Burp Suite - The Basics

What is Burp Suite

Burp Suite is a Java-based framework designed to serve as a comprehensive solution for conducting web application penetration testing

Burp Suite captures and enables manipulation of all the HTTP/HTTPS traffic between a browser and a web server.

Features of Burp Community

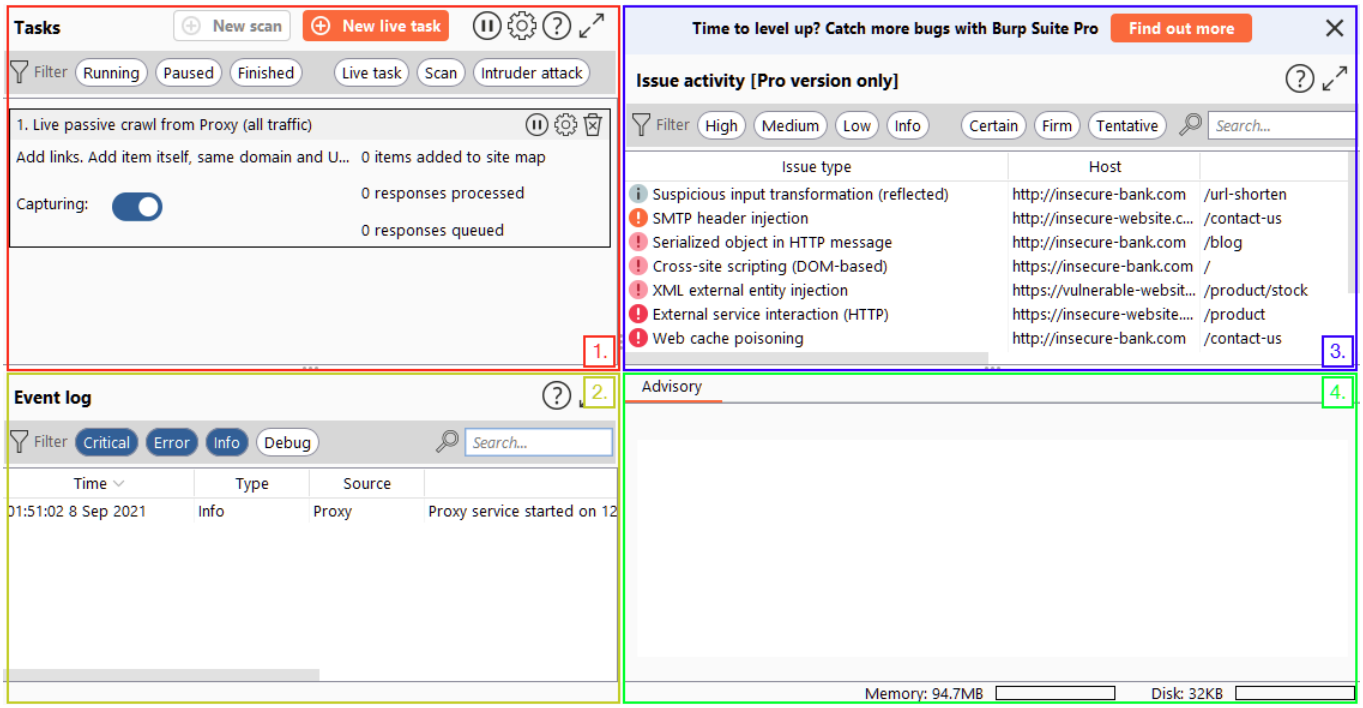
- **Proxy**: The Burp Proxy is the most renowned aspect of Burp Suite. It enables interception and modification of requests and responses while interacting with web applications.
- **Repeater**: Another well-known feature. Repeater allows for capturing, modifying, and resending the same request multiple times. This functionality is particularly useful when crafting payloads through trial and error (e.g., in SQLi - Structured Query Language Injection) or testing the functionality of an endpoint for vulnerabilities.
- **Intruder**: Despite rate limitations in Burp Suite Community, Intruder allows for spraying endpoints with requests. It is commonly utilized for brute-force attacks or fuzzing endpoints.
- **Decoder**: Decoder offers a valuable service for data transformation. It can decode captured information or encode payloads before sending them to the target. While alternative services exist for this purpose, leveraging Decoder within Burp Suite can be highly efficient.
- **Comparer**: As the name suggests, Comparer enables the comparison of two pieces of data at either the word or byte level. While not exclusive to Burp Suite, the ability to send potentially large data segments directly to a comparison tool with a single keyboard shortcut significantly accelerates the process.
- **Sequencer**: Sequencer is typically employed when assessing the randomness of tokens, such as session cookie values or other supposedly randomly generated data. If the algorithm used for generating these values lacks secure randomness, it can expose avenues for devastating attacks.

Installation

To download the latest version of Burp Suite for other systems, you may click this button to go to their download page.

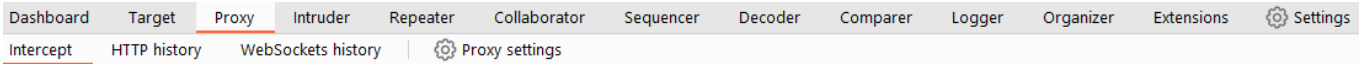
Linux, macOS, and Windows: For other operating systems, PortSwigger provides dedicated installers for Burp Suite Community

The Dashboard



1. **Tasks**: The Tasks menu allows you to define background tasks that Burp Suite will perform while you use the application. In Burp Suite Community, the default “Live Passive Crawl” task, which automatically logs the pages visited, is sufficient for our purposes in this module. Burp Suite Professional offers additional features like on-demand scans.
2. **Event log**: The Event log provides information about the actions performed by Burp Suite, such as starting the proxy, as well as details about connections made through Burp.
3. **Issue Activity**: This section is specific to Burp Suite Professional. It displays the vulnerabilities identified by the automated scanner, ranked by severity and filterable based on the certainty of the vulnerability.
4. **Advisory**: The Advisory section provides more detailed information about the identified vulnerabilities, including references and suggested remediations. This information can be exported into a report. In Burp Suite Community, this section may not show any vulnerabilities.

Navigation

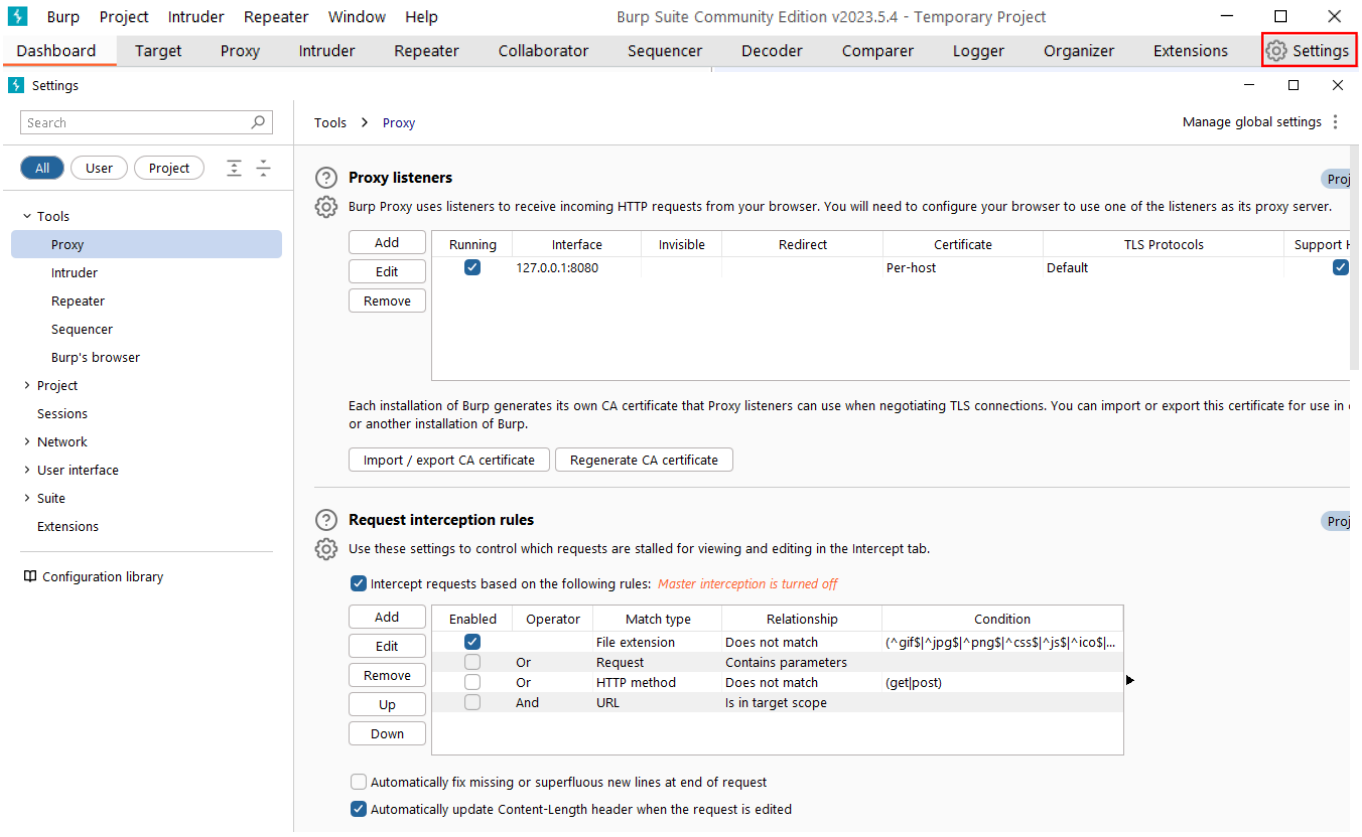


Burp Suite also provides keyboard shortcuts for quick navigation to key tabs. By default, the following shortcuts are available:

Shortcut	Tab
`Ctrl + Shift + D`	Dashboard
`Ctrl + Shift + T`	Target tab
`Ctrl + Shift + P`	Proxy tab
`Ctrl + Shift + I`	Intruder tab
`Ctrl + Shift + R`	Repeater tab

Options

- **Global Settings**: These settings affect the entire Burp Suite installation and are applied every time you start the application. They provide a baseline configuration for your Burp Suite environment.
- **Project Settings**: These settings are specific to the current project and apply only during the session. However, please note that Burp Suite Community Edition does not support saving projects, so any project-specific options will be lost when you close Burp.



In the Settings window, you will find a menu on the left-hand side. This menu allows you to switch between different types of settings, including:

1. **Search**: Enables searching for specific settings using keywords.
2. **Type filter**: Filters the settings for **User** and **Project** options.
   - **User settings**: Shows settings that affect the entire Burp Suite installation.
   - **Project settings**: Displays settings specific to the current project.
3. **Categories**: Allows selecting settings by category.

Introduction to the Burp Proxy

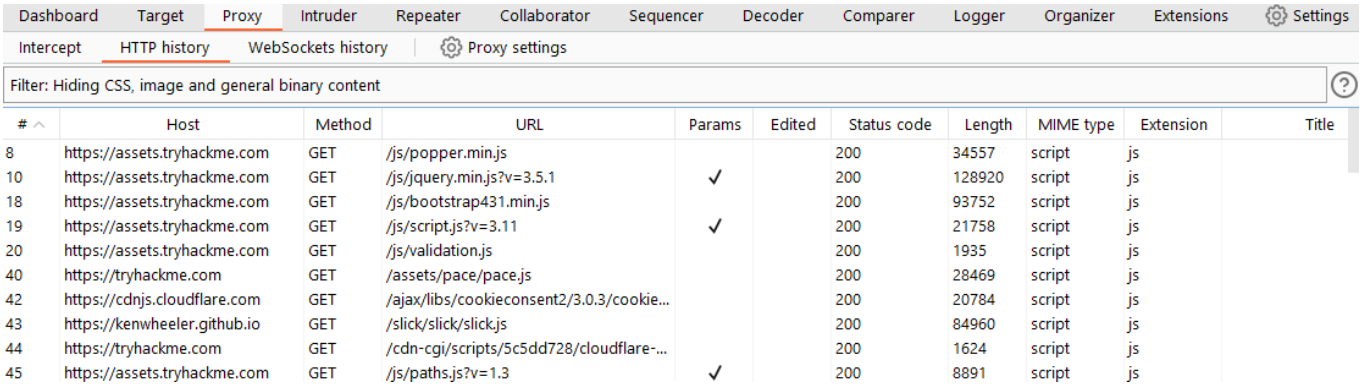
The Burp Proxy is a fundamental and crucial tool within Burp Suite. It enables the capture of requests and responses between the user and the target web server.

Key Points to Understand About the Burp Proxy

Intercepting Requests: When requests are made through the Burp Proxy, they are intercepted and held back from reaching the target server. The requests appear in the Proxy tab, allowing for further actions such as forwarding, dropping, editing, or sending them to other Burp modules. To disable the intercept and allow requests to pass through the proxy without interruption, click the `Intercept is on` button.

- **Taking Control:** The ability to intercept requests empowers testers to gain complete control over web traffic, making it invaluable for testing web applications.
- **Capture and Logging:** Burp Suite captures and logs requests made through the proxy by default, even when the interception is turned off. This logging functionality can be helpful for later analysis and review of prior requests.
- **WebSocket Support:** Burp Suite also captures and logs WebSocket communication, providing additional assistance when analysing web applications.

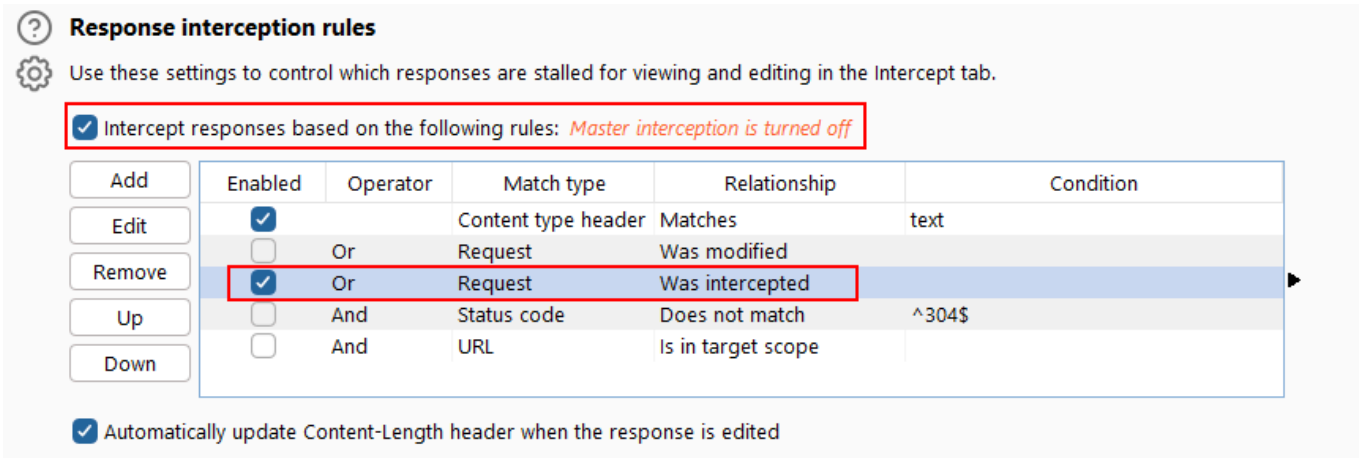
- **Logs and History:** The captured requests can be viewed in the **HTTP history** and **WebSockets history** sub-tabs, allowing for retrospective analysis and sending the requests to other Burp modules as needed.



#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title
8	https://assets.tryhackme.com	GET	/js/popper.min.js			200	34557	script	js	
10	https://assets.tryhackme.com	GET	/js/jquery.min.js?v=3.5.1	✓		200	128920	script	js	
18	https://assets.tryhackme.com	GET	/js/bootstrap431.min.js			200	93752	script	js	
19	https://assets.tryhackme.com	GET	/js/script.js?v=3.11	✓		200	21758	script	js	
20	https://assets.tryhackme.com	GET	/js/validation.js			200	1935	script	js	
40	https://tryhackme.com	GET	/assets/pace/pace.js			200	28469	script	js	
42	https://cdnjs.cloudflare.com	GET	/ajax/libs/cookieconsent2/3.0.3/cookie...			200	20784	script	js	
43	https://kenwheeler.github.io	GET	/slick/slick/slick.js			200	84960	script	js	
44	https://tryhackme.com	GET	/cdn-cgi/scripts/5c5dd728/cloudflare-...			200	1624	script	js	
45	https://assets.tryhackme.com	GET	/js/paths.js?v=1.3	✓		200	8891	script	js	

Some Notable Features in the Proxy Settings

Response Interception: By default, the proxy does not intercept server responses unless explicitly requested on a per-request basis. The "Intercept responses based on the following rules" checkbox, along with the defined rules, allows for a more flexible response interception.



- **Match and Replace:** The "Match and Replace" section in the **Proxy settings** enables the use of regular expressions (regex) to modify incoming and outgoing requests. This feature allows for dynamic changes, such as modifying the user agent or manipulating cookies.

Connecting through the Proxy (FoxyProxy)

Here are the steps to configure the Burp Suite Proxy with FoxyProxy:

1. **Install FoxyProxy:** Download and install the FoxyProxy Basic extension.

   Note: FoxyProxy is already installed on the AttackBox.
2. **Access FoxyProxy Options:** Once installed, a button will appear at the top right of the Firefox browser. Click on the FoxyProxy button to access the FoxyProxy options pop-up.

3. **Create Burp Proxy Configuration:** In the FoxyProxy options pop-up, click the **Options** button. This will open a new browser tab with the FoxyProxy configurations. Click the **Add** button to create a new proxy configuration.
4. **Add Proxy Details:** On the "Add Proxy" page, fill in the following values:
   - Title: `Burp` (or any preferred name)
   - Proxy IP: `127.0.0.1`
   - Port: `8080`
5. **Save Configuration:** Click **Save** to save the Burp Proxy configuration.
6. **Activate Proxy Configuration:** Click on the FoxyProxy icon at the top-right of the Firefox browser and select the `Burp` configuration. This will redirect your browser traffic through `127.0.0.1:8080`. Note that Burp Suite must be running for your browser to make requests when this configuration is activated.
7. **Enable Proxy Intercept in Burp Suite:** Switch to Burp Suite and ensure that Intercept is turned on in the **Proxy** tab.
8. **Test the Proxy:** Open Firefox and try accessing a website, such as the homepage for target IP e.g. `http://10.82.161.246/`. Your browser will hang, and the proxy will populate with the HTTP request. Congratulations, you have successfully intercepted your first request!

- When the proxy configuration is active, and the intercept is switched on in Burp Suite, your browser will hang whenever you make a request.
- Be cautious not to leave the intercept switched on unintentionally, as it can prevent your browser from making any requests.
- Right-clicking on a request in Burp Suite allows you to perform various actions, such as forwarding, dropping, sending to other tools, or selecting options from the right-click menu.

The **Target** tab in Burp Suite provides more than just control over the scope of our testing. It consists of three sub-tabs:

1. **Site map**: This sub-tab allows us to map out the web applications we are targeting in a tree structure. Every page that we visit while the proxy is active will be displayed on the site map. This feature enables us to automatically generate a site map by simply browsing the web application. In Burp Suite Professional, we can also use the site map to perform automated crawling of the target, exploring links between pages and mapping out as much of the site as possible. Even with Burp Suite Community, we can still utilize the site map to accumulate data during our initial enumeration steps. It is particularly useful for mapping out APIs, as any API endpoints accessed by the web application will be captured in the site map.
2. **Issue definitions**: Although Burp Community does not include the full vulnerability scanning functionality available in Burp Suite Professional, we still have access to a list of all the

vulnerabilities that the scanner looks for. The **Issue definitions** section provides an extensive list of web vulnerabilities, complete with descriptions and references. This resource can be valuable for referencing vulnerabilities in reports or assisting in describing a particular vulnerability that may have been identified during manual testing.

3. **Scope settings**: This setting allows us to control the target scope in Burp Suite. It enables us to include or exclude specific domains/IPs to define the scope of our testing. By managing the scope, we can focus on the web applications we are specifically targeting and avoid capturing unnecessary traffic.

The Burp Suite Browser

To start the Burp Browser, click the `Open Browser` button in the proxy tab. A Chromium window will pop up, and any requests made in this browser will go through the proxy.

There are two simple solutions to this:

1. **Smart option:** Create a new user and run Burp Suite under a low-privilege account to allow the Burp Browser to run without issues.
2. **Easy option:** Go to `Settings -> Tools -> Burp's browser` and check the `Allow Burp's browser to run without a sandbox` option. Enabling this option will allow the browser to start without a sandbox. However, please be aware that this option is disabled by default for security reasons. If you choose to enable it, exercise caution, as compromising the browser could grant an attacker access to your entire machine. In the training environment of the AttackBox, this is unlikely to be a significant issue, but use it responsibly.

Scoping and Targeting

Capturing and logging all of the traffic can quickly become overwhelming and inconvenient, especially when we only want to focus on specific web applications. This is where scoping comes in.

By setting a scope for the project, we can define what gets proxied and logged in Burp Suite. We can restrict Burp Suite to target only the specific web application(s) we want to test

However, even if we disabled logging for out-of-scope traffic, the proxy will still intercept everything. To prevent this, we need to go to the **Proxy settings** sub-tab and select `And URL Is in target scope` from the "Intercept Client Requests" section.

Example Attack

In a real-world web app pentest, we would test this for a variety of things, one of which would be Cross-Site Scripting (or XSS)

client-side filters are absurdly easy to bypass. There are a variety of ways we could disable the script or just prevent it from loading in the first place.

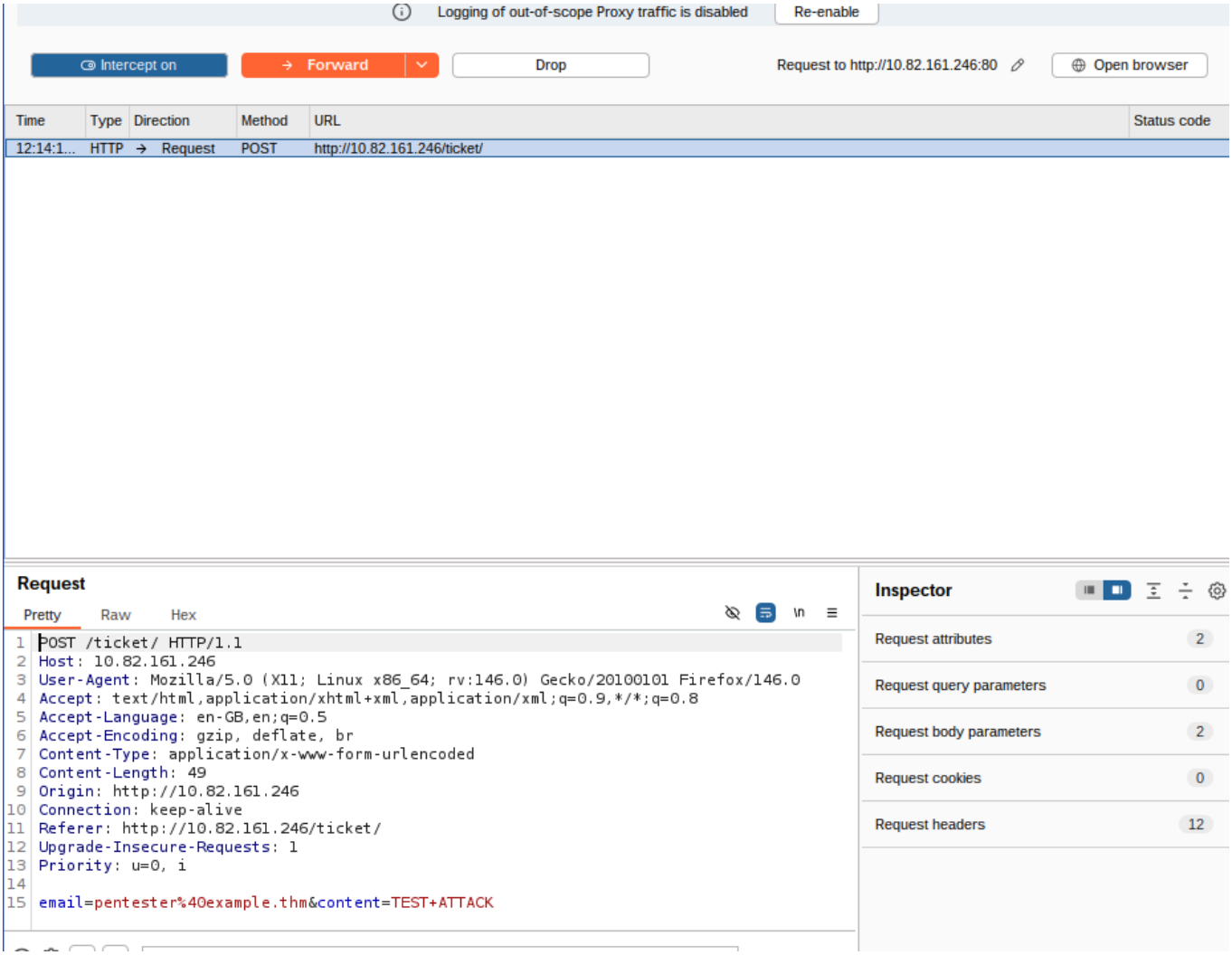
make sure that your Burp Proxy is active and that intercept is on.

Now, enter some legitimate data into the support form. For example: "pentester@example.thm" as an email address, and "Test Attack" as a query.

Submit the form — the request should be intercepted by the proxy.

With the request captured in the proxy, we can now change the email field to be our very simple payload from above: `<script>alert("Succ3ssful XSS")</script>`. After pasting in the payload, we need to select it, then URL encode it with the `Ctrl + U` shortcut to make it safe to send

first we turn on the proxy then make a "legitimate request" the proxy will capture the request so we can edit it



I then edited the request and injected the script into the request i then can forward the request

Request

Pretty Raw Hex

```
1  POST /ticket/ HTTP/1.1
2  Host: 10.82.161.246
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:146.0) Gecko/20100101 Firefox/146.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5  Accept-Language: en-GB,en;q=0.5
6  Accept-Encoding: gzip, deflate, br
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 49
9  Origin: http://10.82.161.246
10 Connection: keep-alive
11 Referer: http://10.82.161.246/ticket/
12 Upgrade-Insecure-Requests: 1
13 Priority: u=0, i
14
15 email=<script>alert("Succ3ssful+XSS")</script>&content=TEST+ATTACK
```