

Exploitation Phase

The attacker needs to exploit the vulnerability to gain access to the system/server.

In this task, we will look at the potential exploitation attempt from the attacker against our web server and see if the attacker got successful in exploiting or not.

To begin our investigation, let's note the information we have so far:

- We found two IP addresses from the reconnaissance phase with sending requests to our server.
- One of the IPs 40.80.148.42 was seen attempting to scan the server with **IP 192.168.250.70**.
- The attacker was using the web scanner Acunetix for the scanning attempt.

STEP 1: Finding the URI which got the multiple brute force attempts?

The screenshot shows a Splunk search interface with the following details:

- Search Bar:** index=botsv1 sourcetype=stream:http dest_ip="192.168.250.70" http_method=POST form_data=*username*passwd* | rex field=form_data "passwd=(?<creds>\w+)" | table src_ip creds
- Results Summary:** 413 events (before 12/13/25 6:32:23.000 AM) No Event Sampling
- Event List:** Shows a list of 413 events. The first few rows are:

src_ip	creds
192.168.250.70	password1
192.168.250.70	password2
192.168.250.70	password3
192.168.250.70	password4
192.168.250.70	password5
192.168.250.70	password6
192.168.250.70	password7
192.168.250.70	password8
192.168.250.70	password9
192.168.250.70	password10
- Time Range:** All time
- Visualizations:** None present in this screenshot.

Since we're working with a brute-forcing attempt, I went right ahead and used the following search query '**index=botsv1 sourcetype=stream:http dest_ip="192.168.250.70"**
http_method=POST form_data=*username*passwd* | rex field=form_data
"passwd=(?<creds>\w+)" | table src_ip creds'. I focused on POST requests containing form data, as login attempts typically transmit credentials this way. The following query was used to extract password submission attempts and identify the source IPs involved. This helps confirm brute-force behaviour by highlighting repeated credential activity.

[Hide Fields](#) [All Fields](#) List Format 20 Per Page [Prev](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#)

a_src_headers 100+	i Time	Event Accept-Encoding: gzip, deflate Host: imreallynotbatman.com Content-Length: 111			
a_src_ip 2 a_src_mac 1 # src_port 100+ status 1 # time_taken 100+ # timeendpos 1 a_timestamp 100+ # timestamppos 1 a_transport 1 a_un 1 a_url_path 1	url	<input type="button" value="X"/> 1 Value, 100% of events Selected Yes No 7ua23			
Reports Top values Top values by time Rare values Events with this field					
Values Count % <table border="1"><tbody><tr><td>/joomla/administrator/index.php</td><td>413</td><td>100%</td></tr></tbody></table>			/joomla/administrator/index.php	413	100%
/joomla/administrator/index.php	413	100%			
transport: tcp uri: /joomla/administrator/index.php uri_path: /joomla/administrator/index.php					

I went to the left and scrolled down the fields to find the uri. I opened it and saw that the value '**/joomla/administrator/index.php**' was shown. I went with it and the answer was correct.

STEP 2: Against which username was the brute force attempt made?

I used the query 'index=botsv1 sourcetype=stream:http dest_ip="192.168.250.70"
`http_method=POST form_data=*username*passwd* | rex field=form_data
"passwd=(?<creds>\w+)" | table _time src_ip uri http_user_agent creds`', but this time appending | `table_time src_ip uri http_user_agent creds`. This query displayed 2 IP addresses responsible for the attempts. (23.22.63.114) which was used automated using Python-urllib/2.7 as the http_user_agent on our server as well as (40.80.148.42) which performed a single attempt using Mozilla/5.0 browser. The password used for this attempt was batman.

a date_wday 1
date_year 1
date_zone 1
a dest_content 1
a dest_headers 1
a dest_ip 1
a dest_mac 1
dest_port 1
duplicate_packets_in 1
duplicate_packets_out 1
a endtime 1
a form_data 1
a http_comment 1
http_content_length 1
a http_content_type 1
a http_method 1
a http_referrer 1
a http_user_agent 1
a index 1
linecount 1

Time Event
dest_mac: 00:0C:29:C4:02:7E
dest_port: 80
duplicate_packets_in: 1

form_data

1 Value, 100% of events Selected Yes No

Reports

Top values Top values by time Rare values

Events with this field

Values	Count	%
username=admin&passwd=batman&option=com_login&task=login& return=aW5kZXgucGhw&e5ec827a3f67ce0fc546d81f7356acc=1	1	100%

packets_out: 5
apply_time: 1677760

I clicked on batman under the filter '**creds**' and clicked on 'view events'. I went to the left side of the page and scrolled through the filters to **form_data** where I found the username - **admin**.

STEP 3: What was the correct password for admin access to the content management system running **imreallynotbatman.com**?

a date_wday 1
date_year 1
date_zone 1
a dest_content 1
a dest_headers 1
a dest_ip 1
a dest_mac 1
dest_port 1
duplicate_packets_in 1
duplicate_packets_out 1
a endtime 1
a form_data 1
a http_comment 1
http_content_length 1
a http_content_type 1
a http_method 1
a http_referrer 1
a http_user_agent 1
a index 1
linecount 1

Time Event
dest_mac: 00:0C:29:C4:02:7E
dest_port: 80
duplicate_packets_in: 1

form_data

1 Value, 100% of events Selected Yes No

Reports

Top values Top values by time Rare values

Events with this field

Values	Count	%
username=admin&passwd=batman&option=com_login&task=login& return=aW5kZXgucGhw&e5ec827a3f67ce0fc546d81f7356acc=1	1	100%

packets_out: 5
apply_time: 1677760

The correct password here was **batman**.

STEP 4: How many unique passwords were used to attempt the brute force attack?

src_ip		creds
40.80.148.42		batman
23.22.63.114		rock
23.22.63.114		sammy
23.22.63.114		cool
23.22.63.114		august
23.22.63.114		baby
23.22.63.114		dave

Looking at the stats. There were a total of '**412**' for the IP address **23.22.63.114**.

STEP 6: What IP address is likely attempting a brute force password attack against **imreallynotbatman.com**?

src_ip		creds
40.80.148.42		batman
23.22.63.114		rock
23.22.63.114		sammy
23.22.63.114		cool
23.22.63.114		august
23.22.63.114		baby
23.22.63.114		dave

The IP address is '**23.22.63.114**'.

STEP 7: After finding the correct password, which IP did the attacker use to log in to the admin panel?

Splunk Search Results					
1 index=botsv1 sourcetype=stream:http dest_ip="192.168.250.70" http_method=POST form_data=username+password+ rex field=form_data "passwd=(?<creds>\w+)" table _time src_ip uri http_user_agent creds					
413 events (before 12/13/25 7:04:47:000 AM) No Event Sampling ▾					
Events (413)	Patterns	Statistics (413)	Visualization	All time ▾	<input type="button" value="Q"/>
20 Per Page ▾	Format	Preview ▾		< Prev 1 2 3 4 5 6 7 ... Next >	
_time	src_ip	uri	http_user_agent	creds	
2016-08-10 21:48:05.858	48.80.148.42	/joomla/administrator/index.php	Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko	batman	
2016-08-10 21:46:51.394	23.22.63.114	/joomla/administrator/index.php	Python-urllib/2.7	rock	
2016-08-10 21:46:51.156	23.22.63.114	/joomla/administrator/index.php	Python-urllib/2.7	sammy	
2016-08-10 21:46:51.154	23.22.63.114	/joomla/administrator/index.php	Python-urllib/2.7	cool	
2016-08-10 21:46:50.873	23.22.63.114	/joomla/administrator/index.php	Python-urllib/2.7	august	
2016-08-10 21:46:50.640	23.22.63.114	/joomla/administrator/index.php	Python-urllib/2.7	baby	
2016-08-10 21:46:50.437	23.22.63.114	/joomla/administrator/index.php	Python-urllib/2.7	duke	

Going back to our first step. I saw that the only IP that was able to do this was – **'40.80.148.42'**

Conclusion Summary

During the exploitation phase, Splunk was used to analyze HTTP POST traffic and identify a brute-force attack targeting the Joomla administrator login page. By examining form data and credential submission attempts, the investigation confirmed repeated password attempts against the admin account. One IP conducted large-scale automated brute-forcing, while a second IP successfully authenticated using the correct credentials. This analysis confirmed that the attacker gained administrative access to the CMS, completing the exploitation stage of the attack lifecycle.