



Name of Capstone Project : -
Burp-Suite

- By Vikram Singh

Agenda

- Introduction of Burp-Suite.

- Types of Burp Suite
- Module of Burp suite

- 1. Dashboard

- 2. Target

- 3. Proxy

- 4. Repeater

- 5. Intruder

- 6. Sequencer

- 7. Decoder

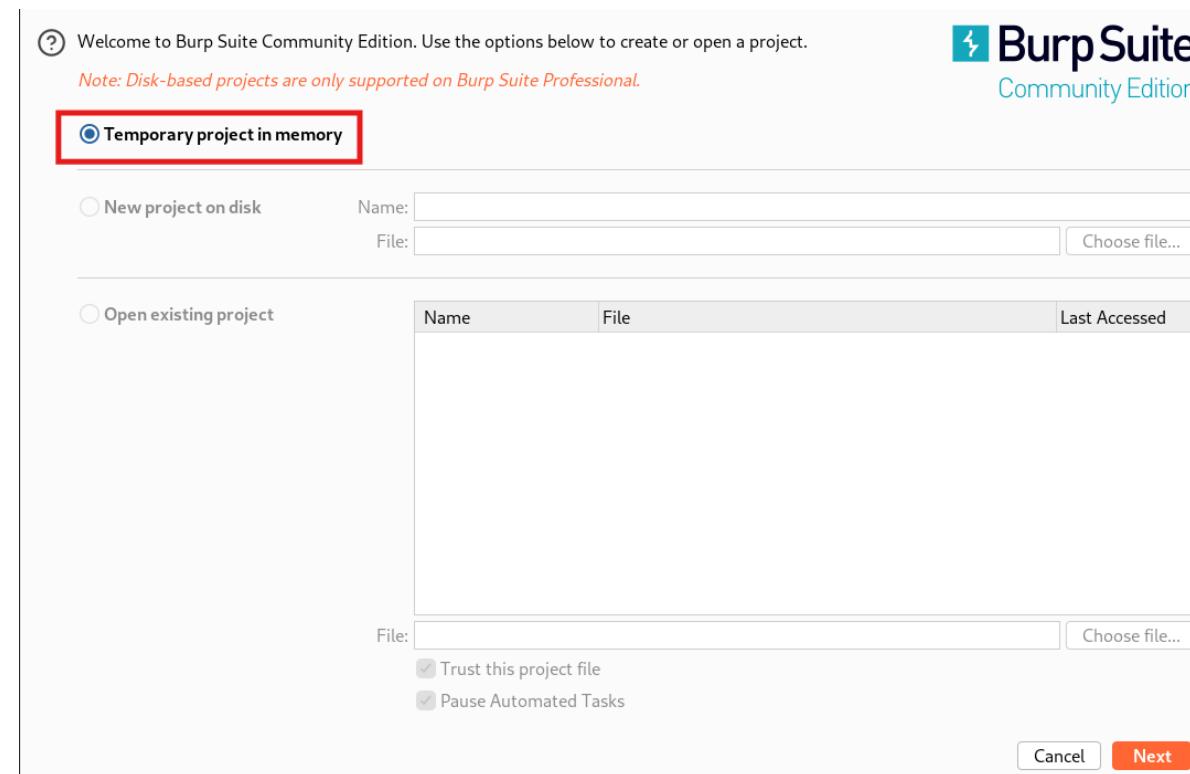
- 8. Comparer

Introduction of Burp-Suite

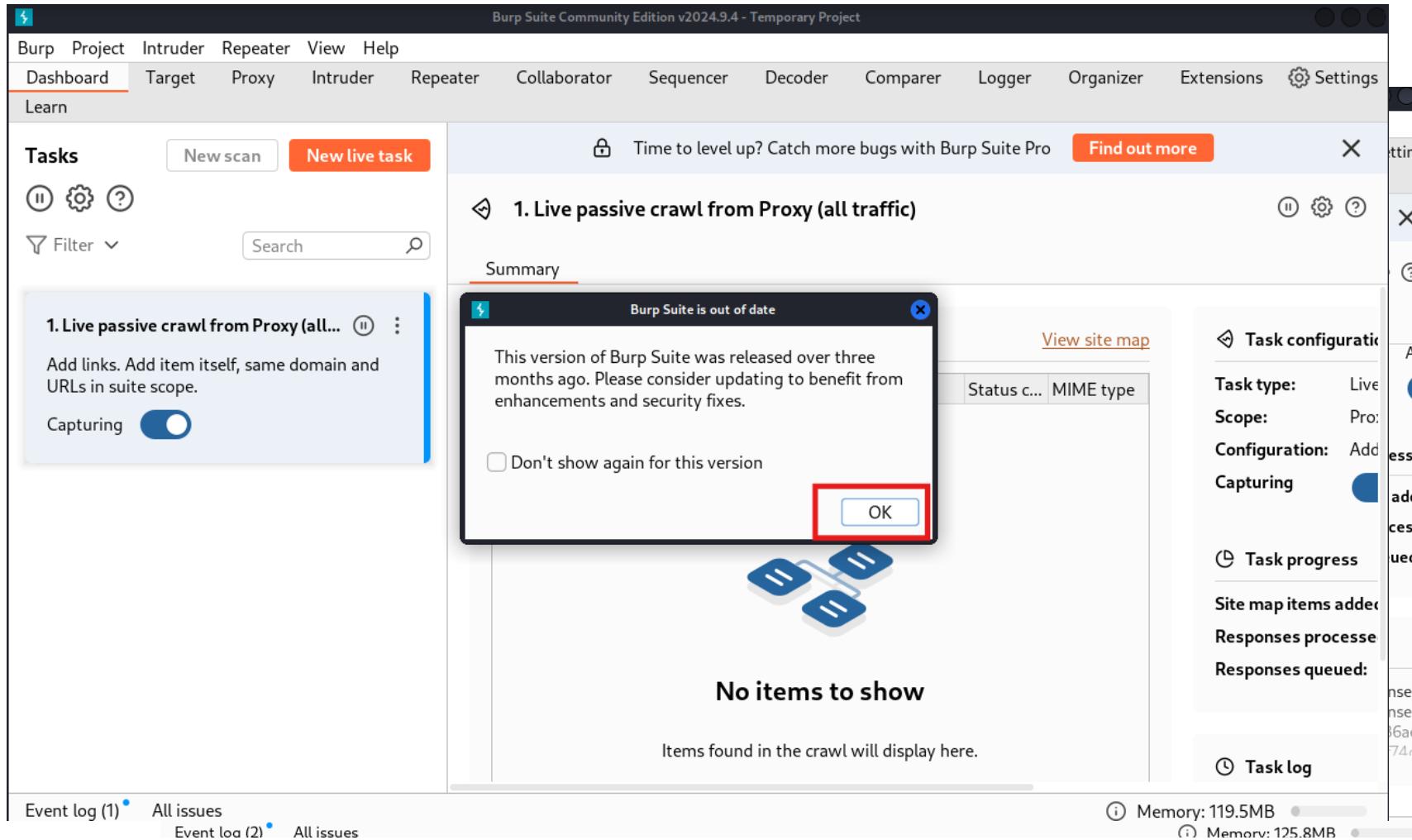
- What is Burp-Suite ?
 - Burp-Suite is a popular cybersecurity tool used for web application testing and find the vulnerability and exploit it.
 - Burp-Suite is known as a Proxy tool its intercept the request between client and server PCs, and It manipulate the request and response by capturing the packet.
 - It is written in Java and Developed by PortSwigger Security.
 - It is available for all Operation Systems like : - Windows, Linux, MacOS.
- **Burp-Suite has three version:**
 1. **Community Edition (Free)** : Basic manual testing tools, includes an Intercepting proxy, repeater, decoder, and comparer.. NO automated scanning or advance tools. Suitable for beginners and learning purpose.
 2. **Professional Edition (Paid)** : Includes everything in the Community Edition, Automated vulnerability scanner for finding security flaws. Intruder tool for brute force, fuzzing, and parameter testing. Pricing : **\$449** usd per user/year
 3. **Enterprise Edition** : It Designed for large-scale automated security testing. Role based access control for teams. **Pricing : \$6995** per year, depending on the number of users.
- **Download links :-**
 - Burp-Suite Community Edition : <https://portswigger.net/burp/releases/professional-community-2025-1-1>
 - Burp-Suite Professional Edition : <https://portswigger.net/burp/releases/professional-community-2025-1-1>
 - Burp-Suite Enterprise Edition : <https://portswigger.net/burp/releases/enterprise-edition-2025-1-1>

I am using Burp-suit Community edition in this Project.

- Temporary project in memory : It create a temporary project in the Community edition Disk based project support in professional edition.



1. Dashboard :- It provides an overview of running tasks, scan results, and vulnerability findings.



2. Target : - The Target tab gives a overview of the target.

- **Site map** : - Its displays a hierarchical view of all domains and endpoints discovered during testing. Allow users to analyze request/response structures, parameters, and vulnerabilities and helps in identifying attack surfaces.

The screenshot shows the Burp Suite interface with the following details:

- Menu Bar:** Burp, Project, Intruder, Repeater, View, Help.
- Toolbar:** Dashboard, Target (highlighted with a red box), Proxy, Repeater, Intruder, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, Settings.
- Sub-Menu:** Extensions, Learn.
- Current Tab:** Site map (highlighted with a red box).
- Section:** Scope, Issue definitions.
- Filter:** Site map filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders.
- Targets:** http://detectportal.firefox.com, http://testphp.vulnweb.com.
- Table:** Host, Method, URL, Params, Length, MIME type, Title, Notes.

Host	Method	URL	Params	Length	MIME type	Title	Notes
http://detectportal.fir...	GET	/canonical.html		298	XML		
http://detectportal.fir...	GET	/success.txt					
http://detectportal.fir...	GET	/success.txt?ipv4	✓	216	text		
http://detectportal.fir...	GET	/success.txt?ipv6	✓	216	text		

- Panels:** Request, Response, Inspector.
- Request Panel:** Shows a list of requests with line numbers and content.
- Response Panel:** Shows a list of responses with line numbers and content.
- Inspector Panel:** Shows sections for Request attributes, Request headers, and Response headers.
- Bottom Status:** Event log (10), All issues, Memory: 135.6MB.

- **Scope** : - In the scope management define which URLs should be included or excluded from testing. Avoid unnecessary or out-of-scope endpoints (3rd party services). Prevents accidental testing of production environments.

The screenshot shows the Burp Suite interface with the 'Target' tab selected. Below it, the 'Scope' tab is highlighted with a red box. The main content area is titled 'Target scope' and contains instructions: '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.' A checkbox for 'Use advanced scope control' is present. The 'Include in scope' section lists a single entry: 'https://portswigger.net/burp/documentation/desktop/tools/proxy'. The 'Exclude from scope' section is empty. At the bottom, there are tabs for 'Event log' and 'All issues', and a status bar showing 'Memory: 155.1MB'.

Burp Suite Community Edition v2025.1.1 - Temporary Project

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

Add	Enabled	Prefix	Include subdomains
	<input checked="" type="checkbox"/>	https://portswigger.net/burp/documentation/desktop/tools/proxy	

Exclude from scope

Add	Enabled	Prefix	Include subdomains

Event log All issues Memory: 155.1MB

- **Scope Definitions** : - It marks security issues like XSS, SQL injection, Broken Authentication etc.

Burp Suite Community Edition v2025.1.1 - Temporary Project

Target Proxy Repeater Intruder Collaborator Sequencer Decoder Comparer Logger Organizer Settings

Dashboard Site map Scope Issue definitions

Issue definitions

This listing contains the definitions of all issues that can be detected by Burp Scanner.

Name	Typical severity	Type index
OS command injection	High	0x00100100
SQL injection	High	0x00100200
SQL injection (second order)	High	0x00100210
ASP.NET tracing enabled	High	0x00100280
File path traversal	High	0x00100300
XML external entity injection	High	0x00100400
LDAP injection	High	0x00100500
XPath injection	High	0x00100600
XML injection	Medium	0x00100700
ASP.NET debugging enabled	Medium	0x00100800
Broken access control	Information	0x00100850
HTTP PUT method is enabled	High	0x00100900
Out-of-band resource load (HTTP)	High	0x00100a00
File path manipulation	High	0x00100b00
PHP code injection	High	0x00100c00
Server-side JavaScript code injection	High	0x00100d00
Perl code injection	High	0x00100e00
Ruby code injection	High	0x00100f00
Python code injection	High	0x00100f10
Expression Language injection	High	0x00100f20
Unidentified code injection	High	0x00101000
Server-side template injection	High	0x00101080

attack even in the event that an attacker circumvents the input validation defenses.

References

- [Web Security Academy: OS command injection](#)

Vulnerability classifications

- [CWE-77: Improper Neutralization of Special Elements used in a Command \('Command Injection'\)](#)
- [CWE-78: Improper Neutralization of Special Elements used in an OS Command \('OS Command Injection'\)](#)
- [CWE-116: Improper Encoding or Escaping of Output](#)
- [CAPEC-248: Command Injection](#)

Typical severity

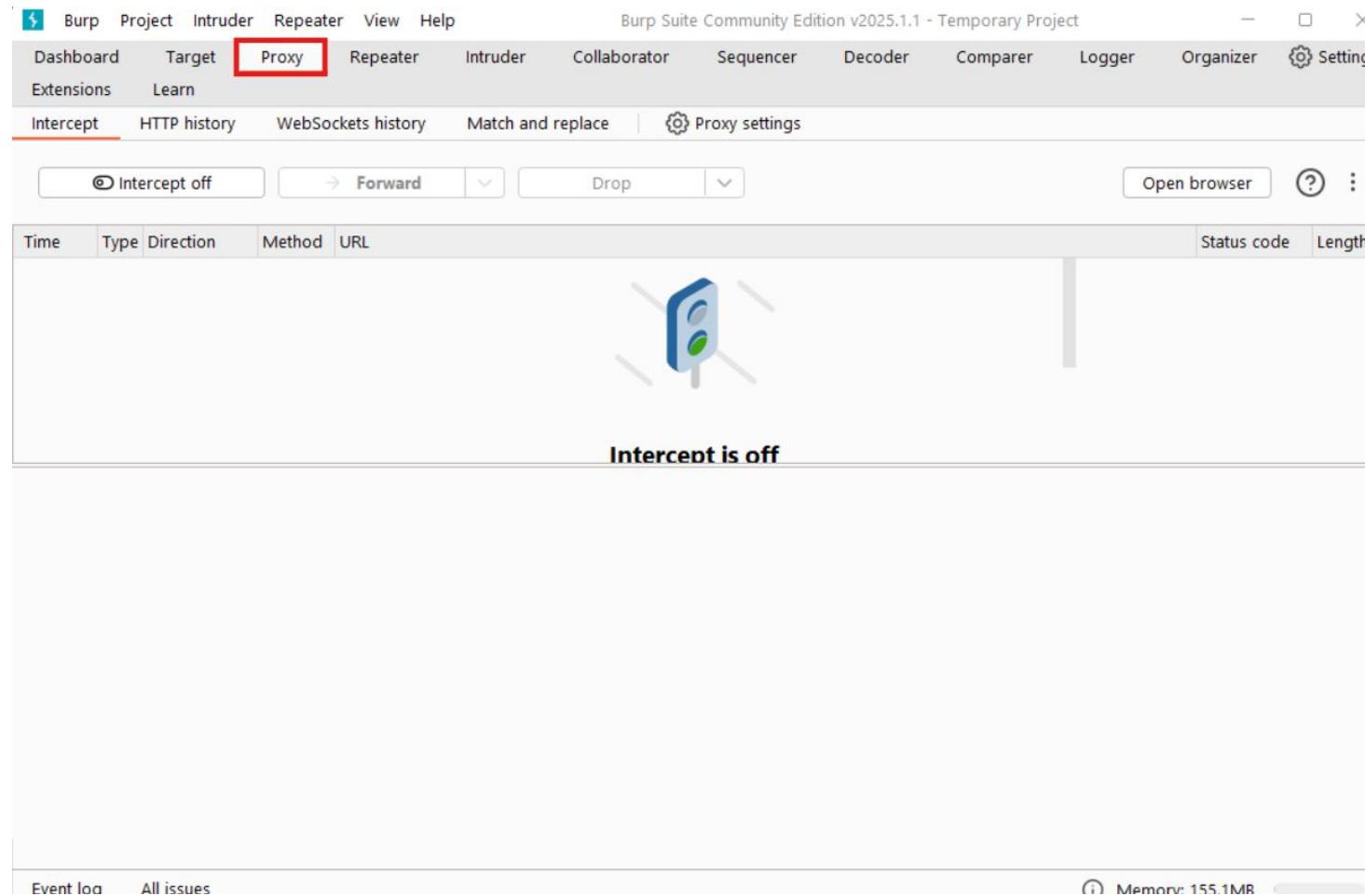
High

Type index

0x00100100

Event log All issues Memory: 155.1MB

3. Proxy : - The Proxy is core feature of the burp-suite. It operates as a web proxy server, and sits as a man-in-the-middle between your browser and destination web servers. Its intercept, modify and analyze HTTP(s) traffic between the browser and the server.



- **Intercept** : - It intercepts HTTP(s) Request & Response.

- Captures traffic between the browser and web server.
- Allows manual modification before forwarding the request/response.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. Below it, the 'Intercept' tab is also highlighted with a red box. The main pane displays a list of network requests captured by the proxy. The first request is highlighted with a blue bar and labeled 'Intercept on'. The status bar at the bottom indicates 'Request to https://www.googletag...'. The table columns are Time, Type, Direction, Method, URL, Status code, and Length.

Time	Type	Direction	Method	URL	Status code	Length
00:21:1...	HT...	→ Request	GET	https://www.googletagmanager.com/gtag/js?id=AW-11422135271		
00:21:1...	HT...	→ Request	POST	https://ps.piwik.pro/ppms.php		
00:21:1...	HT...	→ Request	GET	https://tags.srv.stackadapt.com/sa.jpeg		
00:21:1...	HT...	→ Request	GET	https://tags.srv.stackadapt.com/saq_pxl?uid=SEk-Q5_UkYJF2mT9DoZeJQ&is_js=true&landing_url=https%3A%2F...		
00:21:2...	HT...	→ Request	GET	https://tags.srv.stackadapt.com/js_tracking?url=https%3A%2F%2Fportswigger.net%2Fburp%2Fdocumentation%...		

This screenshot shows the 'Request' and 'Inspector' panes of Burp Suite. The 'Request' pane displays the details of a selected GET request to 'https://www.googletagmanager.com/gtag/js?id=AW-11422135271'. The 'Inspector' pane on the right provides detailed information about the request, including attributes, query parameters, body parameters, cookies, and headers. The status bar at the bottom shows 'Memory: 156.1MB'.

Request

Pretty Raw Hex

```
1 GET /gtag/js?id=AW-11422135271 HTTP/2
2 Host: www.googletagmanager.com
3 Sec-Ch-Ua-Platform: "Windows"
4 Accept-Language: en-US,en;q=0.9
5 Sec-Ch-Ua: "Chromium";v="133", "Not(A:Brand";v="99"
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
7 Sec-Ch-Ua-Mobile: ?0
8 Accept: */
9 Sec-Fetch-Site: cross-site
10 Sec-Fetch-Mode: no-cors
11 Sec-Fetch-Dest: script
12 Sec-Fetch-Storage-Access: active
13 Referer: https://portswigger.net/
```

Event log All issues

Inspector

- Request attributes: 2
- Request query parameters: 1
- Request body parameters: 0
- Request cookies: 0
- Request headers: 16

Memory: 156.1MB

- **HTTP history** : - It contain the history of the web application.
 - It stores all captured request and response for the later analyze.
 - Helps track user interactions, authentication tokens, and API requests.

The screenshot shows the Burp Suite Community Edition interface. The title bar reads "Burp Suite Community Edition v2025.1.1 - Temporary Project". The menu bar includes "Burp", "Project", "Intruder", "Repeater", "View", and "Help". The top navigation bar has tabs for "Dashboard", "Target", "Proxy" (which is highlighted with a red box), "Repeater", "Intruder", "Collaborator", "Sequencer", "Decoder", "Comparer", "Logger", "Organizer", and "Settings". Below this is a secondary navigation bar with "Extensions", "Learn", "Intercept" (highlighted with a red box), "HTTP history" (highlighted with a red box), "WebSockets history", "Match and replace", and "Proxy settings". A filter bar at the top says "Filter settings: Hiding CSS, image and general binary content". The main content area is a table titled "HTTP history" with columns: #, Host, Method, URL, Params, Edited, Status code, Length, MIME type, Extension, and Title. The table contains 16 rows of captured requests. At the bottom, there are links for "Event log" and "All issues", and a status bar showing "Memory: 152.0MB".

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title
1	http://detectportal.firefox.co...	GET	/canonical.html			200	298	XML	html	
2	http://detectportal.firefox.co...	GET	/success.txt?ipv6		✓	200	216	text	txt	
3	http://detectportal.firefox.co...	GET	/success.txt?ipv4		✓	200	216	text	txt	
9	http://detectportal.firefox.co...	GET	/canonical.html			200	298	XML	html	
10	http://wpad	GET	/wpad.dat							dat
11	http://detectportal.firefox.co...	GET	/canonical.html			200	298	XML	html	
12	http://detectportal.firefox.co...	GET	/canonical.html			200	298	XML	html	
13	http://detectportal.firefox.co...	GET	/success.txt?ipv4		✓	200	216	text	txt	
14	http://detectportal.firefox.co...	GET	/success.txt?ipv6		✓	200	216	text	txt	
15	http://detectportal.firefox.co...	GET	/canonical.html			200	298	XML	html	
16	http://detectportal.firefox.co...	GET	/success.txt?inv4		✓	200	216	text	txt	

- WebSockets History : - Intercept WebSocket messages in real-time.
 - Useful for testing real-time application(e.g., chat apps, live notifications).

Burp Suite Community Edition v2025.1.1 - Temporary Project

Burp Project Intruder Repeater View Help

Proxy Repeater Intruder Collaborator Sequencer Decoder Comparer Logger Organizer Settings

Dashboard Target Intercept HTTP history WebSockets history Match and replace Proxy settings



WebSockets history is empty

This displays the history of all WebSocket traffic sent between Burp's browser and your target applications. WebSockets provide long-lived connections for streaming data and other asynchronous traffic.

Learn more Open browser

Event log All issues Memory: 152.0MB

- Match and replace :-

- It automatically modify headers, cookies, or request parameters.
- Example: replace User-Agent headers for user-agent spoofing.

The screenshot shows the Burp Suite interface with the following details:

- Top Navigation Bar:** Burp, Project, Intruder, Repeater, View, Help. The "Proxy" tab is highlighted with a red box.
- Sub-Menu Bar:** Dashboard, Target, **Proxy**, Repeater, Intruder, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, Settings.
- Sub-Sub-Menu Bar:** Intercept, HTTP history, WebSockets history, **Match and replace**, Proxy settings.
- Content Area:**
 - HTTP match and replace rules:** A section with a question mark icon. It contains a note: "Use these settings to automatically replace parts of HTTP requests and responses passing through the Proxy." Below the note is a checkbox labeled "Only apply to in-scope items". To its right is a vertical toolbar with buttons: Add, Edit, Remove, Up, Down.
 - WebSocket match and replace rules:** A section with a question mark icon. It contains a note: "Use these settings to automatically replace parts of WebSocket messages passing through the Proxy." Below the note is a checkbox labeled "Only apply to in-scope items". To its right is a vertical toolbar with buttons: Add, Edit, Remove, Up, Down. Below this is a table with columns: Enabled, Direction, Match, Replace, Type, Comment.
- Bottom Status Bar:** Event log, All issues, Memory: 152.6MB.

- **Proxy Settings** : - In the proxy setting we can add, edit and remove the particular interface which is necessary to intercept the traffic.
 - In this tab we can Import/export CA certificate. CA certificate that proxy listeners can use negotiating TLS (Transport layer security) connection.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. The 'Proxy listeners' section displays a single running listener on port 8080. Below this, a note about CA certificates is present, along with 'Import / export CA certificate' and 'Regenerate CA certificate' buttons. The 'Request interception rules' section shows a rule for intercepting requests based on file extension.

Proxy listeners

Running	Interface	Invisible	Redirect	Certificate	TLS
<input checked="" type="checkbox"/>	127.0.0.1:8080				

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating TLS connections. You can import or regenerate this certificate as required.

Request interception rules

Intercept requests based on the following rules:

Add	Enabled	Operator	Match type	Relationship	Condition
<input checked="" type="checkbox"/>		File extension	Does not match	(^gif\$ ^jpg\$ ^png\$ ^css\$ ^js\$)	
<input type="checkbox"/>	Or	Request	Contains parameters		
<input type="checkbox"/>	Or	HTTP method	Does not match	(get post)	
<input type="checkbox"/>	And	URL	Is in target scope		

- Send request to the Repeater to the proxy.

The screenshot shows the Burp Suite Community Edition interface. The title bar reads "Burp Suite Community Edition v2025.1.1 - Temporary Project". The menu bar includes Burp, Project, Intruder, Repeater, View, and Help. The top navigation bar has tabs for Dashboard, Target, **Proxy**, Repeater, Intruder, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, and Settings. Below the tabs are Intercept, HTTP history, WebSockets history, Match and replace, and Proxy settings. A toolbar below the tabs includes Intercept on, Forward, Drop, Request to http://testphp.vulnweb.com..., Open browser, and a help icon. The main pane displays a table of network traffic with columns for Time, Type, Direction, Method, URL, Status code, and Length. A single row is selected, showing a GET request to "http://testphp.vulnweb.com/login.php". A context menu is open over this row, listing options: Add to scope, Forward, Drop, Add notes, Highlight, Don't intercept requests, Do intercept, Scan, Send to Intruder, **Send to Repeater** (which is highlighted with a red box), Send to Sequencer, Send to Organizer, Send to Comparer, Request in browser, Request attributes, Request query parameters, Request body parameters, Request cookies, and Request headers. The bottom of the interface includes a search bar, a notes section, and memory usage information ("Memory: 161.3MB").

4. Repeater : -

The Repeater tool in Burp-suite is used for manual testing of HTTP(s) requests. It allows security testers to modify and resend individual requests multiple times to analyze how a web application responds.

Key features :-

- Modify and resend Requests** – Edit HTTP headers, parameters, cookies, or body contents.
- Compare Responses** – View differences between multiple request attempts.
- Support for WebSockets**– Modify and resend WebSocket messages.
- Raw, Hex, and Render Views** – Analyze responses in different formats, including rendering as a web page.

Burp Suite Community Edition v2025.1.1 - Temporary Project

Target: <http://testphp.vulnweb.com> | HTTP/1

Repeater

Request

Pretty Raw Hex Render

```
1 GET /login.php HTTP/1.1
2 Host: testphp.vulnweb.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:135.0) Gecko/20100101 Firefox/135.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://testphp.vulnweb.com/
8 Connection: keep-alive
9 Upgrade-Insecure-Requests: 1
10 Priority: u=0,i
11
12
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.19.0
3 Date: Sun, 16 Feb 2025 19:36:18 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: keep-alive
6 X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+
7 Content-Length: 5523
8
9 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
10 "http://www.w3.org/TR/html4/loose.dtd">
11 <html>
12   <!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php"
13 codeOutsideHTMLIsLocked="false" -->
14   <head>
15     <meta http-equiv="Content-Type"
content="text/html;
charset=iso-8859-2">
16
17   <!-- InstanceBeginEditable name="document_title_rgn" -->
```

Inspector

Request attributes 2

Request query parameters 0

Request body parameters 0

Request cookies 0

Request headers 9

Response headers 6

Notes

Done

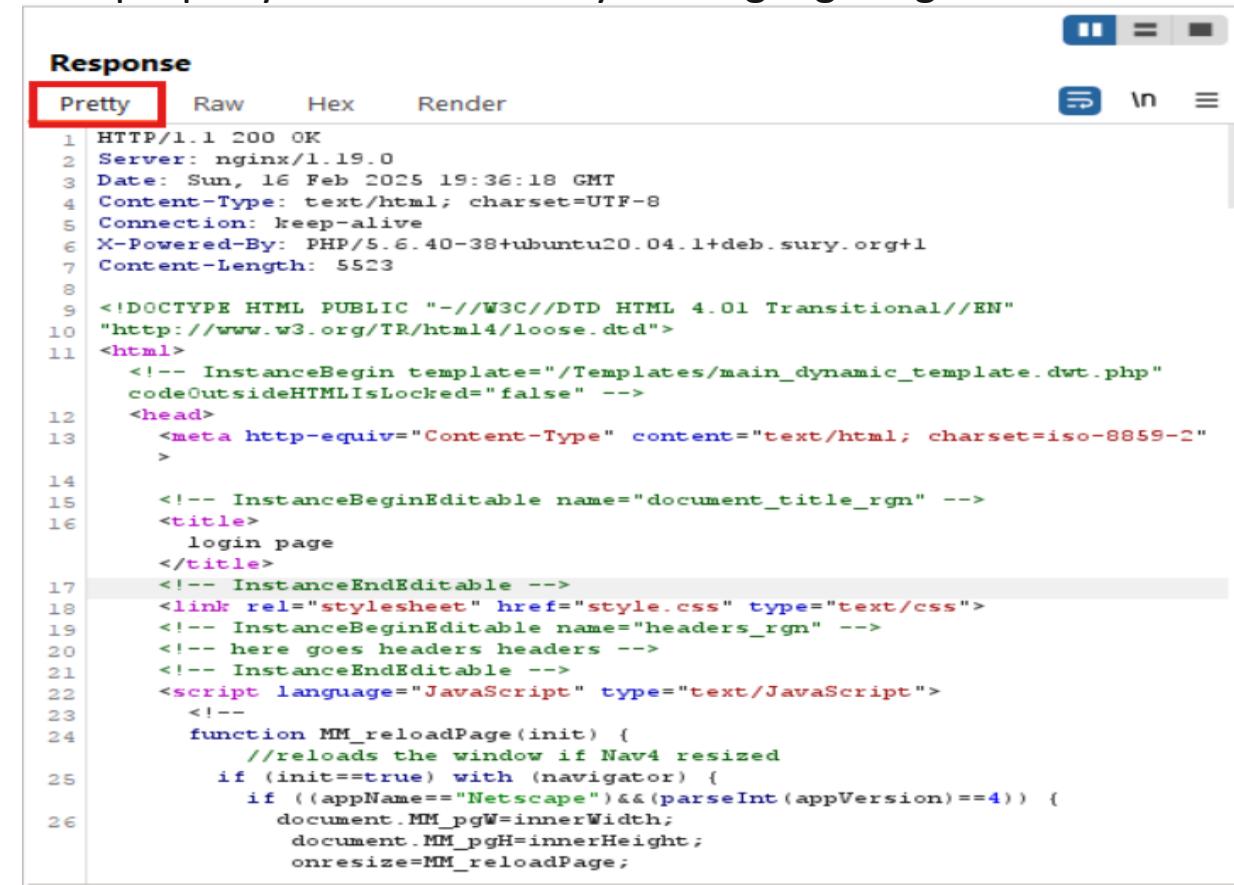
Event log (1) All issues

5,745 bytes | 310 millis

Memory: 162.7MB

➤ Pretty view :-

- Formats and color-codes the response for better readability.
- Works best for **JSON** (JavaScript object notation), **XML** (Extensible markup Language) , **HTML** (Hyper Text Markup Language).
- Helps in quickly understanding API (Application Programming Interface) response and error messages.
- Example: A JSON response will be properly indented with syntax highlighting.



The screenshot shows the 'Response' tab of a browser developer tools window. The 'Pretty' tab is selected, highlighted with a red box. The content is an HTML document with line numbers on the left. Syntax highlighting is applied to various elements: blue for tags like <html>, <head>, <title>, <link>, <script>, and <function>; green for attributes like href="style.css" and type="text/css"; red for content-type="text/html; charset=iso-8859-2"; and purple for comments like <!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php". The code includes standard HTTP headers and an embedded JavaScript function.

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.19.0
3 Date: Sun, 16 Feb 2025 19:36:18 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: keep-alive
6 X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+l
7 Content-Length: 5523
8
9 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
10 "http://www.w3.org/TR/html4/loose.dtd">
11 <html>
12   <!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php"
13   codeOutsideHTMLIsLocked="false" -->
14   <head>
15     <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2"
16   >
17   <!-- InstanceBeginEditable name="document_title_rgn" -->
18   <title>
19     login page
20   </title>
21   <!-- InstanceEndEditable -->
22   <link rel="stylesheet" href="style.css" type="text/css">
23   <!-- InstanceBeginEditable name="headers_rgn" -->
24   <!-- here goes headers headers -->
25   <!-- InstanceEndEditable -->
26   <script language="JavaScript" type="text/JavaScript">
27     <!--
28       function MM_reloadPage(init) {
29         // reloads the window if Nav4 resized
30         if (init==true) with (navigator) {
31           if ((appName=="Netscape") && (parseInt(appVersion)==4)) {
32             document.MM_pgW=innerWidth;
33             document.MM_pgH=innerHeight;
34             onresize=MM_reloadPage;
```

➤ Raw View :-

- Displays the response exactly as received from the server.
- Shows full HTTP Headers, response body, and status codes.
- Useful for debugging HTTP headers, cookies, and redirection.

Use case: Checking for security Headers (e.g., CSP (Content Security Policy), CORS (Cross Origin Resource Sharing), HSTS (HTTP Strict Transfer Security) and debugging raw HTTP response.

```
Response
Pretty Raw Hex Render
14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
<!-- InstanceBeginEditable name="document_title_rgn" -->
<title>login page</title>
<!-- InstanceEndEditable -->
<link rel="stylesheet" href="style.css" type="text/css">
<!-- InstanceBeginEditable name="headers_rgn" -->
<!-- here goes headers headers -->
<!-- InstanceEndEditable -->
<script language="JavaScript" type="text/JavaScript">
<!--
function MM_reloadPage(init) { // reloads the window if Nav4 resized
  if (init==true) with (navigator) (if ((appName=="Netscape")&&(parseInt(
  appVersion)==4)) {
    document.MM_pgW=innerWidth; document.MM_pgH=innerHeight; onresize=
    MM_reloadPage; })
  else if (innerWidth!=document.MM_pgW || innerHeight!=document.MM_pgH)
  location.reload();
}
MM_reloadPage(true);
//-->
</script>
</head>
<body>
<div id="mainLayer" style="position:absolute; width:700px; z-index:1">
<div id="masthead">
  <h1 id="siteName"><a href="https://www.acunetix.com/"></a></h1>
  <h2 id="siteInfo">TEST and Demonstration site for <a href="https://www.acunetix.com/vulnerability-scanner/">Acunetix Web Vulnerability Scanner</a></h2>
  <div id="globalNav">
    <table border="0" cellpadding="0" cellspacing="0" width="100%"><tr>
      <td align="left">
        <a href="#">index.php</a> <a href="#">showcase</a> <a href="#">categories.php</a> <a href="#">services.php</a>
      </td>
    </tr>
  </table>
</div>
</div>
</body>
</html>
```

➤ Hex View :-

- Shows the response in hexadecimal format (useful for binary data).
- Displays both hex values and ASCII (American Standard Code for Information Interchange) representations side by side.
- Useful for inspecting file downloads, encryption, and encoding issues.

Use case : Analyzing binary responses like images, compressed files, or encrypted data.

Response						
	Pretty	Raw	Hex	Render		
00000000	48 54 54 50 2f 31 2e 31	20 32 30 30 20 4f 4b 0d	HTTP/1.1 200 OK			
00000010	0a 53 65 72 76 65 72 3a	20 6e 67 69 6e 78 2f 31	Server: nginx/1			
00000020	2e 31 39 2e 30 0d 0a 44	61 74 65 3a 20 53 75 6e	.19.0 Date: Sun			
00000030	2c 20 31 36 20 46 65 62	20 32 30 32 35 20 31 39	, 16 Feb 2025 19			
00000040	3a 33 36 3a 31 38 20 47	4d 54 0d 0a 43 6f 6e 74	:36:18 GMT Cont			
00000050	65 6e 74 2d 54 79 70 65	3a 20 74 65 78 74 2f 68	ent-Type: text/h			
00000060	74 6d 6c 3b 20 63 68 61	72 73 65 74 3d 55 54 46	tml; charset=UTF			
00000070	2d 38 0d 0a 43 6f 6e 6e	65 63 74 69 6f 6e 3a 20	-8 Connection:			
00000080	6b 65 65 70 2d 61 6c 69	76 65 0d 0a 58 2d 50 6f	keep-alive X-Po			
00000090	77 65 72 65 64 2d 42 79	3a 20 50 48 50 2f 35 2e	wered-By: PHP/5.			
000000a0	36 2e 34 30 2d 33 38 2b	75 62 75 6e 74 75 32 30	6.40-38+ubuntu20			
000000b0	2e 30 34 2e 31 2b 64 65	62 2e 73 75 72 79 2e 6f	.04.1+deb.sury.o			
000000c0	72 67 2b 31 0d 0a 43 6f	6e 74 65 6e 74 2d 4c 65	rg+1 Content-Le			
000000d0	6e 67 74 68 3a 20 35 35	32 33 0d 0a 0d 0a 3c 21	ngth: 5523 <!			
000000e0	44 4f 43 54 59 50 45 20	48 54 4d 4c 20 50 55 42	DOCTYPE HTML PUB			
000000f0	4c 49 43 20 22 2d 2f 2f	57 33 43 2f 2f 44 54 44	LIC "-//W3C//DTD			
00000100	20 48 54 4d 4c 20 34 2e	30 31 20 54 72 61 6e 73	HTML 4.01 Trans			
00000110	69 74 69 6f 6e 61 6c 2f	2f 45 4e 22 0a 22 68 74	itional//EN" ht			
00000120	74 70 3a 2f 2f 77 77 77	2e 77 33 2e 6f 72 67 2f	tp://www.w3.org/			
00000130	54 52 2f 68 74 6d 6c 34	2f 6c 6f 6f 73 65 2e 64	TR/html4/loose.d			
00000140	74 64 22 3e 0a 3c 68 74	6d 6c 3e 3c 21 2d 2d 20	td"> <html><!--			
00000150	49 6e 73 74 61 6e 63 65	42 65 67 69 6e 20 74 65	InstanceBegin te			
00000160	6d 70 6c 61 74 65 3d 22	2f 54 65 6d 70 6c 61 74	mplate="/Templat			
00000170	65 73 2f 6d 61 69 6e 5f	64 79 6e 61 6d 69 63 5f	es/main_dynamic_			
00000180	74 65 6d 70 6c 61 74 65	2e 64 77 74 2e 70 68 70	template.dwt.php			
00000190	22 20 63 6f 64 65 4f 75	74 73 69 64 65 48 54 4d	" codeOutsideHTM			
000001a0	4c 49 73 4c 6f 63 6b 65	64 3d 22 66 61 6c 73 65	LIsLocked="false			
000001b0	22 20 2d 2d 3e 0a 3c 68	65 61 64 3e 0a 3c 6d 65	" --> <head> <me			
000001c0	74 61 20 68 74 74 70 2d	65 71 75 69 76 3d 22 43	ta http-equiv="C			
000001d0	6f 6e 74 65 6e 74 2d 54	79 70 65 22 20 63 6f 6e	ontent-Type" con			

➤ Render View :-

- Renders the response as a web page (if it's HTML-based).
- Helps in testing HTML injection, XSS, and DOM-based vulnerabilities.
- Sometime, content may not load properly if external resources are blocked.

Use case : Verifying if XSS payloads successfully execute in a browser-like environment.

The screenshot shows a web browser interface with a 'Response' tab selected. The 'Render' tab is highlighted with a red box. The page content is as follows:

Acunetix website security
TEST and Demonstration site for [Acunetix Web Vulnerability Scanner](#)

[home](#) | [categories](#) | [artists](#) | [disclaimer](#) | [your cart](#) | [guestbook](#) | [AJAX Demo](#)

If you are already registered please enter your login information below

Username :
Password :

You can also [signup here](#).
Signup disabled. Please use the username **test** and the password **test**.

search art

- [Browse categories](#)
- [Browse artists](#)
- [Your cart](#)
- [Signup](#)
- [Your profile](#)
- [Our guestbook](#)
- [AJAX Demo](#)

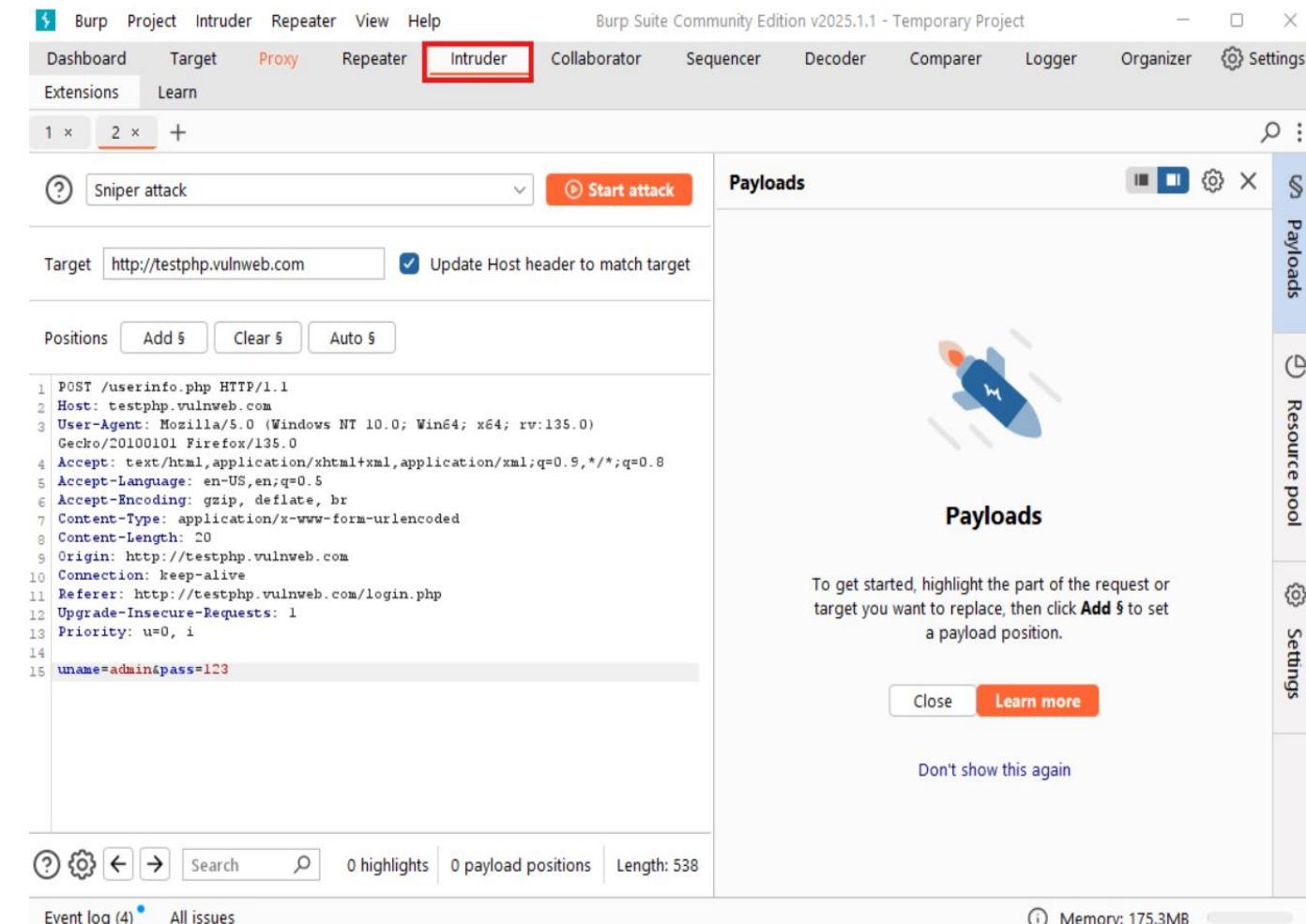
5. Intruder : - The Intruder tool in Burp-Suite is used for automated attacks on web applications. It allows security testers to send multiple requests with different inputs to test for vulnerabilities like Brute Force, SQL Injection, and Parameter fuzzing.

Key Features Of Intruder:-

- **Automated Attacks** – Send multiple payloads to test for vulnerabilities.
- **Four Attack Modes** – Control how inputs are inserted into request.
- **Custom Payloads** – Use wordlists, numbers, or custom payload generators.
- **Results Analysis** – View response status codes, response length, and patterns.
- **Grep Match/Extract** – Find specific keywords or extract data from responses.

In the Intruder we have 4 types of payloads.

- i. Sniper Attack
- ii. Bettering Ram Attack
- iii. Pitchfork Attack
- iv. Cluster Bomb Attack



i. **Sniper Attack** : - The Sniper attack is a simplest mode in Busp Suite Intruder. It tests one input parameter at a time, replacing the marked position with different payloads from a list. This is useful for Fuzzing, testing SQL injection, XSS, and brute-force attacks.

Performing a Brute force attack on Login parameter using sniper attack.

➤ Lets suppose I have a username and I need to brute-force the Password to get login.

- Username = test
- Password = ?
- Website = <http://testphp.vulnweb.com/login.php>

Step 1 : go to the website : - <http://testphp.vulnweb.com/login.php>

The screenshot shows a web browser window with the URL testphp.vulnweb.com/login.php. The page title is "acunetix acuart". The main content area contains a login form with the following fields:

- Username:
- Password:

A red box highlights both the Username and Password input fields. To the right of the form, a message reads: "If you are already registered please enter your login information below:" followed by "Username : test" and "Password : ****". Below the form, another message states: "You can also [signup here](#). Signup disabled. Please use the username **test** and the password **test**."

The left sidebar contains a search bar with "search art" and a "go" button, along with a list of links:

- search art
- go
- Browse categories
- Browse artists
- Your cart
- Signup
- Your profile
- Our guestbook
- AJAX Demo

At the bottom of the sidebar, there is a "Links" section with the following items:

- Security art
- PHP scanner
- PHP vuln help
- Fractal Explorer

Step 2 : On burp-suite capture the request and send it to the Intruder.

The screenshot shows the Burp Suite interface in 'Proxy' mode. A POST request to `http://testphp.vulnweb.com/userinfo.php` is selected in the list of captured requests. The 'Request' tab displays the raw HTTP message, which includes a user-agent header and a parameter `uname=test&pass=1234`. The 'Inspector' tab is open, showing sections for Request attributes, Query parameters, Body parameters, Cookies, and Headers. The 'Notes' tab is also visible. The status bar at the bottom indicates 173.4MB of memory usage.

Image 1

The screenshot shows the same Burp Suite interface as Image 1, but with a context menu open over the selected POST request. The menu options include 'Add to scope', 'Forward', 'Drop', 'Send to Intruder' (which is highlighted in yellow), 'Send to Repeater', 'Send to Sequencer', 'Send to Organizer', 'Send to Comparer', and 'Request in browser'. The status bar at the bottom indicates 175.3MB of memory usage.

Image 2

Step 3 : Click on Intruder tab.

=> Select the Password position where we attempt brute-force.

=> Click on add

=> payload type = simple list

=> Past the guessable password in like in **image 4**

Burp Suite Community Edition v2025.1.1 - Temporary Project

Sniper attack

Target: http://testphp.vulnweb.com

Positions Add \$ Clear \$ Auto \$

```
1 POST /userinfo.php HTTP/1.1
2 Host: testphp.vulnweb.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:135.0)
4 Gecko/20100101 Firefox/135.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 20
10 Origin: http://testphp.vulnweb.com
11 Connection: keep-alive
12 Referer: http://testphp.vulnweb.com/login.php
13 Upgrade-Insecure-Requests: 1
14 Priority: u0, i
15 uname=test&pass=$12345
```

1 highlight | 1 payload position | Length: 540

Image 3

Burp Suite Community Edition v2025.1.1 - Temporary Project

Sniper attack

Target: http://testphp.vulnweb.com

Positions Add \$ Clear \$ Auto \$

```
1 POST /userinfo.php HTTP/1.1
2 Host: testphp.vulnweb.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:135.0)
4 Gecko/20100101 Firefox/135.0
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 20
10 Origin: http://testphp.vulnweb.com
11 Connection: keep-alive
12 Referer: http://testphp.vulnweb.com/login.php
13 Upgrade-Insecure-Requests: 1
14 Priority: u0, i
15 uname=test&pass=$12345
```

Event log (4) | All issues

Memory: 164.4MB

Image 4

Step 4 : Click on Start attack and its appears a new window like image 5

=>After the complete attack.

=> check Status-code, and Length if the Status-code or length is change so check the response of the parameter.

=> Like image 6 checking response.

The screenshot shows the ZAP tool interface after an attack. The title bar says "2. Intruder attack of http://testphp.vulnweb.com". The main area displays a table of attack results:

Request	Payload	Status code	Response...	Error	Timeout	Length	Comment
0		302	670			258	
1	root	302	601			258	
2	!@	302	599			258	
3	wubao	302	639			258	
4	password	302	611			258	
5	123456	302	600			258	
6	admin	302	594			258	

Below the table, there are tabs for "Request" and "Response". The "Response" tab is selected, showing the following raw response:

```
1 POST /userinfo.php HTTP/1.1
2 Host: testphp.vulnweb.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:135.0) Gecko/20100101 Firefox/135.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;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: 20
9 Origin: http://testphp.vulnweb.com
```

At the bottom, there are buttons for "Attack", "Save", and "Settings". A search bar at the bottom right shows "0 highlights".

Image 5

This screenshot shows the same ZAP interface as Image 5, but with specific rows highlighted in yellow. The first row (Request 13) and the last row (Request 5) are highlighted. The highlighted rows show the following data:

Request	Payload	Status code	Response...	Error	Timeout	Length	Comment
13	test	200	593			6266	
0		302	670			258	
1	root	302	601			258	
2	!@	302	599			258	
3	wubao	302	639			258	
4	password	302	611			258	
5	123456	302	600			258	

The "Response" tab is selected, showing the raw response for Request 13:

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.19.0
3 Date: Mon, 17 Feb 2025 10:15:26 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: keep-alive
6 X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
7 Set-Cookie: login=test%2Ftest
8 Content-Length: 6013
9
```

At the bottom, there are buttons for "Attack", "Save", and "Settings". A search bar at the bottom right shows "0 highlights".

Image 6

Step 5 : Username = test, Password = test

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo

search art go

Browse categories

Browse artists

Your cart

Signup

Your profile

Our guestbook

AJAX Demo

Links

Security art

PHP scanner

PHP vuln help

Fractal Explorer

If you are already registered please enter your login information below:

Username: Password:

You can also signup here.
Signup disabled. Please use the username **test** and the password **test**.

Image 7

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo Logout test

search art go

Browse categories

Browse artists

Your cart

Signup

Your profile

Our guestbook

AJAX Demo

Links

Security art

PHP scanner

PHP vuln help

Fractal Explorer

Muhammad Hamza (test)

On this page you can visualize or edit your user information.

Name:

Credit card number:

E-Mail:

Phone number:

Address:

You have 1 items in your cart. You visualize your cart [here](#).

Image 8

II. Battering Ram Attack :-

- It replaces all marked positions in the request with the same payload in each request.
- Useful for testing repeated tokens, session IDs, or authentication fields where the same values is used in multiple places.
- In this we can choose multiple position.

Step 1 : select the Battering ram attack

⇒ Select the position

⇒ Add Password list.

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. The 'Payloads' panel is open, showing a 'Simple list' payload type with a list of credentials:

Position	Value
1	root
2	!@
3	wubao
4	password
5	123456
6	admin
7	12345
8	123456
9	admin
10	123456

The 'Positions' field in the main panel is set to 'Add \$'. The 'Payload position' dropdown is set to 'All payload positions'. The 'Payload type' dropdown is set to 'Simple list'. The 'Payload count' is 20, and the 'Request count' is 20. The 'Payload configuration' section notes that this type lets you configure a simple list of strings that are used as payloads.

Step 2 : Click on Start attack

⇒ Once the attack completed

⇒ Check the Status-code, and length if the Status-code or length is change check the response.

Attack Save 5. Intruder attack of http://testphp.vulnweb.com Attack Save ?

5. Intruder attack of http://testphp.vulnweb.com

Results Positions

Capture filter: Capturing all items Apply capture filter

View filter: Showing all items

Request	Payload	Status code	Response...	Error	Timeout	Length	Comment
0		302	595			258	
1	root	302	595			258	
2	!@	302	597			258	
3	wubao	302	593			258	
4	password	302	604			258	
5	123456	302	599			258	
6	admin	302	599			258	

Request Response

Pretty Raw Hex Render

```
1 HTTP/1.1 302 Found
2 Server: nginx/1.19.0
3 Date: Mon, 17 Feb 2025 10:52:20 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: keep-alive
6 X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+l
7 Location: login.php
8 Content-Length: 14
9
```

?

Search 0 highlights

Finished

Image 1

Attack Save 5. Intruder attack of http://testphp.vulnweb.com Attack Save ?

5. Intruder attack of http://testphp.vulnweb.com

Results Positions

Capture filter: Capturing all items Apply capture filter

View filter: Showing all items

Request	Payload	Status code	Response...	Error	Timeout	Length	Comment
13	test	200	596			6233	
0		302	595			258	
1	root	302	595			258	
2	!@	302	597			258	
3	wubao	302	593			258	
4	password	302	604			258	
5	123456	302	599			258	

Request Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.19.0
3 Date: Mon, 17 Feb 2025 10:52:35 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: keep-alive
6 X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+l
7 Set-Cookie: login=test%2Ftest
8 Content-Length: 5900
9
```

?

Search 0 highlights

Finished

Image 2

III. Pitchfork Attack : - The pitchfork attack type allows you to test multiple parameters simultaneously, but with different values for each parameter. Unlike Sniper (Which tests one position at a time) and Battering Ram (which sends the same value to multiple positions), pitchfork runs Parallel payload lists, assigning a unique value from each list to its corresponding position.

How Pitchfork Works :

- ⇒ Each payload position gets its own separate lists of test values.
- ⇒ Burp takes one value from each list and sends them together in a single request.
- ⇒ The number of requests equals the length of the shortest payload list (unlike Cluster Bomb, which test all combinations)

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. In the main pane, a request is being constructed for a 'Pitchfork attack'. The 'Payloads' section on the right shows a simple list of strings for both 'Payload position 1' and 'Payload type'. The 'Payload configuration' section explains that this type lets you configure a simple list of strings used as payloads. A list of payloads is provided, including 'root', '!@', 'wubao', 'password', '123456', 'admin', and '12345'. The bottom of the screen shows the event log with 4 items and memory usage of 174.8MB.

Burp Suite Community Edition v2025.1.1 - Temporary Project

Intruder

Payloads

Payload position: 1 - test

Payload type: Simple list

Payload count: 20

Request count: 20

Payload configuration

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

Paste
Load...
Remove
Clear
Duplicate
Add
Enter a new item
Add from list... [Pro version only]

Event log (4) All issues

Memory: 174.8MB

• How Pitchfork Attack Works :-

Step 1 :

⇒ Capture the request on burp sent it to the intruder.

⇒ Select attack type : Pitchfork attack

⇒ Select positions uname, pass, => Click on Add\$

⇒ Select Payload position and add wordlist in both position.

E.g. : image 2, and image 3

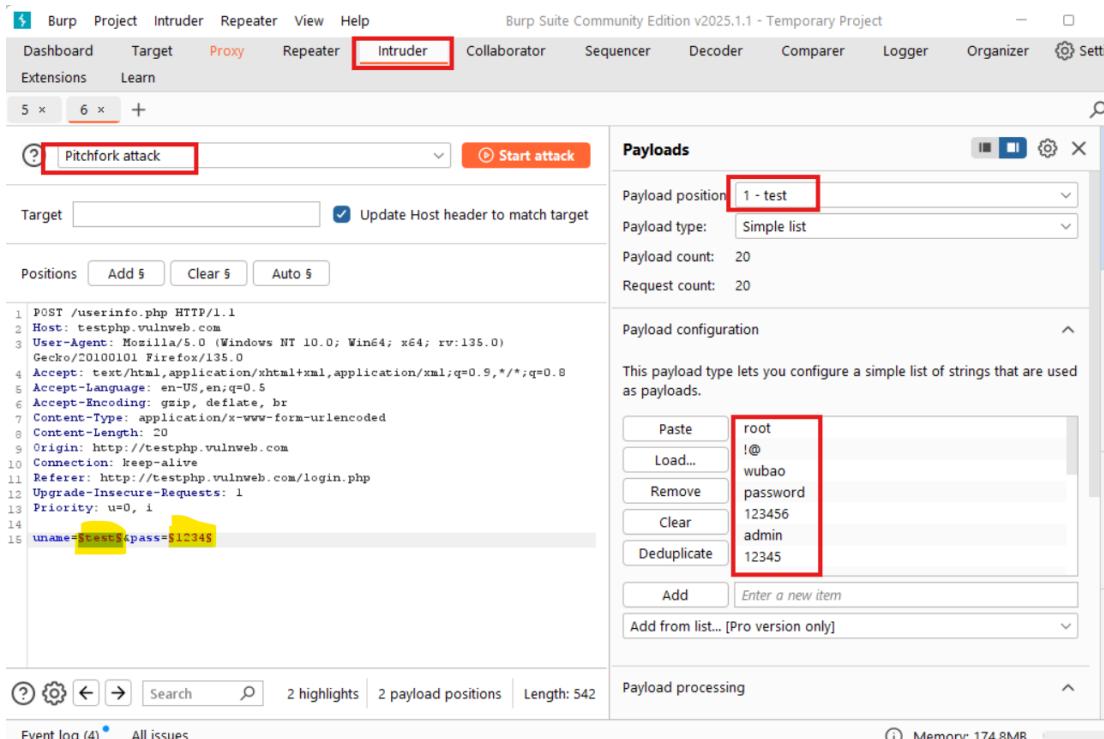


Image 2

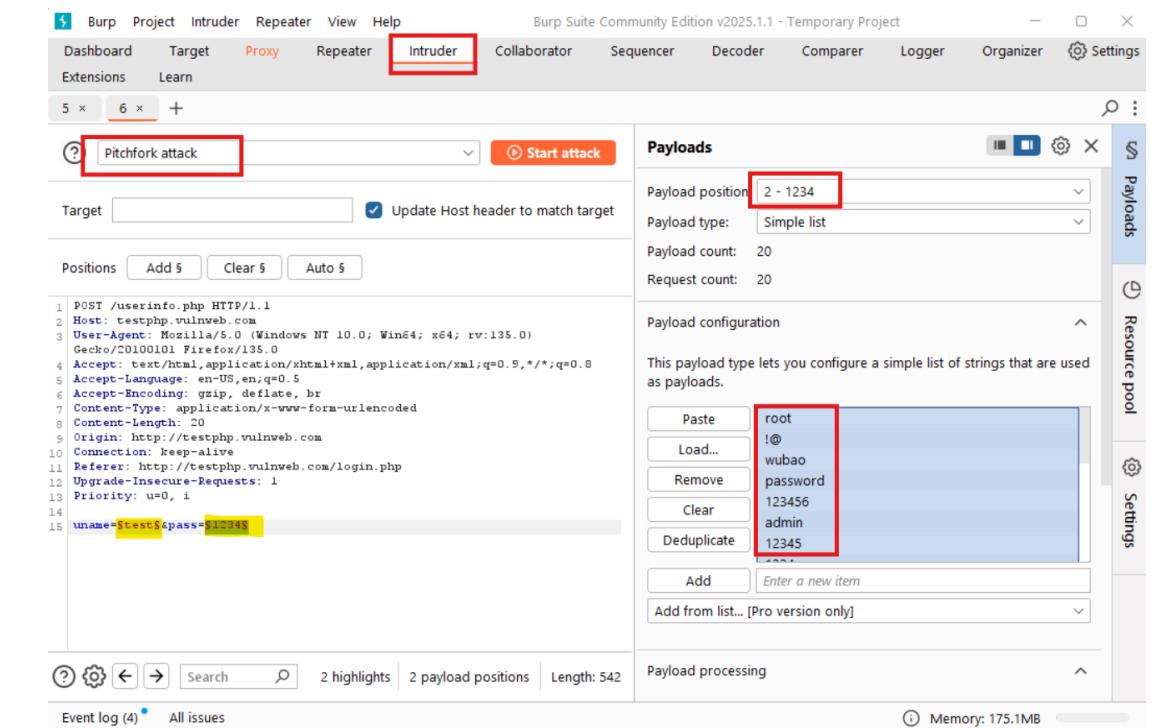


Image 3

- Step 2 :

⇒ Click Start attack

⇒ Image 4 has 302 status code and image 5 has 200 ok status code :

The screenshot shows the "Attack" tab selected in the top bar. The main window title is "6. Intruder attack of http://testphp.vulnweb.com". Below the title, there are two tabs: "Results" (selected) and "Positions". Under "Results", there are two sections: "Capture filter: Capturing all items" and "View filter: Showing all items". The "View filter" section shows a table with columns: Request, Payload 1, Payload 2, Status code, Response, Error, Timeout, Length, and Comm. A row for request 13 has "root" in both Payload 1 and Payload 2, and "302" in the Status code column. The "Response" tab below shows a detailed HTTP response:

```
1 HTTP/1.1 302 Found
2 Server: nginx/1.19.0
3 Date: Tue, 18 Feb 2025 10:29:24 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: keep-alive
6 X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+l
7 Location: login.php
8 Content-Length: 14
9
```

At the bottom, there are navigation icons (back, forward, search, etc.) and a "0 highlights" message.

Image 4

The screenshot shows the "Attack" tab selected in the top bar. The main window title is "6. Intruder attack of http://testphp.vulnweb.com". Below the title, there are two tabs: "Results" (selected) and "Positions". Under "Results", there are two sections: "Capture filter: Capturing all items" and "View filter: Showing all items". The "View filter" section shows a table with columns: Request, Payload 1, Payload 2, Status code, Response, Error, Timeout, Length, and Comm. A row for request 13 has "test" in both Payload 1 and Payload 2, and "200" in the Status code column. The "Response" tab below shows a detailed HTTP response:

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.19.0
3 Date: Tue, 18 Feb 2025 10:29:39 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: keep-alive
6 X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+l
7 Set-Cookie: login=test$!Ctest
8 Content-Length: 6127
9
```

At the bottom, there are navigation icons (back, forward, search, etc.) and a "0 highlights" message.

Image 5

IV. Cluster Bomb Attack :- A Cluster Bomb attack in Burp Suite Intruder is an advance brute-force attack that tests all possible combinations of multiple payloads. Unlike Pitchfork Attack (which runs payload lists in parallel), Cluster Bomb tests every value in one list against every value in another list.

Note :- In the Cluster bomb Attack I am using 20 payload for each position and it make 400 combination to start brut force attack.

The screenshot shows the Burp Suite Community Edition v2025.1.1 interface with the 'Intruder' tab selected. In the 'Payloads' section, 'Payload position 1 - test' is set to 'Simple list' with a payload count of 20 and a request count of 400. The payload list contains 20 items: 'root', '!@', 'wubao', 'password', '123456', 'admin', 'root123', '!', 'lq@w', 'lqaz@wsx', 'idc!', 'admin!@', and 'qwerty'. The 'Payload configuration' panel indicates this type lets you configure a simple list of strings used as payloads. The 'Request' pane shows a POST request to /userinfo.php with various headers and parameters, including 'uname' and 'pass' both set to '123456'. The 'Event log' at the bottom shows 4 issues.

Image 1

This screenshot shows the second payload configuration in the Burp Suite interface. 'Payload position 2 - 1234' is set to 'Simple list' with a payload count of 20 and a request count of 400. The payload list contains 20 items: 'root123', '!', 'lq@w', 'lqaz@wsx', 'idc!', 'admin!@', and 'qwerty'. The 'Payload configuration' panel is identical to the first one. The 'Request' pane shows the same POST request to /userinfo.php with 'uname' and 'pass' both set to '123456'. The 'Event log' at the bottom shows 4 issues.

Image 1

6. Sequencer : - Burp Suite Sequencer is a tool to analyze the randomness (entropy) of the session tokens, CSRF tokens, and other pseudo-random values used in web applications. It helps determine whether an application's tokens are predictable and vulnerable to attacks.

- **How to use Burp Suite Sequencer :-**

Website = <https://ginandjuice.shop>

Creds = carlos/hunter2

Taks = capture tokens

Step 1 : Capture Tokens

=> Open Burp suite and go to the Sequencer tab.

=> Click “start Capture” and select where to capture randomness from:

- Live Interception (capturing session tokens from request).
- Manual input (copy-pasting a set of tokens).
- From a Request (capturing tokens via automated request).

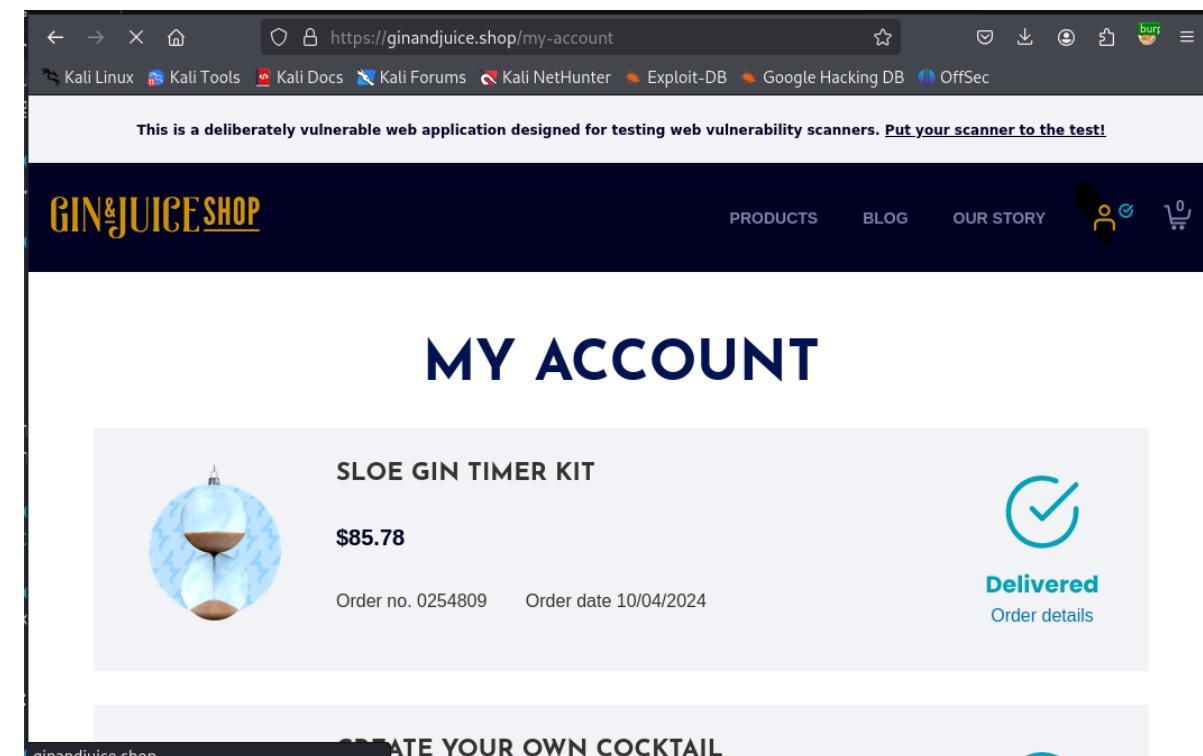


Image 1

Step 2 : Capture cookie and sending towards sequencer.

In image 2

The screenshot shows the Burp Suite interface with the 'Intercept' tab selected. A context menu is open over a selected request to <https://ginandjuice.shop/my-account>. The 'Send to Sequencer' option is highlighted with a red box.

image 2

Step 3 : click on Sequencer tab (image 3)

The screenshot shows the Burp Suite interface with the 'Sequencer' tab selected. A request to <https://ginandjuice.shop/my-account> is selected in the list. The 'Start live capture' button is highlighted with a red box.

Image 3

Step 4 : click on start live capture (image 4)

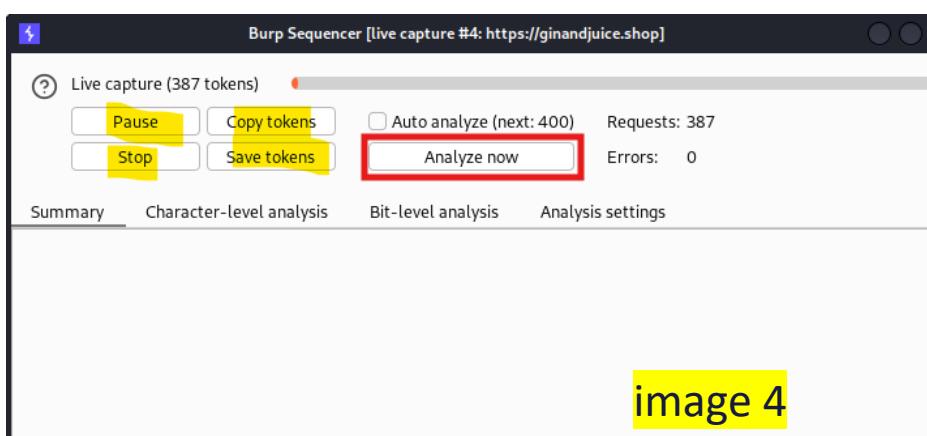
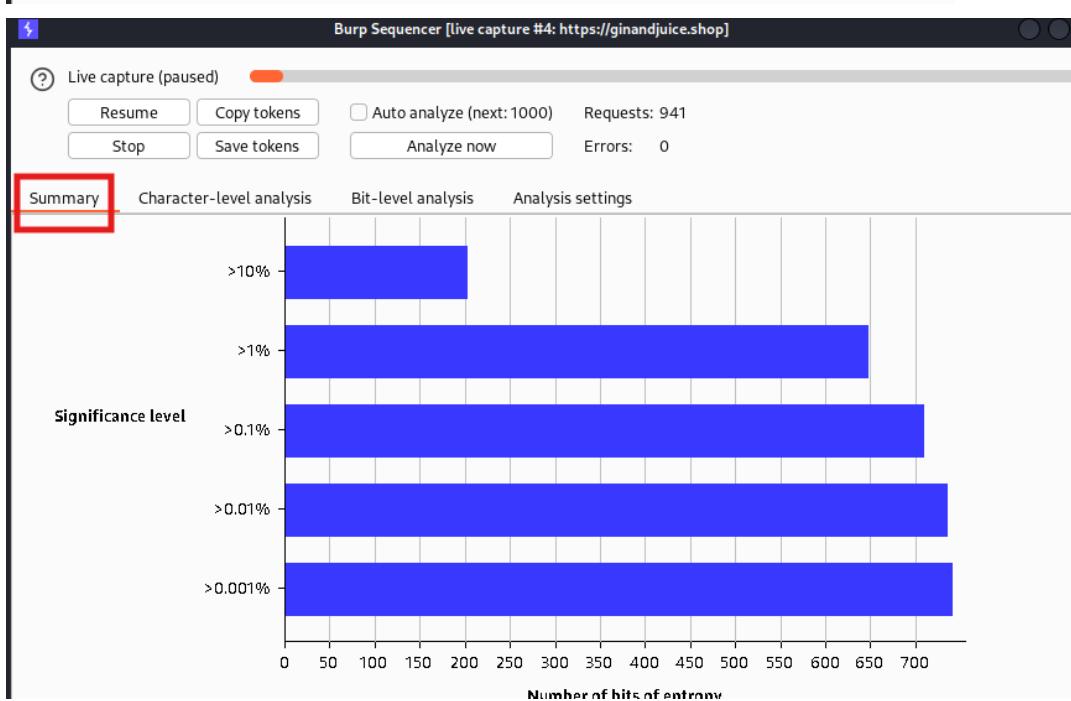


image 4

Step 5 : click on analyze now and see (image 4)



(image 4)

7. Decoder :-

- Burp suite Decoder is a tool used to encode, decode, and transform data between different formats. It helps penetration testers analyze encoded payloads, obfuscated data, and encrypted values used in web applications.
- It can encode and decode format like : URL, HTML, Base64, ASCII hex, Hex, Octal, Binary, and Gzip.

⇒Encoded to URL

<script>alert(1)</script>

Image 7.1

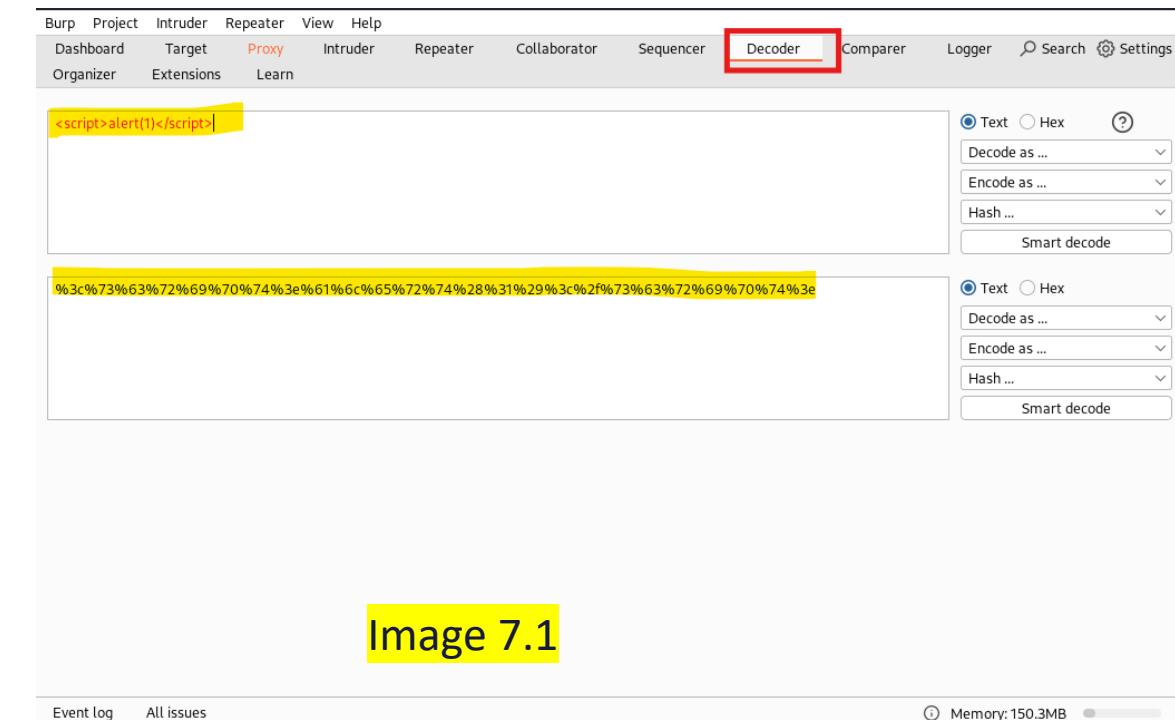


Image 7.1

8. Comparer :- Burp Suite Comparer is a tool used to compare two pieces of data (request, Response, tokens, etc) to identify differences and similarities. It is helpful for analyzing session tokens authentication responses, and code changes in web applications.

How to use Comparer :-

Step 1: Capture the request in Proxy, then send it to the repeater.

see image 1 and 2

The screenshot shows the Burp Suite interface. The top navigation bar has tabs for Burp, Project, Intruder, Repeater, View, Help, and Proxy. The Proxy tab is highlighted with a red box. Below the tabs are buttons for Dashboard, Target, Comparer, Logger, Intruder, Repeater, Collaborator, Sequencer, Decoder, and Search. Under the Target section, there are buttons for Intercept, HTTP history, WebSockets history, Match and replace, and Proxy settings. The main pane displays a list of captured requests. One specific request is highlighted with a red box: "https://demo.testfire.net/bank/showAccount?listAccounts=800001". The Request and Response panes below show the details of this selected request. A context menu is open over the selected request, with the "Send to Repeater" option highlighted with a yellow box. At the bottom, there are buttons for Event log, All issues (11), and a key indicator for Modified, Deleted, and Added items.

Image 1

The screenshot shows the Burp Suite interface with the Repeater tab highlighted with a red box. The main pane displays a captured request and response. The Request pane shows a GET request to "/bank/showAccount?listAccounts=800001" with various headers and a cookie. The Response pane shows the corresponding JSON response. To the right, the Inspector tab is highlighted with a blue box. The Inspector pane contains sections for Request attributes, Request query parameters, Request body parameters, Request cookies, Request headers, and Response headers. The Notes section on the far right is also visible. At the bottom, there are buttons for Event log, All issues (11), and a memory usage indicator.

Image 1

Step 2 : then send the request to the repeater after that modify some content and send again to the comparer to compare both request.

=> we can compare in words and Bytes

Note : If have change 800001 to 899999999

you can see the comparison in image 4

Burp Suite Professional v2024.11.2 - Temporary Project - Licensed to :: SIDDHARTH SANGWAN

Burp Project Intruder Repeater View Help

Proxy Intruder Repeater Collaborator Sequencer Decoder Search Settings

Comparer Logger Organizer Extensions Learn

#	Length	Data
1	768	GET /bank/showAccount?listAccounts=800001
2	771	GET /bank/showAccount?listAccounts=899999999

Paste Load Remove Clear

Select item 2:

#	Length	Data
1	768	GET /bank/showAccount?listAccounts=800001
2	771	GET /bank/showAccount?listAccounts=899999999

Compare ...

Words
Bytes

Event log All issues (11) • Memory: 165.7MB

Image 3

Word compare of #1 and #2 (1 difference)

Length: 768 Length: 771

Text Hex Text Hex

GET /bank/showAccount?listAccounts=800001 HTTP/1.1	GET /bank/showAccount?listAccounts=899999999 HTTP/1.1
Host: demo.testfire.net	Host: demo.testfire.net
Cookie: JSESSIONID=13C702EB0F3D264EBC16AC7D372BEB52; AltOr	Cookie: JSESSIONID=13C702EB0F3D264EBC16AC7D372BEB52; AltOr
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/2010010	User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/2010010
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image
Accept-Language: en-US,en;q=0.5	Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br	Accept-Encoding: gzip, deflate, br
Referer: https://demo.testfire.net/bank/main.jsp	Referer: https://demo.testfire.net/bank/main.jsp
Upgrade-Insecure-Requests: 1	Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document	Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate	Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin	Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1	Sec-Fetch-User: ?1
Priority: u=0, i	Priority: u=0, i
Te: trailers	Te: trailers
Connection: keep-alive	Connection: keep-alive

Key: Modified Deleted Added

Sync views

Image 4

Questions ?

Thank You!