








Browser-in-the-Browser (BitB) Attack: Case Study

Khalid Alissa^{1,*} , Bushra Alhetelah¹ , Ghadeer Alazman¹ , Asma Bader² , Noor Alhomeed² , Layan Almubarak² , Fajer Almulla² 

¹SAUDI ARAMCO Cybersecurity Chair, Networks and Communications Department, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia

²Department of Networks and Communication, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, P.O. Box 1982, Dammam 31441, Saudi Arabia

*Corresponding author: Khalid Alissa, Address, Email: kaalissa@iau.edu.sa

ABSTRACT: Phishing attacks are becoming more sophisticated daily, taking advantage of victims' lack of awareness to steal sensitive information. The browser-in-the-browser (BitB) attack is a novel and sophisticated phishing technique that uses a single sign-on (SSO) popup window that mimics a legitimate browser login to steal a user's credentials. In addition, an attacker can customize the URL shown in the header of the fake login popup to appear as a legitimate link with a padlock symbol. This attack is relatively dangerous as it steals sensitive information and is designed in a way that is hard to detect using HTML, CSS, JavaScript, and social engineering techniques. This paper aims to study and analyze BitB. Also, conduct an experiment on the BitB attack scenario from the attacker and victim's points of view and recommend countermeasures to detect the attack. The results of BitB attack analysis and experiments show that BitB attacks require basic knowledge of phishing tools and programming languages to be implemented by attackers and achieve their goal of stealing sensitive information that allows them to move on to the next stage of their attacks. Further, this paper will be the first academic paper to study a new type of attack due to the lack of available research and documentation, making it a crucial contribution to the field.

KEYWORDS: Phishing Attacks, Browser Attacks, Browser-in-the-Browser Attack, SSO

1. Introduction

Day after day, attackers develop and innovate techniques to deceive users maliciously and cleverly. The Browser-in-the-Browser (BitB) attack is a web attack that simulates a login page with a spoofed legitimate domain to steal user credentials, unlike the traditional phishing websites, which mimic the original web page and have deceptive URLs [1]. This attack mainly targets the Single Sign-On authentication model to obtain sensitive information, specifically the credentials of users [2]. This attack severely threatens web users because users trust Single Sign-On authentication. After all, it saves time and is available on most websites, oblivious to the risks they may face if they do not take countermeasures. Moreover, the BitB attack offers the ability to spoof a legitimate URL by using HTML, CSS, and JavaScript to build a fake Single

Sign-On window displayed to the users, proving that it is easy to fabricate identical popups [3]. The BitB attack is a recent phishing attack in the browser, such as man-in-the-browser and browser-in-the-middle. However, the picture-in-picture attack, in which the attacker embeds a fake website within a legitimate website, and the BitB attack share some things in common [4]. For example, picture-in-picture has an actual outer window and a fake inner window, but the main difference (BitB) uses only the SSO window as a fake one. The Picture-in-Picture fake window is the whole website [5].

Attackers used a BitB attack to target the government and companies' websites, sending them a phishing link with a similar user interface and an inner browser containing a legitimate URL to request the user's credentials, which would later be used to access the user's

account [6]. In the Indian government, systems were blocked and to unlock them needs to pay INR 30000 [6]. Furthermore, in Ukraine, thousands of modems were disconnected from the network affecting the operations of 5,800 wind turbines belonging to the German company. There are temporary techniques done manually to detect BitB attempts on the webpages, such as the SSO popup windows, which cannot be dragged outside the outer window or maximized [7]. Also, the padlock icon in the browser header is a fake picture to mimic valid SSL certificates, and the popup theme differs from the browser or operating system theme [8]. The mentioned indicators require full awareness from the users and professionals in order to detect this type of attack, so it requires real-time solutions to increase awareness and reduce the attack risk before the user becomes a victim.

This paper presents a detailed analysis of the Browser-in-the-Browser attack, addressing a significant gap in knowledge in the field of cybersecurity. By providing the first published paper on this topic, it shows its risk and method of operation, in addition to the experimental results of the attack and ways to address it. The rest of the paper is organized as follows: Section 2 provides an overview of phishing and similar attacks to the BitB attack. Section 3 provides detailed background on the BitB attack, attack implementation requirements, and experimental results of a possible attack scenario. Section 4 concludes the paper.

2. Overview of Phishing

Phishing is the practice of stealing online users' financial and personal information by spoofing legitimate organizations [9]. According to the Anti Phishing Working Group (APWG) 1st Quarter, 2022 report, they recorded around one million phishing attacks. This was the worst phishing quarter that APWG has observed [10]. The phishing attack goes through a life cycle from the planning phase until the launch of the attack or the fraud. Planning is the first phase of the attack where the attacker plans to get the maximum earnings with minimum threats and identify the target. Then, the collection phase is where the attacker gathers information about the target and the methods that will be used in the attack. Finally, the attacker conducts fraud and steals the user's sensitive information [11].

Phishing attacks may result in the theft of data, mainly personal information including login information

and passwords for various online accounts. Mostly, phishers design fake web pages to look like legitimate web pages and have different but deceptive URLs [1]. However, today's attack works differently. The used URL looks correct and safe to the victim. The pop-up can display correct addresses when users hover the mouse over the webpage content links as our main research topic Browser-in-the-Browser attack [7]. So, looking into the URL is not enough as professionals were saying it is.

One of the solutions that help to detect phishing attacks is the browser plugins which are client-side detections that use different detection techniques such as blacklisting, and pattern matching [11]. For example, DontFishMe is a browser plugin that is used to detect online banking phishing websites to alert users before doing any financial transaction. Also, web shield-phishing protection is a plugin that alerts the users if the website is phishing or suspicious by red and green colours in sequence [11].

2.1. Browser-based Phishing Attacks

Web browser-based attacks use browsers, IT parts of web services and content management systems to gather login credentials, steal users' payment details, or infect systems with malware. It is an example of fileless attacks which are dangerous to organizations and difficult to detect. Most of them use browser third-party plug-ins like JavaScript Flash, and ActiveX since behavioural monitoring always leaves some exposure window and there are no links or files for security systems to identify [12]. Browser attacks are very popular and are likely to be successful on systems that have not been adequately hardened against them [13]. It has become an almost daily activity due to the rapid growth of the Internet and the development of the web to become a universal interface for creating many applications [14]. Some of the most popular browsers, like Mozilla Firefox, Microsoft's Edge, and Chrome, now come with at least a basic level of defence against these attacks.

2.1.1. Man-in-the-Middle Attack (MitM)

Man-in-the-Middle (MitM) attack is a web browser-based attack in the field of computer security. This could begin with phishing tactics and in some cases coupled with browser attacks [15]. (MitM) attacks compromise the actual data that flows between endpoints, and the confidentiality, availability, and

integrity of the data itself [16]. The diversity of existing MitM attacks gives witness to the popularity of this attack category. Man-in-the-Browser (MitB) and Browser-in-the-Middle (BitM) are examples of the most common (MitM) attacks aimed at web services.

2.1.2. Man-in-the-Browser Attack (MitB)

The Man-in-the-Browser (MitB) attack is a browser-based attack that uses trojan horses as extensions to target web browsers. The trojan horse does not begin working until the victim connects to the institution's one-time pad (OTP) or public key infrastructure (PKI). Once the victim enters their credentials, the attacker will alter the exterior of the browser's contracts. Such attacks usually target banking organizations and web financial institutions. The victim will not notice the change since it is happening in a real-time manner. This attack is dangerous because the anti-virus cannot detect it and can skip traditional authentication mechanisms such as OTP or two-step verification [17].

2.1.3. Man-in-the-Middle Attack (BitM)

Browser-in-the-Middle (BitM) is like (MitM) in the way it monitors the data flow between the service it accesses and the client, but it avoids some of MitM's common flaws. It could begin with phishing techniques and be combined with the Man-in-the-Browser (MitB) attack in some cases. One of its features is that there is no need for malware to be installed on the victim's machine, and the emphasis is on giving the attacker complete control [13]. The BitM attack replaces the victim's browser with a malicious transparent browser, acting and looking like the desired web page of the target site (e.g., a social network, a banking application, etc.), and hosted on the attack platform, over which the attacker has complete control, and keeping the victim completely unaware of the substitution. The victim will be able to browse the target web application while using a transparent web browser that has been unknowingly exposed by the malicious web server.

2.2. Picture-in-the-Picture Attack

A Picture-in-Picture attack is one of the phishing techniques that was recorded by APWG (Anti-Phishing Working Group) [4] in which the attacker embeds a fake website within a legitimate website as illustrated in Figure 1. This method is as effective as other phishing attacks such as homograph attacks which is using

similar characters to the original domain [18]. Moreover, the main reason behind the name of the Picture-in-Picture attack is that the attacker uses the fake browser address as an image inside the legitimate browser to mislead and lure the users that they are dealing with real websites to steal their sensitive information [19].

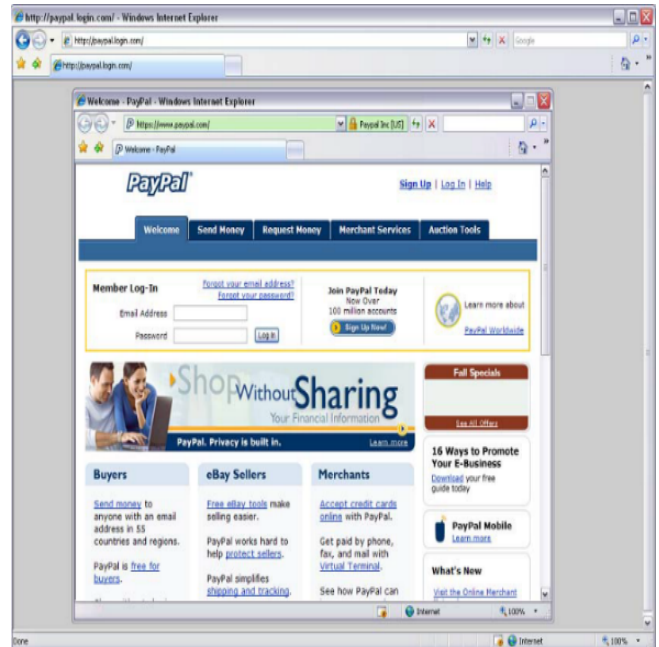


Figure 1: Picture-in-Picture attack. The outer window is real, and the inner window is fake [19].

Furthermore, several indicators help to detect the picture-in-picture attack [5]:

- Maximize the inner window: The fake inner windows cannot be maximized but is not a reliable sign of detection since there are many legitimate websites that use popups with a fixed size.
- Customize browser theme: Some browsers such as Chrome and Firefox provide the ability to change your browser's colours, so if the inner browser is mismatched with the outer browser, then that is a strong indication that there is something suspicious.
- Drag the inner window: The inner window in the picture-in-picture attack cannot be dragged outside the outer windows which could be a good indication of the attack but does not provide additional information about the legitimacy of the window.

One of the detection techniques that will help to mitigate the picture-in-picture attack is PhotoAuth is a two-factor authentication method that prevents real-time phishing attacks by taking picture of the browser address as a second authentication factor of the user authentication to detect the multiple browsers open

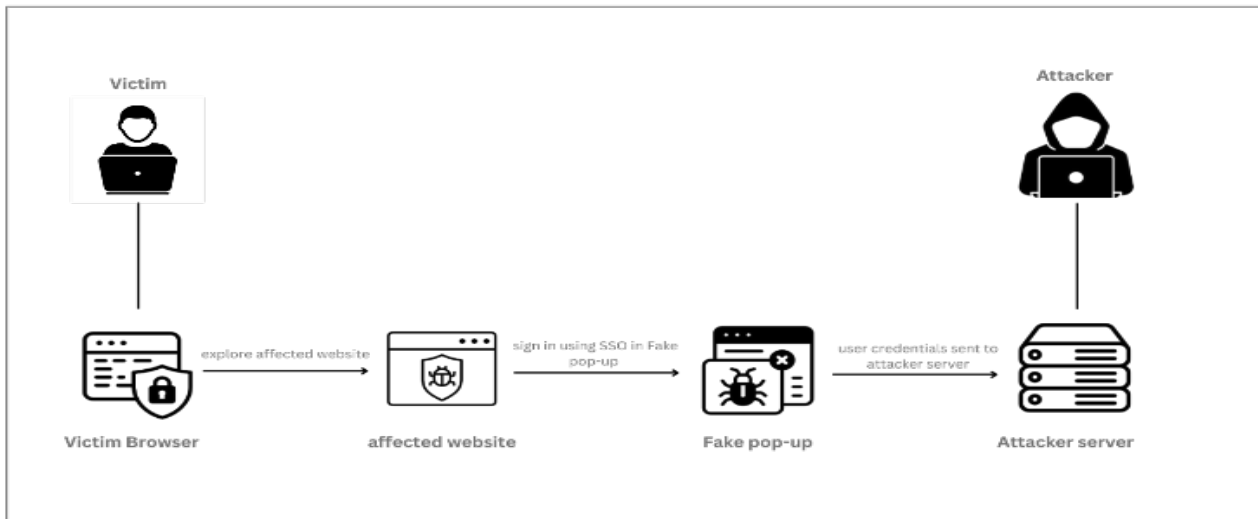


Figure 2: Browser-in-the-Browser (BitB) attack

and determine if the user visiting a real browser or phishing one [18].

2.3. Browser-in-the-Browser Attack (BitB)

A new attack recently appeared known as a Browser-in-the-Browser attack (BitB), which was first reported by a researcher called "Mr.d0x" [8]. This attack takes benefit of the popular third-party single sign-on options rather than the normally time-consuming process of filling out information to create a new account. The BitB attack is used in advance and is a more sophisticated phishing attack that going to trick users, by displaying a fake pop-up window containing a login panel on the visited website, which enables users to log in to several websites using a single account [3].

Among BitB attack features, when users want to sign up on a compromised site, they will be served with a fake bogus pop-up that looks and feels exactly like a legitimate Single Sign-On (SSO) authentication window. The BitB attack simulates a known company that provides SSO services to accomplish the attack, such as a Google, or Apple prompt, or Microsoft with the correct logo, input fields, and address bar, all the interface components they are accustomed to seeing. Also, when users move the mouse over the "Log in" button and the "Forgot password" link, the window can even display the correct addresses. If the user enters his/her credentials into this window, they will be redirected to the cybercriminal's server rather than Google, Apple, or Microsoft [19].

BitB attack usually proceeds in basic steps, and the way how it is embedded on the website differs

from one attacker to another based on many factors. Figure 2 illustrates the possible scenario of a BitB attack.

The following steps show how to perform BitB attacks:

- The BitB attack takes advantage of the (SSO) pop-ups and creating these pop-ups is quite simple by using only HTML, CSS and JavaScript GitHub template provided by "mr.d0x", or designing a new popup from scratch.
- The address bar of the fake popup spoofs the original domain to make the attack more convincing to the users.
- Pointing the iframe to the malicious server hosts the phishing page such as the Gophish toolkit [3]. As an example, `<iframe src=http://www.attacker.com ></iframe>` is to link the fake popup with the phishing website and receives the user's credentials after clicking on the login button [20]. As seen in Figure 3, there are no noticeable differences between the fake and real popups of login with Facebook [3]

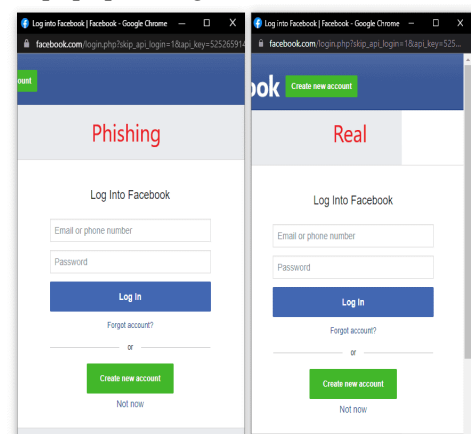


Figure 3: Legitimate SSO login vs fake SSO login [1]

Table 1: Overview of Various Browser-Based Security Attacks

	Description	Frequency	Level of exploitation attack vector	Discovery ability	Affect	Prevention
Man-in-the-Middle Attack (MitM) [21]	In a Man-in-the-Middle attack, someone secretly gets between two talking sides to spy or change the messages.	Broad	Hard	Hard	Normal to Harsh	<ul style="list-style-type: none"> Steer clear of using WiFi undecrypted password networks. Heeding browser alerts that suggest a website is unsafe. Logging off from secure programs while not in use.
Man-in-the-Browser Attack [17]	covertly manipulates online banking transactions by installing malware on the victims device	Broad	Hard	Normal to Hard	Harsh	<ul style="list-style-type: none"> Employ Endpoint Supervision. Secure Browser Extensions. Utilize Secure Banking Utilities.
Browser-in-the-Middle Attack [15]	Using a malicious transparent browser a browser-in-the-middle (BitM) attack places itself between the victims browser and the web server they are accessing.	Broad	Hard	Normal	Harsh	<ul style="list-style-type: none"> Using a secure communication. Implementing network security measure. Using Multi-Factor authentication.
Picture-in-Picture Attack [5]	Involves showing a fake browser window to users that appears to display a legitimate website, so the attacker can the tricks users into thinking they are on a real site when in fact they are on a fraudulent one.	Usual	Hard	Normal	Harsh	<ul style="list-style-type: none"> Ensure that a link that opens an external website opens a new tab, not a new window. Attempt to drag a browser window outside of its parent window. Fake browser windows can't be maximized; therefore, if you find out that you can maximize the window, it might be a fake one.
Browser-in-the-Browser Attack	An advanced type of phishing using 3rd-party single sign-on (SSO) preferences. It works by showing a spoofed pop-up window emulating the real style of third-party SSO login windows each time a user tries to log in to a breached site.	Usual	Normal	Hard	Harsh	<ul style="list-style-type: none"> Check the SSL certificate of the pop-up SSO window Fake browser windows can't be maximized; therefore, if you find out that you can maximize the window, it might be a fake one. Ensure the SSO window does not contain iFrame element in its HTML code.

3. Browser-in-the-Browser Attack

3.1 Real-world BitB attack scenario and detection

Today, browser-in-the-browser attack is a significant threat to many online services. It was first described by the researcher in the Spring of 2022 when he revealed an analysis of the attack and how it works [22]. This attack aimed at government entities, including Ukraine and other such sectors [23]. Since almost all users use (SSO), this attack can affect a large variety of users [3]. One real example that is the latest happening was targeting video gamers, specifically the Steam application for playing, discussing, and creating games. Attackers started targeting victims with direct messages by inviting them to join a team to compete. The shareable link brought the targets to a phishing site for what appears to be an

organization hosting Esports “electronic sports” competitions. Then, the visitors are requested to log in via their Steam account to join a team [24]. The new login page window is a fake window created within the current page using the <iframe> tag in HTML, making it very hard to spot as a phishing attack. The landing pages define the language from the victim's browser preferences and load the correct one, it supports 27 languages. The victim is then prompted to submit the two-factor authentication code on a new form after entering their credentials. To reduce the possibility that the victim would discover the attack, the user is redirected to a legitimate URL address. The victim's login information has already been taken and delivered to the threat actors at this stage. Then attackers modified the victims' email addresses and passwords to

make it more challenging for them to regain control of their accounts [24].

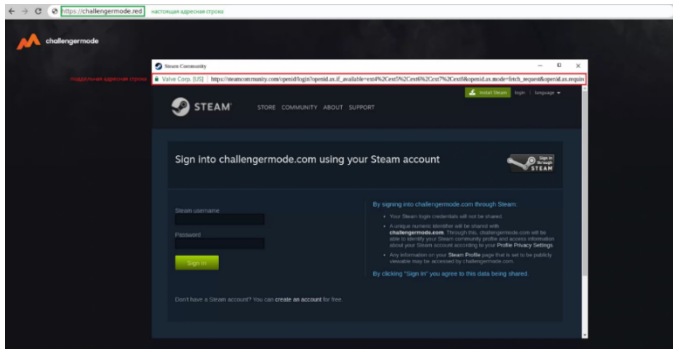


Figure 4: Phishing window created inside the phishing site [24].

There are several temporary techniques to detect BitB attacks on websites which are done manually by users. Below are listings of some indicators that exist some/all of them confirm the BitB attack:

- The (SSO) popup window cannot be dragged out of the outer window [7].
- The lock icon on the popup as seen in Figure 4 is a picture, not a valid SSL certificate.
- Cannot minimize the (SSO) popup window [7].
- Existing the iframe element in the HTML code [8].

Till now no actual prevention techniques developed yet. But there are some known procedures that can reduce the occurrence of such phishing attacks, like verifying a Site's Security and thinking before clicking, etc.

3.1 BitB implementation requirements

A BitB attack's success is determined by how well the SSO popup mimics a legitimate browser login popup, such as the browser header with a padlock symbol, a legitimate URL, operating system, and the use of one of the SSO service providers, such as Google, Facebook, and Steam. The SSO popup is created using HTML, CSS, and JavaScript only, so there is no limitation to the attacker's creativity. The malicious website that is controlled by the attacker can be sent using social engineering with a convincing domain name and offers an SSO option that shows a popup window to steal the user's credentials.

The popup spoofs a legitimate URL for Google login that is placed in the fake browser header with a padlock symbol to lure the user into believing there is a secure connection and uses JavaScript to mimic browser buttons such as close, minimize, and maximize. Also, the popup contains an iframe HTML tag that points to the attacker's

server hosts that mimic the SSO login for Google to be displayed to the users. The iframe phishing link is made by any available phishing tool, and in our case, we will use the PyPhisher tool on the Linux operating system. PyPhisher is a python-based tool that offers phishing links for various social platforms such as Twitter, Facebook, and others [25]. For the sake of clarity, the attack steps are the following:

1. The URL for the malicious website that is controlled by the attacker is sent to the victim via social engineering.
2. The victim enters a malicious website by clicking on the URL. This can be done using any known web browser, such as Chrome, Firefox or any others.
3. The victim then selects the SSO as a login option, which shows a login popup for Google.
4. The phishing link placed in the iframe HTML tag waits for the victim to log in to capture their credentials.
5. The victim provides his or her login information via the username and password parameters.
6. The login popup indicates that there is a login error.
7. The attacker on the background of this scenario captures the credentials from the phishing link and proceeds to the next step of the attack.

3.2 BitB experimental results

In this section, the main objective is to discuss the implementation of the BitB attack from the perspective of both the attacker and the victim. An examination of the steps involved in executing the attack will be conducted, as well as the tools and processes utilized by the attacker. The purpose of this section is to provide a comprehensive and informative discussion that sheds light on the mechanics of the BitB attack by carefully examining these components. Additionally, it provides insight into its potential impact on targeted systems. Overall, this section will serve as a valuable resource for individuals seeking to gain a deeper understanding of the BitB attack and its implications.

The SSO pop-up was selected for the experiment because it allows login at any website through a trusted third party that is not easily suspected by the victim, and it has a known URL link. The same experiment may be tried with any websites that offer third-party SSO services (Facebook, Microsoft, Apple, etc.). The scenario

is the one described in the “Browser-In-The-Browser (BitB) Attack” section. The testbed is set up as follows:

- Victim: The victim visits the attacker's website through the social engineering techniques providing him with the website link or simply by searching the internet and reaching our website.
- Attacker's platform: is set up on GNU/Linux distribution for its easily customizable.
- Web application target: The attacker's website accessible through any browser, has been selected as the target. It is assumed the victim owns an active account within Google which enables them to log in through Google SSO service. Also, google SSO was selected on account of being very popular and the method here in.
- produced applies exactly in the same way to any website that provides SSO service (Microsoft, Apple, etc.).

As highlighted in Figure 5, the user reaches the website that was created by the attacker through browsing the internet or receiving the link using social engineering techniques.

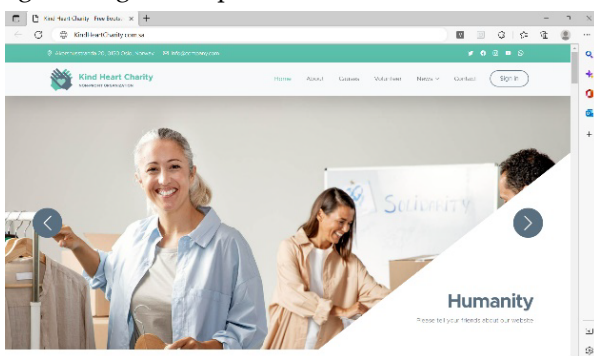


Figure 5: A website made by the attacker

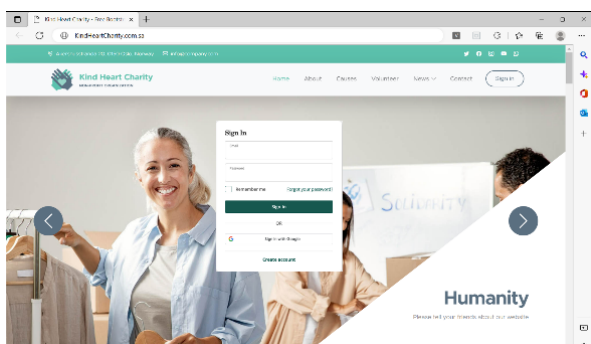


Figure 6: The user tries to sign in to the attacker's

The user tries to log in to the website, he starts filling in the credentials if he/she already has an account on the attacker's website or logs in through Google SSO service. Otherwise, he/she can create an account on the attacker's website. In our case let's assume the victim preferred to sign in through his/her google account.

After the victim clicks on "Sign in with Google" as shown in Figure 6 the fake popup within the attacker's website will appear. The SSO popup is created using HTML, CSS, and JavaScript in a way that looks like the original popup and is hard for the victim to detect as a fake. As shown in Figure 7, the popup's URL looks legitimate and includes a padlock symbol to lure the user into believing there is a secure connection. The phishing link is placed in the iframe HTML tag, so it will appear inside the popup. As demonstrated in Figures 7 and 8, the victim will start entering his credentials by entering his e-mail and then his password. After stealing the victim's credentials, the attacker has the option to perform, depending on the attacker's scenario: either redirect the victim to the Google website or close the fake popup and so on.

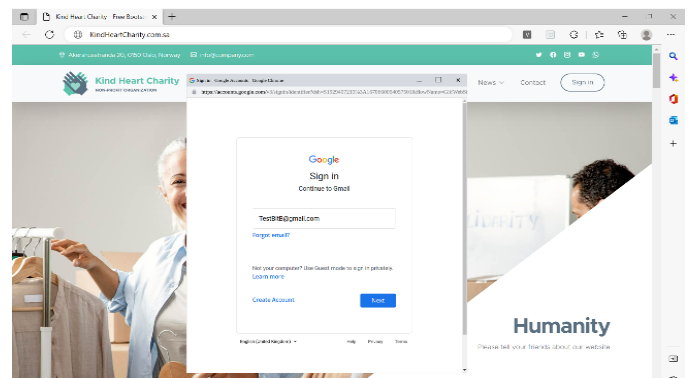


Figure 7: Attacker website provides a fake SSO

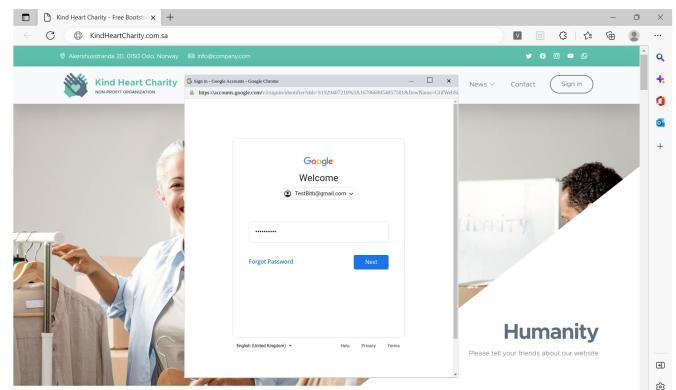


Figure 8: Attacker websites collect users' credentials through a fake SSO (BitB attack)

On the attacker side, all google account credentials of victims will be received as shown in Figure 9. As a further step to bypass two-factor authentication, the attacker can specify a text box for filling in the authentication code which can appear after Figure 7. At this moment the attacker starts filling in these credentials on the legitimate google, and when it asked for the credentials code it will be received by the attacker server when the victim filled it in on the attacker's website.

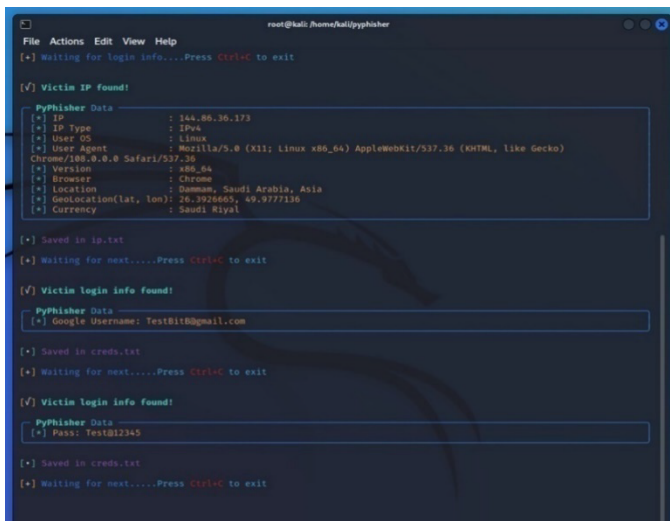


Figure 9: Attacker is capturing the Google account credentials of the victim

As the simple example here documented demonstrates, it was possible to carry on a successful attack without exploiting zero-day or any other known vulnerability at the two endpoints (the victim PC and the official SSO service, instead attacker obtains its own fake SSO) or in the communication channel. Also, the attack was entirely conducted in a remote location, simply by improper use of known technologies.

4. Conclusion

The present study shows that the attack is not easy to discover, since the URL matches the legitimate address. From the user side, the best practice to avoid this kind of attack is to put extreme care into identifying the target SSO service, by trying to interact with the address bar and padlock image before filling in credentials (e.g., if BitB is the case, the address bar will be just a CSS and HTML code not interactable) and, after that, try dragging the suspect window outside the main browser window that contains it. A real browser window will behave independently, while a fake browser window will be “imprisoned” inside the real window it’s shown in.

This paper aims to be the first publication to explain and analyze the recently appeared BitB attack, which is considered a serious threat to web users, especially those who used to log into websites with SSO services. Additionally, it introduces some temporary manual countermeasures to protect against this attack, since no automated solution discovered yet. Furthermore, it shows the experimental results of the BitB attack and its significant impact on a user not sufficiently aware of the risks behind SSO services. Since it aims to steal user credentials as it is in this attack.

In future work, the authors intend to innovate an automatic solution for BitB that prevents and protects the user from such an attack. The solution will be a web extension or plugin that is used to detect BitB attacks based on a set of thoughtful indicators. This will enable the user to be warned before they fall victim to this attack.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

The authors acknowledge SAUDI ARAMCO Cybersecurity Chair for the support.

References

- [1] B. Geyik, b. Erensoy and E. Kocyigit, "Detection of Phishing Websites from URLs by using Classification Techniques on WEKA," Coimbatore, India, 2021.
- [2] S. D. Singh, "BITB (browser in the browser) Attack," InfoSec Write-ups, 14 April 2022. [Online]. Available: <https://infosecwriteups.com/bitb-browser-in-the-browser-attack-e2008c405701>
- [3] V. Lisa, "Browser-in-the-Browser Attack Makes Phishing Nearly Invisible," Threatpost [Blog], 2022.
- [4] R. M. Bian, "Alice in battlefield: an evaluation of the effectiveness of various UI phishing warnings," 2013. [Online]. Available: <https://www.cs.auckland.ac.nz/compsci725s2c/archive/termpapers/725mbian13.pdf>. [Accessed 19 September 2022].
- [5] C. Jackson, D. Simon, D. Tan and A. Barth, "An evaluation of extended validation and picture-in-picture phishing attacks," in International Conference on Financial Cryptography and Data Security, 2007.
- [6] "Novel Phishing Technique Browser-in-the-Browser Attack Targets Government Websites," June 2022. [Online]. Available: <https://cloudsek.com/>. [Accessed 9 October 2022].
- [7] L. Grustniy, "Browser-in-the-browser attack: a new phishing technique," Kasperskay, 25 April 2022. [Online]. Available: <https://www.kaspersky.com/>. [Accessed 16 September 2022].
- [8] Lebedev and D. Eroshev, "Hackers use the browser-in-the-browser technique to steal Steam accounts," 13 September 2022. [Online]. Available: <https://blog.group-ib.com/steam>. [Accessed 21 September 2022].
- [9] K. Jain and B. Gupta, "A survey of phishing attack techniques, defence mechanisms and open research challenges," Enterprise Information Systems, vol. 16, no. 4, pp. 527-565, 2022.
- [10] A.-P. W. Group, "Phishing Activity Trends Report, 1st Quarter 2022," 2022.
- [11] N. Chandru, "A Review on Phishing Attacks and Anti-Phishing Browser Plugins," International Journal of Computer Science & Engineering Technology (IJCSSET), vol. 9, no. 5, pp. 51-58, 2018.
- [12] S. M. Mohamed, N. Abdelbaki and A. F. Shosha, "Digital forensic analysis of web-browser based attacks," in The Steering Committee of The World Congress in Computer Science,

- Computer Engineering and Applied Computing (WorldComp), USA, 2016.
- [13] J. Andress, "Chapter 3 - Authorization and Access Control," in *The Basics of Information Security (Second Edition)*, Syngress, 2014, pp. 39-56.
- [14] G. F. He, T. Zhang, Y. Y. Ma and J. X. Fei, "Protecting User's Privacy from Browser-Based Attacks," in *Applied Mechanics and Materials*, 2014, pp. 941-945.
- [15] F. Tommasi, C. Catalano and I. Taurino, "Browser-in-the-Middle (BitM) attack," *International Journal of Information Security*, vol. 21, no. Springer, pp. 179-189, 2022.
- [16] M. Conti, N. Dragoni and V. Lesyk, "A survey of man in the middle attacks," *IEEE communications surveys*, vol. 18, no. 3, pp. 2027-2051, 2016.
- [17] P. J. Kumar, W. Hu, X. Li and K. Lal, "Mobile Banking Adeptness on Man-In-The-Middle and Man-In-The-Browser Attacks," *IOSR Journal of Mobile Computing \& Application*, vol. 4, pp. 13-19, 2017.
- [18] Y. Sun, S. Zhu, Y. Zhao and P. Sun, "Let Your Camera See for You: A Novel Two-Factor Authentication Method against Real-Time Phishing Attacks," *arXiv preprint arXiv:2109.00132*, 2021.
- [19] D. DAS, "What Is a Browser-in-the-Browser Attack and How Can You Protect Yourself?," *makeuseof*, 24 June 2022. [Online]. Available: <https://www.makeuseof.com/what-is-browser-in-the-browser-attack/>. [Accessed 2022].
- [20] M. G. Alkhozai and O. . A. Batarfi, "Phishing websites detection based on phishing characteristics in the webpage source code," *International Journal of Information and Communication Technology Research*, vol. 1, no. 6, pp. 283-291, 2011.
- [21] E. A. Mallik, "Man-in-the-middle-attack: Understanding in simple words," *International Journal of data and Network Science*, vol. 2, no. 2, pp. 109-134, 2018.
- [22] Mr.d0x, "Browser In The Browser (BITB) Attack," 15 March 2022. [Online]. Available: <https://mrd0x.com/browser-in-the-browser-phishing-attack/>. [Accessed 20 September 2022].
- [23] "Browser in the Browser" attacks: A devastating new phishing technique arises," 1 April 2022. [Online]. Available: <https://www.techrepublic.com/>. [Accessed 19 9 2022].
- [24] B. Toulas, "Hackers steal Steam accounts in new Browser-in-the-Browser attacks," 12 September 2022. [Online]. Available: <https://www.bleepingcomputer.com/news/security/hackers-steal-steam-accounts-in-new-browser-in-the-browser-attacks/>. [Accessed 20 9 2022].
- [25] M. Shariq, "Pyphisher - simple python tool for phishing," *GeeksforGeeks*, 21 April 2022. [Online]. Available: <https://www.geeksforgeeks.org/pyphisher-simple-python-tool-for-phishing/>. [Accessed 10 December 2022].

Copyright: This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY-SA) license (<https://creativecommons.org/licenses/by-sa/4.0/>).