CHAMELEON

FOR OUR SMARTER WORLD

Penetration Testing (MOP)

FAILED

Name: Muhammad Ahmad Rahman
ID: 222035606

Contents

Executive Summary

In this report, I report my findings after doing some different types of penetration testing which includes input validation, security misconfiguration and most importantly, directory transferal.

Introduction:

What is Penetration testing? Penetration testing is a security practice where testers try to breach a system, network, or web application to identify vulnerabilities and ensure security measures are effective.

In this report, I focus on trying to breach into different directories, to check if the application is vulnerable to unauthorized access to files outside of the designated directory. In addition I also try to assess the server's response to various HTTP header manipulations and potential insecure handling of HTTP requests and identify how the web application handles unexpected input through modifications.

Tools used:

The main tool used within this testing is Burp Suite. In Burp Suite, I used tools such as Intruder and Repeater to conduct these tests and find any known vulnerabilities inside the website.

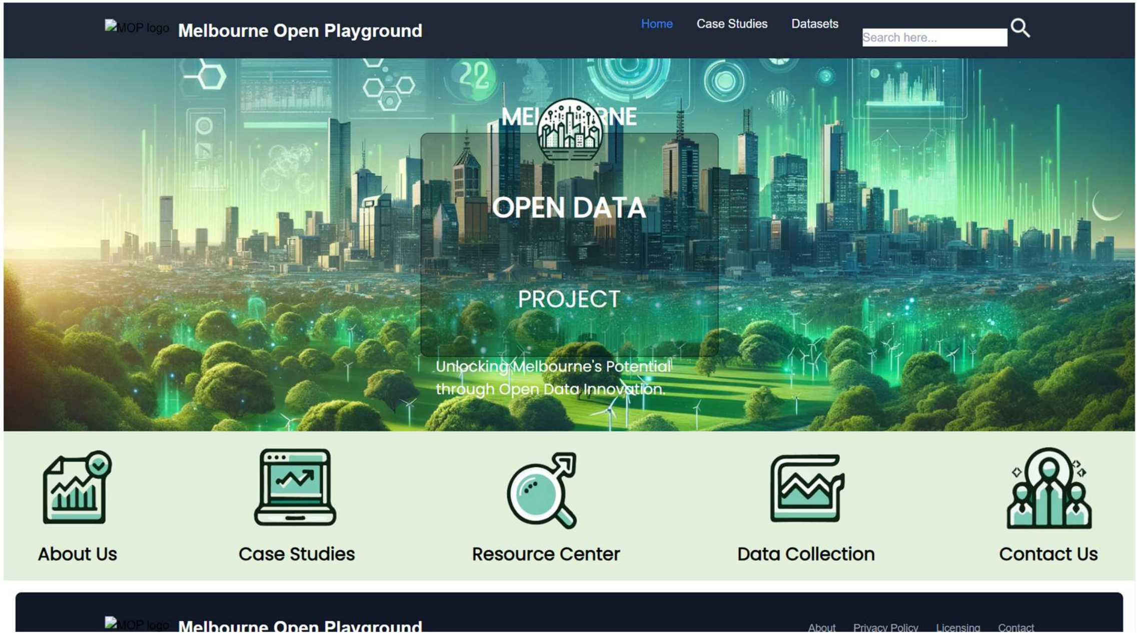
Burp Suite is an integrated platform made up of a suite of tools that work together to support the entire web application testing process, from initial mapping and analysis of an application's attack surface, through finding and exploiting security vulnerabilities.

Burp Repeater is a simple tool for manually modifying and resending individual HTTP requests and analysing the application's responses. It was used extensively in this test to manipulate the requests to explore how different inputs and scenarios affect the application, which is crucial for testing input validation and handling of malformed or malicious data.

Burp Intruder is a tool for performing automated and customized attacks against a web application. It can be used to automate custom attacks on identified parameters, making it invaluable for performing tasks like testing for injections, directory traversal, and error handling in a more systematic and exhaustive manner.

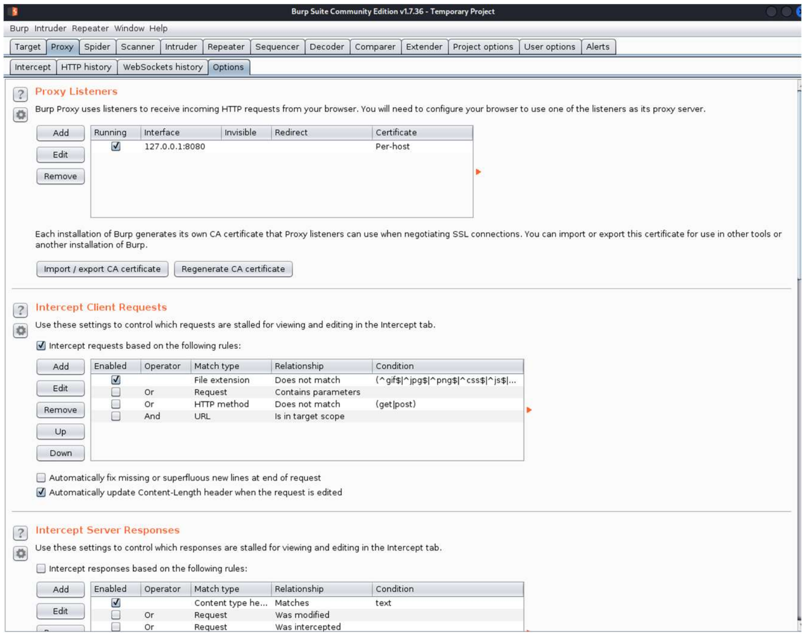
Scope of Testing

The scope of testing for this is the MOP Website



Steps and Results

The first thing I did was to configure Proxy Settings in Burp Suite and set up Burp Proxy to listen on a local port (usually 8080). Ensure it's configured to intercept traffic from your browser.


Burp Suite Community Edition v1.7.36 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Intercept HTTP history WebSockets history Options

Proxy Listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.

Running	Interface	Invisible	Redirect	Certificate
☑	127.0.0.1:8080			Per-host

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating SSL connections. You can import or export this certificate for use in other tools or another installation of Burp.

Intercept Client Requests

Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

☑ Intercept requests based on the following rules:

Enabled	Operator	Match type	Relationship	Condition
☑		File extension	Does not match	(^gif$\|^jpg$|^png$\|^css$|^js$|...
□	Or	Request	Contains parameters	
□	Or	HTTP method	Does not match	(get|post)
□	And	URL	Is in target scope	

□ Automatically fix missing or superfluous new lines at end of request

☑ Automatically update Content-Length header when the request is edited

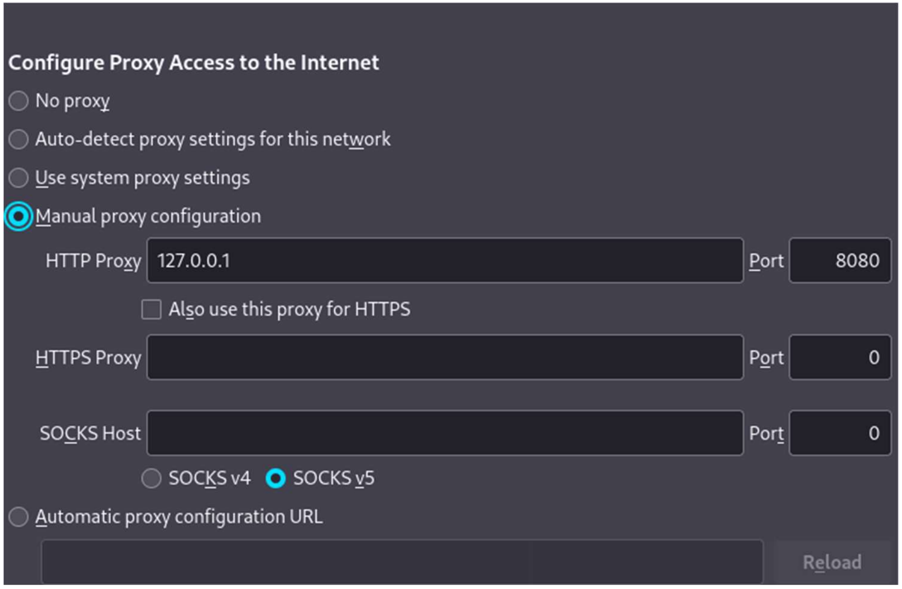
Intercept Server Responses

Use these settings to control which responses are stalled for viewing and editing in the Intercept tab.

□ Intercept responses based on the following rules:

Enabled	Operator	Match type	Relationship	Condition
☑		Content type he...	Matches	text
□	Or	Request	Was modified	
□	Or	Request	Was intercepted	

Then to configure on the browser we set the browser's proxy settings to route traffic through Burp Suite on the same port as set in Burp Proxy. Install the Burp Suite CA certificate in the browser to avoid SSL issues when intercepting HTTPS traffic.



Configure Proxy Access to the Internet

○ No proxy
○ Auto-detect proxy settings for this network
○ Use system proxy settings
● Manual proxy configuration

HTTP Proxy: 127.0.0.1 Port: 8080

□ Also use this proxy for HTTPS

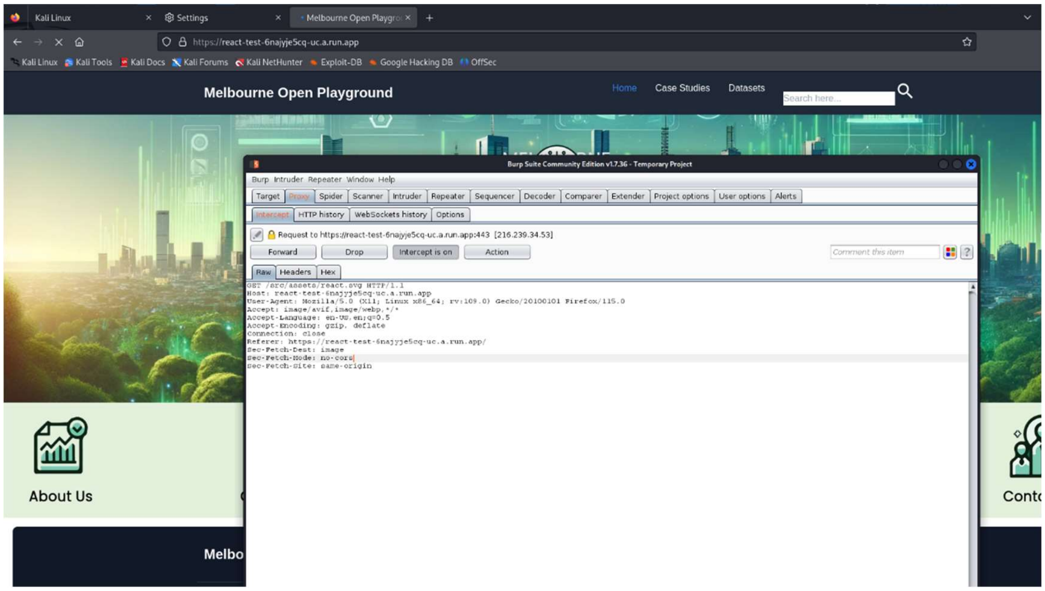
HTTPS Proxy: Port: 0

SOCKS Host: Port: 0

○ SOCKS v4 ● SOCKS v5

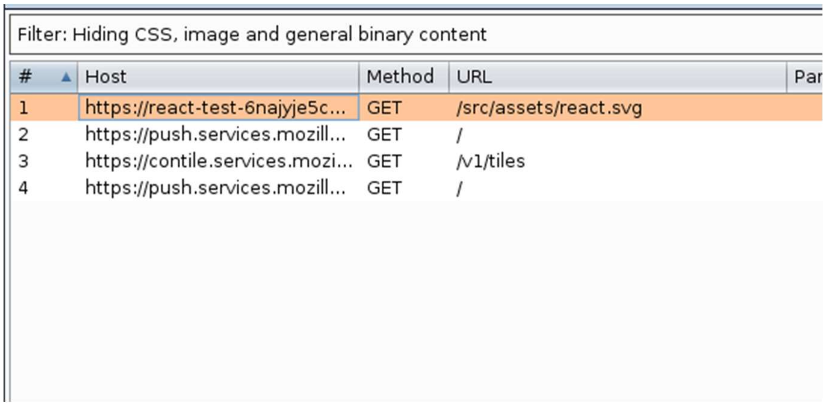
○ Automatic proxy configuration URL

With Burp Proxy running and the browser configured, navigate the target web application. Burp Proxy automatically captures all the incoming and outgoing HTTP/S requests and responses. Examine the details of each request and response in the Proxy's HTTP history tab to understand how the application communicates and to identify potential points of interest for testing.



Request to https://react-test-6najyje5cq-uc.a.run.app:443 [216.239.34.53]

```
GET /src/assets/react.svg HTTP/1.1
Host: react-test-6najyje5cq-uc.a.run.app
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: https://react-test-6najyje5cq-uc.a.run.app/
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
```

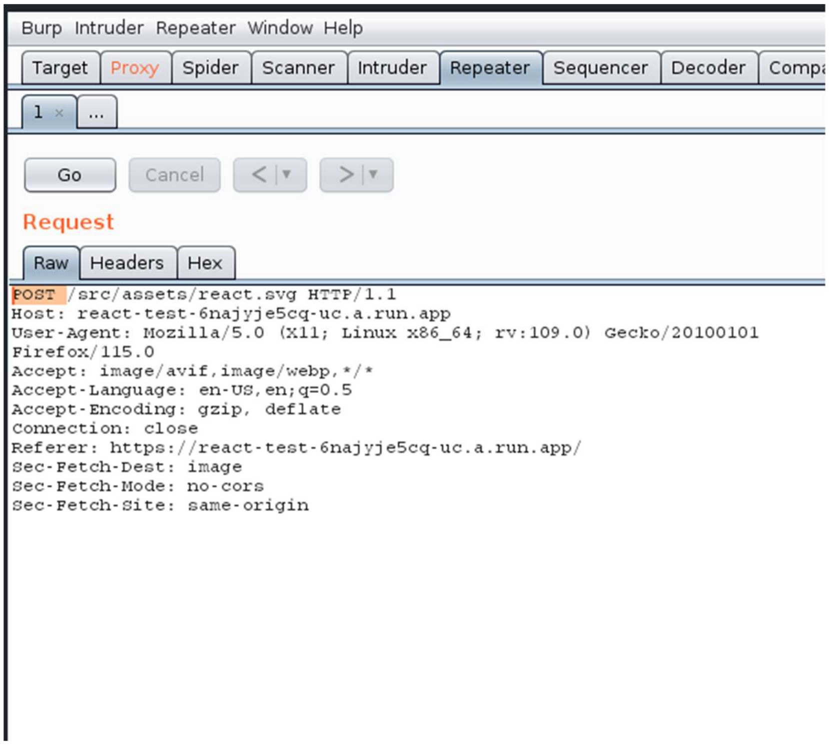


Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Par
1	https://react-test-6najyje5c...	GET	/src/assets/react.svg	
2	https://push.services.mozill...	GET	/	
3	https://contile.services.mozi...	GET	/v1/tiles	
4	https://push.services.mozill...	GET	/	

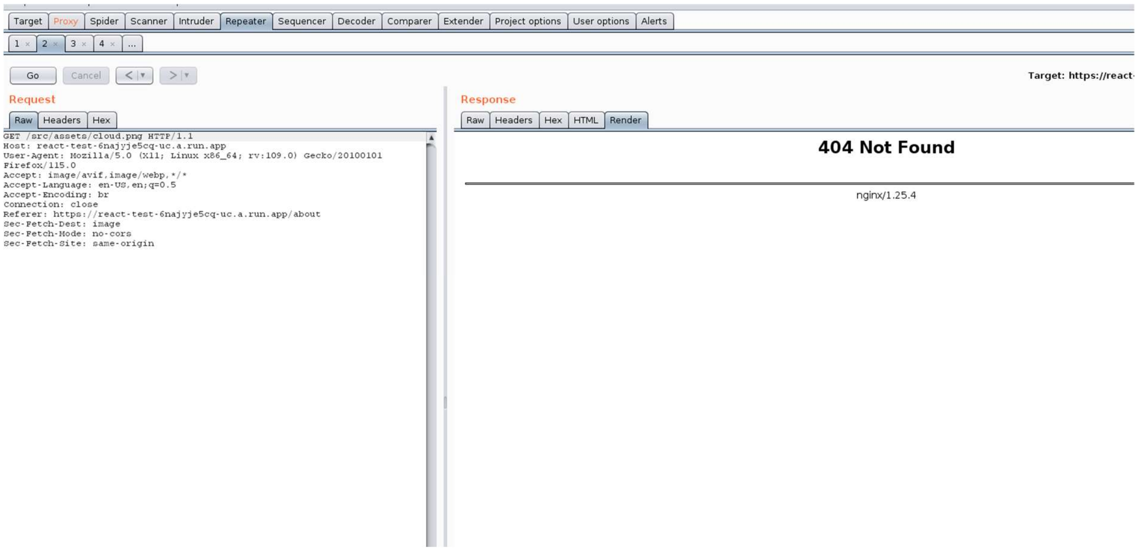
From the traffic captured by Burp Proxy, select specific requests that you want to test more deeply. Right-click on these requests and send them to Burp Repeater.

In Burp Repeater, manually modify the requests to test different vulnerabilities or behaviours.



```
POST /src/assets/react.svg HTTP/1.1
Host: react-test-6najyje5cq-uc.a.run.app
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: https://react-test-6najyje5cq-uc.a.run.app/
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
```

Looking at the results we got an error 404



Target: https://react-

Request

```
GET /src/assets/cloud.png HTTP/1.1
Host: react-test-6najyje5cq-uc.a.run.app
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
Firefox/115.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: br
Connection: close
Referer: https://react-test-6najyje5cq-uc.a.run.app/about
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
```

Response

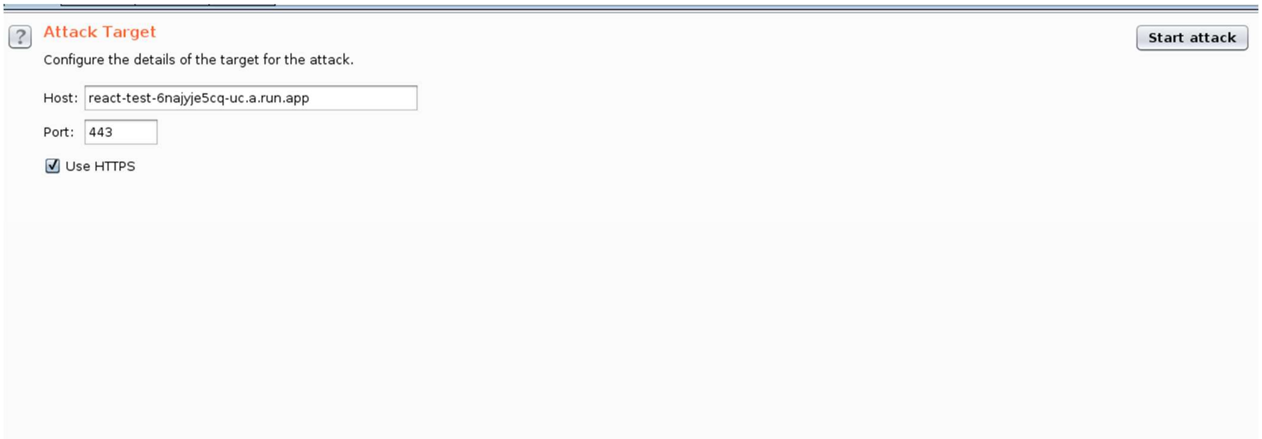
404 Not Found

nginx/1.25.4

This means that the website is stable and secure enough to not get breached by the repeater

Therefore, I tried to use Burp Intruder

Select a request from Burp Proxy or Repeater that involves parameters you wish to test more extensively. Send this request to Burp Intruder.



Attack Target

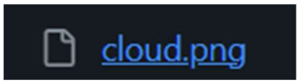
Configure the details of the target for the attack.

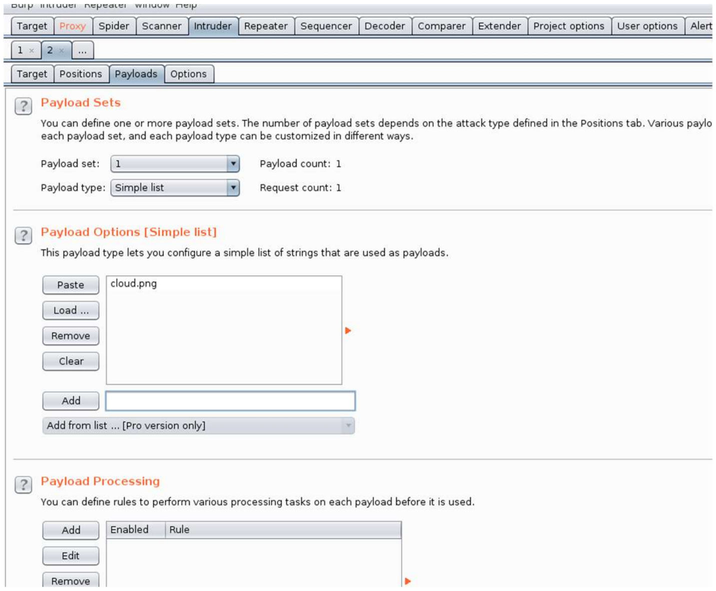
Host: react-test-6najyje5cq-uc.a.run.app

Port: 443

☑ Use HTTPS

I wanted to to directory traversal to another file in the repository called cloud.png




Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alert
1 × 2 × ...
Target Positions Payloads Options
Paste
Load ...
Remove
Clear
Add
Add from list ... [Pro version only]
Add
Edit
Remove

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various paylo
each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 1

Payload type: Simple list Request count: 1

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

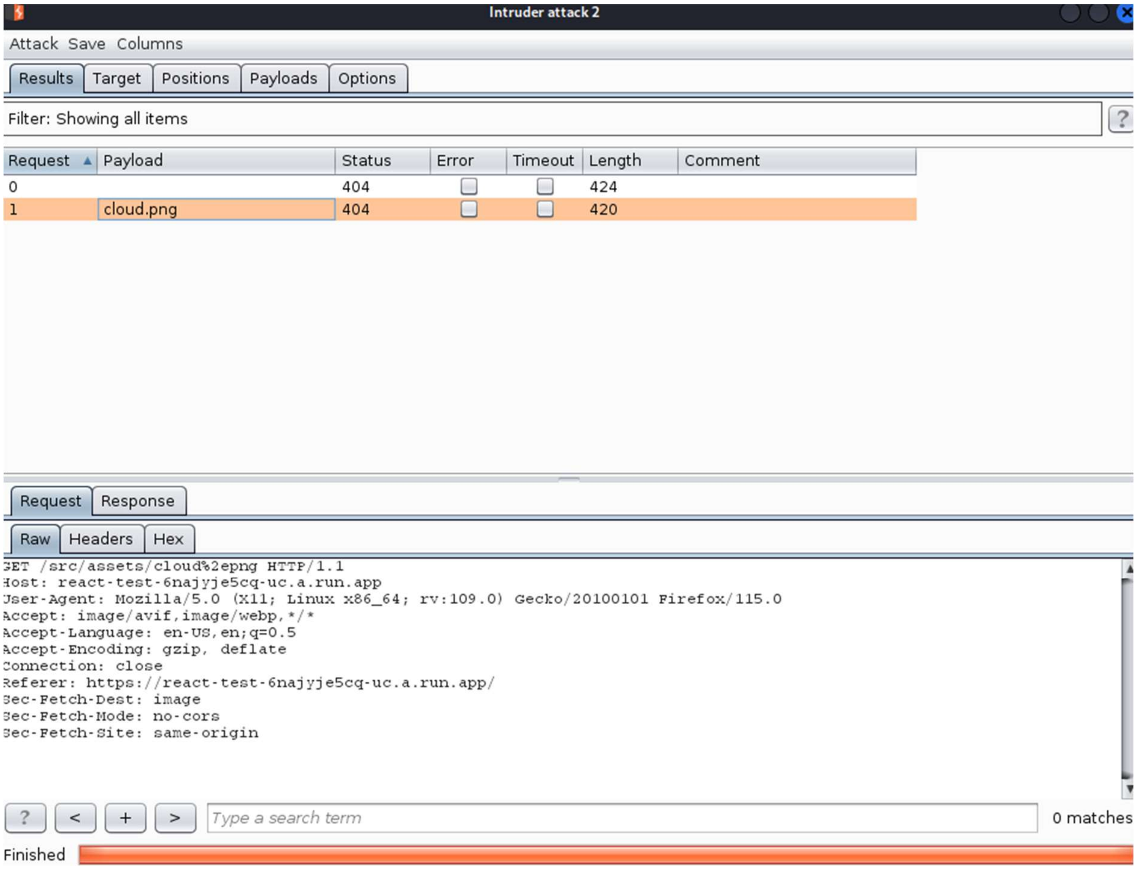
cloud.png

Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

Enabled	Rule

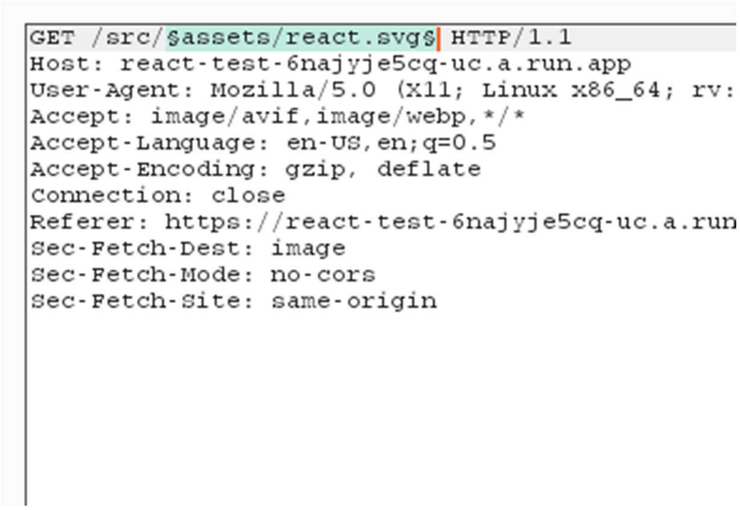
After setting the payloads and targets I ran the attack and it gave me status 404 which indicates that it failed to traverse directories



Filter: Showing all items

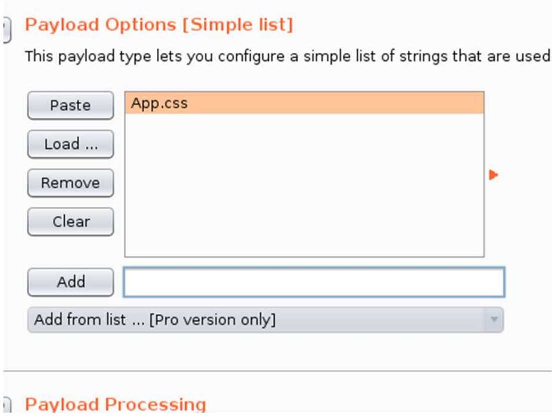
Request ▲	Payload	Status	Error	Timeout	Length	Comment
0		404	□	□	424	
1	cloud.png	404	□	□	420	

```
GET /src/assets/cloud%2epng HTTP/1.1
Host: react-test-6najyje5cq-uc.a.run.app
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: https://react-test-6najyje5cq-uc.a.run.app/
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
```



```
GET /src/$assets/react.svg$ HTTP/1.1
Host: react-test-6najyje5cq-uc.a.run.app
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: https://react-test-6najyje5cq-uc.a.run
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
```

Trying it again, I wanted to traverse to the directory in /src/App.css



Payload Options [Simple list]

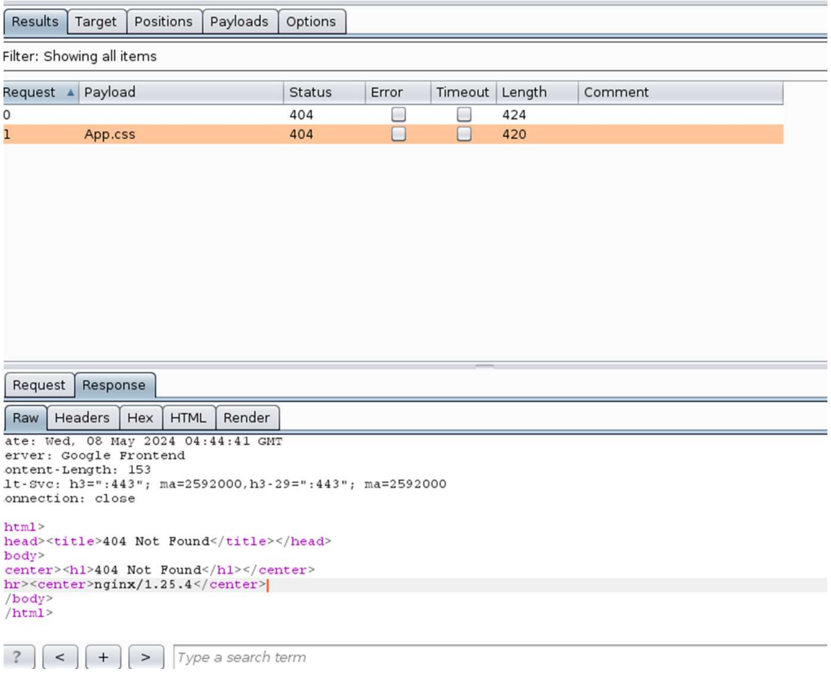
This payload type lets you configure a simple list of strings that are used

App.css

Add from list ... [Pro version only]

Payload Processing

Trying that attack it seemed like there was the same error which means that it was unable to traverse directories



Filter: Showing all items

Request ▲	Payload	Status	Error	Timeout	Length	Comment
0		404	□	□	424	
1	App.css	404	□	□	420	

```
ate: Wed, 08 May 2024 04:44:41 GMT
erver: Google Frontend
ontent-Length: 153
lt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000
onnection: close

html>
head><title>404 Not Found</title></head>
body>
center><h1>404 Not Found</h1></center>
hr><center>nginx/1.25.4</center>
/body>
/html>
```

Conclusion

The conducted penetration testing has provided valuable insights into the security posture of the web application. Key findings from the tests indicated that:

- The application handled directory traversal attempts securely, with no unauthorized access granted to files outside the intended directories, as indicated by consistent 404 Not Found responses to traversal payloads.
- Modifications to HTTP headers and methods did not result in any misbehaviour or information leakage, suggesting robust handling and validation of HTTP requests by the application.
- The server configurations regarding caching, and connection handling appeared to be secure, with no adverse behaviours observed during the tests.

Recommendations

- Continuously monitor and update the application's security configurations to safeguard against emerging threats and vulnerabilities.
- Implement regular security audits and penetration testing cycles to detect and rectify potential security issues proactively.
- Consider extending testing to include more complex attack scenarios and incorporating automated scanning tools to complement manual testing efforts.