

## Burp Suite: Brute Forcing Credentials

### Overview

The Burp Suite software can brute force credentials of services operating on a web server. This is an invasive technique as this involves using the “Repeater” to send GET requests to the web server over-and-over so the user can analyze the response and results of each request.

### Background

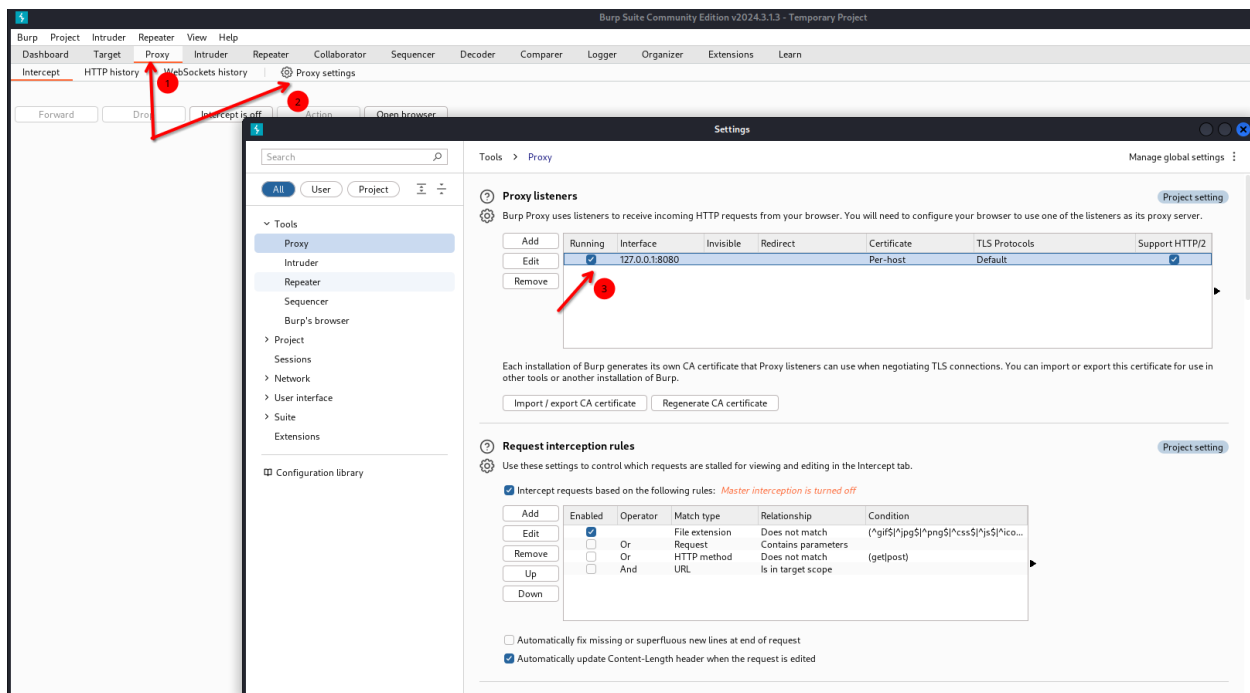
Brute forcing is a common tactic used by penetration testers who are attempting to discover user credentials to gain access to a particular system or service. This tactic usually involves a program that enumerates through a list of users and passwords. However, a true brute forcing attack would involve trying every possible combination of characters.

### Methodology

#### 1. Preparation

Before brute forcing with Burp Suite can occur, the web browser that we are using to visit the target website needs to be configured to use Burp Suite as a Proxy. This will allow Burp Suite to pick up all requests and responses from the requests the browser makes, as Burp Suite will make the requests on the browser’s behalf.

First, a Burp Suite project must be created for the session. Using a temporary project provides enough functionality to perform the brute forcing attack. After the project has finished creating, navigate to the “Proxy” tab, click on the “Proxy Settings” button, and confirm the default 127.0.0.1:8080 entry is present. If it is not, configure an entry to mimic the screenshot below



Now, the browser needs to be configured to utilize Burp as a Proxy. This technique will be demonstrated using Firefox. Navigate to the browser settings and find the proxy settings. Most modern browsers have a search feature in the settings section that can be used to find these settings quicker. When the proxy settings are found, use manual proxy settings to use the default Burp Suite proxy.

Connection Settings

×

Configure Proxy Access to the Internet

☐ No proxy

☐ Auto-detect proxy settings for this network

☐ Use system proxy settings

☒ Manual proxy configuration

HTTP Proxy

127.0.0.1

Port

8080

☐ Also use this proxy for HTTPS

HTTPS Proxy

Port

0

SOCKS Host

Port

0

☐ SOCKS v4

☒ SOCKS v5

☐ Automatic proxy configuration URL

Reload

No proxy for

Example: .mozilla.org, .net.nz, 192.168.1.0/24

Connections to localhost, 127.0.0.1/8, and ::1 are never proxied.

☐ Do not prompt for authentication if password is saved

☐ Proxy DNS when using SOCKS v5

Cancel

OK

Burp Suite and the browser are now officially prepared for a brute forcing attack.

## 2. Execution Steps

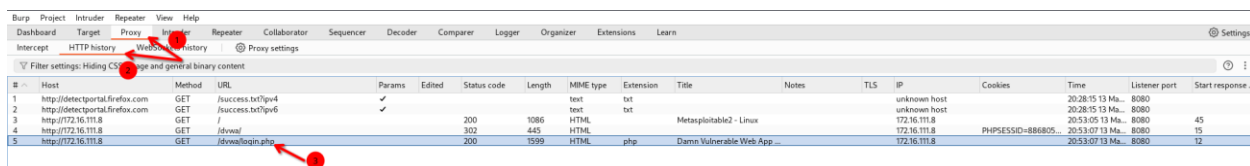
### Step 1: Logging a Vulnerable Request

Burp Suite is used for testing web applications which means the medium of exploitation will be the requests made to the web application. It is common to find web applications that interact with a user database used for storing information about accounts that their

users create. These are accounts that users can login to, and the login page is going to be exploited in this demonstration.

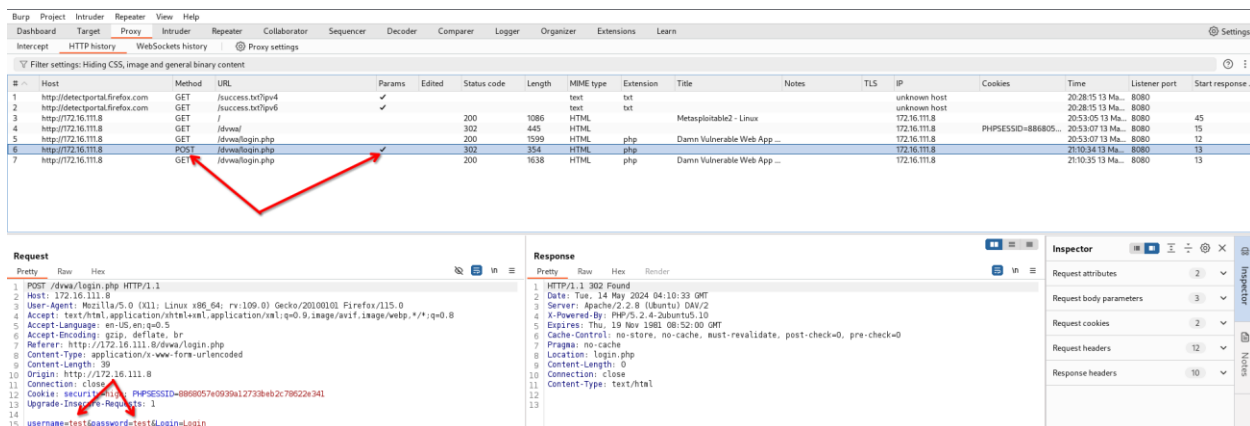
The Metasploitable 2 VM, found on Vulnhub, hosts the Damn Vulnerable Web App (DVWA) which can be used to test various web application exploitation techniques. Before the full app can be accessed, a login page is presented <http://172.16.111.8/dvwa/login.php>. This login page displays the user credentials but, for the purpose of this demonstration, they will be ignored.

Every web request made by the browser is logged by Burp Suite. The GET request that the browser made when retrieving the login page can be viewed by going to the “Proxy” tab and clicking the “HTTP history” tab just below it.



#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port	Start response...
1	http://detectportal.firefox.com	GET	/success.txt?pv=4					text	txt				unknown host		20:28:15 13 Ma...	8080	
2	http://detectportal.firefox.com	GET	/success.txt?pv=6					text	txt				unknown host		20:28:15 13 Ma...	8080	
3	http://172.16.111.8	GET	/			200	1086	HTML		Metasploitable2 - Linux			172.16.111.8		20:53:05 13 Ma...	8080	45
4	http://172.16.111.8	GET	/dvwa/			302	445	HTML					172.16.111.8	PHPSESSID=886805...	20:53:07 13 Ma...	8080	15
5	http://172.16.111.8	GET	/dvwa/login.php			200	1599	HTML	php	Damn Vulnerable Web App ...			172.16.111.8		20:53:07 13 Ma...	8080	12

GET requests are requests that are made from the client to the server, when the client wants to retrieve content from the web server. POST requests are made from the client to the server, but this time the client is providing information to the web server, commonly used when providing login information to the web server. Pick any pair of login credentials and send a test login request to the web server so that Burp Suite can log it. For this demonstration, “test” will be used for the username and password.



#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port	Start response...
1	http://detectportal.firefox.com	GET	/success.txt?pv=4					text	txt				unknown host		20:28:15 13 Ma...	8080	
2	http://detectportal.firefox.com	GET	/success.txt?pv=6					text	txt				unknown host		20:28:15 13 Ma...	8080	
3	http://172.16.111.8	GET	/			200	1086	HTML		Metasploitable2 - Linux			172.16.111.8		20:53:05 13 Ma...	8080	45
4	http://172.16.111.8	GET	/dvwa/			302	445	HTML					172.16.111.8	PHPSESSID=886805...	20:53:07 13 Ma...	8080	15
5	http://172.16.111.8	GET	/dvwa/login.php			200	1599	HTML	php	Damn Vulnerable Web App ...			172.16.111.8		20:53:07 13 Ma...	8080	12
6	http://172.16.111.8	POST	/dvwa/login.php			302	394	HTML	php				172.16.111.8		21:10:34 13 Ma...	8080	13
7	http://172.16.111.8	GET	/dvwa/login.php			200	1638	HTML	php	Damn Vulnerable Web App ...			172.16.111.8		21:10:35 13 Ma...	8080	13

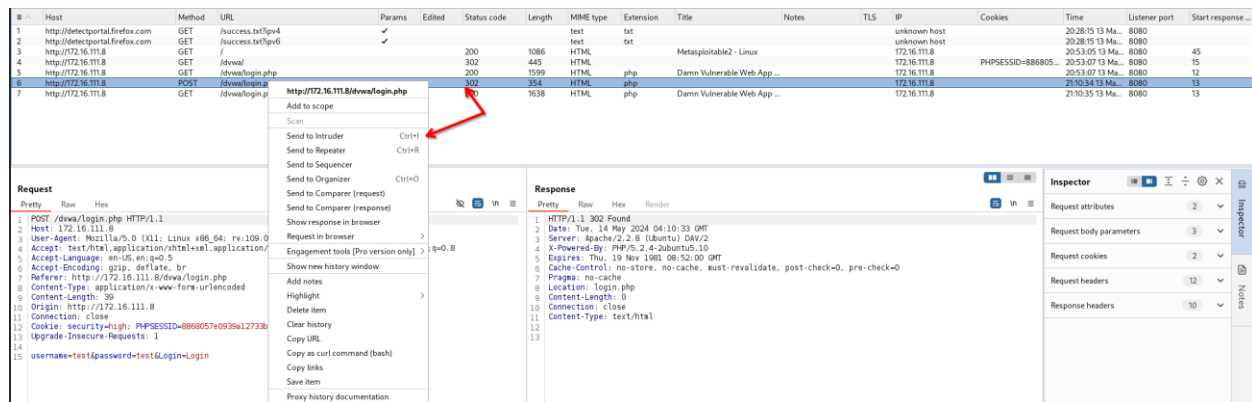
Request	Response
<pre>1 POST /dvwa/login.php HTTP/1.1 2 Host: 172.16.111.8 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0 4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate, br 7 Referer: http://172.16.111.8/dvwa/login.php 8 Content-Type: application/x-www-form-urlencoded 9 Content-Length: 39 10 Origin: http://172.16.111.8 11 Connection: close 12 Cookie: security=1; PHPSESSID=8868057e0939a1273384b2c78622e341 13 Upgrade-Insecure-Requests: 1 14 15 username=test&amp;password=test&amp;login=Login</pre>	<pre>1 HTTP/1.1 302 Found 2 Date: Tue, 14 May 2024 04:10:33 GMT 3 Server: Apache/2.2.8 (Ubuntu) DAV/2 4 X-Powered-By: PHP/5.2.4-2ubuntu5.10 5 Expires: Thu, 19 Nov 1981 08:52:00 GMT 6 Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0 7 Pragma: no-cache 8 Location: login.php 9 Content-Length: 0 10 Connection: close 11 Content-Type: text/html</pre>

Burp Suite has logged a POST request, and it also detects parameters in the request as well. Burp Suite not only shows that a POST request was made, but it also shows the content of both the request and the response. In the “Request” section, there are three parameters passed, two of which are “username” and “password”, which the value “test” was passed for.

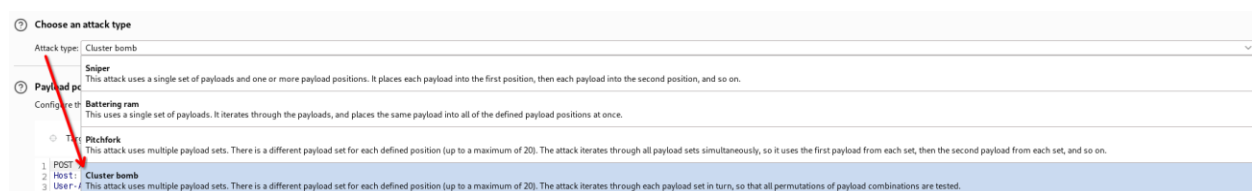
## Step 2: Configuring Payload Request

Now that the POST request has been logged, it needs to be configured into a payload that will be used for the brute force attack. Burp Suite contains a variety of tools that can be helpful when it comes to web application exploitation. One of these tools is “Intruder” which enables the user to configure attacks that send the same HTTP request repeatedly.

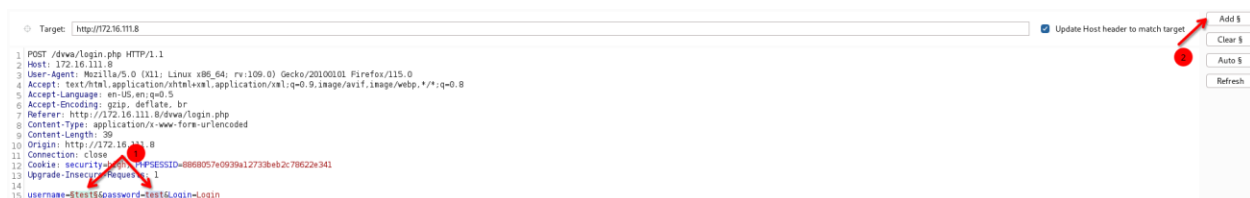
The logged POST request can be sent to intruder by right clicking the request and selecting the “Send to Intruder” option. There is also a hotkey to achieve this same function which is Ctrl+I.



The “Intruder” tab will now be glowing. Navigate to this tab and the logged POST request should appear under the “Payload positions” section. Above that section, there will be a drop-down section where the user can select different attack types, each serving their own purpose. Since multiple parameters are being targeted, the attack type that is most appropriate is “Cluster Bomb”, which will allow the user to utilize multiple payload sets instead of just one.

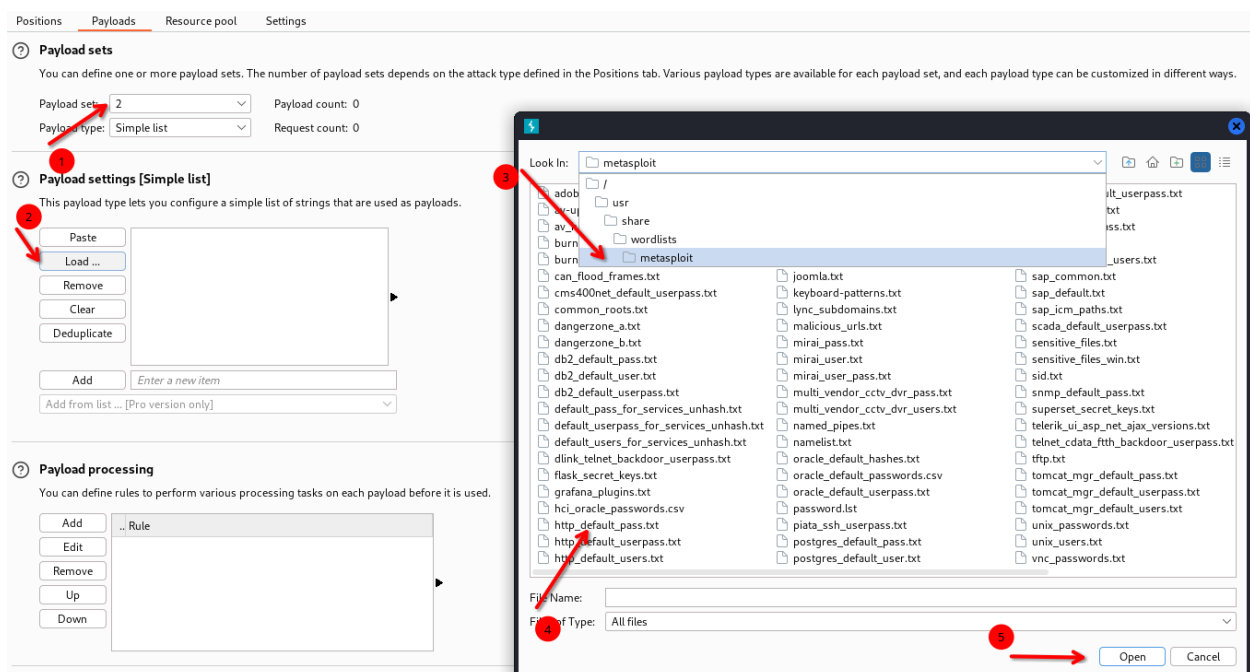


Using the “Cluster Bomb” attack type will allow us to target both the username and the password fields in the same attack. These fields need to be specially marked to tell Burp Suite that they are being targeted. Since the fields aren’t being targeted, the values provided for the fields in the logged POST request are what need to be specially marked. Highlight the values provided for the username and password field and click the “Add \$” button to mark each value.



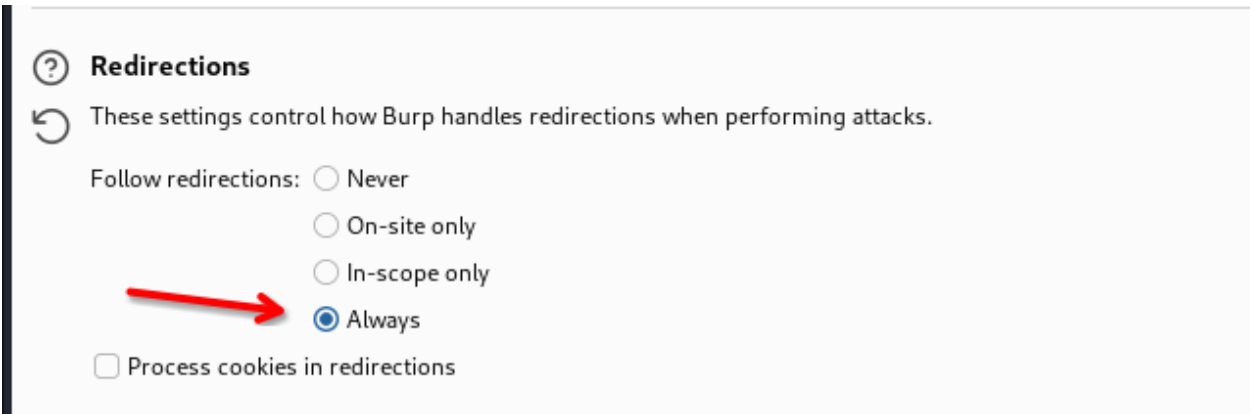
Now that both fields have been specially marked, the payload sets can be configured under the "Payloads" tab which can be found directly above the attack type section. The username value was marked first so that will be payload set 1 which makes the password value payload set 2.

For each payload set, the user can configure a payload type to use for the set. The payload type used for both sets in this demonstration will be simple list. For each set, load a wordlist containing usernames and passwords that the attack will use to enumerate through, during the brute force. Kali Linux installations come with pre-installed wordlists for the most common usernames and passwords used for http users (These can be found under /usr/share/wordlists/metasploit).



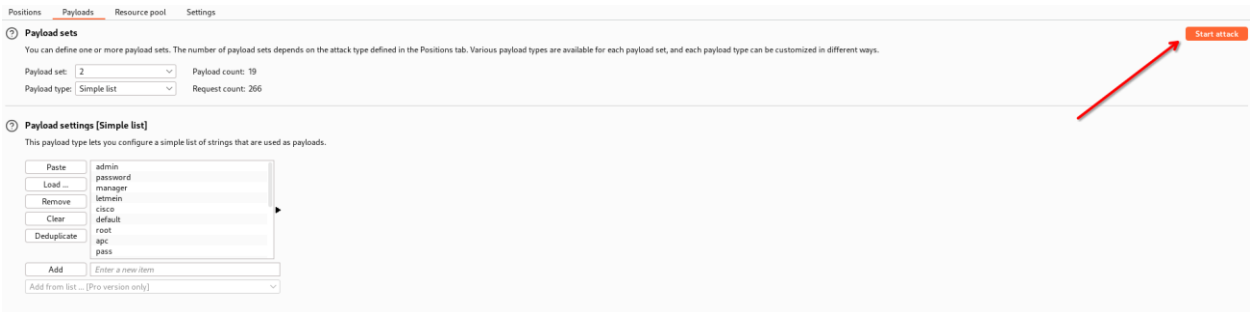
Both payload sets have successfully been configured but there is one more step before the attack is ready to be executed. The attack needs to be configured to follow redirects so that it will be easy to detect when a good set of credentials has been passed. This can be done in the "Settings" tab of the "Intruder" tool. The Settings tab can be found next to the "Payloads" tab

Scroll down to the bottom of the “Settings” tab to find the “Redirections” section and make sure to select the “Always” option so Burp will always follow redirects, during the attack.



Step 3: Analyzing Attack Results

The attack can be executed by clicking the orange “Start Attack” button located on the top right of the “Payload” page.



Now the attack will start executing and results will begin displaying on the screen. The attack will use the lists loaded in each payload set to enumerate through each value and send the same POST request to the web server, logging information about each request made.

Request	Payload 1	Payload 2	Status code	Response received	Error	Redirects followed	Timeout	Length	Comment
0			200	48		1		3291	
1	admin	admin	200	47		1		1675	
2	manager	admin	200	30		1		1675	
3	root	admin	200	27		1		1675	
4	cisco	admin	200	31		1		1675	
5	apc	admin	200	27		1		1675	
6	pass	admin	200	29		1		1675	
7	security	admin	200	35		1		1675	
8	user	admin	200	34		1		1675	
9	system	admin	200	34		1		1675	
10	sys	admin	200	26		1		1675	

Some of the information displayed about each request is what value was tried for each payload, the status code received, and the length of the response that was received. The most important field to pay attention to is the “Length” field. The length of the response received in the initial logged request where wrong credentials were intentionally entered was 1638 bytes. So, it can be assumed that all responses to requests made with incorrect

credentials can be around that length. Most of the requests are receiving responses of around 1675 bytes.

Sort the length column in a descending fashion, so the largest values are shown at the top. This is done because it is expected that the requests that pass valid credentials will follow a redirection, making the received response larger in size.

Results Positions Payloads Resource pool Settings									
▼ Intruder attack results filter: Showing all items									
Request	Payload 1	Payload 2	Status code	Response received	Error	Redirects followed	Timeout	Length	Comment
15	admin	password	200	30		1		4932	
0			200	48		1		3291	
1	admin	admin	200	47		1		1675	
2	manager	admin	200	30		1		1675	
3	root	admin	200	27		1		1675	
4	cisco	admin	200	31		1		1675	
5	apc	admin	200	27		1		1675	
6	pass	admin	200	29		1		1675	
7	security	admin	200	35		1		1675	
8	user	admin	200	34		1		1675	
9	system	admin	200	34		1		1675	

The fifteenth request sent during this attack received a response with a length of 4932 bytes. "admin" was used as the username and "password" was used as the password. What makes this request's length so much larger is the size of the second response. During the other requests that pass incorrect credentials, the redirects are followed back to the login page.

When correct credentials are passed, the second response will be passing content from an entirely different page (The DVWA homepage), making the length of the response much different than the other responses when incorrect credentials were passed. The "Render" tool can be used to help the user see what Burp Suite is seeing. Select the request with the larger length, select "Response 2" tab, and select "Render" just underneath the "Response 2" tab.



Request	Payload 1	Payload 2	Status code	Response received	Error	Redirects followed	Timeout	Length	Comment
15	admin	password	200	30		1		4932	
0			200	48		1		3291	
1	admin	admin	200	47		1		1675	
2	manager	admin	200	30		1		1675	
3	root	admin	200	27		1		1675	
4	cisco	admin	200	31		1		1675	
5	apc	admin	200	27		1		1675	
6	hask	admin	200	29		1		1675	
7	secu	admin	200	35		1		1675	
8	user	admin	200	34		1		1675	
9	system	admin	200	34		1		1675	

Request 1    Response 1    Request 2    Response 2

Pretty    Raw    **Render**

The screenshot shows the DVWA homepage with a sidebar menu on the left containing links like Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP info, About, and Logout. The main content area has a header 'Welcome to Damn Vulnerable Web App!' followed by a warning, disclaimer, and general instructions. At the bottom, a status box indicates 'You have logged in as \'admin\''.

The second response of the fifteenth entry has been rendered to display the DVWA homepage, validating that the credentials passed in the request are valid.

### 3. Validation

To properly validate the findings acquired during the result analysis, the credentials from the request can be used during a login attempt to see if access is granted to the DVWA homepage. This can be done by, first, navigating to the DVWA login page. Once at the homepage, use “admin” for the username field and “password” for the password field. After this information has been entered, the ‘Login’ button can be pressed.

If the credentials are valid, after clicking the “Login” button, the user should be redirected to the DVWA homepage. This should look like the page rendered in the second response rendered in the previous step.



Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout

Username: admin  
Security Level: high  
PHPIDS: disabled

## Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

### WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing [XAMPP](#) onto a local machine inside your LAN which is used solely for testing.

### Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

### General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

You have logged in as 'admin'

Damn Vulnerable Web Application (DVWA) v1.0.7

The acquired credentials have worked and redirection to the DVWA homepage has worked.

## Best Practices

Brute forcing can be invasive. This is because multiple access attempts to the same account can occur over a short period of time and cause the account to become locked out, forcing action from the actual user to unlock the account. To help avoid this, try a spraying approach where only a few passwords are tried for every username. If a valid set of credentials was not detected in the few attempts, move on to the next account.

Rendering pages from response data in Burp Suite is a great way for the user to see exactly what Burp Suite is seeing as the requests are being made and the responses are being received. During this demonstration, the rendering feature allowed the tester to see that the DVWA homepage was rendered after a valid set of credentials was passed. Utilize this

feature, when applicable, to help see how the web application is responding to user testing.

Proxy Settings are configured in the browser to use Burp Suite as a proxy for web requests being made through the browser. After testing has concluded, be sure to disable the browser proxy settings to continue to browse the internet as normal. If the proxy settings are not disabled after Burp Suite is closed, the user may encounter proxy errors when attempting to browse the web normally.

## Conclusion

Brute forcing can be an effective method to gain unauthorized access to an account, by means of enumerating through every possible combination for a password or by providing a pre-defined wordlist that the attack can enumerate through. Though effective, it can be an invasive procedure, as it often involves multiple sign in attempts to a single account, which can cause account lockout, if none of the attempts were successful.

## References

<https://www.vulnhub.com/entry/metasploitable-2,29/>

## Revision History

05/11/2024 - Overview, Background

05/13/2024 - Preparation, Step 1, Step 2, Step 3

05/14/2024 - Best Practices, Conclusion, References