

## AI FOR CYBER SECURITY WITH IBM QRADAR

### ASSIGNMENT-4

**Name: Vishnubhatla V  
L Sruta Keerthi**

#### **BURPSUITE**

##### What is Burp Suite?

Burp Suite is a comprehensive set of cybersecurity tools designed for web application security testing and vulnerability assessment. It is developed by PortSwigger, a UK-based software company. Burp Suite is widely used by security professionals, penetration testers, and ethical hackers to identify and mitigate security vulnerabilities in web applications.

##### Why is Burp Suite used?

Burp Suite is used for several important purposes in the field of cybersecurity:

1. **Web Application Security Testing:** Burp Suite helps security professionals assess the security of web applications by identifying vulnerabilities such as SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and more.
2. **Vulnerability Assessment:** It scans web applications to detect potential security weaknesses, misconfigurations, and other issues that could be exploited by attackers.
3. **Penetration Testing:** Ethical hackers and penetration testers use Burp Suite to simulate attacks on web applications, uncover vulnerabilities, and provide recommendations for remediation.
4. **Security Research:** Researchers use Burp Suite to analyse and study web application security, helping to improve the overall security of web applications.
5. **Web Application Development:** Developers can use Burp Suite to test their own applications during development to catch and fix security issues before they reach production.

##### Features of Burp Suite:

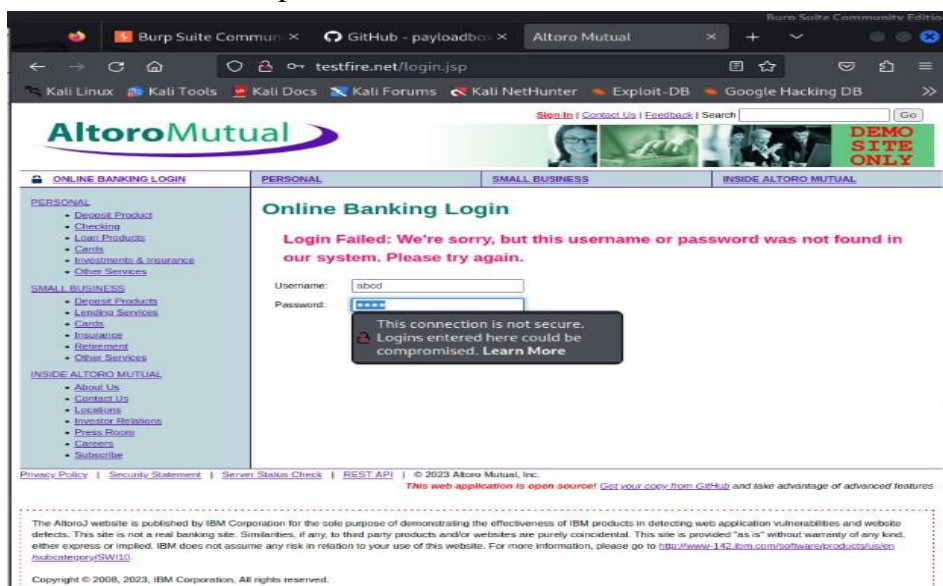
Burp Suite offers a wide range of features to support web application security testing:

1. **Proxy:** Allows intercepting and modifying HTTP/S requests and responses between the client and server, making it possible to analyse and manipulate web traffic.
2. **Scanner:** Automatically scans web applications for common vulnerabilities, such as SQL injection, XSS, and more, providing detailed reports.
3. **Intruder:** Facilitates automated and customizable attacks on web applications to discover vulnerabilities and weak points.
4. **Repeater:** Allows manual modification and resending of individual HTTP/S requests to observe how the application responds, aiding in vulnerability discovery and testing.
5. **Sequencer:** Analyses the randomness of tokens and session identifiers to assess the strength of session management and authentication mechanisms.
6. **Spider:** Crawls and maps the structure of a web application, helping testers understand its functionality and potential attack surfaces.
7. **Decoder:** Provides tools to decode and encode data in various formats, such as Base64 and URL encoding.

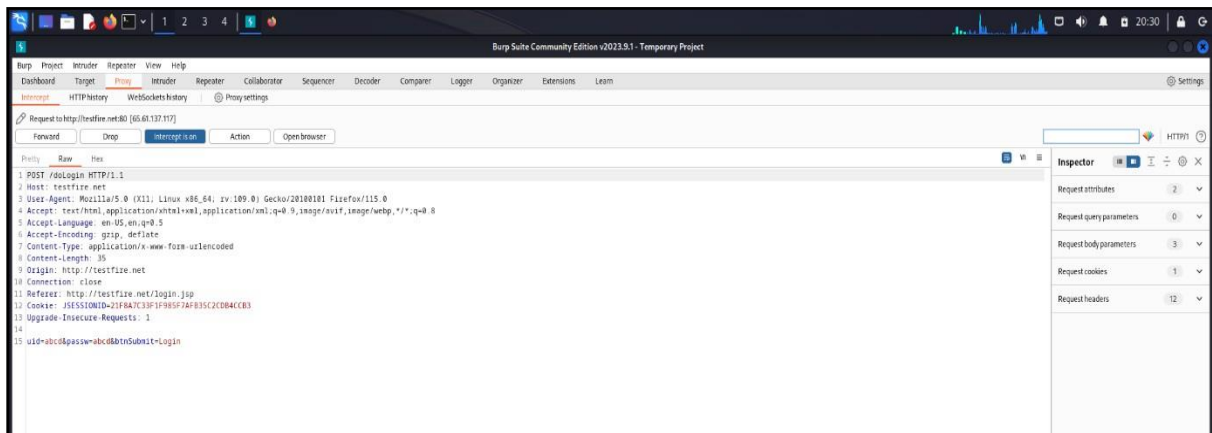
8. Collaborator: Assists in detecting out-of-band vulnerabilities by creating unique payloads that trigger external interactions and reporting the results.
  9. Extensibility: Burp Suite supports the development of custom extensions and plugins, allowing users to add additional functionality.
  10. Reporting: Generates detailed reports with vulnerability findings and recommendations for remediation.
  11. Target Scope Control: Allows users to define the scope of testing by specifying which parts of a web application should be included or excluded.
  12. Session Handling: Manages and maintains user sessions to test authentication and authorization mechanisms thoroughly.
- These features collectively make Burp Suite a powerful tool for identifying and mitigating web application security vulnerabilities.

### Testing the vulnerabilities of testfire.net website

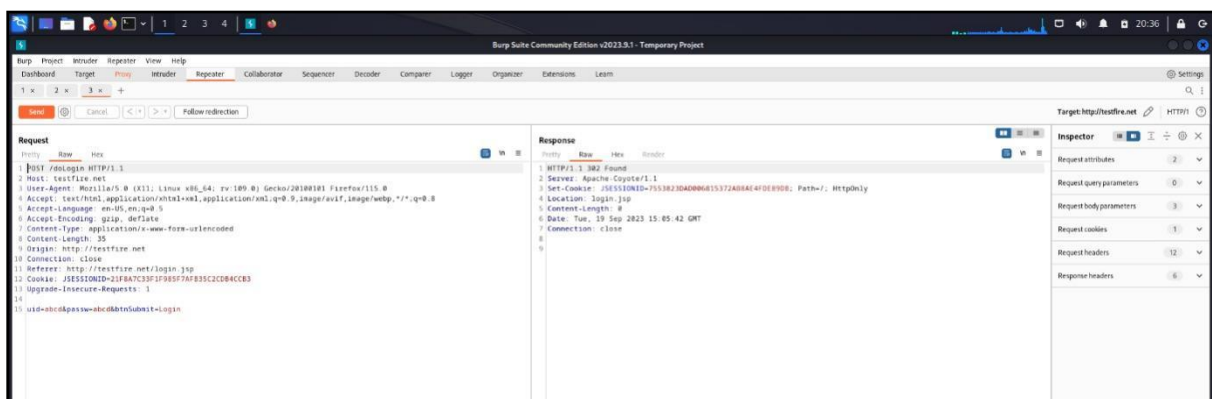
First, the website is opened and credentials are entered.



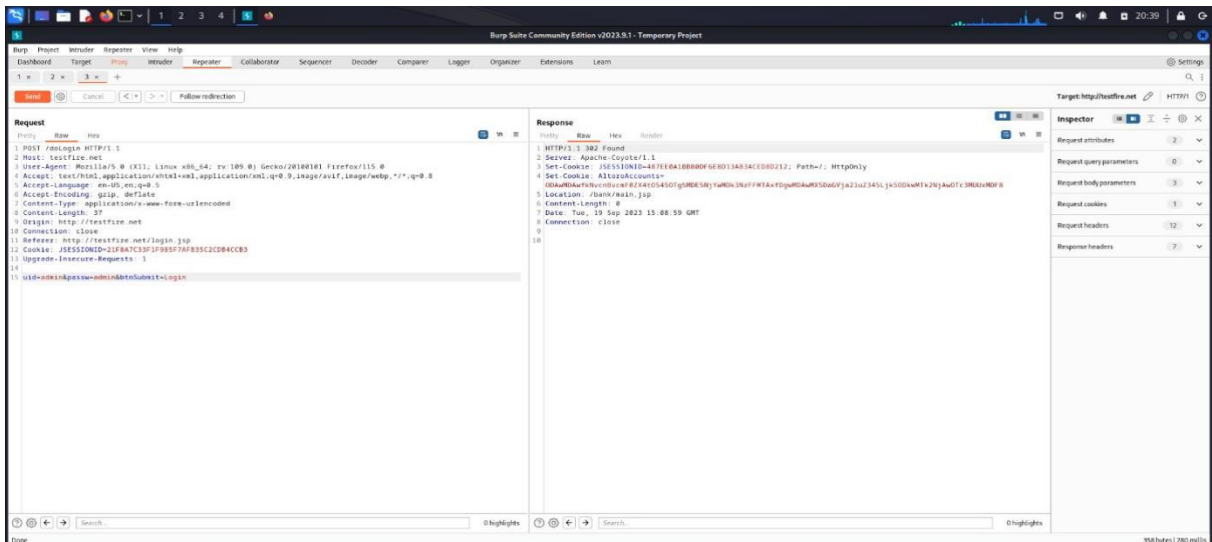
Next, the intercept in the proxy section of Burpsuite is turned on. After that, login button is clicked. After this, the below details are sent to repeater.



Here, as wrong credentials were entered, error is found.

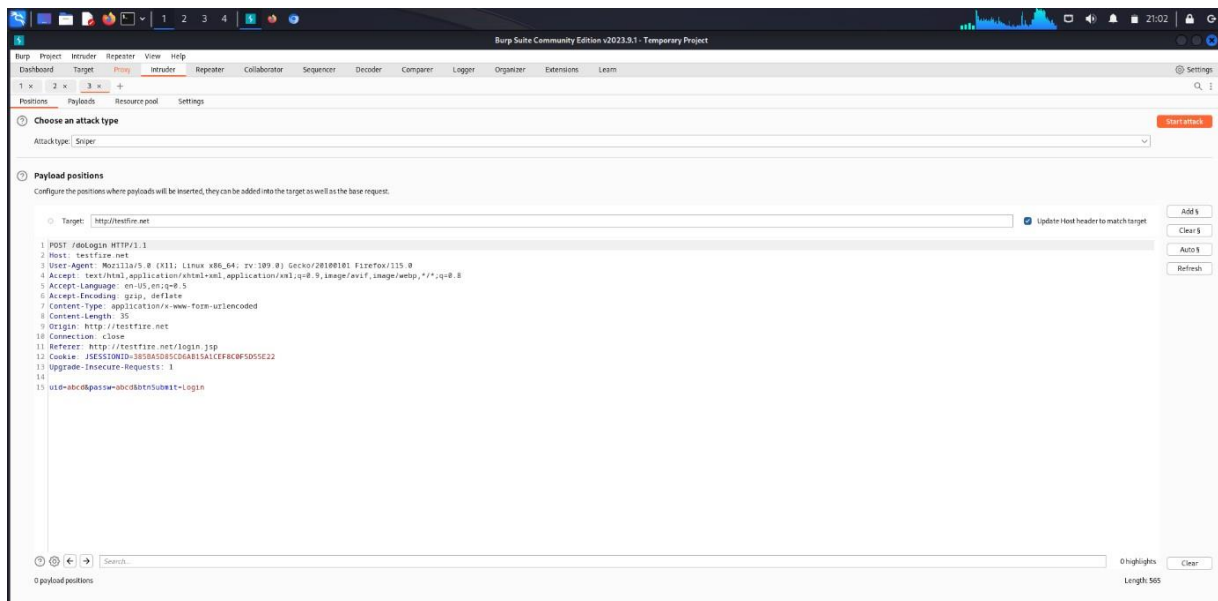


After entering the correct details, we got the source of the information below.

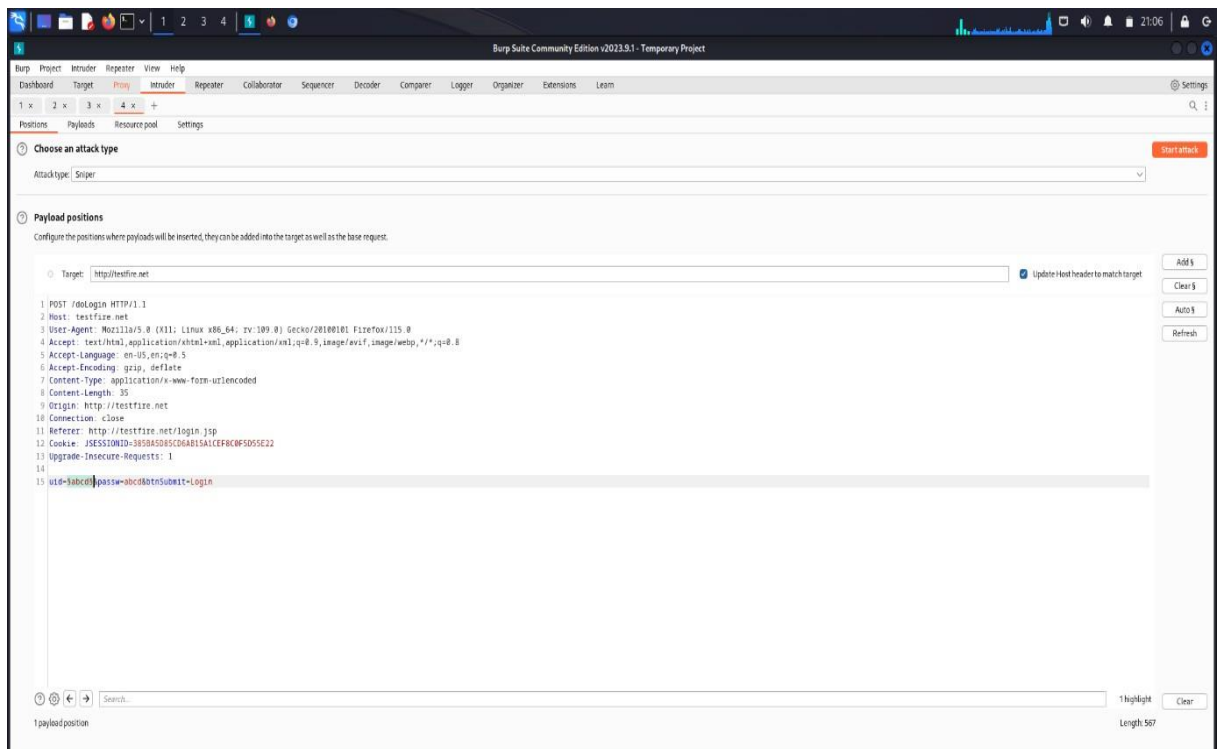


## SQL INJECTION

SQL injection is a technique where a malicious code is injected in a website which leads to hacking of a web page and destroying of database of the website.



SQL injection attack is performed in the intruder tab. For this, the above code was used. The username is added as an element.



In the payloads section, paste the payload code which was copied from github. After this, click on start attack button.



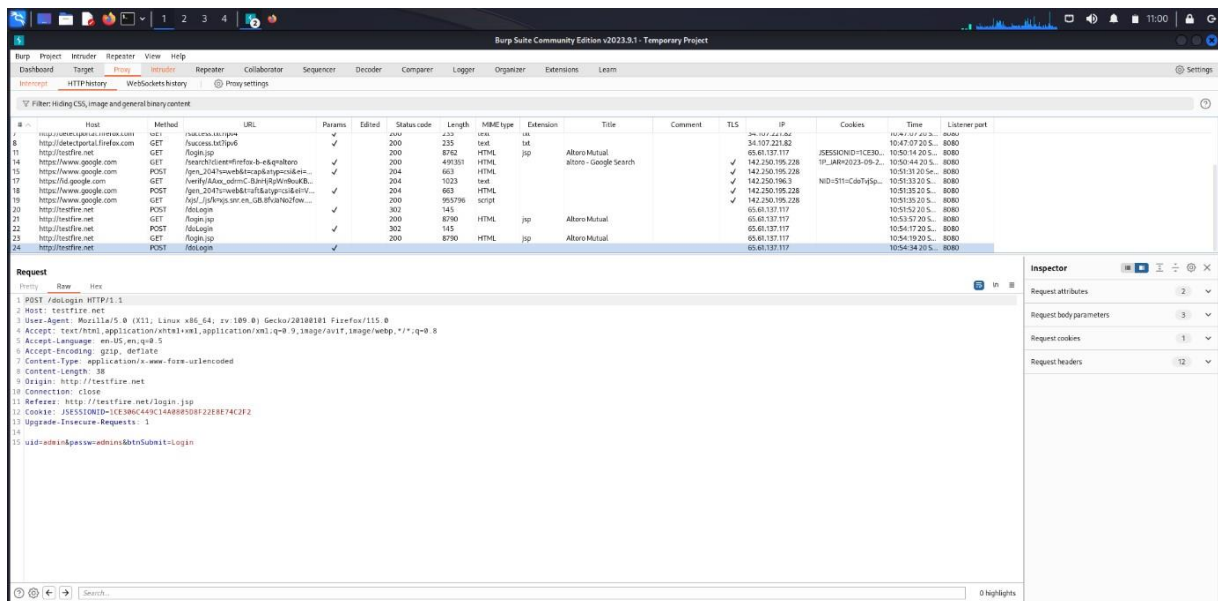
This is the history of the websites searched.

Burp Suite Community Edition v2023.9.1 - Temporary Project														
Dashboard Target Proxy HTTP history WebSockets history Sequencer Decoder Comparer Logger Organizer Extensions Learn														
Filter: Hiding CSS, image and general binary content														
#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Comment	TLS	IP	Time
1	https://content-services.mozilla.c...	GET	/v1/files			200	1923	JSON				✓	34.107.220.239	18:09:13 195s...
2	https://content-signature-2.dh...	GET	/chains/remote-settings/content-signat...			200	5875	script	chain			✓	34.160.144.191	18:09:13 195s...
3	https://firefox.settings.services...	GET	/v1/buckets/main/collections/ms-lang...			200	940	JSON				✓	34.143.100.209	18:09:19 195s...
6	https://firefox.net	GET	/			200	9621	HTML		Altroz-Mutual		✓	65.65.137.177	18:09:21 195s...
15	https://firefox.net	GET	/			200	9621	HTML		Altroz-Mutual		✓	65.65.137.177	18:09:32 195s...
16	https://normandy.cdn.mozilla.net	GET	/api/v1/			200	1208	JSON				✓	35.201.103.21	18:09:36 195s...
17	https://firefox.settings.services...	GET	/v1/buckets/monitor/collections/change...		✓	200	23714	JSON				✓	34.149.100.209	18:09:36 195s...
18	https://services.addons.mozilla.o...	GET	/api/v1/addons/search/?q=firefox-defau...		✓	200	13245	JSON				✓	65.6.112.7	18:09:36 195s...
19	https://pubs.services.mozilla.com	GET	/api/v1/classify_client/			101	245					✓	34.177.68.55	18:09:37 195s...
20	https://classify-client.services.m...	GET	/api/v1/classify_client/			200	326	JSON				✓	34.98.75.36	18:09:37 195s...
21	https://versioncheck.kg.addons...	GET	/update/VersionCheck.php?version=...			200	1886	JSON				✓	34.160.144.191	18:09:38 195s...
22	https://aws5.mozilla.org	GET	/update/IGMP115.1.0/230230724240...		✓	200	1409	XML	xml			✓	35.244.181.201	18:09:39 195s...
23	https://aws5.mozilla.org	GET	/update/SystemAddon115.1.0/2023...		✓	200	471	XML	xml			✓	35.244.181.201	18:09:40 195s...
24	https://content-signature-2.dh...	GET	/chains/normandy-content-signature.m...		✓	304	169		chain			✓	34.160.144.191	18:09:40 195s...
25	https://firefox.settings.services...	GET	/v1/buckets/monitor/collections/change...		✓	200	23714	JSON				✓	34.149.100.209	18:09:40 195s...
26	https://content-signature-2.dh...	GET	/chains/us-content-signature-mozill...		✓	304	169		chain			✓	34.160.144.191	18:09:40 195s...
27	https://firefox.settings.services...	GET	/v1/buckets/main/collections/defaults...		✓	200	2216	JSON				✓	34.149.100.209	18:09:41 195s...
28	https://content-signature-2.dh...	GET	/chains/remote-settings/content-signat...		✓	304	169		chain			✓	34.160.144.191	18:09:43 195s...
29	https://firefox.settings.services...	GET	/v1/buckets/monitor/collections/normandy...		✓	200	45409	JSON				✓	34.149.100.209	18:09:44 195s...
30	https://firefox.settings.services...	GET	/v1/buckets/main/collections/cookie-ba...		✓	200	5227	JSON				✓	34.149.100.209	18:09:46 195s...
31	https://content-signature-2.dh...	GET	/chains/normandy-content-signature.m...		✓	304	169		chain			✓	34.160.144.191	18:09:47 195s...
32	https://beaver.services.mozilla.c...	POST	/download/clientnewclient-auto-flow...		✓	200	206	text				✓	54.185.54.63	18:09:50 195s...
33	https://firefox.settings.services...	GET	/v1/buckets/main/collections/ppi/ppi...		✓	200	18507	JSON				✓	34.149.100.209	18:09:50 195s...
34	https://aws5.mozilla.org	GET	/update/IGMP115.1.0/230230724240...		✓	200	1408	XML	xml			✓	35.244.181.201	18:10:03 195s...
35	https://classify-client.services.m...	GET	/api/v1/classify_client/			200	326	JSON				✓	34.98.75.36	18:10:23 195s...
36	https://firefox.settings.services...	GET	/v1/buckets/monitor/collections/change...		✓	200	23713	JSON				✓	34.149.100.209	18:10:27 195s...
37	https://firefox.settings.services...	GET	/v1/buckets/blocklist/collections/adobe...		✓	200	2302	JSON				✓	34.149.100.209	18:10:38 195s...
38	https://firefox.settings.services...	GET	/v1/buckets/security-state/collections...		✓	200	41214	JSON				✓	34.149.100.209	18:10:39 195s...
39	https://content-signature-2.dh...	GET	/chains/remote-content-signature-mozill...		✓	304	169		chain			✓	34.160.144.191	18:10:39 195s...
40	https://firefox.settings.services...	GET	/v1/buckets/main/collections/cookie-ba...		✓	200	3423	JSON				✓	34.149.100.209	18:10:40 195s...
41	https://tools.github.com	POST	/github/collect		✓	204	609					✓	140.82.112.21	18:10:40 195s...
42	https://api.github.com	POST	/_private/browserstats		✓	200	1087	text				✓	20.207.73.85	18:10:36 195s...
43	https://api.github.com	POST	/_private/browserstats		✓	200	1087	text				✓	20.207.73.85	18:10:02 195s...
44	https://api.github.com	POST	/_private/browserstats		✓	200	1087	text				✓	20.207.73.85	18:10:02 195s...
45	https://github.com	GET	/payloadbox/ig-injection-payload-list			200	32876	HTML		GitHub - payloadbox...		✓	20.207.73.82	18:10:02 195s...
46	https://github.com	GET	/payloadbox/ig-injection-payload-list			200	330773	HTML		GitHub - payloadbox...		✓	20.207.73.82	18:10:03 195s...
47	https://api.github.com	POST	/_private/browserstats		✓	200	1087	text				✓	20.207.73.85	18:10:05 195s...
48	https://github.githubassets.com	GET	/assets/vendor-bundle_modules_github...			200	156478	script	js			✓	185.199.111.154	18:10:08 195s...
50	https://github.githubassets.com	GET	/assets/github-elements-6480e5d4fa...			200	40740	script	js			✓	185.199.111.154	18:10:08 195s...
51	https://github.githubassets.com	GET	/assets/element-negtmty-28097fa2f6...			200	49947	script	js			✓	185.199.111.154	18:10:08 195s...
54	https://github.githubassets.com	GET	/assets/behaviors-56a237e416d9...			200	224066	script	js			✓	185.199.111.154	18:10:10 195s...
55	https://github.com	GET	/payloadbox/ig-injection-payload-list...			302	3058	HTML	svg			✓	20.207.73.82	18:10:10 195s...
56	https://github.githubassets.com	GET	/assets/confirmonly-8bae0038111...			200	8554	script	js			✓	185.199.111.154	18:10:10 195s...
57	https://github.githubassets.com	GET	/assets/vp-runtime-3c4bf2a22b7a...			200	33333	script	js			✓	185.199.111.154	18:10:10 195s...
58	https://github.githubassets.com	GET	/assets/webpack-cdn-070565d9...			200	12380	script	js			✓	185.199.111.154	18:10:10 195s...
59	https://github.githubassets.com	GET	/images/modules/siteicons/funding_...			200	1555	XML	svg			✓	185.199.111.154	18:10:10 195s...
60	https://camo.githubusercontent...	GET	/f997ce5c996a2c78ca442a29f904f...			200	3741	XML				✓	185.199.108.133	18:10:11 195s...

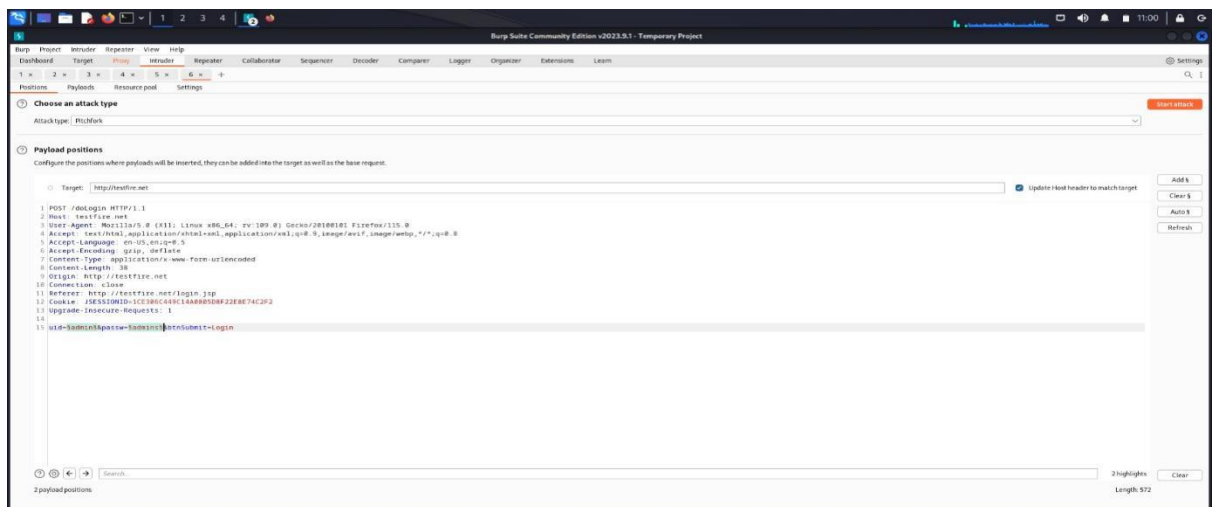
## CREDENTIAL BYPASSING

Credential bypassing is an attack where the attacker does not have the credentials but enters the website by cracking the credentials.

First of all, go to the desired website and turn on the intercept in burpsuite under proxy tab. Under HTTP history, we can see what all websites we have searched for. Right click on the code and send it to the intruder.

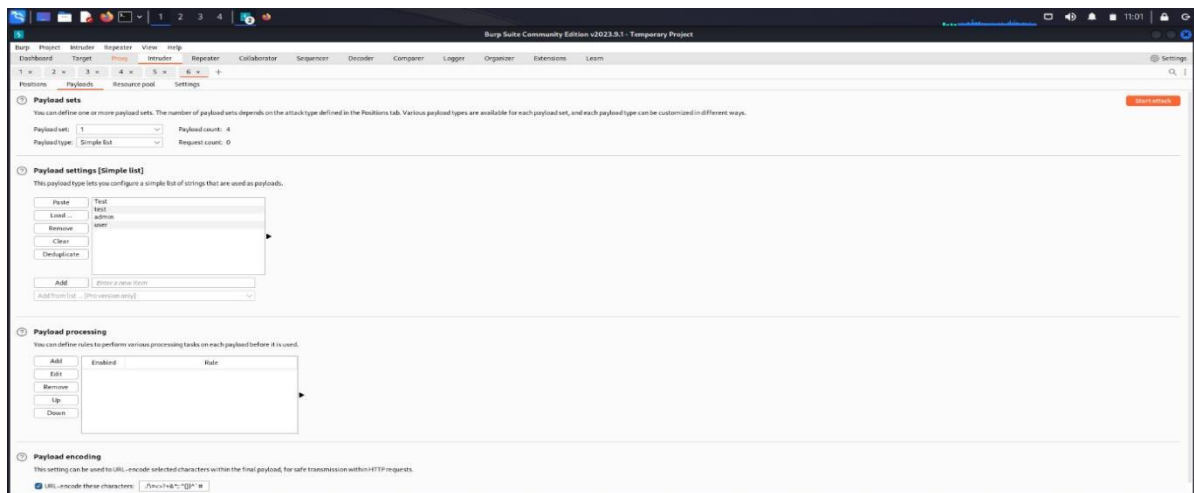


Under intruder tab, select the attack type as pitchfork. Then add the username and password as elements.



Insert the list of usernames and passwords and start the attack.





Among the 4 credentials entered, only one credential has the length of 264 which means it is the correct one.

