

ASSIGNMENT-4

NAME: RAMAR PRIYA MAHA LAKSHMI

REGNO: 21BCE7521

1. Exploring Burp Suite.

Burp Suite is a framework written in Java that aims to provide a one-stop-shop for web application penetration testing. In many ways, this goal is achieved as Burp is very much the industry standard tool for hands-on web app security assessments. Burp Suite is also very commonly used when assessing mobile applications, as the same features which make it so attractive for web app testing translate almost perfectly into testing the APIs (**A**pplication **P**rogramming **I**nterfaces) powering most mobile apps.

At the simplest level, Burp can capture and manipulate all of the traffic between an attacker and a webserver: this is the core of the framework. After capturing requests, we can choose to send them to various other parts of the Burp Suite. This ability to intercept, view, and modify web requests prior to them being sent to the target server (or, in some cases, the responses before they are received by our browser), makes Burp Suite perfect for any kind of manual web app testing.

There are various different editions of Burp Suite available. We will be working with the **Burp Suite Community** edition, as this is free to use for any (legal) non-commercial use. The Burp Suite *Professional* and *Enterprise* editions both require expensive licenses but come with powerful extra features:

- **Burp Suite Professional** is an unrestricted version of Burp Suite Community. It comes with features such as:
 - An automated vulnerability scanner.
 - A fuzzer/bruteforcer that isn't rate limited.
 - Saving projects for future use; report generation.
 - A built-in API to allow integration with other tools.
 - Unrestricted access to add new extensions for greater functionality.

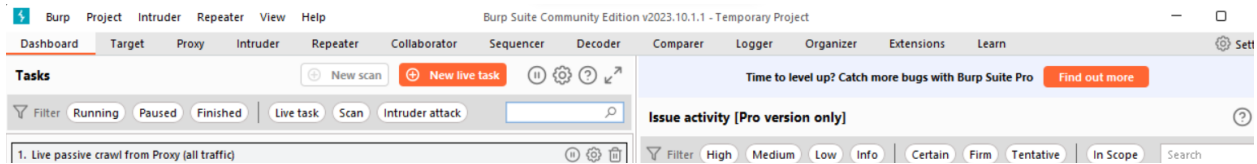
- Access to the Burp Suite Collaborator (effectively providing a unique request catcher self-hosted or running on a Portswigger owned server)
- **Burp Suite Enterprise** is slightly different. Unlike the community and professional editions, Burp Enterprise is used for continuous scanning. It provides an automated scanner that can periodically scan webapps for vulnerabilities in much the same way as **Nessus**. Unlike the other editions of Burp Suite which allow you to perform manual attacks from your own computer, Enterprise sits on a server and constantly scans target web apps for vulnerabilities.

Features of Burp Suite:

Whilst Burp Community has a relatively limited feature-set compared to the Professional edition, it still has many superb tools available. These include:

- **Proxy:** The most well-known aspect of Burp Suite, the Burp Proxy allows us to intercept and modify requests/responses when interacting with web applications.
- **Repeater:** The second most well-known Burp feature **Repeater** allows us to capture, modify, then resend the same request numerous times. This feature can be absolutely invaluable, especially when we need to craft a payload through trial and error (e.g. in an SQLi -- **Structured Query Language Injection**) or when testing the functionality of an endpoint for flaws.
- **Intruder:** Although harshly rate-limited in Burp Community, **Intruder** allows us to spray an endpoint with requests. This is often used for bruteforce attacks or to fuzz endpoints.
- **Decoder:** Though less-used than the previously mentioned features, **Decoder** still provides a valuable service when transforming data either in terms of decoding captured information, or encoding a payload prior to sending it to the target. Whilst there are other services available to do the same job, doing this directly within Burp Suite can be very efficient.
- **Comparer:** As the name suggests, **Comparer** allows us to compare two pieces of data at either word or byte level. Again, this is not something that is unique to Burp Suite, but being able to send (potentially very large) pieces of data directly into a comparison tool with a single keyboard shortcut can speed things up considerably.
- **Sequencer:** We usually use **Sequencer** when assessing the randomness of tokens such as session cookie values or other supposedly random generated data. If the algorithm is not generating

secure random values, then this could open up some devastating avenues for attack.



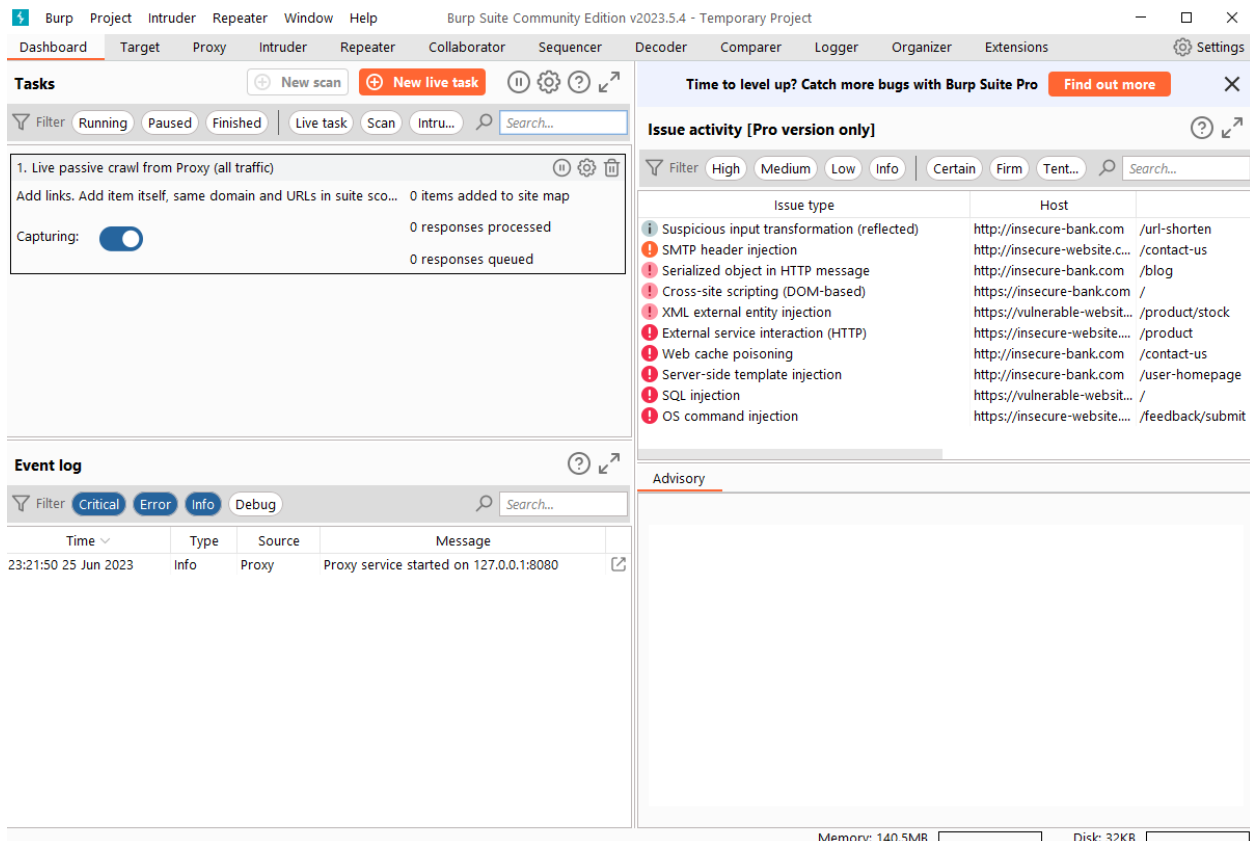
In addition to the myriad of in-built features, the Java codebase also makes it very easy to write extensions to add to the functionality of the Burp framework. These can be written in Java, Python (using the Java Jython interpreter), or Ruby (using the Java Jruby interpreter). The Burp Suite Extender module can quickly and easily load extensions into the framework, as well as providing a marketplace to download third-party modules (referred to as the "BApp Store"). Whilst many of these extensions require a professional license to download and add in, there are still a fair number that can be integrated with Burp Community.

Getting started with Burp Suite:

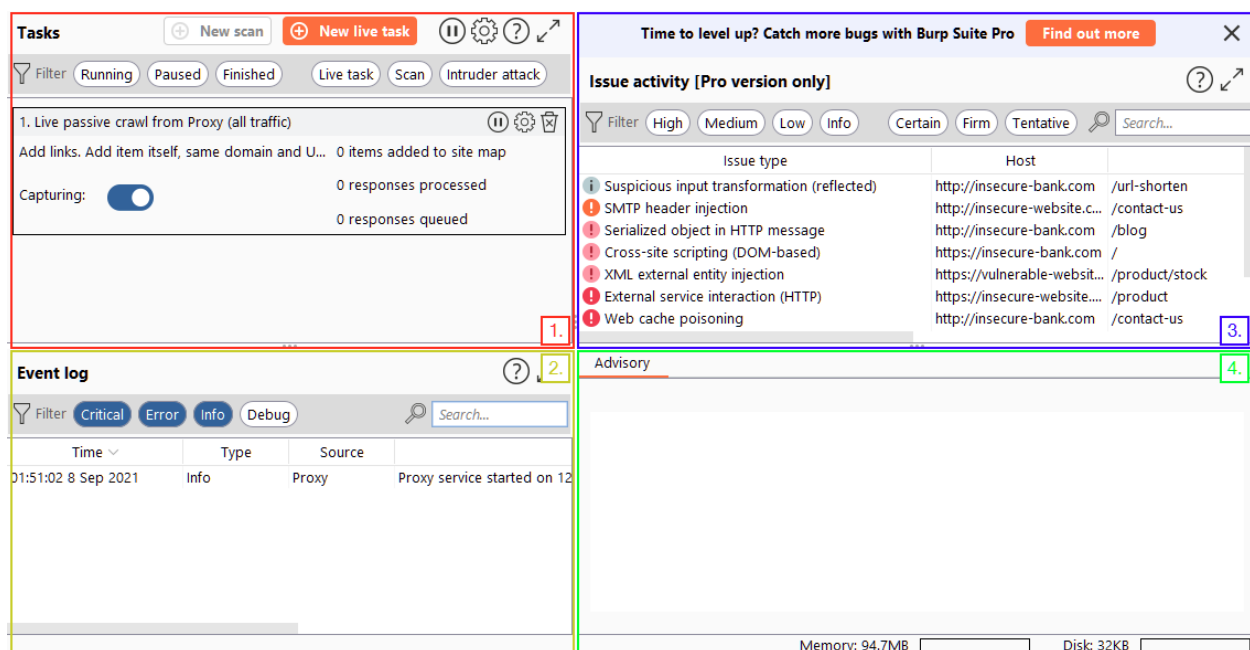
When we open Burp Suite and have accepted the terms and conditions, we are met with a window asking us to select the project type. All we can do here is click "Next"

Click "Start Burp", and the main Burp Suite interface will open!

The BURP Dashboard appears!



In short, the Dashboard interface is split into four quadrants:



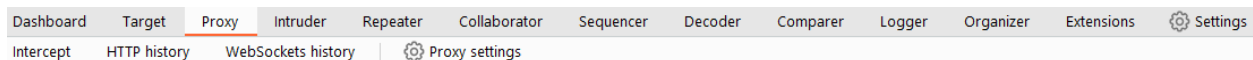
1. The Tasks menu allows us to define background tasks that Burp Suite will run whilst we use the application.
2. The Event log tells us what Burp Suite is doing (e.g. starting the Proxy), as well as information about any connections that we are making through Burp.
3. The Issue Activity section is exclusive to Burp Pro. It won't give us anything using Burp Community, but in Burp Professional it would list all of the vulnerabilities found by the automated scanner. These would be ranked by severity and filterable by how sure Burp is that the component is vulnerable.
4. The Advisory section gives more information about the vulnerabilities found, as well as references and suggested remediations. These could then be exported into a report.

Throughout the various tabs and windows of Burp Suite, you will find little help icons: a question mark within a circle.



Clicking on these will open a new window containing help for the section. These are extremely useful if you're ever stuck and don't know what a feature does.

Navigating around the Burp Suite GUI by default is done entirely using the top menu bars:

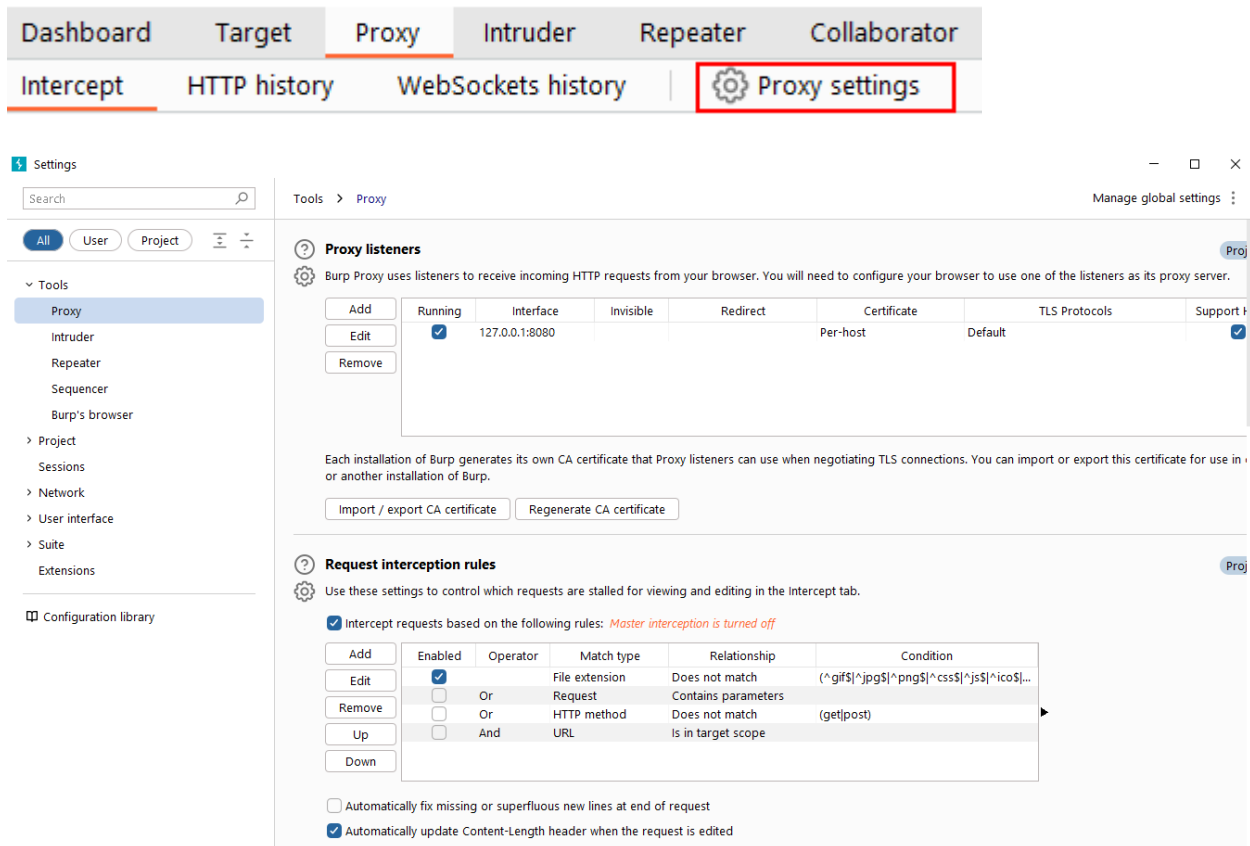


These allow you to switch between modules.

In addition to the menu bar, Burp Suite also has keyboard shortcuts that allow quick navigation to key tabs. By default, these are:

Shortcut	Does
Ctrl + Shift + D	Switch to the Dashboard
Ctrl + Shift + T	Switch to the Target tab
Ctrl + Shift + P	Switch to the Proxy tab
Ctrl + Shift + I	Switch to the Intruder tab
Ctrl + Shift + R	Switch to the Repeater tab

It should be noted that many of the tools in Burp Suite offer shortcuts to specific categories of settings. For example, the Proxy tool includes a "Proxy settings" button which will open the Settings window directly to the section relevant to the proxy.



The Search feature of the settings page is a relatively new addition; however, it is absolutely invaluable, allowing us to search for settings using keywords.

Burp Proxy:

The Burp Proxy is the most fundamental (and most important!) of the tools available in Burp Suite. It allows us to capture requests and responses between ourselves and our target. These can then be manipulated or sent to other tools for further processing before being allowed to continue to their destination.

For example, if we make a request to <https://tryhackme.com> through the Burp Proxy, our request will be captured and won't be allowed to continue to the

TryHackMe servers until we explicitly allow it through. We can choose to do the same with the response from the server, although this isn't active by default. This ability to intercept requests ultimately means that we can take complete control over our web traffic -- an invaluable ability when it comes to testing web applications.

When we first open the Proxy tab, Burp gives us a bunch of useful information and background reading. This information is well worth reading through; however, the real magic happens after we capture a request:



With the proxy active, a request was made to the TryHackMe website. At this point, the browser making the request will hang, and the request will appear in the Proxy tab giving us the view shown in the screenshot above. We can then choose to forward or drop the request (potentially after editing it). We can also do various other things here, such as sending the request to one of the other Burp modules, copying it as a cURL command, saving it to a file, and many others.

When we have finished working with the Proxy, we can click the "Intercept is on" button to disable the Intercept, which will allow requests to pass through the proxy without being stopped.

Burp Suite will still (by default) be logging requests made through the proxy when the intercept is off. This can be very useful for going back and analysing prior requests, even if we didn't specifically capture them when they were made.

Burp will also capture and log WebSocket communication, which, again, can be exceedingly helpful when analysing a web app.

The logs can be viewed by going to the "HTTP history" and "WebSockets history" sub-tabs:

History

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	

It is worth noting that any requests captured here can be sent to other tools in the framework by right-clicking them and choosing "Send to...". For example, we could take a previous HTTP request that has already been proxied to the target and send it to Repeater.

Finally, there are also Proxy specific options, which in the Proxy Settings, accessible by clicking on the "Proxy Settings" button.

These options give us a *lot* of control over how the proxy operates, so it is an excellent idea to familiarise yourself with these.

? 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
Add	<input checked="" type="checkbox"/>		Content type header	Matches	text
Edit	<input type="checkbox"/>	Or	Request	Was modified	
Remove	<input checked="" type="checkbox"/>	Or	Request	Was intercepted	
Up	<input type="checkbox"/>	And	Status code	Does not match	^304\$
Down	<input type="checkbox"/>	And	URL	Is in target scope	

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

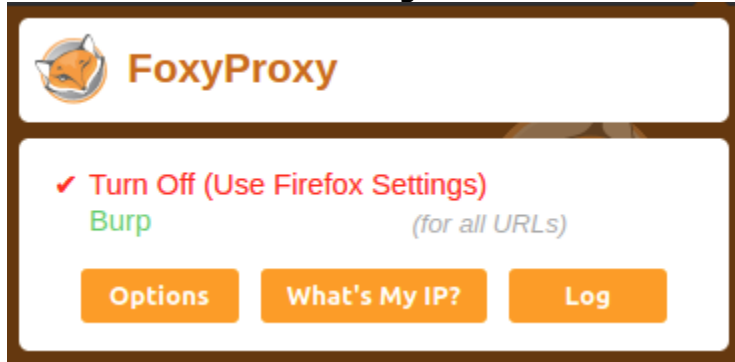
You can make your own rules for most of the Proxy options, so this is one section where looking around and experimenting will serve you very well indeed!

There are two ways to proxy our traffic through Burp Suite.

1. We could use the embedded browser.

2. We can configure our local web browser to proxy our traffic through Burp.

When you click on the FoxyProxy icon at the top of the screen, you will see that there is a configuration available for Burp:

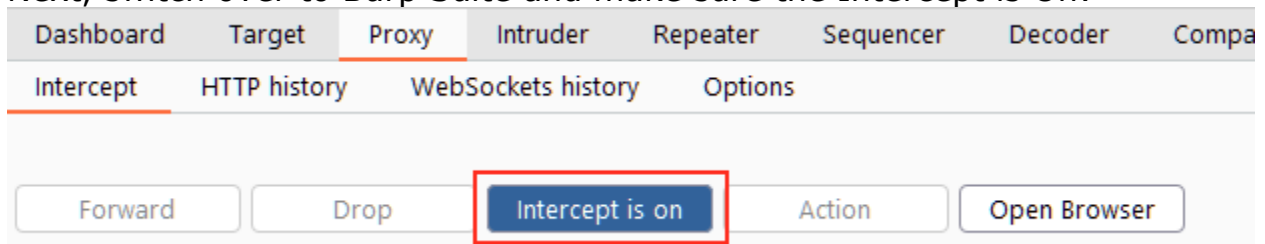


If we click on the "Burp" config, our browser will start directing all of our traffic through 127.0.0.1:8080!

Activate this config now -- the icon in the menu should change to indicate that we have a proxy running:



Next, switch over to Burp Suite and make sure the Intercept is On:



Now, try accessing the homepage for <https://www.google.com> in Firefox. Your browser should hang, and your proxy will populate with the request headers.

From here, you can choose to forward or drop the request. Alternatively, you could send it to another tool or perform any number of other actions by right-clicking on the request and selecting an option from the right-click menu.

We can start the Burp Browser with the "Open Browser" button in the proxy tab:



Intruder:

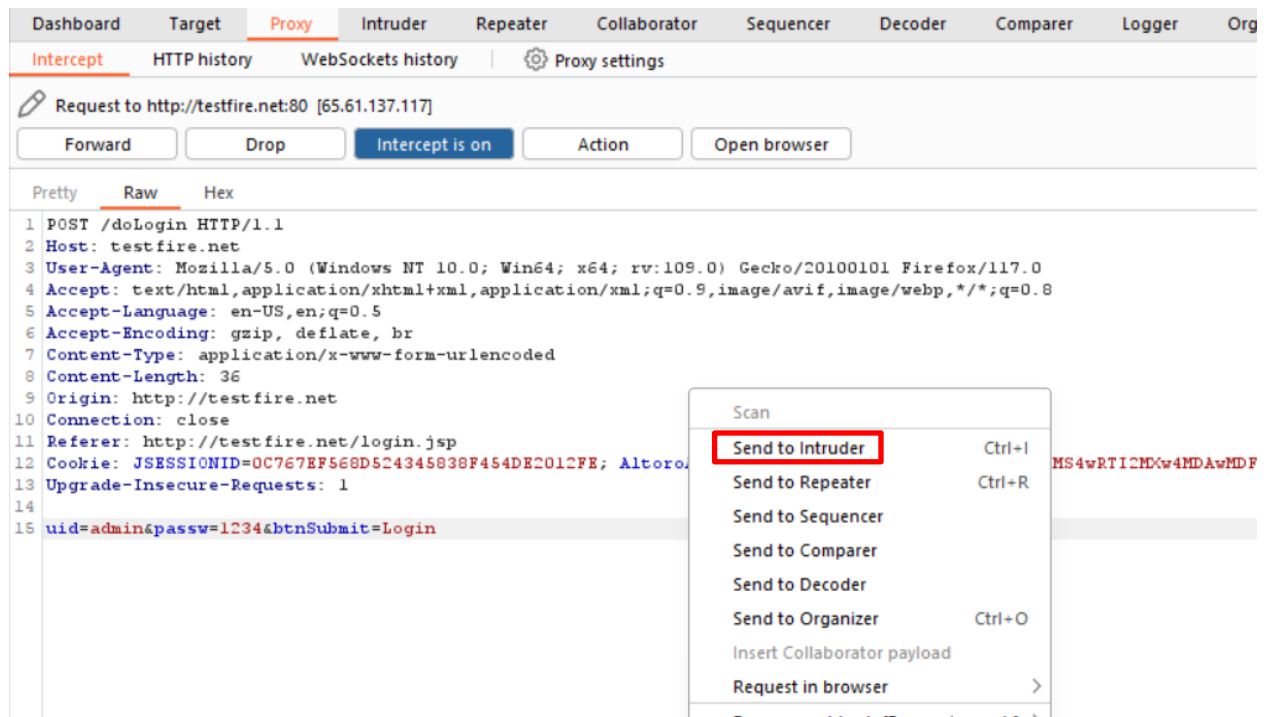
This is a very powerful tool and can be used to carry out different attacks on web applications. It is very easy to configure and you can use it to carry out several testing tasks faster and very effectively. It is a perfect tool that can be used for a brute-force attack and also carry out very difficult blind SQL injection operations.

Burp Suite Intruder mode of operation is usually through HTTP request and modify this request to your taste. This tool can be used for the analysis of the application responses to requests.

There is a need for you to specify some payloads on every attack and the exact location in the base request where the payloads are to be released or placed. We have different ways of building or generating your payloads today. We have payloads like a simple list, username generator, numbers, brute forcer, runtime file, bit flipper, and many.

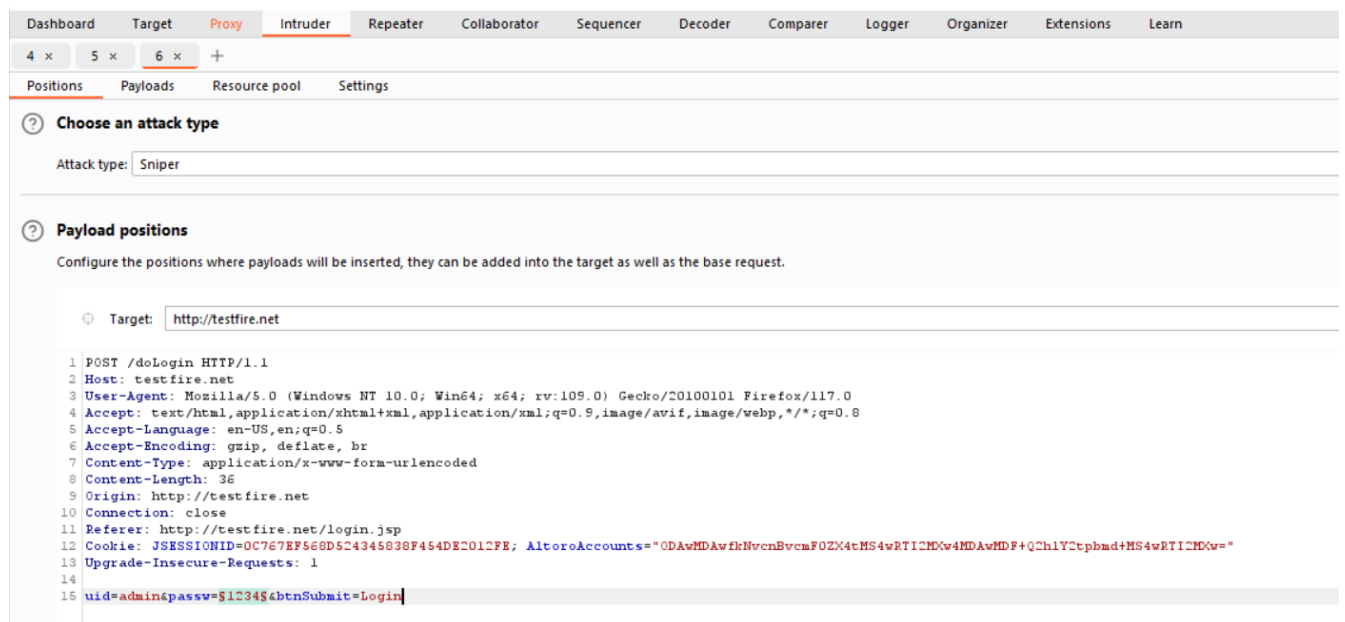
The Burp Suite intruder has different algorithms that help in the placement of these payloads into their exact location.

Burp Suite intruders can be used to enumerate identifiers, extracting useful data, and performing fuzzing operations for vulnerabilities.

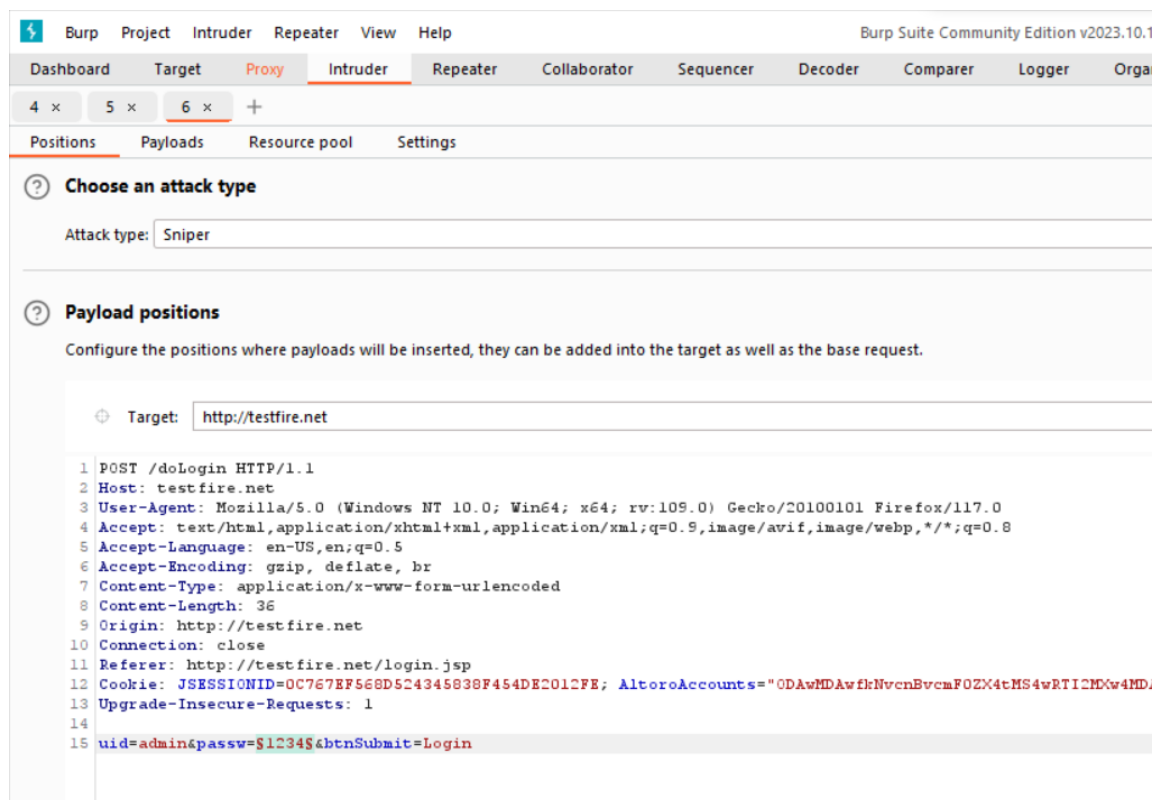


To carry out a successful attack using Burp suite Intruder follow these steps:

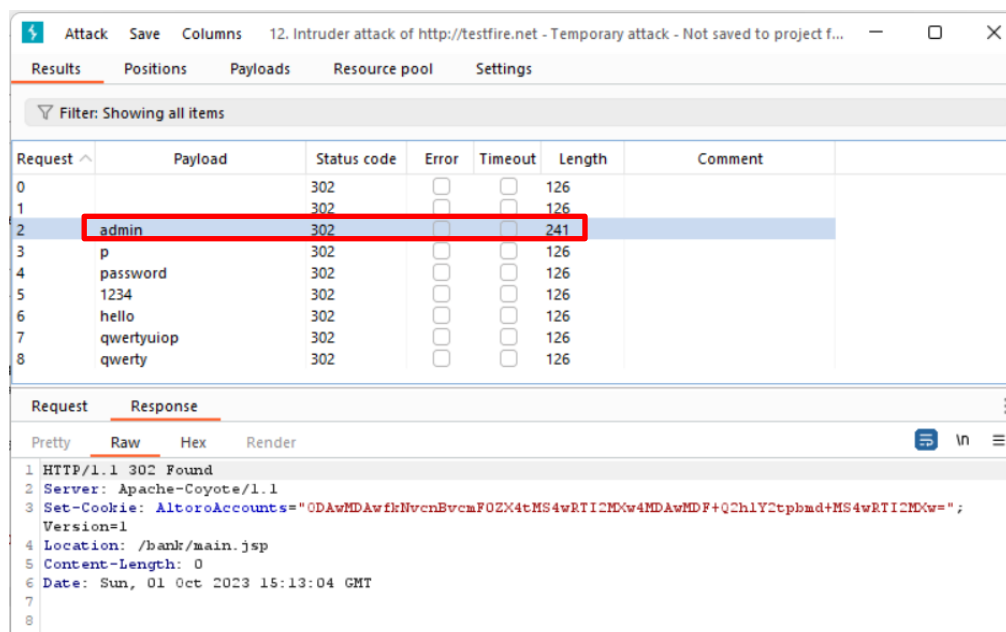
- Find the identifier which most times is highlighted inside the request and also the response confirming the validity.
- Then configure a single payload position that is enough to carry out the attack.



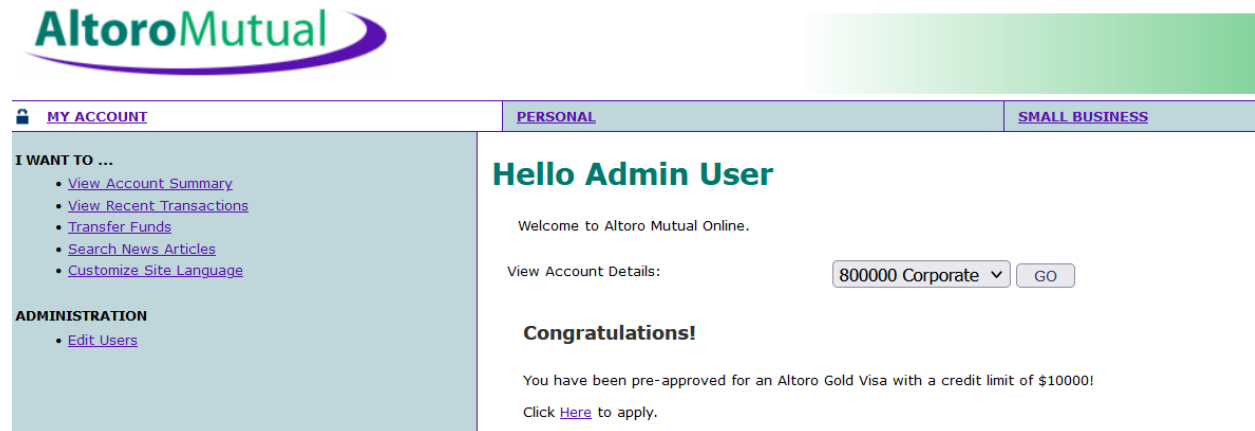
Use the **Payload type** drop-down to generate all identifiers needed to test, using the correct format.



After entering some of these important details to carry out an attack, you can click on the **Start attack** button. The next pop-up page will be the result page, which you will need to analyze.



If you check the above image, you can see that one identifier returns a different HTTP status code or response length, the one that returns different status and length from others is actually the correct password, if you go ahead and use that you will be able to log in.



Repeater:

Burp Suite Repeater is designed to manually manipulate and re-send individual HTTP requests, and thus the response can further be analyzed. It is a multi-task tool for adjusting parameter details to test for input-based issues.

Using Burp Repeater With HTTP request:

If you want to make use of Burp Suite Repeater with an HTTP request, you only need to right-click on the request and select **Send to Repeater**. There is an immediate creation of a new request tab in the Repeater and you will also see all the relevant details on the message editor for further manipulation. You can also open a new Repeater tab manually and select the **HTTP** option.

Request to http://testfire.net:80 [65.61.137.117]

Forward Drop **Intercept is on** Action Open browser

Pretty Raw Hex

```

1 POST /doLogin HTTP/1.1
2 Host: testfire.net
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 36
9 Origin: http://testfire.net
10 Connection: close
11 Referer: http://testfire.net/login.jsp
12 Cookie: JSESSIONID=0C767EF568D524345838F454DE2012FE; AltoroAccounts=
13 Upgrade-Insecure-Requests: 1
14
15 uid=admin&passw=1234&btnSubmit=Login

```

Scan

- Send to Intruder Ctrl+I
- Send to Repeater** Ctrl+R
- Send to Sequencer
- Send to Comparer
- Send to Decoder
- Send to Collaborator Ctrl+O

Sending HTTP Requests

After making all the necessary manipulation to your request it is ready to send, just click the **Send** or **Go** button to send it to the server. The response is displayed on the response panel by the right-hand side. You will also notice that the response message is not editable.

1 x +

Send Cancel < > Follow redirection

Request

Pretty Raw Hex

```

1 POST /doLogin HTTP/1.1
2 Host: testfire.net
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 37
9 Origin: http://testfire.net
10 Connection: close
11 Referer: http://testfire.net/login.jsp
12 Cookie: JSESSIONID=0C767EF568D524345838F454DE2012FE; AltoroAccounts=
13 Upgrade-Insecure-Requests: 1
14
15 uid=admin&passw=admin&btnSubmit=Login

```

Response

Pretty Raw Hex Render

```

1 HTTP/1.1 302 Found
2 Server: Apache-Coyote/1.1
3 Set-Cookie: JSESSIONID=DGBASDE4BAEBID9A5C6323A6F7C8839E; Path=/; HttpOnly
4 Set-Cookie: AltoroAccounts=
5 Location: /bank/main.jsp
6 Content-Length: 0
7 Date: Sun, 01 Oct 2023 16:21:46 GMT
8 Connection: close
9
10

```

Target:

The Burp Suite **Target tab** > **Site map** tool will help you with an overview of all your target application's content and functionality. The left-hand side is in form of a tree view that arranges the content of a URL in a hierarchical order, they are split into domains, directories, folders, and files.

The screenshot shows the Burp Suite interface with the **Target** tab selected. The **Site map** tool is active, displaying a tree view of the target application's structure on the left. The main panel shows a list of URLs and their corresponding requests and responses.

Host	Method	URL	Params	Status	Length	MIME type	Title	Comment	Time
https://content-signature-2.cdn.mozilla.net	GET	/chains/normandy.c...		200	5854	script			20:40:42 1 ...
https://content-signature-2.cdn.mozilla.net	GET	/chains/onecr.conte...		200	5851	script			20:40:47 1 ...
https://content-signature-2.cdn.mozilla.net	GET	/chains/onecr.conte...		200	5838	script			20:40:46 1 ...
https://content-signature-2.cdn.mozilla.net	GET	/chains/normandy.c...		304	169	script			20:40:44 1 ...

The detailed view shows the request and response for the selected URL. The request is a GET request to `/chains/normandy.c...` with a status of 200. The response is a script file with a status of 200 and a MIME type of `text/javascript`.

You can manually map your target application by launching the Burp suite browser either internal browser or the external browser and make sure the proxy interception is turned **OFF** while you browse the entire application manually.

Target Scope

You can configure your target scope by selecting any branch on the **Site map**. Select **Add to scope** or **Remove from the scope** from the menu. You can configure your Site map display filters to show what you want to view and what you want to delete.

The screenshot shows the Burp Suite interface with the **Target** tab selected. The **Site map** tool is active, displaying a tree view of the target application's structure on the left. The main panel shows a list of URLs and their corresponding requests and responses.

Host	Method	URL	Params	Status
http://testfire.net	GET	/bank/main.jsp		200
http://testfire.net	GET	/index.jsp		200
http://testfire.net	GET	/login.jsp		200
http://testfire.net	GET	/search.jsp?query=...		200
http://testfire.net	POST	/doLogin		302
http://testfire.net	POST	/doLogin		302
http://testfire.net	GET	/logout.jsp		302
http://testfire.net	GET	/admin/admin.jsp		200
http://testfire.net	GET	/bank/apply.jsp		200
http://testfire.net	GET	/bank/customize.jsp		200

The context menu for the selected URL `http://testfire.net/` is open, showing options like **Add to scope**, **Remove from scope**, **Scan**, **Engagement tools**, **Compare site maps**, and **Expand branch**.

Sequencer:

Burp Sequencer enables you to analyze the quality of randomness in a sample of tokens. You can use Sequencer to test any tokens that are intended to be unpredictable, such as:

- Session tokens.
- Anti-CSRF tokens.
- Password reset tokens.

Sequencer runs multiple randomness tests against a sample of tokens, then compiles the results to give you an indication of the quality of randomness in the sample.

Decoder:

Burp Decoder enables you to transform data using common encoding and decoding formats. You can use Decoder to:

- Manually decode data.
- Automatically identify and decode recognizable encoding formats, such as URL-encoding.
- Transform raw data into various encoded and hashed formats.

Decoder enables you to apply layers of transformations to the same data. This enables you to unpack or apply complex encoding schemes.

You can send data to Burp Decoder from the message editor in various Burp tools, such as HTTP history. To carry out a data transformation using Burp Decoder:

1. Locate the data that you want to analyze.

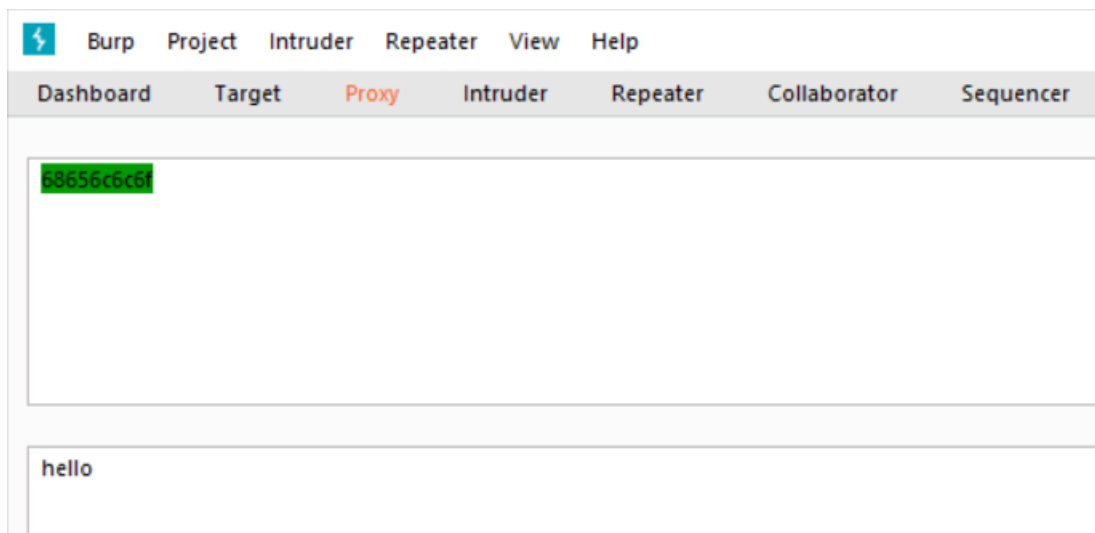
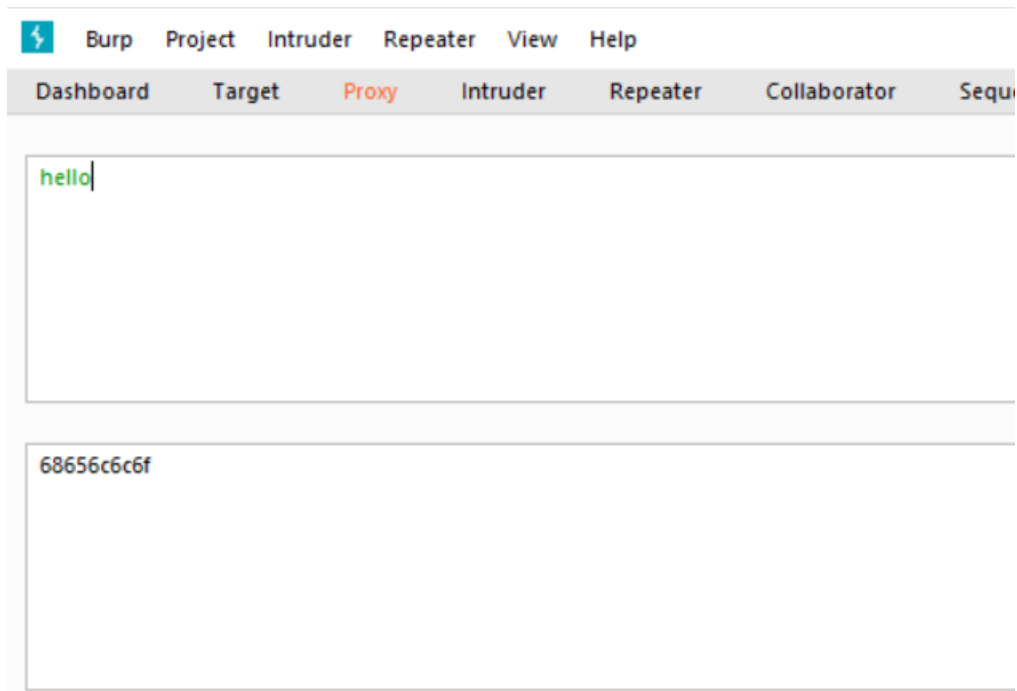
2. Right-click the data in the message editor and select **Send to Decoder**.
3. Go to the **Decoder** tab. The data is in the top panel.
4. Select the operation you want to perform on the data from the controls beside the data panel. For example, **Encode as** or **Smart decode**.

You can view the data in either **Text** or **Hex** form.

- **Decode as** - Apply a decoding function to the data.
- **Encode as** - Apply an encoding function to the data.
- **Hash** - Apply a hash function to the data. The available functions depend upon the capability of your Java platform.
- **Smart decode** - Burp looks for encoded data, and applies layers of decoding until there aren't any further recognizable data formats. This is often useful as an automated first decoding step.

The following decode and encode functions are available:

- URL.
- HTML.
- Base64.
- ASCII hex.
- Hex.
- Octal.
- Binary.
- GZIP.



Logger:

Burp Logger records all the HTTP traffic that Burp Suite generates in real-time. You can use Logger to:

- Study the requests sent by any of Burp's tools or extensions.

- See the requests sent by Burp Scanner in real-time.
- Examine the behavior of extensions.
- Study the requests sent with a session handling rule modification.

By default, Logger displays traffic generated by all the tools in Burp Suite. This includes modified requests, and requests sent by extensions. By contrast, the HTTP history only displays traffic from a browser that is proxied through Burp.

Burp Suite Community Edition v2023.10.1.2 - Temporary Project											
Burp Project Intruder Repeater View Help											
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn											
Capture filter: Logger memory limit set to 100MB Capturing requests up to 1MB; capturing responses up to 1MB											
View filter: Showing all items											
#	Time	Tool	Method	Host	Path	Query	Param count	Status code	Length	Start response timer	Comment
1	20:12:58 1 Oct 2023	Proxy	GET	img-getpocket.cdn...	/404x202/filters:form...		0	200	17548	63	
2	20:12:58 1 Oct 2023	Proxy	GET	img-getpocket.cdn...	/404x202/filters:form...		0	200	16327	142	
3	20:12:58 1 Oct 2023	Proxy	GET	img-getpocket.cdn...	/404x202/filters:form...		0	200	14286	149	
4	20:12:58 1 Oct 2023	Proxy	GET	img-getpocket.cdn...	/404x202/filters:form...		0	200	12507	146	
5	20:12:58 1 Oct 2023	Proxy	GET	img-getpocket.cdn...	/404x202/filters:form...		0	200	14981	151	
6	20:12:58 1 Oct 2023	Proxy	GET	img-getpocket.cdn...	/404x202/filters:form...		0	200	13194	151	
7	20:12:58 1 Oct 2023	Proxy	GET	img-getpocket.cdn...	/404x202/filters:form...		0	200	15816	144	
8	20:12:58 1 Oct 2023	Proxy	GET	img-getpocket.cdn...	/404x202/filters:form...		0	200	7705	147	
9	20:12:58 1 Oct 2023	Proxy	GET	img-getpocket.cdn...	/404x202/filters:form...		0	200	14836	149	
10	20:15:52 1 Oct 2023	Proxy	GET	contile.services.mozl...	/v1/tiles		0	200	1615	235	
11	20:15:54 1 Oct 2023	Proxy	GET	contile-images.servi...	/obgoOYObjFea_bX...		0	200	10733	30	

Conclusion:

Burp Suite is a popular and powerful web security testing tool used by cybersecurity professionals to assess the security of web applications. It provides a range of features, including web vulnerability scanning, intercepting and manipulating web traffic, and automated testing, making it an essential tool for identifying and mitigating security risks in web applications.