iGNITE Technologies

BURPSUITE FOR PENTESTER

LOGGER++

www.hackingarticles.in

Contents

Burp Logger++: A Powerful Extension 3
Setting Up & Navigating 3
Navigating 4
Query-Based Filter 6
Magical Filter 8
Filter Library 14
Regex-Based Filter 17
Export Data Feature 20
Conclusion 24

Burp Logger++: A Powerful Extension

In this article, we'll learn about a powerful Burp Extension cool tool called "Burp Logger++". It is like a super detective for websites, always on the lookout for any hidden problems. It is an extra feature that you can add to Burp, which lots of web experts use to find issues on websites.

Suppose you are a web explorer, and you want to know everything about a website. Burp Logger++ is like your trusty notebook. It is super helpful because it has a magical filter. You can tell it what kind of information you are looking for, and it will only show you those things.

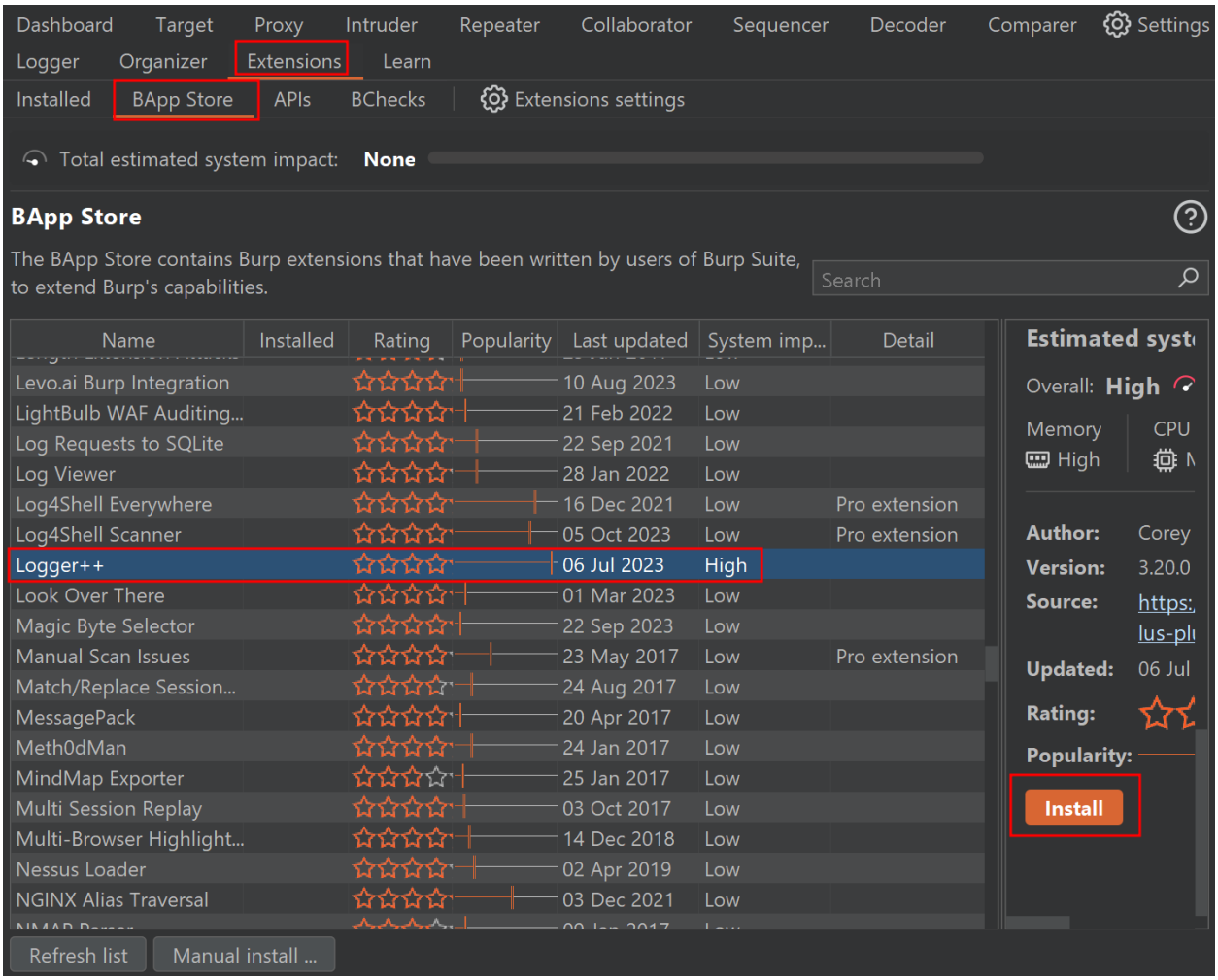
With Burp Logger++, you can also color-code things. Think of it like using different colors to highlight the most important parts of a picture. This helps you spot the important stuff quickly.

- Setting Up & Navigating
- Query-Based Filter
- Filter Library
- Regex-Based Filter
- Export Data Feature

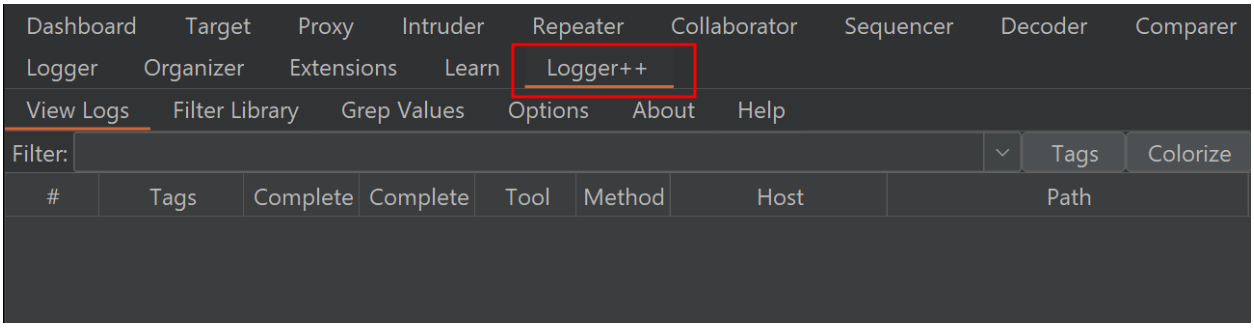
Setting Up & Navigating

You can download and install the extension from the BApp Store. Go to Extensions > Bapp Store. Here, search for Logger++ or simply scroll down.

Click on it, on the right side scroll down and install it.

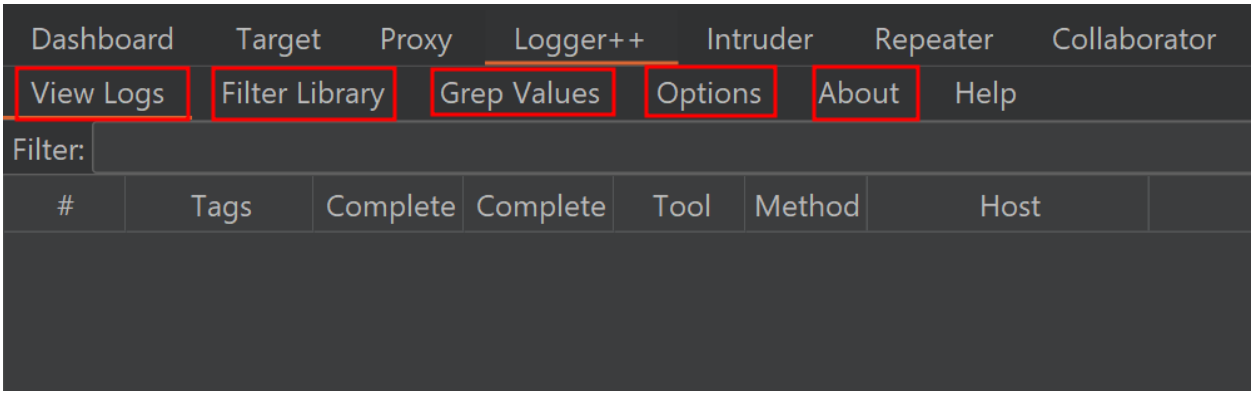


After successfully installation, it will appear on the tool bar.

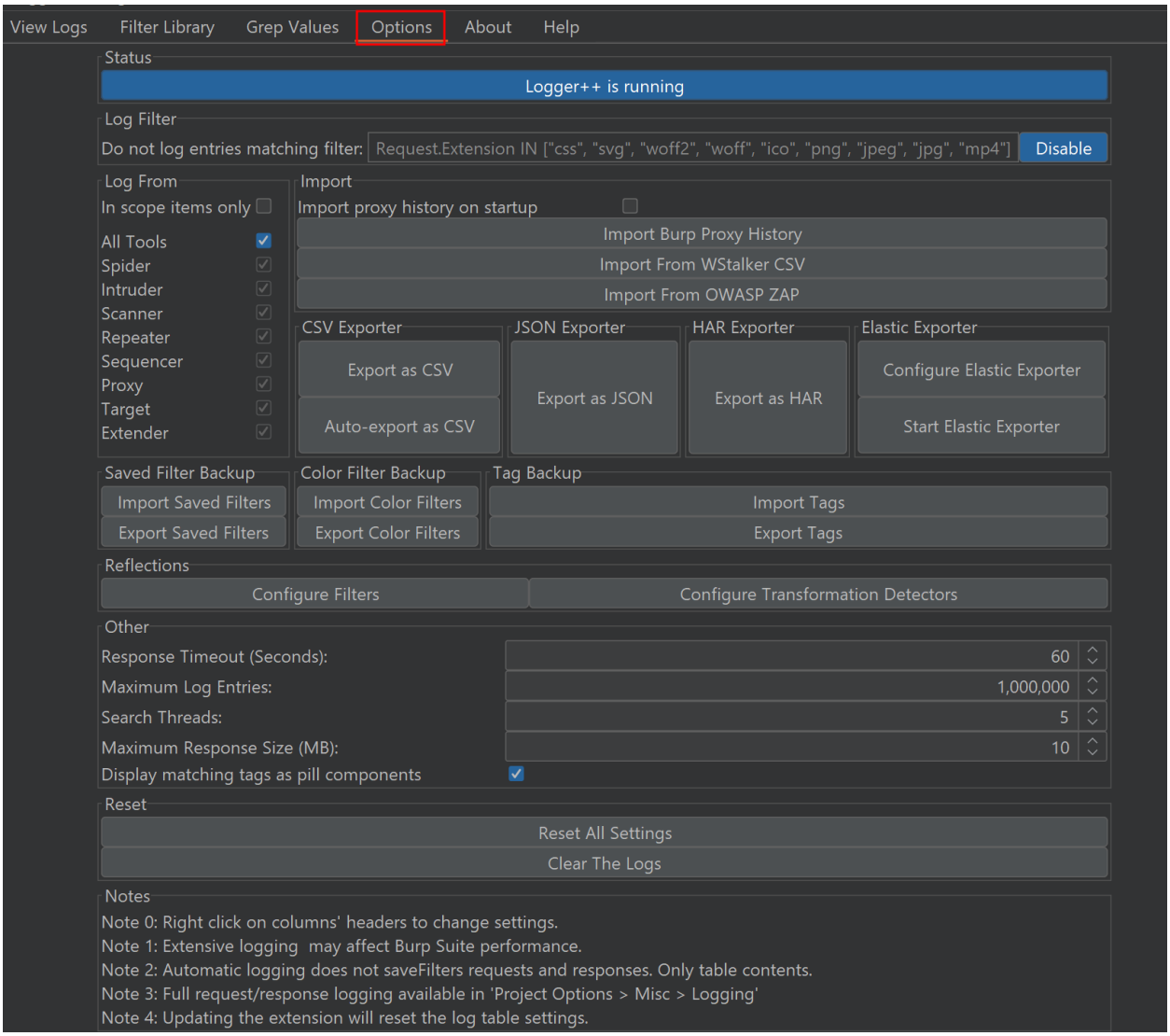


Navigating

There are a lot of options visible to you. First, let's explore the "Options" tab to discover what advanced settings are included in this extension?



Navigate to "Options" to see the various log filter options. It allows you to customize logging setting as per your preference.



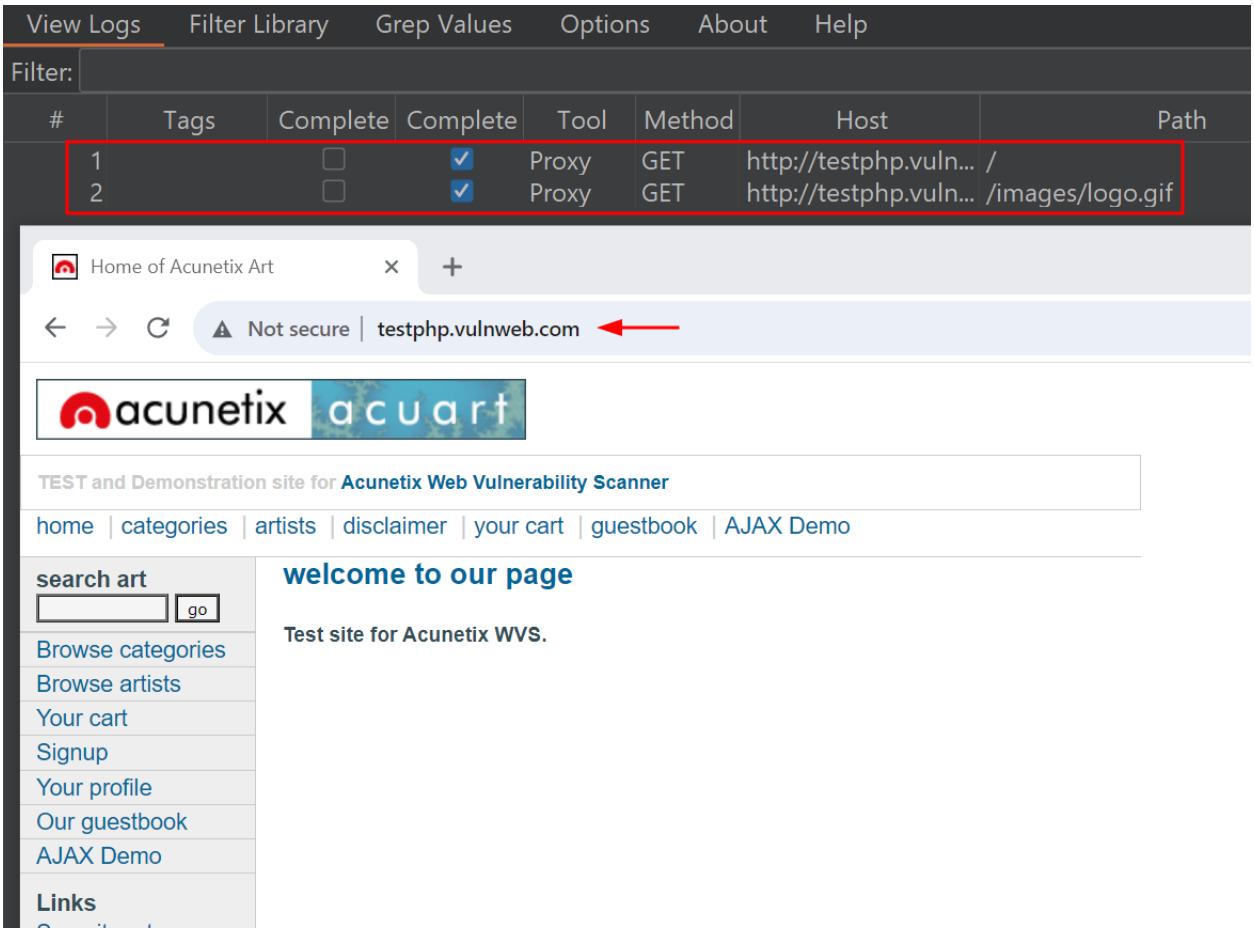
Logger++ is running by default. Here are some others important setting:

- Log Filter: This feature lets you specifically choose the requests that you don't need to record for analysis, or you may turn it off when not in use.
- Log From: It enables you to capture data from the specific logs that you want to capture from.
- Import: You can import log data from CSV and OWASP ZAP reports with this function.
- Export: The log data can be exported for further analysis.

Depending on your preferences, you can use different configuration. We are sticking with the default settings for the time being.

Query-Based Filter

The View Log tab contains all the logs. Using this website "vulnweb" as an example, browse it and simply scan the entire site; all logs will show up here under the View Logs page.

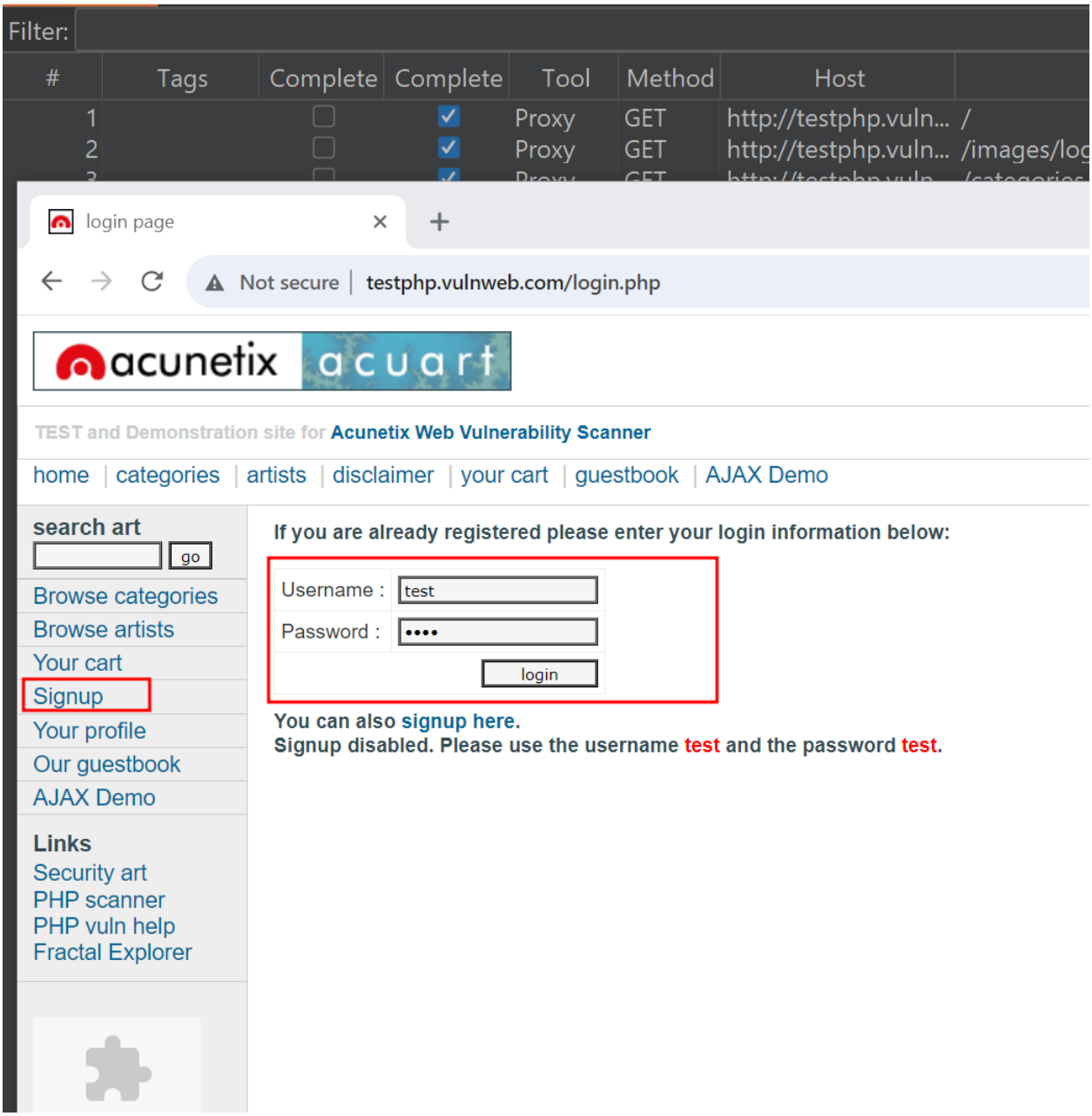


Now, go to Signup. In order to capture the logs for credentials, enter the test login details.

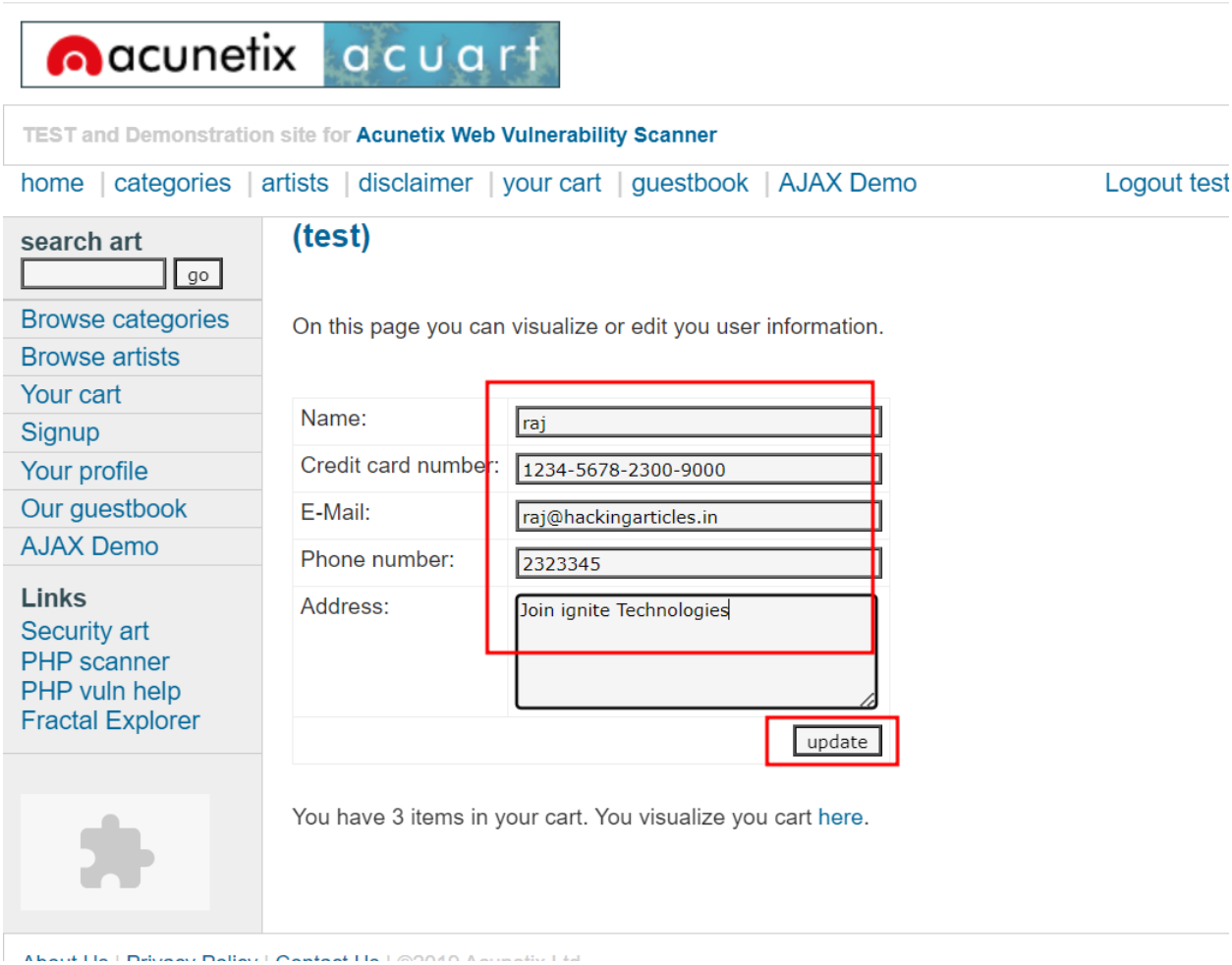
Username: test

Password: test

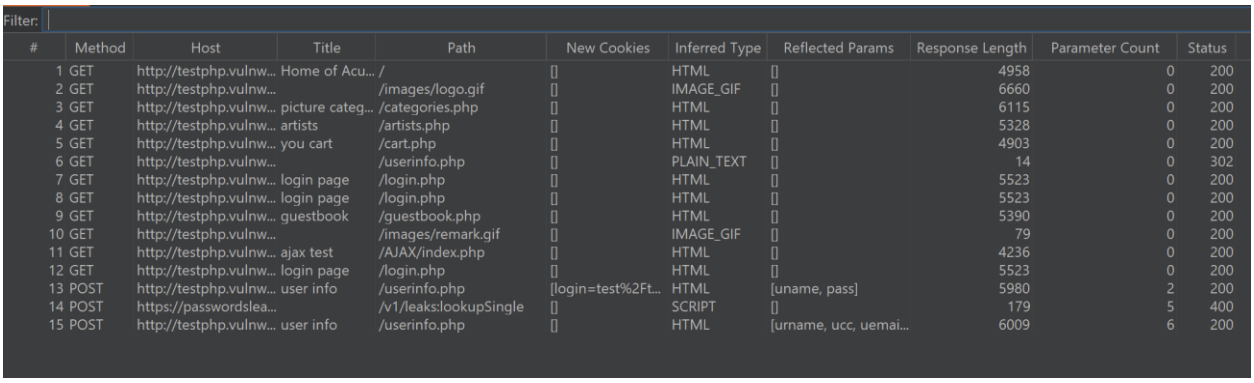
then click on "Login".



Let's update some more details to capture more requests for further analysis.



You can see that all requests have been captured here in View Logs.



Filter:

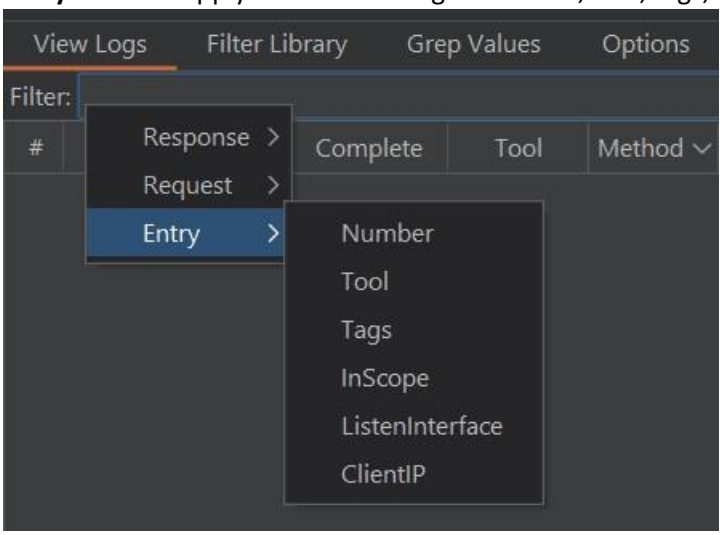
#	Method	Host	Title	Path	New Cookies	Inferred Type	Reflected Params	Response Length	Parameter Count	Status
1	GET	http://testphp.vulnw...	Home of Acu...	/	[]	HTML	[]	4958	0	200
2	GET	http://testphp.vulnw...		/images/logo.gif	[]	IMAGE_GIF	[]	6660	0	200
3	GET	http://testphp.vulnw...	picture categ...	/categories.php	[]	HTML	[]	6115	0	200
4	GET	http://testphp.vulnw...	artists	/artists.php	[]	HTML	[]	5328	0	200
5	GET	http://testphp.vulnw...	you cart	/cart.php	[]	HTML	[]	4903	0	200
6	GET	http://testphp.vulnw...		/userinfo.php	[]	PLAIN_TEXT	[]	14	0	302
7	GET	http://testphp.vulnw...	login page	/login.php	[]	HTML	[]	5523	0	200
8	GET	http://testphp.vulnw...	login page	/login.php	[]	HTML	[]	5523	0	200
9	GET	http://testphp.vulnw...	guestbook	/guestbook.php	[]	HTML	[]	5390	0	200
10	GET	http://testphp.vulnw...		/images/remark.gif	[]	IMAGE_GIF	[]	79	0	200
11	GET	http://testphp.vulnw...	ajax test	/AJAX/index.php	[]	HTML	[]	4236	0	200
12	GET	http://testphp.vulnw...	login page	/login.php	[]	HTML	[]	5523	0	200
13	POST	http://testphp.vulnw...	user info	/userinfo.php	[login=test%2Ft...	HTML	[uname, pass]	5980	2	200
14	POST	https://passwordslea...		/v1/leaks:lookupSingle	[]	SCRIPT	[]	179	5	400
15	POST	http://testphp.vulnw...	user info	/userinfo.php	[]	HTML	[urname, ucc, uemai...	6009	6	200

Magical Filter

You can use filter to selectively view or manipulate HTTP requests and responses. These filters help you focus on specific aspects of the web traffic and are especially useful during security testing. The working is based on query string. It accepts a logical query and returns output based on them.

You have some advanced choices with the filter options:

- **Entry**: You can apply filters according to number, tool, tags, InScope, and other criteria.



- **Request**: It lets you filter just the request itself using many options such as header, body, URL, method, parameters, cookies, etc. As shown below:

View Logs | Filter Library | Grep Values | Options | About | Help

Filter:

#	Method	Host	
1	GET	http://testphp.v	
2	GET	http://testphp.v	
3	GET	http://testphp.vulnw...	picture c
4	GET	http://testphp.vulnw...	artists
5	GET	http://testphp.vulnw...	you cart
6	GET	http://testphp.vulnw...	
7	GET	http://testphp.vulnw...	login pag
8	GET	http://testphp.vulnw...	login pag
9	GET	http://testphp.vulnw...	guestboo
10	GET	http://testphp.vulnw...	
11	GET	http://testphp.vulnw...	ajax test
12	GET	http://testphp.vulnw...	login pag
13	POST	http://testphp.vulnw...	user info
14	POST	https://passwordslea...	
15	POST	http://testphp.vulnw...	user info

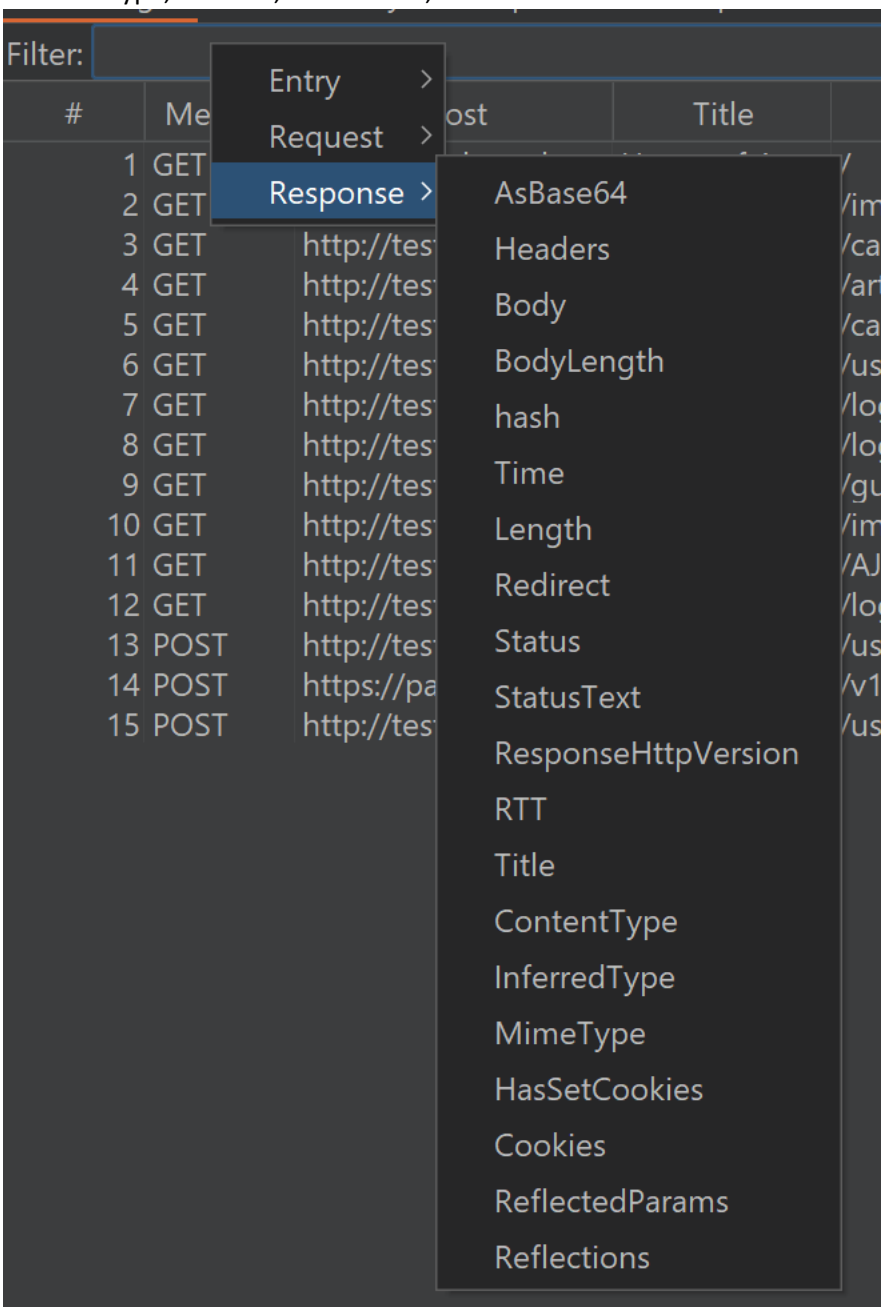
Entry
Request
Response

AsBase64
Headers
Body
BodyLength
Time
Length
Tool
Comment
Complete
URL
Method
Path
Query
PathQuery
Protocol
IsSSL
UsesCookieJar
Hostname
Host
Port
ContentType
RequestHttpVersion
Extension
Referrer
HasParams
HasGetParam
HasPostParam
HasSentCookies
CookieString
ParameterCount
Parameters
Origin

Pretty | Raw | Hex

1

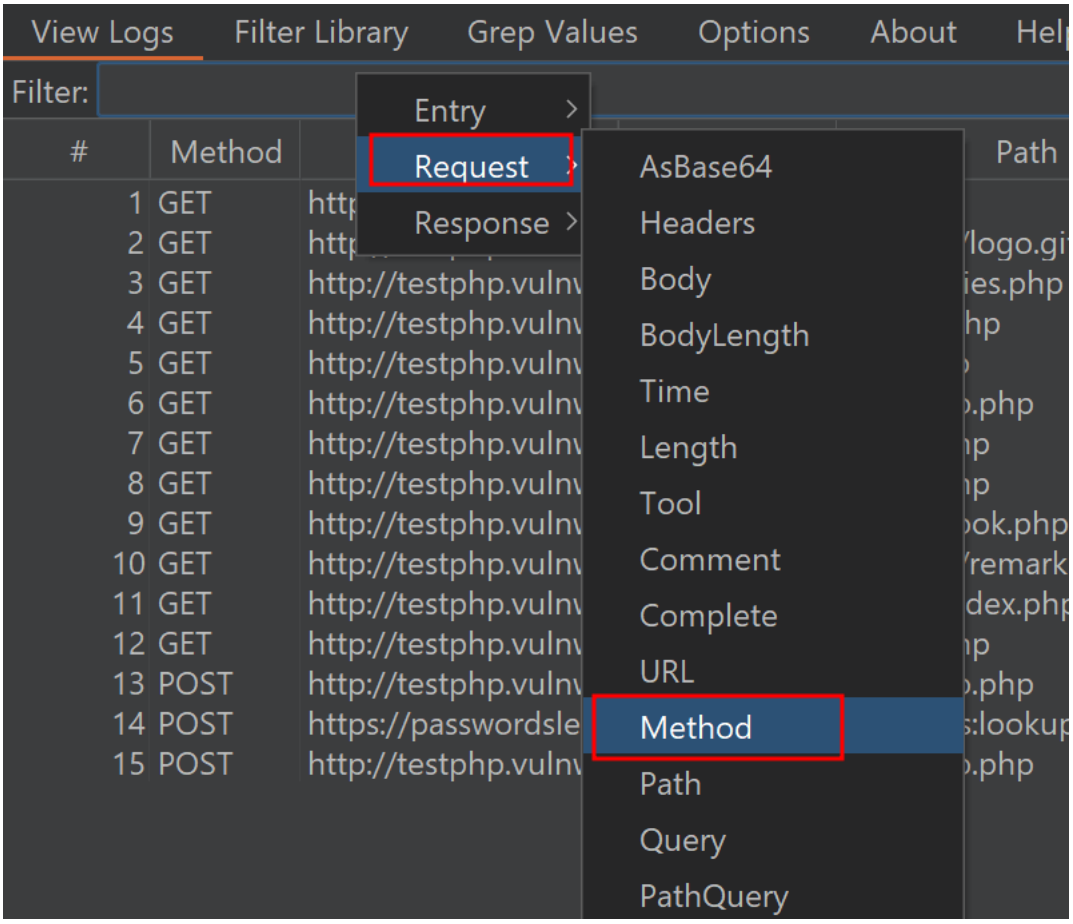
- **Response**: It lets you filter just the response by using various options such as header, body, Inferred Type, Method, Parameters, cookies etc. As shown below:



<u>Scenario 1:</u> Let's suppose you just want to view HTTP POST requests from all logs. It is understood that HTTP POST parameters are in HTTP Request.

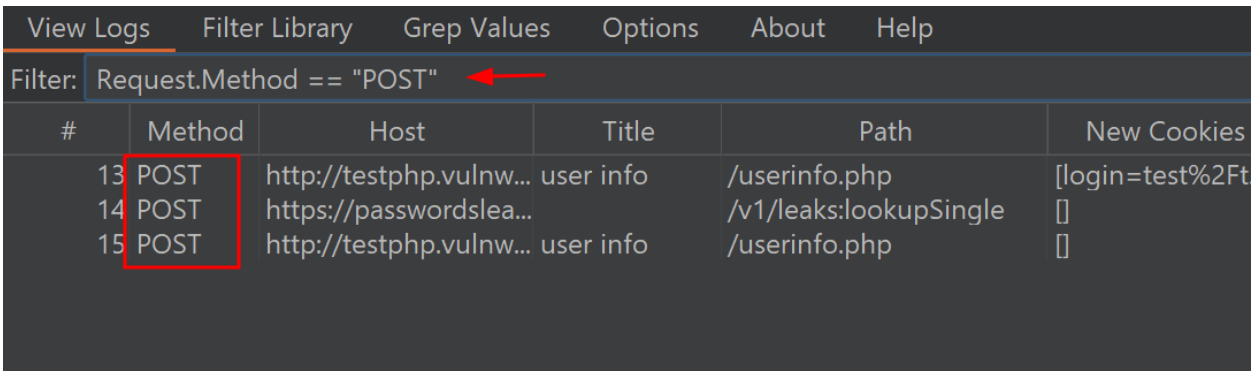
Go to Filter bar > right click > Select Request > Select Method

The method has been chosen and visible in filter bar.



Quary: Request.Method == “POST”

And hit enter. As result, Only HTTP POST Method requests appear.



<u>Scenario 2:</u> Taking another example, suppose we just want to view the requests which contains any username information from all logs.

Go to Filter bar > right click > Select Request > Select Body

Quary: Request.Body CONTAINS “uname”

As a result, the following request is highlighted:

View Logs | Filter Library | Grep Values | Options | About | Help

Filter: Request.Body CONTAINS "uname"

#	Method	Host	Title	Path	New Cool
13	POST	http://testphp.vulnw...	user info	/userinfo.php	[login=test%

Pretty | Raw | Hex

```
1  POST /userinfo.php HTTP/1.1
2  Host: testphp.vulnweb.com
3  Content-Length: 20
4  Cache-Control: max-age=0
5  Upgrade-Insecure-Requests: 1
6  Origin: http://testphp.vulnweb.com
7  Content-Type: application/x-www-form-urlencoded
8  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, lik
9  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,
10 Referer: http://testphp.vulnweb.com/login.php
11 Accept-Encoding: gzip, deflate, br
12 Accept-Language: en-US,en;q=0.9
13 Connection: close
14
15 uname=test&pass=test
```

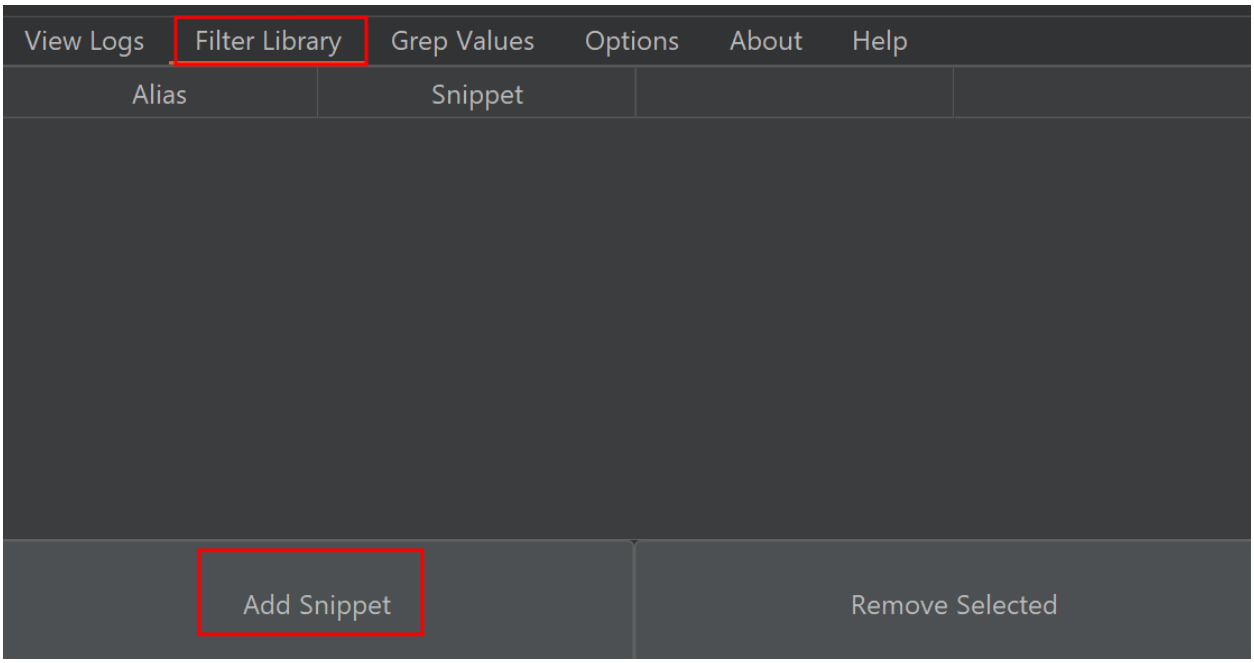
Below are some useful queries which are helpful during penetration testing.

Some Useful Filter Queries

JSON Injection (Check for only one json request)	Response.InferredType == "json"
Injections Attack (Check for HTML, XML, JSON)	Response.InferredType IN ["json", "html", "xml"]
Disclosed Server Information	Response.header CONTAINS “Server:”
Exposed Sensitive File	Response.Body CONTAINS [“.git”, “.config”, “.zip”, “.swf”, “.doc”, “.pdf”, “.xlsx”, “.csv”,]
Exposed Sensitive Path	Request.Path CONTAINS [“/git”, “/etc”, “/var”] Request.Path MATCHES “/account*”
Sensitive Parameter in Query String	Request.Path CONTAINS [“id”, “username”, “password”, “role”, “IsAdmin”]
Sensitive Parameter in Request	Request.Body CONTAINS [“id”, “username”, “password”, “token”, “role”, “EnterpriseID”, “IsAdmin”]
Missing Robots.txt	Request.Path MATCHES “/robots.txt”
CORS Misconfiguration	Response.Header MATCHES “ Access-Control-Allow-Origin: *”
Check for CSRF Token	Request.Method == "POST" AND Request.Body CONTAINS "csrf"
URL Redirection	Request.Path CONTAINS [“redirect=”, “page=”, “url=”, “index.page=”]

Filter Library

We can use the saved or pre-configured filters from the library directly with the help of the Filter Library. When you start testing, you do not have to manually type or remember the query string of filter pattern.

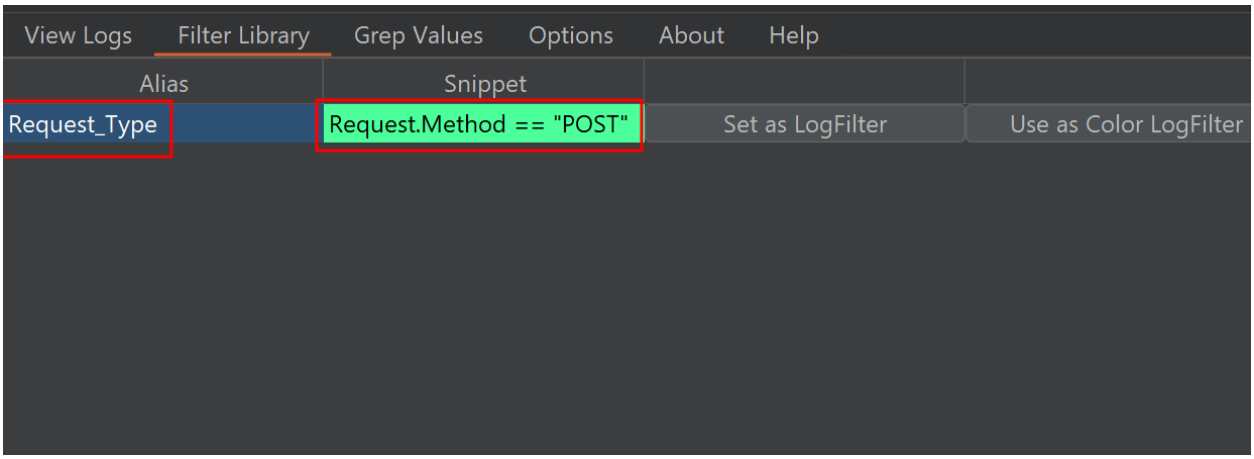


Click on “Add Snippet”. Here are two values that must be added.

- **Alias**: Put any Alias name for your query string.
- **Snippet**: Add query string here.

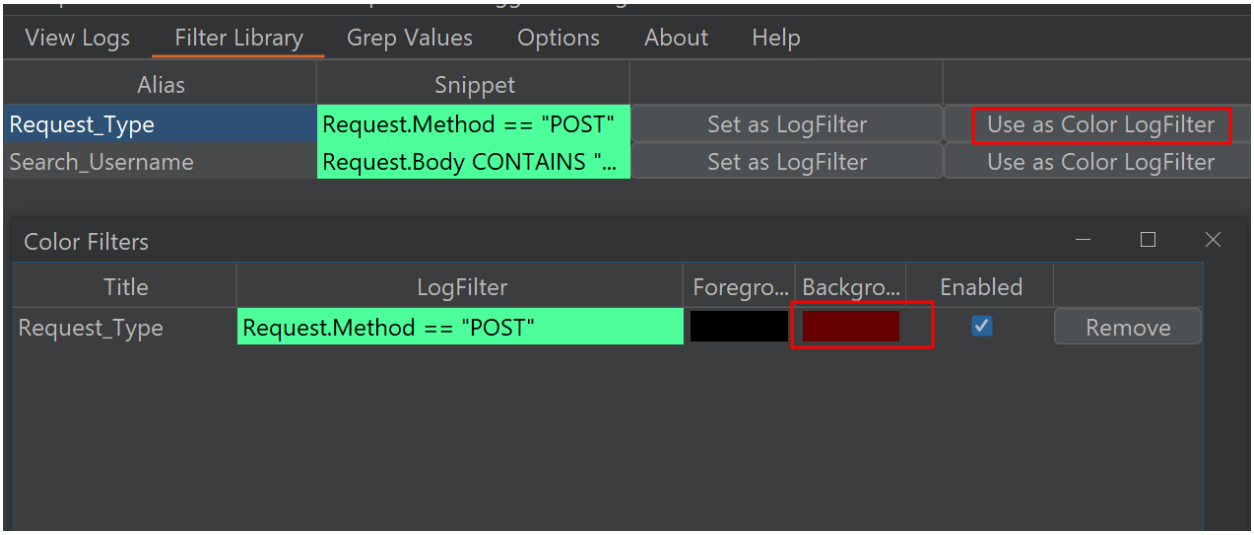
As you can see below, I have added a filter for

Request_Type: Request.Method == “POST”

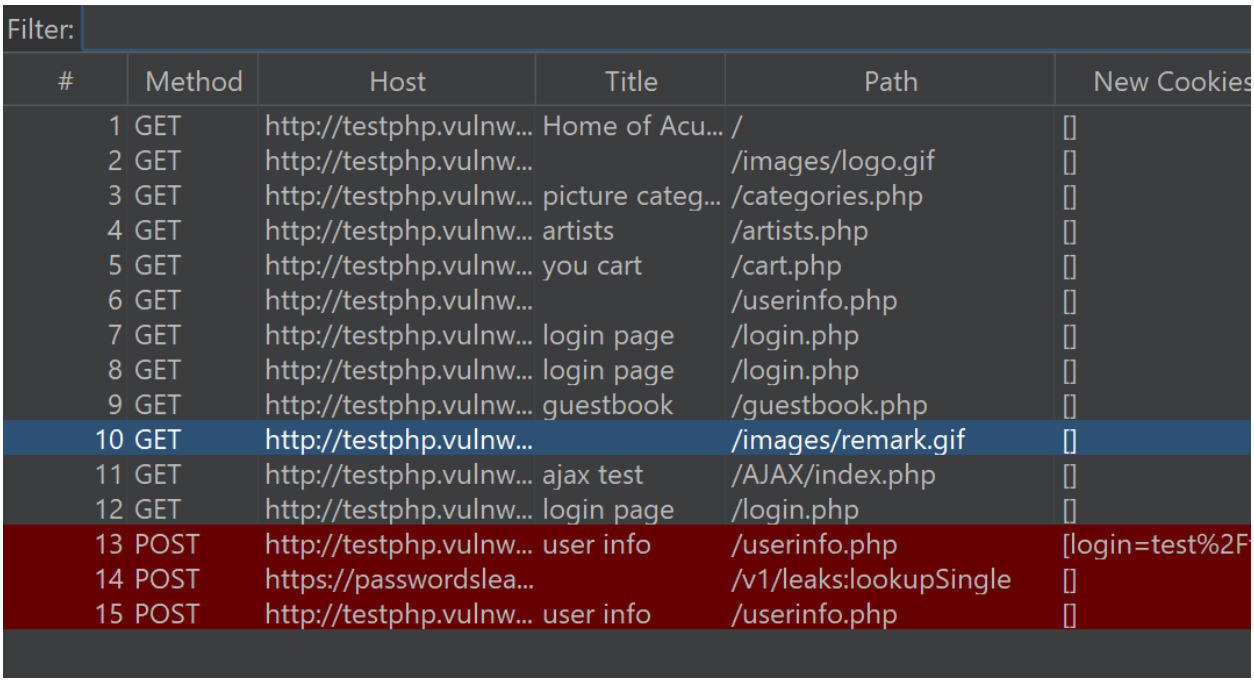


You no longer have to type repeatedly to find only POST requests. You can color-code this request so that the highlighted request stands out among all captured requests on the View Logs page.

Click on Use as Color LogFilter > Select Background Color > check Enable and save it.

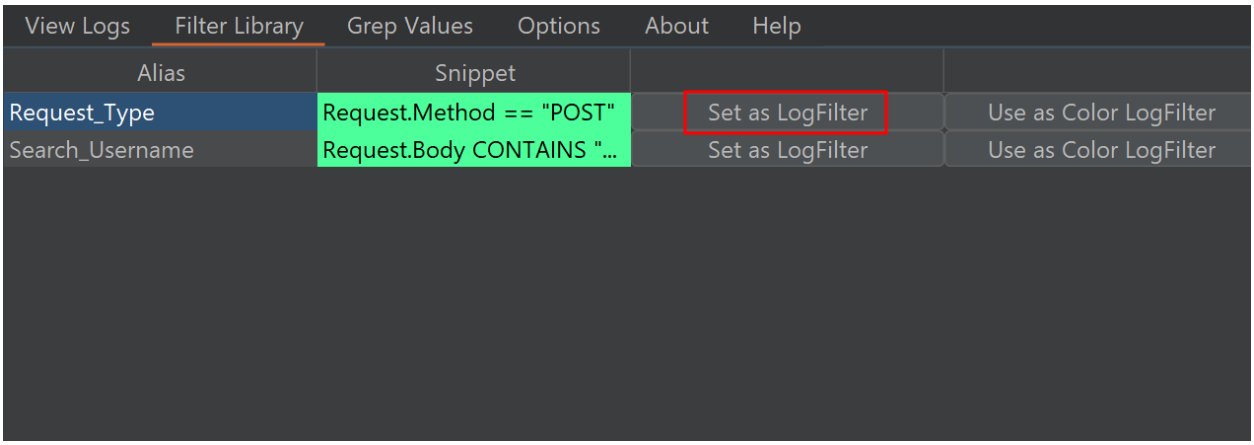


All POST requests are now highlighted in "Dard-Red" on this page.

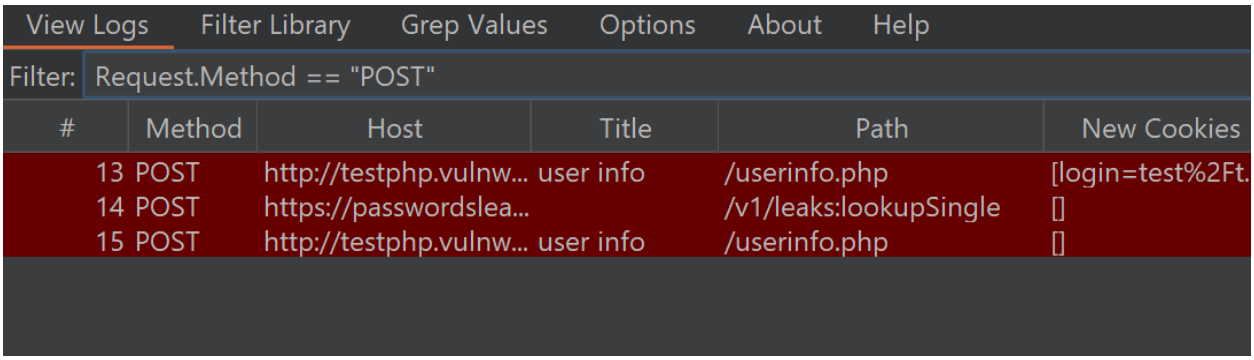


Similarly, you can save whole test scenarios in the Filter Library. There is two ways to call the saved filter:

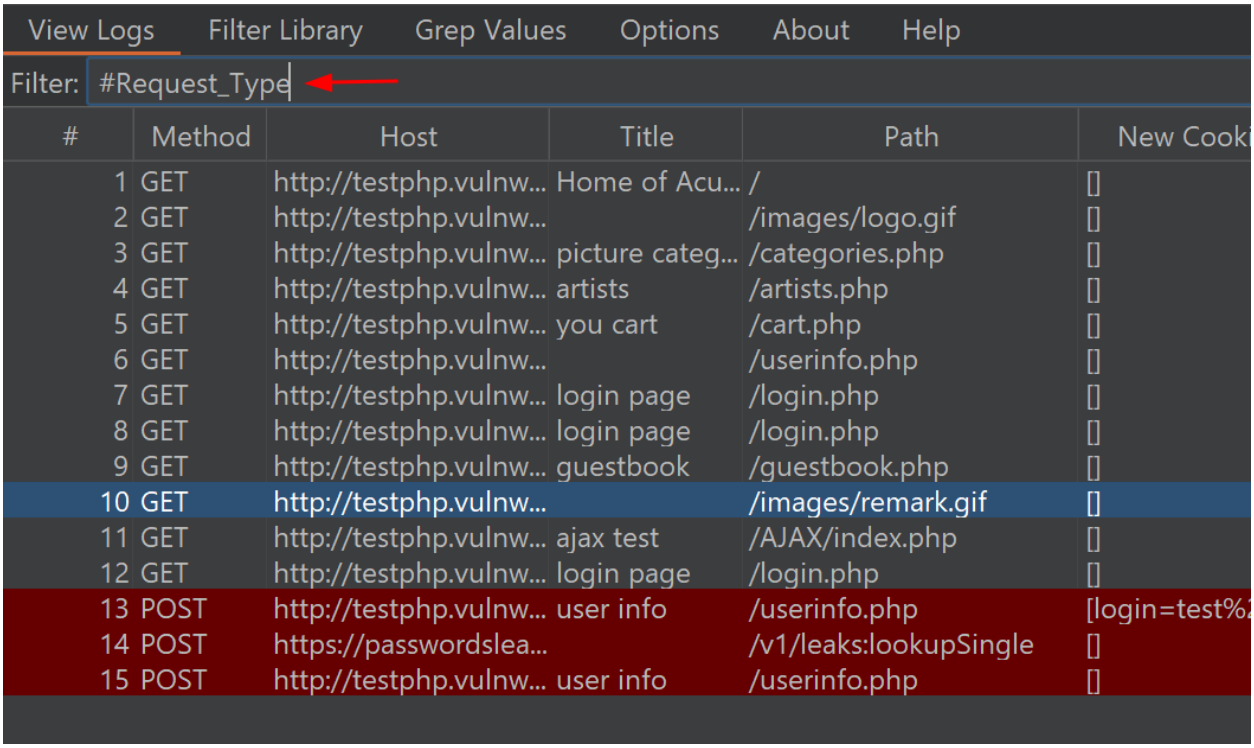
<u>Method 1</u>: In Filter Library, click on Set as LogFilter.



It will directly run the query and the desire result will be displayed.



<u>Method 2:</u> Use "#" with Alias name directly in filter bar.



And hit enter. The equivalent outcome will appear as follows:

View Logs | Filter Library | Grep Values | Options | About | Help

Filter: #Request_Type

#	Method	Host	Title	Path	New Cookies
13	POST	http://testphp.vulnw...	user info	/userinfo.php	[login=test%2Ft...
14	POST	https://passwordslea...		/v1/leaks:lookupSingle	[]
15	POST	http://testphp.vulnw...	user info	/userinfo.php	[]

Regex-Based Filter

Burp Logger's regex filter is a powerful feature that helps web security professionals pinpoint specific data within the vast sea of information during security testing.

You need to specify the regular expression (regex) pattern. This pattern acts like a search query, telling Burp Logger++ what kind of data you want to capture. You can create regex expression pattern to find data as like Email Address, IP Address, Server-side error messages, Software version disclosed, Any API Key exposed etc.

Go to Logger++, click on Grep Values tab. Here, you can see more filters to limit the search criteria.

- Search Response = It will perform search only in responses.
- Search Request = It will perform search only in requests.
- In Scope Only = If you added the target URL in Scope only then it will only search within the scoped target.

For the time being, choose to search through every request and response. Let’s take an example, if you want to find email addresses in web traffic, your regex pattern might look like

Regex: [\w\.-]+@[\w\.-]+.

Directly paste this expression under Regex bar and press enter.

iGNITE Technologies

View Logs | Filter Library | Grep Values | Options | About | Help

Regex: [\w\.-]+@[\w\.-]+.

Results | Unique Results

Entry	Total Matches
http://testphp.vulnweb.com/images/logo.gif	1
http://testphp.vulnweb.com/userinfo.php	1
http://testphp.vulnweb.com/userinfo.php	1

Pretty | Raw | Hex

```
1  POST /userinfo.php HTTP/1.1
2  Host: testphp.vulnweb.com
3  Content-Length: 129
4  Cache-Control: max-age=0
5  Upgrade-Insecure-Requests: 1
6  Origin: http://testphp.vulnweb.com
7  Content-Type:
   application/x-www-form-urlencoded
8  User-Agent: Mozilla/5.0 (Windows NT 10.0;
   Win64; x64) AppleWebKit/537.36 (KHTML,
   like Gecko) Chrome/116.0.5845.111
   Safari/537.36
9  Accept:
   text/html,application/xhtml+xml,applicatio
   n/xml;q=0.9,image/avif,image/webp,image/ap
   ng,*/*;q=0.8,application/signed-exchange;v
   =b3;q=0.7
10 Referer:
   http://testphp.vulnweb.com/userinfo.php
11 Accept-Encoding: gzip, deflate, br
12 Accept-Language: en-US,en;q=0.9
13 Cookie: login=test%2Ftest
14 Connection: close
15
16 urname=raj&ucc=1234-5678-2300-9000&uemail=
   raj%40hackingarticles.in&uphone=2323345&
   uaddress=Join+ignite+Technologies&update=
   update
```

Pretty | Raw | Hex | Render

```
65          <tr>
              <td valign="top">
                Name:
              </td>
              <td>
                <input type="text" value="raj" name="urname" style="
              </td>
            </tr>
66          <tr>
              <td valign="top">
                Credit card number:
              </td>
              <td>
                <input type="text" value="1234-5678-2300-9000" name=
              </td>
            </tr>
67          <tr>
              <td valign="top">
                E-Mail:
              </td>
              <td>
                <input type="text" value="raj@hackingarticles.in" na
              </td>
            </tr>
68          <tr>
              <td valign="top">
                Phone number:
              </td>
              <td>
                <input type="text" value="2323345" name="uphone" sty
              </td>
            </tr>
69          <tr>
```

Consequently, the /userinfo.php request — which includes the email mentioned above is displayed.

You have two ways: Manually search through the complete request/response or click on Unique Result. The results that match the regex expression will be displayed only in Unique Results.

View Logs | Filter Library | Grep Values | Options | About | Help

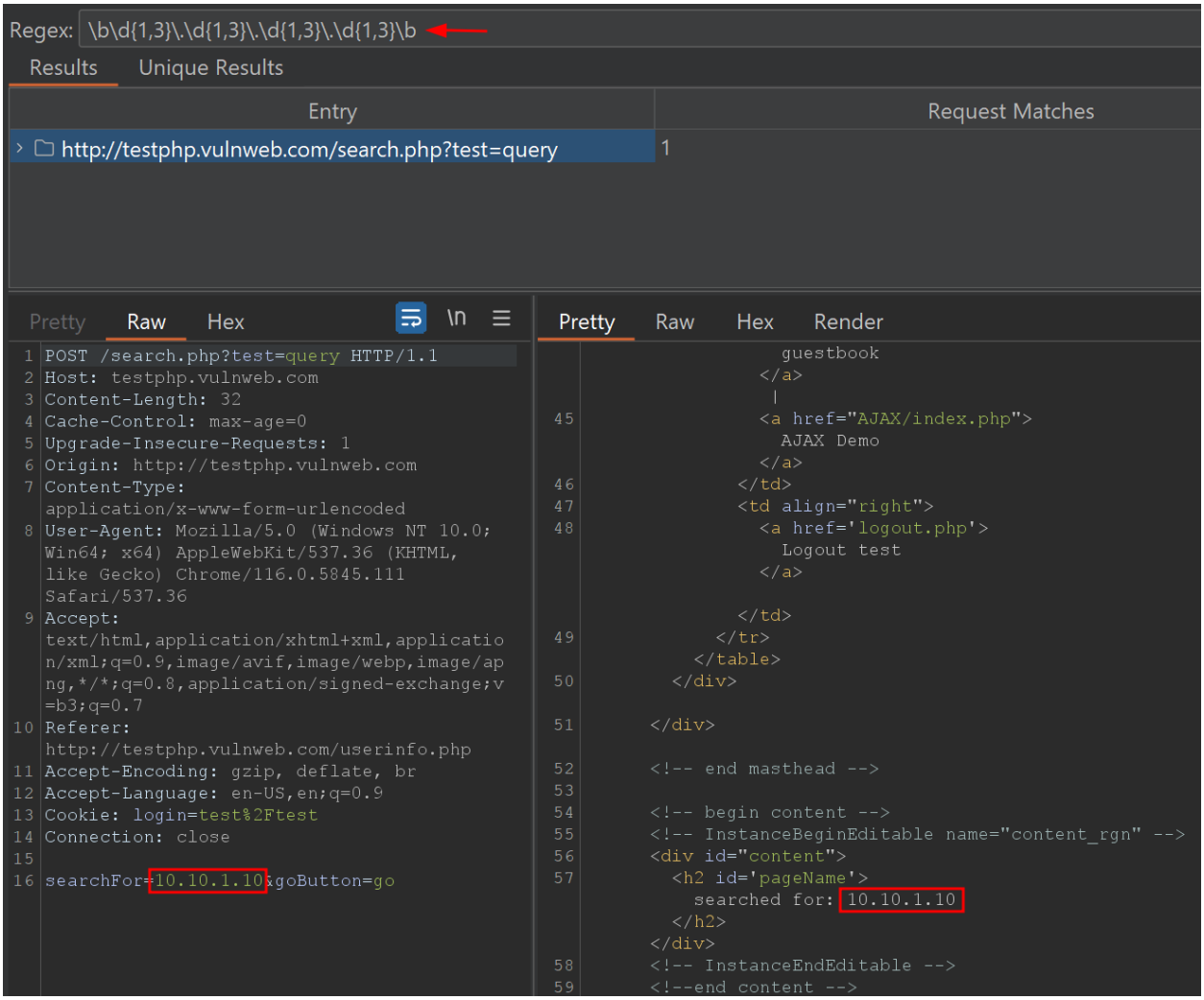
Regex: [\w\.-]+@[\w\.-]+.

Results | Unique Results

Value
A@r□
email@email.com"
raj@hackingarticles.in"

Similarly, Let's check for IP Address also,

Regex Exp: \b\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\b



It is evident that a POST request is being sent through the IP address 10.10.1.10.

In the same way, you can check for other important information like if you want to find the web traffic contains any FTP, HTTP, WWW.

Regex: \b(ftp|www|http)[^\s]+

For reference, the following link includes helpful regex expression to find the disclosed server version within the error information.

https://github.com/lwierzbicki/RegexFinder/blob/main/burp.regex.tsv

Export Data Feature

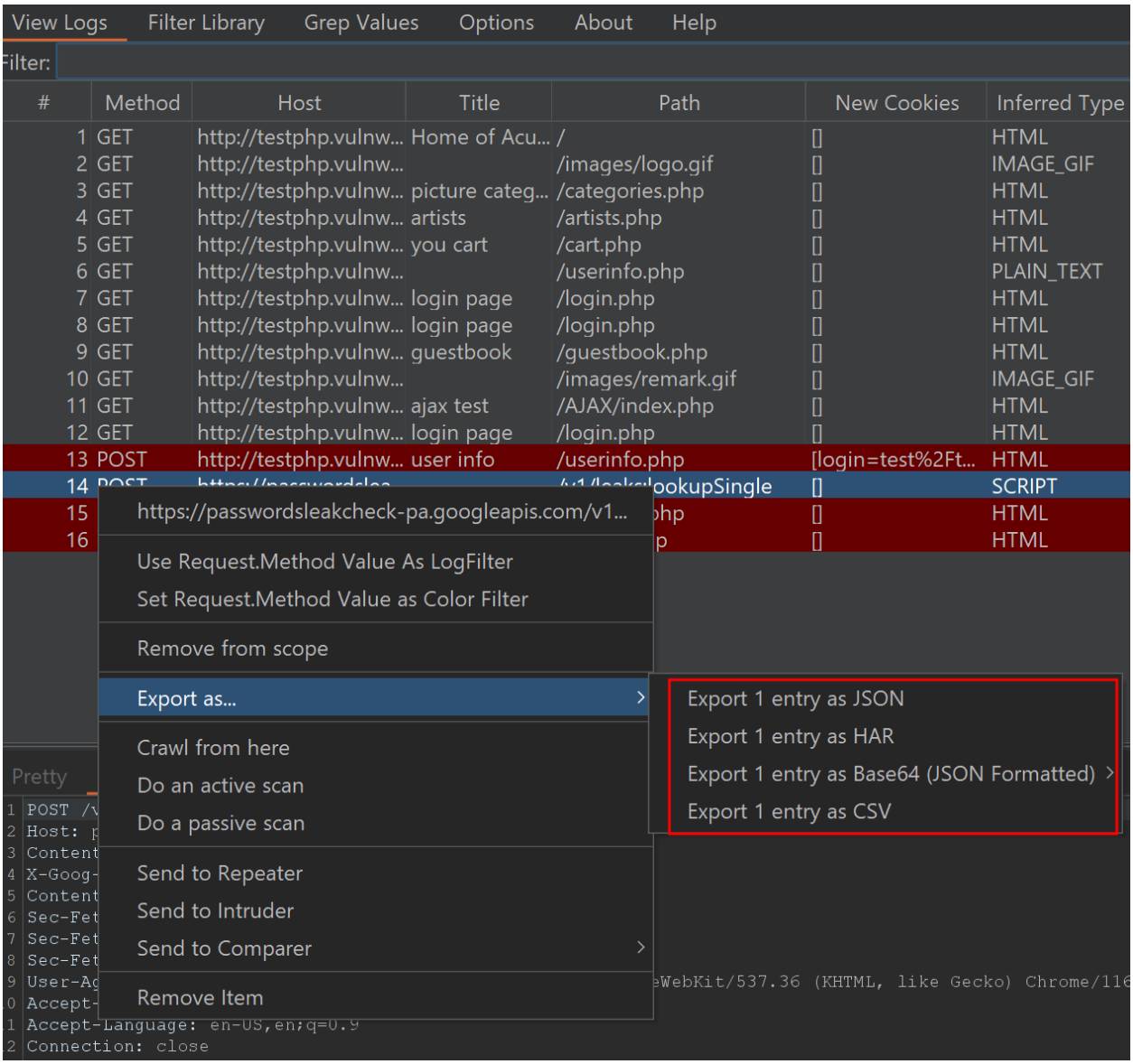
Burp Logger's data export feature is a valuable tool for web security professionals. It allows you to save, analyze, and share the captured data efficiently, making it an essential tool for documenting findings, performing in-depth analysis, and collaborating with others in the field of web security.

Why Export Data Feature is Helpful?

- **Data Preservation**: Exporting data from the Logger++ allows you to save a record of your testing session. This is essential for documentation and analysis.
- **External Analysis**: By exporting data, you can use external tools or software to perform in-depth analysis, generate reports, or share findings with team members.
- **Archiving Evidence**: It helps in preserving evidence of potential vulnerabilities or security issues discovered during testing, which is crucial for audits and compliance.
- **Collaboration**: Exported data can be easily shared with colleagues or experts for collaborative analysis, making it an asset in team-based security testing.
- **Customization**: Depending on the export format chosen, you can tailor the exported data to meet specific reporting or analysis requirements.

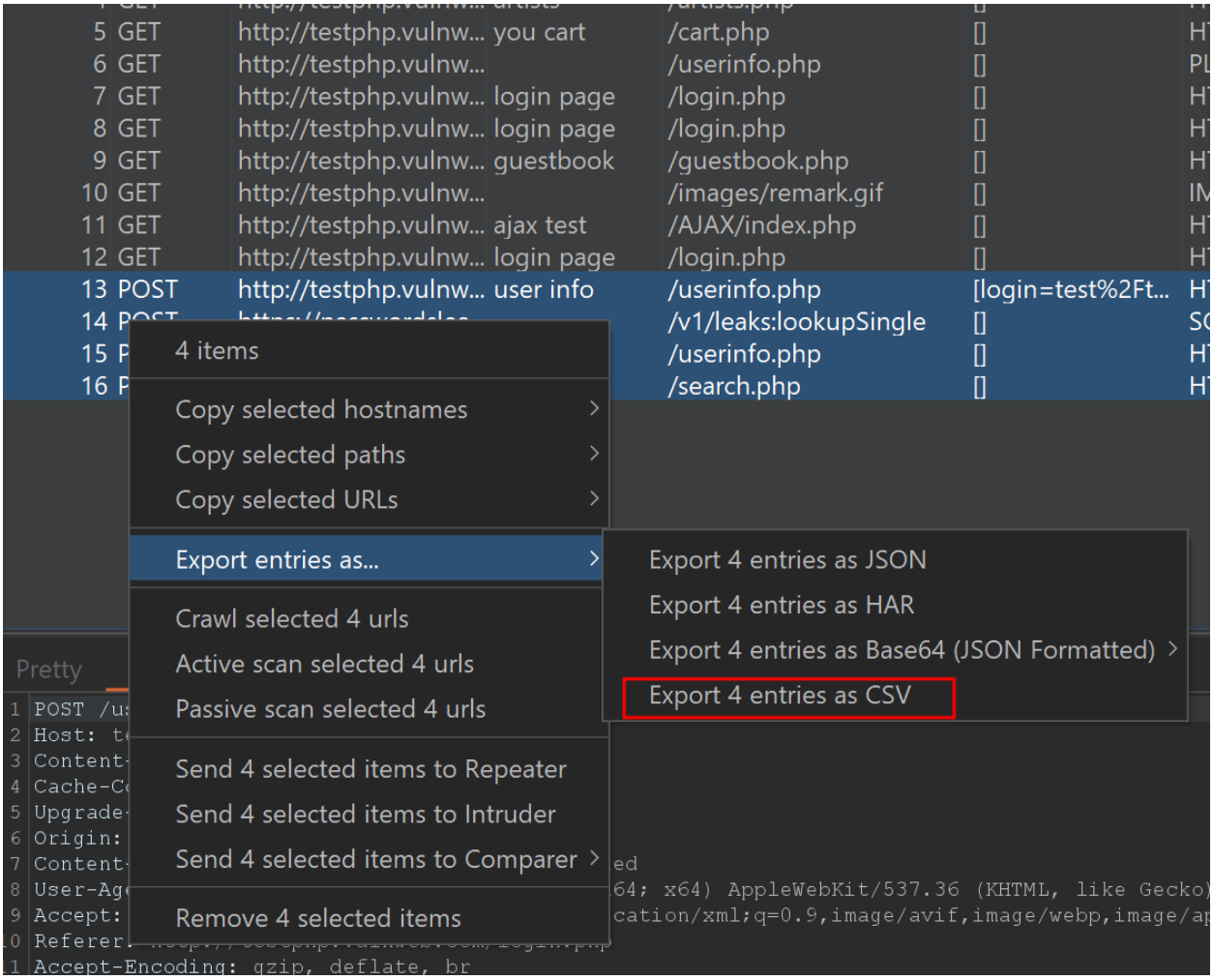
Supported Formats:

- **Base64 JSON Format**: Base64-encoded data is often used to include binary data within a JSON structure.
- **JSON Format**: JSON is a lightweight data-interchange format used for structured data.
- **CSV Format**: CSV files are widely supported and can be opened in spreadsheet software like Microsoft Excel or Google Sheets.
- **HAR Format**: HTTP Archive (HAR) format is used for capturing and storing the performance-related data. The HAR format contains detailed information about HTTP requests and responses.

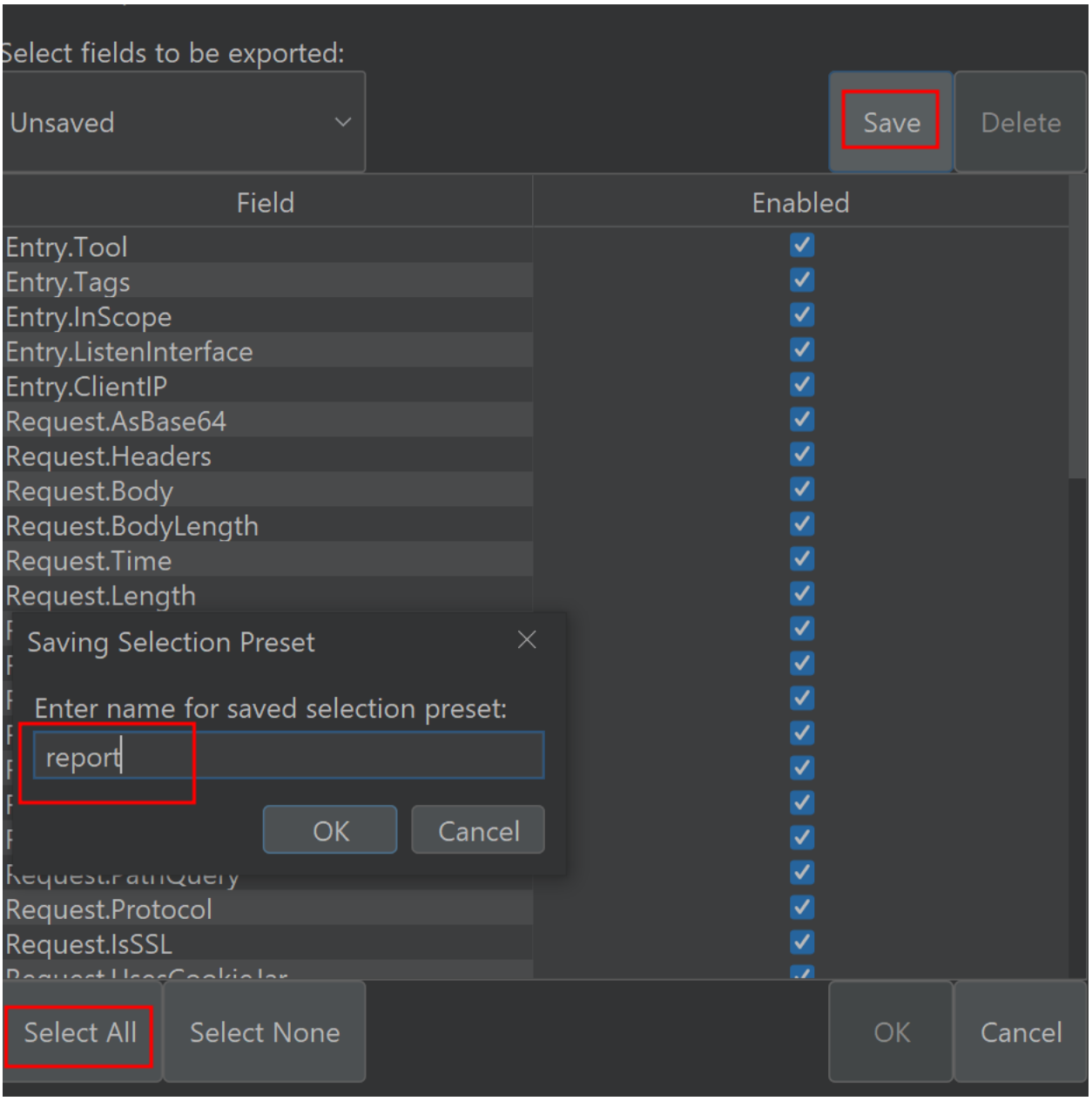


For Example, suppose you want to export all POST requests for further analysis.

Select the associate requests > right click > choose Export entries as > Export as CSV

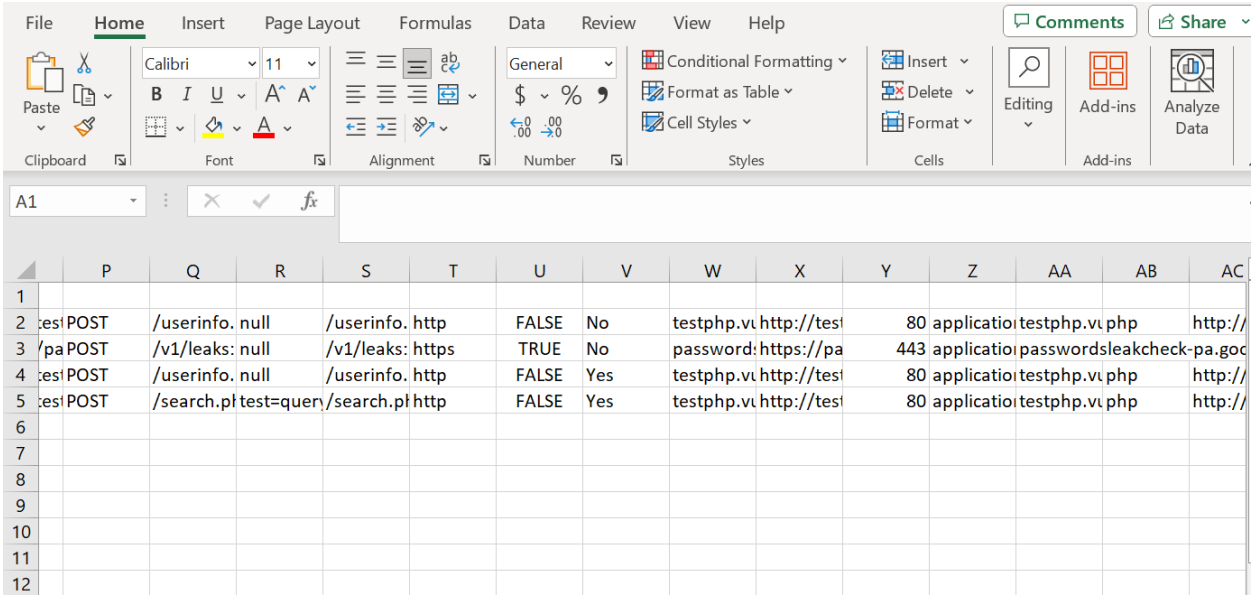


Now Select All > Choose Save > Enter the name and click on Ok.



Save the result to your system offline. You can examine the CSV file; it contains all of the values that you chose to save.

You may select the only required values to store based on your needs.



	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
1														
2	tes	POST	/userinfo.	null	/userinfo.	http	FALSE	No	testphp.vu	http://test	80	applicatio	testphp.vu	php	http://
3	/pa	POST	/v1/leaks:	null	/v1/leaks:	https	TRUE	No	password	https://pa	443	applicatio	passwordsleakcheck-pa.goo	
4	tes	POST	/userinfo.	null	/userinfo.	http	FALSE	Yes	testphp.vu	http://test	80	applicatio	testphp.vu	php	http://
5	tes	POST	/search.ph	test=quer	/search.ph	http	FALSE	Yes	testphp.vu	http://test	80	applicatio	testphp.vu	php	http://
6														
7														
8														
9														
10														
11														
12														

That’s All, Cheers!!!

Conclusion

People like Burp Logger++ because it makes their job easier. It is your smart friend who helps you find hidden treasures (or problems) on websites. With its filters and colorful highlights, has the superpowers for website exploration.

So, if you are a web detective, it is a tool that will make your job much more fun and efficient. Give it a try and see how it can help you on your web adventures!

iGNITE Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials

ADVANCED

- Burp Suite Pro
- Web Services-API
- Pro Infrastructure VAPT
- Computer Forensics
- Android Pentest
- Advanced Metasploit
- CTF

EXPERT

- Red Team Operation
  - APT's - MITRE Attack Tactics
  - Active Directory Attack
  - MSSQL Security Assessment
- Privilege Escalation
  - Windows
  - Linux