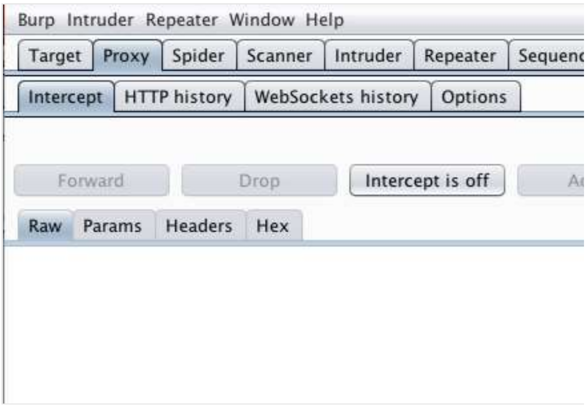
Using Burp to Hack Cookies and Manipulate Sessions

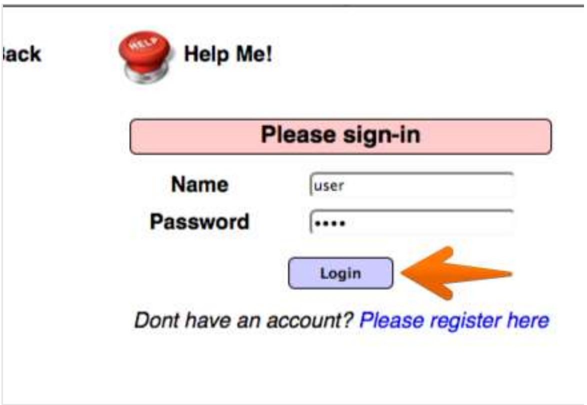
First, ensure that Burp is correctly configured with your browser.

With intercept turned off in the Proxy "Intercept" tab, visit the login page of the 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 A
Raw Params Headers Hex

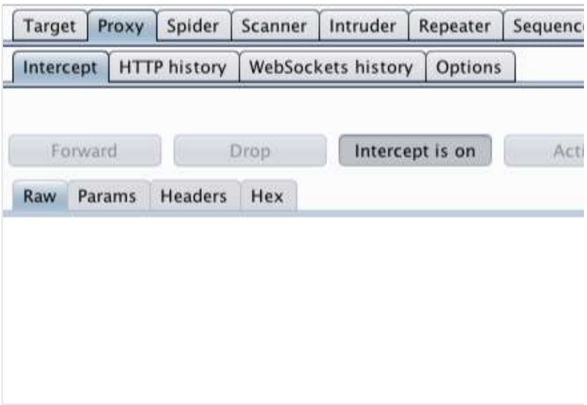
Log in to the application you are testing.

You can log in using the credentials user:user.



Return to Burp.

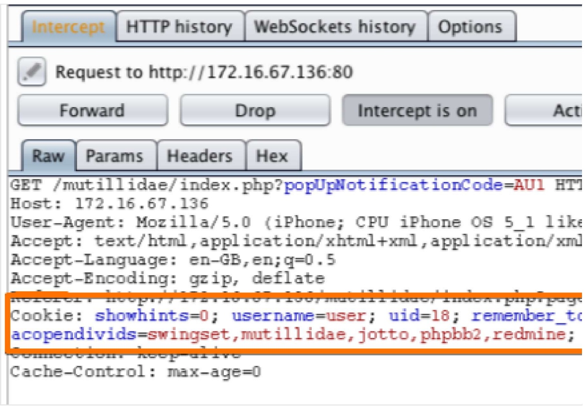
In the Proxy "Intercept" tab, ensure "Intercept is on".


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

Refresh the page in your browser.

The request will be captured by Burp, it can be viewed in the Proxy "Intercept" tab.

Cookies can be viewed in the cookie header.

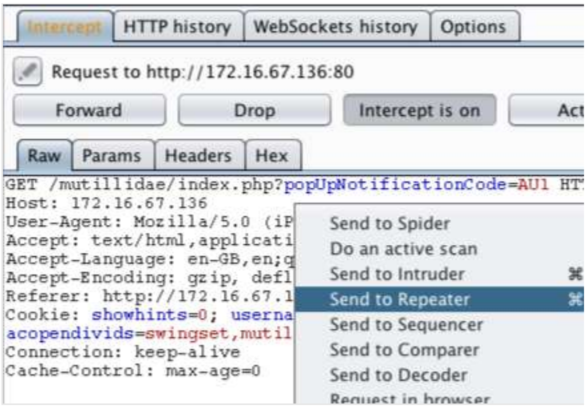


```
GET /mutillidae/index.php?popUpNotificationCode=AU1 HTT
Host: 172.16.67.136
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 5_1 like
Accept: text/html,application/xhtml+xml,application/xml
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://172.16.67.136/mutillidae/index.php?page
Cookie: showhints=0; username=user; uid=18; remember_to
acopendivids=swingset,mutillidae,jotto,phpbb2,redmine;
Connection: keep-alive
Cache-Control: max-age=0
```

We now need to investigate and edit each individual cookie.

Right click anywhere on the request and click "Send to Repeater".

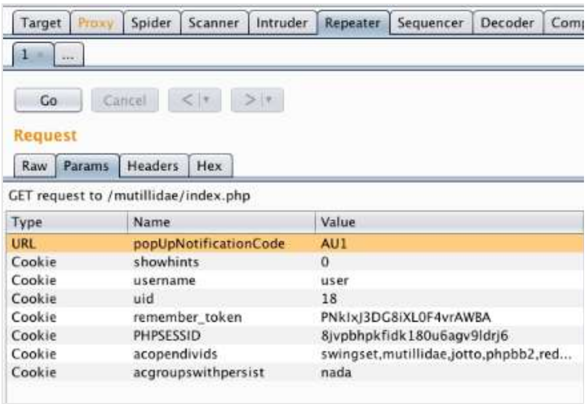
Note: You can also send requests to Repeater via the context menu in any location where HTTP requests are shown, such as the site map or Proxy history.



```
GET /mutillidae/index.php?popUpNotificationCode=AU1 HT
Host: 172.16.67.136
User-Agent: Mozilla/5.0 (iP
Accept: text/html,applicati
Accept-Language: en-GB,en;q
Accept-Encoding: gzip, defl
Referer: http://172.16.67.1
Cookie: showhints=0; userna
acopendivids=swingset,mutil
Connection: keep-alive
Cache-Control: max-age=0
```

Go to the Repeater tab.

The cookies in the request can be edited easily in the "Params" tab.



GET request to /mutillidae/index.php

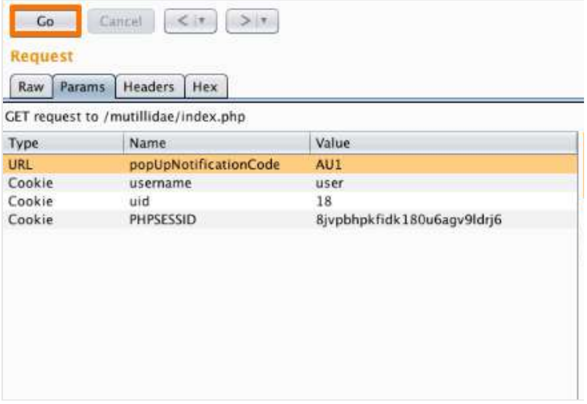
Type	Name	Value
URL	popUpNotificationCode	AU1
Cookie	showhints	0
Cookie	username	user
Cookie	uid	18
Cookie	remember_token	PNkIxJ3DG8iXL0F4vrAWBA
Cookie	PHPSESSID	8jvpbhpkfidk180u6agv9ldrj6
Cookie	acopendivids	swingset,mutillidae,jotto,phpbb2,red...
Cookie	acgroupswithpersist	nada

By removing cookies from the request we can ascertain the function of each cookie.

In this example, if the "username", "uid" and "PHPSESSID" cookies are removed, the session is ended and the user is logged out of the application.

We can use the Repeater to remove cookies and test the response from the server.

Remove and add cookies using the "Add" and "Remove" buttons and use the "Go" button to forward requests to the server.



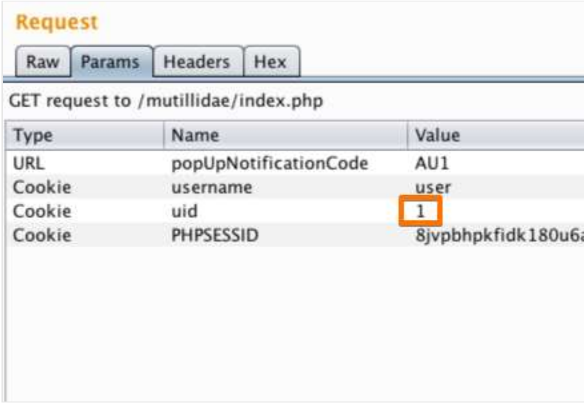
GET request to /mutillidae/index.php

Type	Name	Value
URL	popUpNotificationCode	AU1
Cookie	username	user
Cookie	uid	18
Cookie	PHPSESSID	8jvpbhpkfidk180u6agv9ldrj6

Cookies can be edited in the Request "Params" table.

In this example we have altered the value of the "uid" cookie to 1.

Alter the value then click the "Go" button.



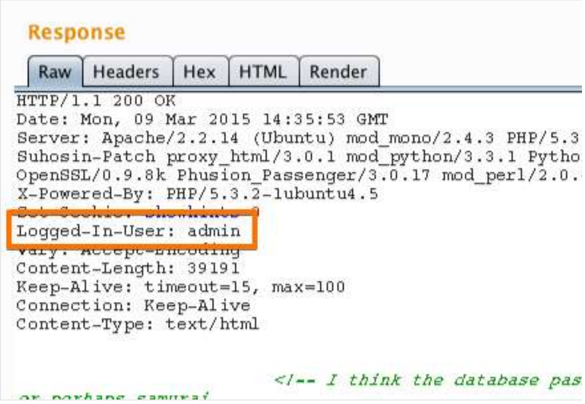
GET request to /mutillidae/index.php

Type	Name	Value
URL	popUpNotificationCode	AU1
Cookie	username	user
Cookie	uid	1
Cookie	PHPSESSID	8jvpbhpkfidk180u6a

The response from the server can be viewed in the "Response" panel in Repeater.

The response shows that by altering the "uid" cookie we have logged in to the application as "admin".

We have used cookies to manipulate the session and access another account with elevated privileges.



```
HTTP/1.1 200 OK
Date: Mon, 09 Mar 2015 14:35:53 GMT
Server: Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3
Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Pytho
OpenSSL/0.9.8k Phusion_Passenger/3.0.17 mod_perl/2.0.
X-Powered-By: PHP/5.3.2-1ubuntu4.5
Set-Cookie: showhints=0
Logged-In-User: admin
Vary: Accept-Encoding
Content-Length: 39191
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

                              <!-- I think the database pas
or perhaps samurai
```

Related articles:

- Getting started with Burp Proxy
- Using Burp Intruder
- Using Burp Repeater
- Getting started with Burp Scanner
- Using Burp to attack session management

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

PortSwigg

Follow us