iGNITE Technologies

BURP SUITE FOR PENTESTER PAYLOAD PROCESSING

WWW.HACKINGARTICLES.IN

TABLE OF CONTENTS

Abstract

Today we are going to discuss the "Payload Processing" option of Burpsuite which is advanced functionality comes under the **Intruder Tab** in order to perform fuzzing or a brute force attack.
However, you might be aware about how fuzzing is done with the Intruder tab, so let's explore some of its advanced techniques, such that it could enhance our bug hunting methodologies.

Payload Processing

Payload Processing

Payload Processing can be defined as when payloads are generated using **payload types**, they can be further **manipulated or filtered** using various **processing rules** and **payload encoding**.

Payload Processing Rules

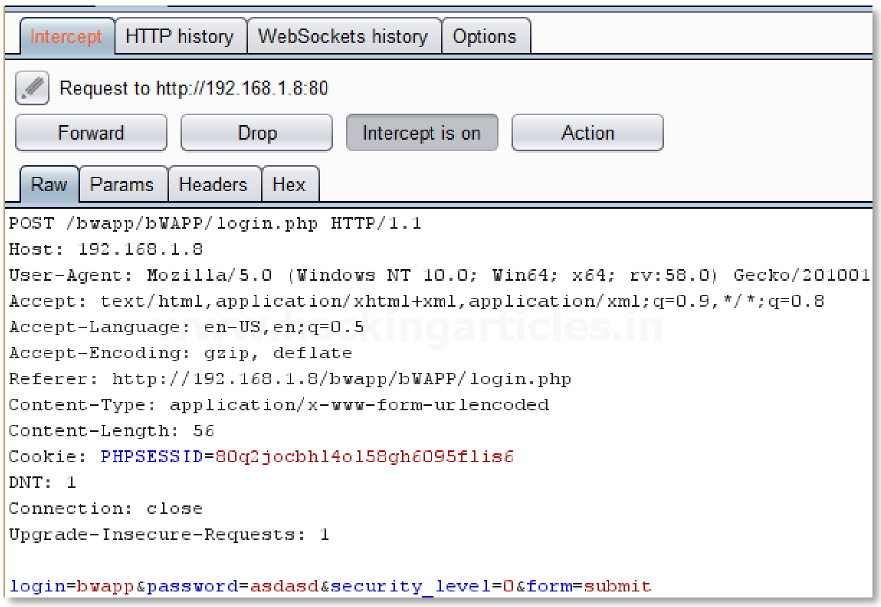
These rules are defined to perform a various processing task on each payload before it is used. These rules are executed in a sequence, and they can be used to help debug any problem with the configuration. Payload processing rules are useful in situations where you need to generate different payloads, or where we want to wrap payloads within a wider structure or encoding scheme.

There are 12 types of payload processing rules available:

- **Add prefix**
- **Add suffix**
- **Match / Replace**
- **Substring**
- **Reverse substring**
- **Modify case**
- **Encode**
- **Decode**
- **Hash**
- **Add raw payload**
- **Skip if matches regex**
- **Invoke Burp extension**

Add Prefix

This processing rule adds up a prefix before the payload.
First, we have intercepted the request of the login page in the **Bwapp LAB**, where we have given default username and wrong password. Then click on login, the burp suite will capture the request of the login page in the intercept tab.



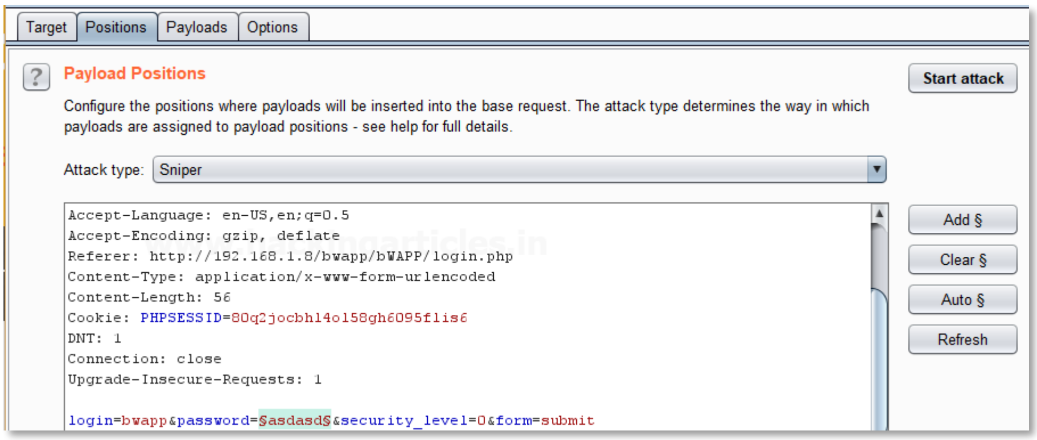
Request to http://192.168.1.8:80

```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/201001
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.8/bwapp/bWAPP/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 56
Cookie: PHPSESSID=80q2jocbh14o158gh6095flis6
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

login=bwapp&password=asdasd&security_level=0&form=submit
```

Send the captured request to the **Intruder** by clicking on the Action Tab and follow given below step. Now open the **Intruder tab** then select **Positions tab** and you can observe the highlighted password and follow the given below step for selecting payload position.

- Press on the **Clear button** given at right of the window frame.
- Now we will select the fields where we want to attack and i.e. the password filed and click on **Add button.**
- Choose the **Attack type** as **a sniper**
- In the given below image, we have selected a password that means we will need one dictionary files for a password.



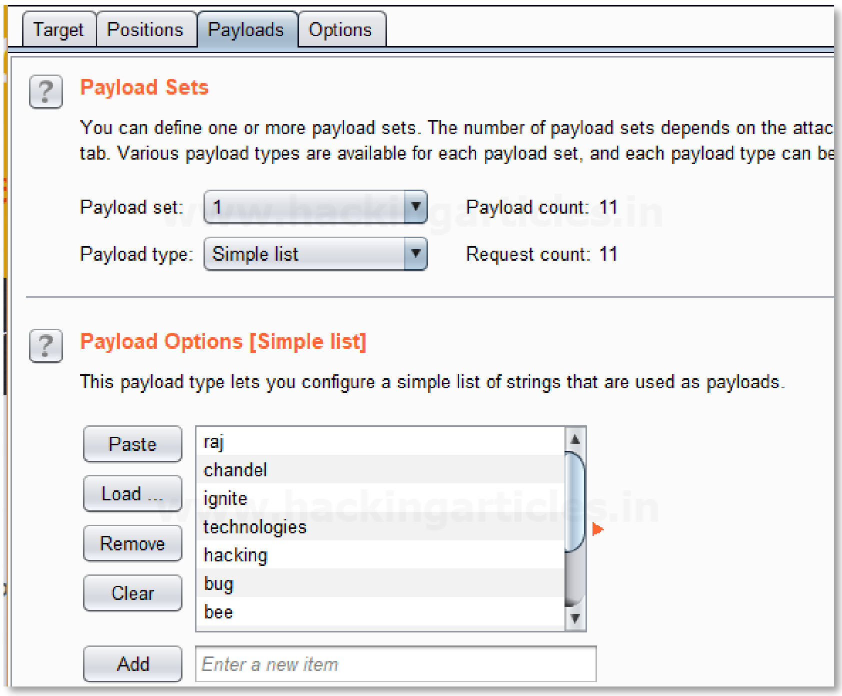
Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Sniper

```
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.8/bwapp/bWAPP/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 56
Cookie: PHPSESSID=80q2jocbh14o158gh6095flis6
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

login=bwapp&password=§asdasd§&security_level=0&form=submit
```

Now click on **payloads option** after selecting payload position. Then select the **Payload type** as **Simple list,** where we have added a dictionary by clicking on **Load** button. We can either load the dictionary or we can manually add input strings using the **Add button** in the **payload options** as shown in the image.



Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attac
tab. Various payload types are available for each payload set, and each payload type can be

Payload set: 1 Payload count: 11

Payload type: Simple list Request count: 11

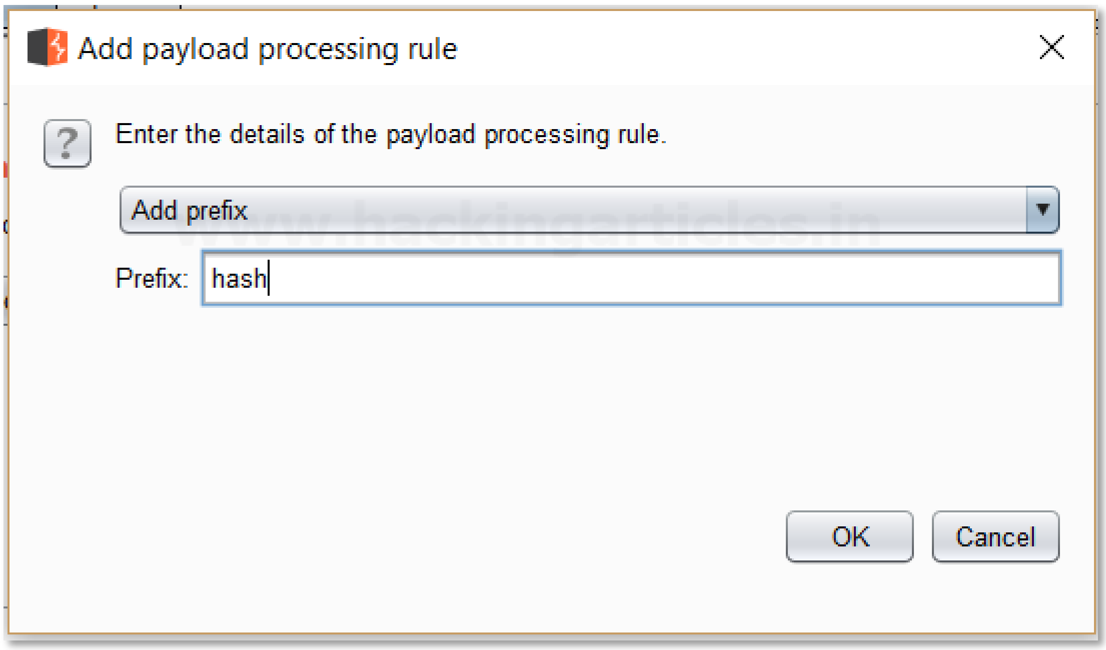
Payload Options [Simple list]

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

raj
chandel
ignite
technologies
hacking
bug
bee

Enter a new item

Before executing the attack we have added a **payload processing rule** to the payload type which is **Add Prefix** and we have given an input string "hash" which is added as a **prefix** with every input strings in the dictionary, as shown in the **result window** of the **attack**.
Select **Start Attack** in the **Intruder menu** as shown in the image.



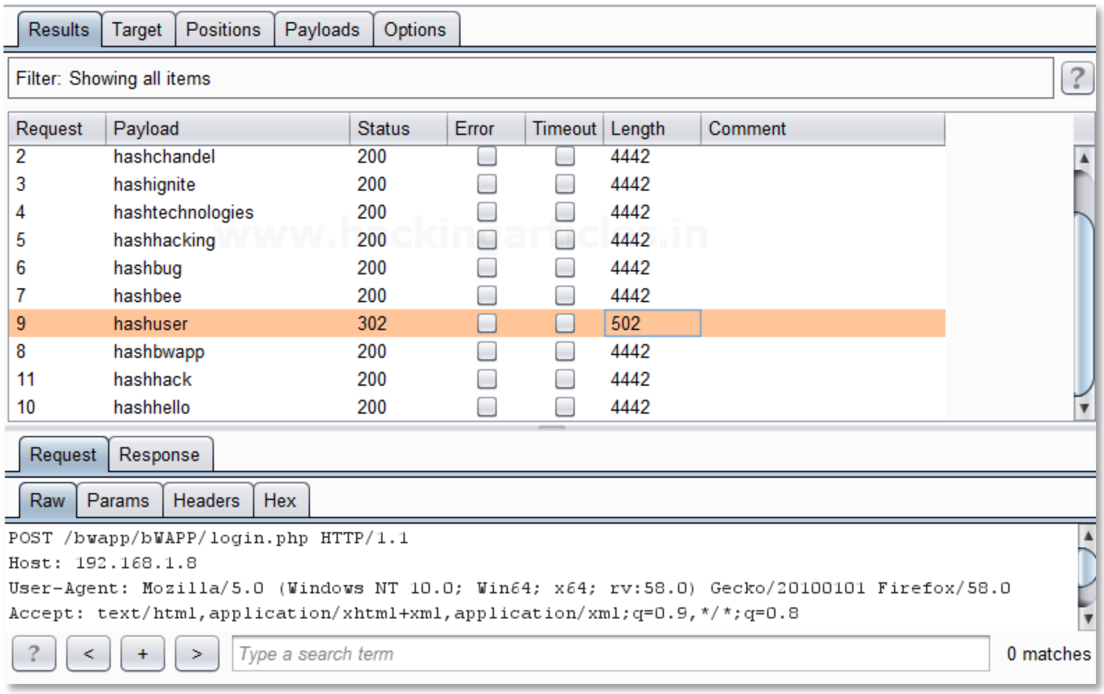
Add payload processing rule

Enter the details of the payload processing rule.

Add prefix

Prefix: hash

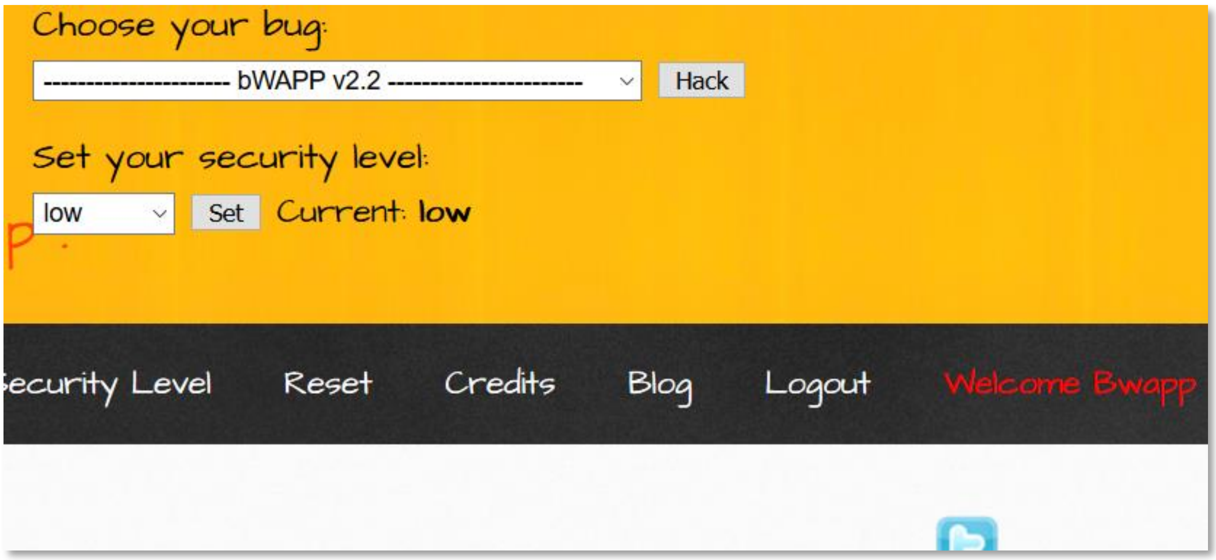
Sit back and relax because now the burp suite will do its work, match the password which will give you the correct password. The moment it will find the correct value, it will change the value of length as shown in the image.



Request	Payload	Status	Error	Timeout	Length	Comment
2	hashchandel	200	☐	☐	4442	
3	hashignite	200	☐	☐	4442	
4	hashtechnologies	200	☐	☐	4442	
5	hashhacking	200	☐	☐	4442	
6	hashbug	200	☐	☐	4442	
7	hashbee	200	☐	☐	4442	
9	hashuser	302	☐	☐	502	
8	hashbwapp	200	☐	☐	4442	
11	hashhack	200	☐	☐	4442	
10	hashhello	200	☐	☐	4442	

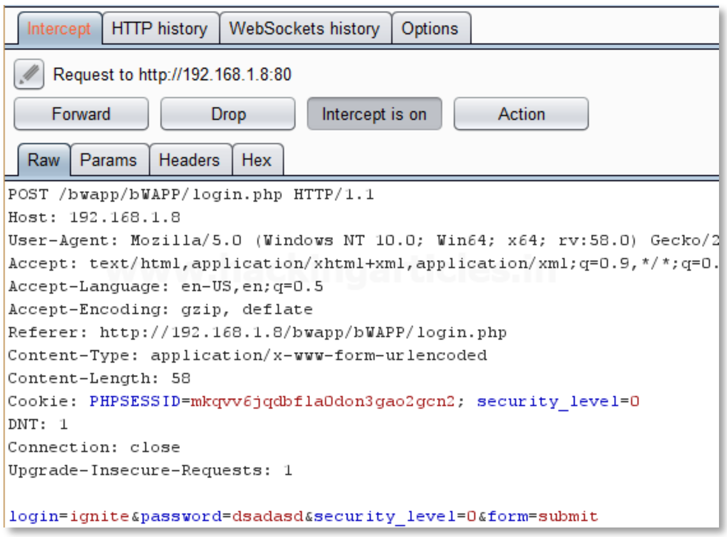
```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

And to confirm the **password matched**, we will give the password in the **Bwapp LAB login page**, which will successfully log us into the **Bwapp lab**. This shows our success in the attack as shown in the image.



Add Suffix

This processing rule adds up a suffix after the payload.
First, we have intercepted the request of the login page in the **Bwapp LAB**, where we have given default username and wrong password. Then click on login, the burp suite will capture the request of the login page in the intercept tab.

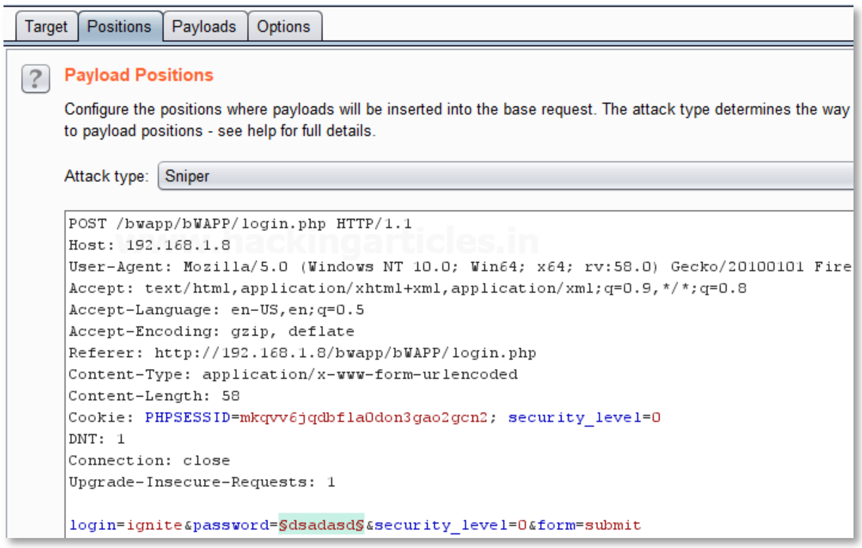


```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/2
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.8/bwapp/bWAPP/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 58
Cookie: PHPSESSID=mkqvv6jqdbfla0don3gao2gcn2; security_level=0
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

login=ignite&password=dsadasd&security_level=0&form=submit
```

Send the captured request to the **Intruder** by clicking on the Action Tab and follow given below step. Now open the **Intruder tab** then select **Positions tab** and you can observe the highlighted password and follow the given below step for selecting payload position.

- Press on the **Clear button** given at right of the window frame.
- Now we will select the fields where we want to attack and i.e. the password filed and click on **Add button.**
- Choose the **Attack type** as a **sniper**
- In the given below image, we have selected a password that means we will need one dictionary files for a password.



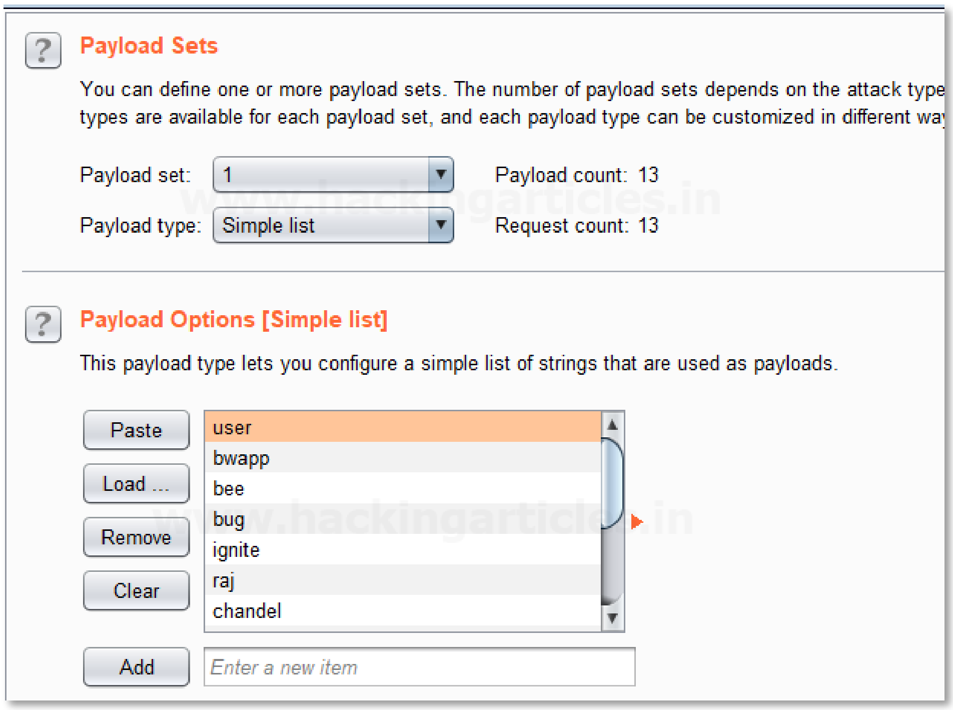
Configure the positions where payloads will be inserted into the base request. The attack type determines the way to payload positions - see help for full details.

Attack type: Sniper

```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Fire
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.8/bwapp/bWAPP/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 58
Cookie: PHPSESSID=mkqvv6jqdbfla0don3gao2gcn2; security_level=0
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

login=ignite&password=§dsadasd§&security_level=0&form=submit
```

Now click on **payloads option** after selecting payload position. Then select the **Payload type** as **Simple list,** where we have added a dictionary by clicking on **Load** button. We can either load the dictionary or we can manually add input strings using the **Add button** in the **payload options** as shown in the image.


Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type
types are available for each payload set, and each payload type can be customized in different way

Payload set: 1 Payload count: 13

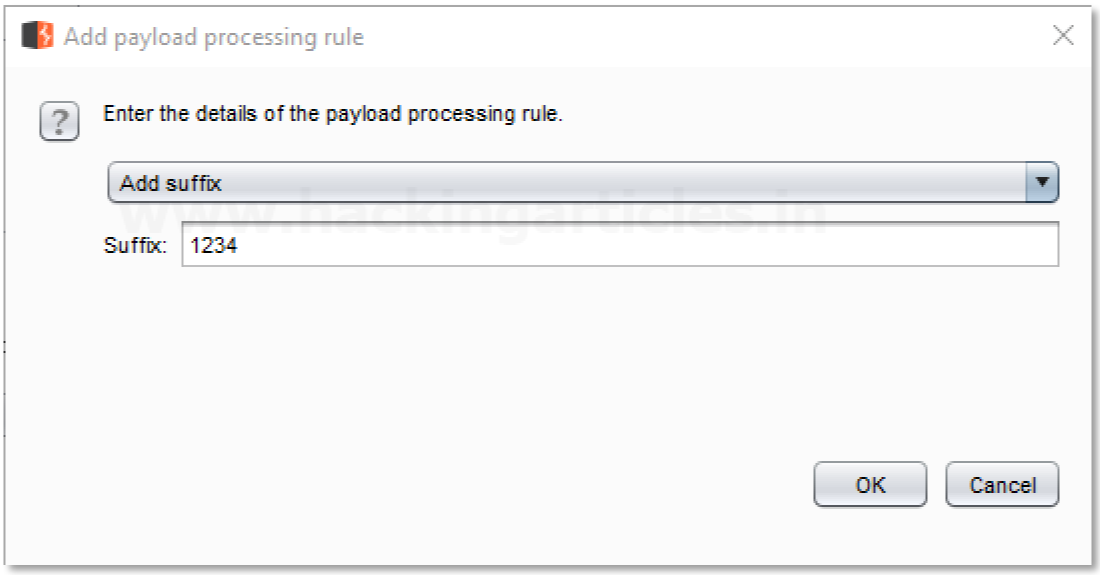
Payload type: Simple list Request count: 13

Payload Options [Simple list]

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

```
user
bwapp
bee
bug
ignite
raj
chandel
```

Before executing the attack we have added a **payload processing rule** to the payload type which is **Add Suffix** and we have given an input string “1234” which is added as a **suffix** with every input strings in the dictionary, as shown in the **result window** of the **attack**.
Select **Start Attack** in the **Intruder menu** as shown in the image.

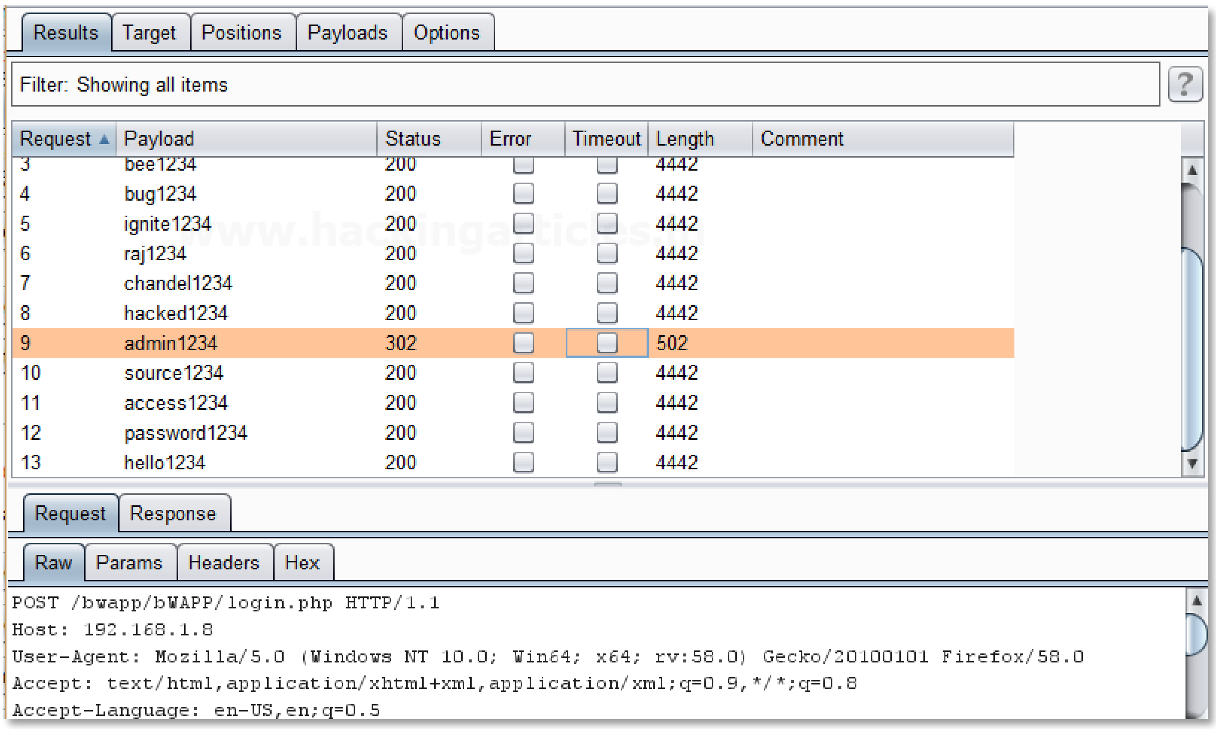

Add payload processing rule

Enter the details of the payload processing rule.

Add suffix

Suffix: 1234

Sit back and relax because now the burp suite will do its work, match the password which will give you the correct password. The moment it will find the correct value, it will change the value of length as shown in the image.
Use this combination of username and password for login to verify your brute force attack for the correct password.



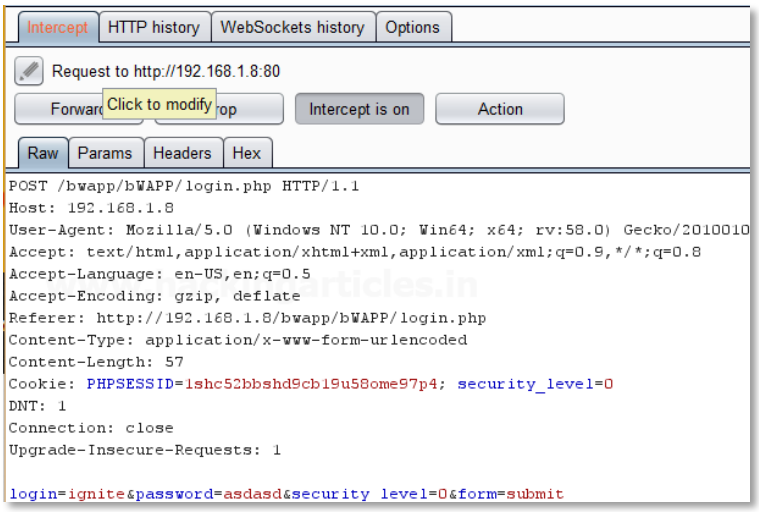
Request	Payload	Status	Error	Timeout	Length	Comment
3	bee1234	200	☐	☐	4442	
4	bug1234	200	☐	☐	4442	
5	ignite1234	200	☐	☐	4442	
6	raj1234	200	☐	☐	4442	
7	chandel1234	200	☐	☐	4442	
8	hacked1234	200	☐	☐	4442	
9	admin1234	302	☐	☐	502	
10	source1234	200	☐	☐	4442	
11	access1234	200	☐	☐	4442	
12	password1234	200	☐	☐	4442	
13	hello1234	200	☐	☐	4442	

```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
```

Match / Replace

This processing rule is used to replace any part of the payload that match a specific regular expression, with a string.

First, we have intercepted the request of the login page in the **Bwapp LAB**, where we have given default username and wrong password. Then click on login, the burp suite will capture the request of the login page in the intercept tab.

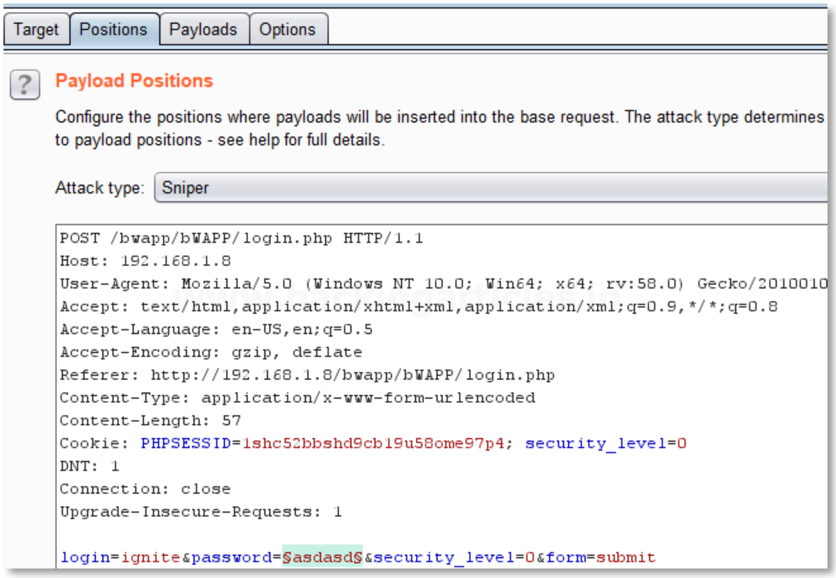


```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/2010010
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.8/bwapp/bWAPP/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 57
Cookie: PHPSESSID=1shc52bbshd9cb19u58ome97p4; security_level=0
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

login=ignite&password=asdasd&security_level=0&form=submit
```

Send the captured request to the **Intruder** by clicking on the Action Tab and follow given below step. Now open the **Intruder tab** then select **Positions tab** and you can observe the highlighted password and follow the given below step for selecting payload position.

- Press on the **Clear button** given at right of the window frame.
- Now we will select the fields where we want to attack and i.e. the password filed and click on **Add button.**
- Choose the **Attack type** as **the sniper**
- In the given below image, we have selected a password that means we will need one dictionary files for the password.



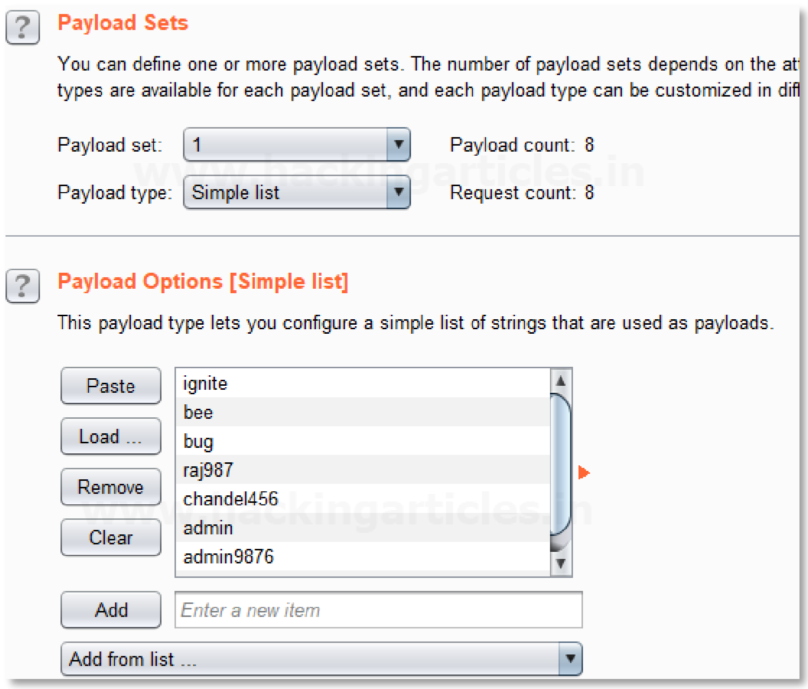
Configure the positions where payloads will be inserted into the base request. The attack type determines to payload positions - see help for full details.

Attack type: Sniper

```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/2010010
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.8/bwapp/bWAPP/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 57
Cookie: PHPSESSID=1shc52bbshd9cb19u58ome97p4; security_level=0
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

login=ignite&password=§asdasd§&security_level=0&form=submit
```

Now click on **payloads option** after selecting payload position. Then select the **Payload type** as **Simple list,** where we have added a dictionary by clicking on **Load** button. We can either load the dictionary or we can manually add input strings using the **Add button** in the **payload options** as shown in the image.


Payload Sets

You can define one or more payload sets. The number of payload sets depends on the at
types are available for each payload set, and each payload type can be customized in dif

Payload set: 1 Payload count: 8

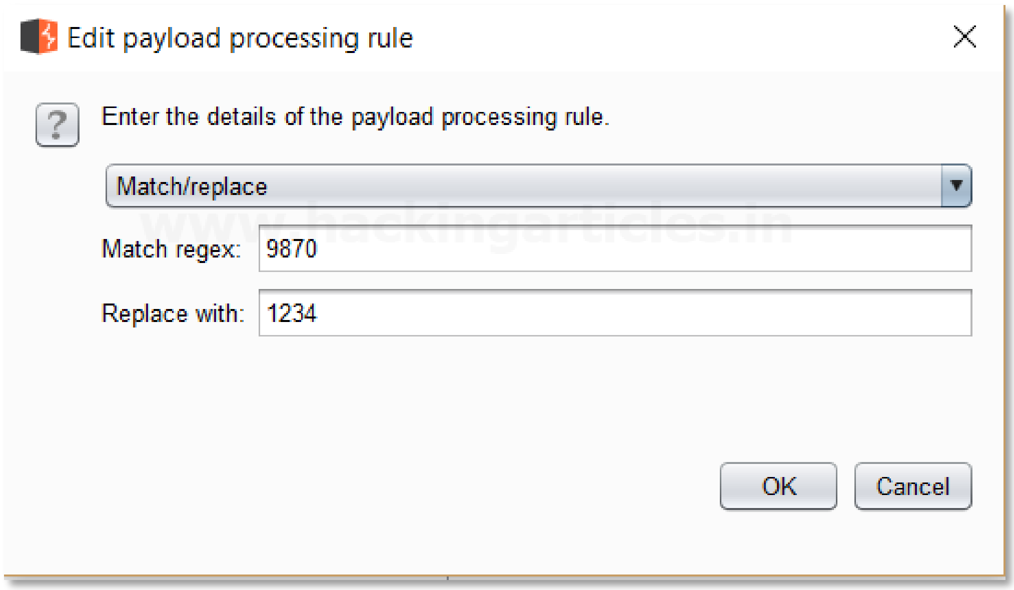
Payload type: Simple list Request count: 8

Payload Options [Simple list]

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

```
ignite
bee
bug
raj987
chandel456
admin
admin9876
```

Before executing the attack we have added a **payload processing rule** to the payload type which is **Match / Replace** and we have given an input "9870" in the **Match Regex** which will match the input given with the input strings in the dictionary, if the there is a certain match than it will replace it with the input "1234" given in the **Replace with** as shown in the image.
Select **Start Attack** in the **Intruder menu**.


Edit payload processing rule

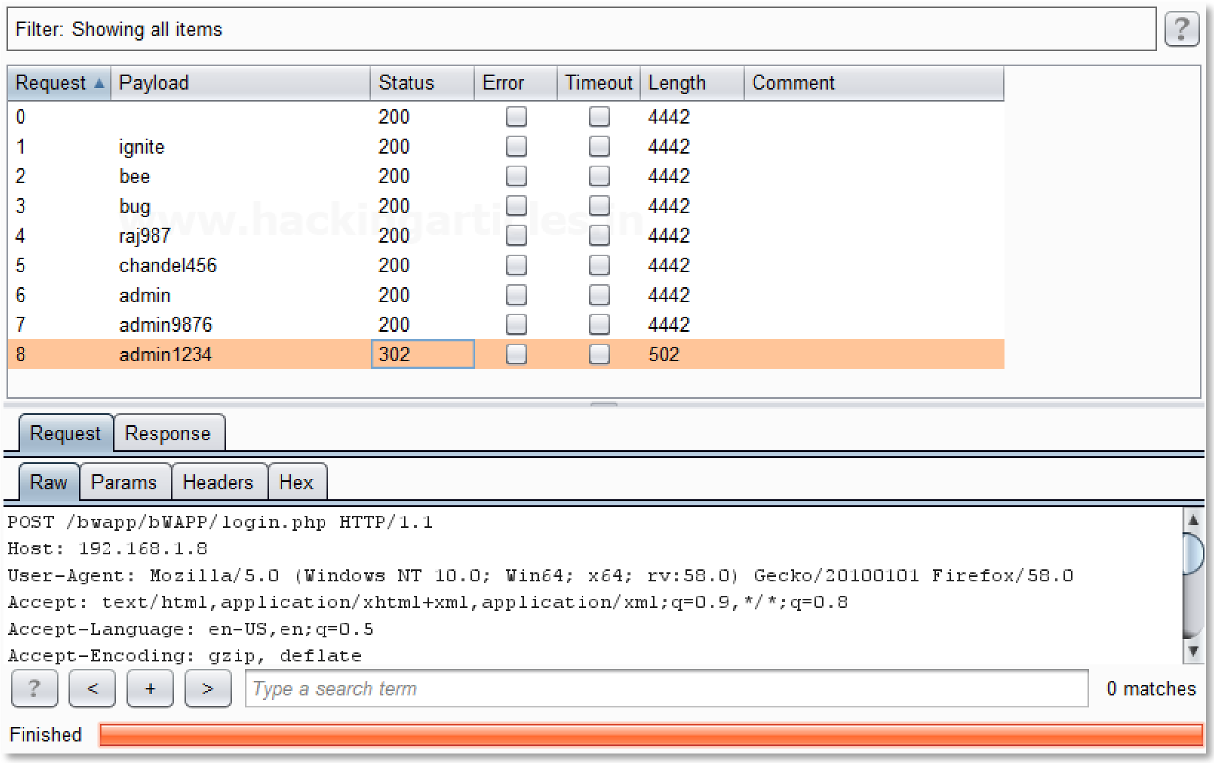
Enter the details of the payload processing rule.

Match/replace

Match regex: 9870

Replace with: 1234

Sit back and relax because now the burp suite will do its work, match the password which will give you the correct password. The moment it will find the correct value, it will change the value of length as shown in the image.
Use this combination of username and password for login to verify your brute force attack for the correct password.

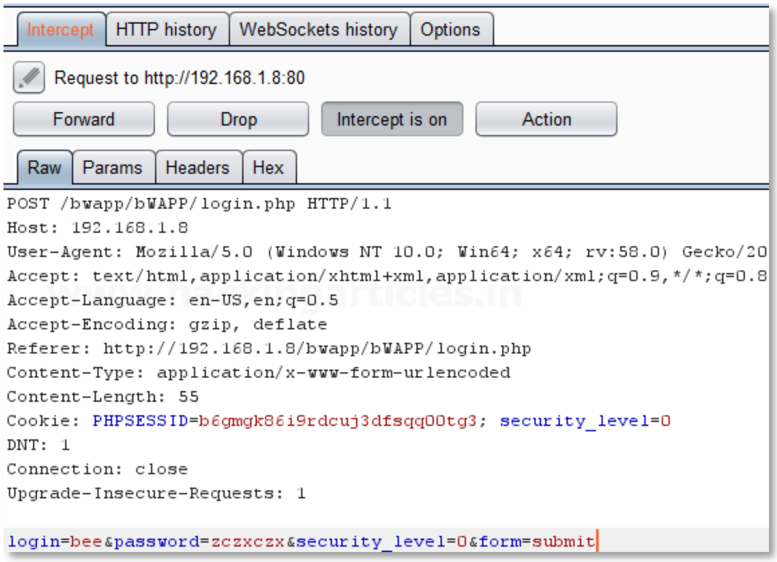


Request	Payload	Status	Error	Timeout	Length	Comment
0		200	☐	☐	4442	
1	ignite	200	☐	☐	4442	
2	bee	200	☐	☐	4442	
3	bug	200	☐	☐	4442	
4	raj987	200	☐	☐	4442	
5	chandel456	200	☐	☐	4442	
6	admin	200	☐	☐	4442	
7	admin9876	200	☐	☐	4442	
8	admin1234	302	☐	☐	502	

```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
```

Substring

This processing rule is used to extracts a sub-portion of the payload, starting from a specified offset up to a specified length. Here the offset and length are counted from the front.
First, we have intercepted the request of the login page in the **Bwapp LAB**, where we have given default username and wrong password. Then click on login, the burp suite will capture the request of the login page in the intercept tab.

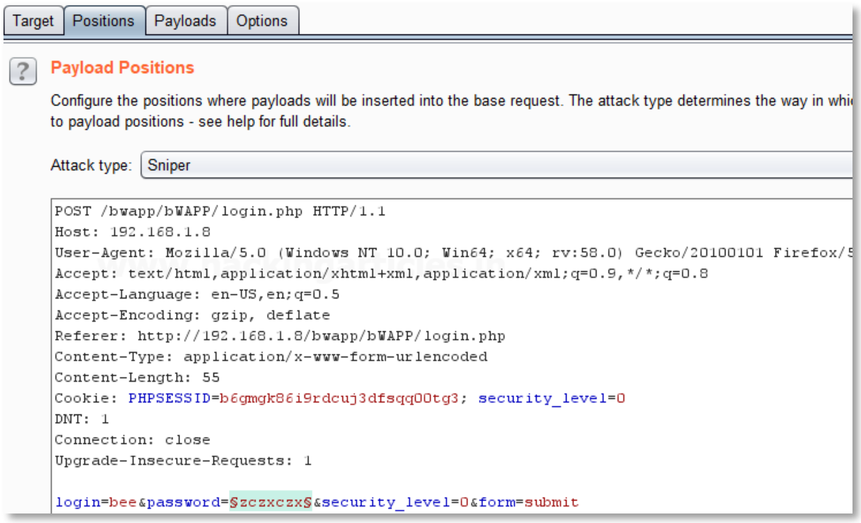


```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.8/bwapp/bWAPP/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 55
Cookie: PHPSESSID=b6gmgk86i9rdcuj3dfsqq00tg3; security_level=0
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

login=bee&password=zczxczx&security_level=0&form=submit
```

Send the captured request to the **Intruder** by clicking on the Action Tab and follow given below step. Now open the **Intruder tab** then select **Positions tab** and you can observe the highlighted password and follow the given below step for selecting payload position.

- Press on the **Clear button** given at right of the window frame.
- Now we will select the fields where we want to attack and i.e. the password filed and click on **Add button.**
- Choose the **Attack type** as **a sniper**
- In the given below image, we have selected a password that means we will need one dictionary files for the password.



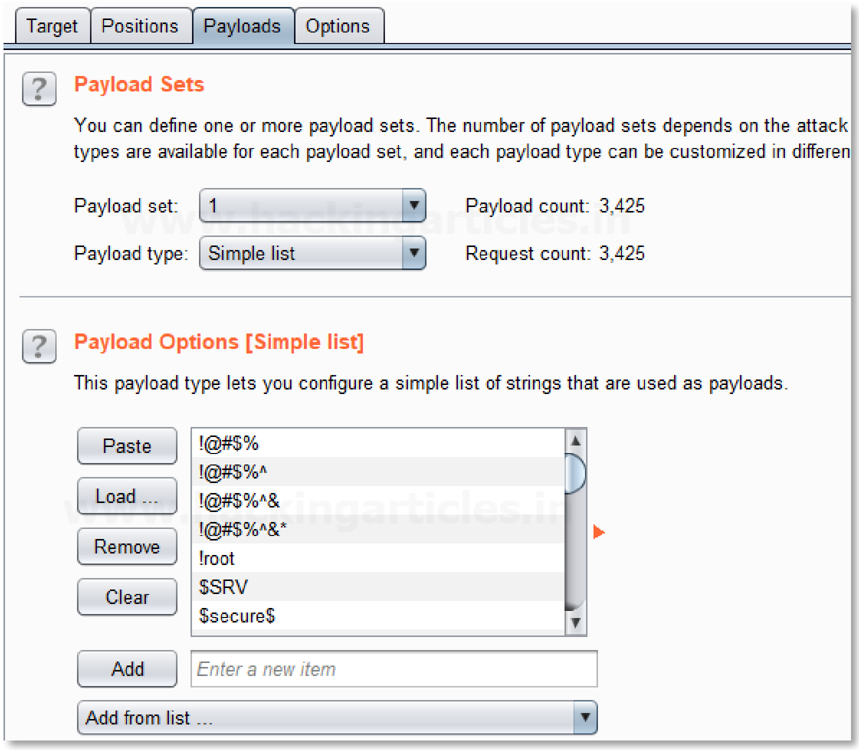
Configure the positions where payloads will be inserted into the base request. The attack type determines the way in whi
to payload positions - see help for full details.

Attack type: Sniper

```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/5
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.8/bwapp/bWAPP/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 55
Cookie: PHPSESSID=b6gmgk86i9rdcuj3dfsqq00tg3; security_level=0
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

login=bee&password=§zczxczx§&security_level=0&form=submit
```

Now click on **payloads option** after selecting payload position. Then select the **Payload type** as **Simple list,** where we have added a dictionary by clicking on **Load** button. Here we had added dictionary using the option "**Add from list**" as shown below in the given image.



Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack types are available for each payload set, and each payload type can be customized in differen

Payload set: 1 Payload count: 3,425

Payload type: Simple list Request count: 3,425

Payload Options [Simple list]

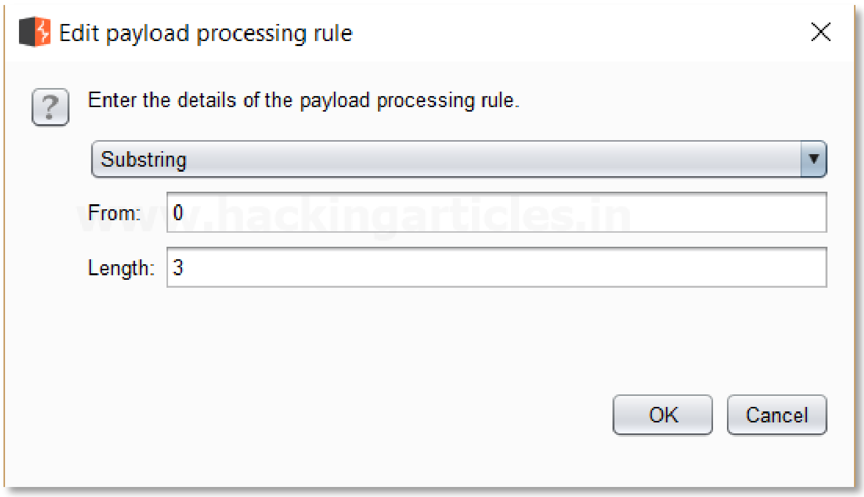
This payload type lets you configure a simple list of strings that are used as payloads.

```
!@#$%
!@#$%^
!@#$%^&
!@#$%^&*
!root
$SRV
$secure$
```

Enter a new item

Add from list ...

Before executing the attack we have added a **payload processing rule** to the payload type which is**Substring** and we have given an input "0" in **From option** which specifies the offset and an input "3" in the**Length option** which specifies the length of the input strings.
For example, if "password" is a word in a dictionary and we had applied above filter so it will place alphabet **p = 0; a = 1; s = 2 and s = 3** hence it will read only **pass** from whole word "password".
The length specified will select only those inputs having the specific length and other lower or greater length inputs are discarded as shown in the result window of the attack.
Select **Start Attack** in the **Intruder menu**.



Edit payload processing rule

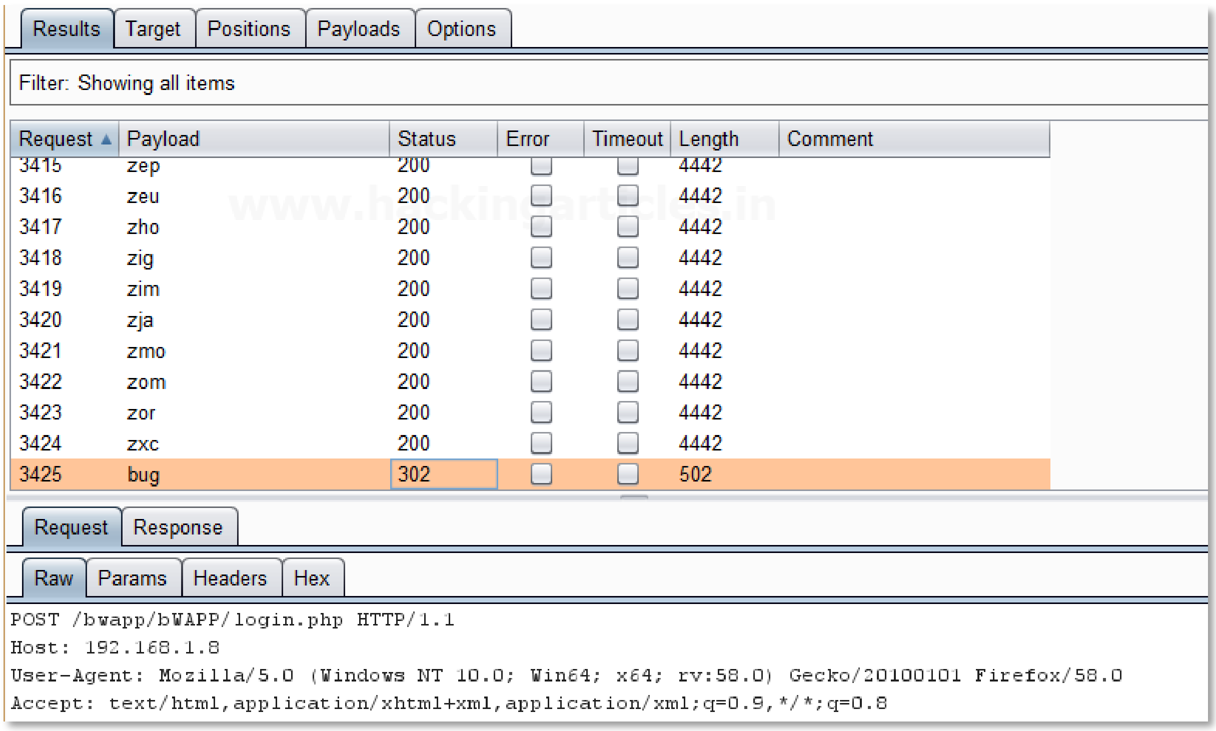
Enter the details of the payload processing rule.

Substring

From: 0

Length: 3

Sit back and relax because now the burp suite will do its work, match the password which will give you the correct password. The moment it will find the correct value, it will change the value of length as shown in the image.
Use this combination of username and password for login to verify your brute force attack for the correct password.



Results | Target | Positions | Payloads | Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
3415	zep	200	☐	☐	4442	
3416	zeu	200	☐	☐	4442	
3417	zho	200	☐	☐	4442	
3418	zig	200	☐	☐	4442	
3419	zim	200	☐	☐	4442	
3420	zja	200	☐	☐	4442	
3421	zmo	200	☐	☐	4442	
3422	zom	200	☐	☐	4442	
3423	zor	200	☐	☐	4442	
3424	zxc	200	☐	☐	4442	
3425	bug	302	☐	☐	502	

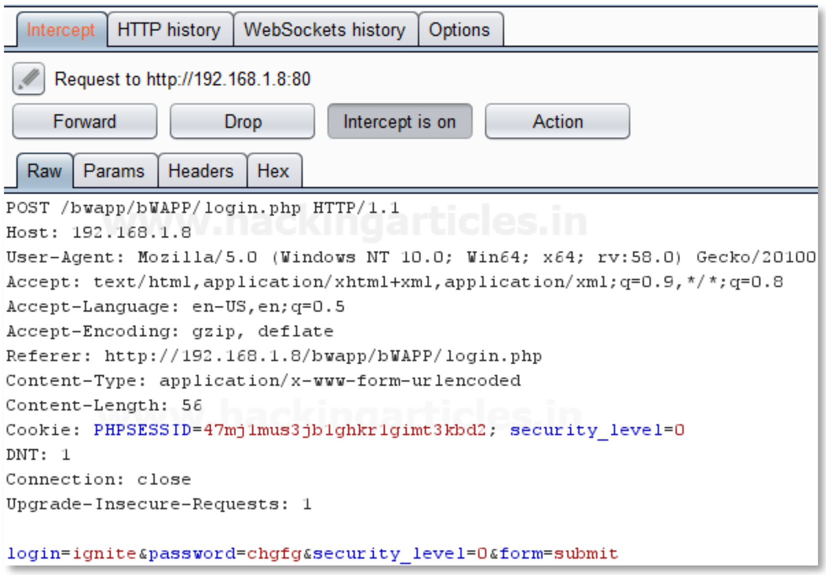
Request | Response

Raw | Params | Headers | Hex

```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

Reverse Substring

This processing rule is used as a substring rule, but the end offset is specified counting backward from the end of the payload, and the length is counted backward from the end offset.
First, we have intercepted the request of the login page in the **Bwapp LAB**, where we have given default username and wrong password. Then click on login, the burp suite will capture the request of the login page in the intercept tab.

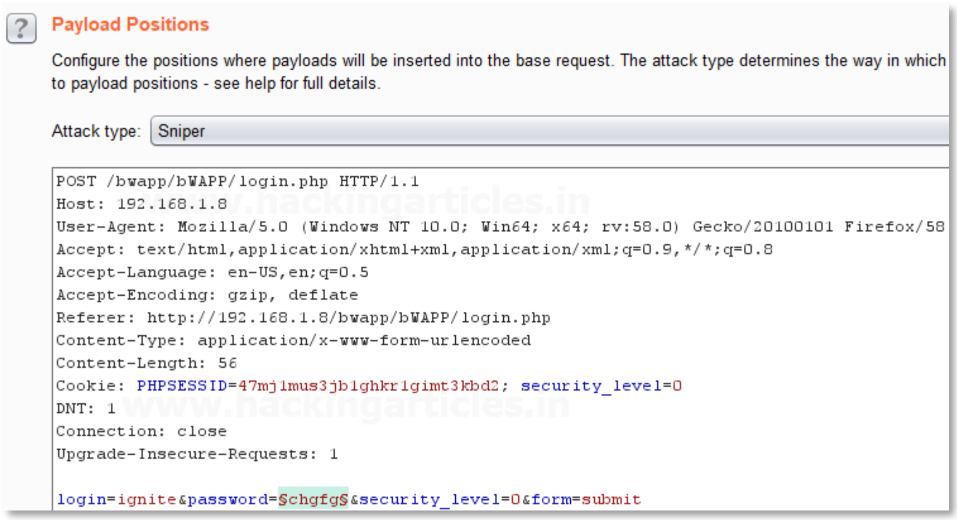


```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.8/bwapp/bWAPP/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 56
Cookie: PHPSESSID=47mj1mus3jb1ghkr1gimt3kbd2; security_level=0
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

login=ignite&password=chgfg&security_level=0&form=submit
```

Send the captured request to the **Intruder** by clicking on the Action Tab and follow given below step. Now open the **Intruder tab** then select **Positions tab** and you can observe the highlighted password and follow the given below step for selecting payload position.

- Press on the **Clear button** given at right of the window frame.
- Now we will select the fields where we want to attack and i.e. the password filed and click on **Add button.**
- Choose the **Attack type** as the **sniper**
- In the given below image, we have selected a password that means we will need one dictionary files for a password.



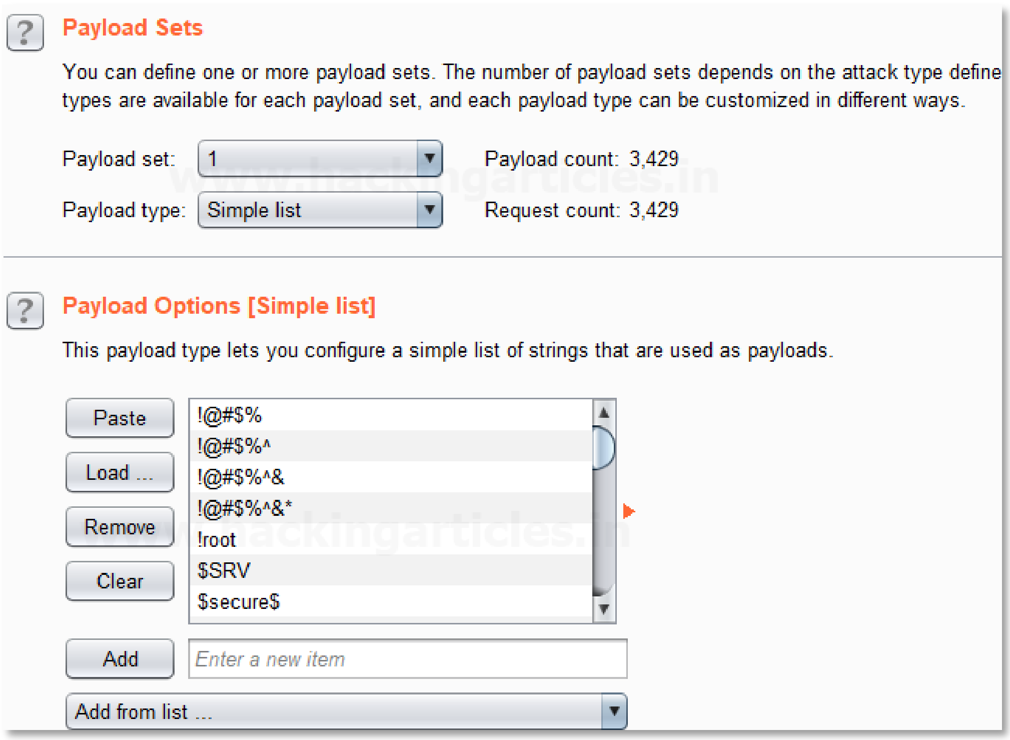
Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which to payload positions - see help for full details.

Attack type: Sniper

```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.8/bwapp/bWAPP/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 56
Cookie: PHPSESSID=47mj1mus3jb1ghkr1gimt3kbd2; security_level=0
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

login=ignite&password=§chgfg§&security_level=0&form=submit
```

Now click on **payloads option** after selecting payload position. Then select the **Payload type** as **Simple list,** where we have added a dictionary by clicking on **Load** button. Here we had added dictionary using the option "**Add from list**" as shown below in the given image.


Payload Sets

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

Payload set: 1 Payload count: 3,429

Payload type: Simple list Request count: 3,429

Payload Options [Simple list]

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

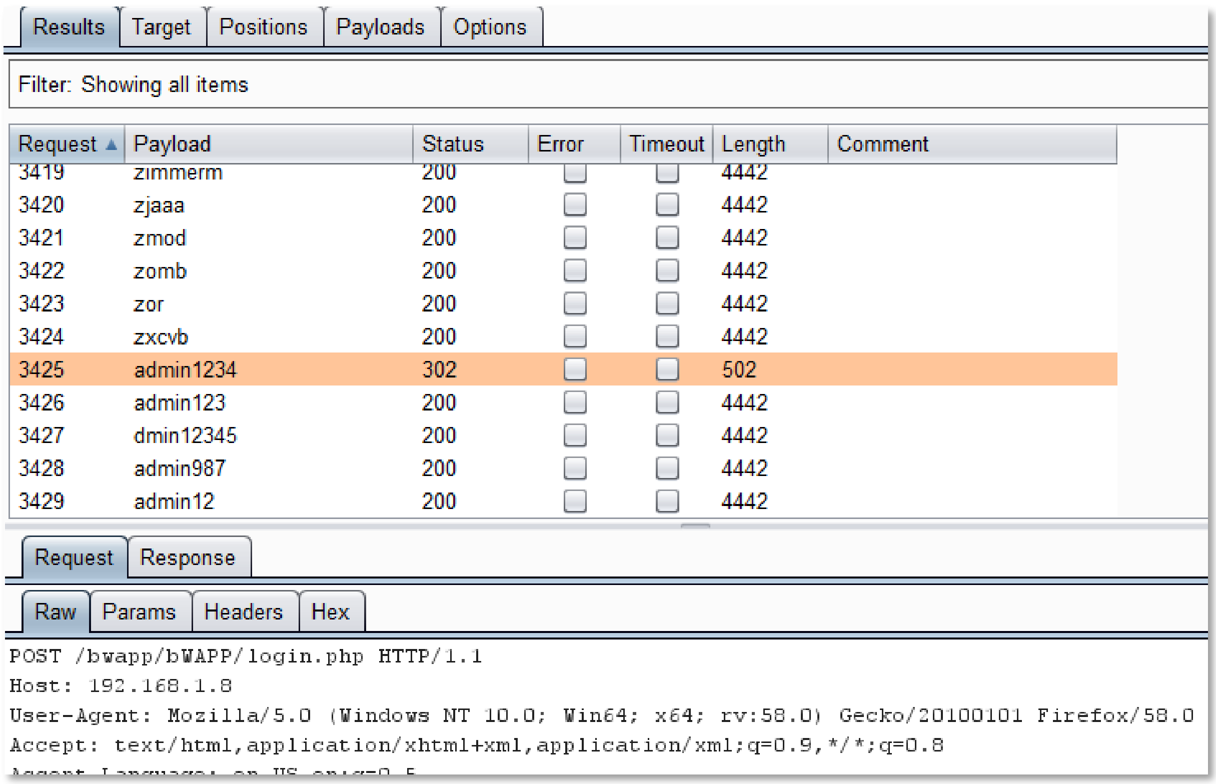
```
!@#$%
!@#$%^
!@#$%^&
!@#$%^&*
!root
$SRV
$secure$
```

Before executing the attack we have added a **payload processing rule** to the payload type which is **Reverse Substring** and we have given an input "2" in **From option** which specifies the offset and an input "9" in the **Length option** which specifies the length of the input strings and they are similar to the Substring rule but it works from backward of an offset and the length is counted backward where the offset ends.

For example if "admin123456" is word in dictionary and we had applied above filter so it will place alphabet **4 = 0; 3 = 1 ; 2 = 2 ; 1 = 3 ; n = 4 ; i = 5 ; m = 6 ; d = 7 ; d = 8 ; a = 9** hence it will read only '**admin1234'** from whole word "admin123456".

The length specified will select only those inputs having the specific length and other lower or greater length inputs are discarded as shown in the result window of the attack.

Select **Start Attack** in the **Intruder menu**.
Sit back and relax because now the burp suite will do its work, match the password which will give you the correct password. The moment it will find the correct value, it will change the value of length as shown in the image.
Use this combination of username and password for login to verify your brute force attack for the correct password.



Request ▲	Payload	Status	Error	Timeout	Length	Comment
3419	zimmerm	200	☐	☐	4442	
3420	zjaaa	200	☐	☐	4442	
3421	zmod	200	☐	☐	4442	
3422	zomb	200	☐	☐	4442	
3423	zor	200	☐	☐	4442	
3424	zxcvb	200	☐	☐	4442	
3425	admin1234	302	☐	☐	502	
3426	admin123	200	☐	☐	4442	
3427	dmin12345	200	☐	☐	4442	
3428	admin987	200	☐	☐	4442	
3429	admin12	200	☐	☐	4442	

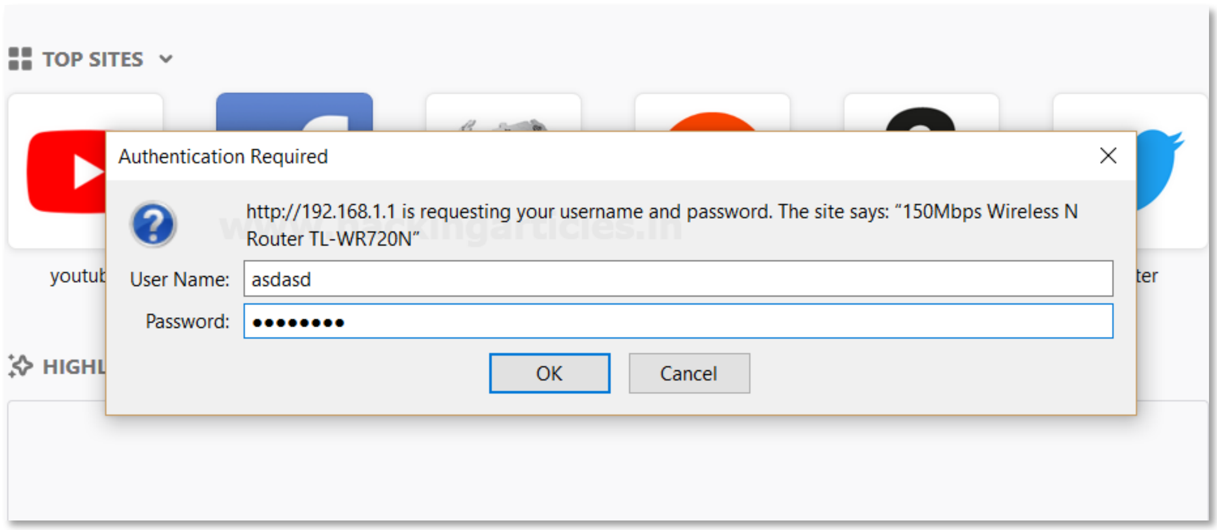
```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

Modify Case

This processing rule can be used to modify the case of the payload if needed. This rule has the same options available for the **Case Modification** payload type.

Payload Encode

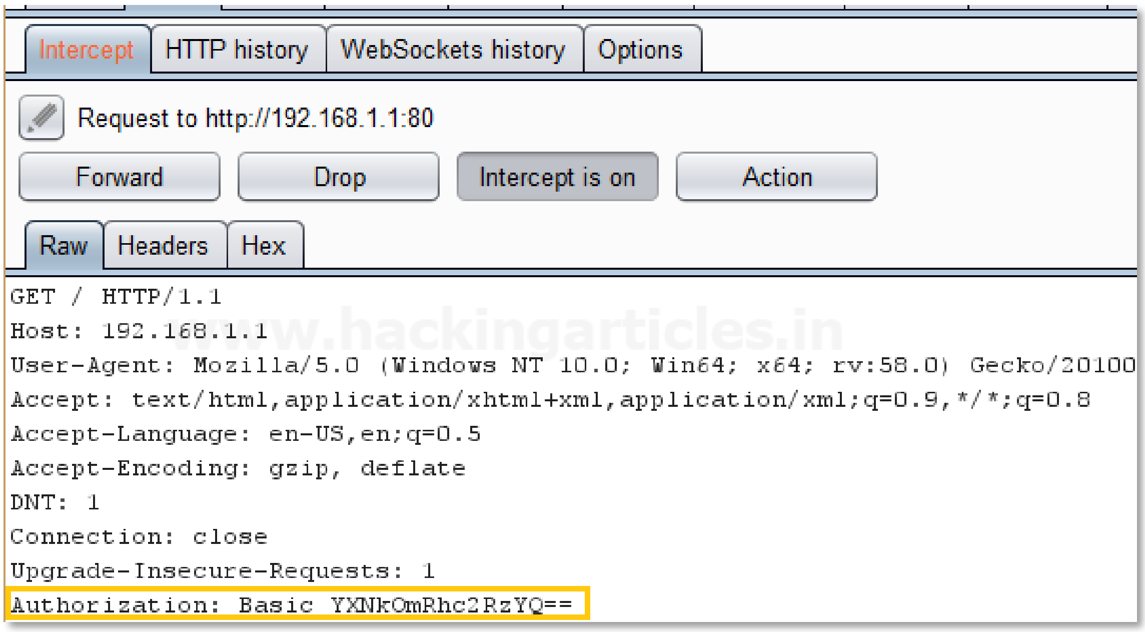
The processing rule can be used to encode the payload using various schemes such as URL, HTML, Base64, ASCII hex or constructed strings.
Let's start!!
First, we have intercepted the request of the login page of the router by giving its default IP which is **192.168.1.1**, where we have given an invalid username and password. Then click on login, the burp suite will capture the request of the login page in the intercept tab.


Authentication Required

http://192.168.1.1 is requesting your username and password. The site says: "150Mbps Wireless N Router TL-WR720N"

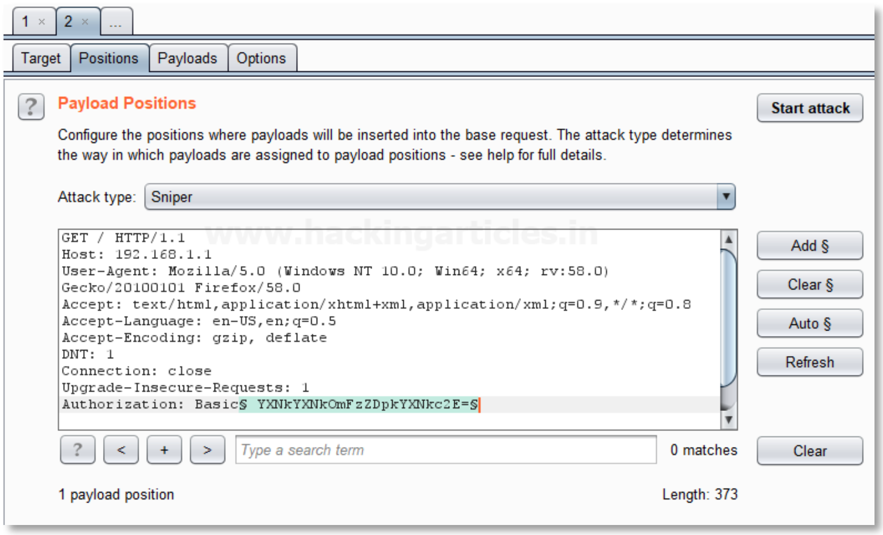
User Name: asdasd
Password: ●●●●●●●●●

Thus the sent request will be captured by burp suite which you can see in the given below image. In the screenshot, I had highlighted some value in the last line. Here it tells the type of authentication provided by the router is **basic** and if you have read above theory of basic authentication I had described that it is **encoded** in **base 64**



```
GET / HTTP/1.1
Host: 192.168.1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
Authorization: Basic YXNkOmRhc2RzYQ==
```

Send the captured request to the **Intruder** by clicking on the Action Tab and follow given below step. Now open the **Intruder tab** then select **Positions tab** and you can observe the highlighted password and follow the given below step for selecting payload position.
Press on the **Clear button** given at right of the window frame.
Now **select** the **encoded value of authentication** for payload position and **click** to **ADD button** on the left side of the frame.
Choose the **Attack type** as



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.

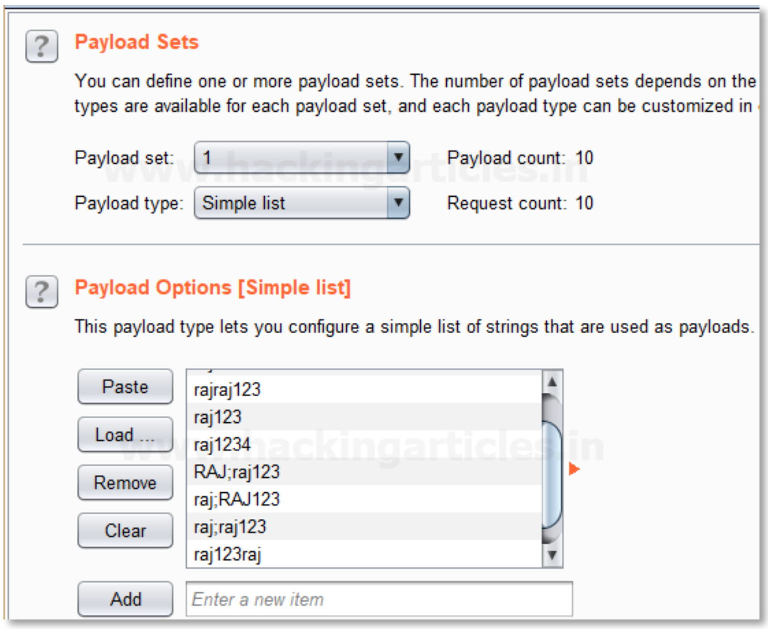
Attack type: Sniper

```
GET / HTTP/1.1
Host: 192.168.1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0)
Gecko/20100101 Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
Authorization: Basic§ YXNkYXNkOmFzZDpkYXNkc2E=§
```

Type a search term 0 matches

1 payload position Length: 373

Now click on **payloads option** after selecting payload position. Then select the **Payload type** as **Simple list,** where we have added a dictionary by clicking on **Load** button. We can either load the dictionary or we can manually add input strings using the **Add button** in the **payload options** as shown in the image.
The base64 encoded value of Authentication is a combination of username and password now the scenario is to generate the same encoded value of authentication with help of user password dictionary, therefore I have made a **dictionary.**



You can define one or more payload sets. The number of payload sets depends on the types are available for each payload set, and each payload type can be customized in

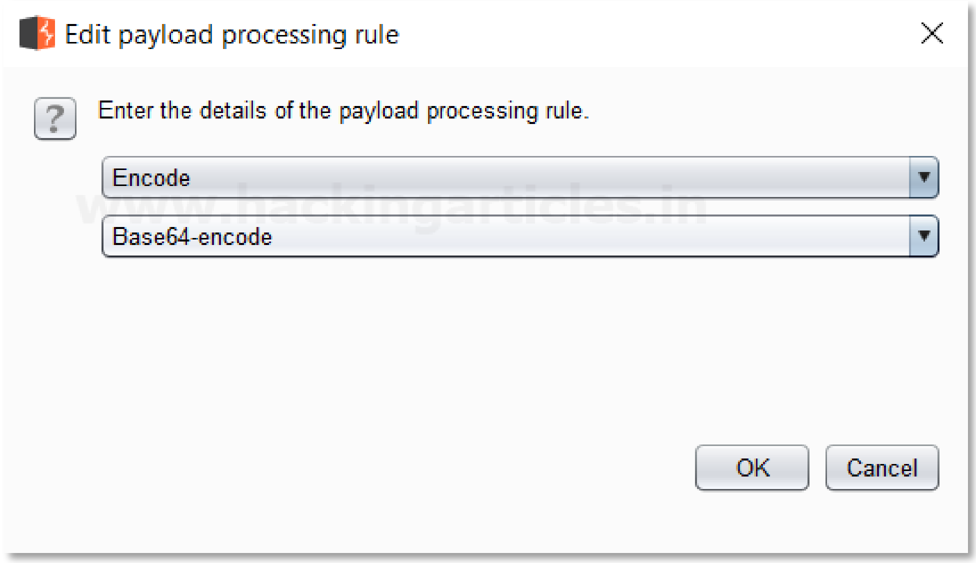
Payload set: 1 Payload count: 10
Payload type: Simple list Request count: 10

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

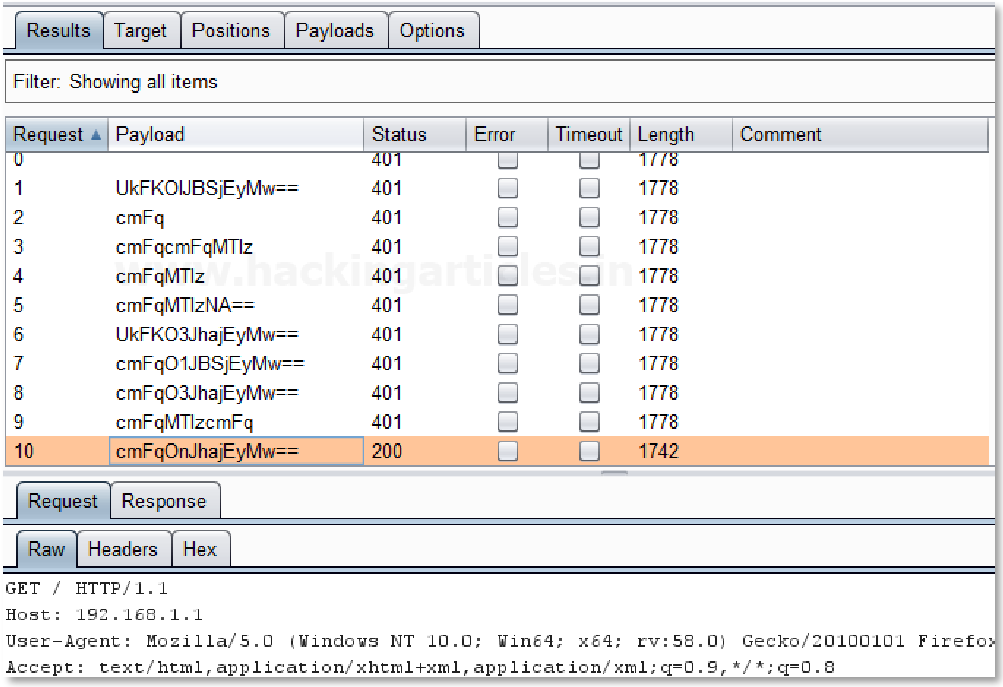
rajraj123
raj123
raj1234
RAJ;raj123
raj;RAJ123
raj;raj123
raj123raj

Enter a new item

Before executing the attack we have added a **payload processing rule** to the payload type which is **Encode**and we have selected **“Base64 encode”** scheme because we know router takes the value in **Base64**.
Select **Start Attack** in the **Intruder menu** as shown in the image.


Edit payload processing rule
Enter the details of the payload processing rule.
Encode
Base64-encode
OK
Cancel

Sit back and relax because this will start brute force attack and try to match string for user authentication. In the screenshot, you can the **status** and **length** of the **highlighted value** is **different** from the rest of the values. This means we can use this encoded value to bypass the user authentication which occurs from request number 10. Now check the username and password of 10th line in the dictionary.
And to confirm the **username** and **password matched**, we will give the password in the **Router’s Login Page**, which will successfully log us into the **Router’s Configuration Page**. This shows our success in the attack as shown in the image.



Request ▲	Payload	Status	Error	Timeout	Length	Comment
0		401	☐	☐	1778	
1	UkFKOlJBSjEyMw==	401	☐	☐	1778	
2	cmFq	401	☐	☐	1778	
3	cmFqcmFqMTIz	401	☐	☐	1778	
4	cmFqMTIz	401	☐	☐	1778	
5	cmFqMTIzNA==	401	☐	☐	1778	
6	UkFKO3JhajEyMw==	401	☐	☐	1778	
7	cmFqO1JBSjEyMw==	401	☐	☐	1778	
8	cmFqO3JhajEyMw==	401	☐	☐	1778	
9	cmFqMTIzcmFq	401	☐	☐	1778	
10	cmFqOnJhajEyMw==	200	☐	☐	1742	

```
GET / HTTP/1.1
Host: 192.168.1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

Decode

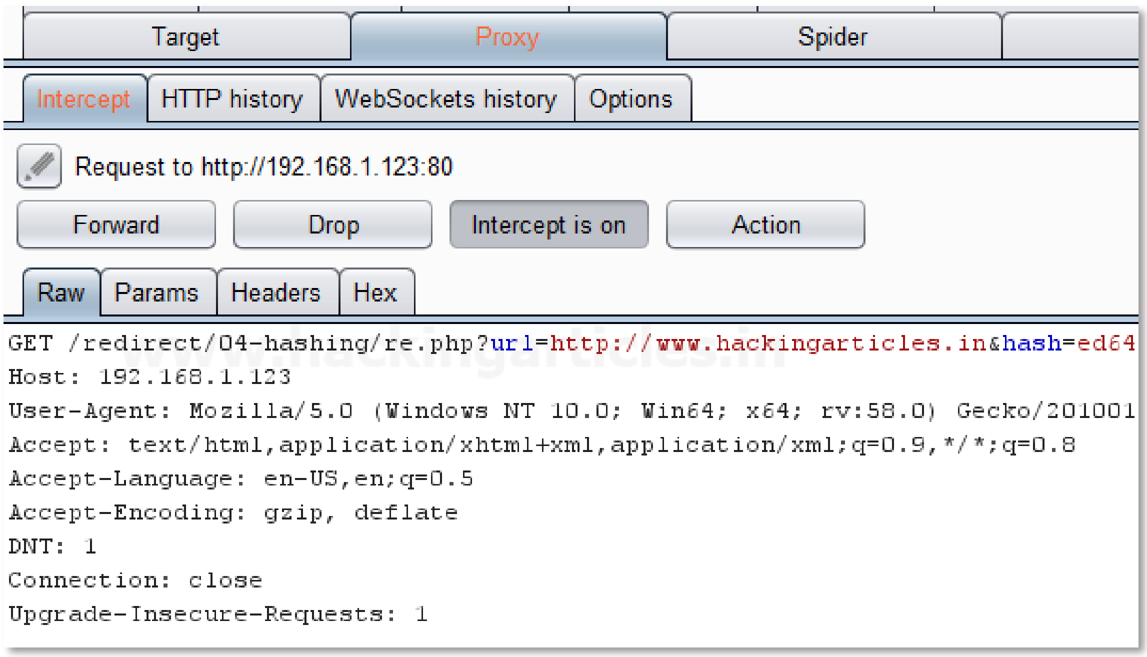
This processing rule can be used to decode the payload using various schemes: URL, HTML, Base64 or ASCII hex. As we know decoding is nothing but reversing the encoding. It can be used in the opposite way in which encoding is carried out.

Hash

This processing rule can be used to carry out a hashing operation on the payload. There are 7 types of hashing algorithms are available in this payload processing rule which is as follows:

- SHA-384
- SHA-224
- SHA-256
- MD5
- MD2
- SHA
- SHA-512

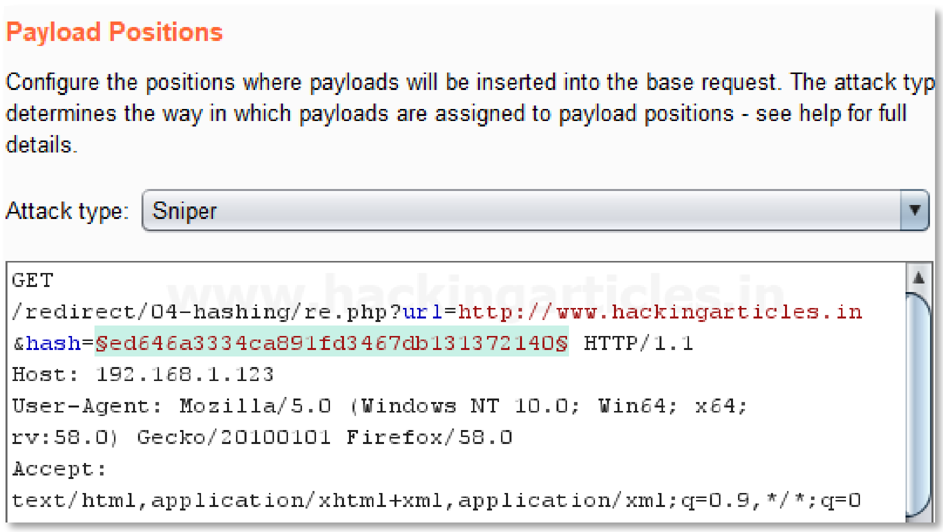
First, we have intercepted the request of the **Redirection Link** designed to find **redirection vulnerabilities** in the **LAB** created by us and in the hash value of the URL we have given a wrong hash value of **HTTP://www.google.com** in place of the **actual hash value** of the **HTTP://www.hackingarticles.in** in the **URL** of the redirecting page. We have simply clicked on the Redirection link as shown in the image; the burp suite will capture the request of the redirecting page in the intercept tab.



```
GET /redirect/04-hashing/re.php?url=http://www.hackingarticles.in&hash=ed64
Host: 192.168.1.123
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/201001
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
```

Send the captured request to the **Intruder** by clicking on the Action Tab and follow given below step. Now open the **Intruder tab** then select **Positions tab** and you can observe the highlighted password and follow the given below step for selecting payload position.

- Press on the **Clear button** given at right of the window frame.
- Now we will select the fields where we want to attack which is the **hash value** of the redirecting page and then click on **the Add button.**
- Choose the **Attack type** as a sniper.



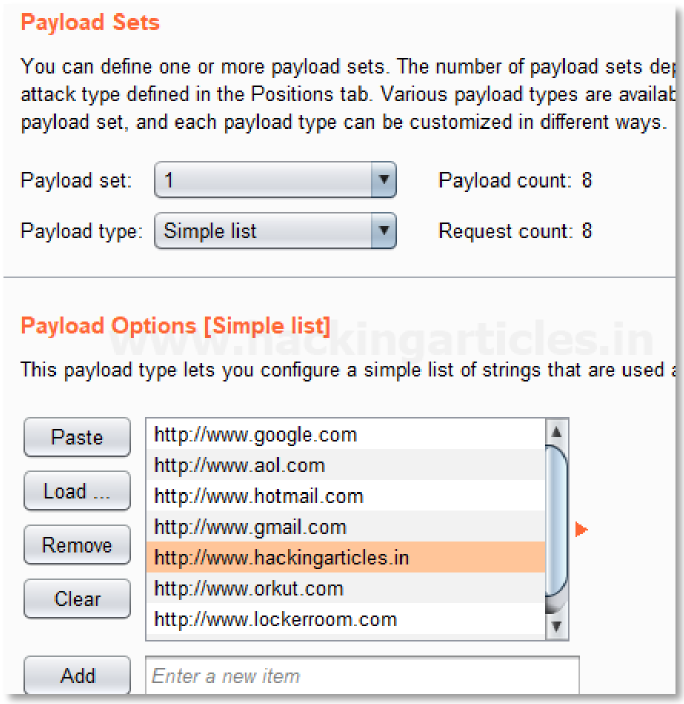
Payload Positions

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

Attack type: Sniper

```
GET
/redirect/04-hashing/re.php?url=http://www.hackingarticles.in
&hash=§ed646a3334ca891fd3467db131372140§ HTTP/1.1
Host: 192.168.1.123
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:58.0) Gecko/20100101 Firefox/58.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0
```

Then select the **Payload type** as **Simple list,** where we have added a dictionary by clicking on **Load** button. We can either load the dictionary or we can manually add input strings using the **Add button** in the**payload options** as shown in the image.



Payload Sets

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

Payload set: 1 Payload count: 8

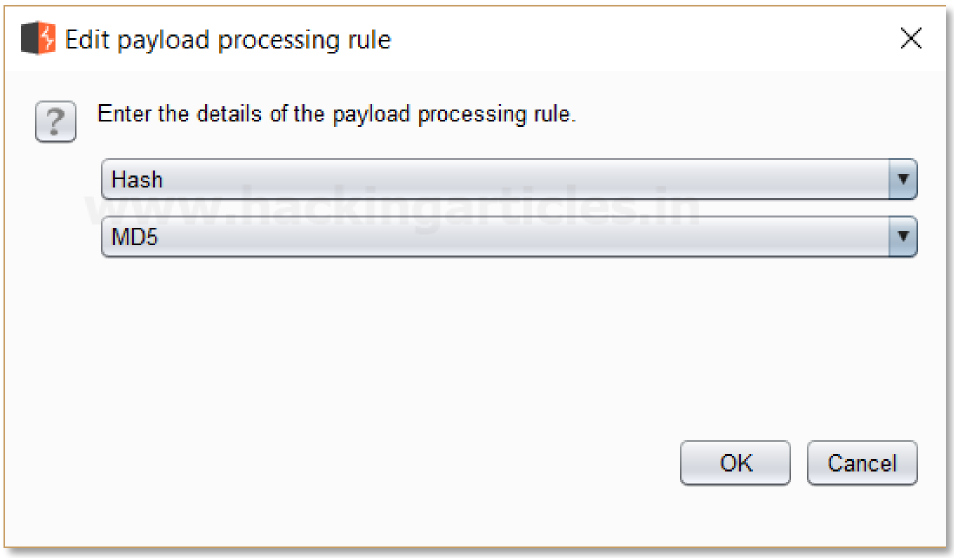
Payload type: Simple list Request count: 8

Payload Options [Simple list]

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

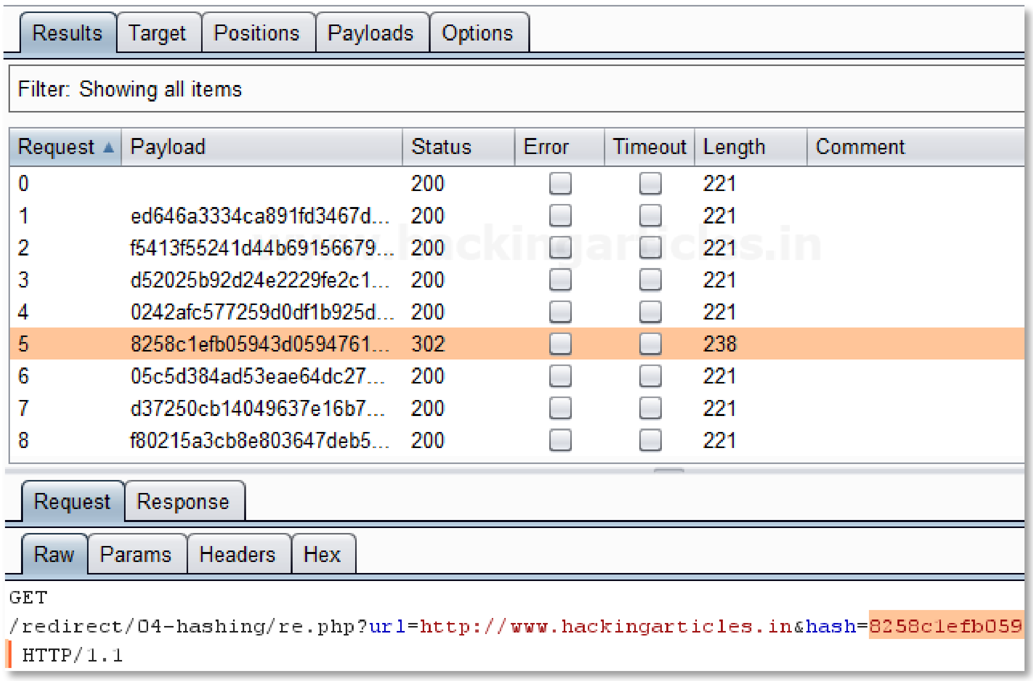
http://www.google.com
http://www.aol.com
http://www.hotmail.com
http://www.gmail.com
http://www.hackingarticles.in
http://www.orkut.com
http://www.lockerroom.com

Before executing the attack we have added a **payload processing rule** to the payload type which is **Hash**and then we have selected **MD5** which is a commonly used algorithm for converting URL of the websites into a Hash MD5 value. As you can see the **input strings** of the dictionary are in a simple text form, but this processing rule converts it into Hash MD5 values which can be seen in **result window** of the **attack**.
Select **Start Attack** in the **Intruder menu** as shown in the image.


Edit payload processing rule
Enter the details of the payload processing rule.
OK
Cancel

Hash
MD5

Sit back and relax because now the burp suite will do its work, match the Hash MD5 of the Redirecting Page which will give you the correct MD5 value. The moment it will find the correct value, it will change the value of length as shown in the image.

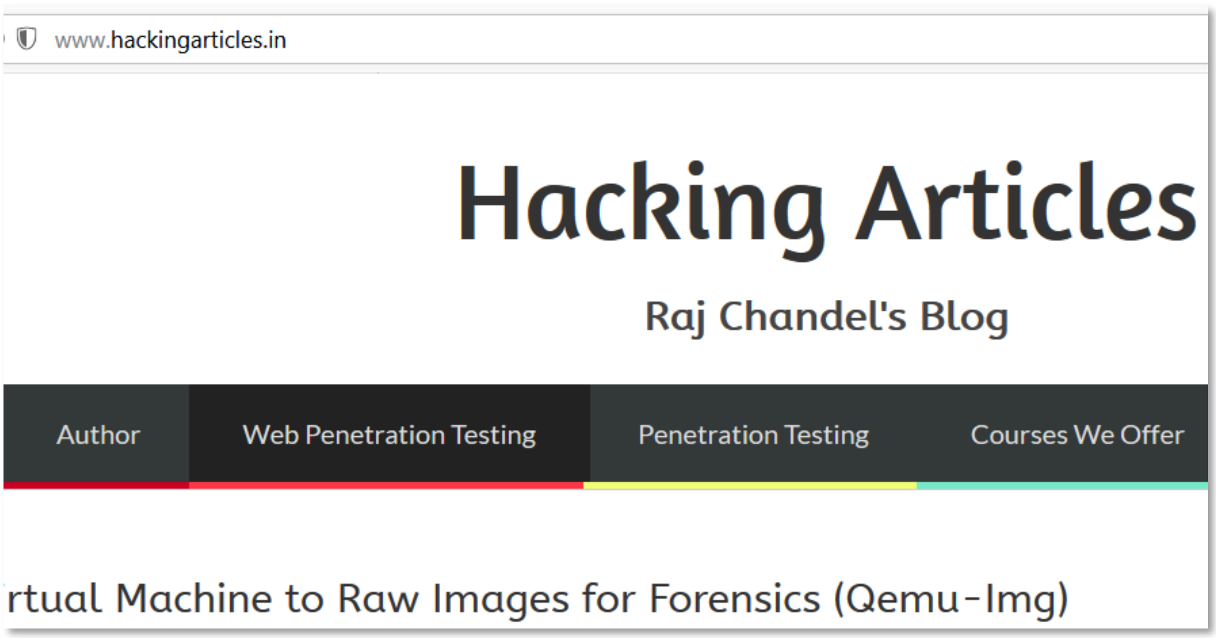


Filter: Showing all items

Request ▲	Payload	Status	Error	Timeout	Length	Comment
0		200	☐	☐	221	
1	ed646a3334ca891fd3467d...	200	☐	☐	221	
2	f5413f55241d44b69156679...	200	☐	☐	221	
3	d52025b92d24e2229fe2c1...	200	☐	☐	221	
4	0242afc577259d0df1b925d...	200	☐	☐	221	
5	8258c1efb05943d0594761...	302	☐	☐	238	
6	05c5d384ad53eae64dc27...	200	☐	☐	221	
7	d37250cb14049637e16b7...	200	☐	☐	221	
8	f80215a3cb8e803647deb5...	200	☐	☐	221	

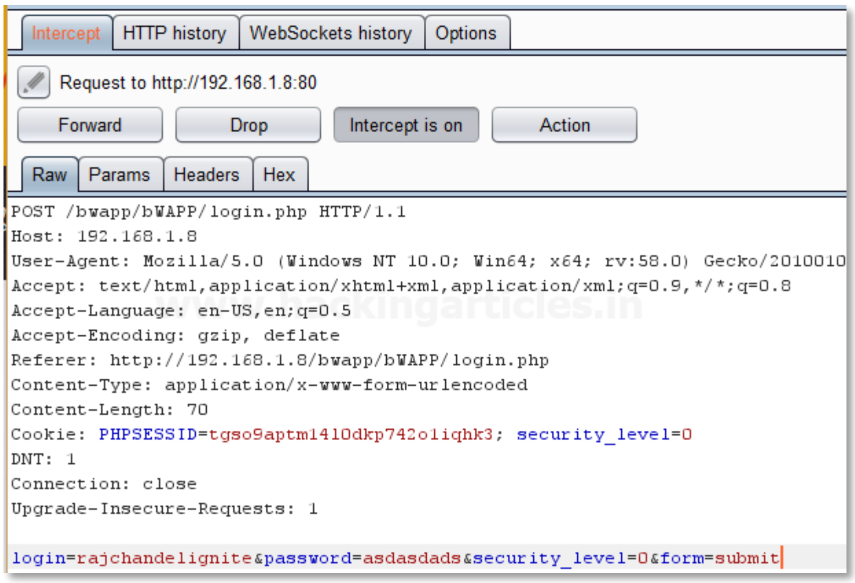
```
GET
/redirect/04-hashing/re.php?url=http://www.hackingarticles.in&hash=8258c1efb059
HTTP/1.1
```

The **Hash MD5 value**, we will give the Hash value in the **URL** of the redirecting page which is **HTTP://www.hackingarticles.in**, which will successfully redirect us to **HTTP://www.hackingarticles.in**. This shows our success in the attack as shown in the image.



Add Raw Payload

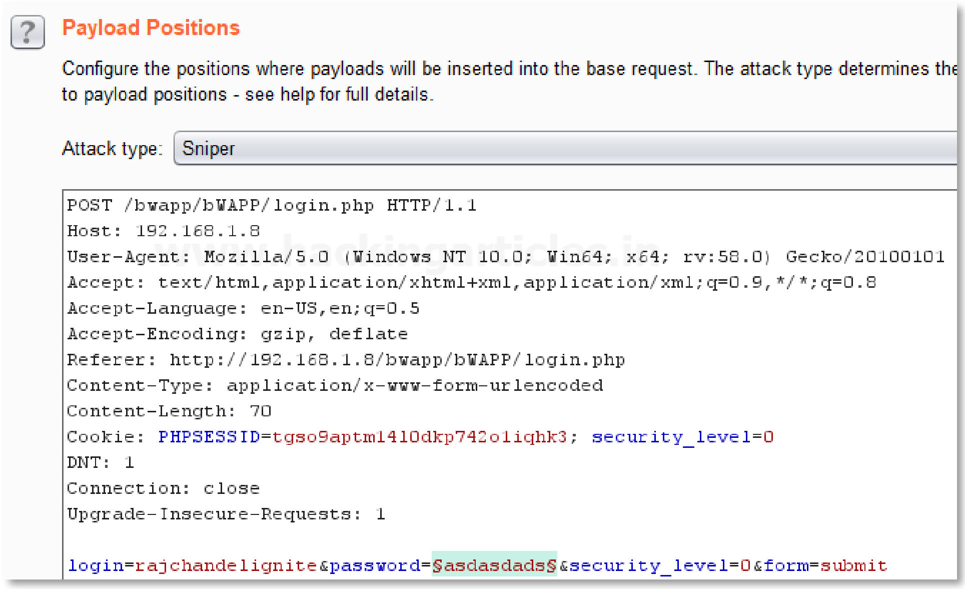
This processing rule can be used to add raw payload value before or after the current processed value. For example, it can come in handy whenever we want to submit the same payload in both raw and hashed form.
First, we have intercepted the request of the login page in the **Bwapp LAB**, where we have given default username and wrong password. Then click on login, the burp suite will capture the request of the login page in the intercept tab.



```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/2010010
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.8/bwapp/bWAPP/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 70
Cookie: PHPSESSID=tgso9aptm1410dkp742o1iqhk3; security_level=0
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

login=rajchandelignite&password=asdasdads&security_level=0&form=submit
```

Send the captured request to the **Intruder** by right-clicking on the space and selecting **Send to Intruder**option or simply press **Ctrl + i**. Now open the **Intruder tab** then select **Positions tab** and the following will be visible. Choose the **Attack type** as **Sniper.** Press on the **Clear button** as shown in the image. Now we will select the fields where we want to attack which is the password and click on **Add button.**



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

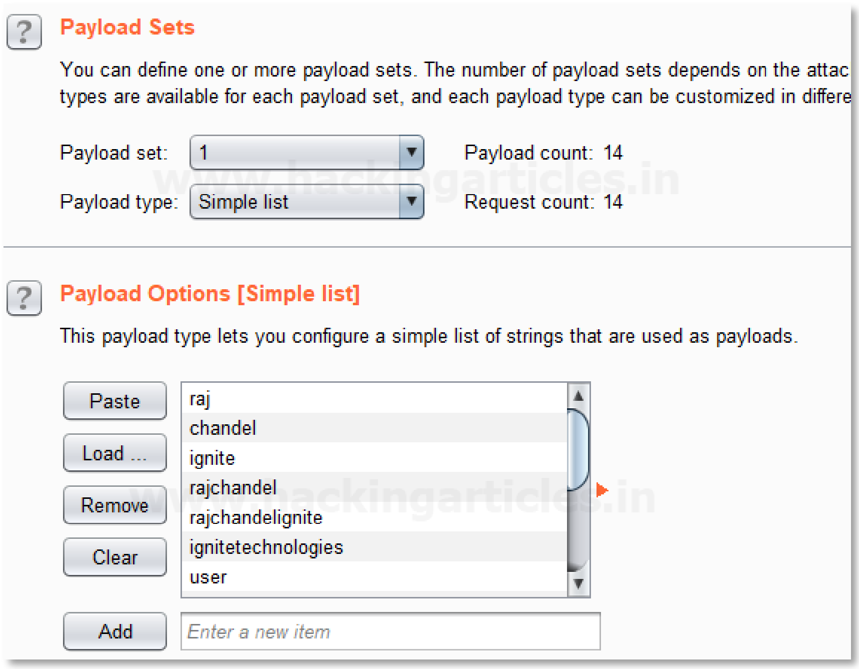
Attack type: Sniper

```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.8/bwapp/bWAPP/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 70
Cookie: PHPSESSID=tgso9aptm1410dkp742o1iqhk3; security_level=0
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

login=rajchandelignite&password=§asdasdads§&security_level=0&form=submit
```

Send the captured request to the **Intruder** by clicking on the Action Tab and follow given below step. Now open the **Intruder tab** then select **Positions tab** and you can observe the highlighted password and follow the given below step for selecting payload position.

- Press on the **Clear button** given at right of the window frame.
- Now we will select the fields where we want to attack and i.e. the password filed and click on **Add button.**
- Choose the **Attack type** as
- In the given below image, we have selected a password that means we will need one dictionary files for a password.


Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attac
types are available for each payload set, and each payload type can be customized in differe

Payload set: 1 Payload count: 14

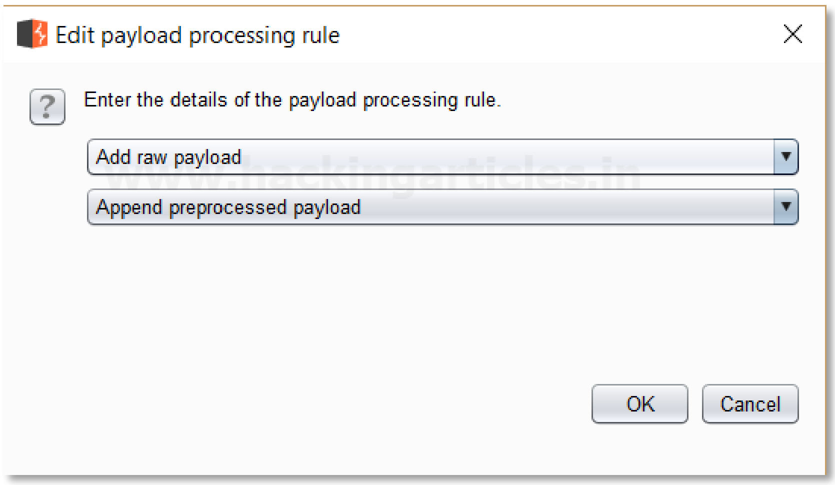
Payload type: Simple list Request count: 14

Payload Options [Simple list]

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

```
raj
chandel
ignite
rajchandel
rajchandelignite
ignitetechnologies
user
```

Before executing the attack we have added a payload processing rule to the payload type which is **Add Raw Payload** and then we have selected **Append Pre-processed Payload**. This adds a **raw payload value**before and after the **current processed value**. As you can see the **input strings** of the dictionary as a single input string is **repeated twice** which can be seen in **result window** of the **attack**.
Select **Start Attack** in the **Intruder menu** as shown in the image.

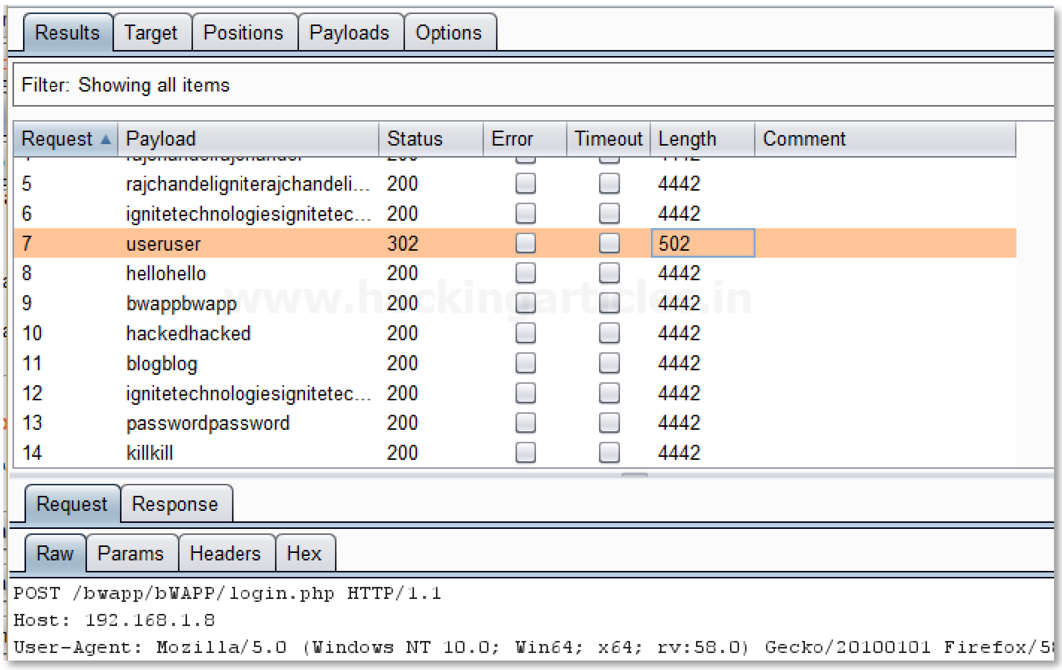

Edit payload processing rule

Enter the details of the payload processing rule.

Add raw payload

Append preprocessed payload

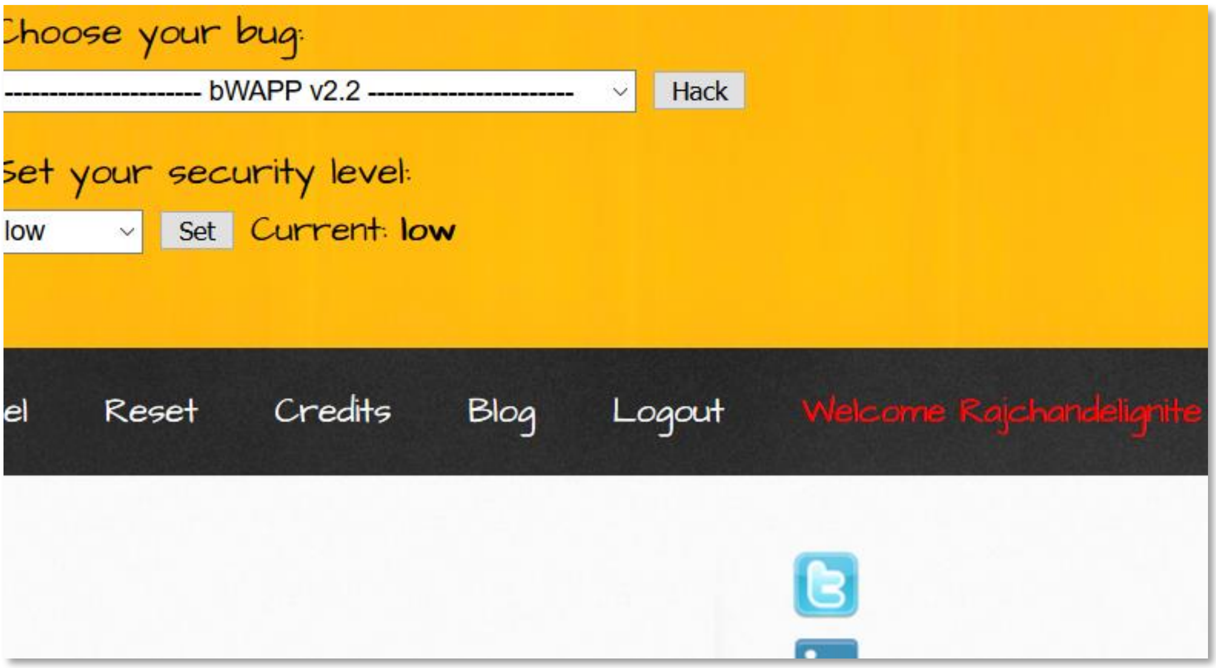
Sit back and relax because now the burp suite will do its work, match the password which will give you the correct password. The moment it will find the correct value, it will change the value of length as shown in the image.


Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
5	rajchandeligniterajchandeli...	200	☐	☐	4442	
6	ignitetechnologiesignitetec...	200	☐	☐	4442	
7	useruser	302	☐	☐	502	
8	hellohello	200	☐	☐	4442	
9	bwappbwapp	200	☐	☐	4442	
10	hackedhacked	200	☐	☐	4442	
11	blogblog	200	☐	☐	4442	
12	ignitetechnologiesignitetec...	200	☐	☐	4442	
13	passwordpassword	200	☐	☐	4442	
14	killkill	200	☐	☐	4442	

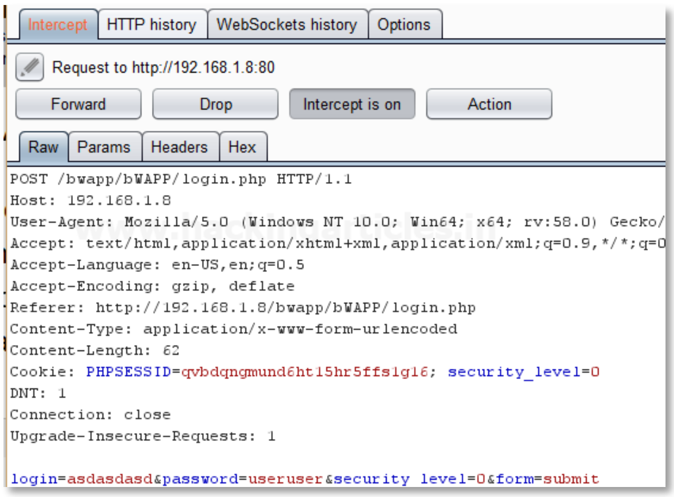
```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/5
```

And to confirm the **password matched**, we will give the password in the **Bwapp LAB login page**, which will successfully log us into the **Bwapp lab**. This shows our success in the attack as shown in the image.



Skip if Matches Regex

This processing rule can be used to check the current processed value matches a specified regular expression, and if it matches it will skip the payload and will move onto the next one. For example, Suppose we have a parameter value that has a minimum length and want to skip values in the list that are shorter than the minimum length defined.
First, we have intercepted the request of the login page in the **Bwapp LAB**, where we have given default username and wrong password. Then click on login, the burp suite will capture the request of the login page in the intercept tab.

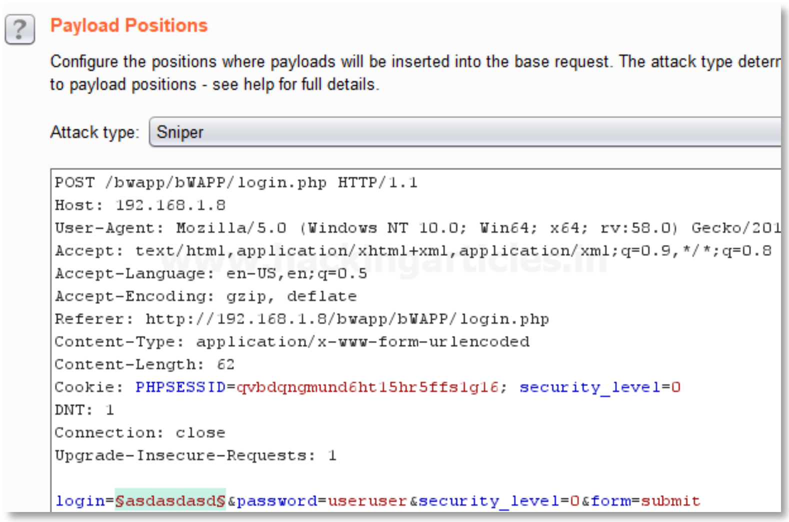


```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.8/bwapp/bWAPP/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 62
Cookie: PHPSESSID=qvbdqngmund6ht15hr5ffs1g16; security_level=0
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

login=asdasdasd&password=useruser&security_level=0&form=submit
```

Send the captured request to the **Intruder** by clicking on the Action Tab and follow given below step. Now open the **Intruder tab** then select **Positions tab** and you can observe the highlighted password and follow the given below step for selecting payload position.

- Press on the **Clear button** given at right of the window frame.
- Now we will select the fields where we want to attack and i.e. the password filed and click on **Add button.**
- Choose the **Attack type** as
- In the given below image, we have selected a password that means we will need one dictionary files for a password.



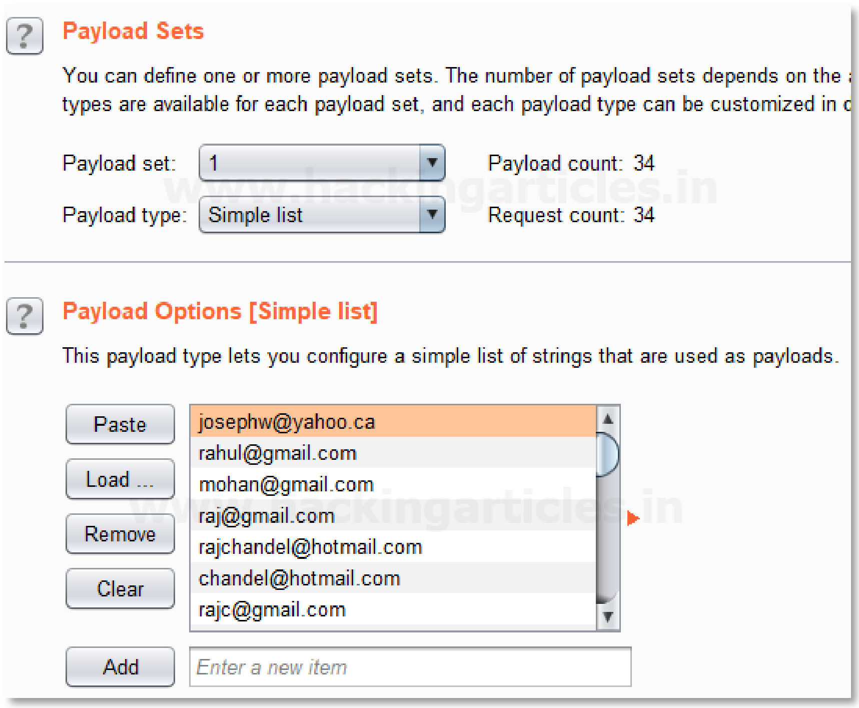
Configure the positions where payloads will be inserted into the base request. The attack type deterr
to payload positions - see help for full details.

Attack type: Sniper

```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/201
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.8/bwapp/bWAPP/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 62
Cookie: PHPSESSID=qvbdqngmund6ht15hr5ffs1g16; security_level=0
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

login=§asdasdasd§&password=useruser&security_level=0&form=submit
```

Then select the **Payload type** as **Simple list,** where we have added a dictionary by clicking on **Load** button. We can either load the dictionary or we can manually add input strings using the **Add button** in the**payload options** as shown in the image.


Payload Sets

You can define one or more payload sets. The number of payload sets depends on the
types are available for each payload set, and each payload type can be customized in

Payload set: 1 Payload count: 34

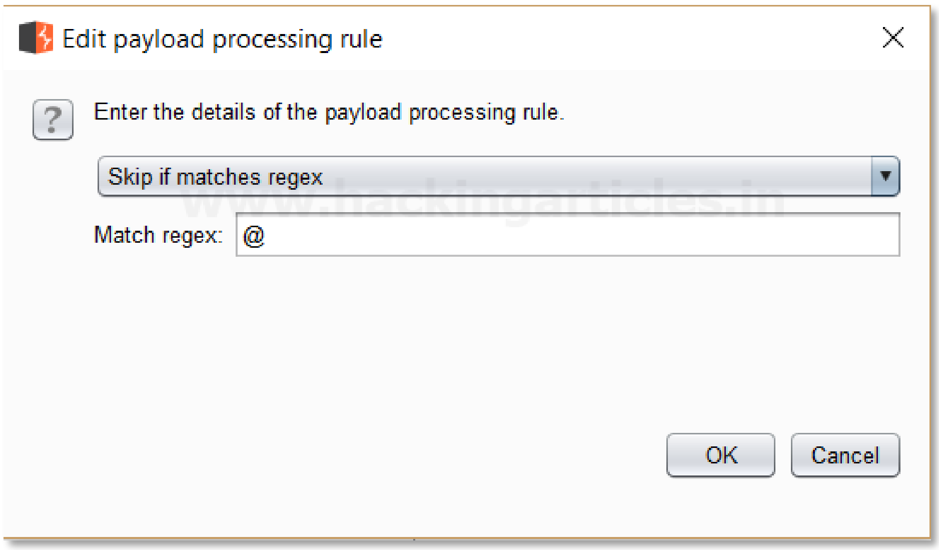
Payload type: Simple list Request count: 34

Payload Options [Simple list]

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

josephw@yahoo.ca
rahul@gmail.com
mohan@gmail.com
raj@gmail.com
rajchandel@hotmail.com
chandel@hotmail.com
rajc@gmail.com

Before executing the attack we have added a payload processing rule to the payload type which is **Skip if Matches Regex** where we have given an input of **{@}** in the **match regex** field. Here we see that as per this rule if the input is given matches with any of the input strings in the dictionary it simply skip that value and move on to next.
Now Select **Start Attack** in the **Intruder menu** as shown in the image.

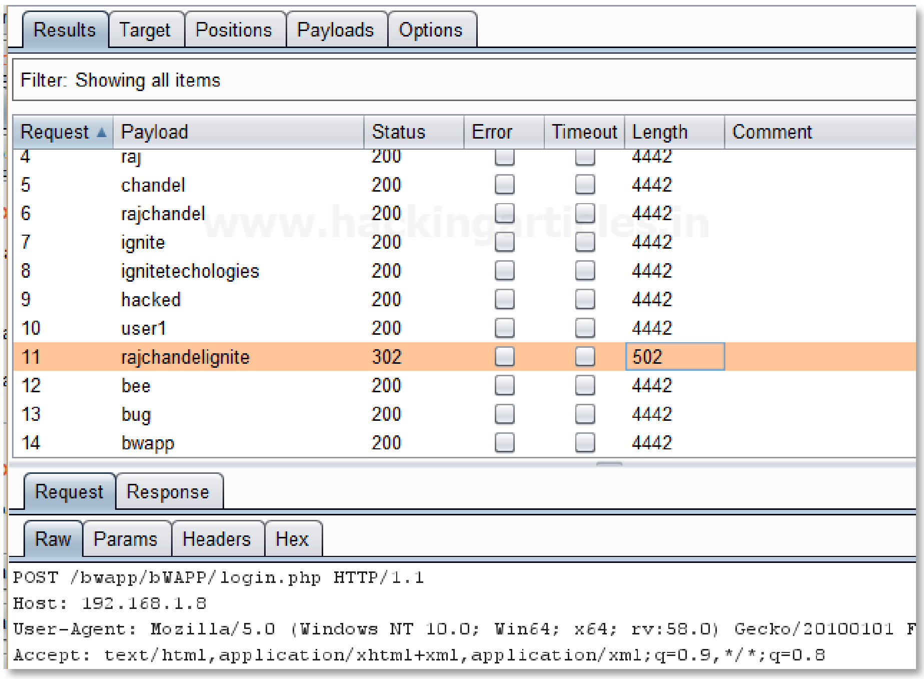

Edit payload processing rule

Enter the details of the payload processing rule.

Skip if matches regex

Match regex: @

Sit back and relax because now the burp suite will do its work, match the password which will give you the correct password. The moment it will find the correct value, it will change the value of length as shown in the image.



Request	Payload	Status	Error	Timeout	Length	Comment
4	raj	200	☐	☐	4442	
5	chandel	200	☐	☐	4442	
6	rajchandel	200	☐	☐	4442	
7	ignite	200	☐	☐	4442	
8	ignitetechologies	200	☐	☐	4442	
9	hacked	200	☐	☐	4442	
10	user1	200	☐	☐	4442	
11	rajchandelignite	302	☐	☐	502	
12	bee	200	☐	☐	4442	
13	bug	200	☐	☐	4442	
14	bwapp	200	☐	☐	4442	

```
POST /bwapp/bWAPP/login.php HTTP/1.1
Host: 192.168.1.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 F
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

About Us

About Us

"Simple training makes Deep Learning"

"IGNITE" is a worldwide name in IT field. As we provide high-quality cybersecurity training and consulting services that fulfil students, government and corporate requirements.
We are working towards the vision to "Develop India as a Cyber Secured Country". With an outreach to over eighty thousand students and over a thousand major colleges, Ignite Technologies stood out to be a trusted brand in the Education and the Information Security structure.

We provide training and education in the field of Ethical Hacking & Information Security to the students of schools and colleges along with the corporate world. The training can be provided at the client's location or even at Ignite's Training Center.
We have trained over 10,000 + individuals across the globe, ranging from students to security experts from different fields. Our trainers are acknowledged as Security Researcher by the Top Companies like - Facebook, Google, Microsoft, Adobe, Nokia, Paypal, Blackberry, AT&T and many more. Even the trained students are placed into a number of top MNC's all around the globe. Over with this, we are having International experience of training more than 400+ individuals.

The two brands, Ignite Technologies & Hacking Articles have been collaboratively working from past 10+ Years with about more than 100+ security researchers, who themselves have been recognized by several research paper publishing organizations, The Big 4 companies, Bug Bounty research programs and many more.

Along with all these things, all the major certification organizations recommend Ignite's training for its resources and guidance.
Ignite's research had been a part of number of global Institutes and colleges, and even a multitude of research papers shares Ignite's researchers in their reference.

iGNITE Technologies

What We Offer

Ethical Hacking

The Ethical Hacking course has been structured in such a way that a technical or a non-technical applicant can easily absorb its features and indulge his/her career in the field of IT security.

Bug Bounty 2.0

A bug bounty program is a pact offered by many websites and web developers by which folks can receive appreciation and reimbursement for reporting bugs, especially those affecting to exploits and vulnerabilities.
Over with this training, an indivisual is thus able to determine and report bugs to the authorized before the general public is aware of them, preventing incidents of widespread abuse.

Network Penetration Testing 2.0

The Network Penetration Testing training will build up the basic as well advance skills of an indivisual with the concept of Network Security & Organizational Infrastructure. Thereby this course will make the indivisual stand out of the crowd within just 45 days.

iGNITE Technologies

Red Teaming

This training will make you think like an "Adversary" with its systematic structure & real Environment Practice that contains more than 75 practicals on Windows Server 2016 & Windows 10. This course is especially designed for the professionals to enhance their Cyber Security Skills

CTF 2.0

The CTF 2.0 is the latest edition that provides more advance module connecting to real infrastructure organization as well as supporting other students preparing for global certification. This curriculum is very easily designed to allow a fresher or specialist to become familiar with the entire content of the course.

Infrastructure Penetration Testing

This course is designed for Professional and provides an hands-on experience in Vulnerability Assessment Penetration Testing & Secure configuration Testing for Applications Servers, Network Deivces, Container and etc.

Digital Forensic

Digital forensics provides a taster in the understanding of how to conduct investigations in order for business and legal audien ces to correctly gather and analyze digital evidence.