All this text wasn’t written by me (Tiago a.k.a TigaxMT).

All of it was transcribed by me from HackerOne video lessons.

All credits and thanks go to them.

Server-Side Request Forgery

SETTING THE STAGE

Many web applications will take a URL from a user and do something with it, e.g. request an image to create a thumbnail, check that a link is valid, call webhooks, and more.
While this is by no means a universal behavior, it's something you'll see in a lot of software.

CONTRIVED EXAMPLE

Imagine a reporting dashboard with the ability to poll a URI and pull in the CSV data that it returns.
Every N minutes, it makes a web request and then updated your dashboard show a fresh table.

Many things can go wrong as a result of this.
What if you entered file://etc/passwd ? You may be able to directly read files from the server, if this isn't properly handled.

What if you instead sent this:
http://localhost/admin/add_user?username=cody&password=HackerOne!

The server could make requests to internal services for you; things that you would never be able to ordinarily access due to firewall rules.

PORT SCANNING

If you can control the port on which requests are made, you can pretty trivially do port scans of the internal network. If you see that the request return immediately, this will typically indicate that port is closed. If the request hangs for a few seconds before returning, this often indicates that the port is open but not handling HTTP requests.

DIFFICULTY

SSRF bugs are really simple in concept, but they can be surprisingly difficult to find in practice. This is because they often require traversing many levels of indirection.
You may need to set up some feature, then upload a specific file that contains SSRF payload, then trigger some processing. These bugs often evade discovery for many years because of this.

MITIGATION

There’s no silver bullet for SSRF mitigation, but some combination of these can be effective:

- Limit connections to only port 80(HTTP) and 443(HTTPS) to prevent port scanning
- Resolve the IP of the target host, to ensure that the IP is for that of an external host
- Disable access to any protocol scheme that is not HTTP or HTTPS

REAL WORLD EXAMPLE

A great real-world example is this report for Shopify by HackerOne user floyd, exploiting SSRF via an SVG file: https://hackerone.com/reports/223203