Password Cracking Web Forms with Hydra and Burp Suite

02/09/2021

Prerequisites:

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

Kali Linux

Both virtual machines need to be installed and on the same network.

Burp Suite setup:

To set up Burp Suite you need to add and setup foxy proxy on kali's web browser.

Go to settings, add-ons, and search for and install foxy proxy.

Next, select "options" from the proxy logo in your browser tool bar.

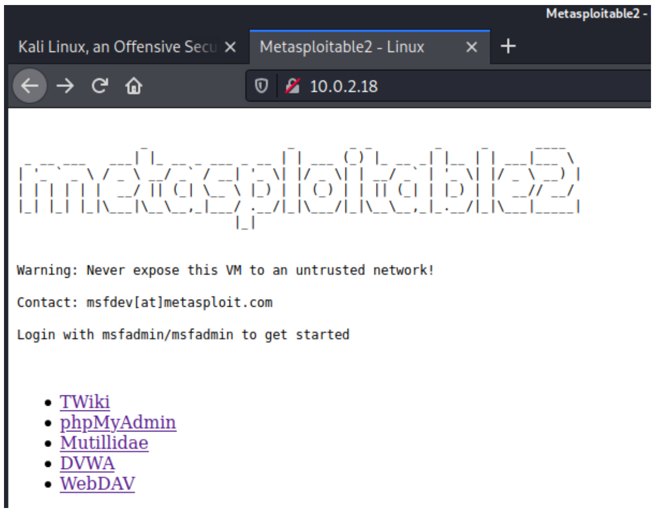
Select the "add" option in the upper left corner and add a proxy labeled "Burp Suite" with the default type of "HTTP" and use 127.0.0.1 for the IP address and port 8080 as the port. Leave the other options blank. For now, leave the proxy set to "Turn off" in the main drop-down menu for it from the toolbar.

Metasploitable setup:

Start a new machine on VirtualBox and mark "Other Linux 64-bit" as the type. Leave defaults except for when it asks for a disk select "Use an existing virtual hard disk file" and select the .vmdk file for Metasploitable.

You can either nmap your network from Kali (recommended), or you can login to the Metasploitable machine with msfadmin:msfadmin credentials to determine the IP.

Once the IP is determined, navigate to that IP in the web browser of Kali and select the "DVWA" hyperlink.



```
Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started
```

- TWiki
- phpMyAdmin
- Mutillidae
- DVWA
- WebDAV

Hydra Hacking:

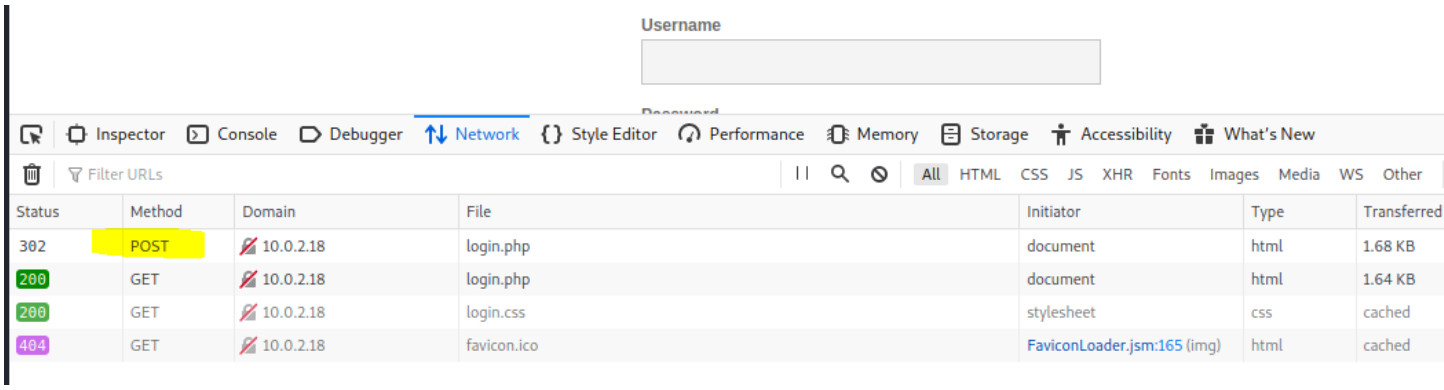
Hydra is tricky to use as it is a command line tool, but once you know the proper way to build the commands it isn't so bad.

For our purposes, we first need to do some recon.

With the DVWA login open in Kali's browser, right click on the username field, and select "inspect element".

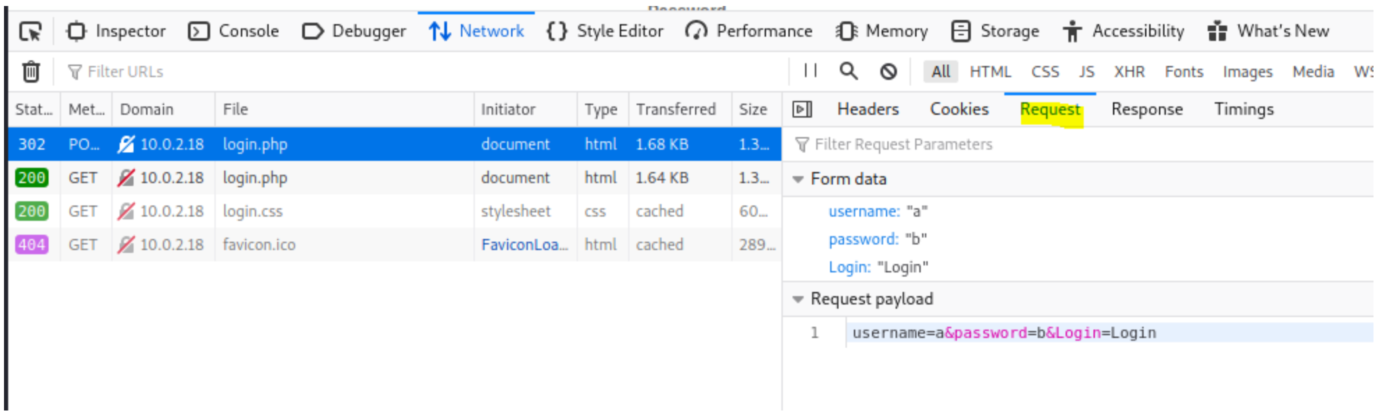
On the top pane of this developer toolbox, select the "Network" option.

Leaving the developer's tools open, attempt to login with a random one-digit username and password. You should receive a POST method now in the developer tools window:



Status	Method	Domain	File	Initiator	Type	Transferred
302	POST	10.0.2.18	login.php	document	html	1.68 KB
200	GET	10.0.2.18	login.php	document	html	1.64 KB
200	GET	10.0.2.18	login.css	stylesheet	css	cached
404	GET	10.0.2.18	favicon.ico	FaviconLoader.jsm:165 (img)	html	cached

From this POST request we need the information available on the "Request" tab to the right:



Stat...	Met...	Domain	File	Initiator	Type	Transferred	Size
302	PO...	10.0.2.18	login.php	document	html	1.68 KB	1.3...
200	GET	10.0.2.18	login.php	document	html	1.64 KB	1.3...
200	GET	10.0.2.18	login.css	stylesheet	css	cached	60...
404	GET	10.0.2.18	favicon.ico	FaviconLoa...	html	cached	289...

Form data

```
username: "a"
password: "b"
Login: "Login"
```

Request payload

```
username=a&password=b&Login=Login
```

The "Form data" will be used in building our Hydra command. We also need to not the directory of the site this form is on as /dvwa/login.php.

Another important thing to note here is the "Login failed" response that the page gives when the login attempt does not work.

Let's breakdown all the switches we will be using for Hyrda:

-L - Specifies a word list is being used for the "user" parameter.

-P - Specifies that a word list is being used for the "password" parameter.

http-post-form - Specifies the type of request, “post” in our case.

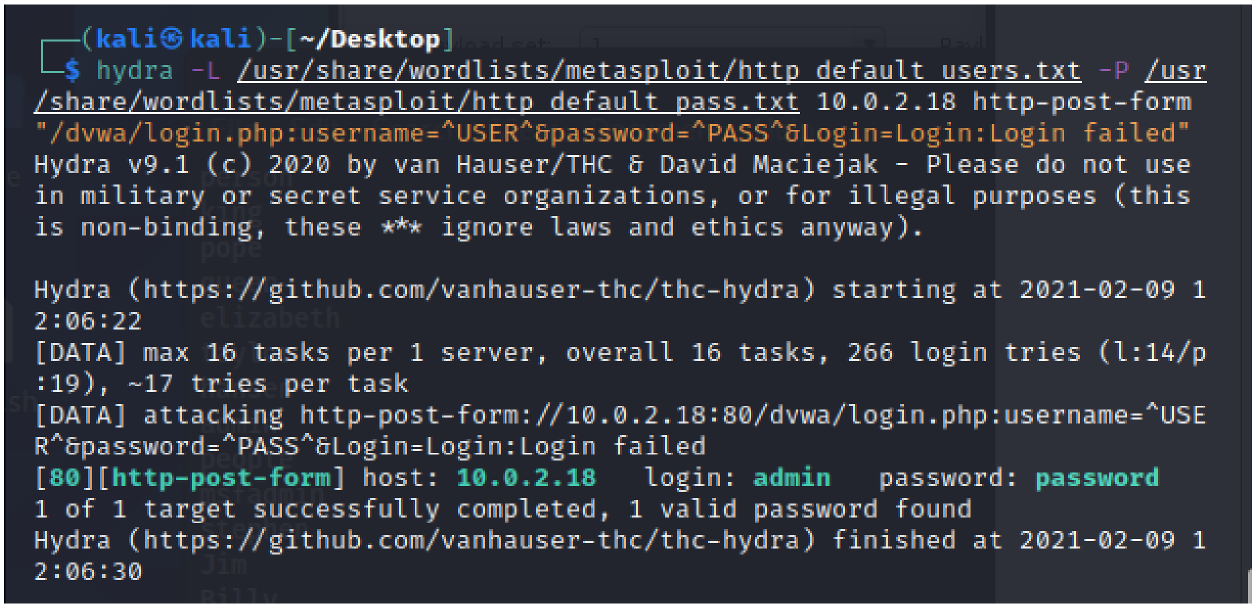
“/dvwa/login/php:username=^USER^&password=^PASS^&Login=Login failed” - this specifies that our variables between the carets, and uses the form data and the failed login responses we collected above.

The wordlists we are going to use for this command are on /usr/share/wordlists/metasploit and are the: http_default_users.txt for Users and http_default_pass.txt for Password.

When we combine all these with the IP of our Metasploitable machine we get (in my instance 10.0.2.18 is the IP of this machine, yours will probably be different):

Hyrda -L /usr/share/wordlists/metasploit/http_default_users.txt -P /usr/share/wordlists/metasploit/http_default_pass.txt 10.0.2.18 http-post-form “/dvwa/login.php:username=^USER^&password=^PASS^&Login=Login:Login failed”

It should quickly spit out your results:



```
┌──(kali㉿kali)-[~/Desktop]
└─$ hydra -L /usr/share/wordlists/metasploit/http_default_users.txt -P /usr
/share/wordlists/metasploit/http_default_pass.txt 10.0.2.18 http-post-form
"/dvwa/login.php:username=^USER^&password=^PASS^&Login=Login:Login failed"
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use
in military or secret service organizations, or for illegal purposes (this
is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-02-09 1
2:06:22
[DATA] max 16 tasks per 1 server, overall 16 tasks, 266 login tries (l:14/p
:19), ~17 tries per task
[DATA] attacking http-post-form://10.0.2.18:80/dvwa/login.php:username=^USE
R^&password=^PASS^&Login=Login:Login failed
[80][http-post-form] host: 10.0.2.18   login: admin   password: password
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-02-09 1
2:06:30
```

Awesome, now let’s login and get setup for the next attack. Once logged in, set the DVWA security to low from the sidebar, and select the “Brute Force” option.

Once that is done, go ahead and select “Burpsuite” from the foxy proxy drop down on the browser’s toolbar.

Burp Suite cracking:

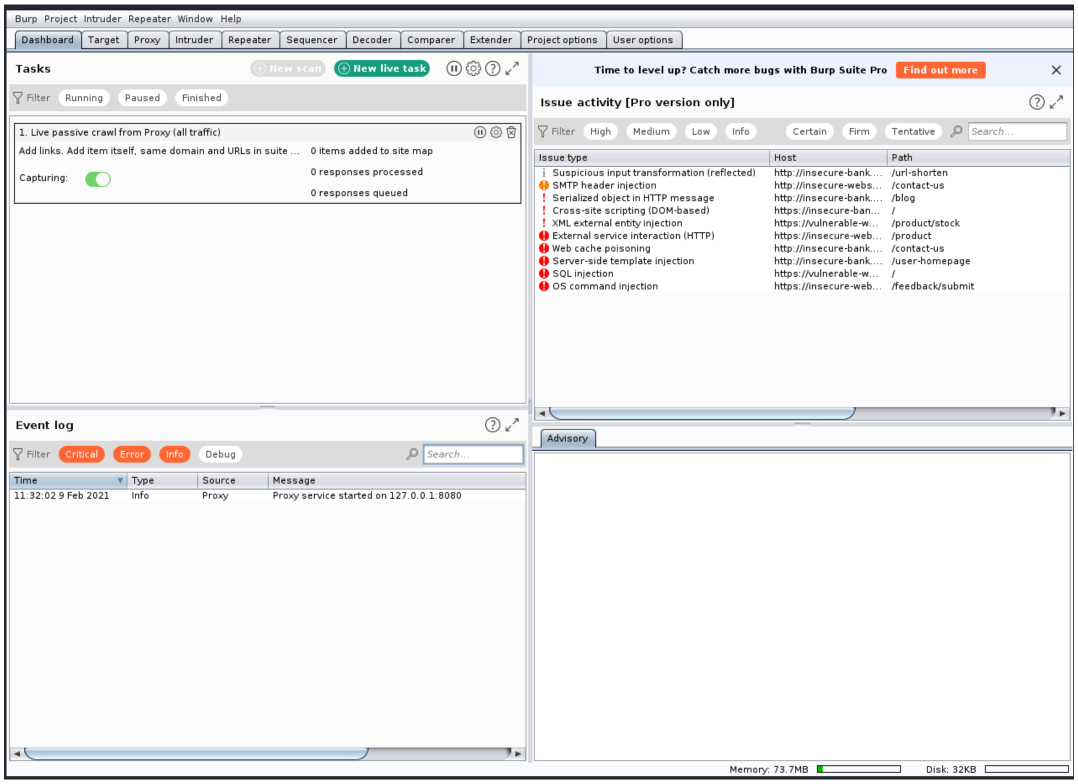
Burp Suite is unfortunately slow on purpose to make you pay for it. As a result, we are going to have to create our own very small, short wordlists to demonstrate how it works or else it will take a long time.

Create one list with the name “user” with the following words: user, root, admin, super, manager.

Create another list called “pass” with the following words: password, password1, secret, toor, letmein.

Save these two files wherever, just remember the path.

Next, start Burp Suite and leave all the default settings and skip ahead until you get the main screen:


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
Tasks
New scan
New live task
Filter Running Paused Finished
Time to level up? Catch more bugs with Burp Suite Pro
Find out more
Filter High Medium Low Info Certain Firm Tentative
Search...
Event log
Filter Critical Error Info Debug
Advisory
Memory: 73.7MB
Disk: 32KB

1. Live passive crawl from Proxy (all traffic)
Add links. Add item itself, same domain and URLs in suite ...
0 items added to site map
0 responses processed
0 responses queued
Capturing:

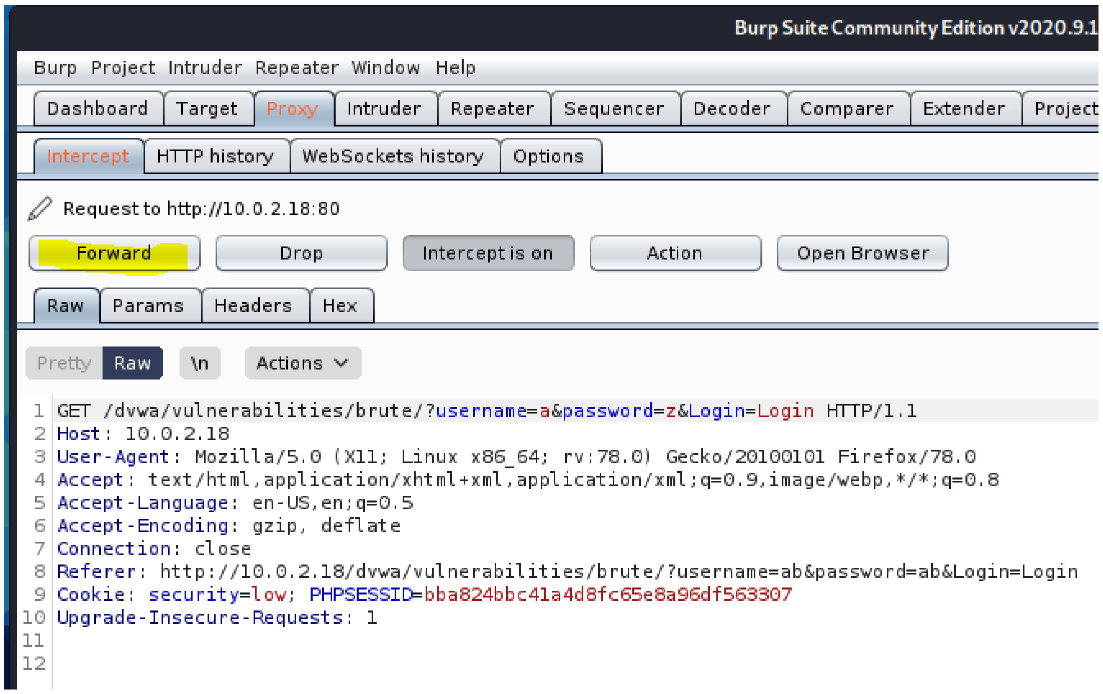
Issue activity [Pro version only]

Issue type	Host	Path
Suspicious input transformation (reflected)	http://insecure-bank....	/url-shorten
SMTP header injection	http://insecure-webs...	/contact-us
Serialized object in HTTP message	http://insecure-bank....	/blog
Cross-site scripting (DOM-based)	https://insecure-ban...	/
XML external entity injection	https://vulnerable-w...	/product/stock
External service interaction (HTTP)	https://insecure-web...	/product
Web cache poisoning	http://insecure-bank....	/contact-us
Server-side template injection	http://insecure-bank....	/user-homepage
SQL injection	https://vulnerable-w...	/
OS command injection	https://insecure-web...	/feedback/submit

Time	Type	Source	Message
11:32:02 9 Feb 2021	Info	Proxy	Proxy service started on 127.0.0.1:8080

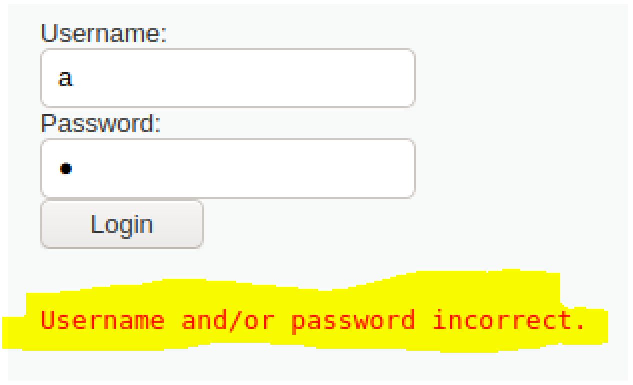
From here, select the “Proxy” tab.

On the DVWA page, try and login with single letter digits for username and password. One, you do this, you will need to go back to the “Intercept” tab of the “Proxy” menu in Burpsuite and hit the “Forward” Button a few times (until you no longer can):


Burp Suite Community Edition v2020.9.1
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project
Intercept HTTP history WebSockets history Options
Request to http://10.0.2.18:80
Forward Drop Intercept is on Action Open Browser
Raw Params Headers Hex
Pretty Raw \n Actions

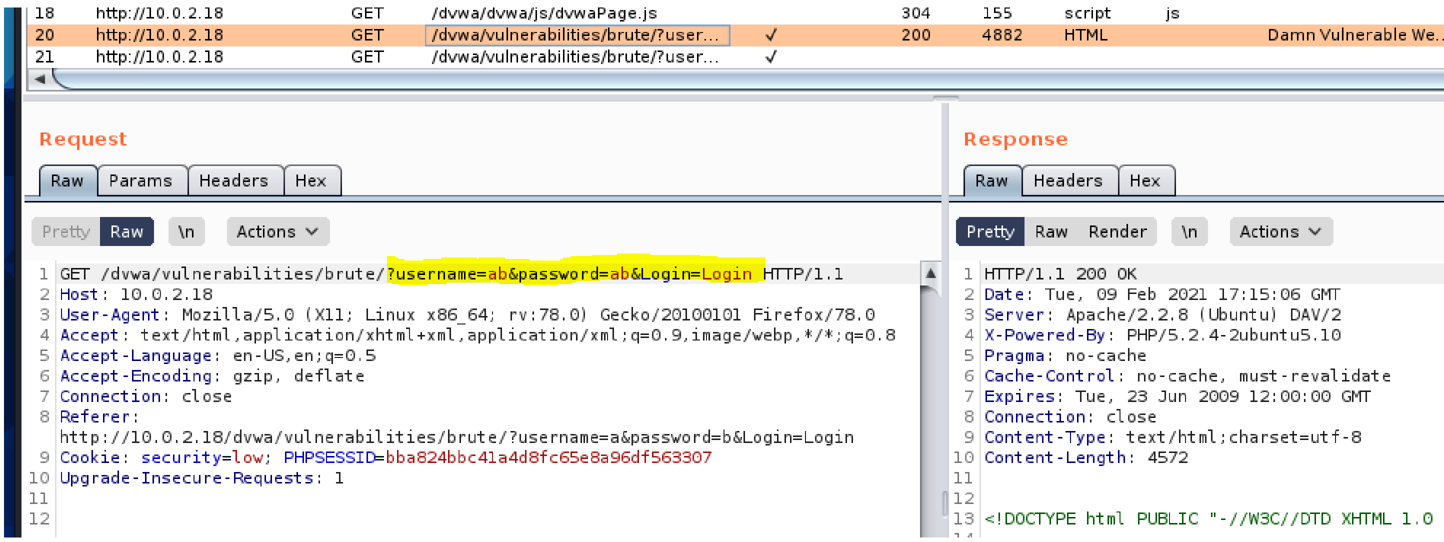
```
GET /dvwa/vulnerabilities/brute/?username=a&password=z&Login=Login HTTP/1.1
Host: 10.0.2.18
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://10.0.2.18/dvwa/vulnerabilities/brute/?username=ab&password=ab&Login=Login
Cookie: security=low; PHPSESSID=bba824bbc41a4d8fc65e8a96df563307
Upgrade-Insecure-Requests: 1


```

Go back to the page and take note of the failed login message.



```
Username and/or password incorrect.
```

Next, return to Burp Suite and select the "HTTP" history tab of the Proxy menu and you should have a GET request that looks like this (see the username and password at the top):



```
18  http://10.0.2.18  GET  /dvwa/dvwa/js/dvwaPage.js                304  155   script  js
20  http://10.0.2.18  GET  /dvwa/vulnerabilities/brute/?user...  ✓  200  4882  HTML        Damn Vulnerable We...
21  http://10.0.2.18  GET  /dvwa/vulnerabilities/brute/?user...  ✓
```

```
GET /dvwa/vulnerabilities/brute/?username=ab&password=ab&Login=Login HTTP/1.1
Host: 10.0.2.18
User-Agent: Mozilla/5.0 (X11; Linux x86_64; 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
Connection: close
Referer:
http://10.0.2.18/dvwa/vulnerabilities/brute/?username=a&password=b&Login=Login
Cookie: security=low; PHPSESSID=bba824bbc41a4d8fc65e8a96df563307
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Date: Tue, 09 Feb 2021 17:15:06 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Pragma: no-cache
Cache-Control: no-cache, must-revalidate
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Connection: close
Content-Type: text/html;charset=utf-8
Content-Length: 4572


<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
```

Once you find this, select the “actions” drop down menu on the **left** and select “send to intruder”.

Navigate to the “Intruder’ top at the top, leave the target and port the defaults, and select the “Positions” tab.

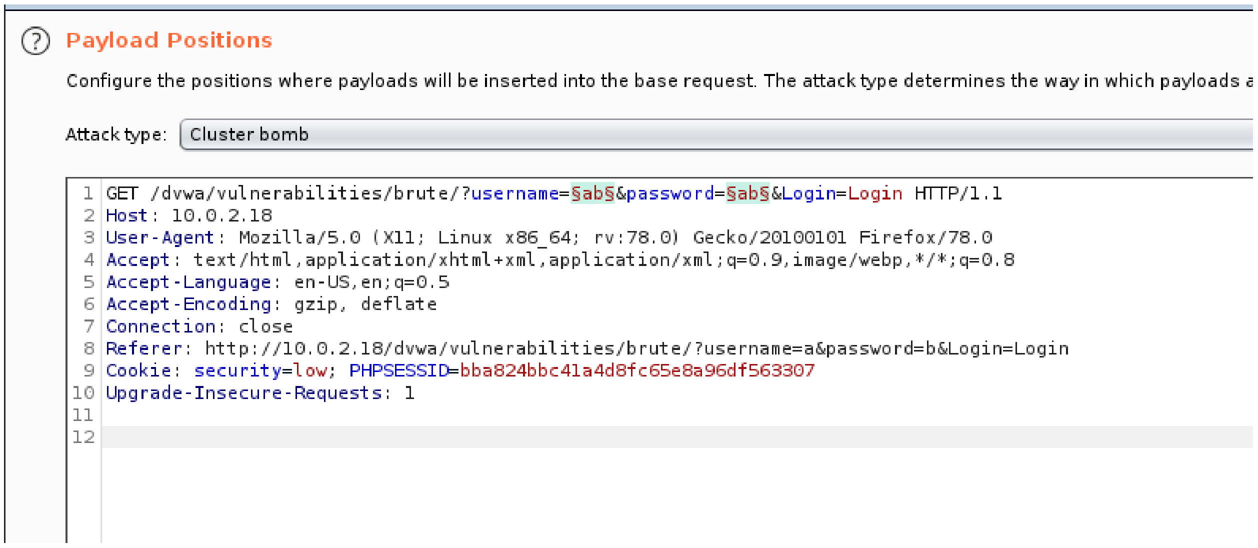
Select the “Clear” button on the right side of the screen.

Next, highlight the text between “=” after username and the “&” before password and click add. This tells Burp Suite that this will be a variable.

Do the same thing with the password field.

Next, select the “Cluster Bomb” attack type from the drop-down menu. This indicates we will be using multiple payloads (different wordlists).

Payload Positions should look like this:



Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads a

Attack type: Cluster bomb

```
GET /dvwa/vulnerabilities/brute/?username=§ab§&password=§ab§&Login=Login HTTP/1.1
Host: 10.0.2.18
User-Agent: Mozilla/5.0 (X11; Linux x86_64; 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
Connection: close
Referer: http://10.0.2.18/dvwa/vulnerabilities/brute/?username=a&password=b&Login=Login
Cookie: security=low; PHPSESSID=bba824bbc41a4d8fc65e8a96df563307
Upgrade-Insecure-Requests: 1
```

Select the “Payload” tab of the Intruder menu.

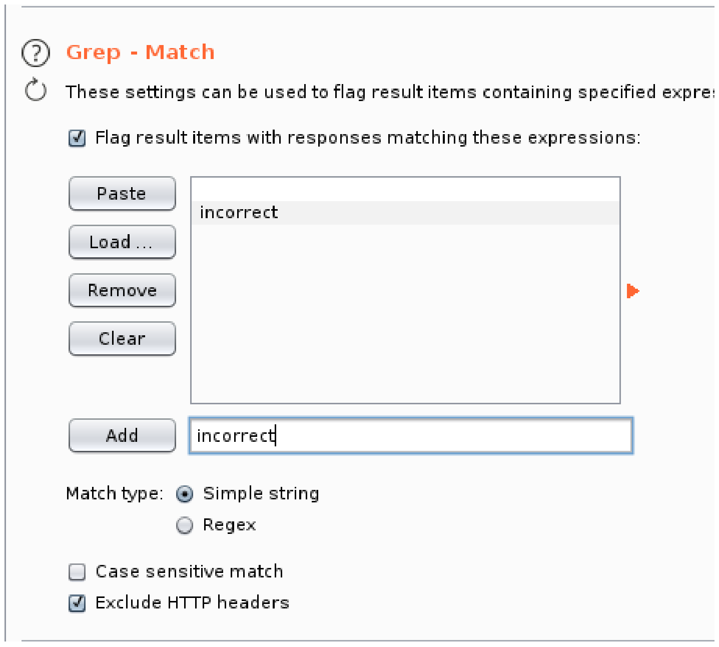
Under “Payload Sets”, make sure the “1” is selected.

In “Payload Options”, select the “Load” button and add the “user” wordlist you made earlier.

Next, Select the “2” option from the “Payload Sets” menu.

Repeat the process from above to add the “pass” wordlist from earlier.

Now select the “Options” tab of the Intruder menu. Scroll down to “Grep Match”. Clear everything in there out and add a keyword from our “Username and/or password incorrect.” failed login response from earlier with the default options:


Grep - Match

These settings can be used to flag result items containing specified expre

[✓] Flag result items with responses matching these expressions:

incorrect

Add: incorrect

Match type: (●) Simple string ( ) Regex

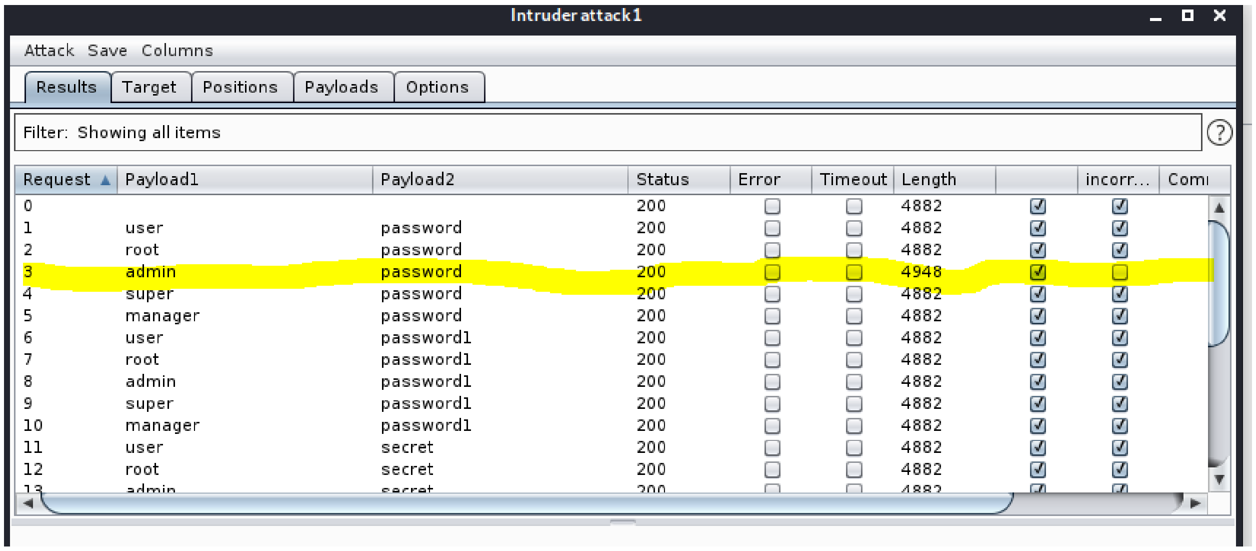
[ ] Case sensitive match

[✓] Exclude HTTP headers

Ready to go!

Click “Start Attack” on the upper right side.

You should see a list start to appear on the right side. Notice that one does not have the box under “incorrect” check marked?


Intruder attack1

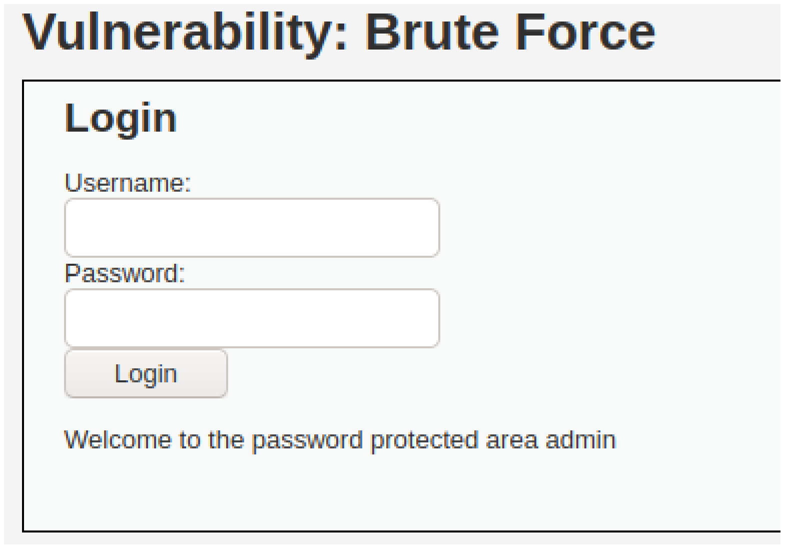
Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Request	Payload1	Payload2	Status	Error	Timeout	Length		incorr...	Comı
0			200	☐	☐	4882	☑	☑	
1	user	password	200	☐	☐	4882	☑	☑	
2	root	password	200	☐	☐	4882	☑	☑	
3	admin	password	200	☐	☐	4948	☑	☐	
4	super	password	200	☐	☐	4882	☑	☑	
5	manager	password	200	☐	☐	4882	☑	☑	
6	user	password1	200	☐	☐	4882	☑	☑	
7	root	password1	200	☐	☐	4882	☑	☑	
8	admin	password1	200	☐	☐	4882	☑	☑	
9	super	password1	200	☐	☐	4882	☑	☑	
10	manager	password1	200	☐	☐	4882	☑	☑	
11	user	secret	200	☐	☐	4882	☑	☑	
12	root	secret	200	☐	☐	4882	☑	☑	
13	admin	secret	200	☐	☐	4882	☑	☑	

That means we god a result that was did not have "incorrect" in the response. They are the same credentials from earlier, let's give them a try (Turn off the proxy first):


Vulnerability: Brute Force

Login

Username:

Password:

Welcome to the password protected area admin

Success!