Conduct a manual security testing for a local web application or an API using proxy tools like burp suite/paros etc and provide a report.

BURPSUITE

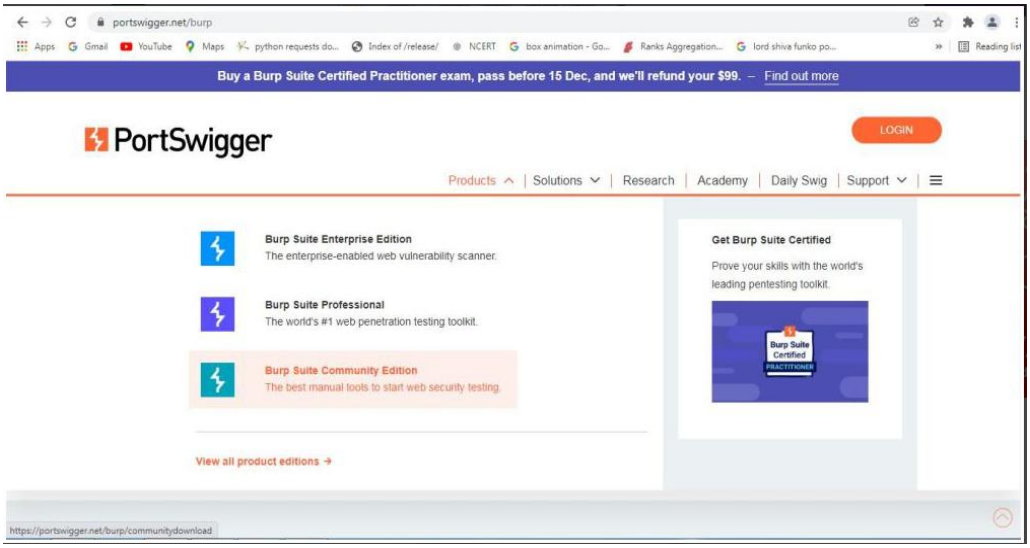
Burp or Burp Suite is a set of tools used for penetration testing of web applications. The tool is written in Java and developed by PortSwigger Web Security. The tool has three editions: a **Community Edition** that can be downloaded free of charge, a **Professional Edition** and an **Enterprise Edition** that can be purchased after a trial period. It is a collection of different tools which are brought together in a single application for performing security testing of Web applications. Burp Suite is widely used by penetration testers to test and identify different vulnerabilities which are present in web applications and exploit them to fix those security issues. Burp Suite has a large number of features which include proxy, intruder, repeater, sequencer, decoder, compare, and many more.

1. Installing Burp Suite on Windows:

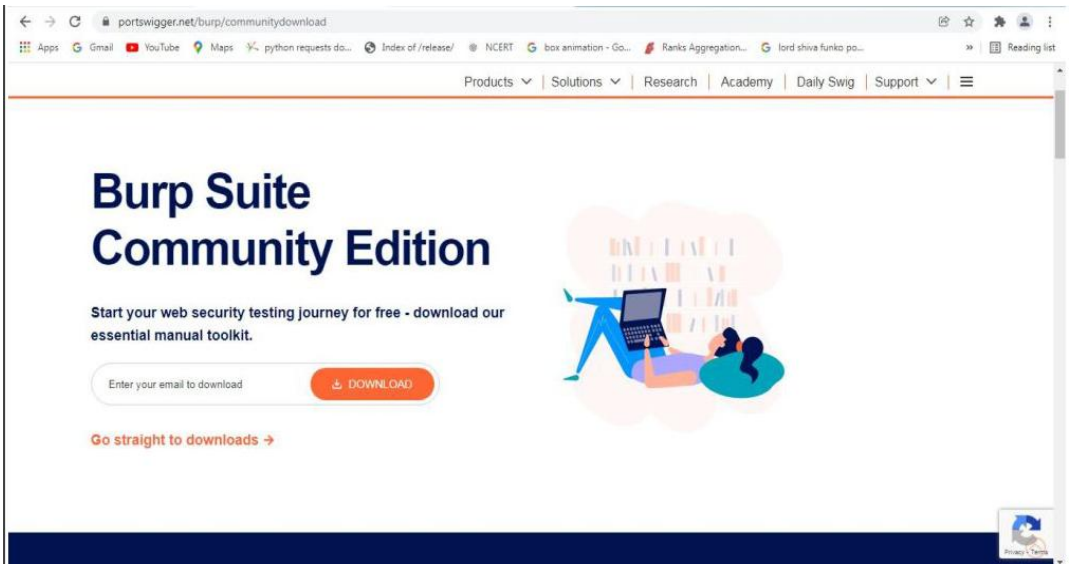
Follow the below steps to install Burp Suite on Windows:

Step 1: Visit the official Burp Suite website (https://portswigger.net/burp) using any web browser.

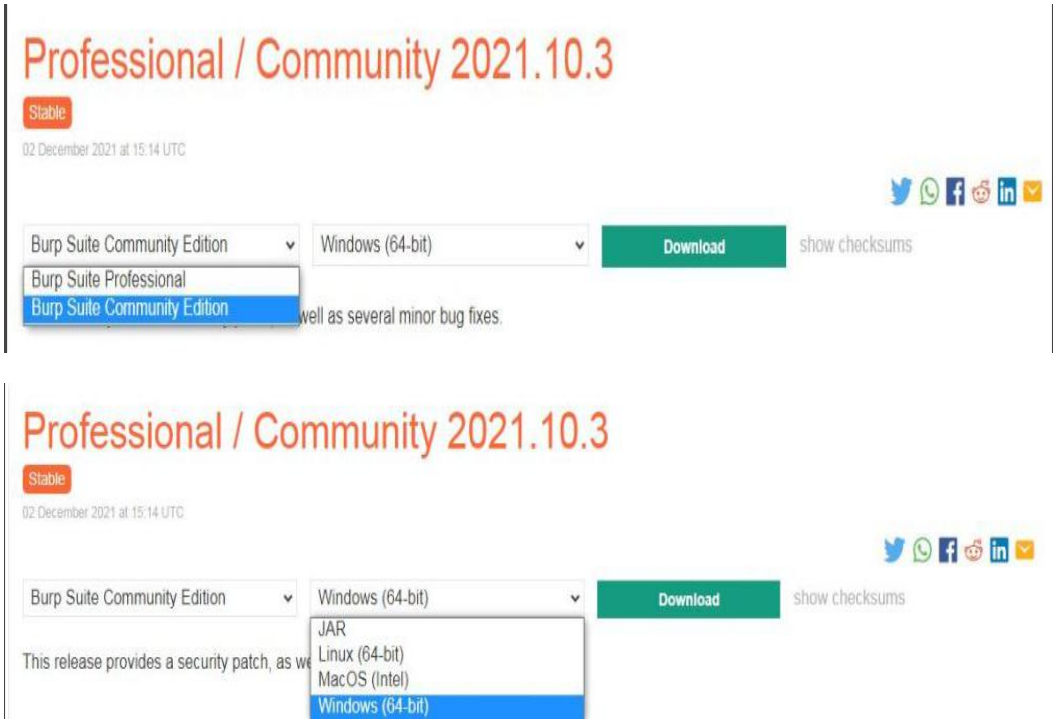
Step 2: Click on Products, a list of different Burp Suites will open, choose Burp suite Community Edition as it is free, click on it.



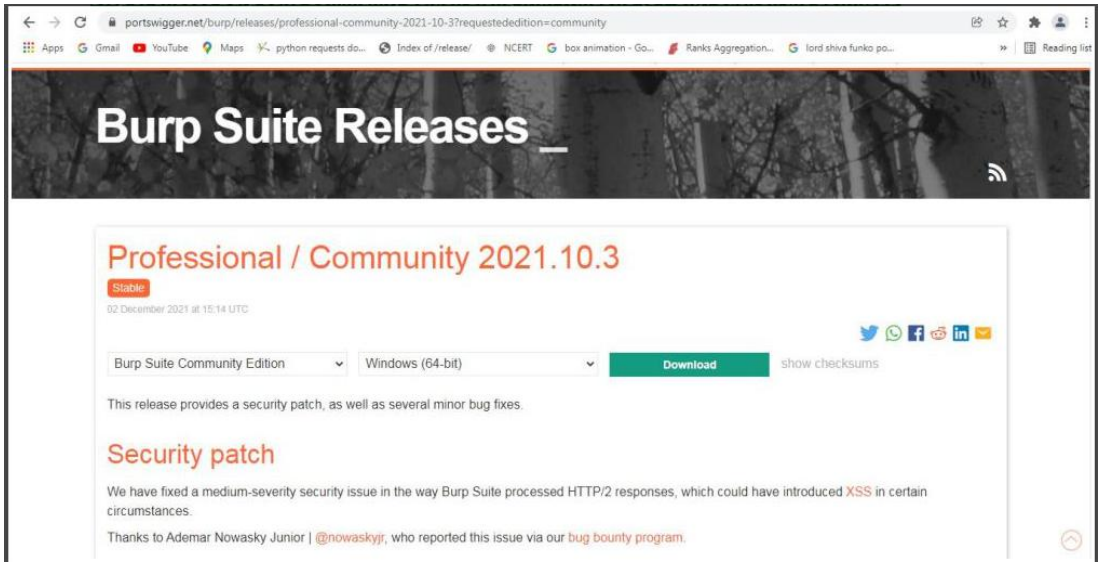
Step 3: New webpage will open, which will ask for email id, and other option is Go Straight to downloads. Click on Go straight to downloads.



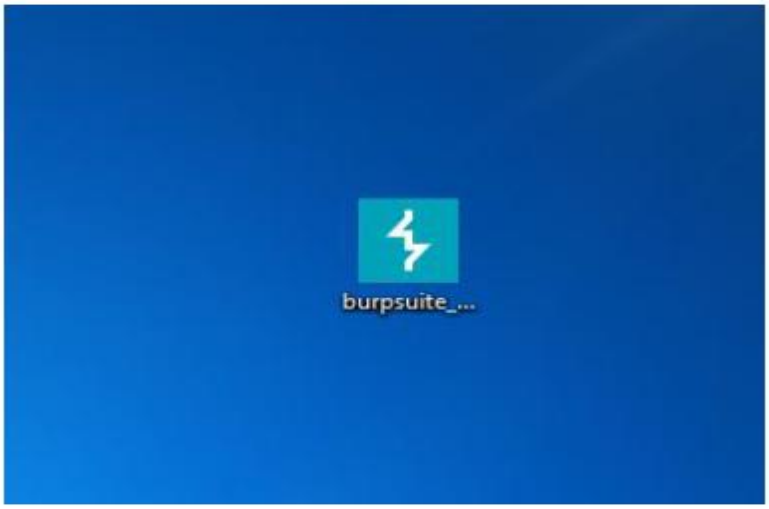
Step 4: After clicking on Go straight to downloads new webpage will open which will contain two versions of burp suite one is Burp suite community edition and the other is burp suite professional along with compatibility for different operating systems.



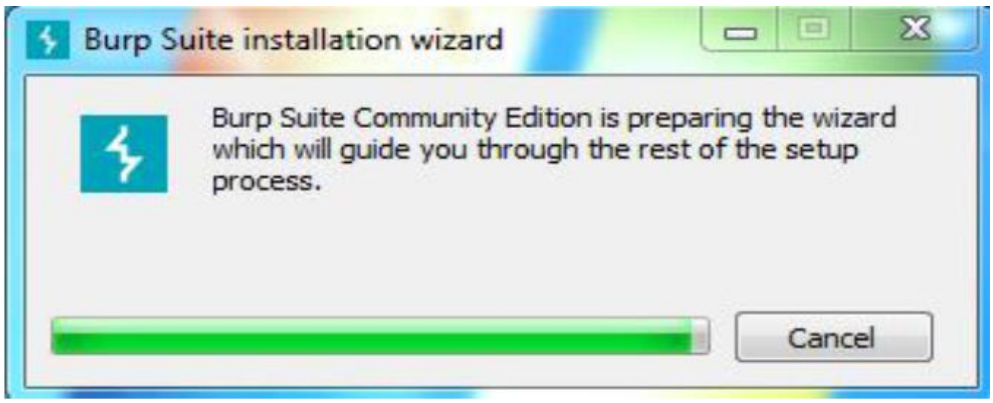
Step 5: Choose Burp suite Community Edition along with Windows (64-bit). Click on the download button, downloading of the executable file will start shortly. It is a big 210 MB file that will take some time depending on download speed.



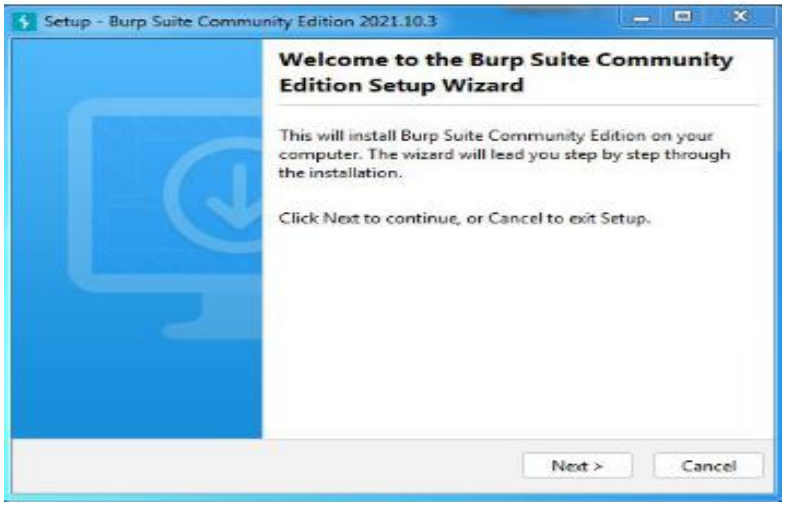
Step 6: Now check for the executable file in downloads in your system and run it.



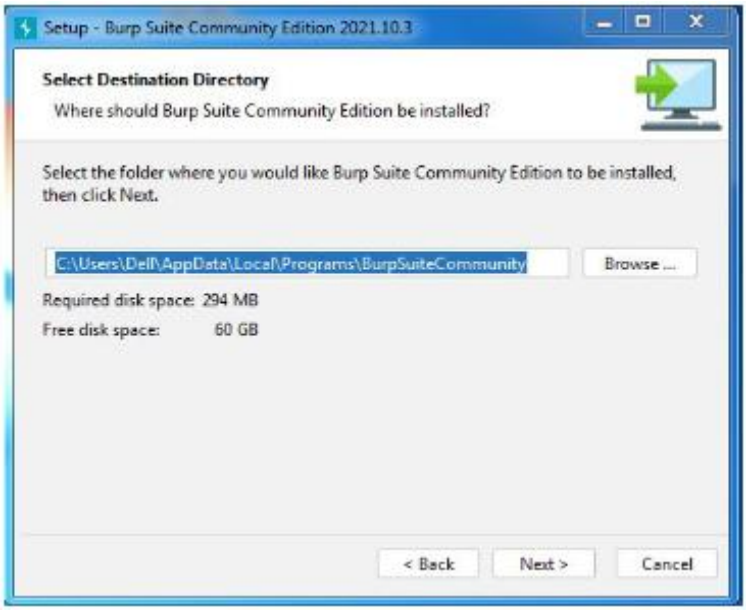
Step 7: Loading of Installation Wizard will appear which will take a few seconds.



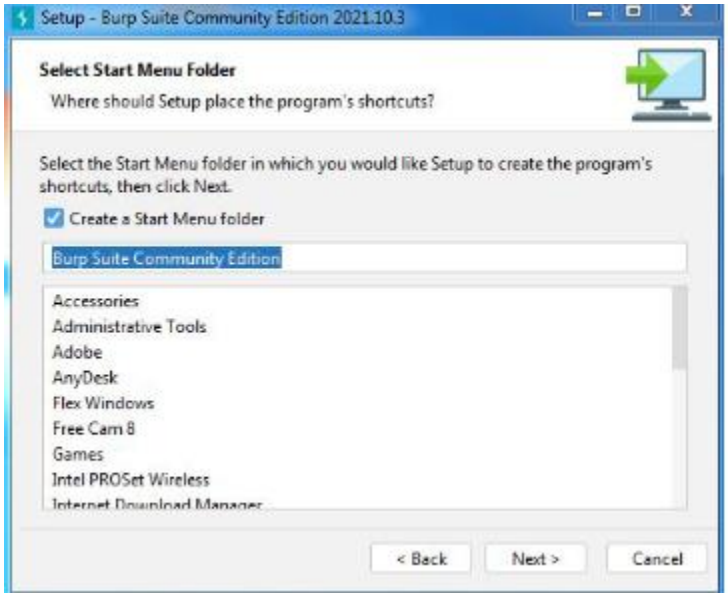
Step 8: After this Setup screen will appear, click on Next.



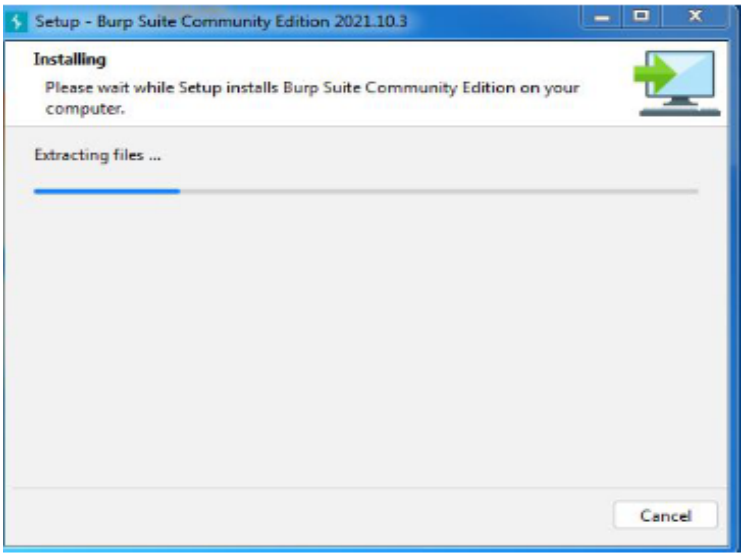
Step 9: The next screen will be of installing location so choose the drive which will have sufficient memory space for installation. It needed a memory space of 294 MB.



Step 10: Next screen will be of choosing Start menu folder so don't do anything just click on Next Button.



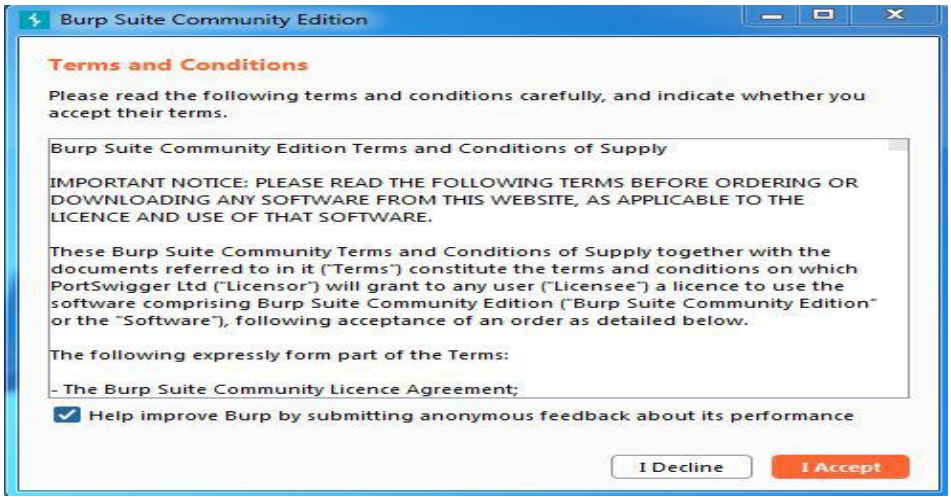
Step 11: After this installation process will start and will hardly take a minute to complete the installation. Application



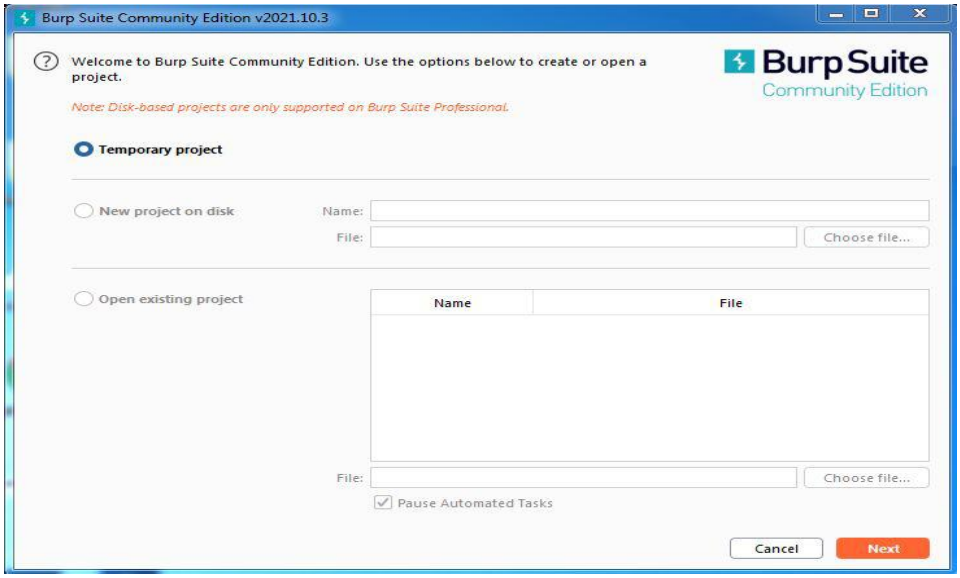
Step 12: Click on Finish after the installation process is complete.

Step 13: Burp suite is successfully installed on the system and an icon is created on the desktop.

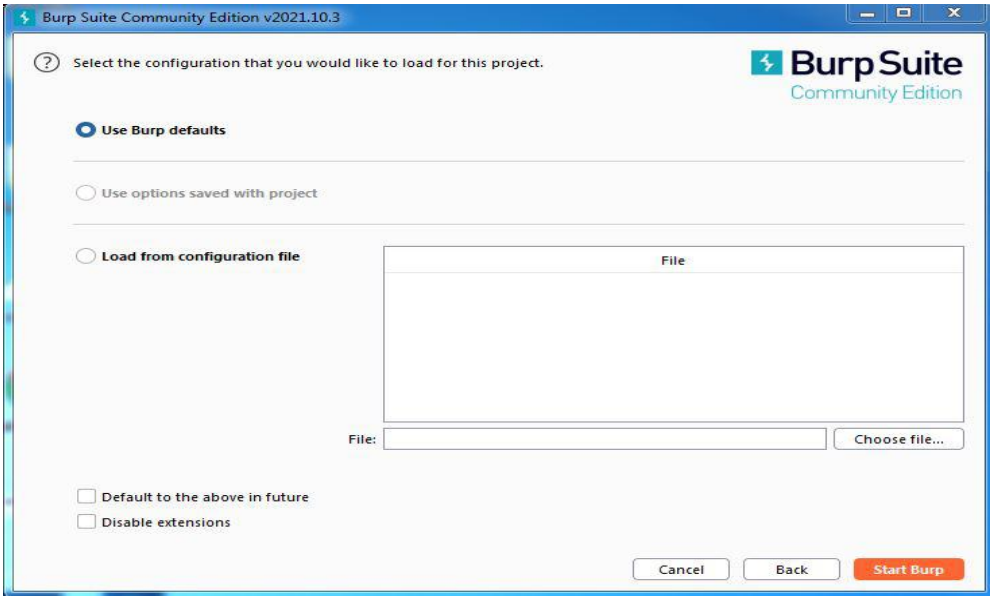
Step 14: Run the software, screen containing terms and conditions will appear Click on I Accept.



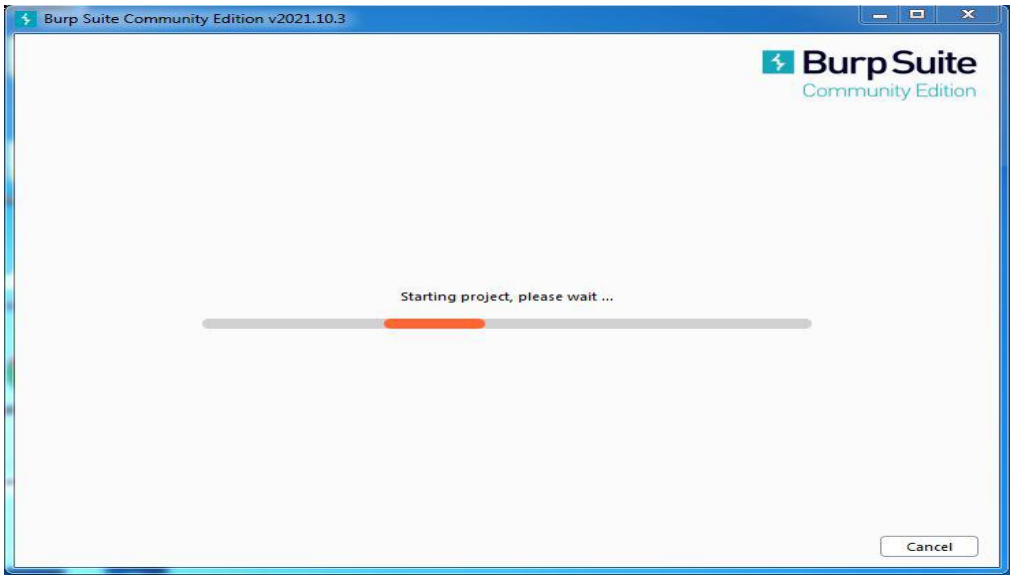
Step 15: New screen containing information regarding the project will appear, Choose temporary project and click Next.



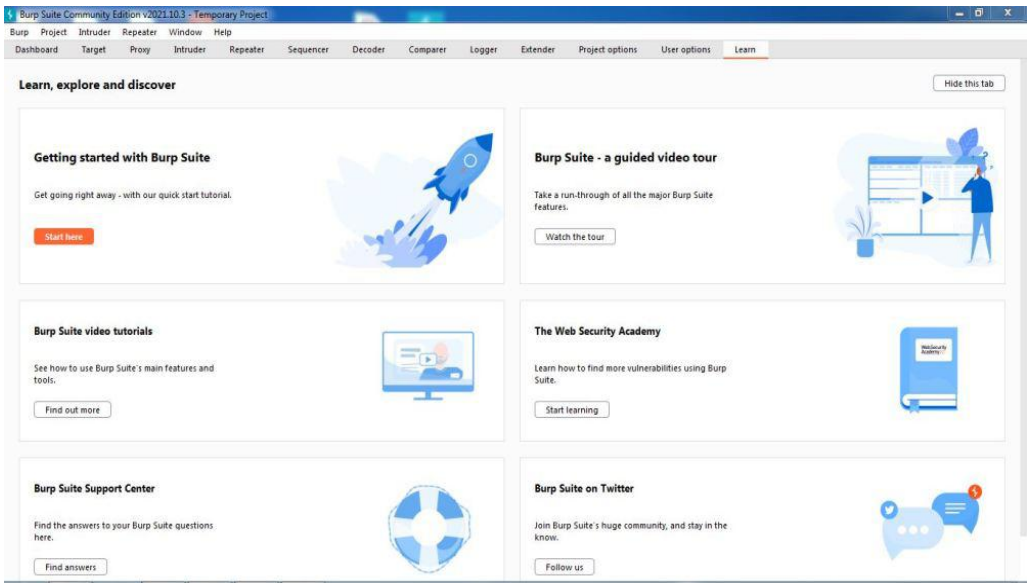
Step 16: Next screen is about using default settings or loading from configuration file, click on Use Burp Defaults.



Step 17: Project will start loading.



Step 18: Finally new project window will appear.



Congratulations!! At this point, you have successfully installed Burp Suite on your windows system.

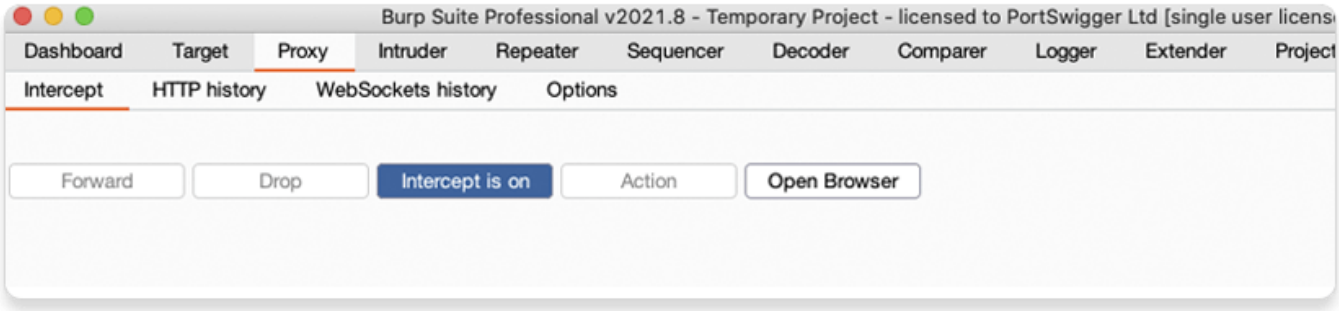
2. Intercept HTTP traffic with Burp Proxy

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.

Step 1: Launch Burp's browser

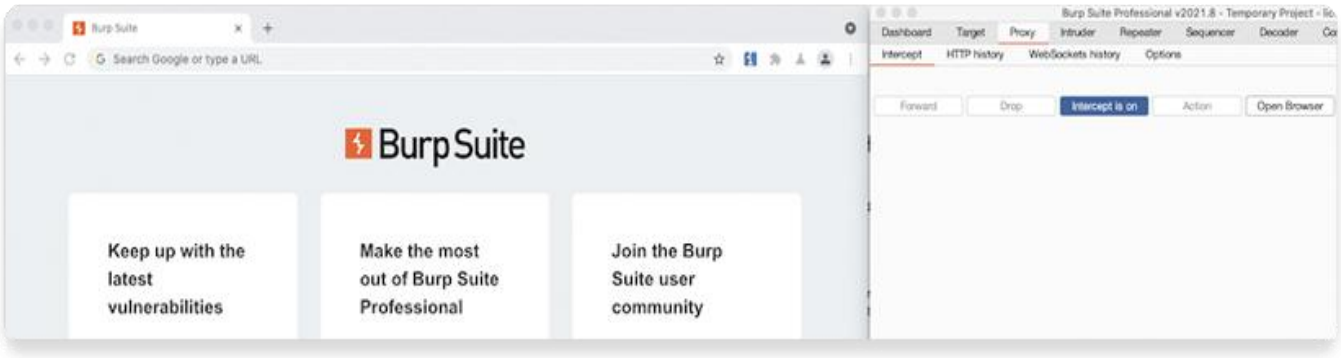
Go to the **Proxy > Intercept** tab.

Click the **Intercept is off** button, so it toggles to **Intercept is on.**



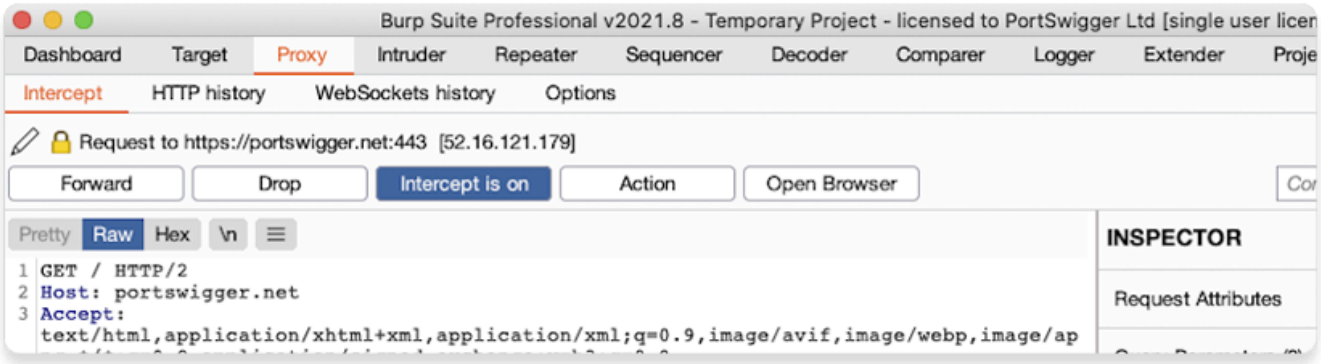
Click **Open Browser**. This launches Burp's browser, which is preconfigured to work with Burp right out of the box.

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



Step 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.



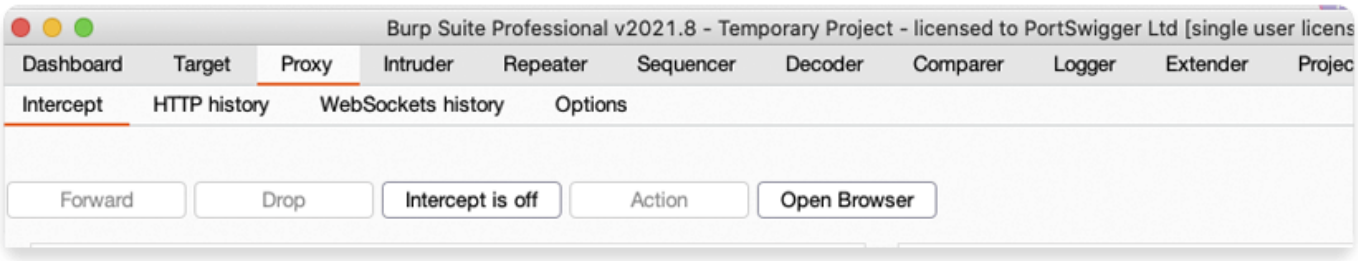
The request is held here so that you can study it, and even modify it, before forwarding it to the target server.

Step 3: Forward the request

Click the **Forward** button several times to send the intercepted request, and any subsequent ones, until the page loads in Burp's browser.

Step 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. Click the **Intercept is on** button so that it now says **Intercept is off**.

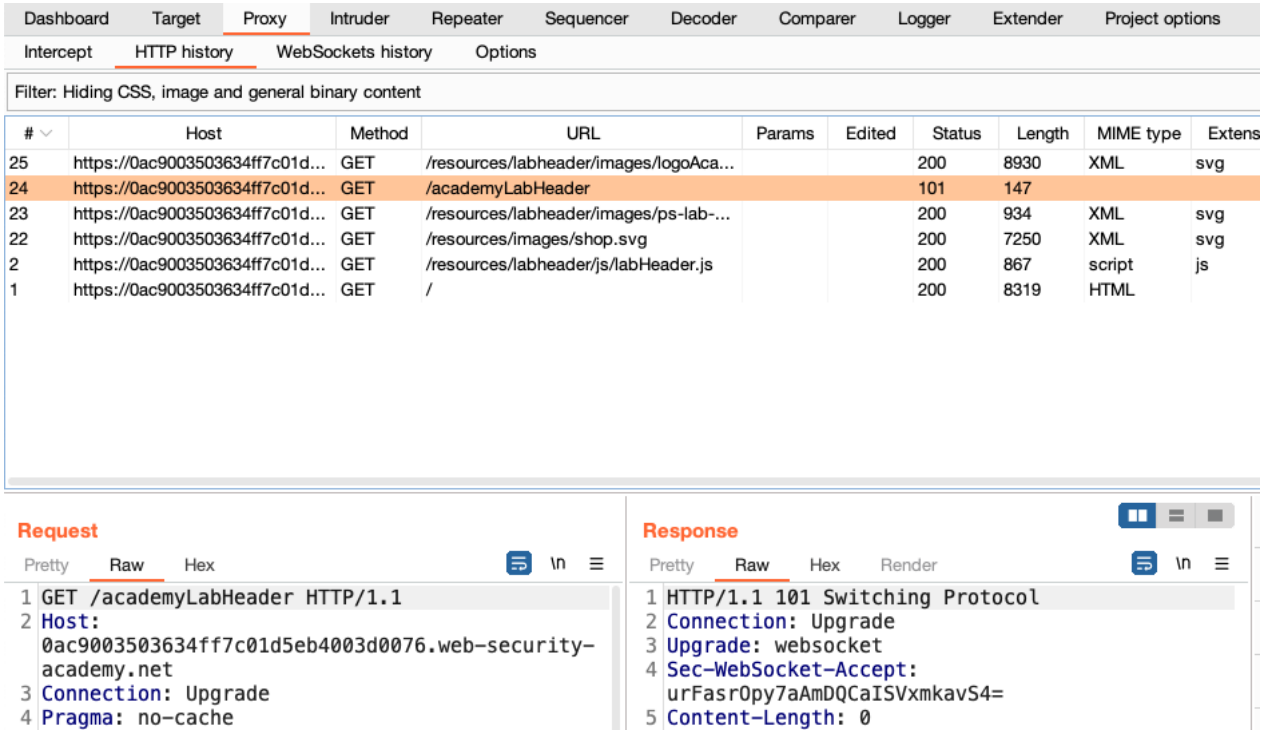


Go back to the browser and confirm that you can now interact with the site as normal.

Step 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 interception 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

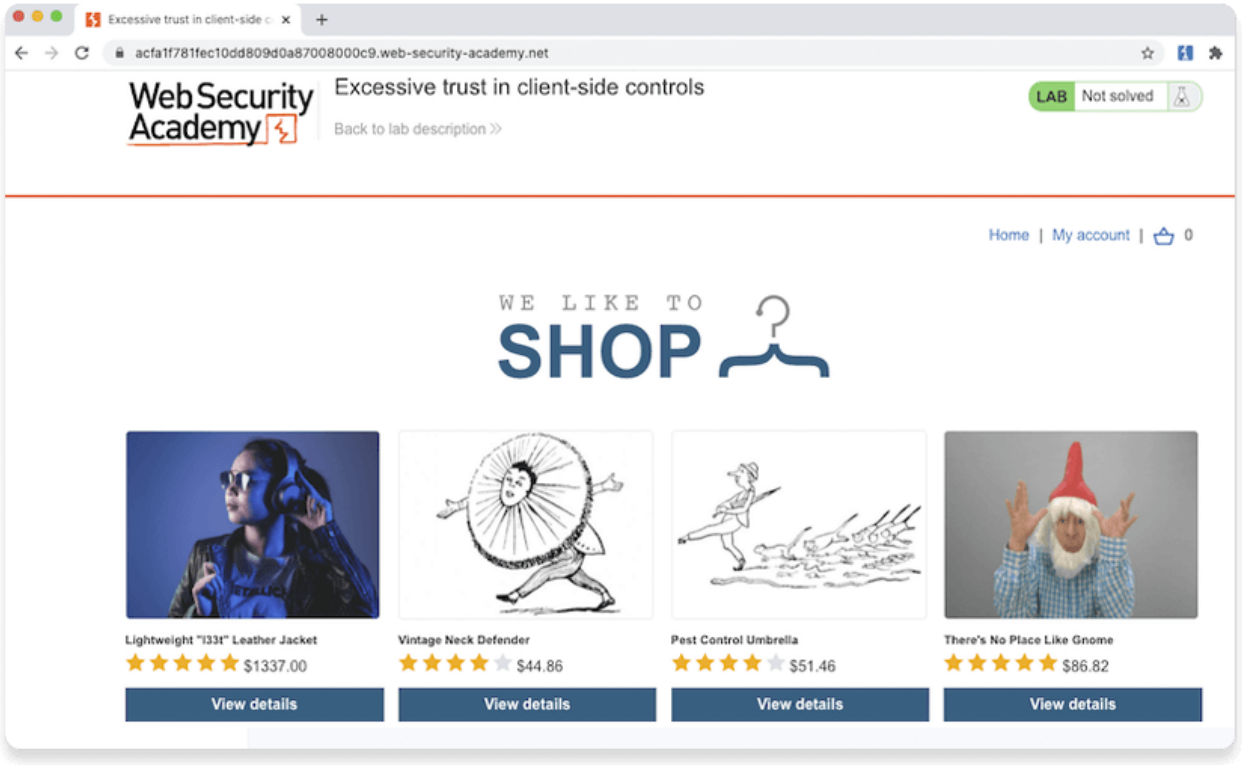
Step 1: Access the vulnerable website in Burp's browser

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.



Step 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

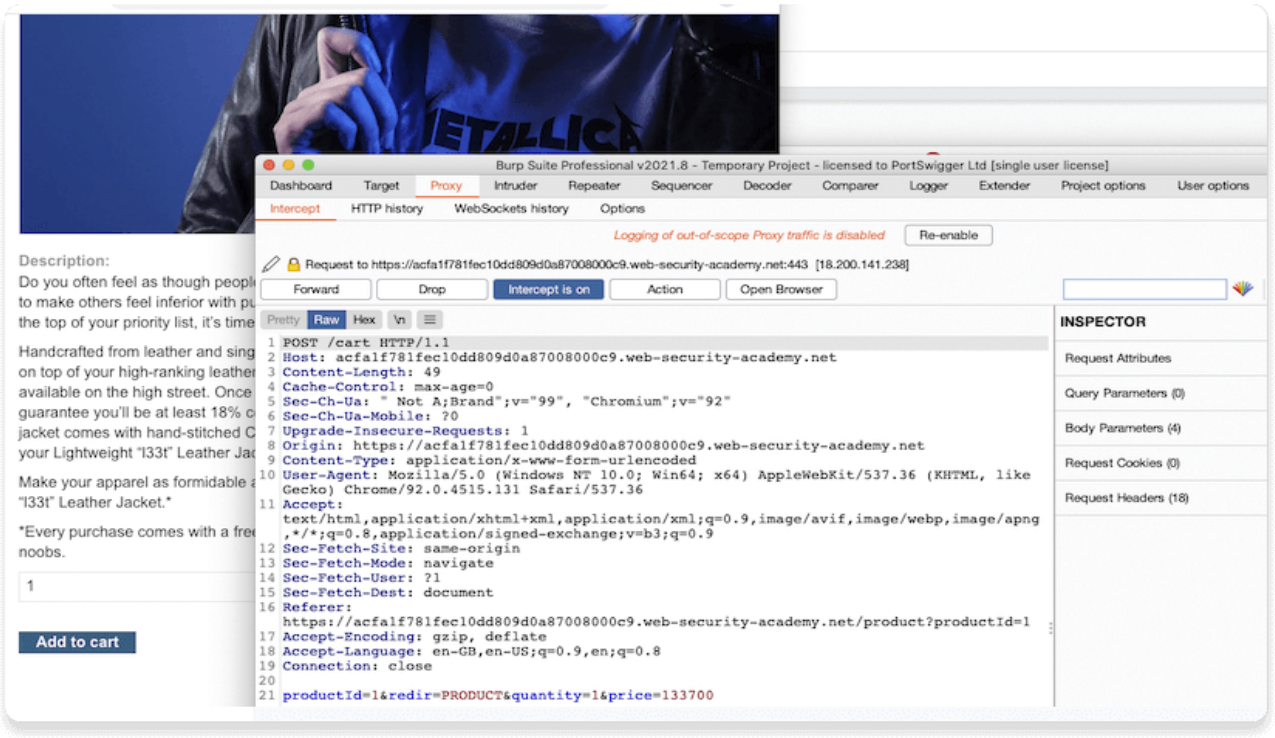
Notice that you have just $100 of store credit.

Step 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**.

Step 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.



Note

You may initially see a different request on the **Proxy > Intercept** tab if the browser is doing something else in the background. In this case, just click **Forward** until you see the POST /cart request as shown in the screenshot above.

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.

Step 5: Modify the request

Change the value of the price parameter to 1 and click **Forward** to send the modified request to the server.

```
20
21 productId=1&redir=PRODUCT&quantity=1&price=1
```

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

Step 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.

Note

There is no way to modify the price via the web interface. You were only able to make this change thanks to Burp Proxy.

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

Congratulations, you've also just solved your first Web Security Academy lab! You've also learned how to intercept, review, and manipulate HTTP traffic using Burp Proxy.

4. Set the target scope

Step 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
```

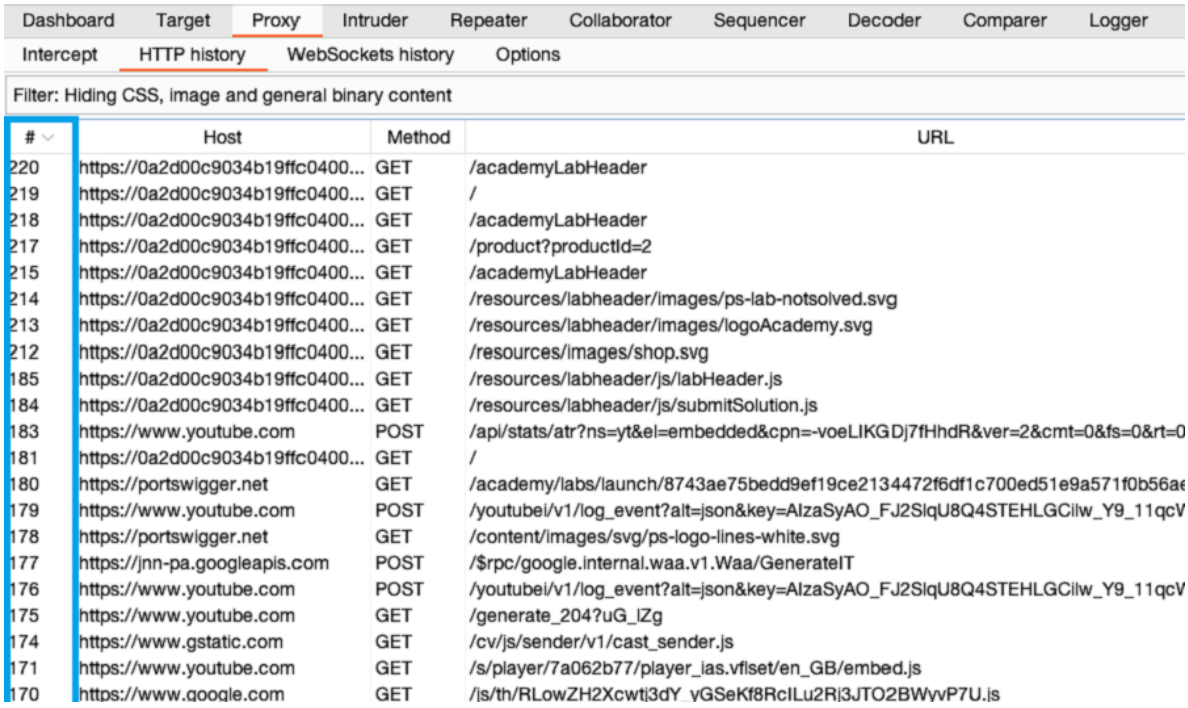
When the page loads, click **Access the lab**. If prompted, log in to your portswigger.net account. After a few seconds, you will see your own instance of a fake shopping website.

Step 2: Browse the target site

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

Step 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.

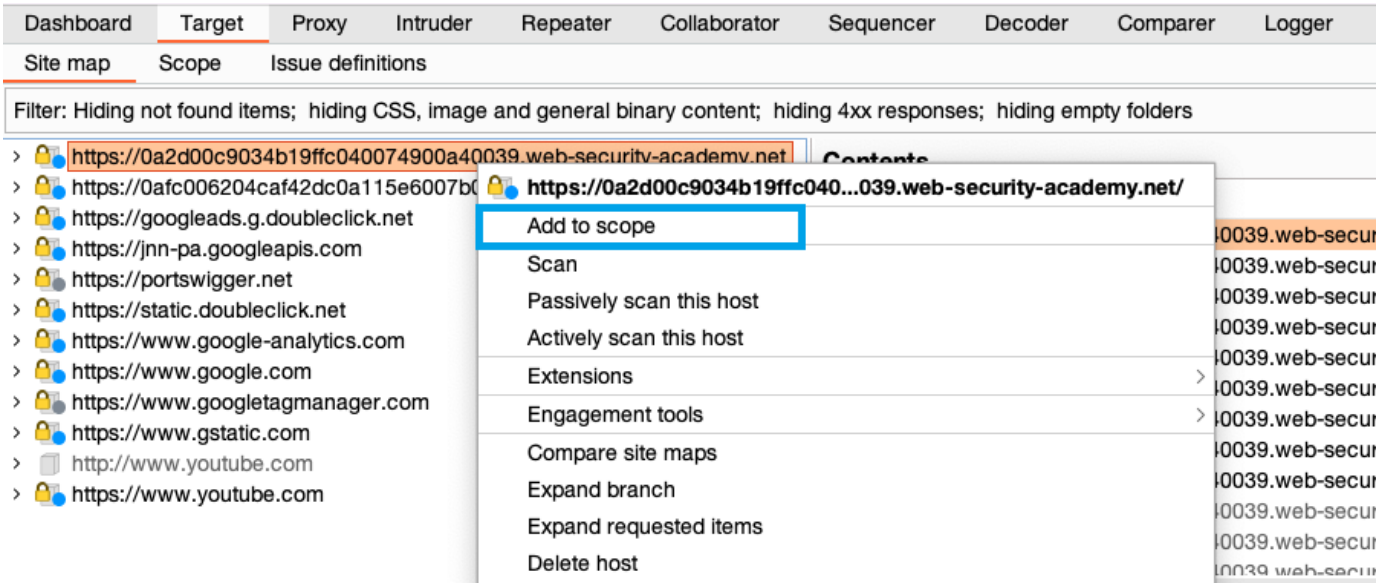


#	Host	Method	URL
220	https://0a2d00c9034b19ffc0400...	GET	/academyLabHeader
219	https://0a2d00c9034b19ffc0400...	GET	/
218	https://0a2d00c9034b19ffc0400...	GET	/academyLabHeader
217	https://0a2d00c9034b19ffc0400...	GET	/product?productId=2
215	https://0a2d00c9034b19ffc0400...	GET	/academyLabHeader
214	https://0a2d00c9034b19ffc0400...	GET	/resources/labheader/images/ps-lab-notsolved.svg
213	https://0a2d00c9034b19ffc0400...	GET	/resources/labheader/images/logoAcademy.svg
212	https://0a2d00c9034b19ffc0400...	GET	/resources/images/shop.svg
185	https://0a2d00c9034b19ffc0400...	GET	/resources/labheader/js/labHeader.js
184	https://0a2d00c9034b19ffc0400...	GET	/resources/labheader/js/submitSolution.js
183	https://www.youtube.com	POST	/api/stats/atr?ns=yt&el=embedded&cpn=-voeLIKGDj7fHhdR&ver=2&cmt=0&fs=0&rt=0
181	https://0a2d00c9034b19ffc0400...	GET	/
180	https://portswigger.net	GET	/academy/labs/launch/8743ae75bedd9ef19ce2134472f6df1c700ed51e9a571f0b56ae
179	https://www.youtube.com	POST	/youtubei/v1/log_event?alt=json&key=AIzaSyAO_FJ2SlqU8Q4STEHLGCilw_Y9_11qcV
178	https://portswigger.net	GET	/content/images/svg/ps-logo-lines-white.svg
177	https://jnn-pa.googleapis.com	POST	/$rpc/google.internal.waa.v1.Waa/GenerateIT
176	https://www.youtube.com	POST	/youtubei/v1/log_event?alt=json&key=AIzaSyAO_FJ2SlqU8Q4STEHLGCilw_Y9_11qcV
175	https://www.youtube.com	GET	/generate_204?uG_lZg
174	https://www.gstatic.com	GET	/cv/js/sender/v1/cast_sender.js
171	https://www.youtube.com	GET	/s/player/7a062b77/player_ias.vflset/en_GB/embed.js
170	https://www.google.com	GET	/js/th/RLowZH2Xcwtj3dY_yGSeKf8RcILu2Ri3JTO2BWyvP7U.js

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.

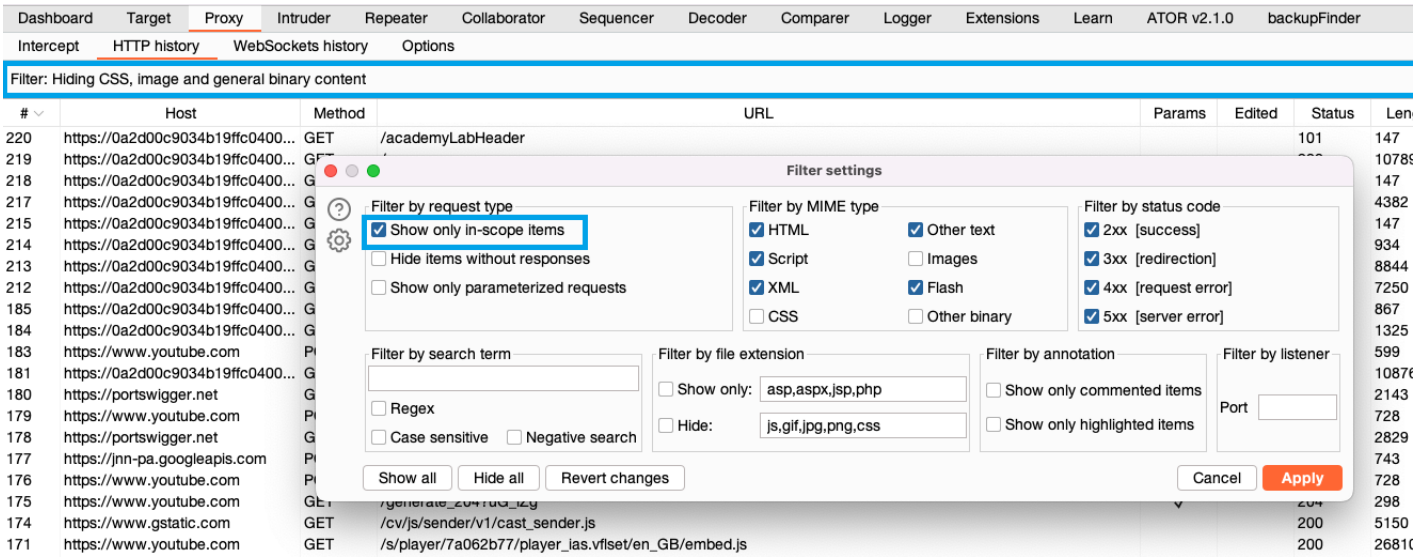
Step 4: Set the target scope

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.



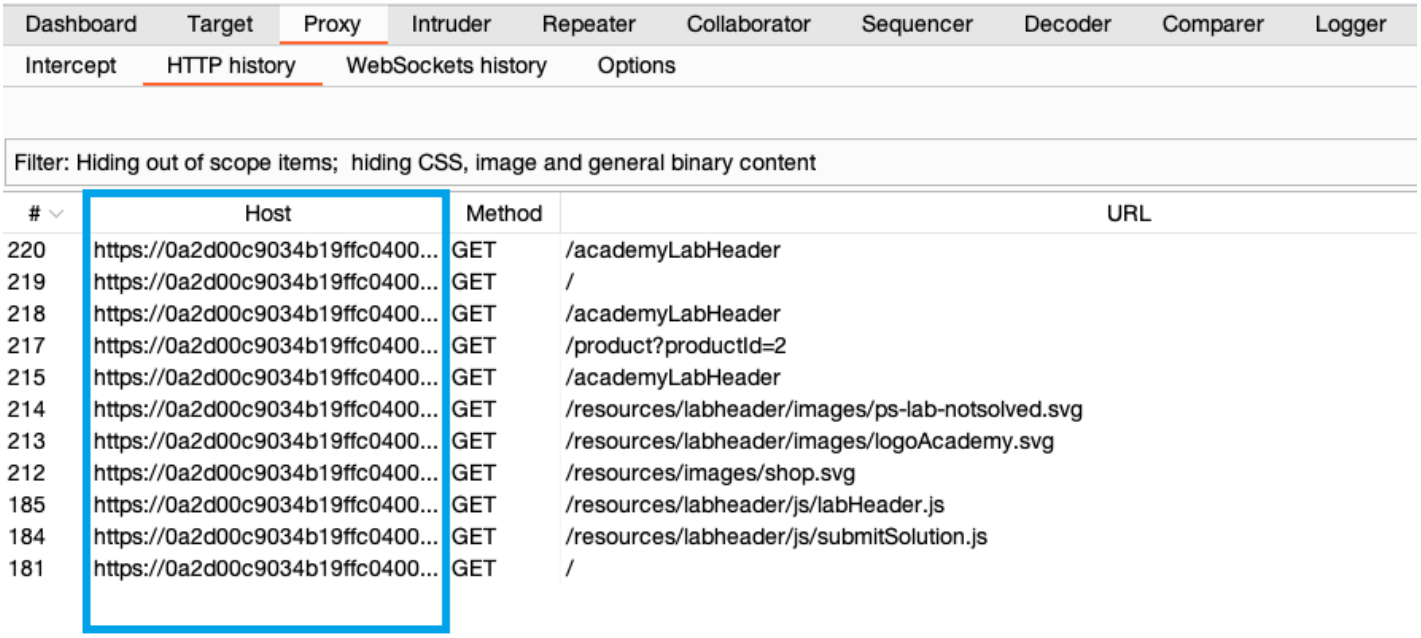
Step 5: Filter HTTP history

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.



Congratulations, you've successfully set the target scope and used it to simplify your HTTP history. In the next section you'll build on this work to complete the lab.

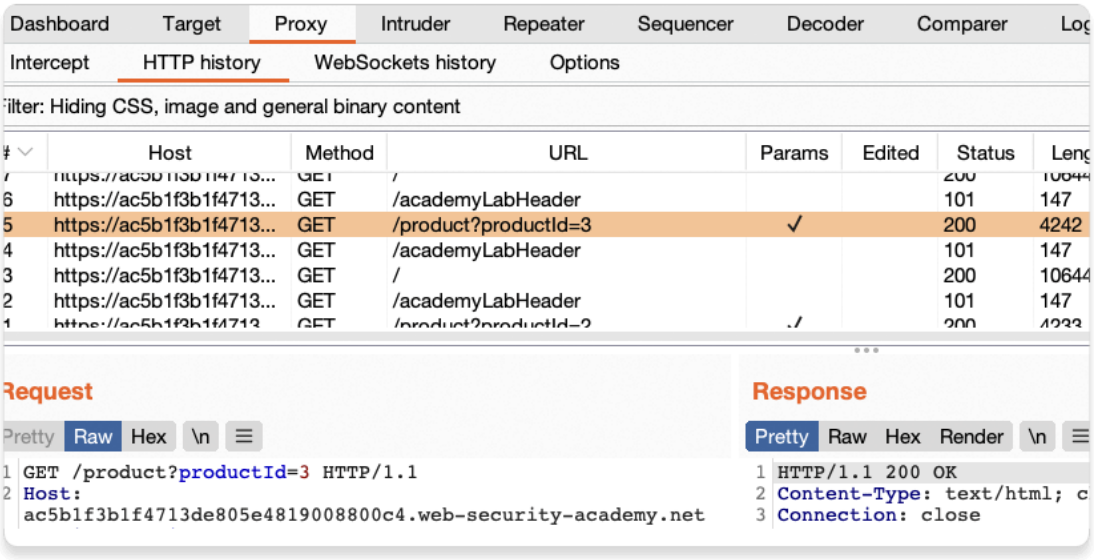
5. Reissue requests with Burp Repeater

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.

Step 1: Identify an interesting request

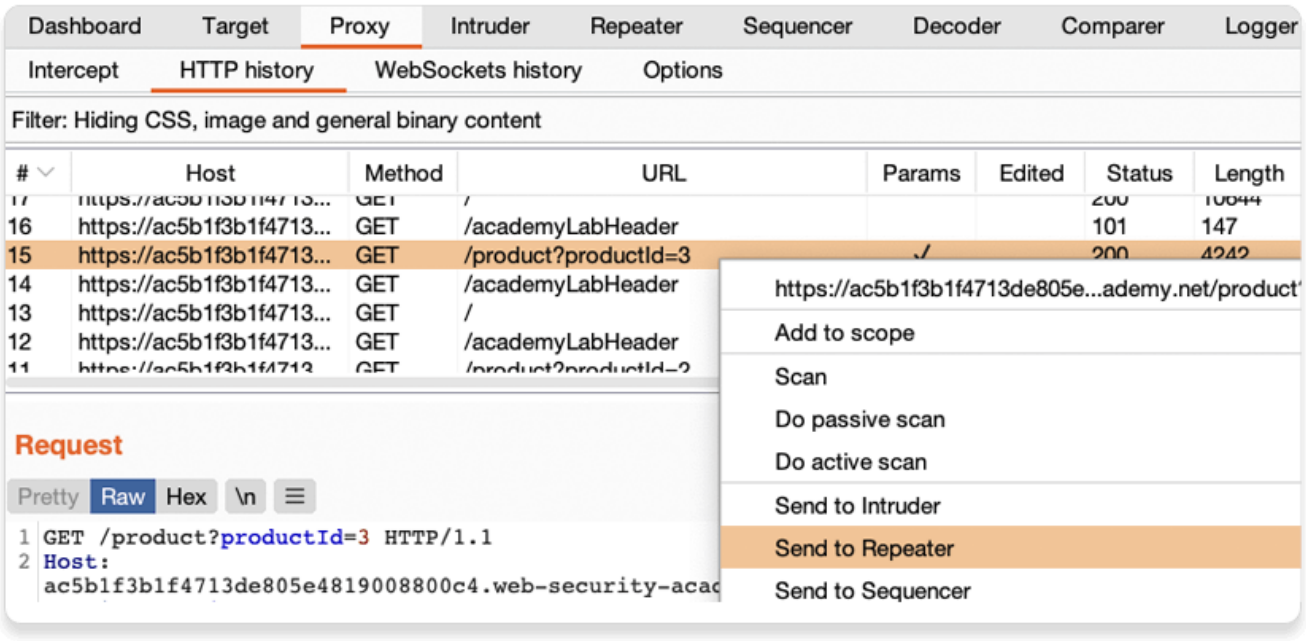
In the previous tutorial, you browsed a fake shopping website. Notice that each time you accessed a product page, the browser sent a GET /product request with a productId query parameter.



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

Step 4: Send the request to Burp Repeater

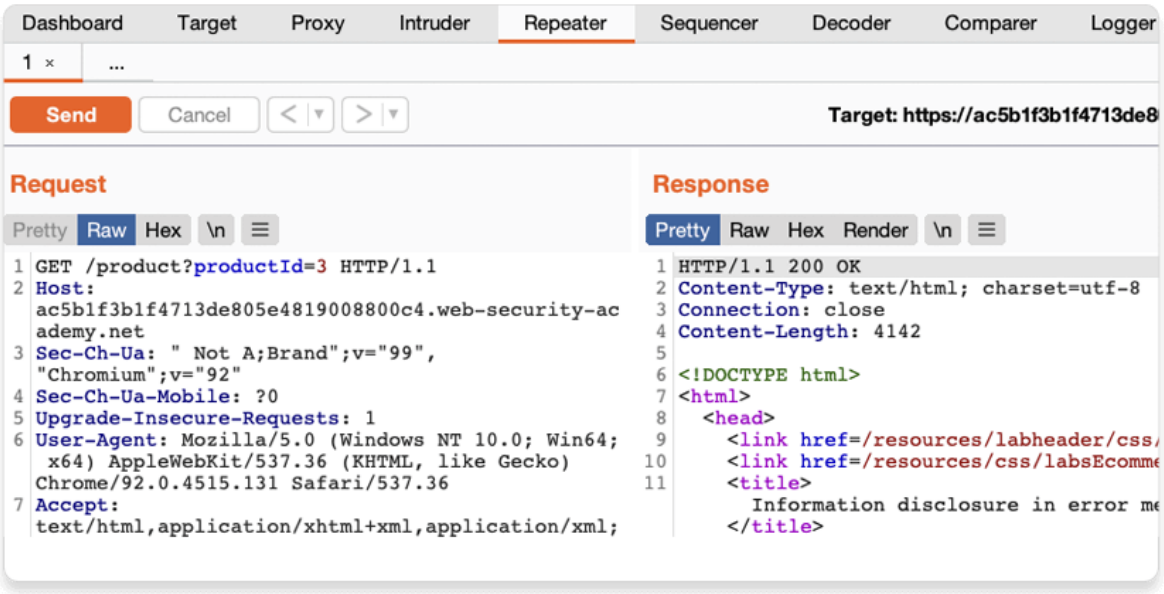
Right-click on any of the GET /product?productId=[...] requests and select **Send to Repeater**.



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

Step 5: 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.

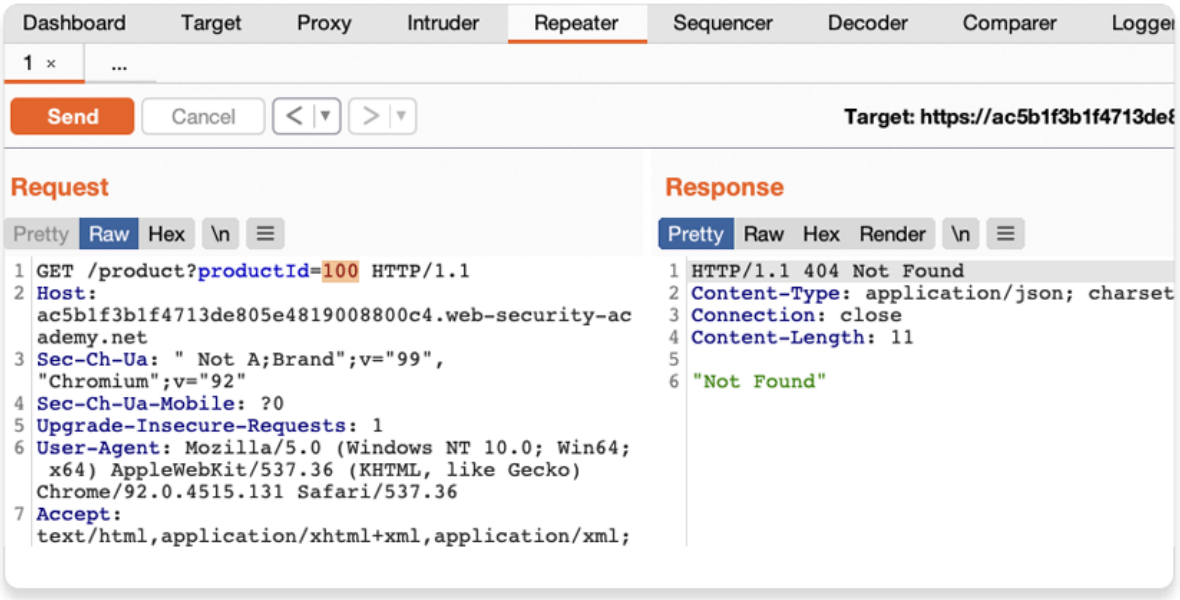


Testing different input with Burp Repeater

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.

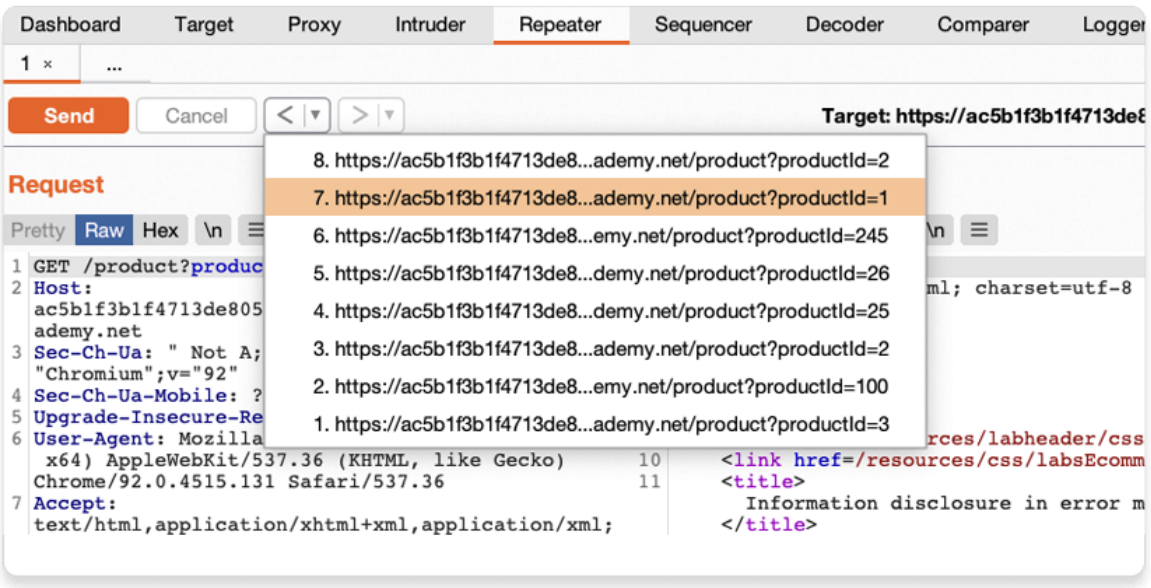
Step 1: Resend the request with different input

Change the number in the productId parameter and resend the request. Try this with a few arbitrary numbers, including a couple of larger ones.



Step 2: 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.

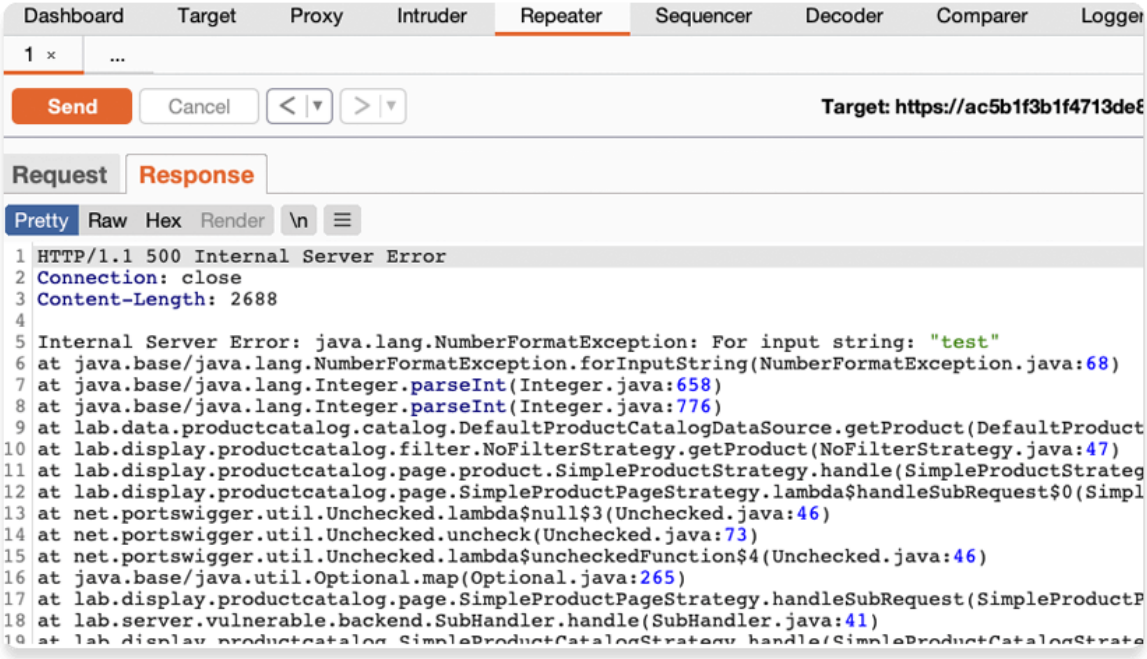
Step 3: Try sending unexpected input

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.

Step 4: 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.

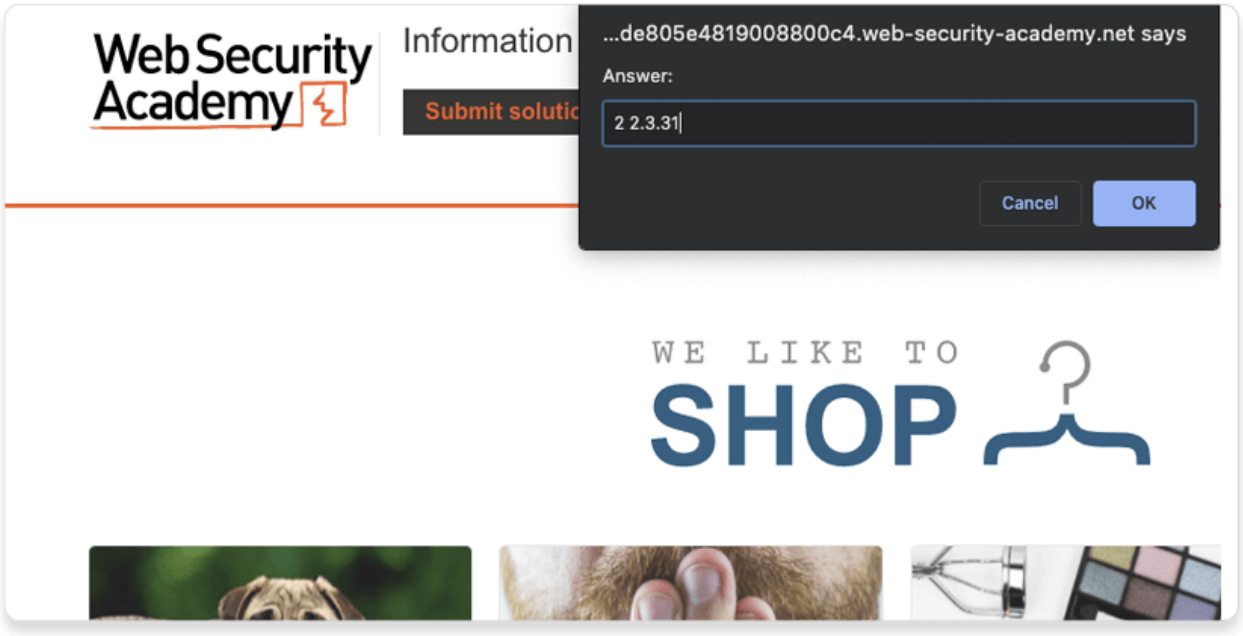


Notice that the response tells you that the website is using the Apache Struts framework - it even reveals which version.

```
37 at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:628)
38 at java.base/java.lang.Thread.run(Thread.java:835)
39
40 Apache Struts 2 2.3.31
```

In a real scenario, this kind of information could be useful to an attacker, especially if the named version is known to contain additional vulnerabilities.

Go back to the lab in Burp's browser and click the **Submit solution** button. Enter the Apache Struts version number that you discovered in the response (2 2.3.31).



Congratulations, that's another lab under your belt! You've used Burp Repeater to audit part of a website and successfully discovered an information disclosure vulnerability.