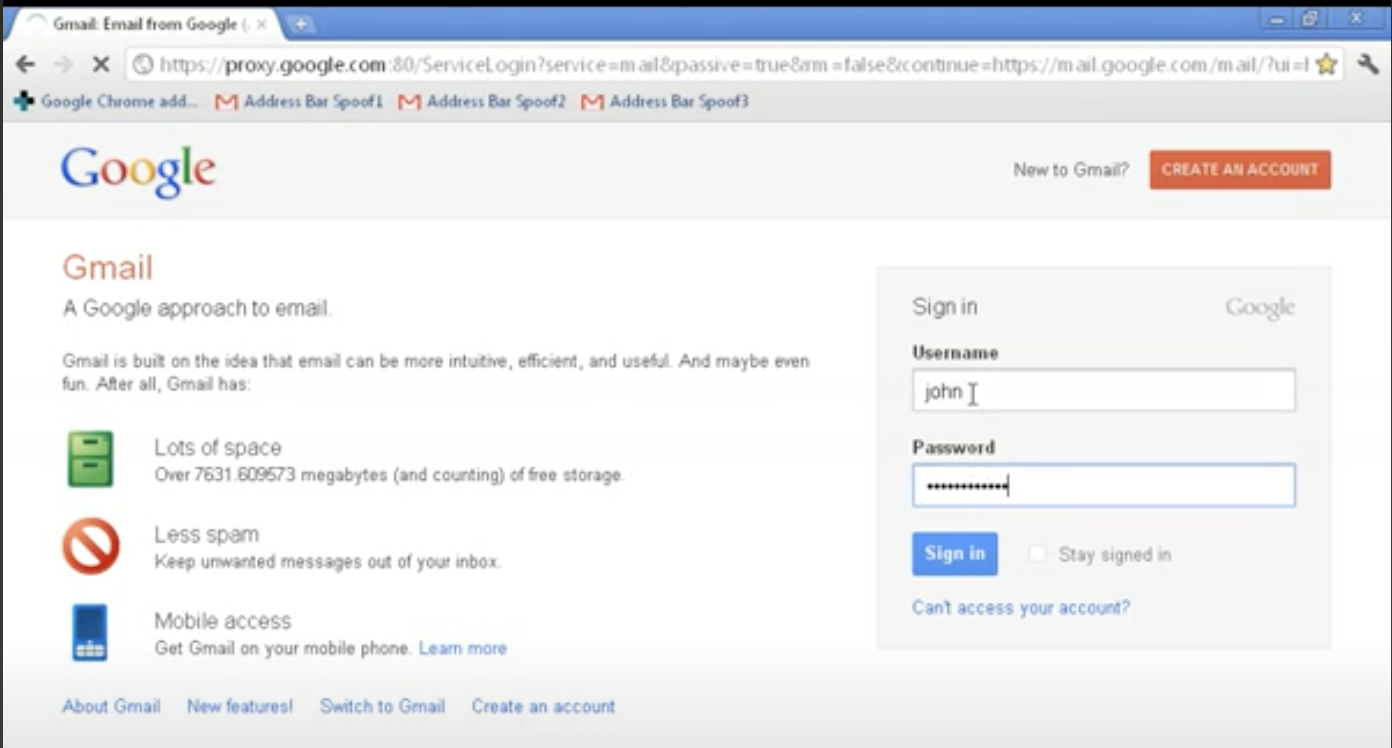


Figure , Address Bar Spoof in Chrome 14 and a Fake Login Page

### 2.3.3 Visual Spoof and Homograph Attack by Chris Weber

Back in 2010 ICANN approved full IDN top-level domains (TLDs), allowing non-Latin characters in domain extensions [27]. With this approval it introduced a new range of spoofing attacks notably Chris Weber inside a Unicode Security Guide repo [28] showcases multiple attack scenarios inside web browser address bars and also in operation system level.

Several attack vectors have been identified, including non-Unicode lookalikes, where character combinations such as “rn” resemble “m,” and Unicode confusable, where visually similar characters from different scripts (e.g., “A,” “Α,” and “А”) are used deceptively. Inside Figure 8 it represents what visually appear as two identical domain names, however, the second contains the U+0261 LATIN SMALL LETTER SCRIPT G which is different domain from google.com.

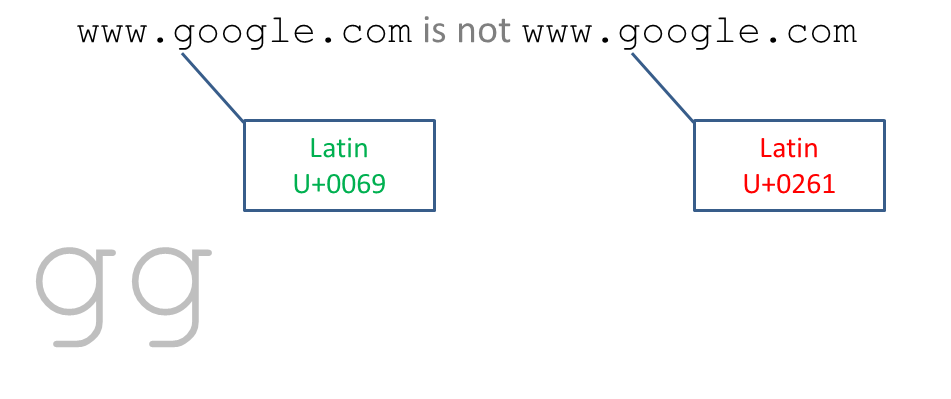


Figure , Two identical google domains but using a deceptive g character [28]

Inside the report Chris mentions many more attack scenarios but we only focus on browser address part also he mentions that currently there isn’t any mitigation for these kinds of attacks and it’s up to the user to differentiate, in next chapters we go through current mitigations and browsers support.

With the wide range of Unicode characters even if you can’t get identical character to spoof you can still use look alike character as seen in Figure 9 which a spoofed domain uses homograph (visual spoofing) attack to show as mozilla.org but the “i” character is different.

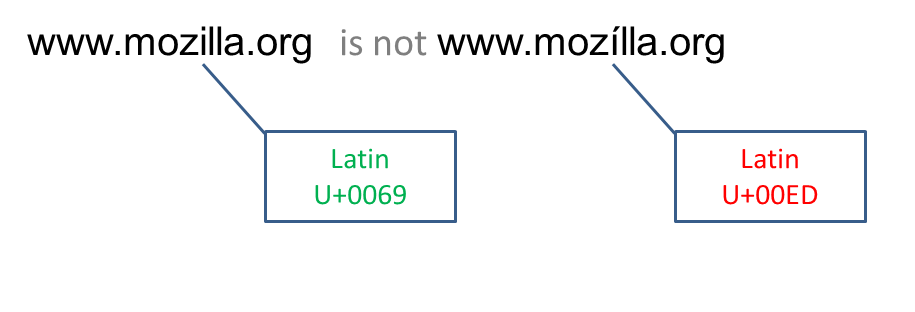


Figure , Mozila.org spoof using Latin \u00ED character [28]

# Bibliography

|  |  |
| --- | --- |
| [1] | TrendMicro. [Online]. Available: https://www.trendmicro.com/vinfo/us/security/definition/address-bar-spoofing. |
| [2] | Renwa, "Address Bar Spoofing Repo," [Online]. Available: https://github.com/RenwaX23/Address\_Bar\_Spoofing. |
| [3] | P. &. S. M. Stavroulakis, "Handbook of information and communication security," *Springer Science & Business Media.,* 2010. |
| [4] | Z. H. C. N. L. &. K. I. Alkhalil, "Phishing attacks: A recent comprehensive study and a new anatomy.," *Frontiers in Computer Science,* vol. 3, no. 563060, 2021. |
| [5] | A. Dubey, "Website Spoofing (With Examples)," [Online]. Available: https://bolster.ai/glossary/website-spoofing. |
| [6] | F. A. Almubarak, "Browser-in-the-Browser (BitB) Attack: Case Study.". |
| [7] | F. B. o. I. (FBI), "Business E-mail Compromise The 12 Billion Dollar Scam.," 2018. [Online]. Available: https://www.ic3.gov/Media/Y2018/PSA180712. |
| [8] | "Chromium Issue Tracker," [Online]. Available: https://issues.chromium.org/issues. |
| [9] | "Bugzilla & Firefox Issue Tracker," [Online]. Available: https://bugzilla.mozilla.org/home. |
| [10] | "Hackerone Hacktivity," [Online]. Available: https://hackerone.com/hacktivity/overview. |
| [11] | ZeroFox, Phishing Campaign Glossary, [Online]. Available: https://www.zerofox.com/glossary/phishing-campaign/. |
| [12] | Kaspersky, 2023. [Online]. Available: https://securelist.com/spam-phishing-scam-report-2022/108692/. |
| [13] | StationX, 2024. [Online]. Available: https://www.stationx.net/phishing-statistics/. |
| [14] | M. Z. S. W. A. M. N. A. A. R. S. &. A. W. Nadeem, "Phishing attack, its detections and prevention techniques.," *Int. J. Wirel. Secur. Netw,* vol. 1, no. 13-25, 2023. |
| [15] | MDN, "Populating the page: how browsers work," [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/Performance/How\_browsers\_work. |
| [16] | N. Kim, "What Happens When You Type a URL Into Your Browser?," 2022. [Online]. Available: https://systemdesign.one/what-happens-when-you-type-url-into-your-browser/#fn:1. |
| [17] | MDN, "TCP handshake," [Online]. Available: https://developer.mozilla.org/en-US/docs/Glossary/TCP\_handshake. |
| [18] | K. Lee, "What happens when you type a URL into your browser?," Amazon AWS, 21 August 2021. [Online]. Available: https://aws.amazon.com/blogs/mobile/what-happens-when-you-type-a-url-into-your-browser. |
| [19] | FreeCodeCamp, "What happens when you click on a URL in your browser," 2019. [Online]. Available: https://www.freecodecamp.org/news/what-happens-when-you-hit-url-in-your-browser/. |
| [20] | I. Kareem and A. Obaid, "Fraud usage detection in Internet users based on log data.," *The International Journal of Nonlinear Analysis and Applications (IJNAA),* Vols. 12. 2008-6822, 2021. |
| [21] | MalwareBytes, "Address Bar Spoofing Glossary," [Online]. Available: https://www.malwarebytes.com/glossary/address-bar-spoofing. |
| [22] | Chromium, "Guidelines for URL Display," [Online]. Available: https://chromium.googlesource.com/chromium/src/+/HEAD/docs/security/url\_display\_guidelines/url\_display\_guidelines.md. |
| [23] | N. W. Group, "RFC 1035," November 1987. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc1035. |