

# Web Application Security Assessment using Burp Suite

## Introduction:

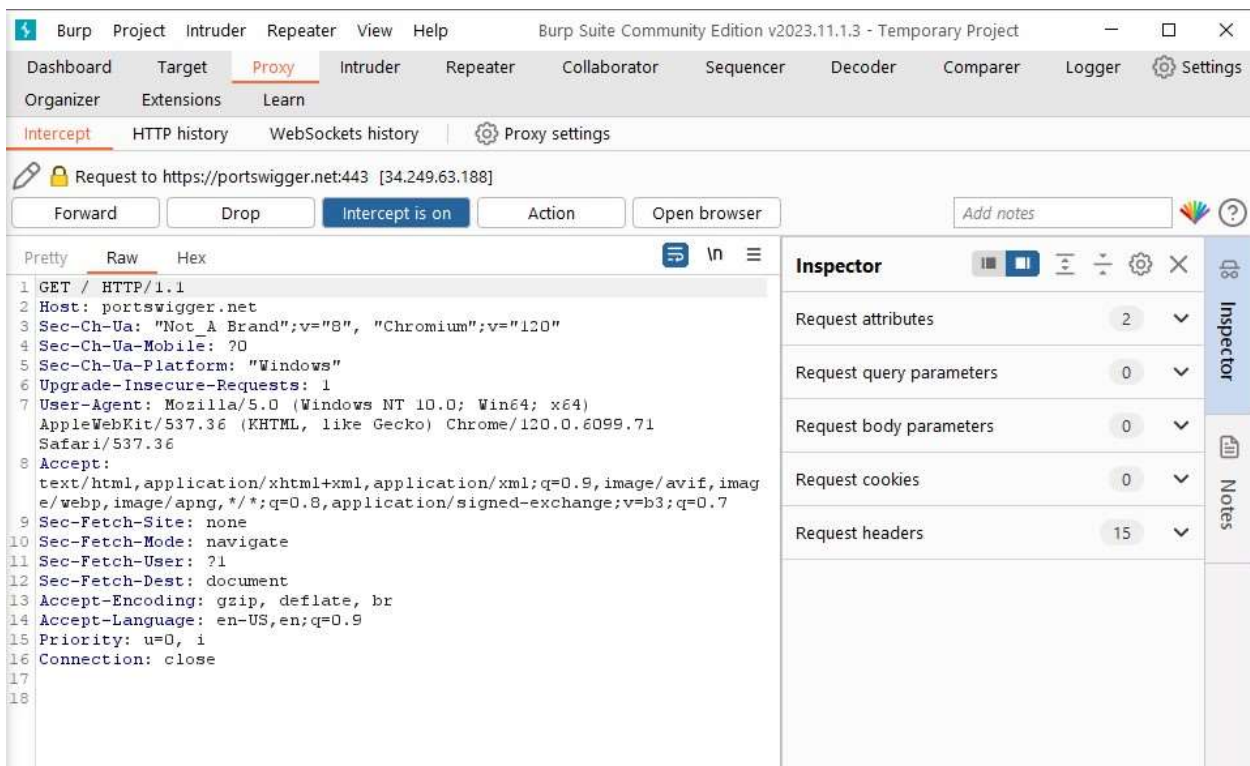
Burp Suite is a tool widely used in the cyber security field to identify and address security vulnerabilities in web applications. Developed by PortSwigger, Burp Suite provides a comprehensive platform providing a set of tools for identifying and addressing vulnerabilities, allowing professionals to proactively test and enhance the security posture of web applications. Some of these tools include a proxy, scanner, repeater and crawl to name a few. I will highlight these tools and more in this project as well as others.

## Methodology

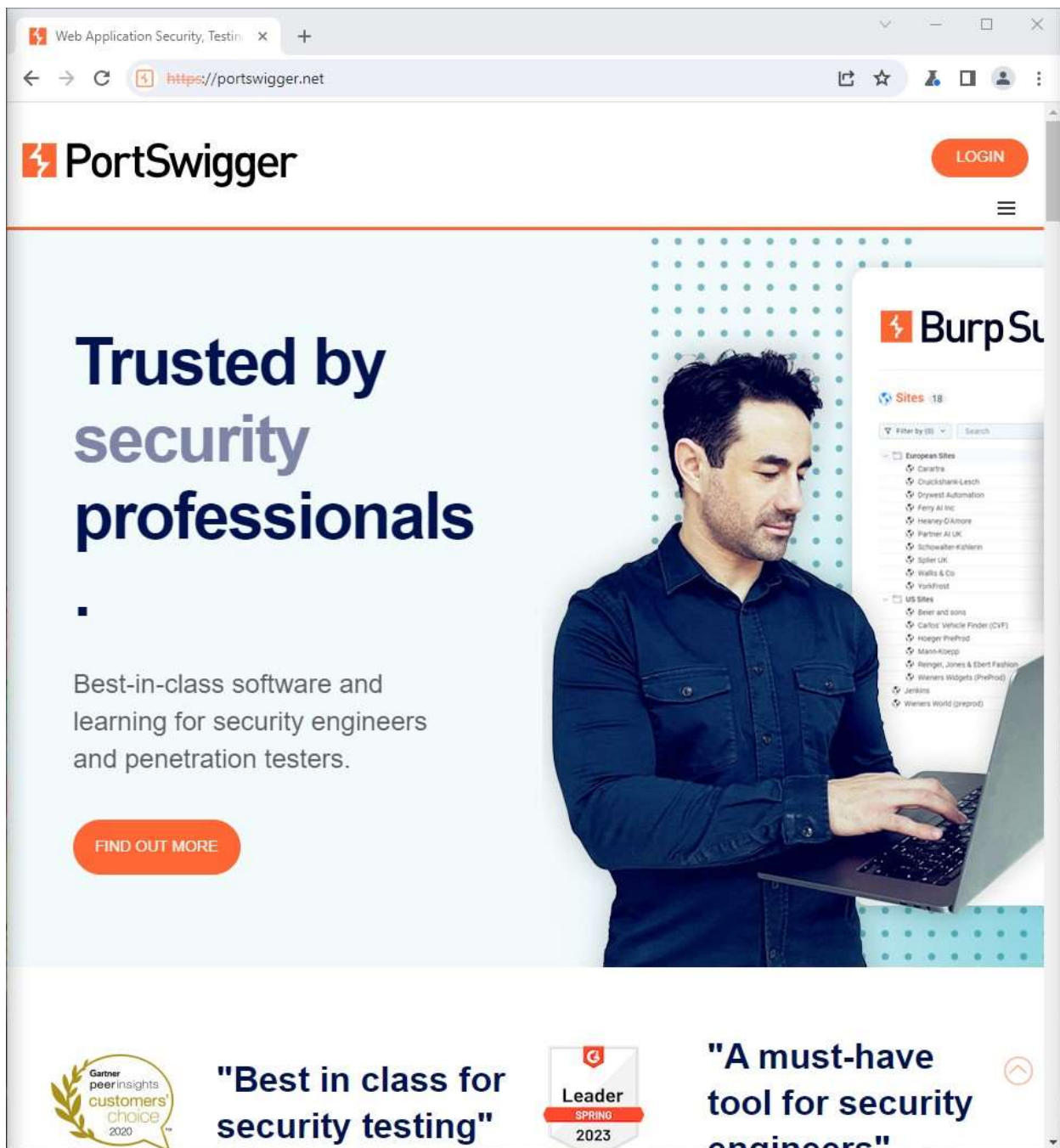
In this project I will be running Burp Suite using Burp Suite Community and Professional for scanning. For the target website that I will be using will be <https://portswigger.net>, and [ginandjuice.shop](https://ginandjuice.shop), which are vulnerable websites provided from Burp Suite. As listed in the introduction I will be using a list of tools for this project starting with the proxy, Burp Proxy acts as a link between the user's browser and the target web application. It allows users to intercept and modify HTTP(S) requests and responses. This feature is crucial for understanding how web applications work and identifying potential vulnerabilities. Doing activities like forwarding requests, viewing history, modifying request, and exploiting vulnerabilities. Following the Proxy is the Burp Repeater. The Repeater tool allows users to manually resend HTTP(S) requests to the server with modifications. It's useful for testing and exploiting vulnerabilities by manipulating parameters and observing the application's response. After using the Repeater I will run a vulnerability scan, you must be using Burp Suite Professional for this. If you need a license key portswigger gives a free trial where all you have to do is give you email address (note they do not accept Gmail accounts). The Scanner module automates the process of identifying security vulnerabilities in web applications. It uses a wide range of checks to find common issues such as SQL injection, cross-site scripting (XSS), and more. Users can customize scan configurations to suit their testing needs. While running the scan the tool Crawl will be in action as well. Which is used for automated crawling of web applications, helping to discover and map the structure of the application. It can follow links and identify new URLs, providing a comprehensive view of the target. And lastly you will generate your report.

## Procedure

First thing I am going to intercept request with Burp Proxy. An interception in this sense is a HTTP(S) request and responses sent between Burp's browser and the target server. To get started download and install community or professional. In these first actions performed it doesn't matter which is chosen, it will only matter later when generating a scan. Once installed launch the Burp's browser to do this go to the Proxy tab and click the Intercept tab. There you will click the "Intercept is off" button which will turn the intercept on, then click "open browser". Now we will intercept a request. Via the Burp's browser attempt to visits "<https://portswigger.net>". Notice in Burp Suite you were able to intercept the HTTP request which was issued by the browser before it could reach the server, example shown here:



This can be modified before forwarding it to the target server if you would like but in this case we are going to forward the request. To do so click forward to send the intercepted request, the page should load in Burp's browser if not click the forward button again until Burp's browser is populated. Once that happens the burp browser should look like this:



Now go back to the intercept tab and switch off interception. You don't want to keep interception on constantly because a large number of requests from a browser usually sends constantly while you navigate on a website and don't want to have intercepted all of them. Once intercept is off the burp browser should still be operating. Now view the HTTP history start by (in Burp Suite) go to the Proxy tab then go to the HTTP history tab. Here you can see all history that has passed through Burp (even while interception was switched off)(Appendix. A).

Click any entry in the history to view the raw HTTP request as well as the response from the server should look like this:

The screenshot displays the Burp Suite interface with the 'Request' and 'Response' tabs selected. The 'Request' tab shows a GET request to `/content/images/svg/icons/enterprise.svg` with various headers and cookies. The 'Response' tab shows an HTTP/2 200 OK response with headers indicating the content type is `image/svg+xml` and the server is `Kestrel`. Both panels have tabs for 'Pretty', 'Raw', and 'Hex' views. The bottom of the interface features navigation icons, a search bar, and a '0 highlights' indicator.

### Request

1 GET  
/content/images/svg/icons/enterprise.svg HTTP/2  
2 Host: portswigger.net  
3 Cookie: SessionId=CfDJ8ITOG3g5p3JKtNNyCP8Y3%2Fi15nvYNHg67K745e5S%2FMyAri2ut%2Fcqk78qNhtM5OvGSr170%2BvtBC2XqxHKJGqT18LwOOWnMLAlbzDGSZRh2PpGM2%2FjpC9XwHwceU6Y2AfhUk%2BuL4um2RK%2BFxw1XXP3KYf4W65f1ZgaURJe2ruIm6YQ;  
AWSALBAPP-0=\_remove\_; AWSALBAPP-1=\_remove\_; AWSALBAPP-2=\_remove\_; AWSALBAPP-3=\_remove\_  
4 Sec-Ch-Ua: "Not A Brand";v="8", "Chromium";v="120"  
5 Sec-Ch-Ua-Mobile: ?0  
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.71 Safari/537.36  
7 Sec-Ch-Ua-Platform: "Windows"  
8 Accept: image/avif,image/webp,image/apng,image/svg+xml,image/\*,\*/\*;q=0.8  
9 Sec-Fetch-Site: same-origin  
10 Sec-Fetch-Mode: no-cors  
11 Sec-Fetch-Dest: image  
12 Referer: https://portswigger.net/  
13 Accept-Encoding: gzip, deflate, br  
14 Accept-Language: en-US,en;q=0.9  
15 Priority: u=2, i  
16

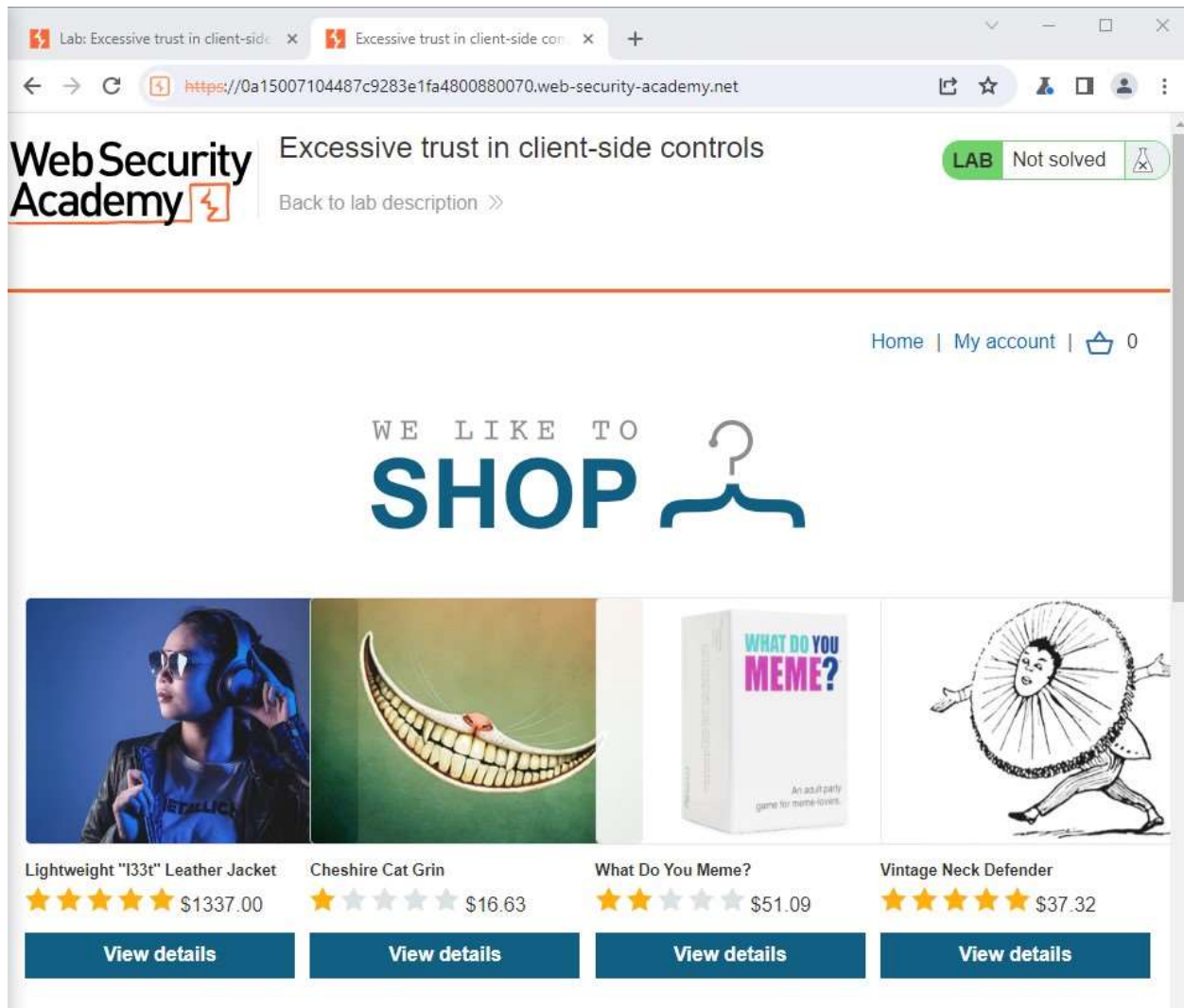
### Response

1 HTTP/2 200 OK  
2 Date: Fri, 05 Jan 2024 20:05:08 GMT  
3 Content-Type: image/svg+xml  
4 Content-Length: 554  
5 Server: Kestrel  
6 Accept-Ranges: bytes  
7 Cache-Control: must-revalidate, max-age=0  
8 Etag: "1da31b64eff142a"  
9 Last-Modified: Mon, 18 Dec 2023 13:30:04 GMT  
10 Strict-Transport-Security: max-age=31536000; preload  
11 X-Content-Type-Options: nosniff  
12 X-Frame-Options: SAMEORIGIN  
13 X-Xss-Protection: 1; mode=block  
14 Content-Security-Policy: default-src 'none';base-uri 'none';child-src 'self' https://www.youtube.com/embed/;connect-src 'self' https://ps.containers.piwik.pro https://ps.piwik.pro https://www.google.com/recaptcha/;font-src 'self';frame-src 'self' https://www.youtube.com/embed/ https://www.google.com/recaptcha/;img-src 'self' https://i.ytimg.com/;media-src 'self' https://d21v5rjx8s17cr.cloudfront.net/ https://d2g1lb374o3yzk.cloudfront.net/;script-src 'self'

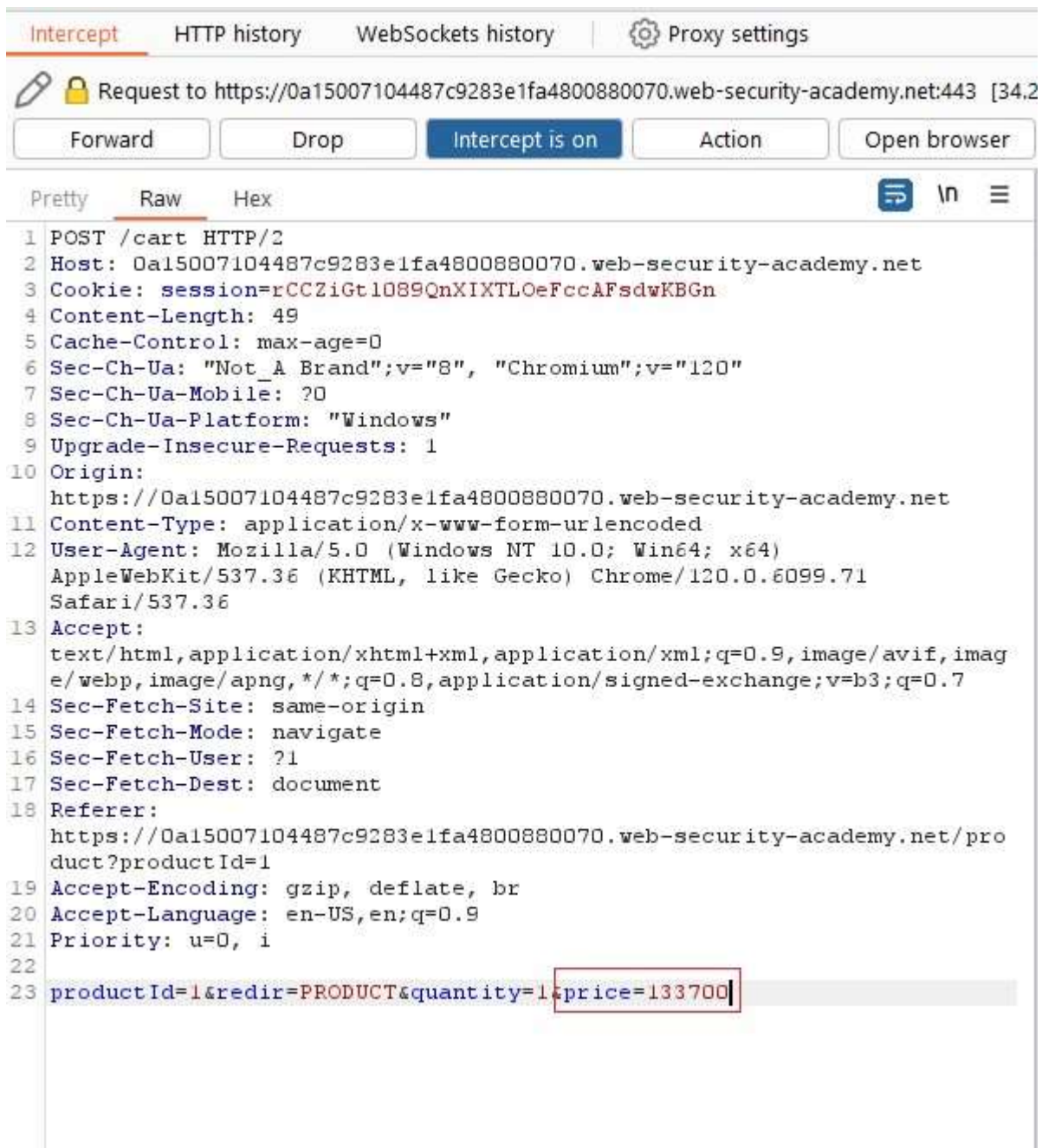
When cyber professionals do this they are looking to dig through website per usual and study the interactions between Burp's browser and the server afterward. Next I am going to modify an intercepted request in Burp Proxy, which enables you to manipulate request differently then the website expected to see how it responds. To do this, access the vuln website in Burp's Browser, make sure intercept is off in proxy and click intercept, launch the burp's browser and use it to visit "<https://portswigger.net/websecurity/logic-flaws/examples/lab-logic-flaws-excessive-trust->



in-client-side-controls". Once loaded click access the lab, then you will be sent to port swagger login page (create a account if necessary), sign in once that is done you will be sent to instance of fake shopping site like the one shown here:



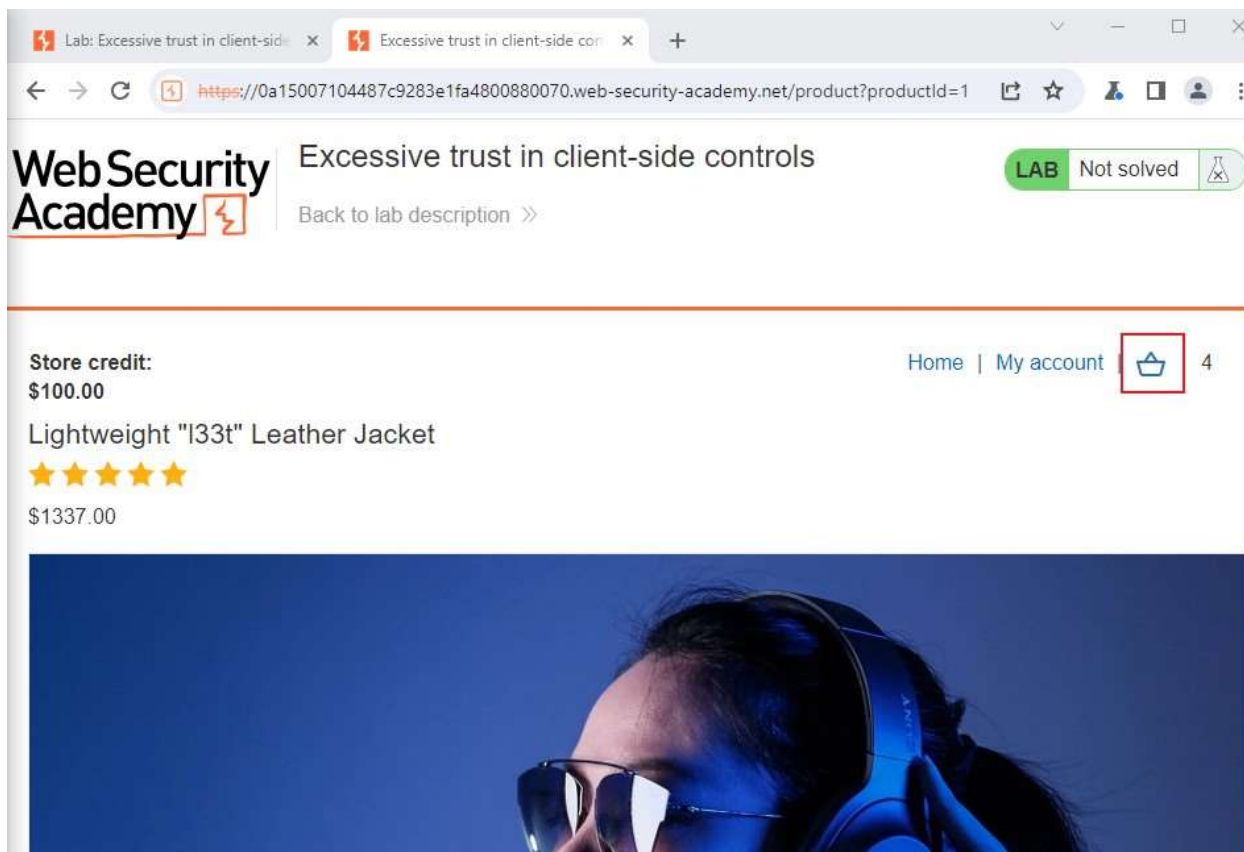
Now login to shopping account go to “my account” and use the credentials, username: wiener and password: peter then find something to buy. Notice there is currently have \$100 in store credit. Now click home to go back to the home page. Now click “view details” for the “lightweight “I33t” Leather Jacket”. Now Study the add to cart function to do this, in Burp go to the Proxy tab followed by the Intercept tab and switch intercept to on. Now moving in the Burp’s Browser add the jacket the cart. When this is done there should receive POST /cart request in Burp. Notice there is a parameter in the body called price, which is for the price of the item in cents shown here:



Now we to modify the request, change the value for the price parameter to 1 then click forward to send the request to the server (Appendix. B)

Then proceed the turn the intercept off again. Since we have a vulnerability it's time to exploit it.

In burps browser click the based icon to view cart.

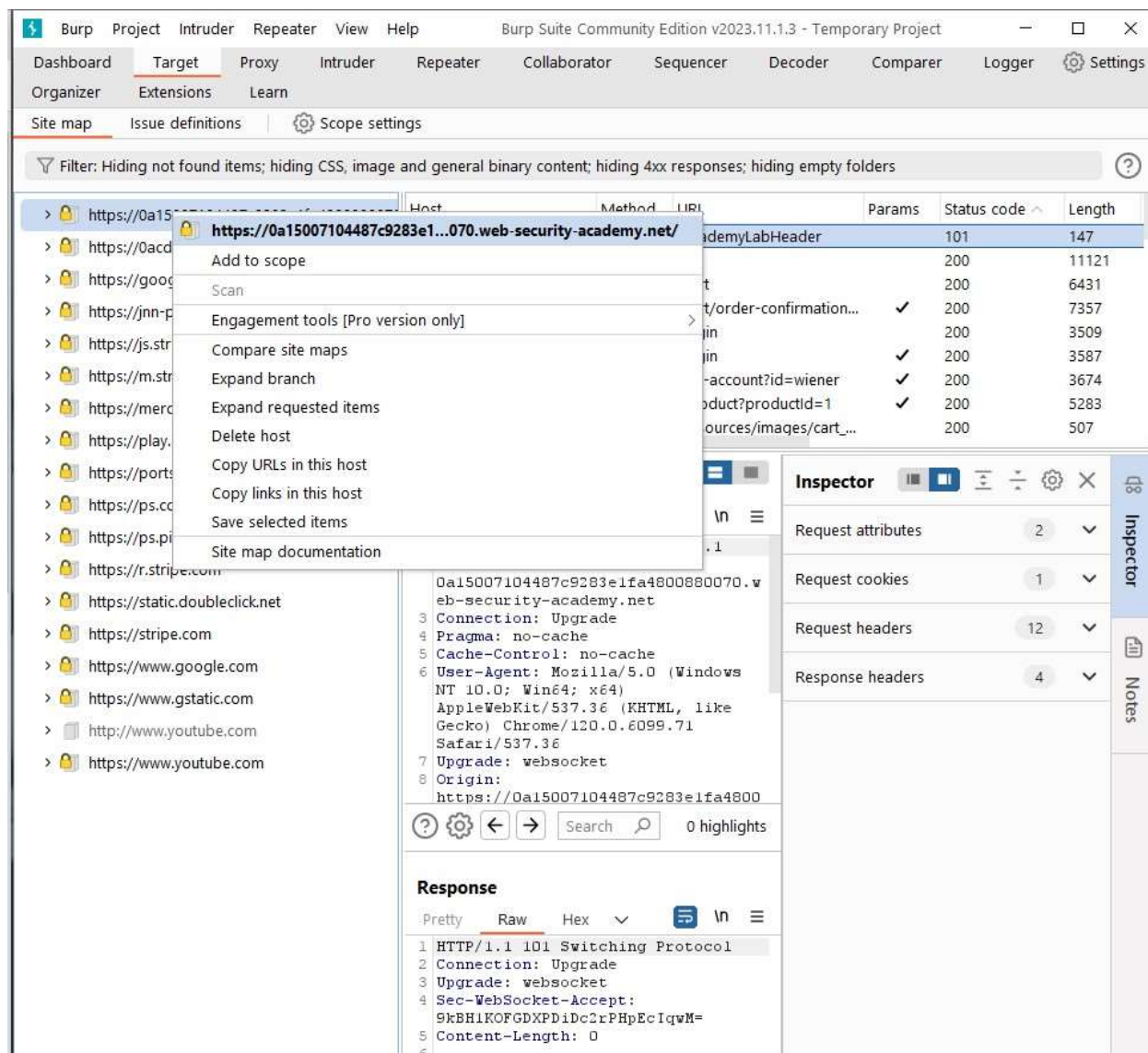


Notice the jacket was added for 1 cent.

Name	Price	Quantity
Lightweight "I33t" Leather Jacket	\$0.01	<div>- 4 +</div> <div>Remove</div>

Moving forward it's time to set the target scope the objective of setting a target scope is to work in Burp Suite. The target scope tells Burp which URL and host wants to be tested, it enables us to filter out noise generated by the browser and other sites to focus on the traffic we want. To do this, launch burp's browser and use it to visit ["https://portswigger.net/web-security/information-disclosure/exploiting/lab-infoleak-in-error-messages"](https://portswigger.net/web-security/information-disclosure/exploiting/lab-infoleak-in-error-messages), click access the lab and browse the target site, do some shopping and click on a couple of the product pages. Now to study the HTTP history in Burp go to the proxy tab then go to the HTTP history tab, to make it easier to read click the entry '#' to re-order the requests to go to descending order (see most recent at the top). Notice that the HTTP history shows details

about each request made including requests to thirdparty websites that you're not interested in (Google Analytics / YouTube). Now to set the target scope, head to the target tab then to site map tab, where you can see on the left side you can find the panel to see the list of hosts that the browser has interacted with. To add scope right click the node for the target site and click add to scope and proceed to click yes to exclude out-of-scope traffic.



Now to filter HTTP history, this will cut time down drastically and makes it easier to evaluate the entries. Now head back to the proxy tab then to HTTP history. Click on the display filter above the HTTP history and select "Show only in-scope items"



Filter settings: Hiding out of scope items; hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Ext
443	https://0a15007104487c9283e1f...	GET	/resources/images/cart_blue.svg			200	507	XML	svg
442	https://0a15007104487c9283e1f...	GET	/resources/images/shop.svg			200	7258	XML	svg
439	https://0a15007104487c9283e1f...	GET	/resources/labheader/js/labHeader.js			200	987	script	js
435	https://0a15007104487c9283e1f...	GET	/			200	11121	HTML	
285	https://0a15007104487c9283e1f...	GET	/product?productId=1	✓		200	5283	HTML	
283	https://0a15007104487c9283e1f...	GET	/product?productId=1	✓		200	5283	HTML	
282	https://0a15007104487c9283e1f...	GET	/			200	11121	HTML	

**Configure filter**

Settings mode | Bambda mode

**Filter by request type**

- ☒ Show only in-scope items
- ☐ Hide items without responses
- ☐ Show only parameterized requests

**Filter by MIME type**

- ☒ HTML
- ☒ Script
- ☒ XML
- ☐ CSS
- ☒ Other text
- ☐ Images
- ☒ Flash
- ☐ Other binary

**Filter by status code**

- ☒ 2xx [success]
- ☒ 3xx [redirection]
- ☒ 4xx [request error]
- ☒ 5xx [server error]

**Filter by search term**

☐ Regex

☐ Case sensitive ☐ Negative search

**Filter by file extension**

☐ Show only: asp,aspx,jsp,php

☐ Hide: js,gif,jpg,png,css

**Filter by annotation**

☐ Show only commented items

☐ Show only highlighted items

**Filter by listener**

Port

Show all | Hide all | Revert changes | Convert to Bambda | Cancel | Apply

Notice that it now only shows entries from the website that were targeted, all other entries have been taken out of the result list. Doing this improves time and efficiency due to only showing the entries that you are interested in.

The screenshot displays the Burp Suite interface. The top menu bar includes Burp, Project, Intruder, Repeater, View, and Help. The main toolbar shows various tools like Dashboard, Target, Proxy, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, and Settings. The 'Proxy' tab is active, and the 'HTTP history' sub-tab is selected. A filter settings bar indicates 'Hiding out of scope items; hiding CSS, image and general binary content'. Below this is a table of HTTP history entries. A red box highlights a specific entry (ID 285) with the URL '/product?productId=1'. Below the table, the 'Request' and 'Response' panels are visible. The 'Request' panel shows a GET request to '/content/images/svg/icons/enterprise.svg'. The 'Response' panel shows an HTTP/2 200 OK response with various headers including Date, Content-Type, Server, and Cache-Control. The 'Inspector' panel on the right shows the request and response details.

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Ext
443	https://0a15007104487c9283e1f...	GET	/resources/images/cart_blue.svg			200	507	XML	svg
442	https://0a15007104487c9283e1f...	GET	/resources/images/shop.svg			200	7258	XML	svg
439	https://0a15007104487c9283e1f...	GET	/resources/labheader/js/labHeader.js			200	987	script	js
435	https://0a15007104487c9283e1f...	GET	/			200	11121	HTML	
285	https://0a15007104487c9283e1f...	GET	/product?productId=1		✓	200	5283	HTML	
283	https://0a15007104487c9283e1f...	GET	/product?productId=1		✓	200	5283	HTML	
282	https://0a15007104487c9283e1f...	GET	/			200	11121	HTML	
281	https://0a15007104487c9283e1f...	GET	/			200	11121	HTML	
280	https://0a15007104487c9283e1f...	GET	/product?productId=1		✓	200	5283	HTML	
279	https://0a15007104487c9283e1f...	POST	/cart		✓	302	100		
278	https://0a15007104487c9283e1f...	POST	/cart		✓	302	100		

**Request**

```

1 GET
2 /content/images/svg/icons/enterprise.svg HTTP/2
3 Host: portswigger.net
4 Cookie: SessionId=CfDj8ITOG3gSp3JKtNNyCP8Y3%2F115nvYNHg67K745e5S%2FMyAr12ut%2F0qk78qNhtM5OvGSr170%2BvtBC2XqxHKJGqT18Lw0OWnMLAlbzDGS2Rh2PpGM2%2FjpC9XwHwceU6Y2AfHuk%2BuL4um2RK%2BFxw1XXP3KYt4W65f1ZgaURJe2ruIm6YQ; AWSALBAPP-0=_remove_; AWSALBAPP-1=_remove_; AWSALBAPP-2=_remove_; AWSALBAPP-3=_remove_
5 Sec-Ch-Ua: "Not A Brand";v="8", "Chromium";v="100"
6 Sec-Ch-Ua-Mobile: ?0
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)

```

**Response**

```

1 HTTP/2 200 OK
2 Date: Fri, 05 Jan 2024 20:05:08 GMT
3 Content-Type: image/svg+xml
4 Content-Length: 554
5 Server: Kestrel
6 Accept-Ranges: bytes
7 Cache-Control: must-revalidate, max-age=0
8 Etag: "1da31b64eff142a"
9 Last-Modified: Mon, 18 Dec 2023 13:30:04 GMT
10 Strict-Transport-Security: max-age=31536000; preload
11 X-Content-Type-Options: nosniff
12 X-Frame-Options: SAMEORIGIN
13 X-Xss-Protection: 1; mode=block
14 Content-Security-Policy: default-src 'self';

```

**Inspector**

- Request attributes: 2
- Request cookies: 5
- Request headers: 21
- Response headers: 20

A key component in Burp is the Burp Repeater which lets you reissue requests over and over again, which will help evaluating results in which the target website's output is in response to different inputs without having to intercept the request each time which will make it easier to probe for vulnerabilities or confirm once's that were identified by Burp Scanner. To do this first, identify a request. Last tutorial, noticed each time accessing a product page there were GET /product request with a productId query parameter:

Burp Suite Community Edition v2023.11.1.3 - Temporary Project

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Settings

Intercept HTTP history WebSockets history Proxy settings

Filter settings: Hiding out of scope items; hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Ext
443	https://0a15007104487c9283e1f...	GET	/resources/images/cart_blue.svg			200	507	XML	svg
442	https://0a15007104487c9283e1f...	GET	/resources/images/shop.svg			200	7258	XML	svg
439	https://0a15007104487c9283e1f...	GET	/resources/labheader/js/labHeader.js			200	987	script	js
435	https://0a15007104487c9283e1f...	GET	/			200	11121	HTML	
285	https://0a15007104487c9283e1f...	GET	/product?productId=1		✓	200	5283	HTML	
283	https://0a15007104487c9283e1f...	GET	/product?productId=1		✓	200	5283	HTML	
282	https://0a15007104487c9283e1f...	GET	/			200	11121	HTML	
281	https://0a15007104487c9283e1f...	GET	/			200	11121	HTML	
280	https://0a15007104487c9283e1f...	GET	/product?productId=1		✓	200	5283	HTML	
279	https://0a15007104487c9283e1f...	POST	/cart		✓	302	100		
278	https://0a15007104487c9283e1f...	POST	/cart		✓	302	100		

**Request**

```

1 GET
/content/images/svg/icons/enterprise.svg HTTP/2
Host: portswigger.net
Cookie: SessionId=CfDJ8ITOG3g5p3JKtNNyCP8Y3%2Fi15nvYNHge7K745e5S%2FMyAri2ut%2Fcqk78qNhtM5OvGSr170%2BvtBC2XqxHKJGqT18LwOOwNMLAlbzDGS2Rh2PpGM2%2Fjpc9XwHwceU6Y2AfhuK%2BuL4um2RK%2BFxw1XXP3KYf4W65f12gaURJe2ruIm6YQ; AWSALBAPP-0=remove; AWSALBAPP-1=remove; AWSALBAPP-2=remove; AWSALBAPP-3=remove
Sec-Ch-Ua: "Not A Brand";v="8", "Chromium";v="120"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)

```

**Response**

```

1 HTTP/2 200 OK
2 Date: Fri, 05 Jan 2024 20:05:08 GMT
3 Content-Type: image/svg+xml
4 Content-Length: 554
5 Server: Kestrel
6 Accept-Ranges: bytes
7 Cache-Control: must-revalidate, max-age=0
8 Etag: "1da31b64eff142a"
9 Last-Modified: Mon, 18 Dec 2023 13:30:04 GMT
10 Strict-Transport-Security: max-age=31536000; preload
11 X-Content-Type-Options: nosniff
12 X-Frame-Options: SAMEORIGIN
13 X-Xss-Protection: 1; mode=block
14 Content-Security-Policy: default-src 'self'

```

**Inspector**


Request attributes 2

Request cookies 5

Request headers 21


Response headers 20






Send the request to Burp Repeater, right click on GET /product?productId=[...] request and select "Send to Repeater" (Appendix. C.). Head to the repeater tab, notice there's a request entry.


 Burp Project Intruder Repeater View Help
 Burp Suite Community Edition v2023

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer  
 Organizer Extensions Learn

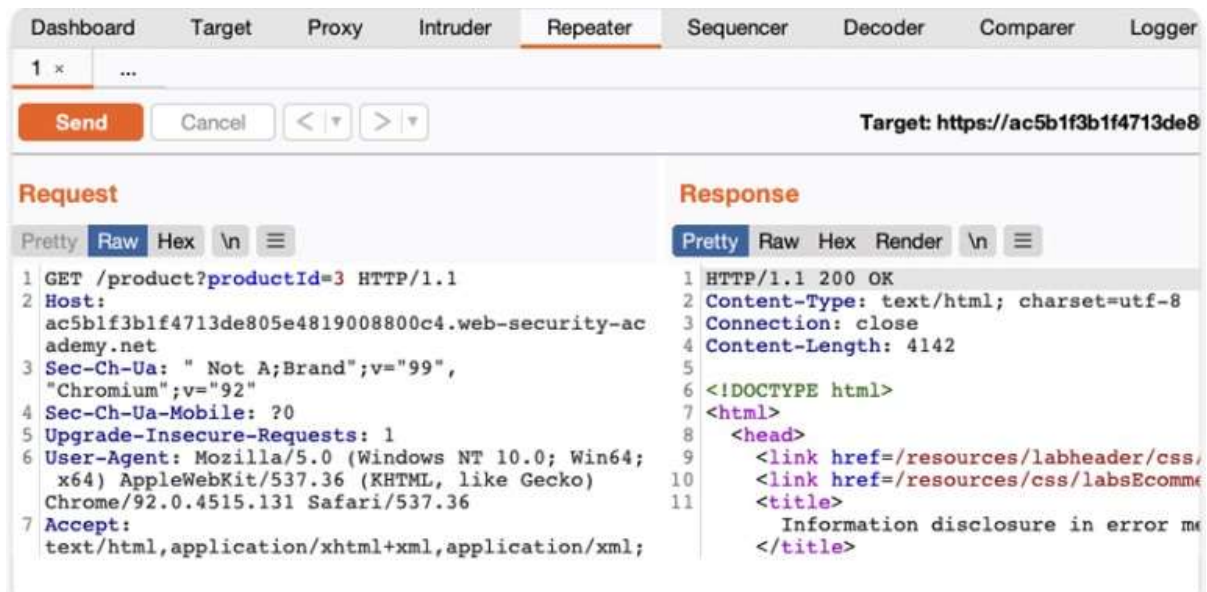
1 x +

Send

Cancel
< ▾
> ▾
Target: https://0a15007104487c9283e1fa480080070.web-security-academy.net

Request	Response
<div style="display: flex; justify-content: space-between; align-items: center; border-bottom: 1px solid #ccc; padding-bottom: 5px;"> <span>Pretty <u>Raw</u> Hex</span> <span>    </span> </div> <pre> 1 GET /product?productId=1 HTTP/2 2 Host:   0a15007104487c9283e1fa480080070.web-security-academy.net 3 Cookie: session=   rCCZiGt1089QnXIXTL0eFccAFsdwKBGn 4 Sec-Ch-Ua: "Not_A_Brand";v="8",   "Chromium";v="120" 5 Sec-Ch-Ua-Mobile: ?0 6 Sec-Ch-Ua-Platform: "Windows" 7 Upgrade-Insecure-Requests: 1 8 User-Agent: Mozilla/5.0 (Windows NT   10.0; Win64; x64)   AppleWebKit/537.36 (KHTML, like   Gecko) Chrome/120.0.6099.71   Safari/537.36 9 Accept:   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image   /webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 10 Sec-Fetch-Site: same-origin 11 Sec-Fetch-Mode: navigate 12 Sec-Fetch-Dest: document 13 Referer:   https://0a15007104487c9283e1fa480080070.web-security-academy.net/ 14 Accept-Encoding: gzip, deflate, br 15 Accept-Language: en-US,en;q=0.9 16 Priority: u=0, i 17 18                     </pre>	<div style="display: flex; justify-content: space-between; align-items: center; border-bottom: 1px solid #ccc; padding-bottom: 5px;"> <span>Pretty <u>Raw</u> Hex ▾</span> <span>   </span> </div>

Send the request and view the response, click send and view the response from the server:

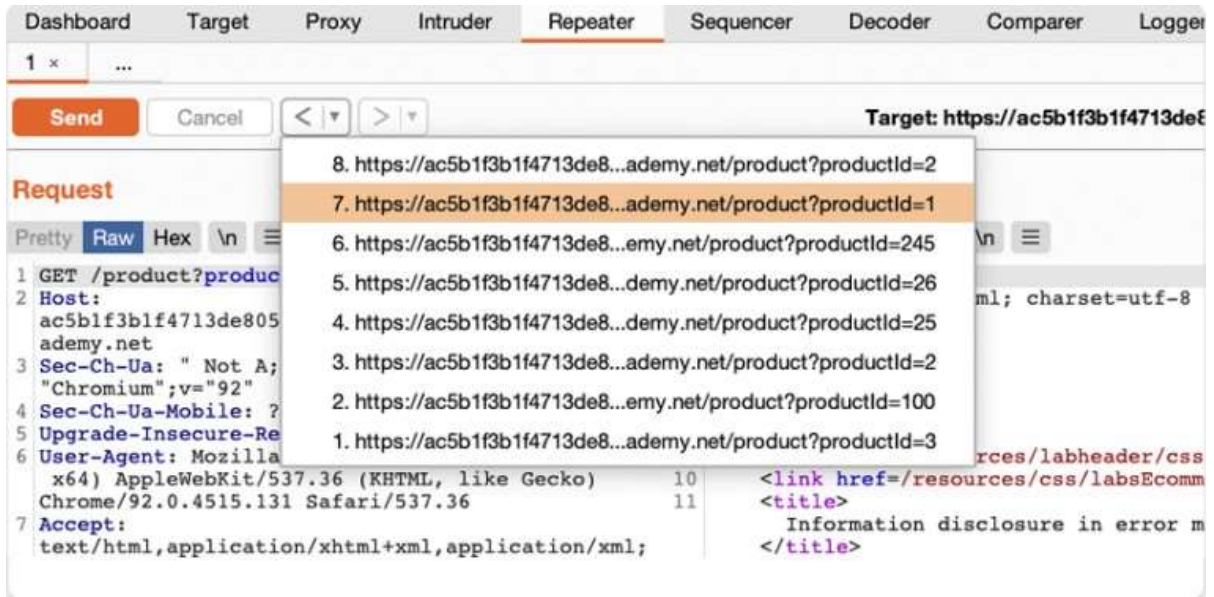




Now let's Test different input with Burp Repeater by resending the same request with different input, you will be able to identify and confirm different input-based vulnerabilities. Let's resend the request with different input, change the number in the productId parameter and resend the request :



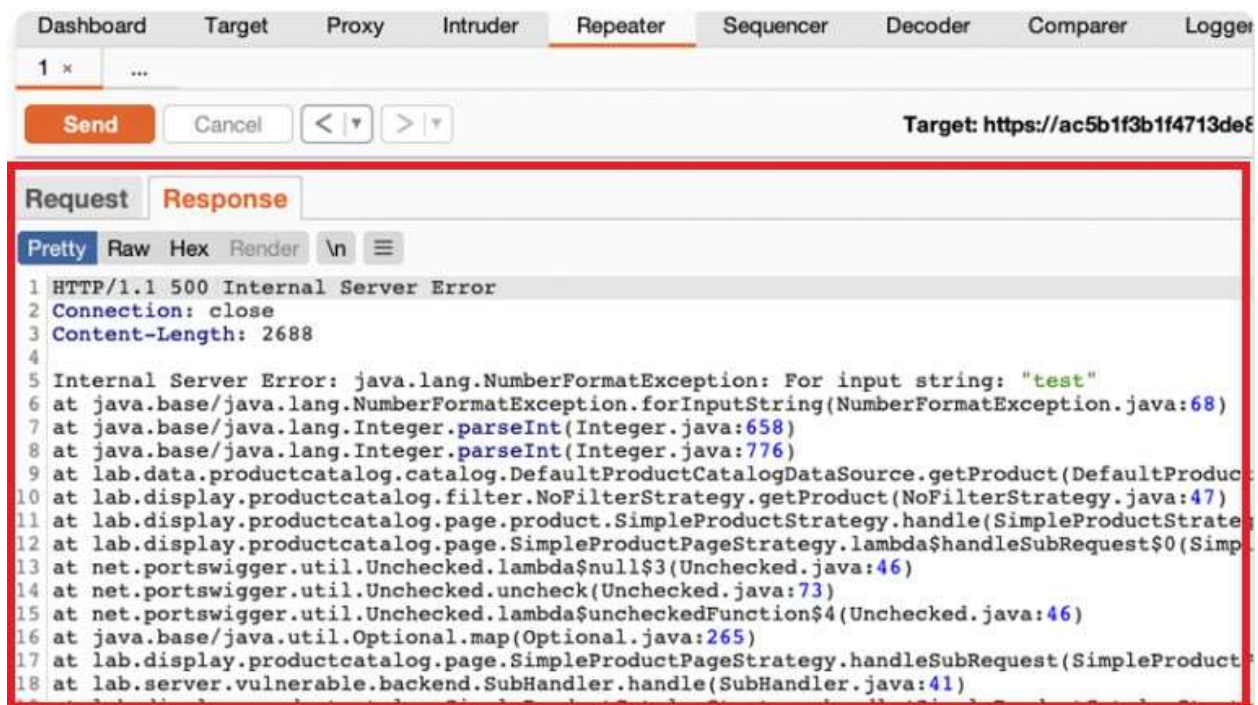
View the request history, use the arrows to move to previous request sent along with matching responses. The drop down menu next to each arrows also let you move to a specific request via history:



Notice in the responses, you can request different product pages by entering their ID, but you'll receive a Not Found response if the server cannot find that item with the ID number given. Let's try sending unexpected input. Notice how productId expects an int, try adding a string to what happens:



Notice an exception is caused, which an error response is issued:



The exception also shows website is using Apache Struts framework (shows its version):

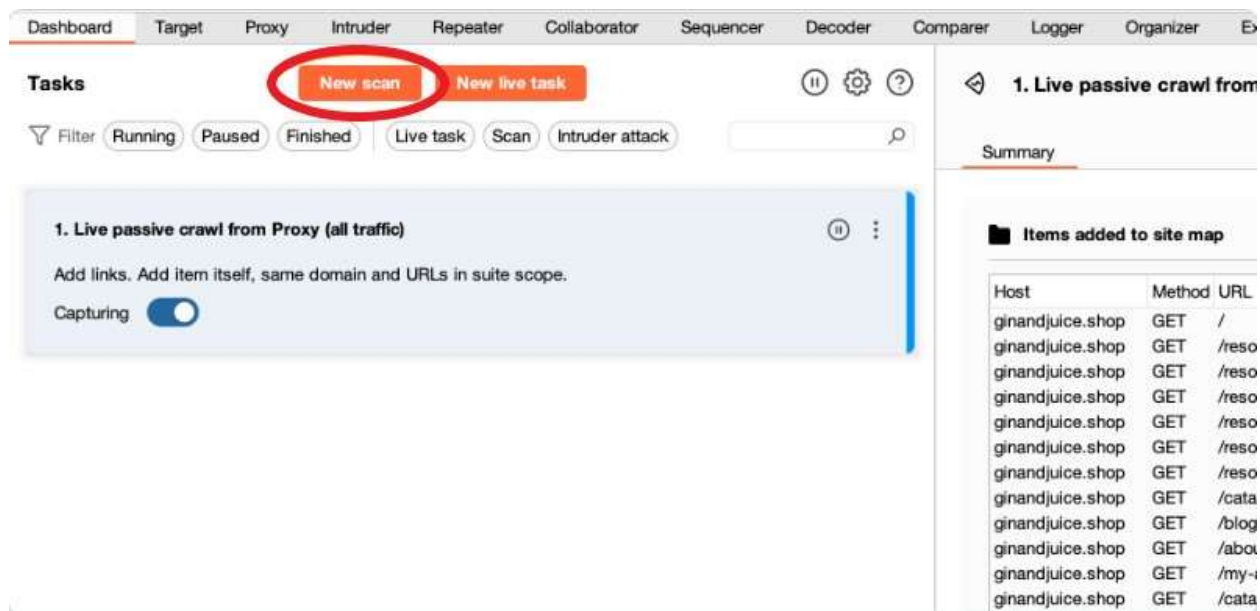
```
37 at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:628)
38 at java.base/java.lang.Thread.run(Thread.java:835)
39
40 Apache Struts 2 2.3.31
```

This info can be exploited by attackers with the version number the attack can look up exploits to run against the website to gain information. Let's head back to the Burp's browser and click on the Submit solution button and enter the Apache Struts version number found in the response:

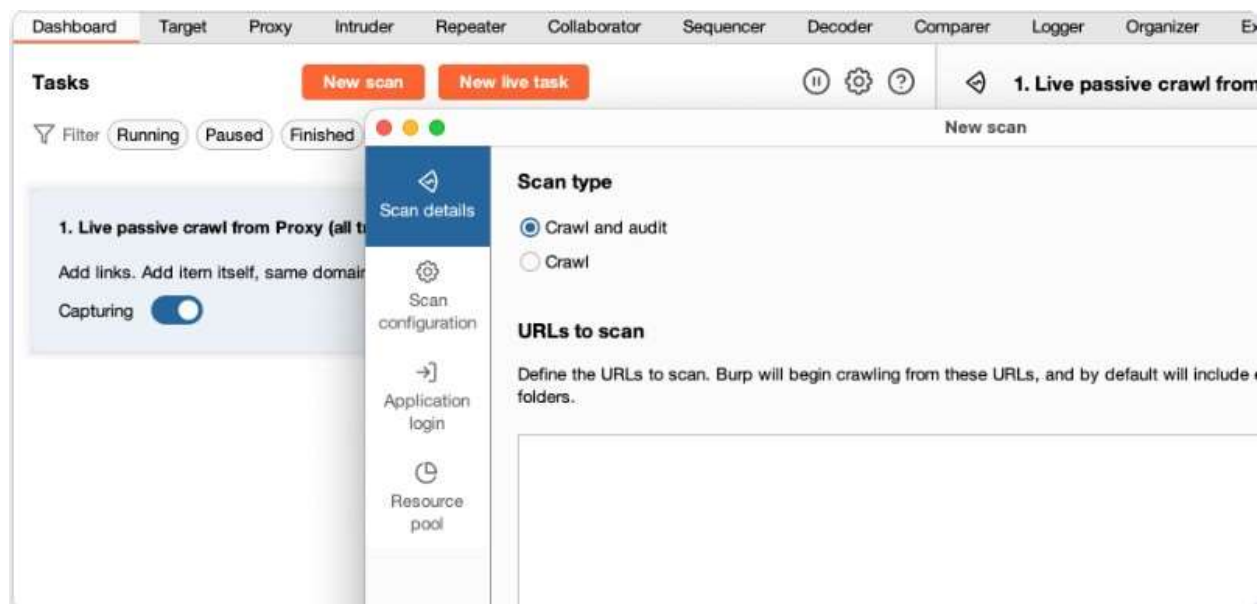


Now we are going to do vulnerability scanning, you must be running Burp Suite Pro to do this. Start by opening the scan launcher, go to the dashboard tab then to new scan:

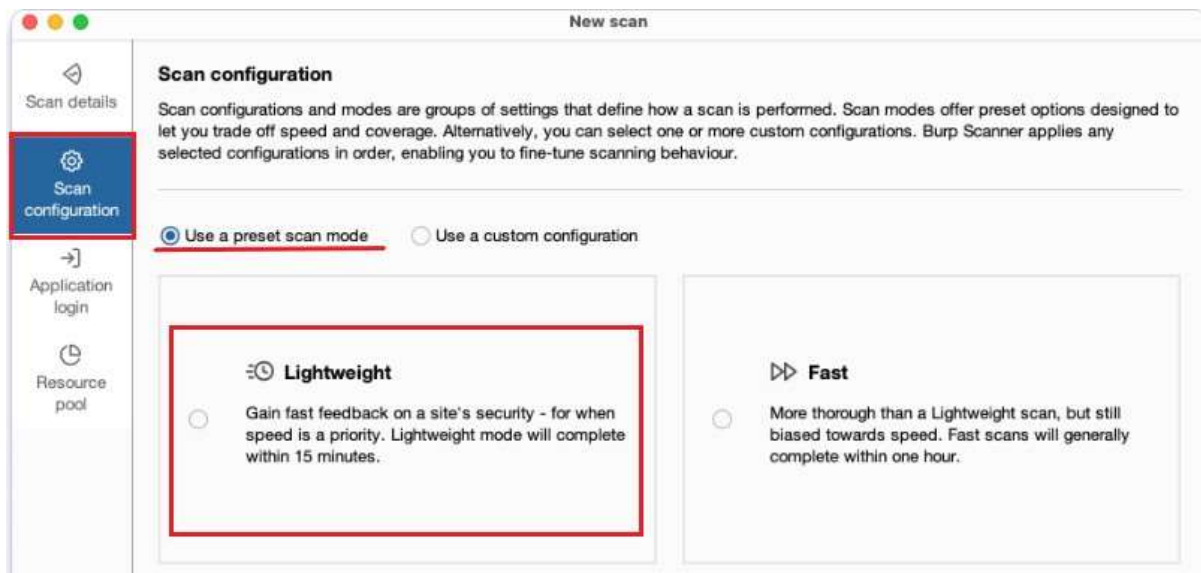




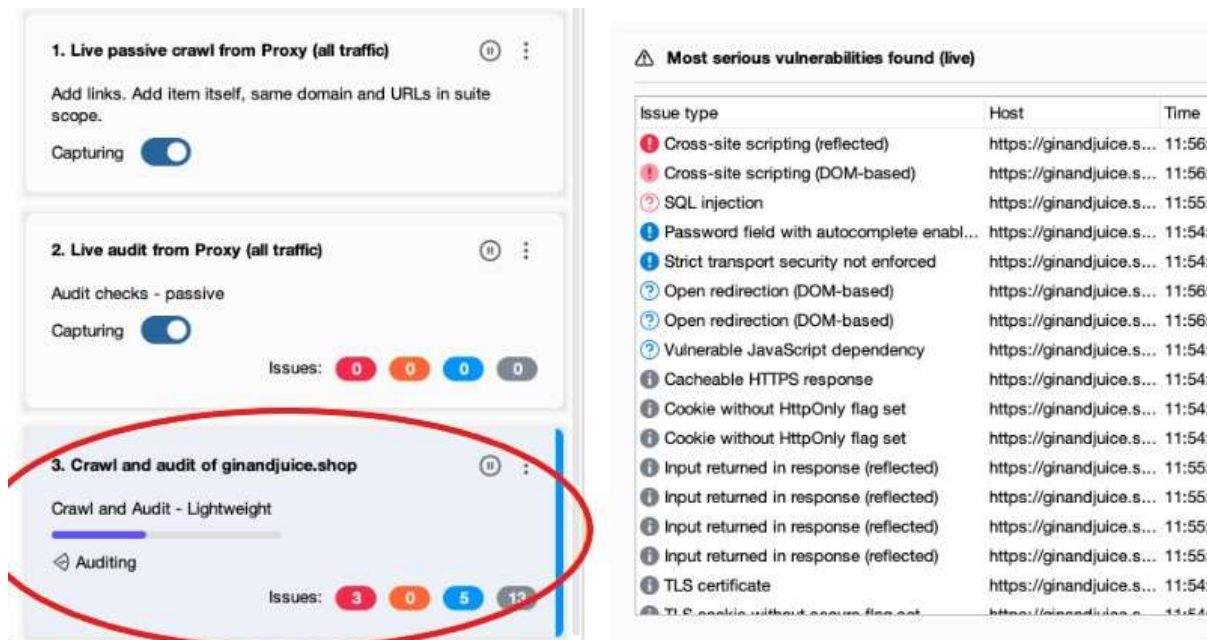
This is followed by the scan launch dialog box opening, this is where the configurations are made for the scans, enter the URL of the target site. In the “URLs to scan” paste in the url ginandjuice.shop



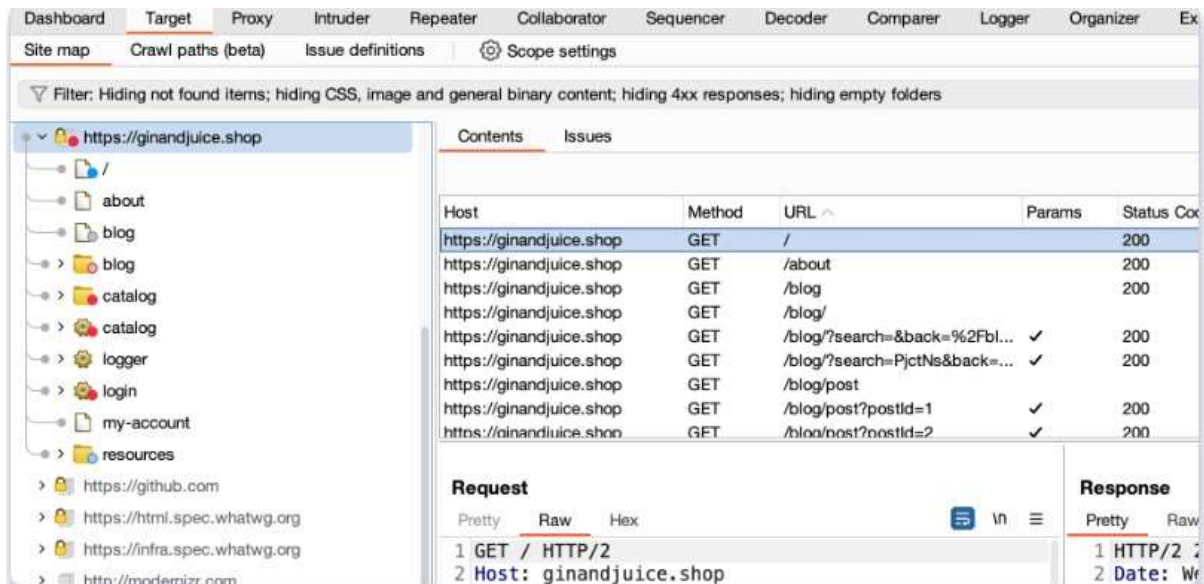
To configure the scan, click Scan Configuration, this is where you can specify what actions you want the Burp Scanner. Make sure that “Use a preset scan mode is selected and click “lightweight”, this scan is for getting a high-level overview of a target as quickly as possible (max time is 15 min):



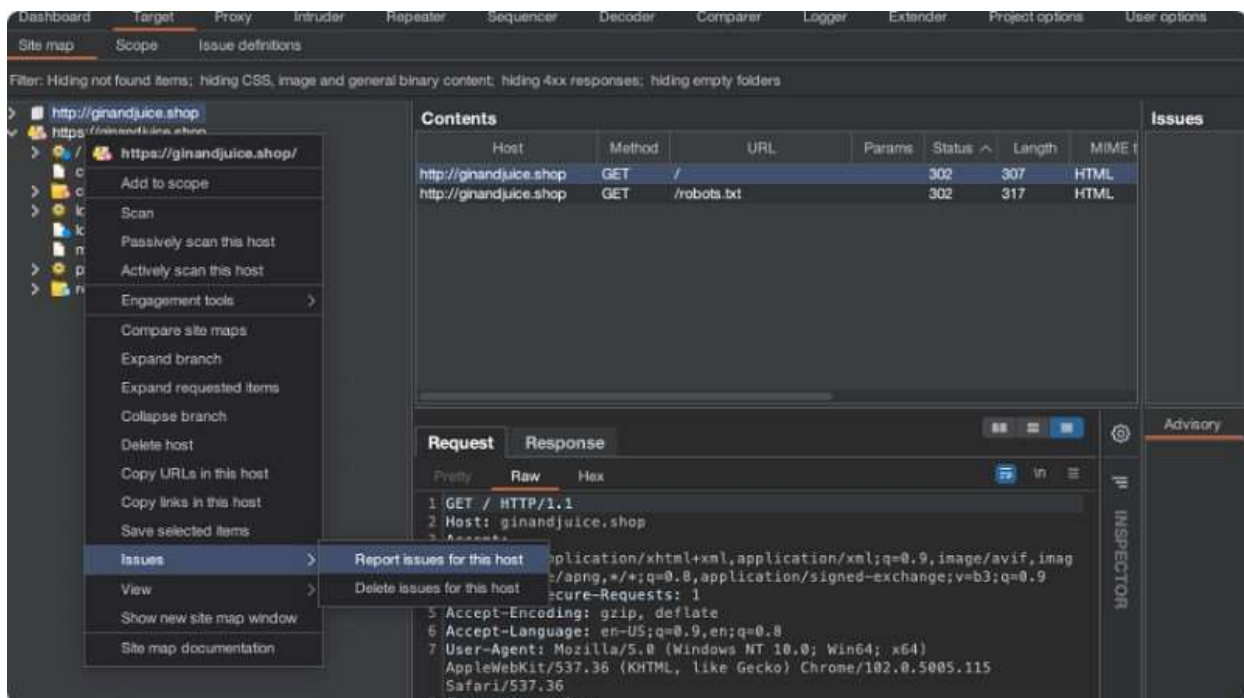
Now launch the scan, click ok, scans will start. The task you added will be added to the Dashboard (if you want to see more info can click on the task):



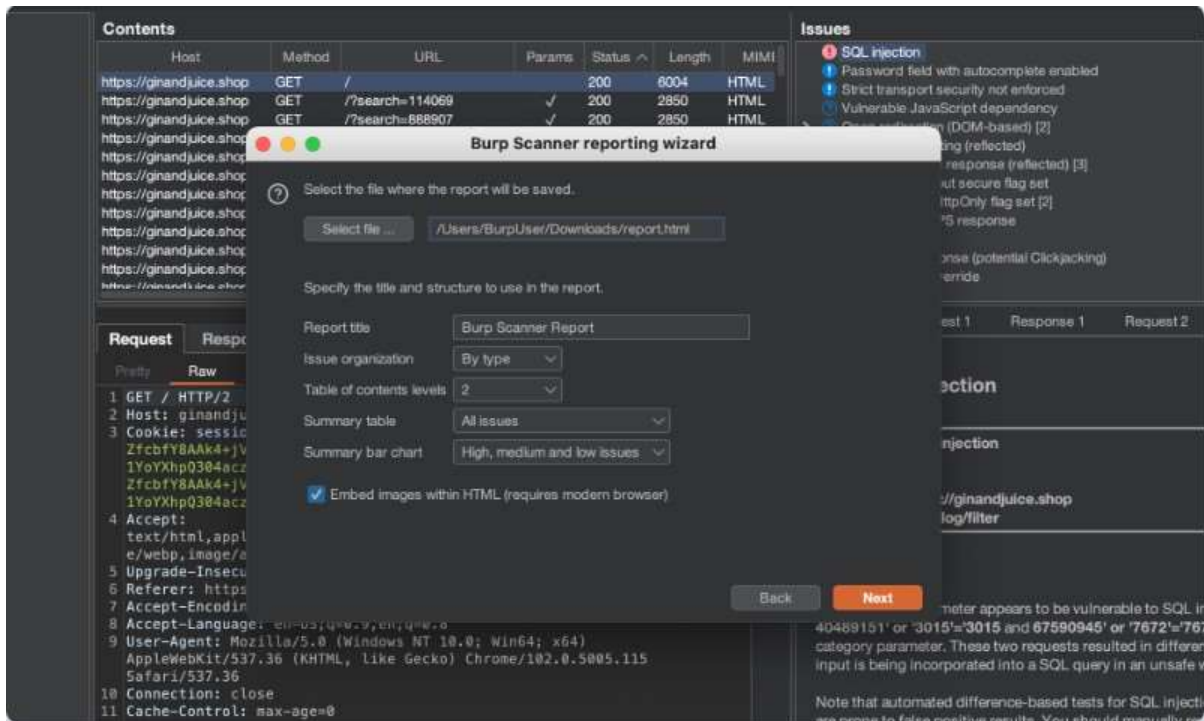
While the scans are running the crawl is in action. To see this go to the target then the site map tab. Expand the node for "ginandjuice.shop" there you can see all the contents the crawl has discovered to this point:



Moving to the identified issues, you can monitor status of scans (Issues → select scan from Tasks), once crawl is finished the scanner will start to audit for vulnerabilities. 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. Now to generate a report. Select relevant issue, go to "target" then on "site map" right-click on the entry for "https://ginandjuice.shop", click issues then report issues for this host:



To configure the report options, once it is selected a “Burp Scanner reporting wizard” pops up, click next (defaults until you’re prompted to enter a filename and location for the report):



Choose where you want to save the file and then click save and then next. Then view and share report, an example is shown here:



## Summary

The table below shows the numbers of issues identified in different categories. Issues are classified according to severity as High, Medium, Low or Information. This reflects the likely impact of each issue for a typical organization. Issues are also classified according to confidence as Certain, Firm or Tentative. This reflects the inherent reliability of the technique that was used to identify the issue.

		Confidence			
		Certain	Firm	Tentative	Total
Severity	High	3	4	0	7
	Medium	0	0	0	0
	Low	1	1	2	4
	Information	17	3	1	21

The chart below shows the aggregated numbers of issues identified in each category. Solid colored bars represent issues with a confidence level of Certain, and the bars fade as the confidence level falls.



## Mitigations

### 1. SQL Injection:

- Use parameterized queries or prepared statements to ensure input validation.
- Employ stored procedures to limit direct SQL code execution.
- Regularly update and patch database management systems.

### 2. Cross-Site Scripting (XSS):

- Validate and sanitize user inputs, escaping special characters.
- Implement Content Security Policy (CSP) headers to restrict script sources.
- Encode output data before rendering it in the browser.

### 3. Cross-Site Request Forgery (CSRF):

- Use anti-CSRF tokens to validate the origin of requests.
- Ensure that sensitive actions (e.g., changing passwords) require additional authentication.
- Implement SameSite cookie attributes to control cookie behavior.

4. **Security Misconfigurations:**

- Regularly audit and review server configurations.
- Disable unnecessary services and features.
- Follow the principle of least privilege for user permissions.

5. **Session Management Issues:**

- Use secure, random session identifiers.
- Implement session timeout mechanisms.
- Regularly rotate session tokens and credentials.

6. **Sensitive Data Exposure:**

- Encrypt sensitive data, both in transit (using HTTPS) and at rest.
- Avoid storing sensitive information unless absolutely necessary.
- Use strong, industry-standard encryption algorithms.

7. **Insecure Direct Object References (IDOR):**

- Implement proper access controls and authorization mechanisms.
- Use indirect references, such as tokens or unique identifiers, instead of relying on sequential or easily guessable values.
- Regularly audit and review access controls.

8. **Unvalidated Redirects and Forwards:**

- Avoid using user input to construct URLs for redirects.
- Implement a whitelist of allowed redirect destinations.
- Provide users with a confirmation step before performing redirects.

9. **File Upload Vulnerabilities:**

- Implement proper file type verification and validation.
- Store uploaded files in a secure location with restricted access.
- Regularly scan uploaded files for malicious content.

## 10. Security Headers:

- Implement security headers, such as Content Security Policy (CSP), StrictTransport-Security (HSTS), and X-Content-Type-Options.
- Regularly review and update security header configurations.

It's important to note that the remediation process may vary based on the specific technologies and frameworks used in the web application. Additionally, organizations should follow best practices, stay informed about security updates, and conduct regular security assessments to maintain a robust defense against evolving threats.

## Conclusion

In this project, multiple findings and vulnerabilities were prevalent post test using Burp Suite. Attacks like Cross-Site Scripting, Sensitive Data Exposure, and Unvalidated Redirects and Forwards were all shown here. An attacker could use vulnerabilities found in this lab and exploit them for personal gain which could ultimately degrade the posture of a website and protection of assets for a company. Using the interceptor I was able to gain request sent, then I was able to modify the request to change information on that website, then I was able to use repeater to change the productID type which then gave me an error and I was able to exploit the website. And finally I was able to run a scan on another website using Burp Suite Pro, where I was able to find issues and remediation to them and generate a report on that scan.

## Appendix

### A.

Burp Suite Community Edition v2023.11.1.3 - Temporary Project									
<div> <div> Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Settings </div> <div> Organizer Extensions Learn </div> </div>									
<div> Intercept HTTP history WebSockets history Proxy settings </div>									
<div> Filter settings: Hiding CSS, image and general binary content </div>									
#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Ext
1	https://portswigger.net	GET	/			200	48562	HTML	
2	https://portswigger.net	GET	/content/images/svg/icons/enterprise....			200	2075	XML	svg
4	https://portswigger.net	GET	/content/images/svg/icons/profession...			200	1919	XML	svg
7	https://portswigger.net	GET	/content/images/svg/icons/community...			200	2075	XML	svg
8	https://portswigger.net	GET	/mega-nav/images/dastardly.svg			200	1895	XML	svg
10	https://portswigger.net	GET	/bundles/public/staticcms.js?v=zXANA...		✓	200	23142	script	js
12	https://portswigger.net	GET	/content/images/logos/portswigger-lo...			200	4778	XML	svg
17	https://portswigger.net	GET	/images/company-logos/amazon.svg			200	6706	XML	svg
18	https://portswigger.net	GET	/images/company-logos/nasa.svg			200	7233	XML	svg
19	https://portswigger.net	GET	/images/company-logos/axa.svg			200	2968	XML	svg
20	https://portswigger.net	GET	/images/company-logos/fedex.svg			200	4154	XML	svg

**B.**



Intercept HTTP history WebSockets history Proxy settings

Request to https://www.youtube.com:443 [172.217.0.78]

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```

20 Sec-Fetch-Site: same-origin
21 Sec-Fetch-Mode: cors
22 Sec-Fetch-Dest: empty
23 Referer:
  https://www.youtube.com/embed/DqhZ9cDfEBw?origin=https://portswigger.
  net&rel=0&enablejsapi=1
24 Accept-Encoding: gzip, deflate, br
25 Accept-Language: en-US,en;q=0.9
26 Priority: u=4, i
27
28 {
  "context":{
    "client":{
      "hl":"en",
      "gl":"US",
      "clientName":56,
      "clientVersion":"1.20240102.01.00",
      "configInfo":{
        "appInstallData":
          "CODY4awGEOe6rwUQv6OwBRDQjbAFEJT6_hIQu6OwBRC9tq4FEKGSsAUQvYqW
          BRComrAFEKaBsAUQpcL-EhDpjLAFELiLrgUQmvCvBRDh8q8FEM2VsAUQmPz-E
          hDJ968FEPX5rwUQ6-j-EhC36v4SEL2ZsAUQt52wBRCIh7AFEI-isAUQ5LP-Eh
          DZya8FELmE_xIQqfevBRDqw68FEMyu_hIQzN-uBRDtorAFEKihsAUQ1YiwBRD
          T4a8FEN3o_hIQvwmvBRCEi7AFE0eG_xIQ_IWwBRC8-a8FEKKBsAUQrLevBRC3
          768FENqYsAUQ65OuBRCooLAFEIjjrwUQmZSwBRCu1P4SEKuCsAUQx4OwBRCNo
          rAFEMf8tyI%3D"
      },
      "userAgent":
        "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (
        KHTML, like Gecko) Chrome/120.0.6099.71 Safari/537.36",
      "connectionType":"CONN_CELLULAR_4G"
    },
    "thirdParty":{
      "embedUrl":"https://portswigger.net/"
    }
  },
  "events":[
    {
      "eventTimeMs":1704488034949,
      "latencyActionTicked":{
        "tickName":"vc",
        "clientActionNonce":"1Y7GO_inKRRNQUJC"
      },
      "context":{
        "lastActivityMs":"3044"
      }
    }
  ],
  "serializedClientEventId":{
    "serializedEventId":"YGyYZar2Ora__tcP-6-TkAw",
    "clientCounter":"24433"
  }
}

```

C.

The screenshot displays the Burp Suite interface with the 'Proxy' tab selected. A list of HTTP requests is shown, with request 285 highlighted. A context menu is open over request 285, listing various actions such as 'Send to Repeater', 'Send to Sequencer', and 'Send to Organizer'. The 'Request' and 'Response' panes are visible at the bottom, showing the raw HTTP data for the selected request.

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Ext
277	https://0a15007104487c9283e1f...	GET	/product?productId=1		✓	200	5283	HTML	
280	https://0a15007104487c9283e1f...	GET	/product?productId=1		✓	200	5283	HTML	
283	https://0a15007104487c9283e1f...	GET	/product?productId=1		✓	200	5283	HTML	
285	https://0a15007104487c9283e1f...	GET	/product?productId=1		✓	200	5283	HTML	
473	https://0a15007104487c9283e1f...	GET	/product?pro					HTML	
477	https://0a15007104487c9283e1f...	GET	/product?pro					HTML	
264	https://0a15007104487c9283e1f...	GET	/my-account?					HTML	
257	https://0a15007104487c9283e1f...	GET	/my-account						
258	https://0a15007104487c9283e1f...	GET	/login					HTML	
261	https://0a15007104487c9283e1f...	POST	/login					HTML	
263	https://0a15007104487c9283e1f...	POST	/login						

**Request**

```

1 GET /product?productId=1 HTTP/2
2 Host: 0a15007104487c9283e1fa4800880070.web-security-academy.net
3 Cookie: session=rCCZiGt1089QnXIXTL0eFccAFsdwKBGn
4 Sec-Ch-Ua: "Not A Brand";v="8", "Chromium";v="120"
5 Sec-Ch-Ua-Mobile: ?0
6 Sec-Ch-Ua-Platform: "Windows"
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.71 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-Dest: document
13 Referer: https://0a15007104487c9283e1fa4800880070.web-security-academy.net/
14 Accept-Encoding: gzip, deflate, br
15 Accept-Language: en-US,en;q=0.9
16 Priority: u=0, i
17
18

```

**Response**

```

1 HTTP/2
2 Content-Type: text/html; charset=utf-8
3 X-Frame-Options: DENY
4 Content-Security-Policy: script-src 'self' https://resources.labheader.js/labHeader.js
5
6 <!DOCTYPE html>
7 <html>
8 <head>
9 <title>Academy Lab Header</title>
10 <meta charset="utf-8">
11 <script src="/resources/labheader.js/labHeader.js"></script>
12 <div id="academyLabHeader">
13 <div class="container">
14 <div class="logo">
15 <div class="title-container">
16 <h2>

```