6ZC-07

Analysis and countermeasure of cross-site leak attacks

Chen Duo† Noriaki Yoshiura†

Department of Information and Computer Sciences, Saitama University†

1 Introduction

XS-Leaks (Cross-Site Leaks) is a type of cyber attack involving cross-site scripting to exfiltrate sensitive information from a website. XS-Leaks attacks involve injecting malicious code into a web page, which can be executed by a user's browser when the page is loaded.

Google has published technical information and related papers on XS-Leaks. Google also analyzed XS-Leaks and proposed various attack techniques and countermeasures [1,2].

In an XS-Leaks attack, an attacker injects a script that can obtain sensitive information, such as user credentials, from a website and exfiltrate it to a server controlled by the attacker. This can be done using various methods. Using an iframe is one of the methods to send information to a remote server. XS-Leaks attacks can be hazardous because it is hard to detect them. XS-Leaks attacks do not typically leave any visible trace on websites. Website developers and administrators must prevent XS-Leaks attacks and protect websites against XS-Leaks. Countermeasures can include input validation, sanitization, and proper use of content security policies.

This paper selects one attack method called iframe counting from XS-Leaks; Concretely, this paper implements XS-Leaks attacks using native JavaScript and tests the attacks on some famous websites.

2 XS-Leaks Attacks

Luka Knittel analyzed XS-leaks attack methods and proposed 14 new attack methods [3]. There are six elements in XS-Leaks attacks.

(1) Vulnerable Web: Vulnerable Web is where attackers want to exfiltrate some information.
(2) Attacker's Web: Attackers' Web is the web that contains exploits. Victims access the Attacker's Web.
(3) Inclusion Method: The inclusion Method is used to load the Vulnerable Web from the Attacker's Web.
(4) Leak Technique: After victims access the vulnerable web, the Leak Technique is used to differentiate the potential statuses of the user on the vulnerable web.
(5) States: Vulnerable web has two possible states. The states depend on the victim on the vulnerable web.
(6) Detectable Differences: Detectable Differences are used so that the attacker decides the user's status for vulnerable webs.

3 Proposal Method

To realize the attack, this paper implements a phishing website. The front end and back end are implemented based on native JavaScript. We can configure the node.js environment to attack target websites. In this research, window.open method is chosen to open target websites, and iframe counting is the leak technique used to leak information of web users. By using window.open and iframe counting, we can open the target webpage and count the number of iframe on the web page.

3.1 Inclusion Method - window.open

The window.open method is a JavaScript method that allows a web page to open a new browser window or a new tab in the same browser window. It can open a new window with a specified URL or a new one with a specified size and appearance. This research uses window.open method on the phishing website. After the user opens the phishing website, the target website will be opened automatically, and we can count the number of the iframe.

3.2 Leak Technique – iframe counting

An iframe (inline frame) is an HTML element that allows a web page to embed another HTML document within it. It is often used to display content from another website on a page, such as a video from YouTube or a map from Google Maps. Contents in the iframe are treated as a separate document and can be styled and interacted with independently of the parent page. Since iframe is widely used on websites, we can find that the number of iframe often changes when other pages are opened on the website or when the webpage jumps. Changes in the number of iframes will cause web users to expose some of their information. By counting the number of iframe on the page before the user logs in and the number of iframe after the user logs in, we can find whether the user has logged in to the website and thus determine whether the user has a member of the website.

3.3 Attack Flow

According to the attack elements mentioned in section 2, the proposed attack has six parts.

(1) Vulnerable Web: Vulnerable Web is the login page of a target website.
(2) Attacker's Web: A fishing website can cheat the user, and the user will open it.

(3) Inclusion Method: On the fishing website, this research uses the window.open method to open the target website.
(4) Leak Technique: This research records the number of iframe on the webpage before and after the user logins.
(5) States: In a target website, the number of iframe is different between before and after login.
(6) Detectable Differences: If the number of iframe changes, the attacker can know whether the user has been login the website and whether the user has a membership to this website.

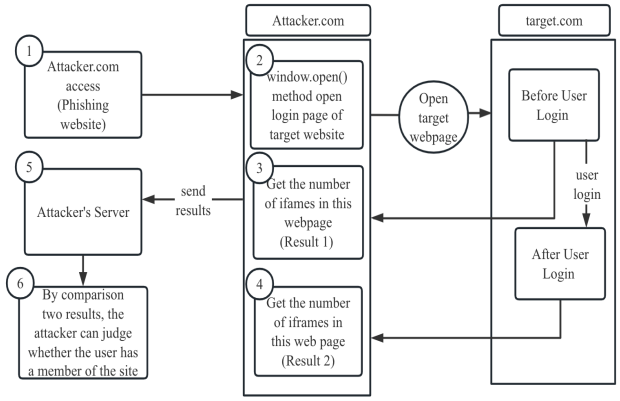

Figure.1 Attack flow of an XS-Leaks attack through iframe counting

As shown in Figure.1, the attack page will automatically open the target website's login page after a victim opens the phishing website. The attack page will record the number of the iframe and send "result 1" to the attacker's server or record it in the attacker's file. After the victim login on to the target webpage, the attack page will record the number of the iframe and then send "result 2" to the attacker's server. The attacker can compare two results; if the number of the iframe has been changed, the victim logins on to the webpage, and the attacker knows that the victim has a membership on this website. If the number of the iframe does not change, the attacker can know that the victim does not log in to the webpage.

4 Experiment and Evaluation

This paper first tested a well-known website A to verify the proposed method. Before the test attack, we knew that the login page of this website A did not use iframe, but the page after login used four iframes to embed advertisements. Table 1(1) shows the result of the test attack.

Consequently, this paper tested ten websites and used three different browsers Chrome, Firefox, and Safari. However, the difference in browsers does not change the experiment's results.

No	Types of the website	Test website	Before login	After login	Success or failure
(1)	Business	A site	0	4	✔
(2)	eCommerce	B site	0	2	✔
(3)	Mail	C site	0	6	✔
(4)		D site	2	4	✔
(5)	Entertainment	E site	0	0	✕
(6)	Bank	F site	2	3	✔
(7)		G site	0	2	✔
(8)	Business	H site	4	4	✕
(9)	Video sharing	I site	2	3	✔
(10)	Service provider	J site	2	6	✔

Table.1 Results of the test attacks (Chrome, Firefox, and Safari obtained the same results)

According to the test results in Table 1, we can find that the number of iframes on some websites changes before and after login. Through these changes, the attacker can infer whether the victim has logged into the website and whether the victim has a membership of s website. Especially the website of a bank also has this kind of problem. If the attacker knows that the victim has a bank account, the attacker can carry out more attacks to obtain the victims' properties.

As shown in table 1, the proposed method cannot attack the E site and H site. E site does not use iframe in its website. H site uses the same number of iframe on its website, and the attacker cannot know whether the victim logs in. Although there is no single way to defend against this attack directly, we can avoid this attack in several ways:

(1) Do not use iframe on web pages.
(2) Use the same number of the iframe on web pages.
(3) Set browsers so that the browsers cannot open other pages or window.open is prohibited.

5 Conclusion

This paper has proposed an XS-Leaks (Cross-Site Leaks) attack using iframe counting to expose a web user's login to a target website and whether the user has a membership of the target website. This paper test ten different websites and three different web browsers. In addition, three simple defense methods have been proposed.

References

[1] https://xsleaks.dev,XS-Leaks Wiki, 2021.

[2] Luan Herrera. XS-Leaks in redirect flows. https://blog.lbherrera.me/posts/ XS-Leaks in redirect flows

[3] Lukas Knittel, Christian Mainka, Marcus Niemietz. XSinator.com: From a Formal Model to the Automatic Evaluation of Cross-Site Leaks in Web Browsers. (CCS)2020.