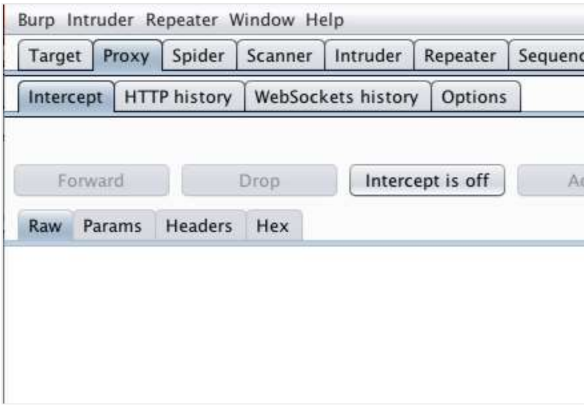
# Using Burp Scanner to Find Cross-Site Scripting (XSS) Issues

XSS vulnerabilities occur when an application includes attacker-controllable data in a response sent to the browser without properly validating or escapir content. Cross-site scripting attacks may occur anywhere that an application includes in responses data that originated from any untrusted source.

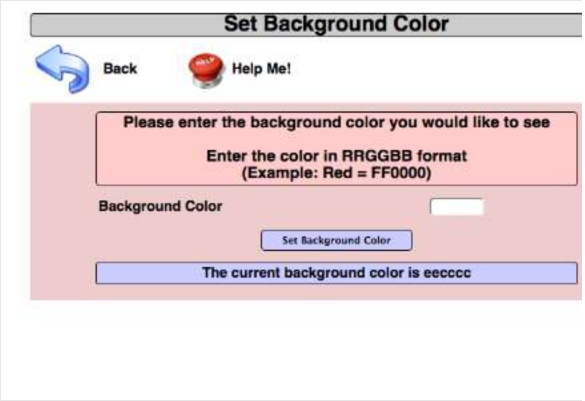
In this example we will demonstrate how to use Burp Scanner to test for XSS vulnerabilities. The example uses a version of "Mutillidae" taken from OWA Broken Web Application Project. Find out how to download, install and use this project.

First, ensure that Burp is correctly configured with your browser.

With Burp Proxy "Intercept" turned off, visit the web application you are testing in your browser.


Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequenc
Intercept HTTP history WebSockets history Options
Forward
Drop
Intercept is off
Ac
Raw Params Headers Hex

Visit the page of the website you wish to test for XSS vulnerabilities.



Please enter the background color you would like to see

Enter the color in RRGGBB format
(Example: Red = FF0000)

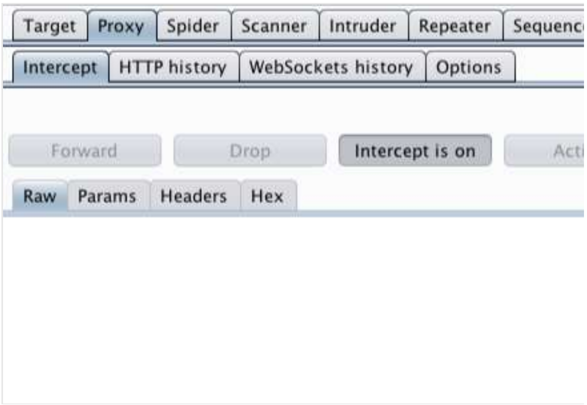
Background Color

Set Background Color

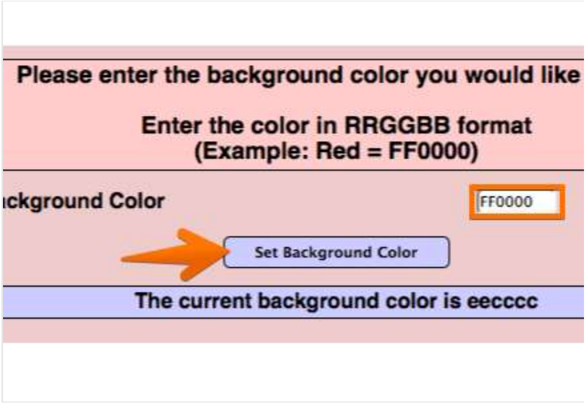
The current background color is eecccc

Return to Burp.

In the Proxy "Intercept" tab, ensure "Intercept is on".


Target Proxy Spider Scanner Intruder Repeater Sequenc
Intercept HTTP history WebSockets history Options
Forward
Drop
Intercept is on
Acti
Raw Params Headers Hex

Enter some appropriate input in to the web application and submit the request.



Please enter the background color you would like

Enter the color in RRGGBB format
(Example: Red = FF0000)

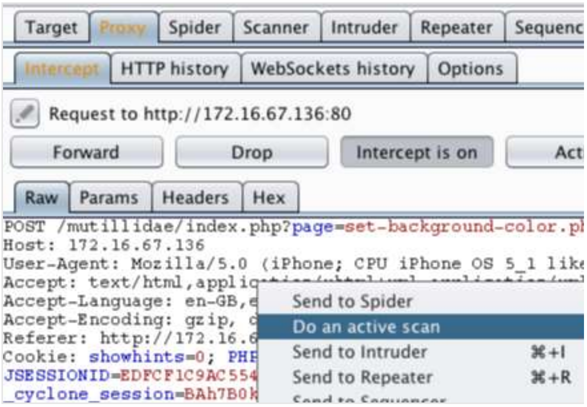
ickground Color FF0000

The current background color is eecccc

The request will be captured by Burp. You can view the HTTP request in the Proxy "Intercept" tab.

Right click on the request to bring up the context menu and click "Do an active scan" to send the request to Burp Scanner.

You can also locate the relevant request in various Burp tabs without having to use the intercept function, e.g. requests are logged and detailed in the "HTTP history" tab beneath the "Proxy" tab.



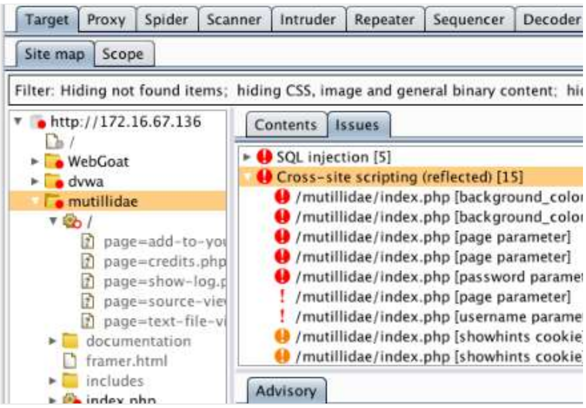
Request to http://172.16.67.136:80

```
POST /mutillidae/index.php?page=set-background-color.ph
Host: 172.16.67.136
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 5_1 like
Accept: text/html,applic
Accept-Language: en-GB,e
Accept-Encoding: gzip, d
Referer: http://172.16.6
Cookie: showhints=0; PHP
JSESSIONID=EDFCF1C9AC554
_cyclone_session=BAh7B0k
```

Once the scan is complete go to the Target "Site map" tab.

In this example the Scanner found a number of reflected XSS issues.

You can click on the arrow next to the issue to expand the section and view each individual issue.



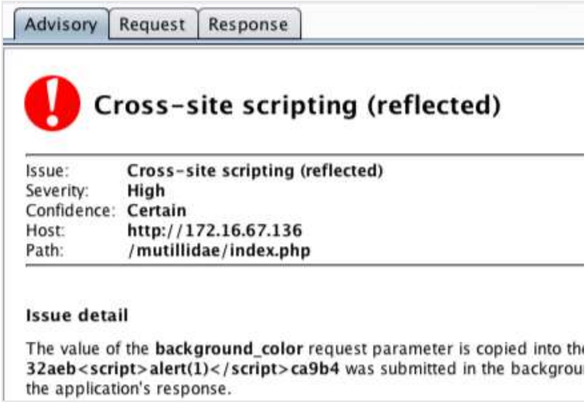
Filter: Hiding not found items; hiding CSS, image and general binary content; hi

```
▼ http://172.16.67.136
    /
  ▸ WebGoat
  ▸ dvwa
  ▾ mutillidae
    ▾ /
        page=add-to-yo
        page=credits.php
        page=show-log.p
        page=source-vie
        page=text-file-vi
    ▸ documentation
      framer.html
    ▸ includes
    ▸ index.php
```

```
▸ SQL injection [5]
▾ Cross-site scripting (reflected) [15]
    /mutillidae/index.php [background_colo
    /mutillidae/index.php [background_colo
    /mutillidae/index.php [page parameter]
    /mutillidae/index.php [page parameter]
    /mutillidae/index.php [password parame
    /mutillidae/index.php [page parameter]
    /mutillidae/index.php [username parame
    /mutillidae/index.php [showhints cookie
    /mutillidae/index.php [showhints cookie
```

After clicking on an individual issue the Scanner UI provides an advisory section regarding the specific issue.

You can also view the request and response from the simulated attack.



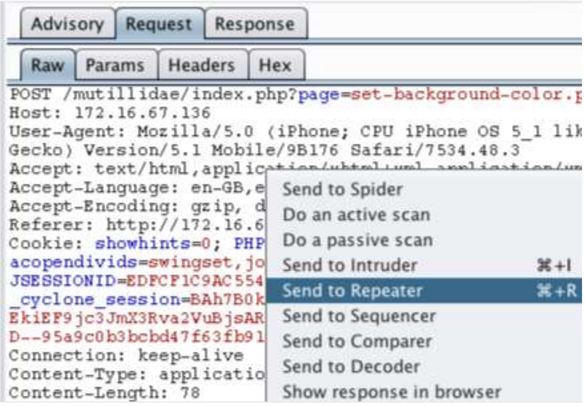
Cross-site scripting (reflected)

| | |
|---|---|
| Issue: | Cross-site scripting (reflected) |
| Severity: | High |
| Confidence: | Certain |
| Host: | http://172.16.67.136 |
| Path: | /mutillidae/index.php |

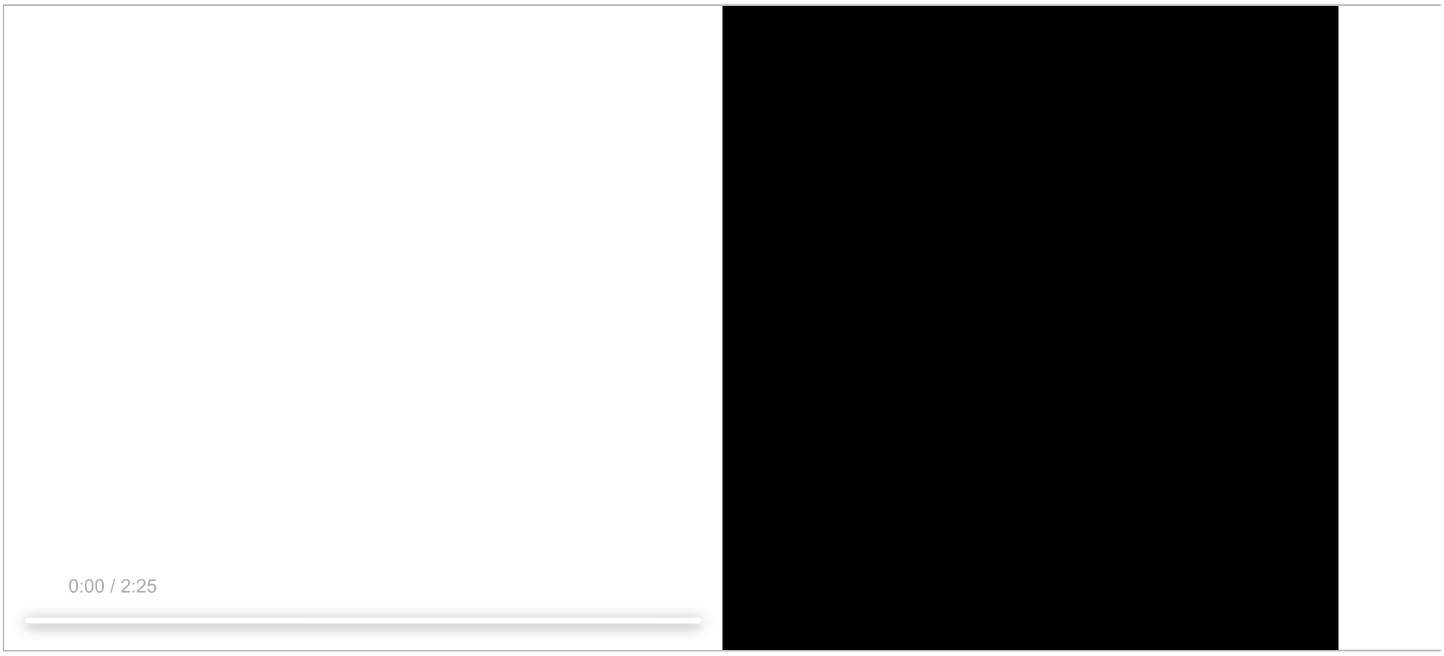
Issue detail

The value of the **background_color** request parameter is copied into th
**32aeb<script>alert(1)</script>ca9b4** was submitted in the backgrou
the application's response.

Furthermore, you can send the request to Burp Repeater for manual examination of the issue.



```
POST /mutillidae/index.php?page=set-background-color.p
Host: 172.16.67.136
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 5_1 lik
Gecko) Version/5.1 Mobile/9B176 Safari/7534.48.3
Accept: text/html,applic
Accept-Language: en-GB,e
Accept-Encoding: gzip, d
Referer: http://172.16.6
Cookie: showhints=0; PHP
acopendivids=swingset,jo
JSESSIONID=EDFCF1C9AC554
_cyclone_session=BAh7B0k
EkiEF9jc3JmX3Rva2VuBjsAR
D--95a9c0b3bcbd47f63fb91
Connection: keep-alive
Content-Type: applicatio
Content-Length: 78
```



Related articles:

- Getting started with Burp Proxy
- Using Burp Repeater
- Getting started with Burp Scanner

**Burp Suite**
Web vulnerability scanner
Burp Suite Editions
Release Notes

**Vulnerabilities**
Cross-site scripting (XSS)
SQL injection
Cross-site request forgery
XML external entity injection
Directory traversal
Server-side request forgery

**Customers**
Organizations
Testers
Developers

**Company**
About
PortSwigger News
Careers
Contact
Legal
Privacy Notice

**Insights**
Web Security Academy
Blog
Research
The Daily Swig

PortSwig

Follow us