

# Time-Based SSRF

We can also determine the existence of an SSRF vulnerability by observing time differences in responses. This method is also helpful for discovering internal services.

Let us submit the following document to the PDF application of the previous section and observe the response time.

Code: **html**

```
<html>
  <body>
    <b>Time-Based Blind SSRF</b>
    
  </body>
</html>
```

Request

PrettyRawHex\n

15-----40369444
744947618934290228237
Content-Disposition: form-data; name="file"; filename="exfil.html"
Content-Type: text/html
<html>
 <body>
 <b>Time-Based Blind SSRF</b>
 
 </body>
</html>
25-----40369444
744947618934290228237--
26

Response

PrettyRawHexRender\n

1 HTTP/1.0 302 FOUND
2 Content-Type: text/html; cha
3 Content-Length: 242
4 Location: http://ubuntu-web.
5 Server: Werkzeug/2.0.2 Pythc
6 Date: Fri, 22 Oct 2021 18:04
7
8 <!DOCTYPE HTML PUBLIC "-//W3
9 <title>
 Redirecting...
10 </title>
11 <h1>
 Redirecting...
12 </h1>
13 <p>
 You should be redirected a
14

INSPECTOR

Request Attributes
Query Parameters (0)
Body Parameters (1)
Request Cookies (0)
Request Headers (12)
Response Headers (5)

Done462 bytes | 10,084 millis

We can see the service took 10 seconds to respond to the request. If we submit a valid URL inside the HTML document, it will take less time to respond. Remember that `internal.app.local` was a valid internal application (that we could access through SSRF in the previous section).

Request

PrettyRawHex\n

14-----403694447449476189
34290228237
Content-Disposition: form-data; name="file"; filename="exfil.html"
Content-Type: text/html
<html>
 <body>
 <b>Time-Based Blind SSRF</b>
 
 </body>
</html>
25-----403694447449476189
34290228237--
26

Response

PrettyRawHexRender\n

1 HTTP/1.0 302 FOUND
2 Content-Type: text/html; charset=utf-8
3 Content-Length: 242
4 Location: http://ubuntu-web.lalaguna.loc
5 Server: Werkzeug/2.0.2 Python/3.8.12
6 Date: Fri, 22 Oct 2021 18:26:20 GMT
7
8 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
9 <title>
 Redirecting...
10 </title>
11 <h1>
 Redirecting...
12 </h1>
13 <p>
 You should be redirected automaticall
14

INSPECTOR

Request Attributes
Query Parameters (0)
Body Parameters (1)
Request Cookies (0)
Request Headers (12)
Response Headers (5)

Done462 bytes | 413 millis

In some situations, the application may fail immediately instead of taking more time to respond. For this reason, we need to observe the time differences between requests carefully.

PreviousNext

Mark Complete & Next

## Table of Contents

Introduction to Server-Side Attacks

## Abusing Intermediary Applications

AJP Proxy

Nginx Reverse Proxy & AJP

Apache Reverse Proxy & AJP

## Server-Side Request Forgery (SSRF)

Server-Side Request Forgery (SSRF) Overview

SSRF Exploitation Example

Blind SSRF

Blind SSRF Exploitation Example

Time-Based SSRF

## Server-Side Includes (SSI) Injection

Server-Side Includes Overview

SSI Injection Exploitation Example

## Edge-Side Includes (ESI) Injection

Edge-Side Includes (ESI)

## Server-Side Template Injections

Introduction to Template Engines

SSTI Identification

SSTI Exploitation Example 1

SSTI Exploitation Example 2

SSTI Exploitation Example 3

## Extensible Stylesheet Language Transformations Server-Side Injections

Attacking XSLT

## Skills Assessment

Server-Side Attacks - Skills Assessment

## My Workstation

OFFLINE

Start Instance

/ 1 spawns left

Powered by

HACKTHEBOX