Swanirman Sunirmit Publications of Research Volume 4, Issue 3-September 2024 | [2024-25]
ISSN [Online]: 2583-2654

Utility of CSRF Token in Application Security

[1]Ajit Saha,
Scientist-F, National Informatics Centre,
A-Block, CGO-Complex,
Lodhi Road, New Delhi-110003

Abstract: Cross Site Request Forgery (CSRF) is a web application vulnerability that allows an attacker to induce application users to perform actions that do not intent to perform. It allows an attacker to bypass the same origin policy which is design to prevent interference from other website. The basic mechanism to prevent CSRF vulnerability is to make the request as a legitimate request. Only the legitimate request should come to server. Now how we can prove the request is legitimate.

The legitimacy may be checked by using two things

- The request should be generated from same origin.
- Every request should be uniquely identified by Server.

The authentication and authorization provides authenticity of the request but CSRF provides validity of request whether the request has been originated from same server or not. The CSRF token can prevent the CSRF attack.

I. INTRODUCTION

Common protections from CSRF attacks.

The attackers perform CSRF attack bypassing the anti-CSRF measures implemented in server and clients.

The most common practices for protecting applications from CSRF attacks are as follows :

CSRF tokens - A CSRF token is a short-life, unique, secret, and unpredictable alpha-numeric value that is generated by the server-side application and shared with the client. The client must include the CSRF token in the request and submit to server. The server will check the CSRF token and accept the request otherwise server will reject the request. This makes it very difficult for an attacker to construct a valid request on behalf of the victim.

SameSite cookies - SameSite prevents the browser from sending this cookie along with cross-site requests. The main goal is to mitigate the risk of cross-origin information leakage. It also provides some protection against cross-site request forgery attacks. Possible values of attribute of the cookie are none, lax, strict.

The strict attribute of cookie will prevent sending cookie to cross-site target server. The strict attributed cookies can only be send to same origin.

A bank website however most likely doesn't want to allow any transactional pages to be linked from external sites so the strict flag would be most appropriate here.

The lax value provides a reasonable balance between security and usability for websites that want to maintain user's logged-in session after the user arrives from an external link.

The none value won't give any kind of protection. The browser attaches the cookies in all cross-site browsing contexts.

The default value of the SameSite attribute differs from browser to browser, therefore it is better to explicitly set the value of the attribute.

Referrer-based validation - Some applications make use of the HTTP Referrer header to attempt to defend against CSRF attacks, normally by verifying that it can be ensure that the request originated from the application's own domain. If referrer url is something else then the request can be rejected.

ISSN [Online]: 2583-2654

II. WHAT IS CSRF TOKEN?

A CSRF token is a short life unique security number design to protect web application from unauthorized or malicious requests. It is a special type of token, often referred to as a synchronized token or challenged token, that verifies the legitimacy of the request generated by the user. Each CSRF token is unique to the user request. This token is very crucial for security.

III. FEATURES OF CSRF TOKEN

(1) A CSRF token is a large number and unpredictable in nature.
(2) Making it extremely difficult for attackers to guess or replicate.
(3) The random number generator should use maximum entropy.
(4) It's lifetime should be very short.
(5) CSRF token should be send as a http request header to the server.

IV. CSRF TOKENS ARE CAPABLE TO PROVIDE SECURITY FROM VARIOUS ATTACKS.

4.1 CSRF token can prevent XSS attacks.

Cross-Site Scripting (XSS) is a prevalent critical vulnerability within the realm of web security. An attacker skilfully inject malicious script into a trusted website, subsequently browser will execute the scripts. The consequences of XSS attacks are diverse from the theaft of sensitive information to the defacement of entire website.

There are three main type of XSS attack.

- Stored XSS
- Reflected XSS
- DOM based XSS

Stored XSS: This type of XSS attack involves the permanent storage of malicious scripts on the server. When a user accesses a specific page, these scripts are served, resulting in a attack with far-reaching implications. This type of vulnerability can not be protected by CSRF token.

Reflected XSS: In this scenario, the injected script becomes a tool of attacker. The server, unwillingly, reflects this script to the user's browser and effects start. Normally this script are non-persistent and attacker may send a devastating script and may cause a dangerous effect of it.This type of vulnerability can be protected by CSRF token validation.

DOM based XSS: The attack takes a dynamic turn within the Document Object Model (DOM) of the victim's browser. Operating within this virtual representation of the web page, malicious actors manipulate the structure and content dynamically, potentially wreaking havoc on the user experience. This type of vulnerability can not be protected by CSRF token.

Not all but some XSS attacks can be prevented by using CSRF tokens. Consider a simple reflected XSS vulnerability :

https://myserver.com/status?message=<script>/* +Bad+stuff+here...+*/</script>

Now, suppose that the vulnerable function includes a CSRF token:

https://myserver.com/status?csrf-token=CI12345432a4d2Foz&message=<script>/* +Bad+stuff+here...+*/</script>

Assuming that the server properly validates the CSRF token, and rejects requests without a valid token, then the token does prevent exploitation of the XSS vulnerability. The clue here is in the name: "cross-site scripting", at least in its reflected form, involves a cross-site request. By preventing an attacker from forging a cross-site request, the application prevents trivial exploitation of the XSS vulnerability.

- If a reflected XSS vulnerability exists anywhere else on the site within a function that is not protected by a CSRF token, then that XSS can be exploited in the normal way.
- If an exploitable XSS vulnerability exists anywhere on a site, then the vulnerability can be

leveraged to make a victim user perform actions even if those actions are themselves protected by CSRF tokens. In this situation, the attacker's script can request the relevant page to obtain a valid CSRF token, and then use the token to perform the protected action.

- CSRF tokens do not protect against stored XSS vulnerabilities. If a page that is protected by a CSRF token, , then that XSS vulnerability can be exploited in the usual way, and the XSS payload will execute when a user visits the page.

Therefore it is better not to use GET method which can give chance to exploit the reflected XSS.

4.2 CSRF token can prevent DOS and DDOS attacks.

CSRF token can protect applications from DOS and DDOS attack. CSRF tokens have maximum entropy and lifetime is very short. The basic principle of protection frpm dos ans ddos attacks is a single CSRF token for a single transaction in an application. Every application has a fixed number of forms. Each form has its own CSRF token. The tentative time of form filling is the lifetime of CSRF token. No CSRF token generation request will be accepted during the lifetime of CSRF token. If lifetime is over then only new request will be accepted by server. The CSRF token generation will be protected by referrer URL, session Id and the application form id. When a token generation request comes to server , Server will check whether the token is live for the form or not. If token is live then no further token generation request will be accepted. Request will be rejected by server and no repeated request will be accepted by server. Hence DOS and DDOS attacks are not possible by sending repeated requests.

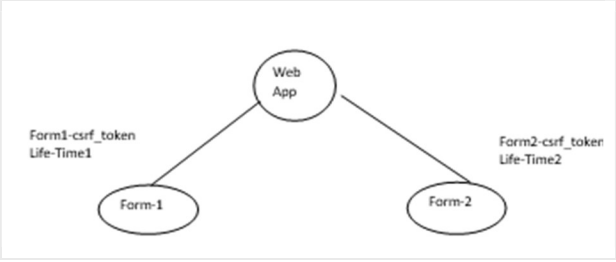

Fig 1

V. CONCLUSION

CSRF token is an important security utility, it can provide a number of protection against various security vulnerability. The technique of implementation of CSRF token depends on the application requirement. While effective, tokens can be exposed at a number of points, including in browser history, HTTP log files, network appliances logging the first line of an HTTP request and referrer headers, if the protected site links to an external URL. These potential weak spots make tokens a less than full-proof solution.

REFERENCES

Have a look on the following articles

- ***Bright articles on CSRF token***
- ***Synopsys : Cross Site Request Forgery***

[1]**Ajit Saha:** The Author is a Scientist and interested in doing Research and Development. Interested area is exploring upcoming technology and identify security breaches of the technology to mitigate security risk.
.
The author has 30 years of long Research and Development experience for various government organizations. The author has long experience on Cyber Security and ethical hacking, works as a Deputy Chief Information Security office in various Ministry. The author has big contribution on design of Cyber Security Framework by following ISO/IEC/27001 family of standard. The author has long teaching experience, had taken part-time classes in various renown Institutions like IETE, JNU, Jamia Hamdard University etc.

An author passionate about cyber security because of several key factors:

1. **Personal Experience**: They may have had a personal encounter with cyber threats, such as a data breach or a security incident, which made them realize the importance of protecting digital assets. This personal experience can create a deep, personal commitment to improving cybersecurity measures.
2. **Growing Threat Landscape**: The increasing prevalence of cyber threats and attacks in today's digital world can be a significant motivator. The complexity and scale of cyber threats—from identity theft to ransomware—highlight the critical need for robust security measures, fueling a passion to tackle these challenges.
3. **Intellectual Challenge**: Cyber security is a field filled with complex problems and constantly evolving technologies. For those with a keen interest in problem-solving and staying ahead of adversaries, the intellectual challenge and the need to continuously learn and adapt can be highly stimulating and fulfilling.
4. **Protecting Privacy and Safety**: A strong desire to protect individuals' privacy and safety online can drive passion. Authors in this field often aim to safeguard personal information and ensure that digital interactions are secure, which can be deeply satisfying and align with their values.
5. **Impact and Relevance**: Cyber security has a direct impact on almost every aspect of modern life, from personal finances to national security. The relevance of the field and its potential to make a significant difference in protecting people and organizations can be a powerful motivator.
6. **Innovative Solutions**: The field of cybersecurity is continually evolving, with new technologies and methods emerging to combat cyber threats. The opportunity to be at the forefront of innovation and to contribute to the development of cutting-edge solutions can be a major driving force.