By : https://www.linkedin.com/in/nikitha-yasala-0413a817b/

What is SSRF

Server-Side Request Forgery (SSRF) is a type of security vulnerability that allows an attacker to make requests from a vulnerable server to other internal or external services. This can lead to various malicious outcomes, such as:

1. Accessing Internal Resources: If a web application can make requests to internal services (like databases or internal APIs), an attacker might exploit SSRF to gain access to sensitive information.

2. Port Scanning: Attackers can use SSRF to probe for open ports on the internal network, potentially discovering additional services to target.

3. Exfiltration of Data: By making requests to internal services, attackers can retrieve data that should not be accessible externally.

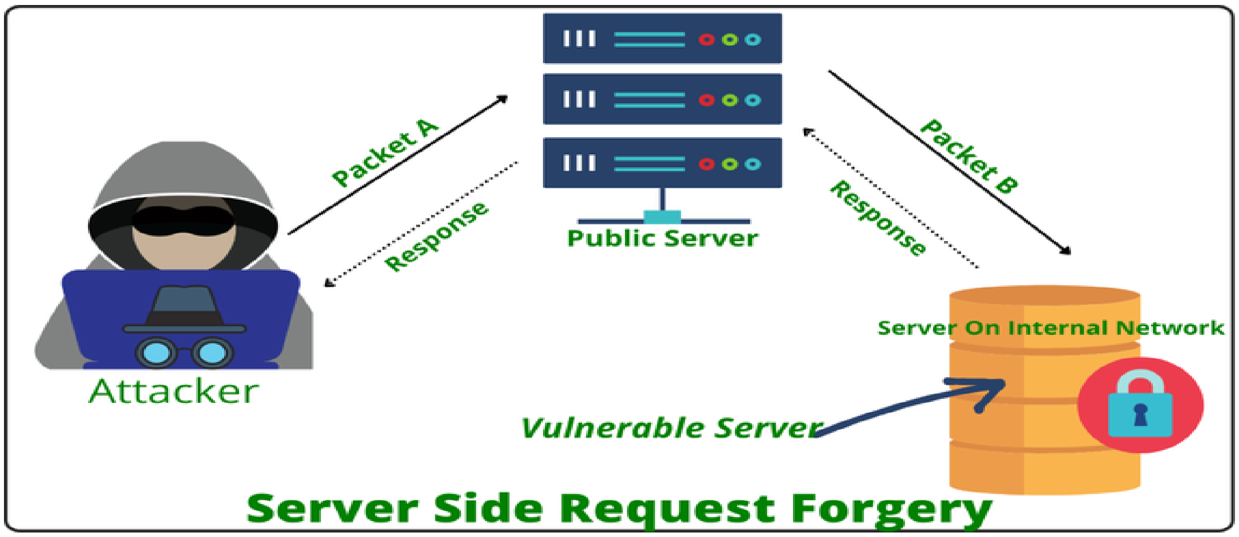
4. Bypassing Security Controls: SSRF can allow attackers to bypass network security measures that only allow certain traffic, enabling them to interact with systems that are otherwise protected.

How SSRF Works:

1. Vulnerability: An application accepts user input (like a URL) and makes requests based on that input without proper validation or sanitation.

2. Exploitation: An attacker crafts a request that the server processes, potentially pointing it to internal services or other locations.

3. Impact: Depending on the server's permissions and the resources it can access, the attacker can retrieve sensitive information or perform other malicious actions.



Prevention Measures:

Input Validation: Ensure that user input is properly validated and sanitized. Reject requests to internal addresses or sensitive endpoints.

Network Segmentation: Limit the ability of the server to make requests to internal services through network configuration.

Use of White lists: Implement white listing for URLs that the application can access, ensuring that only approved endpoints can be contacted.

Understanding SSRF is crucial for developers and security professionals to protect applications from these vulnerabilities.

Types of SSRF

Server-Side Request Forgery (SSRF) can be classified into several types based on how the vulnerability is exploited and the impact it can have. Here are some common types:

1. Internal SSRF

This type targets internal services within the same network as the vulnerable server. Attackers can access internal APIs, databases, or other services that are not exposed to the internet.

2. External SSRF

In this scenario, the attacker tricks the server into making requests to external services. This can be used to access third-party APIs or services that the server interacts with, potentially leading to data leakage.

3. Blind SSRF

In blind SSRF, the attacker does not receive direct feedback from the vulnerable application about the success of the request. Instead, they may rely on timing, response codes, or other indirect indicators to infer information about the internal services.

4. DNS Rebinding

This involves manipulating DNS responses so that the server makes requests to an attacker's server that then point to internal resources. This can help bypass same-origin policies and allow the attacker to access internal data.

5. HTTP Request Smuggling

In some cases, attackers can exploit SSRF to perform HTTP request smuggling, which involves sending crafted requests that confuse the server and potentially manipulate how requests are processed.

6. Port Scanning

Using SSRF, attackers can probe internal networks for open ports and running services, helping them to map the network and find additional attack vectors.

7. File Retrieval

This type involves tricking the server into fetching files from internal resources. Attackers may use SSRF to retrieve sensitive files such as configuration files, credentials, or other confidential information.

8. Denial of Service (DoS)

An attacker may exploit SSRF to overwhelm internal services by making numerous requests, potentially leading to denial of service for legitimate users.

Prevention and Mitigation

To protect against these types of SSRF attacks, organizations should implement robust input validation, use network segmentation, and apply strict whitelisting policies for outgoing requests.

Understanding the various types of SSRF can help developers and security teams implement effective defenses and safeguard their applications.

Impact and mitigation measures of SSRF

Impact of SSRF

Server-Side Request Forgery (SSRF) can have several serious impacts, including:

1. Unauthorized Access: Attackers can access internal services that should not be exposed, potentially gaining sensitive information like API keys or database credentials.

2. Data Leakage: SSRF can lead to the exfiltration of sensitive data from internal systems or databases, compromising confidentiality.

3. Network Scanning: Attackers can map out internal network services and their configurations, making it easier to identify additional targets.

4. Denial of Service (DoS): By sending numerous requests to internal services, attackers can overwhelm resources, leading to service outages.

5. Bypassing Security Controls: SSRF can be used to circumvent firewall rules or other security mechanisms designed to restrict access to internal resources.

6. Web Application Compromise: In some cases, SSRF can lead to further attacks on the web application, potentially allowing remote code execution or other severe vulnerabilities.

Mitigation Strategies

To effectively mitigate SSRF vulnerabilities, consider the following strategies:

1. Input Validation and Sanitization:

Validate and sanitize all user inputs, especially URLs. Reject any requests that contain unexpected patterns or point to internal IP addresses.

2. Use of White lists:

Implement a white list for allowed domains or IP addresses. Only permit requests to known, trusted services.

3. Network Segmentation:

Segregate sensitive internal services from public-facing servers. Use firewalls to restrict access between segments.

4. Limit Server Permissions:

Run the application with the least privileges necessary. This limits the potential damage if an SSRF attack is successful.

5. Monitor and Log Requests:

Implement logging of outgoing requests to detect any suspicious activity. Regularly review logs for anomalies.

6. DNS Control:

Use a DNS service that prevents DNS rebinding attacks, ensuring that requests cannot be redirected unexpectedly.

7. Implement Rate Limiting:

Apply rate limiting to outgoing requests to prevent abuse through excessive requests, which can lead to DoS attacks.

8. Use of Security Tools:

Employ web application firewalls (WAFs) and other security tools that can help detect and block SSRF attempts.

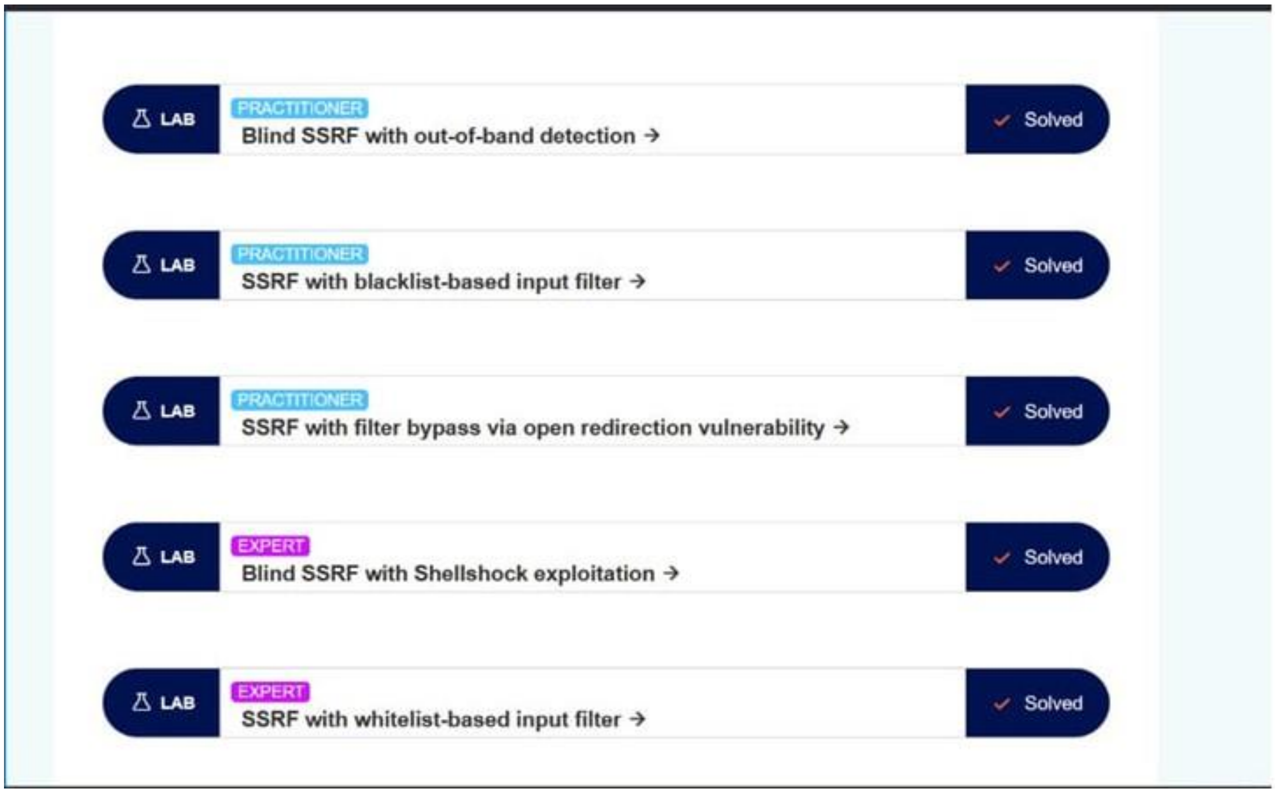
9. Regular Security Audits:

Conduct regular security assessments and penetration testing to identify and remediate potential SSRF vulnerabilities.

By understanding the impact of SSRF and implementing these mitigation strategies, organizations can significantly reduce their risk and enhance their overall security posture.

Practical:

https://portswigger.net/web-security/all-labs#server-side-request-forgery-ssrf



LAB PRACTITIONER Blind SSRF with out-of-band detection → ✓ Solved

LAB PRACTITIONER SSRF with blacklist-based input filter → ✓ Solved

LAB PRACTITIONER SSRF with filter bypass via open redirection vulnerability → ✓ Solved

LAB EXPERT Blind SSRF with Shellshock exploitation → ✓ Solved

LAB EXPERT SSRF with whitelist-based input filter → ✓ Solved

REFERENCES:

https://www.acunetix.com/blog/articles/server-side-request-forgery-vulnerability/

https://brightsec.com/blog/ssrf-attack/

https://brightsec.com/blog/7-ssrf-mitigation-techniques-you-must-know/

https://portswigger.net/web-security/all-labs#server-side-request-forgery-ssrf