CSRF & SSRF

In this article we going to learn about CSRF and SSRF, both of these vulnerabilities take advantage of how server process URLs. These are very common and well known vulnerabilities and understanding about these vulnerabilities is very important for web application penetration testing.

First let’s get an introduction about these attacks.

**CSRF:**

* CSRF stands for cross-site request forgery.
* This vulnerability was present in OWASP top 10 list, but was removed after in the edition came after 2017.
* This vulnerability is still present in 5% of the web applications.
* CSRF happen at the client side, in technical terms the forgery happens at the client side.
* The main purpose of CSRF attacks is to force user to take undesirable actions on their online account.
* Suppose we have a website named “example.com” and we are logged in to that website, in this case there will be some cookies that will be stored and these cookies will be use when we make some request to the website to perform some action. Now what happens in CSRF a malicious website can perform the same action using cookies related to the example.com. And this could happen if somehow we visit that malicious website or click on the URL of that website.
* There are many actions that could be performed on a website using this vulnerability such as “changing the password” or “making online transactions”. The action is highly depends on the nature and use case of that website.
* CSRF attacks work because the user is already authenticated to the target website and the forced request includes the cookies containing session information.
* Standard CSRF attacks assume that a user is already authenticated to a website, but CSRF attacks can also be “stored.”

**SSRF**

* SSRF stands for server-side request forgery.
* SSRF attacks are designed to exploit how a server processes external information.
* If an attacker can modify the target URL, they can potentially exfilterate sensitive information from the website or inject untrusted input into it.
* Suppose we have a website named “example.com”, it uses an API to fetch some data from the web server. If we make a request to localhost/admin and get the response than this is a serious problem.
* If we can access the admin panel using the localhost than we do not need to authenticate to use admin panel, because here server is making a request to the localhost to access the /admin page and it is displaying the result on the client side.
* The vulnerability associated with SSRF vulnerabilities is not limited to data exfiltration. In some cases application may be designed to read data from URL. If this URL is trusted, the application may not perform data validation. This could allow an attacker to provide malicious input that could exploit a buffer overflow, integer overflow, SQL injection or other vulnerability in the web application.
* The impact of SSRF vulnerability can be significant. A recent example of an attack exploiting SSRF (and the difficulty of protecting against it) is the Capital One data breach(https://www.americanbanker.com/news/capital-one-to-pay-80m-in-connection-with-massive-data-breach#:~:text=The%20hack%20compromised%20personal%20data,were%2080%2C000%20bank%20account%20numbers.), which expose the personal information of 106 million people.

**Difference between CSRF and SSRF:** Both CSRF and SSRF vulnerabilities take advantage of how a web server handles URLs. However, they differ on the bases of target of the attack and purpose of the attack.

**Attack Target**

* The target of a CSRF attack is the user. While it accomplished suing flaws in how the website is designed, its purpose is to perform legitimate but unauthorized actions on the user’s account with the web-based service.
* SSRF on the other hand is designed to primarily target the server. While, in the long run, the attack may affect users of the service, the primary purpose of the attack is theft of sensitive information on the server or exploiting other vulnerabilities by using SSRF to bypass input validation countermeasures.

**Attack purpose**

* In the case of SSRF, the primary purpose of the attack is to gain access to sensitive information/data. This could be performed directly (by forcing it to write data to an attacker-supplied URL) or indirectly (by allowing exploitation of a vulnerability that can be used to steal data).
* CSRF vulnerabilities, on the other hand, do not provide an attacker with any access to sensitive data. While the attacker forces a user’s browser to visit the target site, the actual request and response are performed independently. Even if the attack results in sensitive data being sent in response to the malicious request, this data only goes to the target user’s computer, not the attacker’s. The purpose of exploiting CSRF vulnerability is to force the target user to take action in the attacker’s interest, like changing an account password to one known to the attacker.

**Conclusion:** While CSRF and SSRF vulnerabilities are very different, they both enable by the same problem “a failure to properly use URLs by the server.” When looking for potential vulnerabilities in a website, examining how the website uses URLs and the types, formats and destination of request made to or by it can help in identification of these vulnerabilities.

So this was all about CSRF and SSRF. Hope you liked it and learned something new from it.

If you have any doubt, question, quires related to this article or just want to share something with me, than please feel free to contact me.