Burp Suite Pocket Guide

Table of Contents

1. Burp Suite Overview

1.1. What is Burp Suite?

Burp Suite is a comprehensive platform for performing security testing of web applications. It includes a variety of tools with unique functionalities that work together seamlessly to support the entire testing process, from initial mapping and analysis of an application's attack surface to finding and exploiting security vulnerabilities.

1.2. Editions of Burp Suite

Community Edition: This is the free version of Burp Suite. It provides essential manual tools, such as the Proxy, Intruder, Repeater, and Decoder. However, it lacks the automated scanning and some advanced features found in the Professional and Enterprise editions.

Professional Edition: A paid version that offers additional features, including the automated scanner, advanced manual tools, and support for extensions. This edition is aimed at security professionals who need to perform more comprehensive and efficient testing.

Enterprise Edition: This edition is designed for organizations needing to scale their security testing across many applications. It offers continuous, automated scanning and integrates with CI/CD pipelines, making it suitable for large enterprises.

1.3. Use Cases

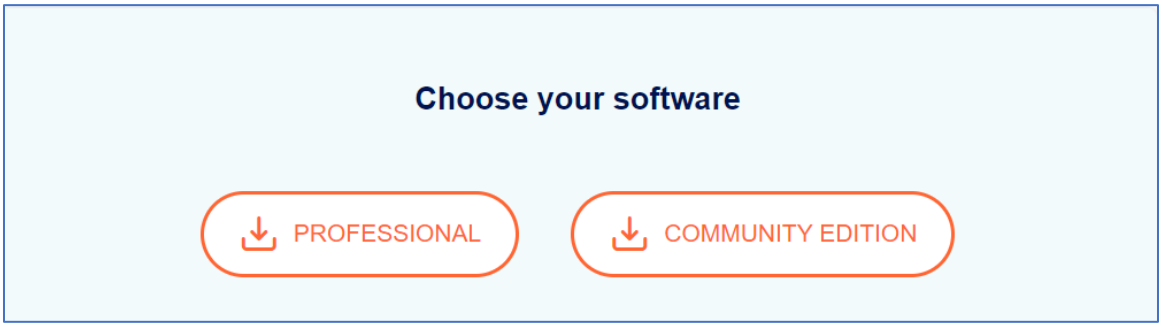
Penetration Testing: Burp Suite is widely used by penetration testers to identify and exploit vulnerabilities in web applications.

Security Audits: Organizations use Burp Suite for regular security audits to ensure their applications remain secure over time.

Bug Bounty Hunting: Security researchers and bug bounty hunters use Burp Suite to find and report vulnerabilities in web applications for rewards.

1.4. Download and Installation Steps

First of all, Visit the official Burp Suite website. Download the appropriate version (Community or Professional).



Run the installer and launch Burp Suite. When asked to select a project file and configuration, just click **Next** and then Start Burp to skip this for now.

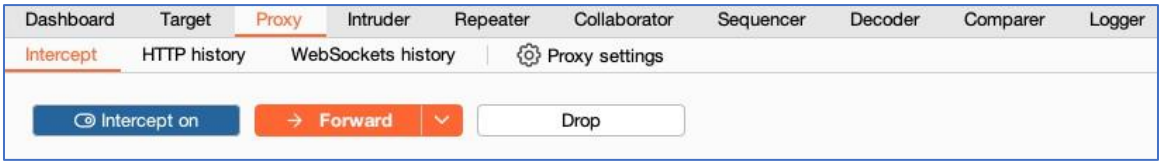
2. Intercept HTTP traffic with Burp Proxy

Burp Proxy lets you intercept HTTP requests and responses sent between Burp's browser and the target server. This enables you to study how the website behaves when you perform different actions.

2.1. Launch Burp's browser

Go to the Proxy > Intercept tab.

Set the intercept toggle to Intercept on.

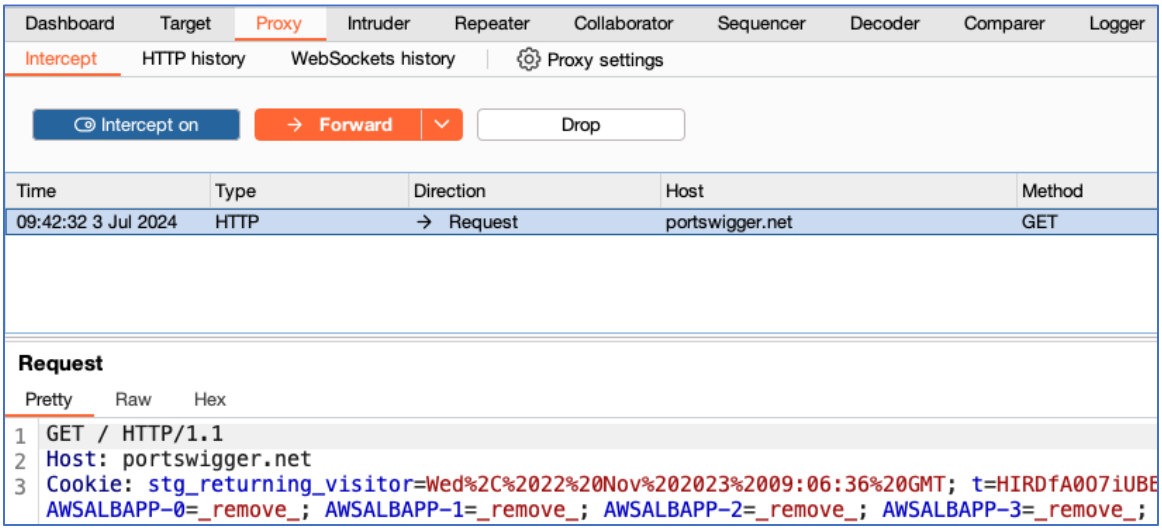


Click Open Browser. This launches Burp's browser, which is preconfigured to work with Burp right out of the box.

Position the windows so that you can see both Burp and Burp's browser.

2.2. Intercept a request

Using Burp's browser, try to visit **https://portswigger.net** and observe that the site doesn't load. Burp Proxy has intercepted the HTTP request that was issued by the browser before it could reach the server. You can see this intercepted request on the Proxy > Intercept tab.



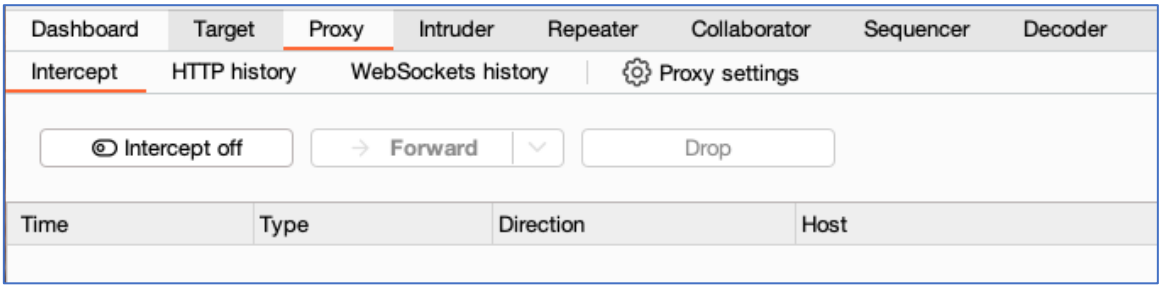
The request is held here so that you can study it, and even modify it, before forwarding it to the target server.

2.3. Forward the request

Click the Forward button to send the intercepted request. Click Forward again to send any subsequent requests that are intercepted, until the page loads in Burp's browser. The Forward button sends all the selected requests.

2.4. Switch off interception

Due to the number of requests browsers typically send, you often won't want to intercept every single one of them. Set the intercept toggle to Intercept off.

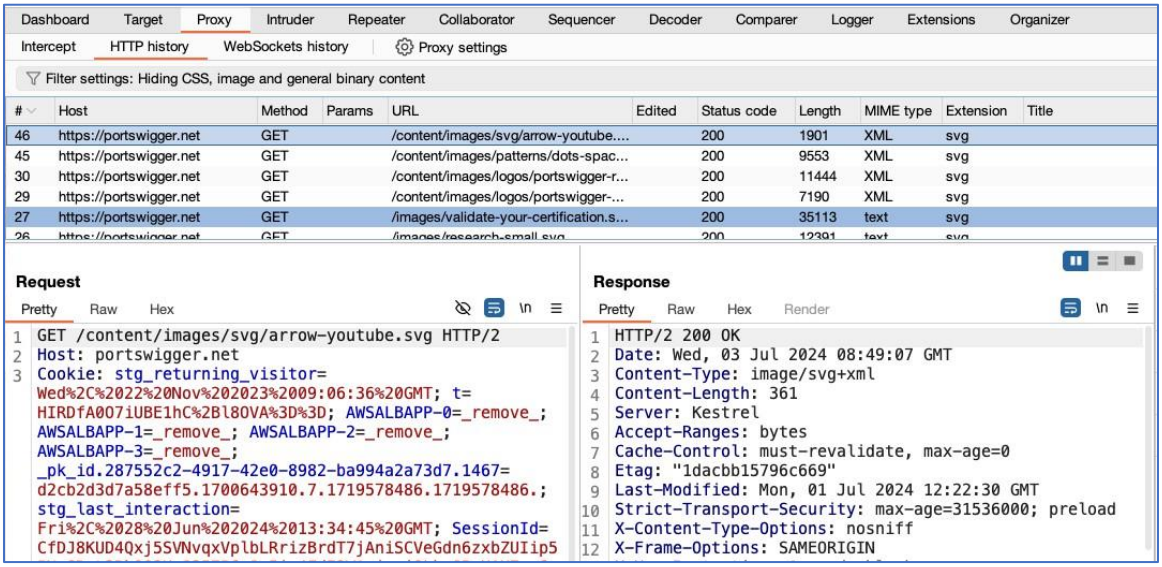


Go back to the browser and confirm that you can now interact with the site as normal.

2.5. View the HTTP history

In Burp, go to the Proxy > HTTP history tab. Here, you can see the history of all HTTP traffic that has passed through Burp Proxy, even while intercept was switched off.

Click on any entry in the history to view the raw HTTP request, along with the corresponding response from the server.



This lets you explore the website as normal and study the interactions between Burp's browser and the server afterward, which is more convenient in many cases.

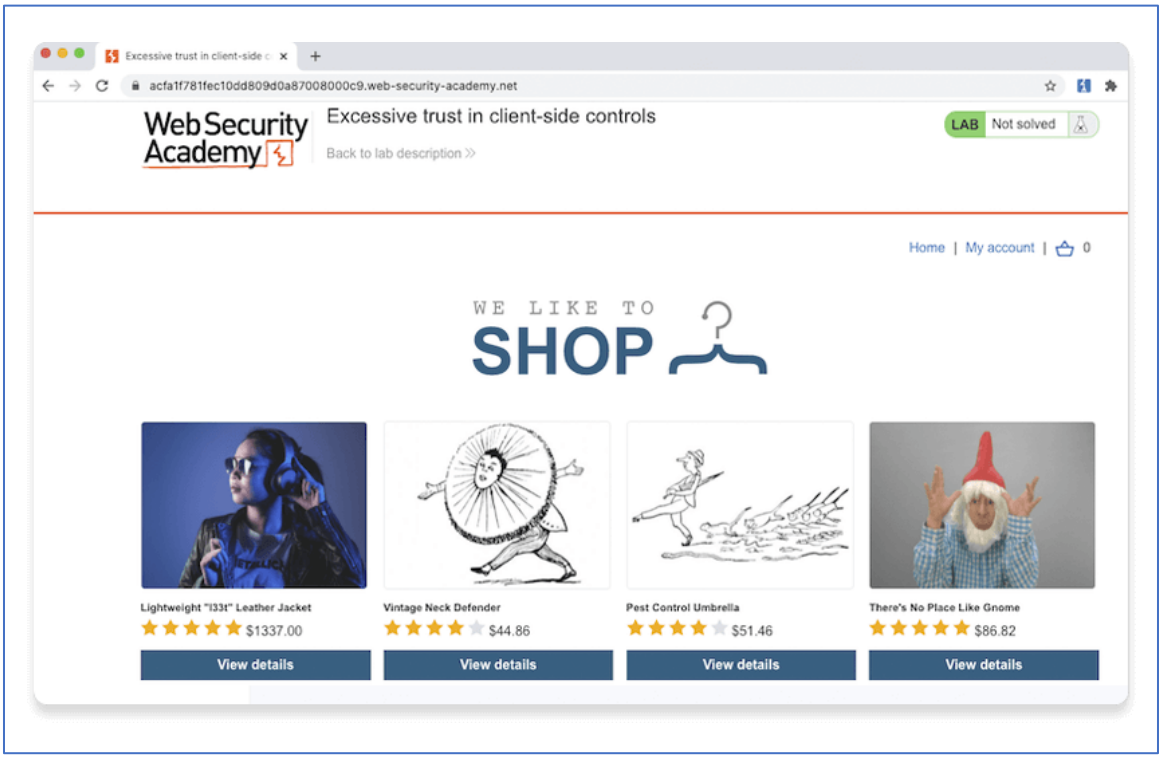
3. Modifying HTTP requests with Burp Proxy

This enables you to manipulate these requests in ways that the website isn't expecting, in order to see how it responds. Using one of our deliberately vulnerable websites, known as "labs", you'll see how this can help you identify and exploit real vulnerabilities.

3.1 Access the vulnerable website in Burp's browser

In Burp, go to the Proxy > Intercept tab and make sure interception is switched off. Launch Burp's browser and use it to visit the following URL: https://portswigger.net/web-security/logic-flaws/examples/lab-logic-flaws-excessive-trust-in-client-side-controls

When the page loads, click Access the lab. If prompted, log in to your portswigger.net account. After a few seconds, you will see your own instance of a fake shopping website.



3.2 Log in to your shopping account

On the shopping website, click My account and log in using the following credentials:

Username: **wiener**

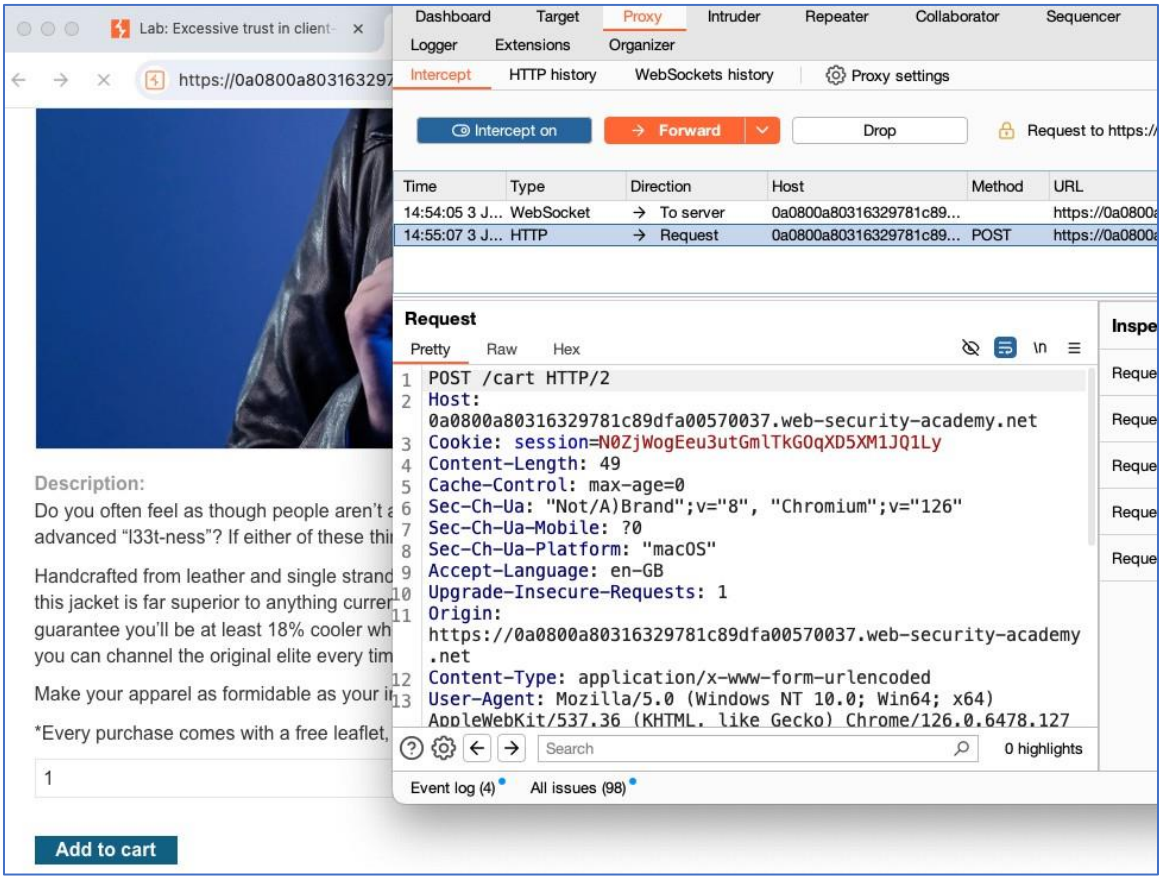
Password: **peter**

3.3 Find something to buy

Click Home to go back to the home page. Select the option to view the product details for the Lightweight "l33t" leather jacket.

3.4 Study the add to cart function

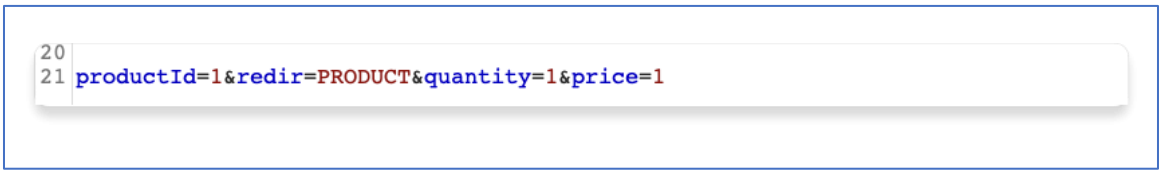
In Burp, go to the Proxy > Intercept tab and switch interception on. In the browser, add the leather jacket to your cart to intercept the resulting POST /cart request.



Study the intercepted request and notice that there is a parameter in the body called price, which matches the price of the item in cents.

3.5 Modify the request

Change the value of the price parameter to 1 and click Forward > Forward all to send the modified request to the server, along with any other intercepted requests.



Switch interception off again so that any subsequent requests can pass through Burp Proxy uninterrupted.

3.6 Exploit the vulnerability

In Burp's browser, click the basket icon in the upper-right corner to view your cart. Notice that the jacket has been added for just one cent.

Click the **Place order** button to purchase the jacket for an extremely reasonable price.

4 Set the target scope

The target scope tells Burp exactly which URLs and hosts you want to test. This enables you to filter out the noise generated by your browser and other sites, so you can focus on the traffic that you're interested in.

4.1 Launch Burp's browser

Launch Burp's browser and use it to visit the following URL: https://portswigger.net/web-security/information-disclosure/exploiting/lab-infoleak-in-error-messages

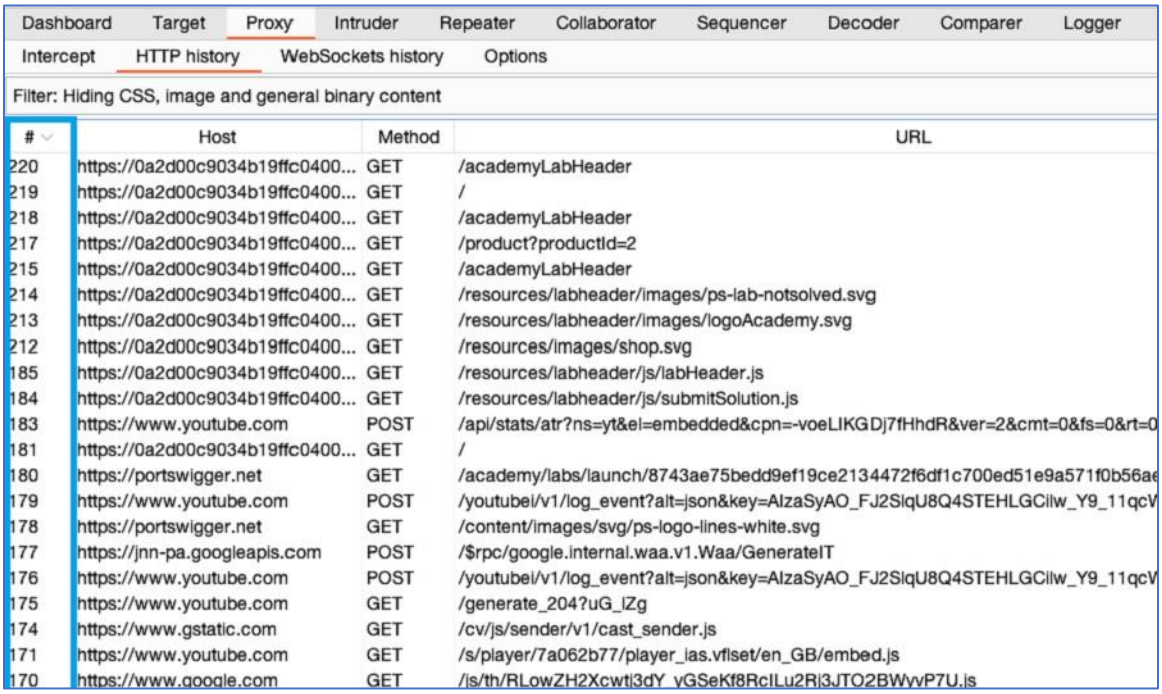
When the page loads, click Access the lab. If prompted, log in to your portswigger.net account. After a few seconds, you will see your own instance of a fake shopping website.

4.2 Browse the target site

In the browser, explore the site by clicking on a couple of the product pages.

4.3 Study the HTTP history

In Burp, go to the Proxy > HTTP history tab. To make this easier to read, keep clicking the header of the leftmost column (#) until the requests are sorted in descending order. This way, you can see the most recent requests at the top.

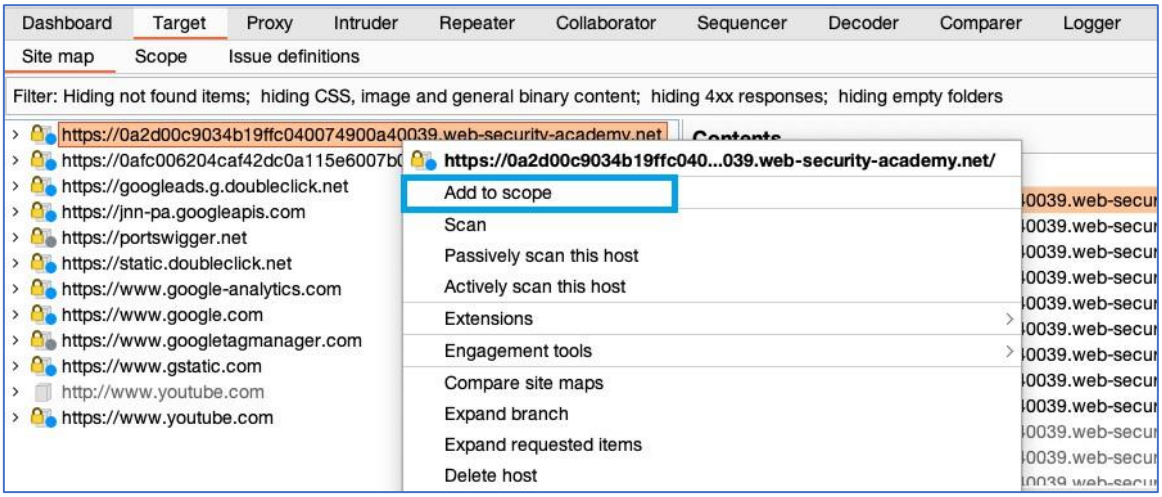


Notice that the HTTP history shows details about each request that the browser has made, including requests to third-party websites that you're not interested in, such as YouTube and Google Analytics.

4.4 Set the target scope

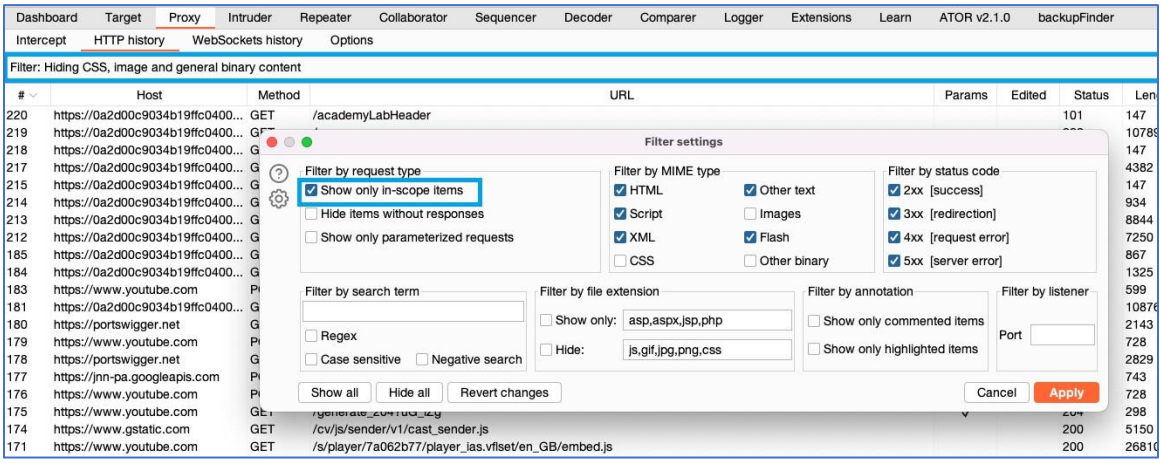
Go to Target > Site map. In the left-hand panel you can see a list of hosts that your browser has interacted with.

Right-click on the node for the target site and click Add to scope. When prompted in a pop-up window, click Yes to exclude out-of-scope traffic.



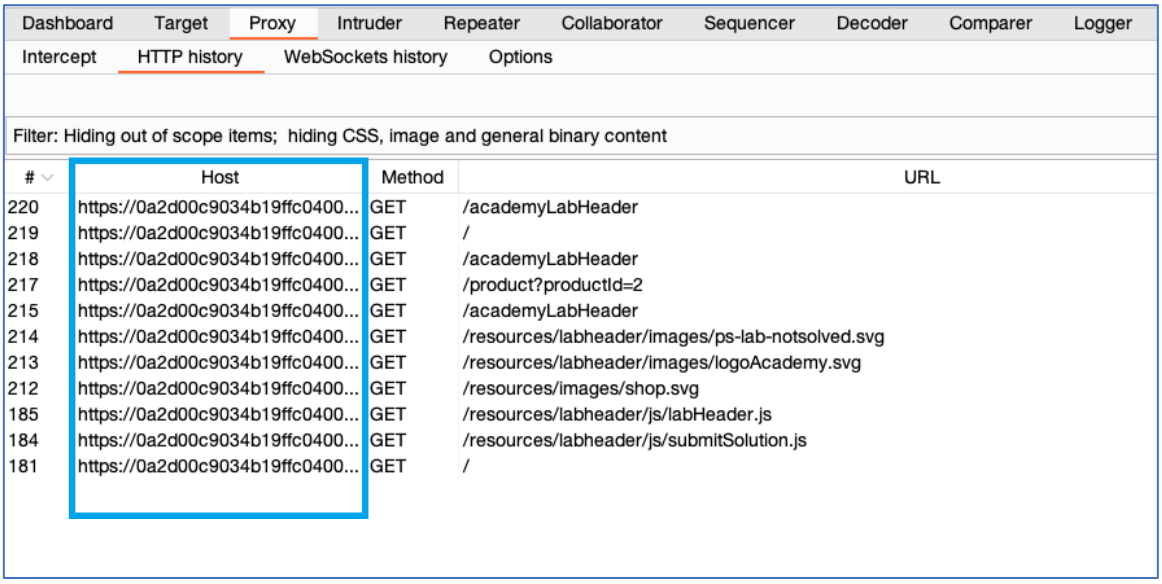
4.5 Filter HTTP history

Click on the display filter above the HTTP history and select Show only in-scope items.



Scroll back through your HTTP history. Notice that it now only shows entries from the target website. All other entries have been hidden.

This greatly simplifies the history to only include items you're interested in. If you continue to browse the target site, notice that out-of-scope traffic is no longer logged in the site map or proxy history.



5 Reissue requests with Burp Repeater

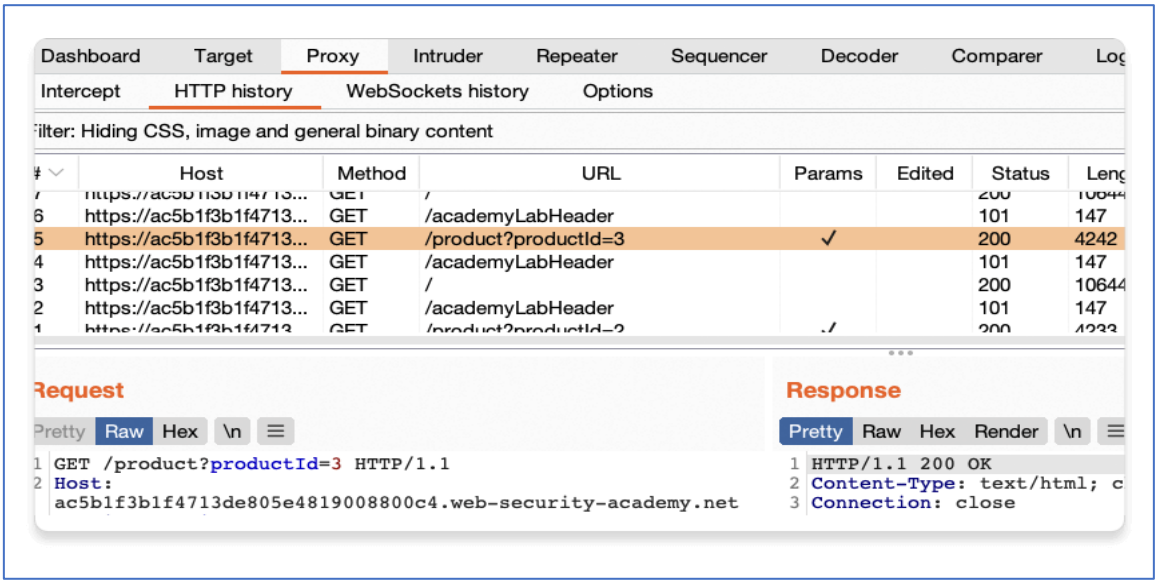
This lets you study the target website's response to different input without having to intercept the request each time. This makes it much simpler to probe for vulnerabilities, or confirm ones that were identified by Burp Scanner.

5.1 Sending a request to Burp Repeater

The most common way of using Burp Repeater is to send it a request from another of Burp's tools. In this example, we'll send a request from the HTTP history in Burp Proxy.

5.2 Identify an interesting request

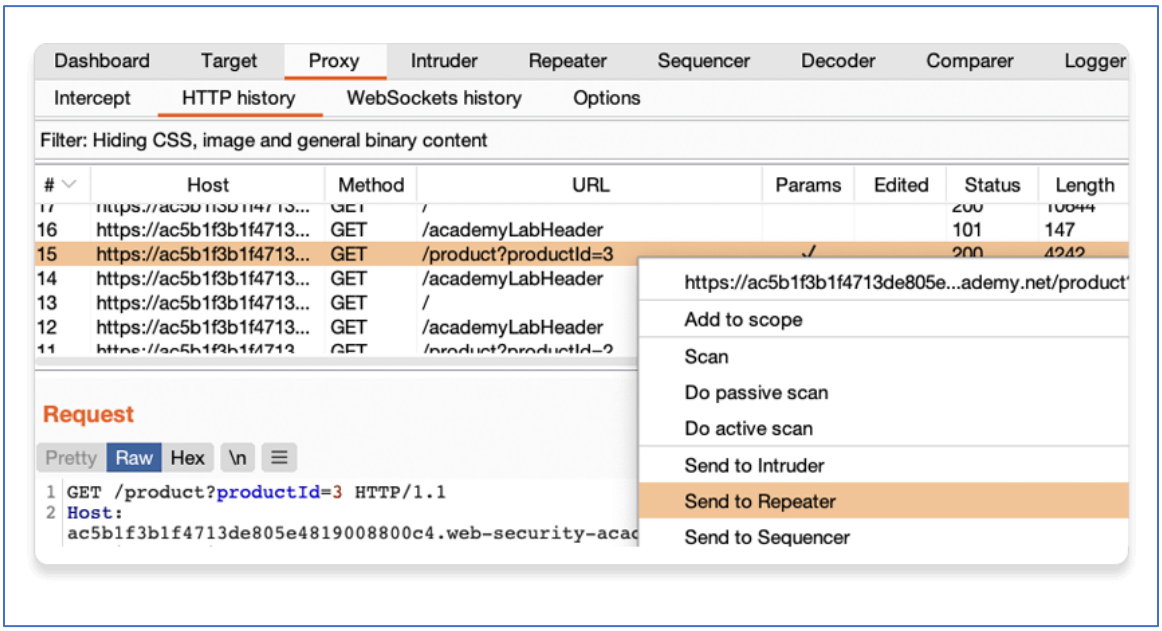
In the previous tutorial, you browsed a fake shopping website. Notice that each time you accessed a product page, the browser sent a GET /product request with a productId query parameter.



Let's use Burp Repeater to look at this behavior more closely.

5.3 Send the request to Burp Repeater

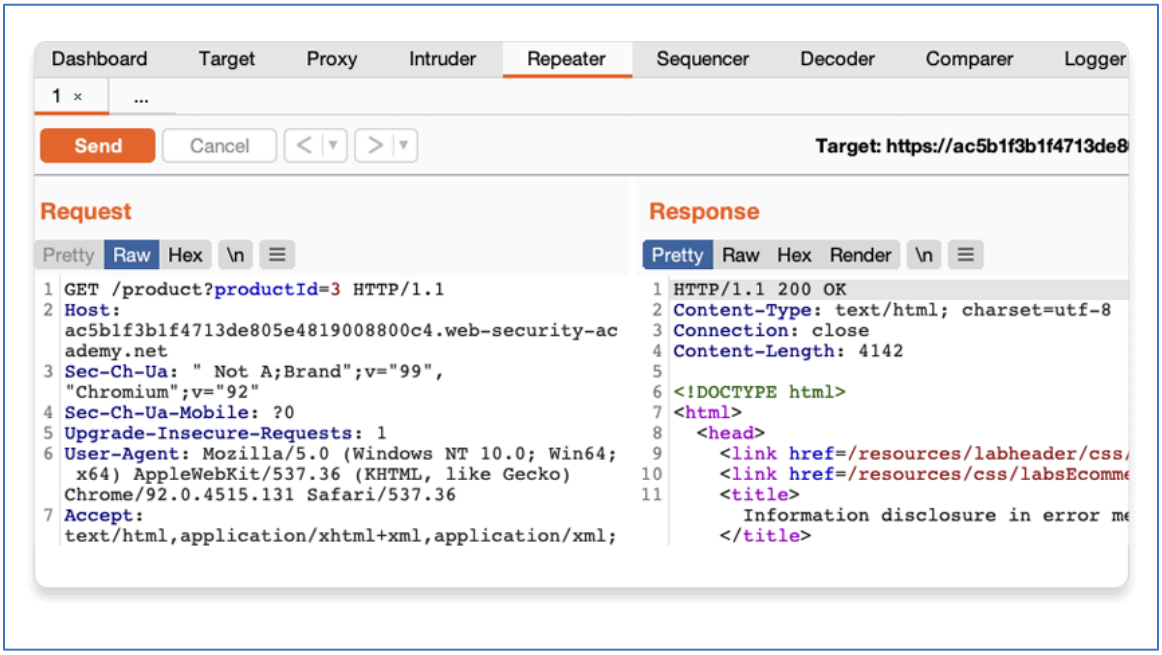
Right-click on any of the GET /product?productId=[...] requests and select Send to Repeater.



Go to the **Repeater** tab to see that your request is waiting for you in its own numbered tab.

5.4 Send the request and view the response

Click Send and view the response from the server. You can resend this request as many times as you like and the response will be updated each time.



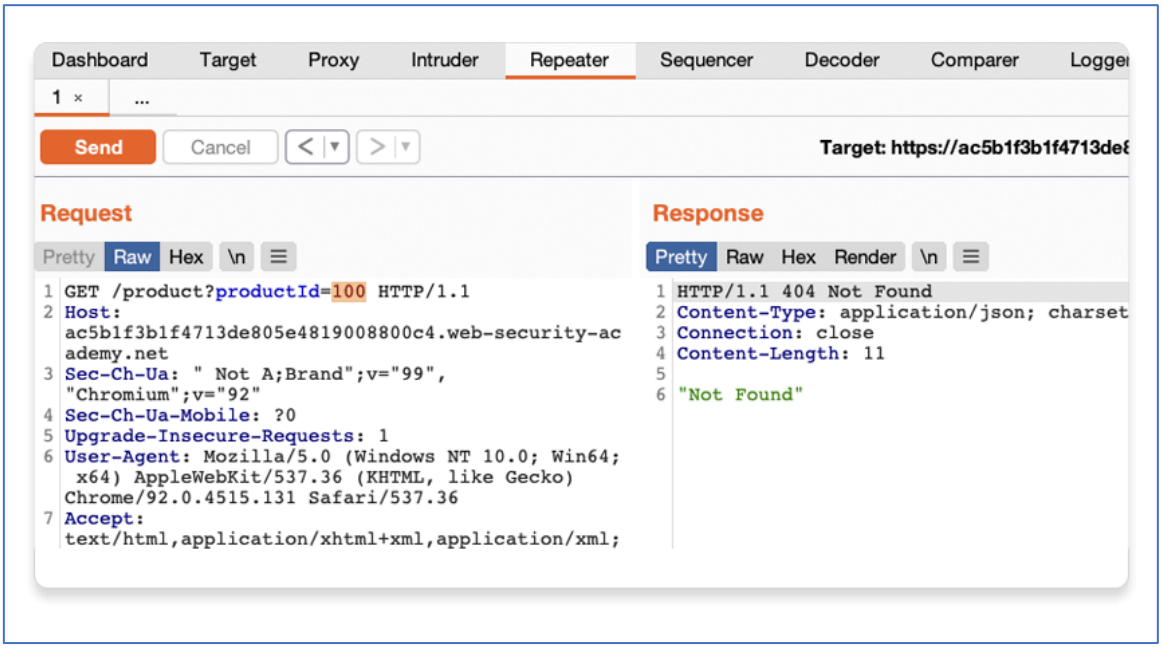
5.5 Testing different input with Burp Repeater

By resending the same request with different input each time, you can identify and confirm a variety of input-based vulnerabilities. This is one of the most common tasks you will perform during manual testing with Burp Suite.

5.6 Resend the request with different input

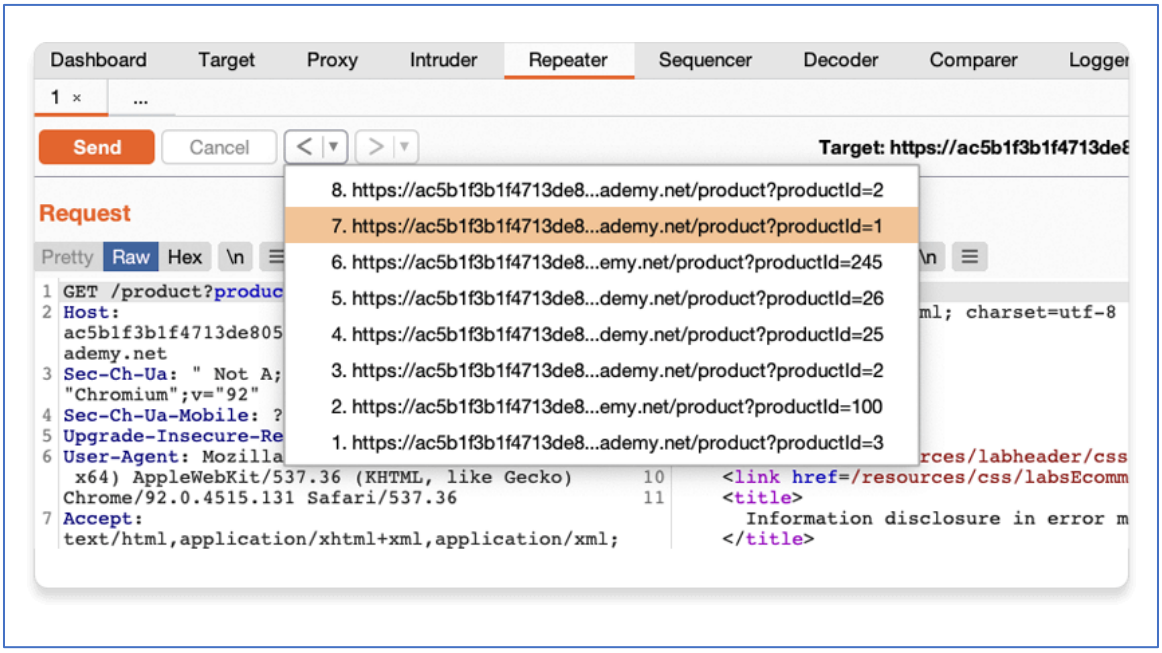
Change the number in the productId parameter and resend the request.

Try this with a few arbitrary numbers, including a couple of larger ones.



5.7 View the request history

Use the arrows to step back and forth through the history of requests that you've sent, along with their matching responses. The drop-down menu next to each arrow also lets you jump to a specific request in the history.



This is useful for returning to previous requests that you've sent in order to investigate a particular input further.

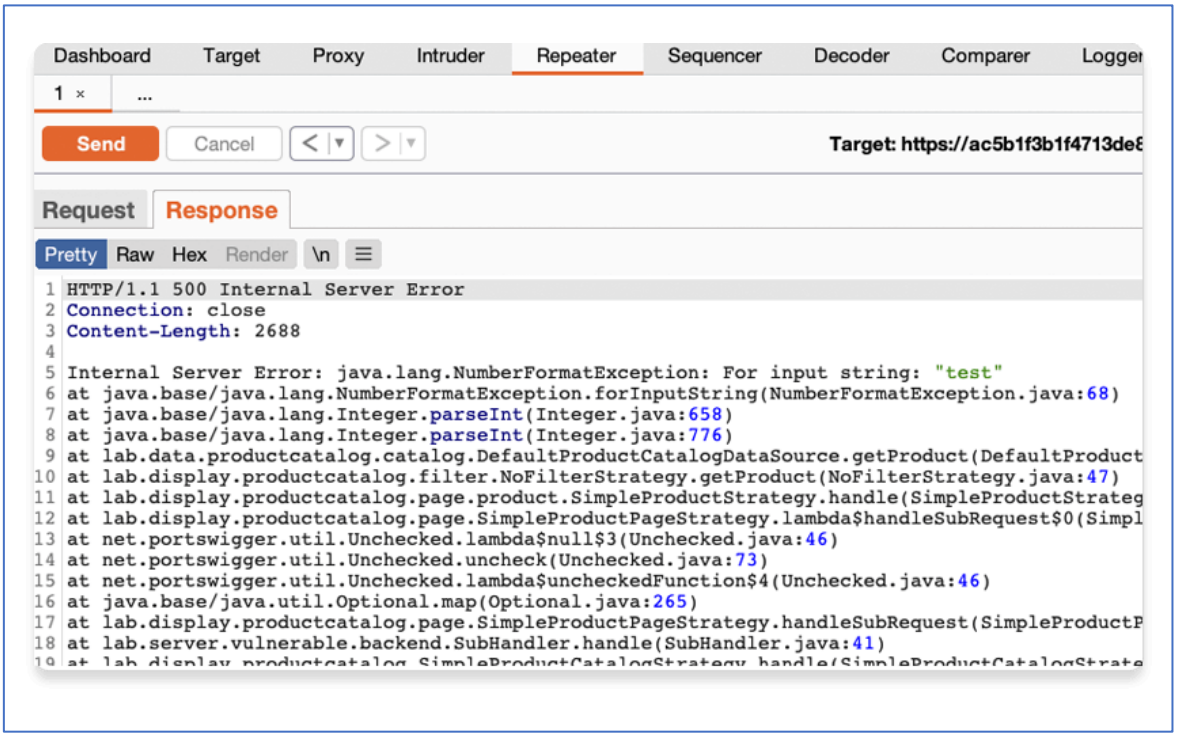
Compare the content of the responses, notice that you can successfully request different product pages by entering their ID, but receive a Not Found response if the server was unable to find a product with the given ID. Now we know how this page is supposed to work, we can use Burp Repeater to see how it responds to unexpected input.

5.8 Try sending unexpected input

The server seemingly expects to receive an integer value via this productId parameter. Let's see what happens if we send a different data type. Send another request where the productId is a string of characters.

5.9 Study the response

Observe that sending a non-integer productId has caused an exception. The server has sent a verbose error response containing a stack trace.



Notice that the response tells you that the website is using the Apache Struts framework - it even reveals which version.

6 Run your first scan

Burp Scanner can be used as both a fully automated scanner and a powerful means of augmenting your manual testing workflow. The list of vulnerabilities that Burp Scanner can detect is constantly growing. We work closely with our world-class research team to make sure that it stays up to speed with the latest techniques for finding both classic bugs and newly discovered vulnerabilities alike.

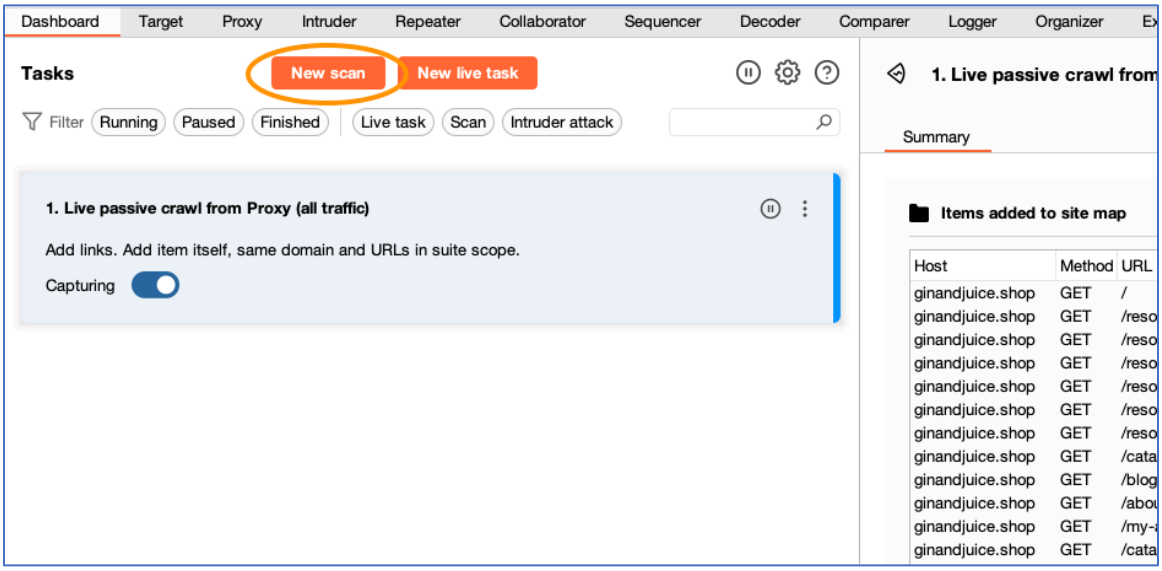
Scanning a website involves two phases:

1. **Crawling for content and functionality:** Burp Scanner first navigates around the target site, closely mirroring the behavior of real users. It catalogs the structure and content of the site, and the paths used to navigate it, in order to build a comprehensive map of the site.

2. **Auditing for vulnerabilities:** The audit phase of a scan involves analyzing the website's behavior to identify security vulnerabilities and other issues. Burp Scanner employs a wide range of techniques to deliver a high-coverage, accurate audit of the target.

6.1 Open the scan launcher

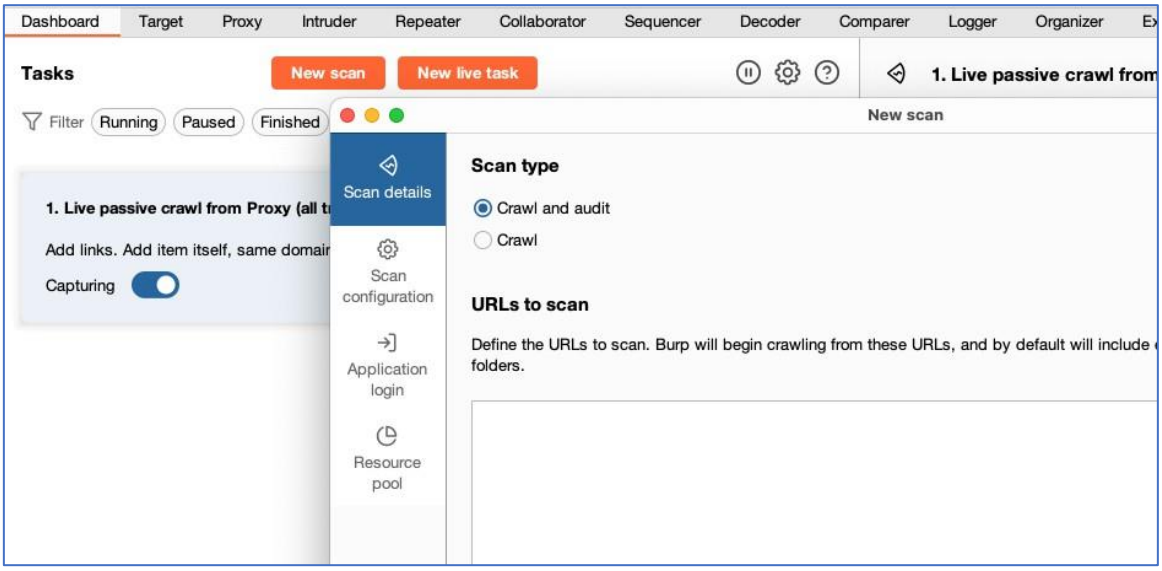
Go to the Dashboard tab and select **new** scan.



The Scan launcher dialog opens. This is where you can adjust various settings to control Burp Scanner's behavior.

6.2 Enter the URL of the target site

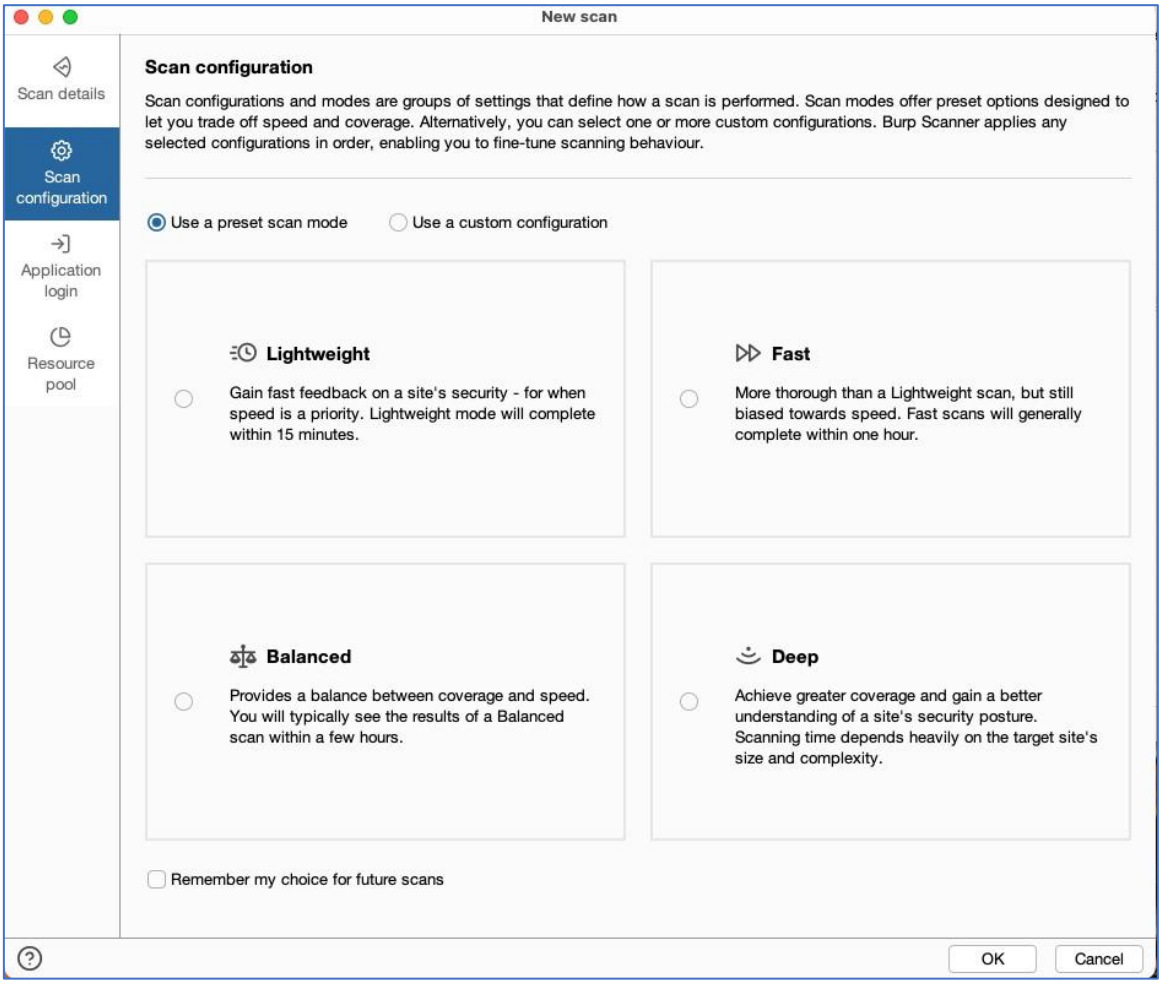
In the **URLs to scan** field, enter ginandjuice. shop. If necessary, remove the URL for the website that you set as a target scope in the earlier tutorial, Leave all the other settings as their default for now.



6.3 Configure the scan

Select Scan configuration. From here, you can fine-tune many aspects of Burp Scanner's behavior to suit different use cases and target sites

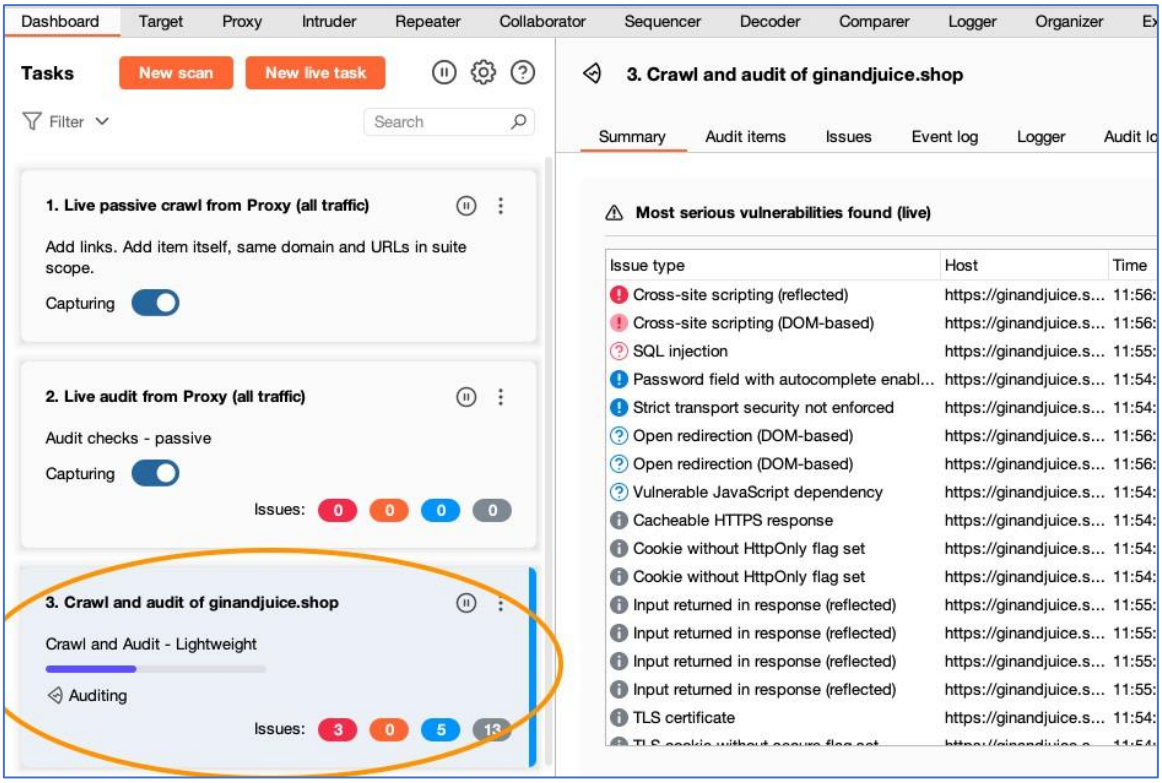
Make sure that use a preset scan mode is selected and click Lightweight. The Lightweight scan mode is intended to give a very high-level overview of a target as quickly as possible. Scans using this mode run for a maximum of 15 minutes.



6.4 Launch the scan

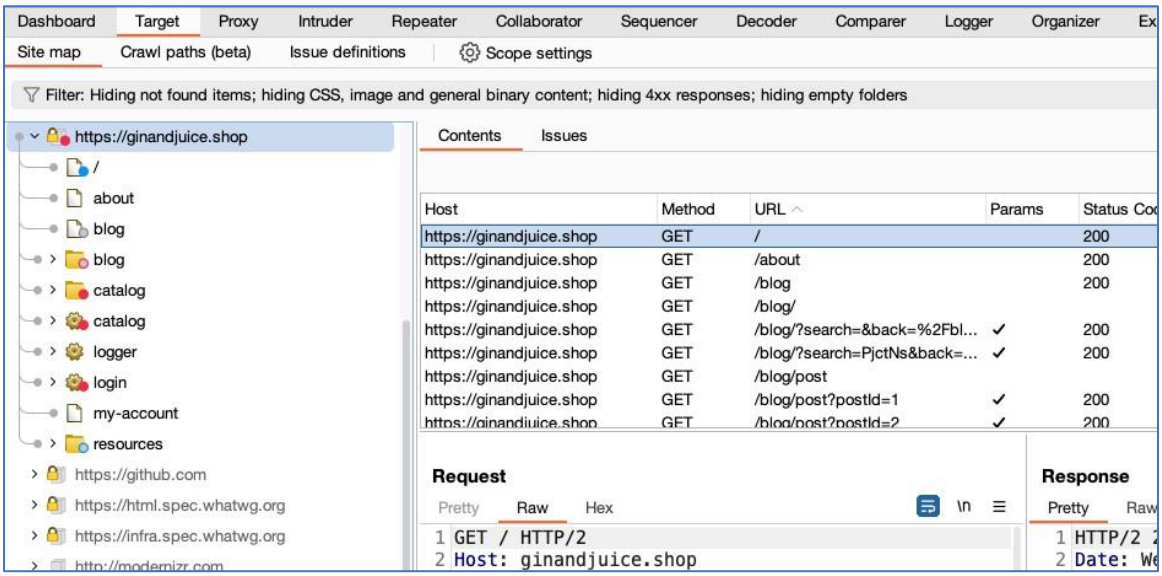
Click **OK** to launch the scan. Burp Scanner begins crawling from the URL you entered in the previous step.

Notice that a new task has been added to the **Dashboard** to represent this scan. You can select the task to view more details about its status and what it's currently doing.



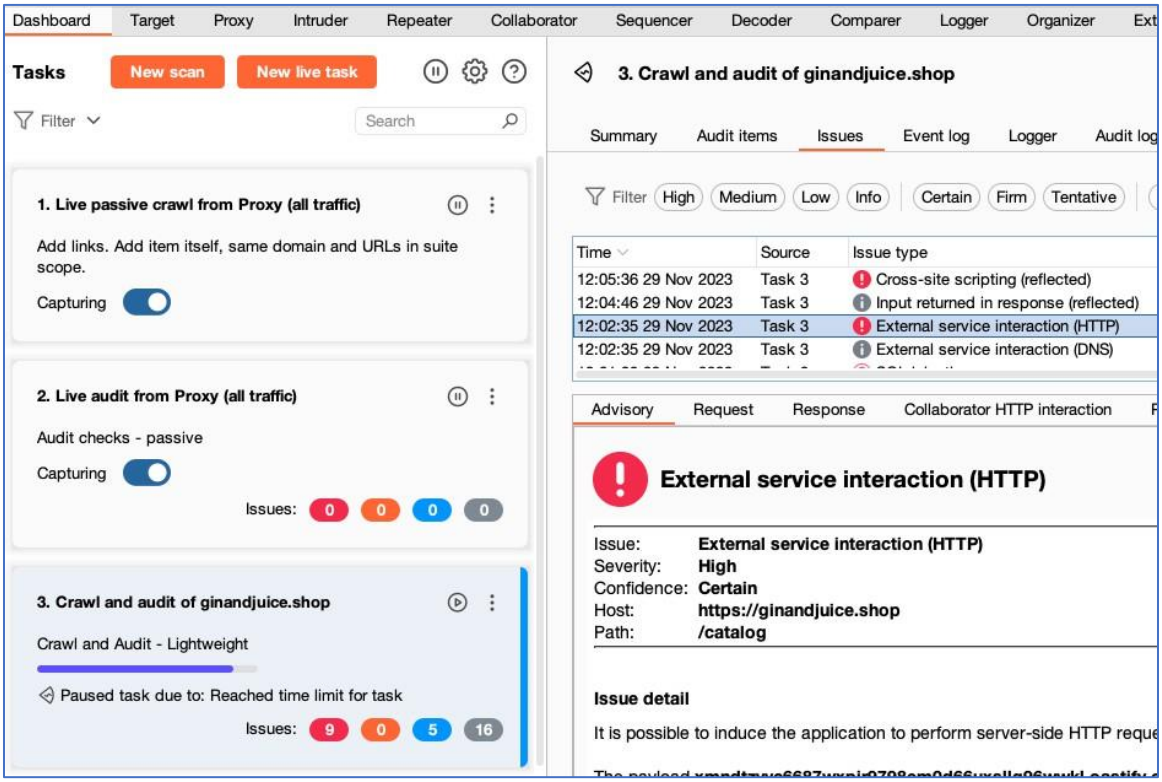
6.5 See the crawl in action

Go to the Target > Site map tab and notice the new entry for ginandjuice. shop. Expand this node to see all of the content that the crawler has managed to discover so far. If you wait a few seconds, you'll see the map being updated in real time.



6.6 View the identified issues

Monitor the scan's status in the Dashboard tab. After a minute or two, the crawl will finish and Burp Scanner will begin auditing for vulnerabilities. To monitor the scan for any issues it finds, select the scan from the Tasks list. In the main panel, go to the Issues tab.



If you select an issue, you can see an **Advisory** tab, which contains key information about the issue type, including a detailed description and some remediation advice. Next to this are several tabs that provide evidence that Burp Scanner found for this issue. This is typically a **Request** and **Response** but will differ depending on the issue type.