**Server-Side Request Forgery (SSRF**) is a request-based attack in which an intruder intercepts with the API request from an application to the server and modifies the application generated HTTP request to get access to unauthorized information such as administration credentials and such.

SSRF can be generic and blind, while in blind SSRF vulnerability, the attacker doesn’t receive any response from the spoofed server. Based on the location of the application hosted, SSRF can be orchestrated on the internal server (localhost), external third-party server and external network.

SSRF = Server-Side Request Forgery. An attacker uses or abuses server functionality to access or modify resources. Attacker targets applications that support data imports from URLs or allows them to read data form URLs.

Normal SSRF: In this SSRF vulnerability, the response to the API request is sent back to the app and the attacker can see the response as well.

Blind SSRF: In blind SSRF, the response is not sent back to the application. As such, the attacker has to device ways to confirm the vulnerability.

SSRF attack against the hosting server: In this type of attack, the attacker sends HTTP request to the server that’s hosting the application, via the loopback net interface. It involves supplying a URL with a hostname like 127.0.0.1 or localhost.

How to avoid SSRF

* Never trust user input
* Use whitelist for IP and domains.
* Always validate if response has the expected format and content.

Attackers can get access to URLs in the server & re-exploit the vulnerability of the server to modify the server content. Attackers can also scan for other applications running in the same network using SSRF vulnerability by scanning the ports.

SSRF attacks can be done in the following ways:

* Within the hosting server
* Within the network
* Within different network/server

SSRF attack against the hosting server is made by making an HTTP request to the server using its loopback network interface. This involves URL with hostname 127.0.0.1 or localhost. Since the attacker uses the URL via the application, the server bypasses the access control.

Bypassing a blacklist filter: To bypass a blacklist filter, the keyword which is blacklisted (for example- admin) needs to be double encoded so that the server does not find the word in the compromised URL. With a regex search the server normally looks for the blacklisted word in the URL to block the URL.