Investigating Web Attacks

Module 09

Investigating Web Attacks

Web attacks are mainly intended to disrupt corporate web resources. Preventing attacks on web applications should be one of the top priorities for any organization. Investigating web attacks involves analyzing web server and local system logs to identify the type and source of the attacks.

ICON KEY

Valuable information

Test your knowledge

Web exercise

Workbook review

Lab Scenario

Forensic investigators need to investigate web attacks on a web server to identify the type of web attack and its severity. As a part of this investigation, investigators must analyze the web server's log files, IDS, and SIEM tools which contains the IP address, browser, and system information of the attacker. An example of a severe, large-scale web attack would be a scenario where the website of a renowned banking organization is brought down by attackers and the entire data on it, including account holders' details, employers' and employees' details, etc., is leaked online on the dark web. To investigate this web attack, investigators would need to analyze the IDS logs, web server logs, and SIEM alerts. By analyzing these details, they can determine the types of web attacks that have been attempted by the attackers. These may include cross-site scripting, SQL injection, brute-force, and various other attacks.

As an expert forensic investigator, you should know how to investigate web-related attacks.

Lab Objectives

The objective of this lab is to help the students understand how to investigate various types of web attacks, including:

- Detection and analysis of an XSS attack by examining Apache logs, ModSecurity logs, and IIS logs
- Detection and analysis of a SQL injection attack by examining Apache logs
- Detection and analysis of a directory traversal attack by examining Apache logs, ModSecurity logs, and IIS logs
- Detection and analysis of a command injection attack by examining Apache logs
- Detection and analysis of an XXE attack by examining Apache logs
- Detection and analysis of a brute-force attack by examining Apache logs and IIS logs

Lab Environment

To carry out this lab, you need:

- A computer running **Windows Server 2016** virtual machine
- A web browser with internet access
- Administrative privileges to install and run the tools

Tools demonstrated in this lab are available in C:\CHFI-Tools\CHFIv10 Module 09 Investigating Web Attacks

Lab Duration

Time: 50 minutes

Overview of Investigating Web Attacks

Web attacks are of different types. A **denial-of-service (DoS) attack** can be cited as an example of a web attack. In DoS attacks, visitors to a website are denied access to any information or services available on the website. In such cases, visitors may report the unavailability of online services which they were trying to access. Attackers can also use other techniques, such as SQL injection, command injection, brute-force attacks etc., to exploit any vulnerable website.

An indication of a web attack can be the redirection of a user to an unknown website. Unusually slow network performance and frequent server reboots may also indicate a web attack. In case a web attack is suspected, investigators need to examine the log data from various sources to understand where the attack originated and mitigate it at the earliest.

Lab Tasks

Recommended lab to assist you in investigating web attacks:

- Identifying and Investigating Web Application Attacks Using Splunk

Lab Analysis

Analyze and document the results related to the lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Lab

1

Identifying and Investigating Web Application Attacks Using Splunk

Log files record events and activities that take place in an operating system or during the runtime of a computer software or service. Forensic examination of these log files help investigator identify malicious activities that take place on the web server.

ICON KEY

Valuable information

Test your knowledge

Web exercise

Workbook review

Lab Scenario

An e-commerce company has suffered a cyber-attack on its web application that lists and sells various goods/products. The attackers not only took down the website but also stole personally identifiable information of its users/customers. The company sought the services of the cyber-forensics wing of the state law enforcement department to crack the case, retrieve customer-specific stolen information, secure its systems and all customer-related information, and thus help restore the customers' faith in their business.

To investigate the case and determine the types of web attacks that have been carried out by the attackers on the company's web application, Robert, the forensic investigator, now needs to examine and analyze log files from the Apache server, IIS server, and ModSecurity web application firewall using SIEM tool.

Lab Objectives

The objective of this lab is to help you understand how to examine log files and look for artifacts pertaining to web application attacks with SIEM tool.

Lab Environment

- A computer running **Windows Server 2016** virtual machine
- Administrative privileges to execute the commands
- A web browser with internet access
- **Splunk Enterprise** already installed on the machine

Note: You can download the latest version of **Splunk Enterprise** from the link https://www.splunk.com/en_us/download/splunk-enterprise.html

If you are using the latest version of software for this lab, then the steps and screenshots demonstrated in the lab might differ

Note: Make sure that **Real Time Protection** is disabled in **Windows 10** virtual machine (if it is running) before beginning this lab.

Lab Duration

Time: 50 minutes

Overview of the Lab

This lab familiarizes you with the process of examining log files generated by servers and web application firewalls to check for indicators of web attacks using SIEM tools like **Splunk Enterprise**.

Note: In this lab, we will be examining logs generated by **Apache** and **IIS Servers** as well as logs generated by **ModSecurity** web application firewall. The **Apache** and **ModSecurity** logs being examined in this lab have been generated on an Ubuntu machine, while the **IIS logs** have been generated on a Windows Server 2016 virtual machine.

Through the course of this lab, you will come across instances where you need to apply plain-text and encoded filters to fetch specific log entries pertaining to various web attacks. In cases where the **Splunk Enterprise** application does not show you suggestions through dropdowns for the filters that you are typing into its New Search field, you will have to manually type them and then click on the Search icon. The number of log entries obtained upon uploading an evidence file or upon applying filters in Splunk Enterprise might vary in your lab environment.

Lab Tasks

TASK 1

Launch Splunk Enterprise

1. In this lab, we will be using the **Splunk Enterprise** application to detect and examine various types of web attacks.
2. Login to **Windows Server 2016** virtual machine.
3. To launch **Splunk Enterprise**, launch a web browser, type **10.0.0.16:8000** in the browser's address bar and press **Enter**. Splunk login page appears, enter the login credentials (as you had set in **Lab 1** of **CHFIv10 Labs Module 8 Network Forensics**) and then click **Sign In**.

Note: If you face any issue launching Splunk, launch a command prompt and issue the command **"C:\Program Files\Splunk\bin\splunk.exe" start**. Splunk login page appears in a new tab.

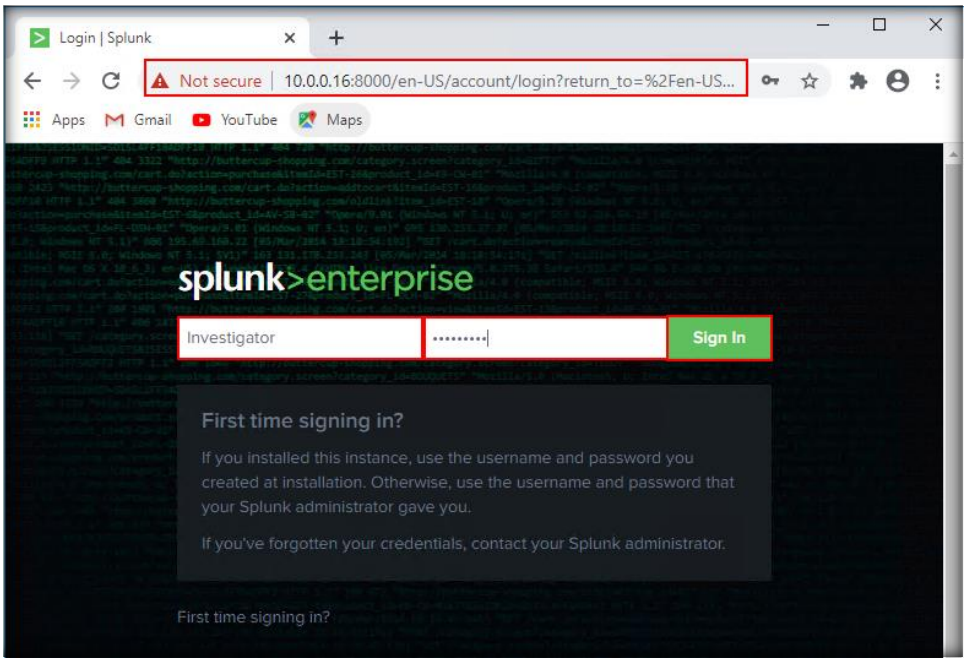

FIGURE 1.1: Log in to Splunk Enterprise application

TASK 2

Detect and Examine XSS Attack via Log Files

4. Upon logging in to **Splunk Enterprise**, you will see its homepage, where you will find the **Add Data** option, as shown in the screenshot below. Click on **Add Data**.

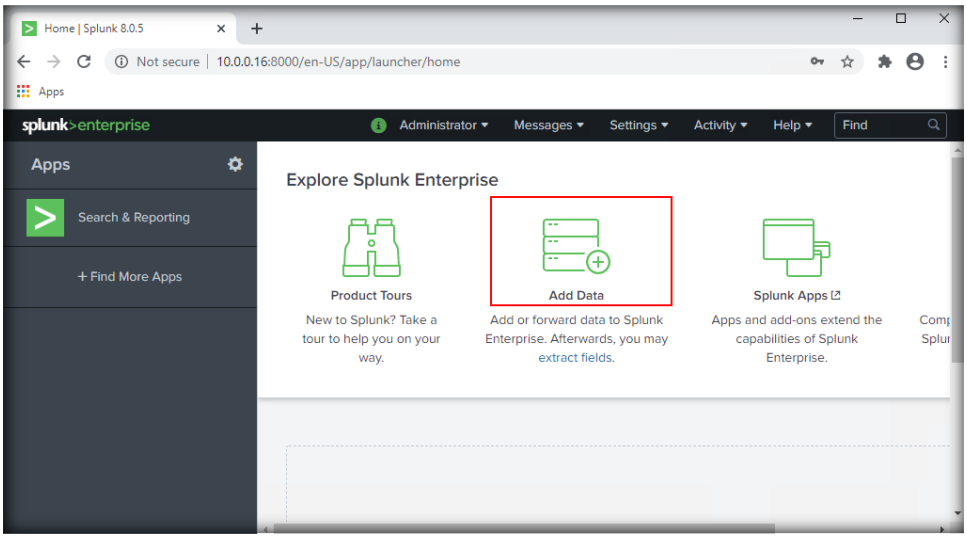

FIGURE 1.2: Add Data

5. If the **Welcome, Administrator** window appears, click **Skip**.

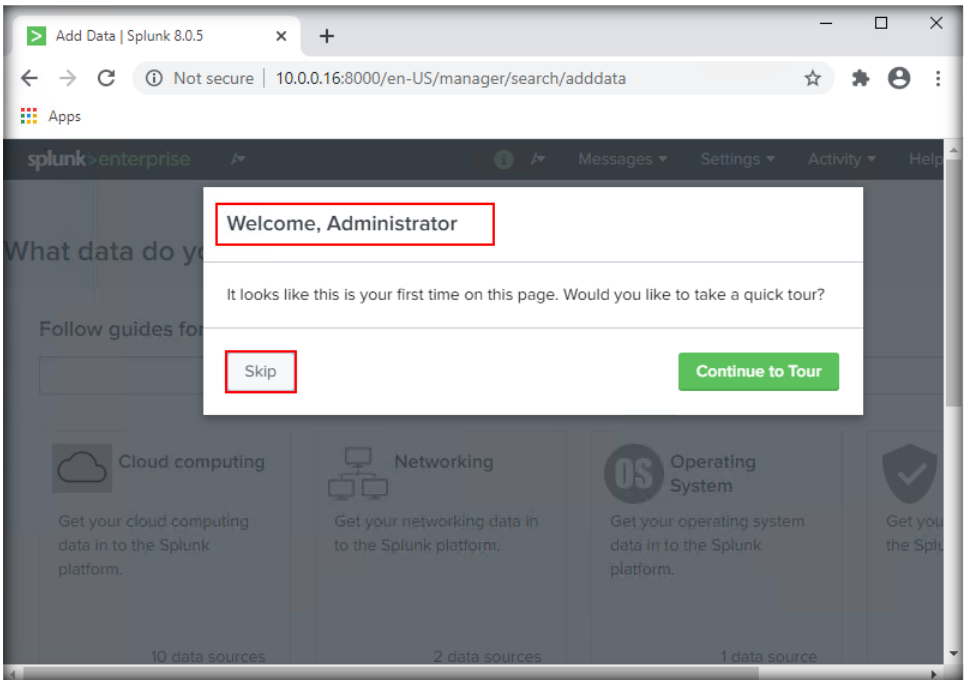

FIGURE 1.3: Welcome window

6. Now, scroll down the homepage to find and click the **Upload files from my computer** option.

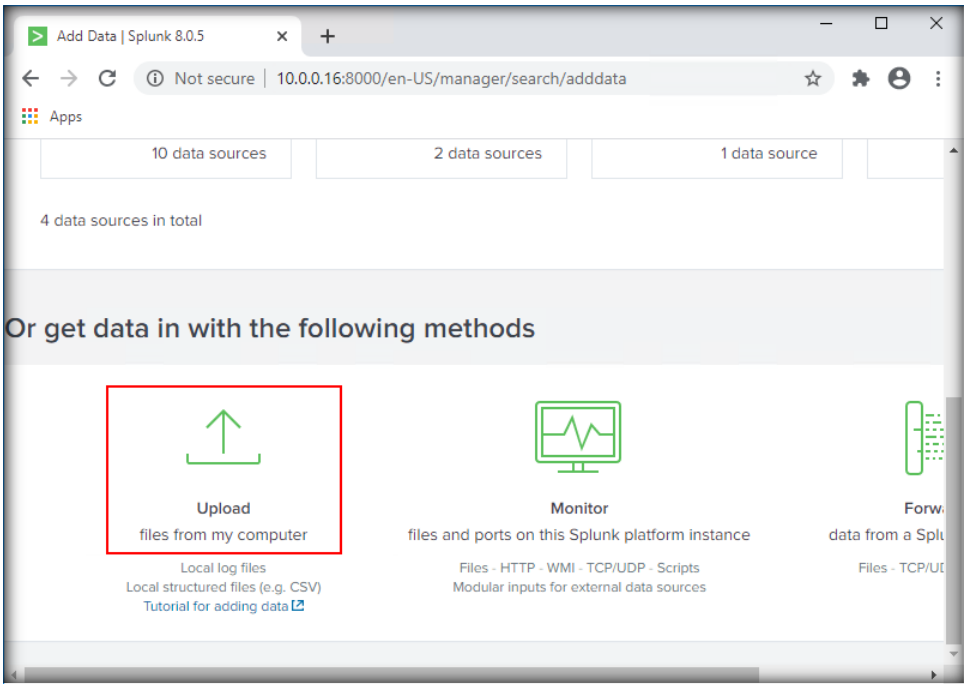

FIGURE 1.4: Click Upload

7. The **Select Source** section will now appear. Click **Select File**, as shown in the screenshot below:

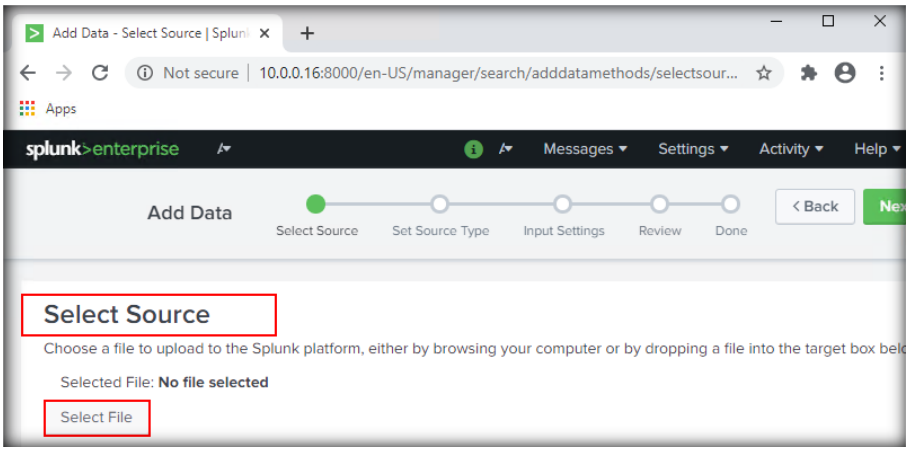

FIGURE 1.5: Go to Select Source and click on Select File

8. When an **Open** window appears, navigate to **C:\CHFI-Tools\Evidence Files\Log Files\Apache Logs** and select **XSS**. Click **Open** to upload the file.

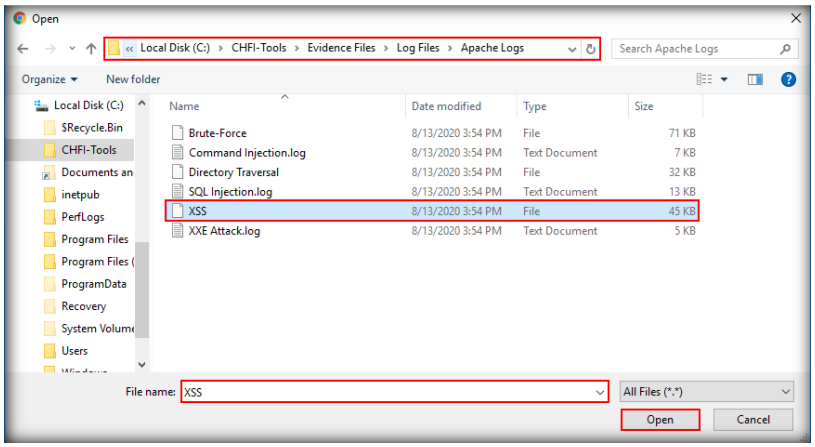

FIGURE 1.6: Select XSS file and click Open

9. The **XSS** file will now be uploaded successfully. Click **Next** to continue.

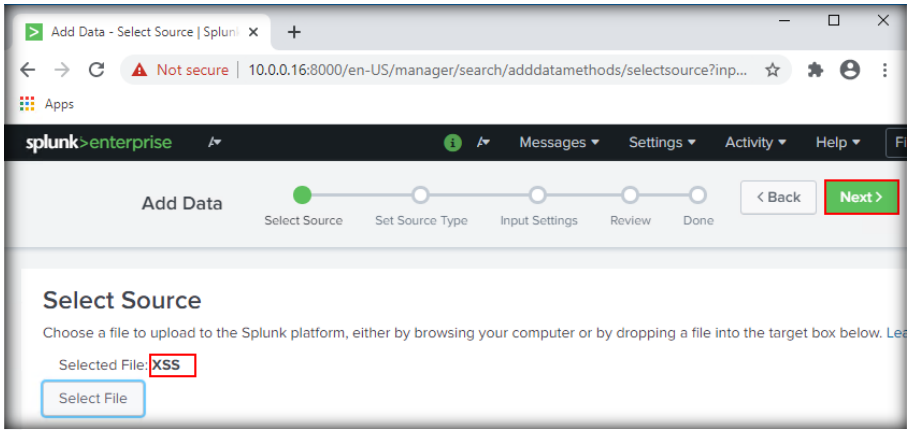

FIGURE 1.7: Click Next

10. In the next step, you will see the **Set Source Type** section. Click **Save As**, as shown in the following screenshot:

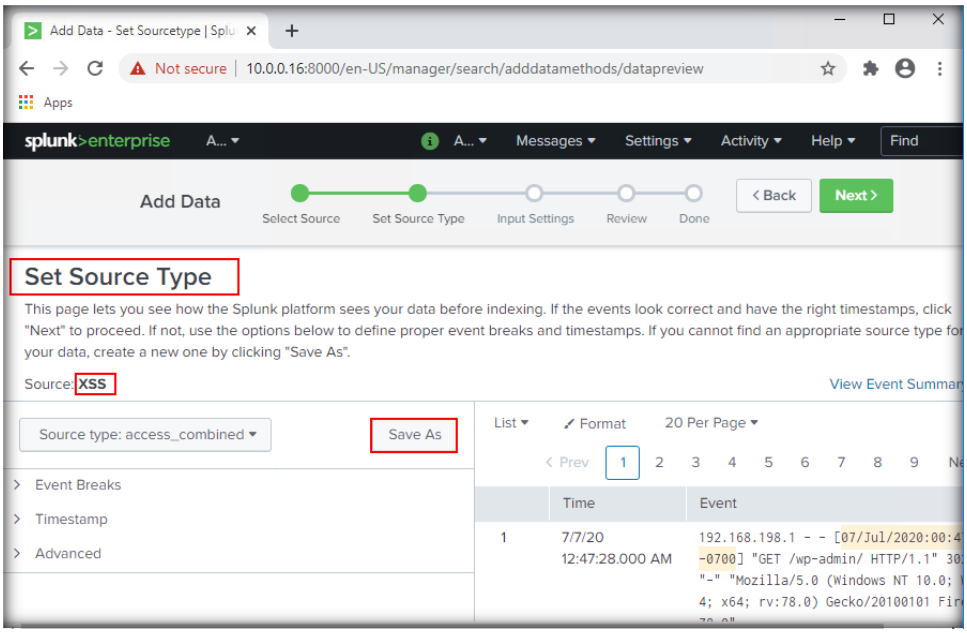

FIGURE 1.8: Click Save As in Set Source Type section

11. The **Save Source Type** window will now appear. Enter **XSS (Apache Logs)** in the **Name** field. Then, click **Save** to continue.

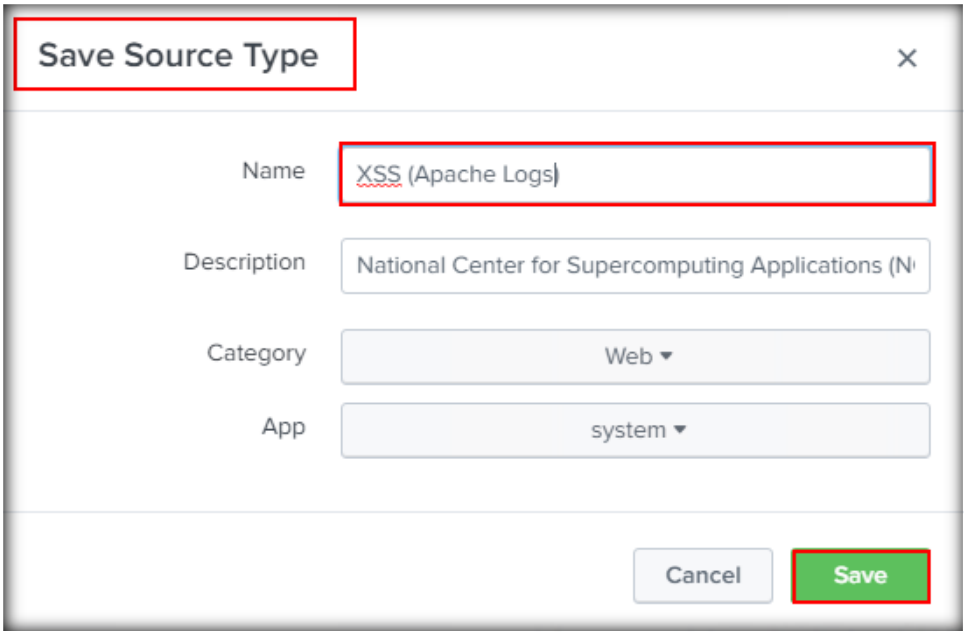

FIGURE 1.9: Enter details in Save Source Type window and click Save

12. You will now get back to the **Set Source Type** section. Click **Next** to proceed further, as shown in the screenshot:

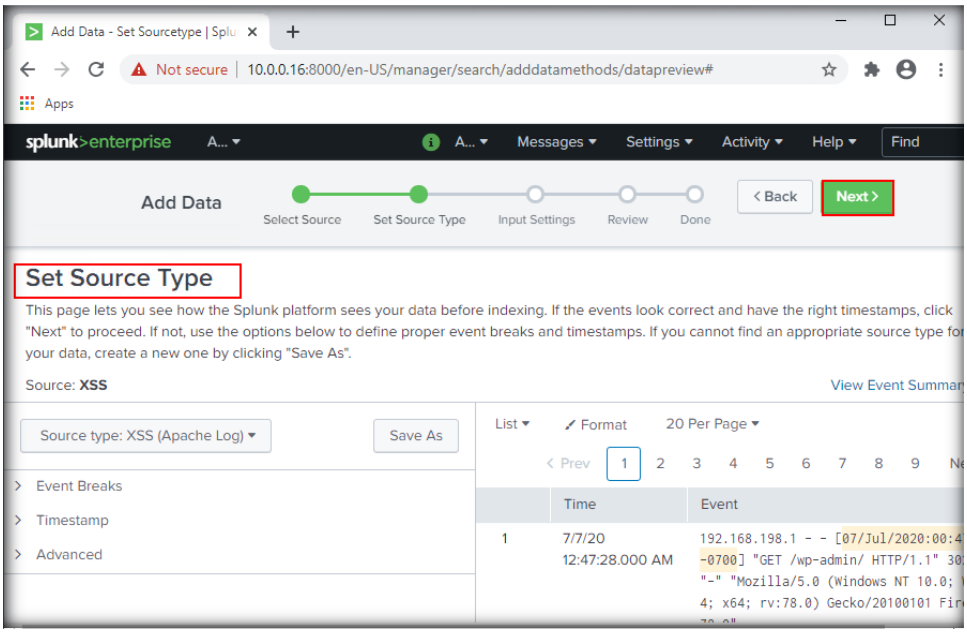

FIGURE 1.10: Click Next in Set Source Type section

13. In the next step, you will see the **Input Settings** section. Click **Review** to proceed further.

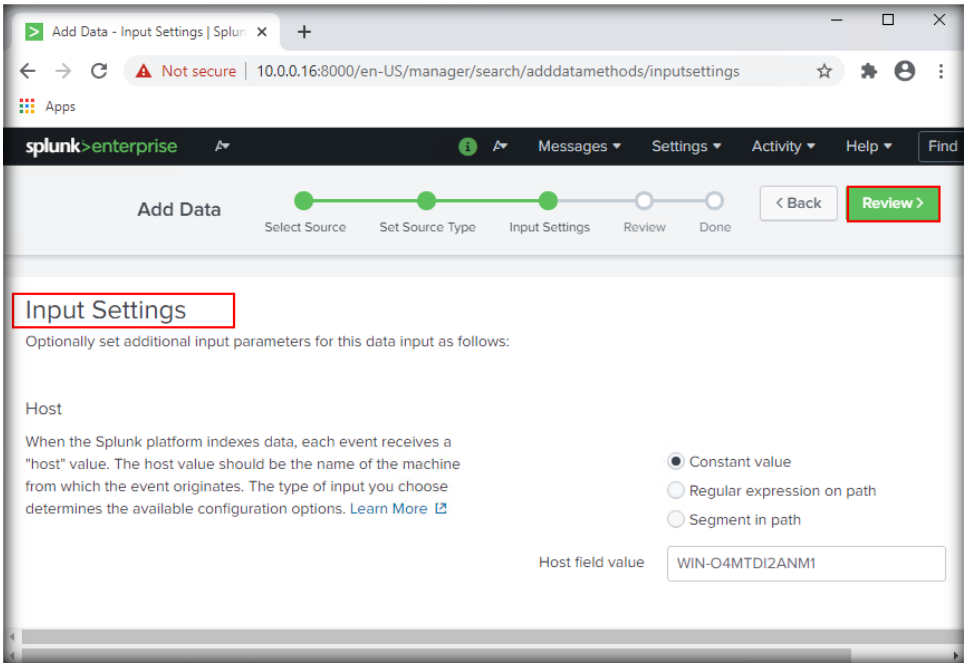

FIGURE 1.11: Click on Review in the Input Settings section

14. In the next step, you will see the **Review** section. Ensure that the details under the **Review** section are correct and then click **Submit** to proceed further. If necessary, apply corrections to the details by revisiting the previous sections by clicking on **Back**, and then, come back again to the **Review** section and click Submit to proceed further.

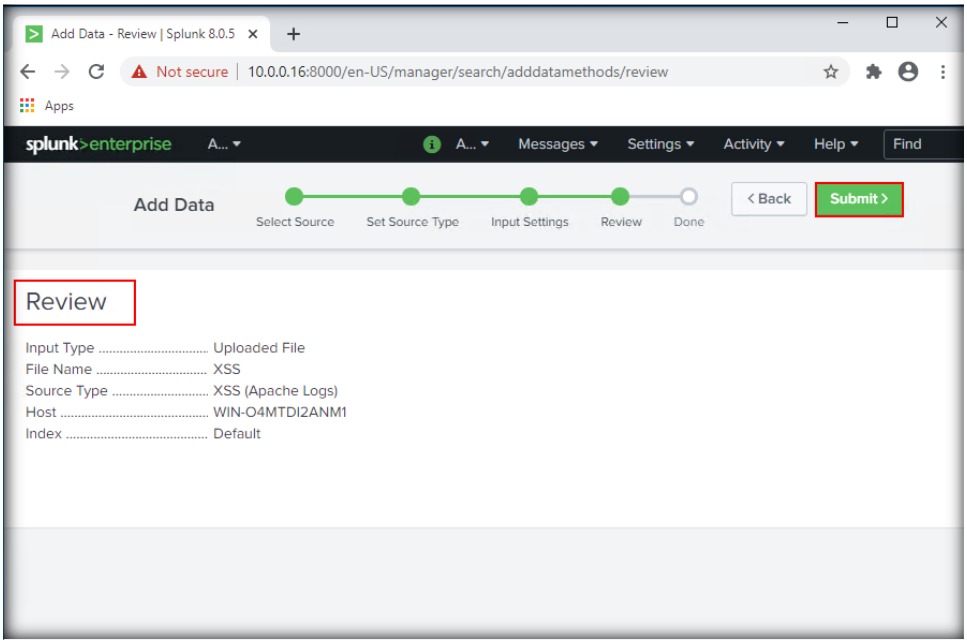

FIGURE 1.12: Review the details and click Submit

15. After clicking **Submit**, the application window will display the following message: **File has been uploaded successfully**. Now, click **Start Searching** to examine the log file.

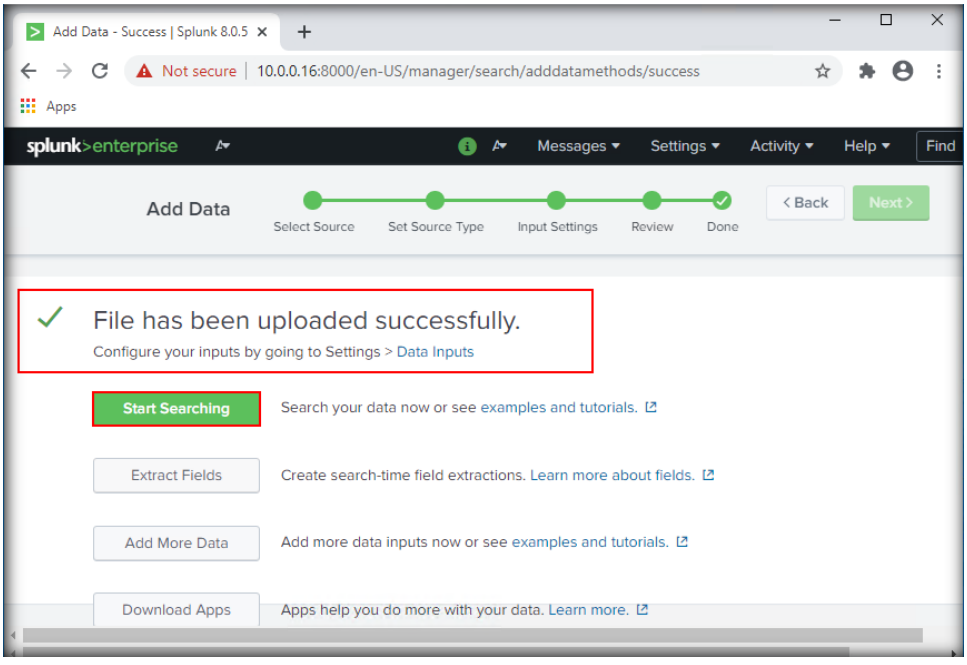

FIGURE 1.13: Click Start Searching

16. If you see the **Welcome, Administrator** window, click **Skip**.

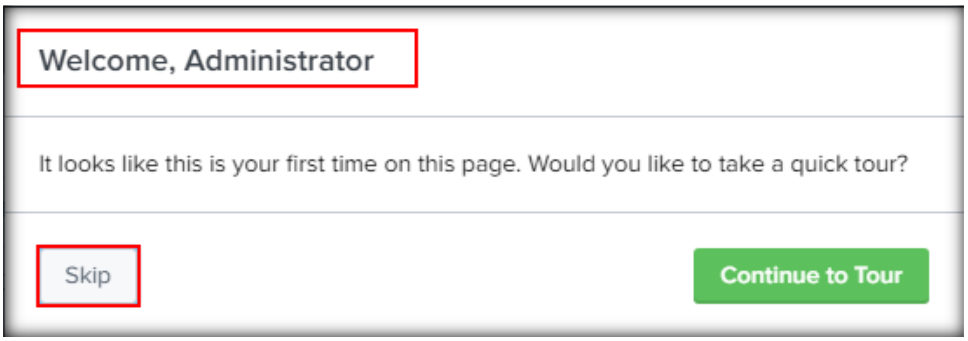

FIGURE 1.14: Skip the Welcome, Administrator window

17. You will now see the log entries from the uploaded file displayed in **Splunk Enterprise** window under the **Events** tab, as shown in the screenshot below. Our task now is to examine the log file for signs of an **XSS attack**.

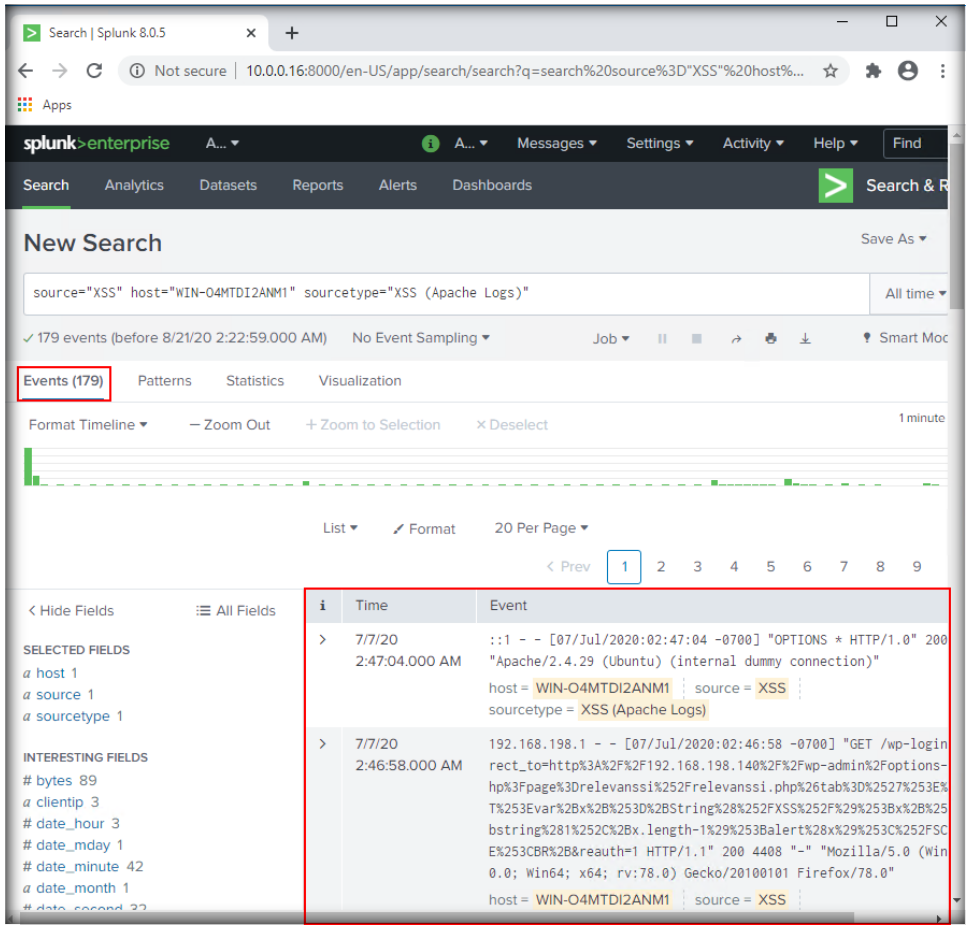

FIGURE 1.15 Log entries displayed in Splunk

18. To examine the log file for signs of an **XSS** attack, we will first apply the **<script>** filter. To do this, type **<script>** in the search box and then click the **Search** icon. However, no results are displayed upon applying this filter, as seen in the screenshot below:

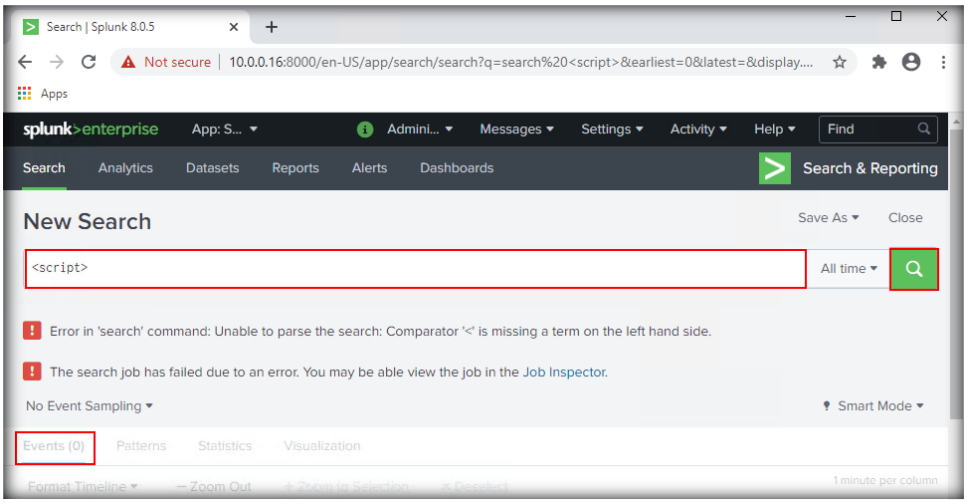

FIGURE 1.16: Use filter <script>

19. Now, we will apply an encoded form of **<script>** filter to look for results pertaining to **XSS** attacks. To apply the encoded filter, type **%3c** in the search box. Upon typing **%3c**, we can observe that a search term suggestion relating to **%3c** is displayed through a drop-down which appears under the **New Search** box. The suggestion reads **%3c%2fscript%3e%3cbr**. Select the suggestion and then click the **Search** icon.

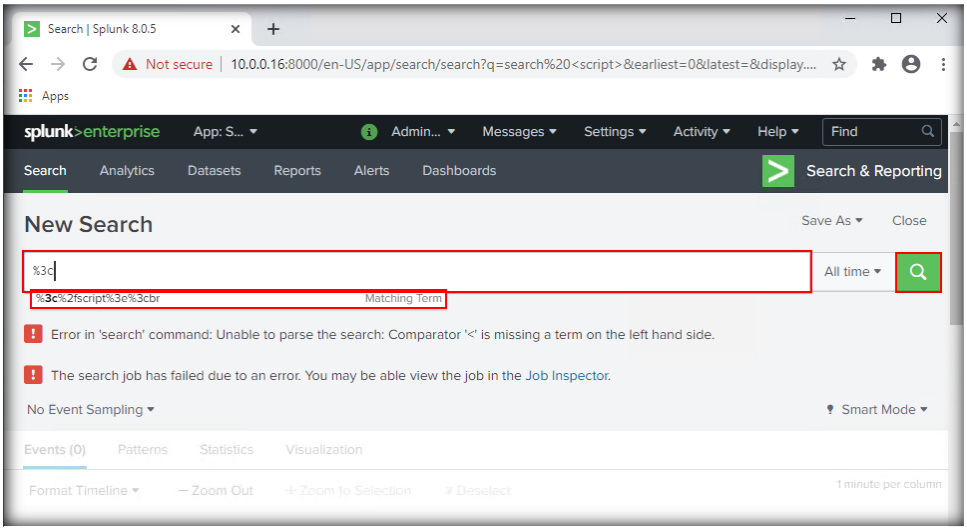

FIGURE 1.17: Enter the encoded filter

Note: In case you do not see the search term suggestion **%3c%2fscript%3e%3cbr** in the drop-down appearing below the **New Search** box, manually type the entire filter (i.e., **%3c%2fscript%3e%3cbr**) in the **New Search** box and then click on the **Search** icon.

20. We can now see a couple of log entries displayed as results under the **Events** tab, as shown in the screenshot below:

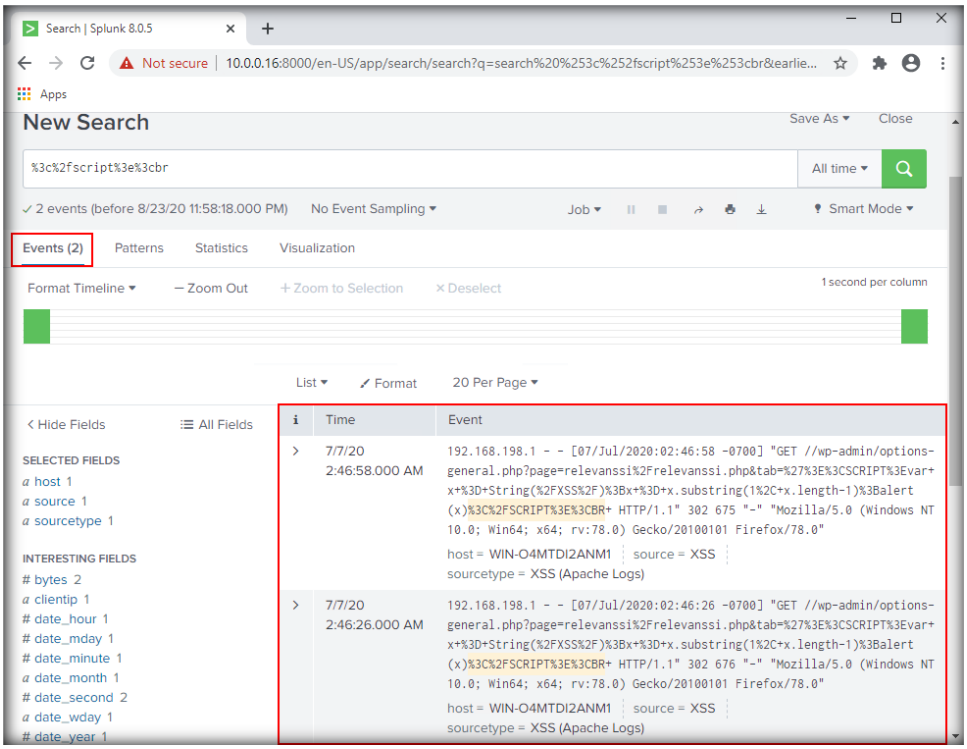

FIGURE 1.18: Results using encoded filter

Note: For the ease of demonstration, we have used a log file containing entries specific only to recorded web attacks. In real-time scenarios, however, you might see several log entries upon applying a filter and you might have to scroll down the window to manually look for the log entry containing indicators of an **XSS** attack.

21. When we examine the first result obtained upon applying the encoded filter, we can see the presence of an encoded script in the query string. The encoded script is highlighted in the screenshot below:

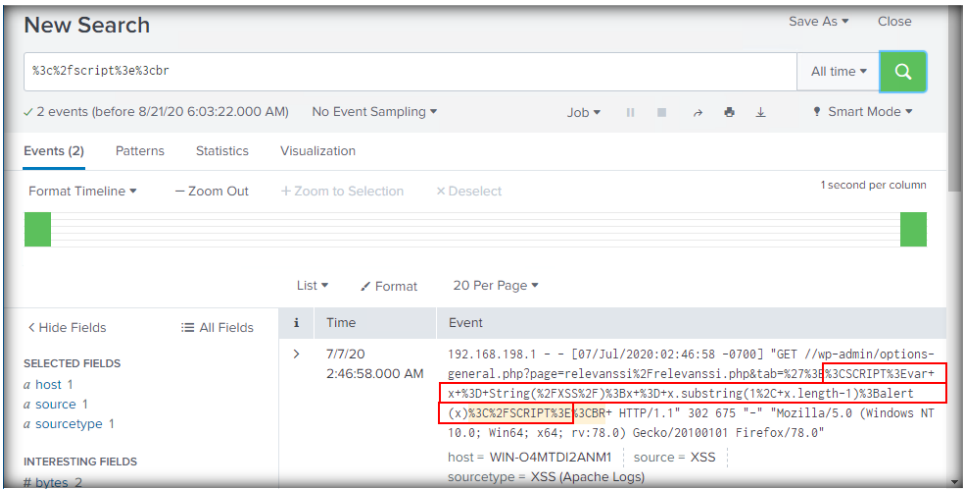

FIGURE 1.19: Encoded script found in log entry after applying the filter

22. The encoded script reads as **%3CSCRIPT%3Evar+x+%3D+String(%2FXSS%2F)%3Bx+%3D+x.substring(1%2C+x.length-1)%3Balert(x)%3C%2FSCRIPT%3E**. To determine if the encoded script is malicious, we can decode it by referring to the table provided below:

Encoded Value	Decoded Character
%3C	<
%20	space
%22	“
%3A	:
%3E	>
%2F	/
%26	&
%3B	;
%3D	=
%2C	,
%27	‘
%2D	-

TABLE 1.1: HTML URL decoder table

23. With the help of the table above, we can decode the above encoded script as **<SCRIPT>var x = String(/XSS/);x = x.substring(1, x.length-1);alert(x)</SCRIPT>** (**+** will be converted to **Space** while decoding encoded JavaScript). By examining the decoded JavaScript, we can infer that the log entry under examination indicates an XSS attack attempt on the webpage.
24. Based on a detailed observation of the log entry, we can summarize our findings as follows:
    A. Date and time of the attack: **07th July 2020** and **2:46:58 AM**
    B. IP address of the attacker: **192.168.198.1**
    C. The webpage which the attacker had targeted: **//wp-admin/options-general.php?** (this indicates it was an attack on the **admin** page of a **WordPress** site)
    D. Malicious script used in the query string: **%3CSCRIPT%3Evar+x+%3D+String(%2FXSS%2F)%3Bx+%3D+x.substring(1%2C+x.length-1)%3Balert(x)%3C%2FSCRIPT%3E**

E. HTTP status code **302**: This indicates URL redirection. This indicates that the attacker gained admin privileges by using the malicious **XSS** script.

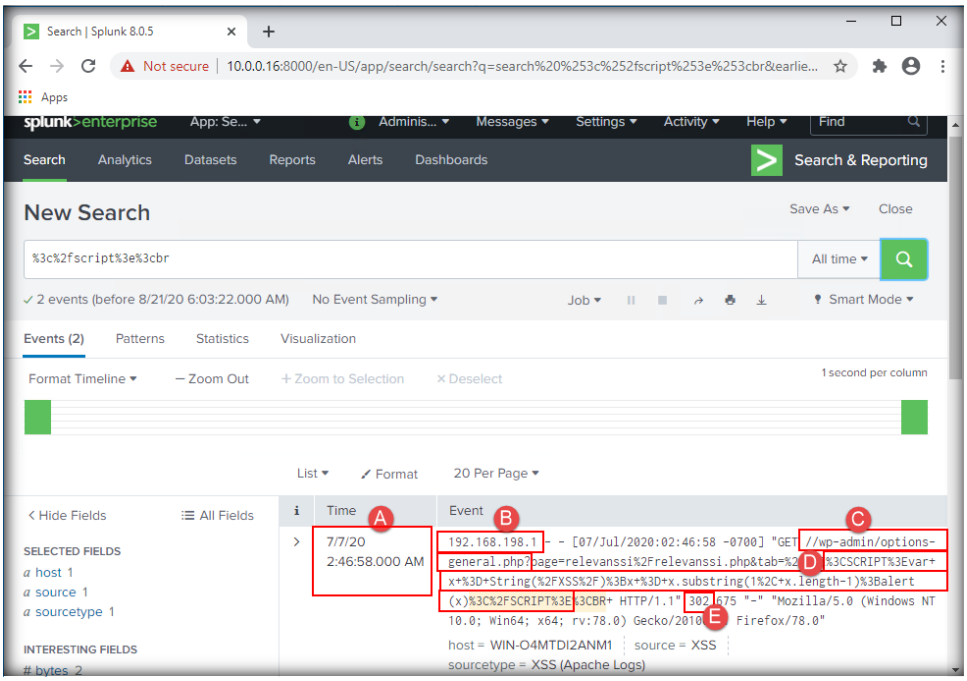

FIGURE 1.20: Detailed analysis of XSS attack

25. We will now examine the **XSS** attack log file generated by **ModSecurity**, an open-source web application firewall.

26. Click on **splunk>enterprise** at the top left of the **Splunk Enterprise** webpage.

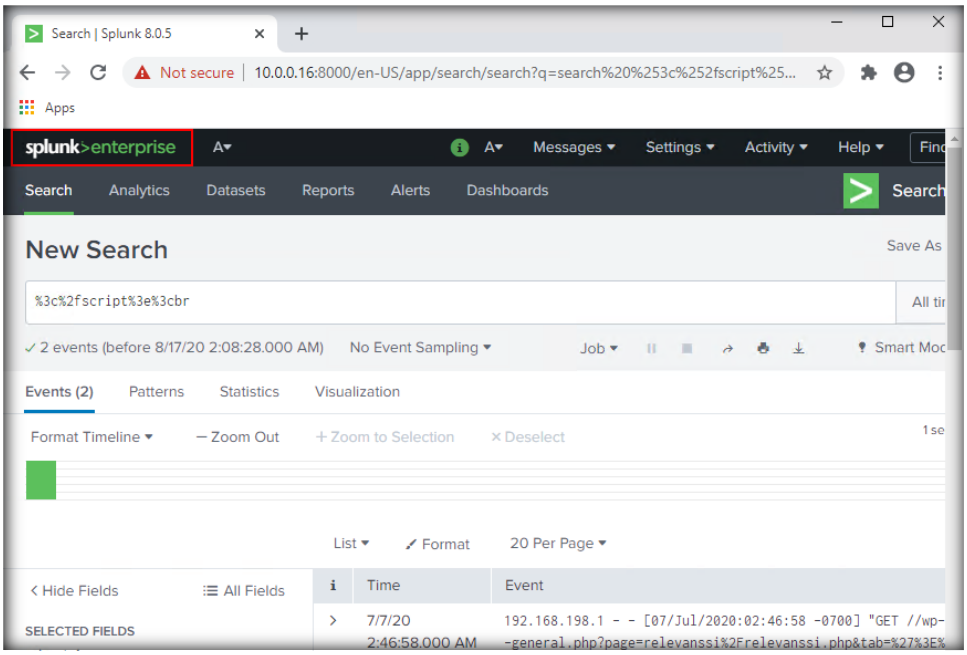

FIGURE 1.21: Click splunk>enterprise

27. You will now be directed to the homepage of **Splunk Enterprise** again. We are now going to examine the **XSS** attack log file from **ModSecurity Logs**. To be able to upload this evidence file, click on the **Add Data** icon on the homepage and then follow the steps meant for uploading it, as you did while uploading the **XSS** evidence file in the previous case.

28. When the **Open** window appears, navigate to **C:\CHFI-Tools\Evidence Files\Log Files\ModSecurity Logs**, select **XSS,** and click **Open** to upload the file.

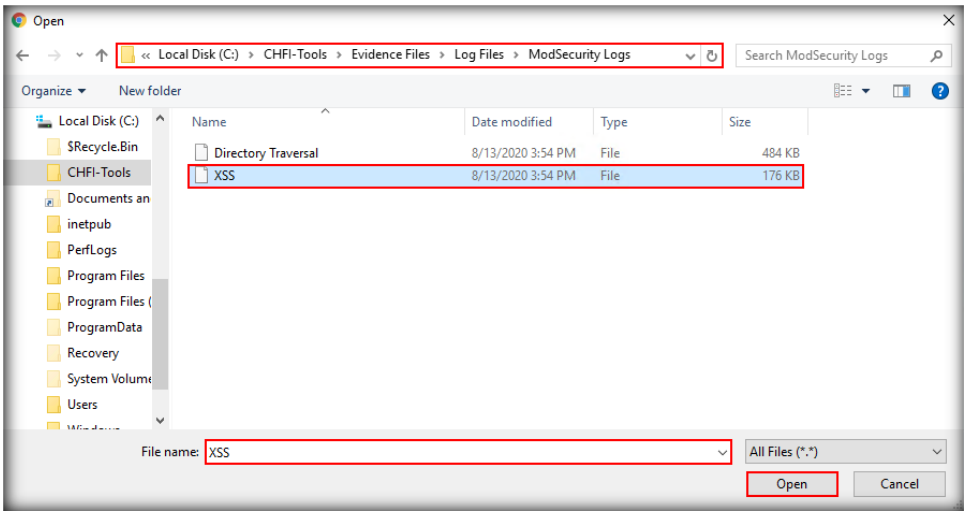

FIGURE 1.22: Select XSS file and click Open

29. The **XSS** file will be uploaded successfully. Click **Next** to continue.

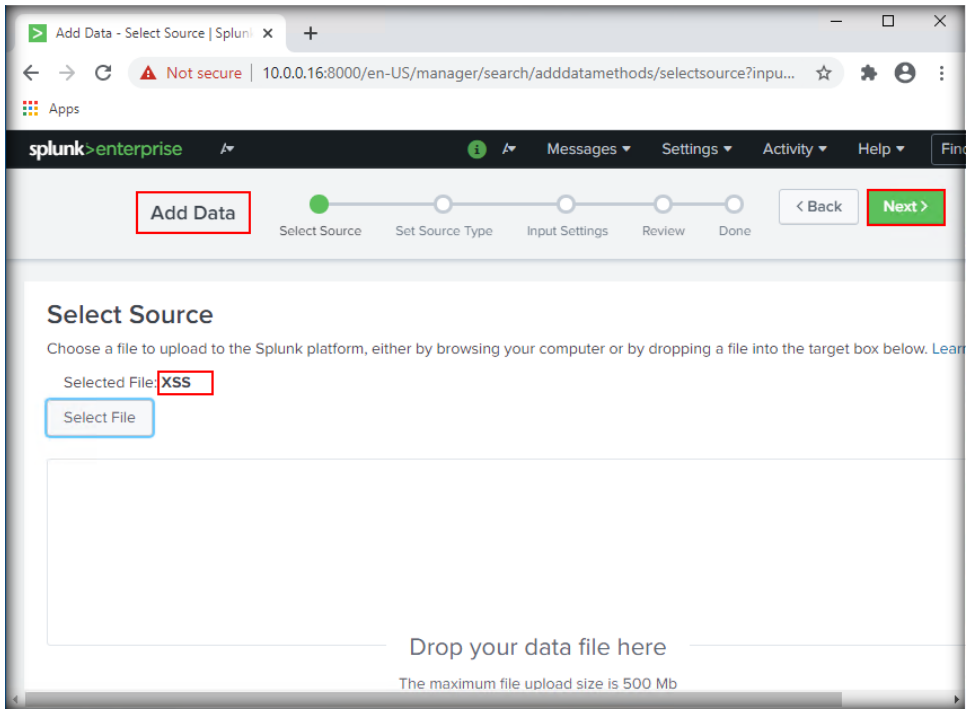

FIGURE 1.23: Click Next

30. In the next step, you will see the **Set Source Type** section. Click the **Save As** button.

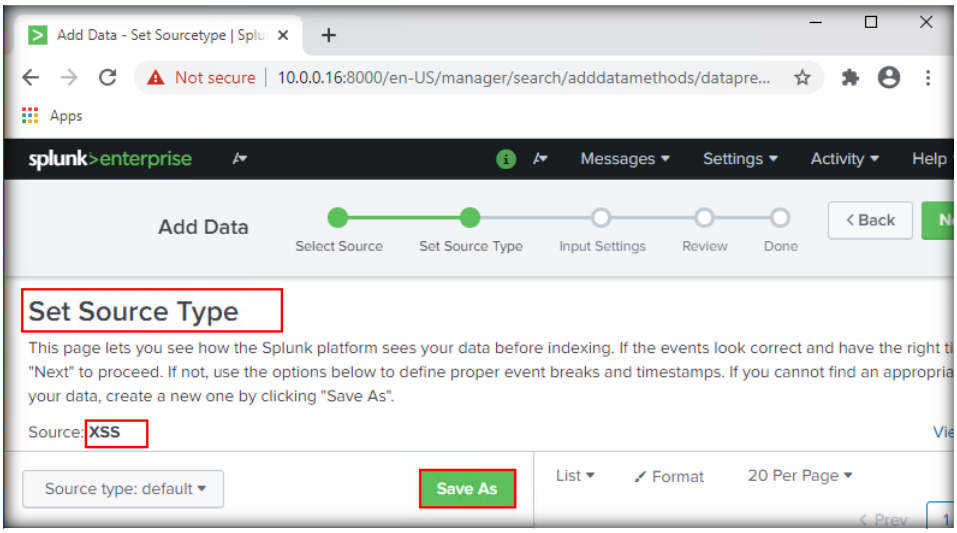

FIGURE 1.24: Click Save As in Set Source Type section

31. The **Save Source Type** window will now appear. In the **Name** field, enter **XSS (ModSecurity Logs)**. Click **Save** to continue.

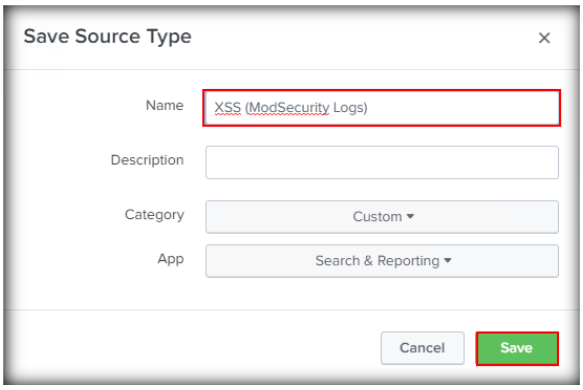

FIGURE 1.25: Enter details in Save Source Type window and click Save

32. You will now get back to the **Set Source Type** section again. Click **Next** to proceed further.

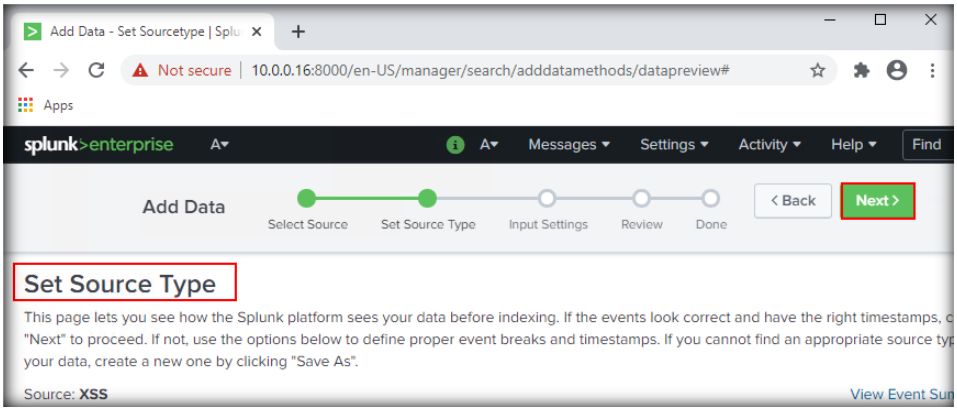

FIGURE 1.26: Click Next

33. In the next step, you will see the **Input Settings** section. Click **Review** to proceed further.

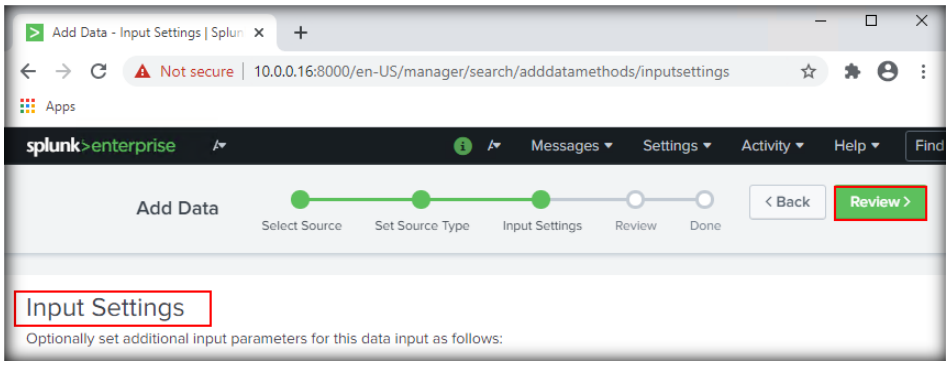

FIGURE 1.27: Click on Review in Input Settings section

34. In the next step, the **Review** section will appear. Ensure that the details under the **Review** section are correct and then click **Submit** to proceed further.

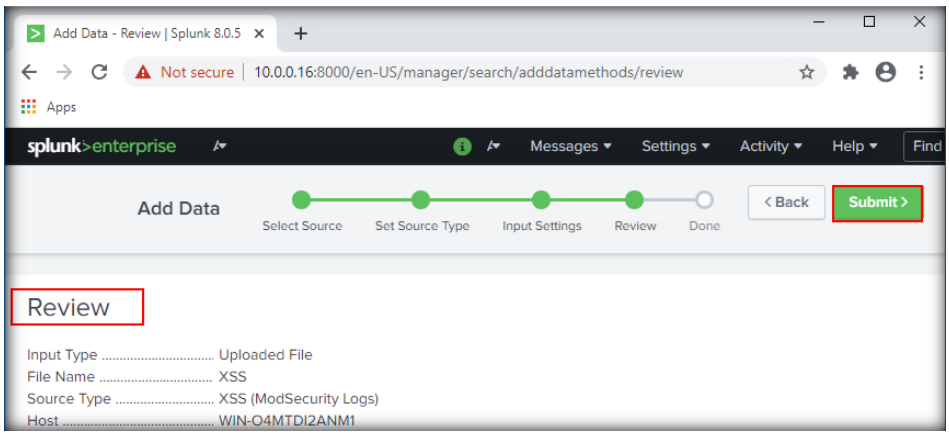

FIGURE 1.28: Review the details and click Submit

35. Upon clicking **Submit**, the application window will display the following message: **File has been uploaded successfully**. Now, click **Start Searching** to fetch the log entries from the log file.

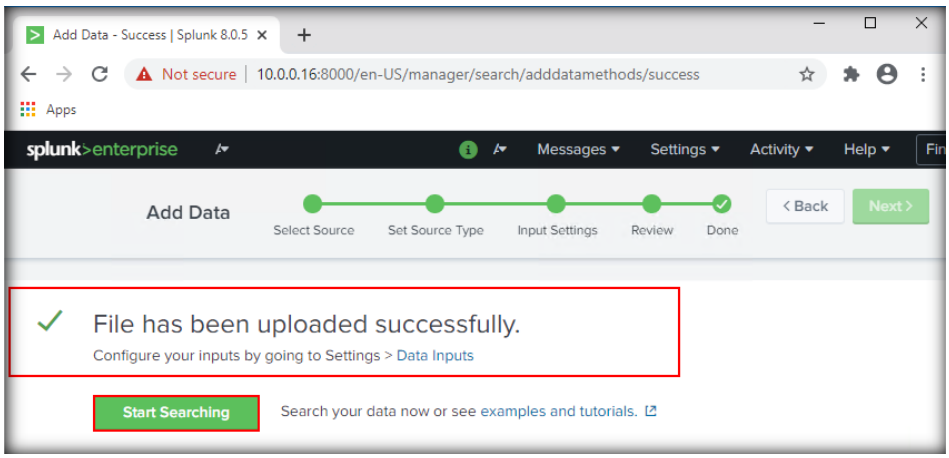

FIGURE 1.29: Click Start Searching

36. All entries from the log file will now show up in **Splunk Enterprise** under the **Events** tab.

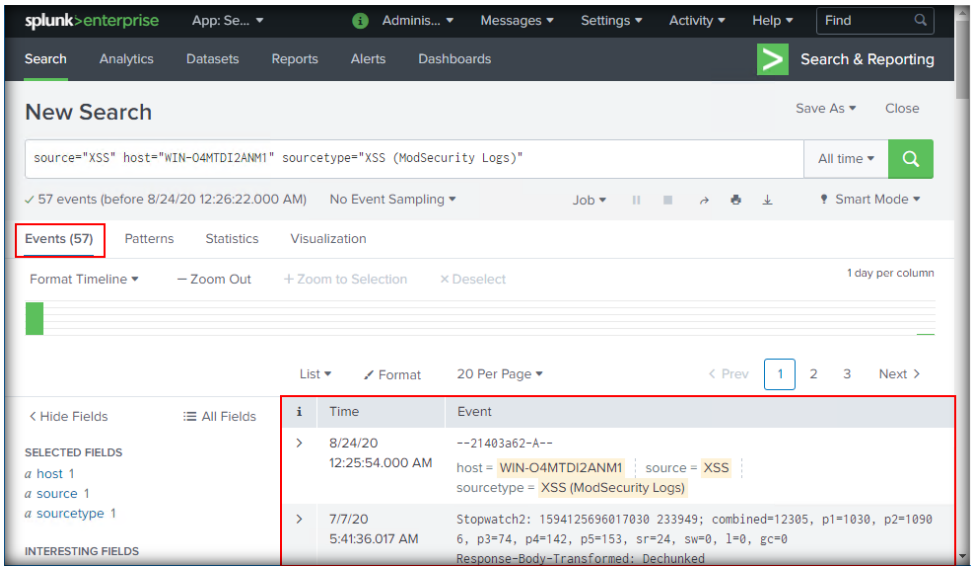

FIGURE 1.30: Log entries displayed in Splunk

37. To examine the log file for **XSS** attack, we will first apply the plain-text filter **<script>**. However, upon applying the plain-text filter, we see that no results are generated.

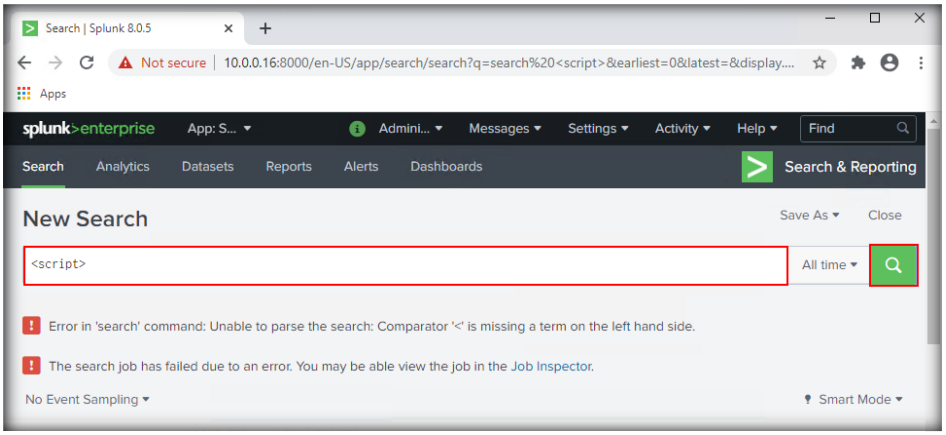

FIGURE 1.31: Use filter <script>

Note: Upon applying the above-mentioned filter, you might view entries from previously sourced log file. In such a case, you need to specify the current log file along with the search filter, for example, **<script> sourcetype="XSS (ModSecurity Logs)"**, to fetch relevant results.

38. Now, we will apply an encoded form of **<script>** filter to look for results related to **XSS** attack. To apply the encoded filter, type **%3c** in the search box. Upon typing **%3c**, we can observe that a search term suggestion relating to **%3c** is displayed through a drop-down which appears under the **New Search** box. The suggestion reads **%3c%2fscript%3e%3cbr**, which is an encoded variant of the **<script>** filter. Select the suggestion and then click the **Search** icon.

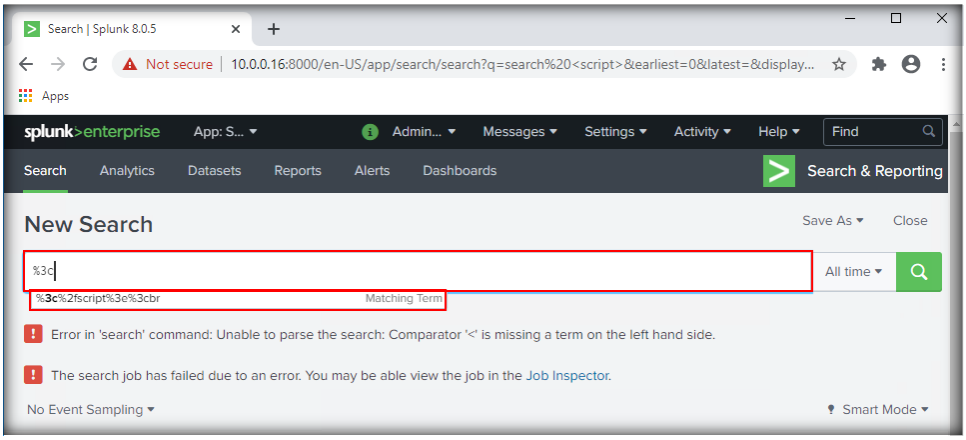

FIGURE 1.32: Enter the encoded filter

Note: Upon applying the above-mentioned filter, you might view entries from previously sourced log file. In such a case, you need to specify the current log file along with the search filter, for example, **%3c%2fscript%3e%3cbr sourcetype="XSS (ModSecurity Logs)"**, to fetch relevant results.

39. Upon applying the encoded filter, we observe that a few log entries are displayed as results under the **Events** tab.

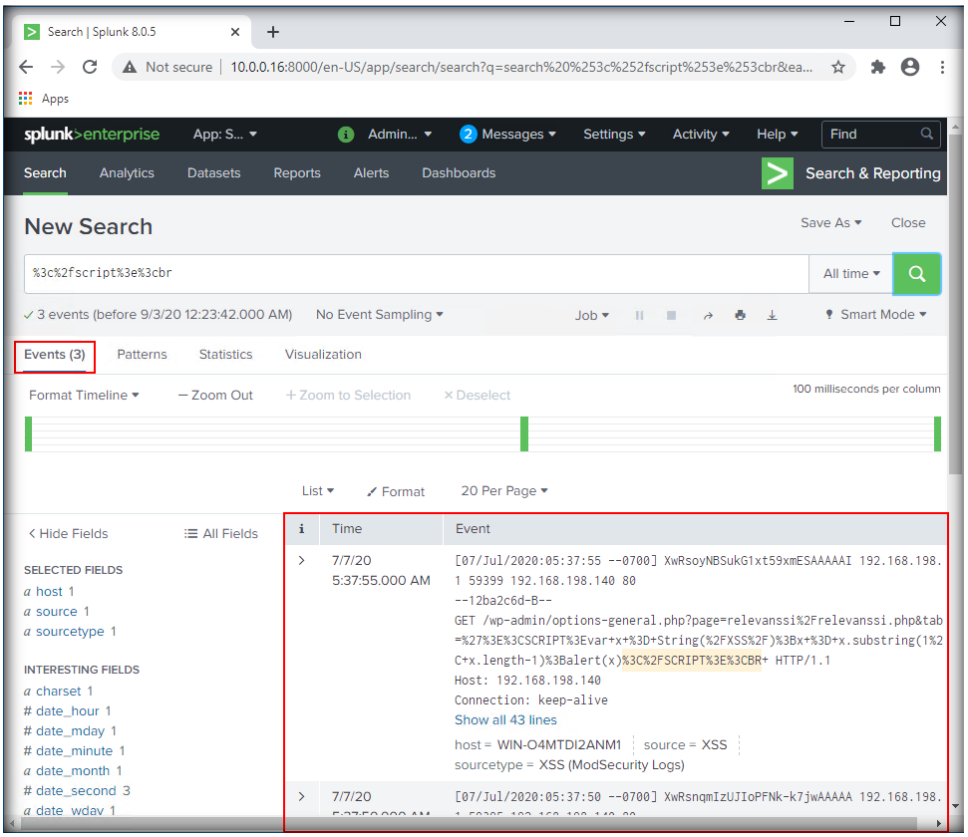

FIGURE 1.33: Results found after using the encoded filter

Note: For the ease of demonstration, we have used a log file containing entries specific only to recorded web attacks. In real-time scenarios, however, you might see several log entries upon applying a filter and you might have to scroll down the window to manually look for the log entry containing indications of an **XSS** attack.

40. Upon examining the result, we notice the presence of an encoded script in the query string. The script reads **%3CSCRIPT%3Evar+x+%3D+String(%2FXSS%2F)%3Bx+%3D+x.substring(1%2C+x.length-1)%3Balert(x)%3C%2FSCRIPT%3E**.

41. When we decode this encoded malicious script with the help of the decoder table provided under the steps for investigating the **XSS** log file from **Apache Logs**, we can similarly see the presence of the following HTML tags in the JavaScript: **<SCRIPT>var x = String(/XSS/);x = x.substring(1, x.length-1);alert(x)</SCRIPT>**. Upon observing the presence of encoded HTML tags, we can infer that the script is malicious and has been used for an **XSS** attack.

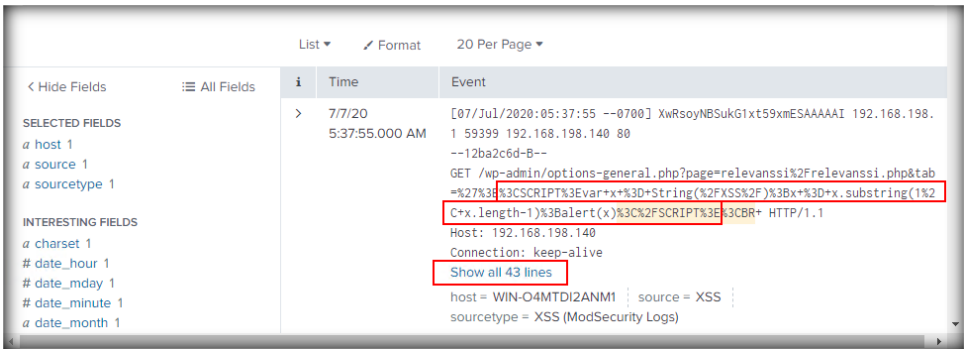

FIGURE 1.34: Malicious Script related to XSS attack

Note: You must click on the **Show all 43 lines** option to be able to view all components of the log entry, as indicated in the screenshot above. The log contents that appear upon clicking the **Show all 43 lines** option are described in the subsequent step.

42. A detailed observation of the contents in the log entry provides us with the following findings, as marked in the screenshots below:

    A. Date and time of the attack: **07th July 2020** and **05:37:55 AM**

    B. IP address of the attacker: **192.168.198.1**

    C. The webpage targeted by the attacker using **GET** request: **/wp-admin/options-general.php?** (this indicates that it was an attack on the webpage of a **WordPress** site)

    D. Malicious script used in the query string: **%3CSCRIPT%3Evar+x+%3D+String(%2FXSS%2F)%3Bx+%3D+x.substring(1%2C+x.length-1)%3Balert(x)%3C2FSCRIPT%3E**

    E. IP Address of the Host: **192.168.198.140**

F. HTTP **403** Forbidden message: This indicates the access to requested resource has been blocked by **ModSecurity** firewall.

G. Message generated by server in response to attacker's GET request: **You don't have permission to access this resource**.

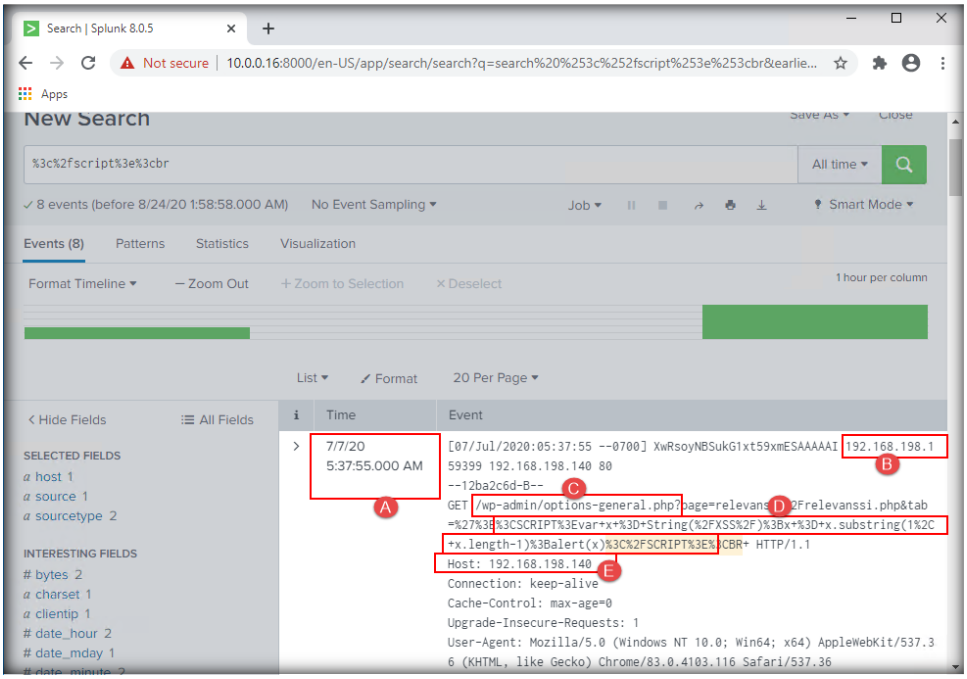

FIGURE 1.35: Detailed analysis of XSS attack

Note: Scroll down the application window to find the observations **F** and **G**, as indicated in the screenshot below:

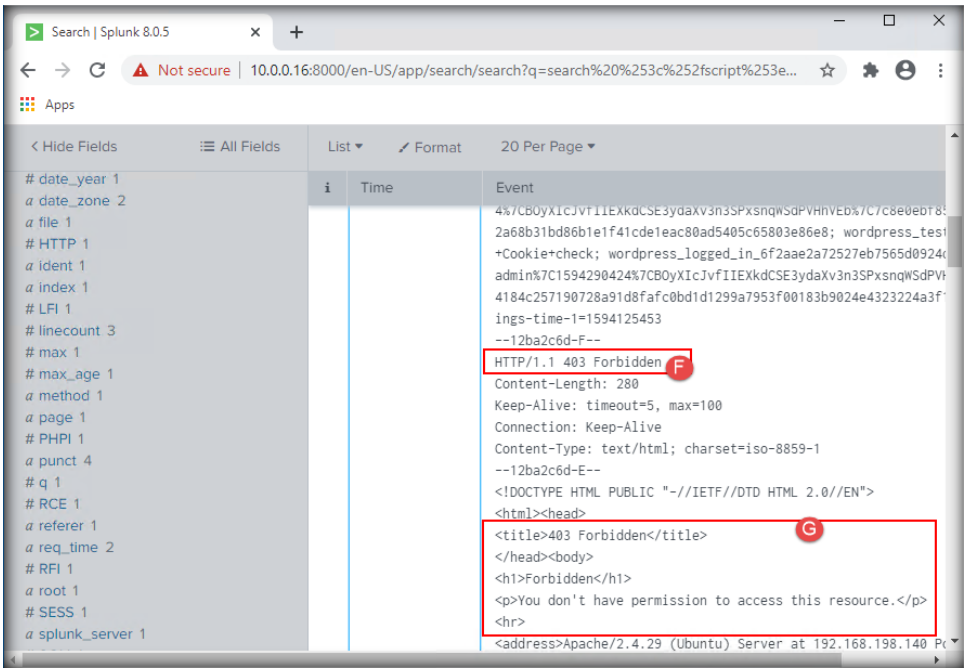

FIGURE 1.36: Access Forbidden message

43. We will now examine the **XSS** attack log file generated by the **IIS** server.
44. Click **splunk>enterprise** at the top left of the **Splunk Enterprise** webpage.

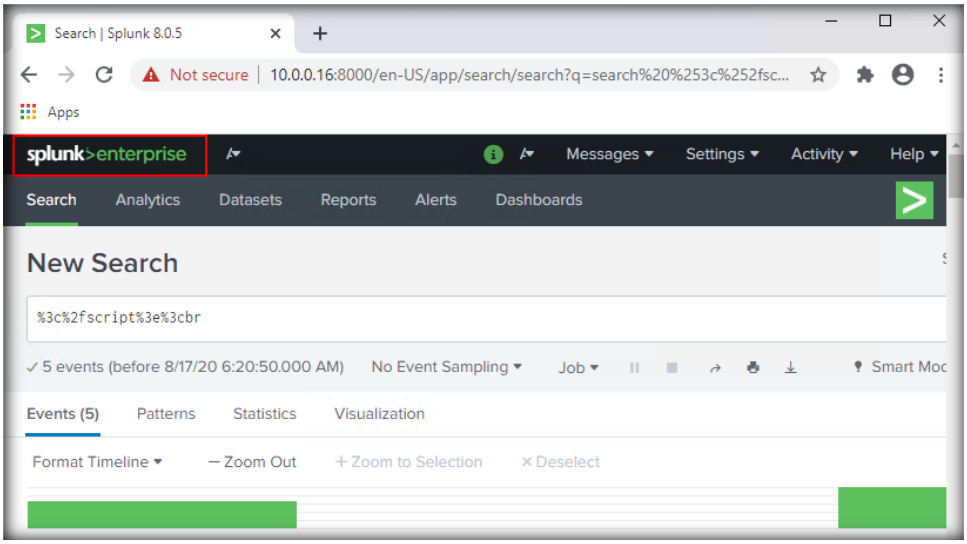

FIGURE 1.37: Click splunk>enterprise

45. You will be directed to the homepage of **Splunk Enterprise**. We are now going to examine the **XSS.txt** log file from **IIS Logs**. To be able to upload this evidence file, click on the **Add Data** icon on the homepage and then follow the steps meant for uploading it, as you did in previous cases.
46. When the **Open** window appears, navigate to **C:\CHFI-Tools\Evidence Files\Log Files\IIS Logs** and select **XSS.txt**. Click **Open** to upload the file.

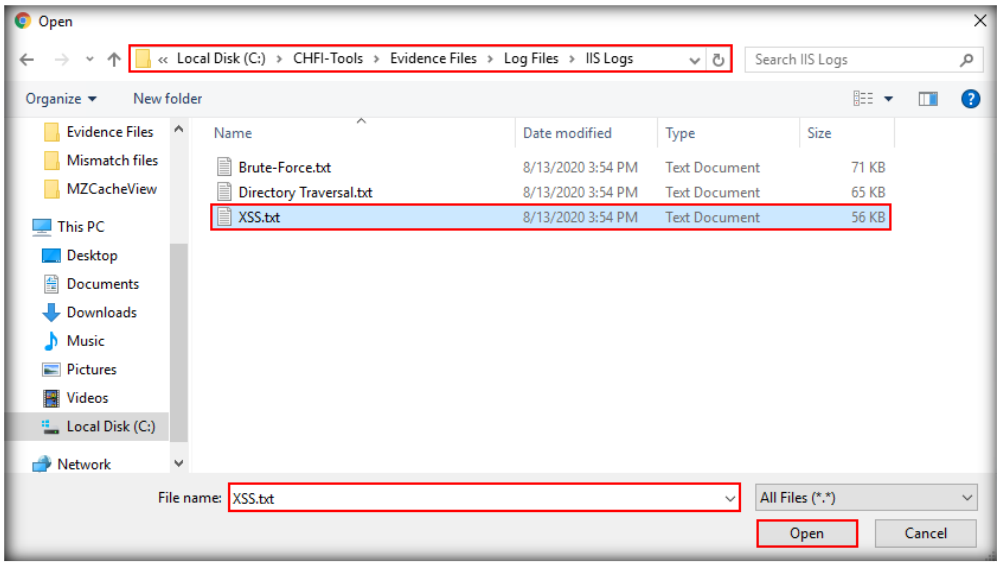

FIGURE 1.38: Select XSS.txt and click Open

47. The **XSS.txt** file will be uploaded successfully. Click **Next** to continue.

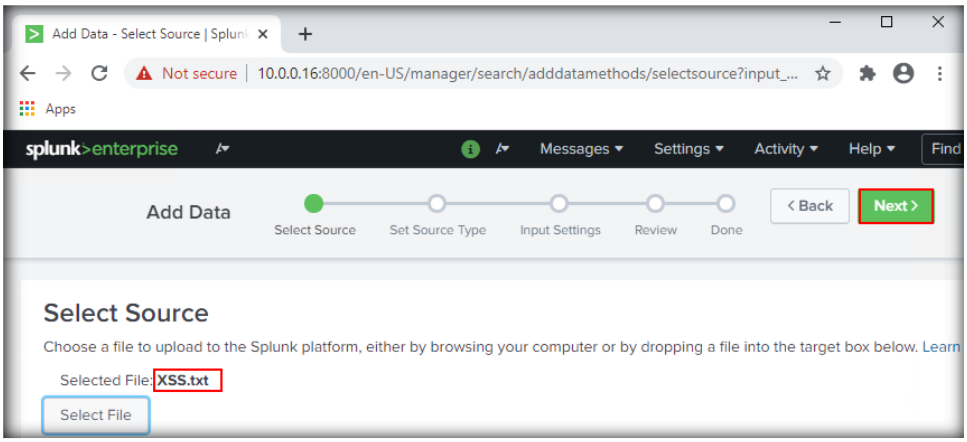

FIGURE 1.39: Click Next

48. In the next step, you will see the **Set Source Type** section. click **Save As**, as shown in the following screenshot:

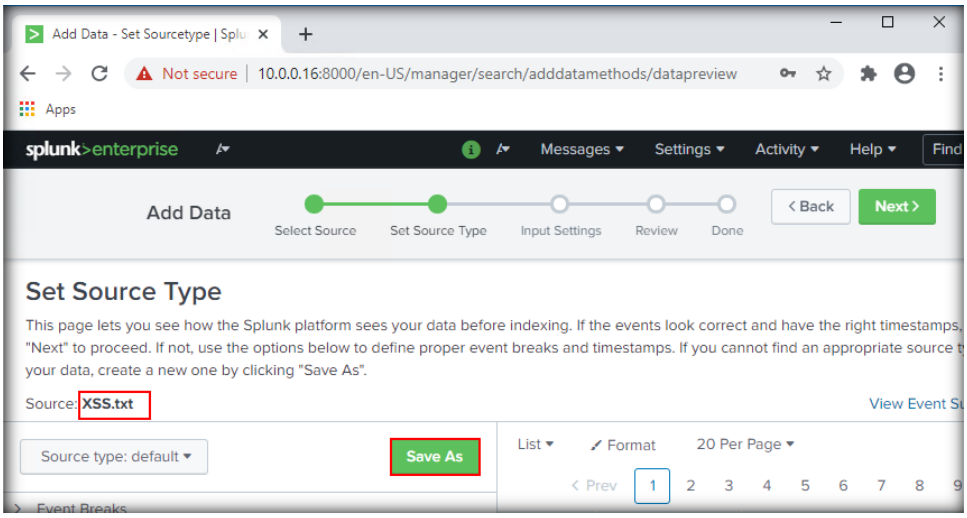

FIGURE 1.40: Click Save As in Set Source Type section

49. The **Save Source Type** window will appear now. In the **Name** field, enter **XSS.txt (IIS Logs)**. Then, click **Save** to continue.

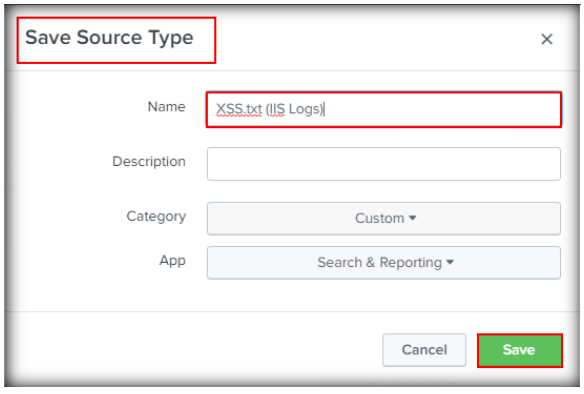

FIGURE 1.41: Enter details in Save Source Type window and click Save

50. You will now get back to the **Set Source Type** section again. Click **Next** at the top right of the page to proceed further.

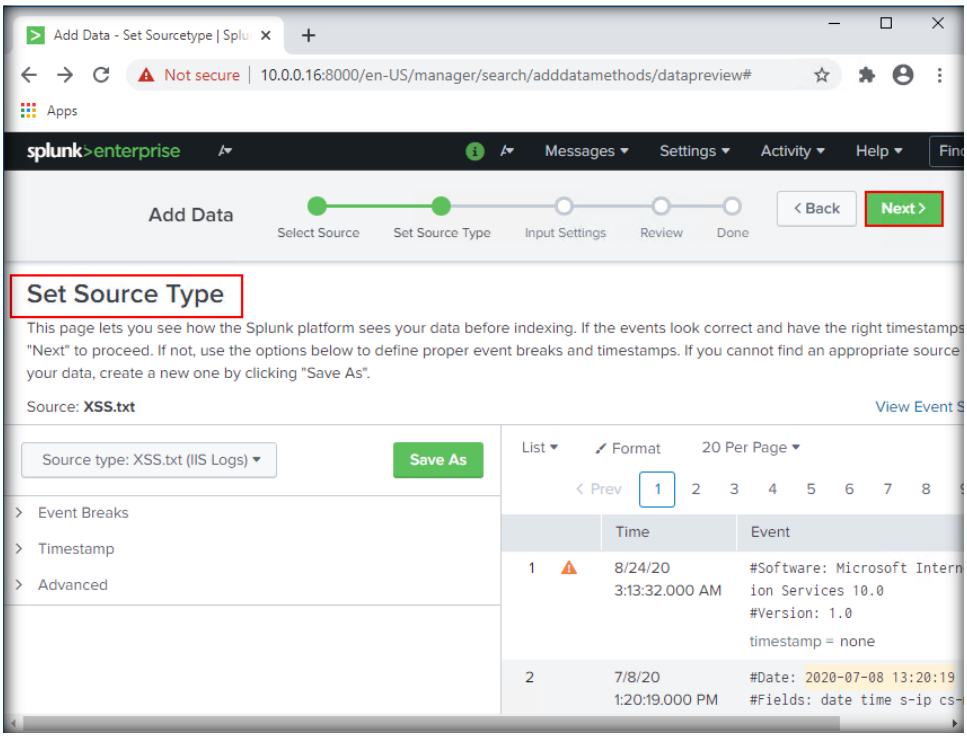

FIGURE 1.42: Click Next in Set Source Type section

51. In the next step, you will see the **Input Settings** section. Click **Review** at the top right of the page to proceed further.

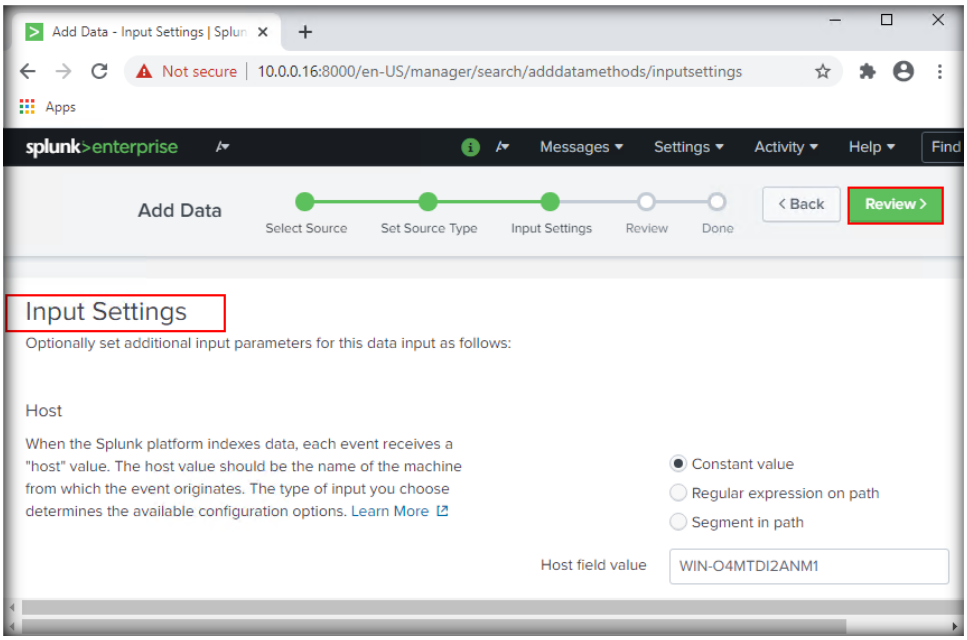

FIGURE 1.43: Click on Review in the Input Settings section

52. In the next step, the **Review** section will appear. Ensure that the details under the **Review** section are correct and then click **Submit** to proceed further.

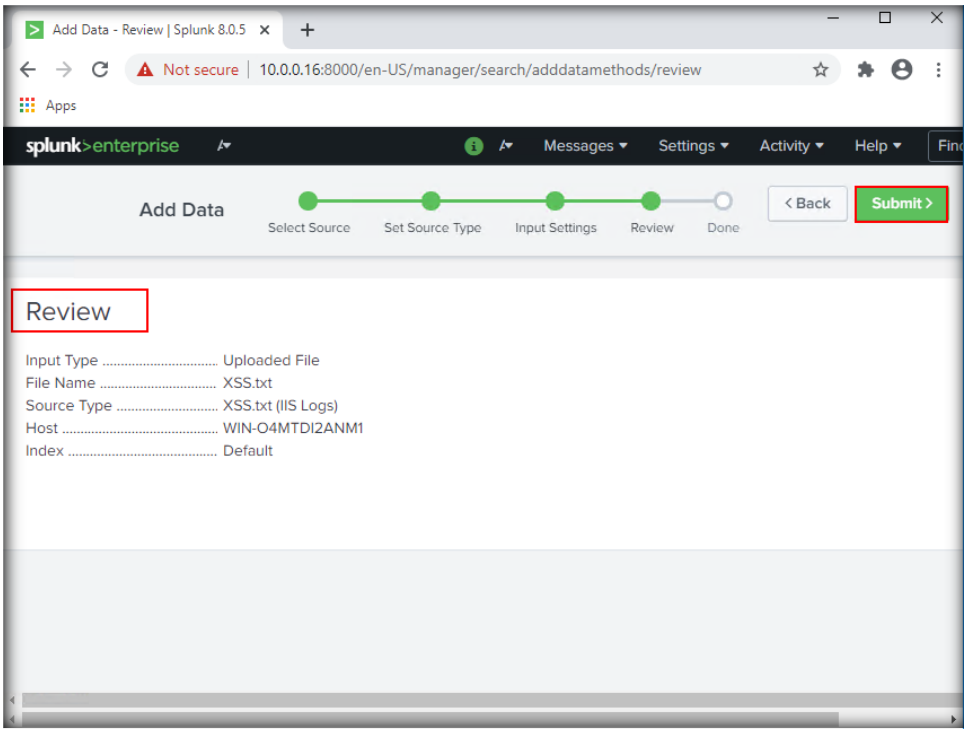

FIGURE 1.44: Review the details and click Submit

53. Upon clicking **Submit**, the application window will display the following message: **File has been uploaded successfully**. Now, click **Start Searching** to fetch all log entries from the log file.

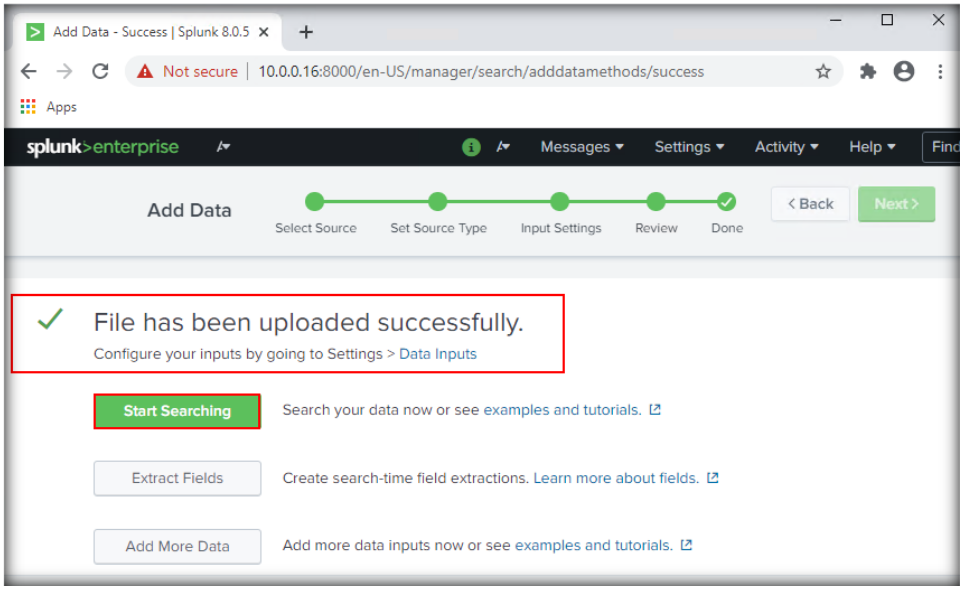

FIGURE 1.45: Click Start Searching

54. The uploaded log file will now be opened in **Splunk Enterprise** and its entries will be displayed under the **Events** tab, as shown in the screenshot below:

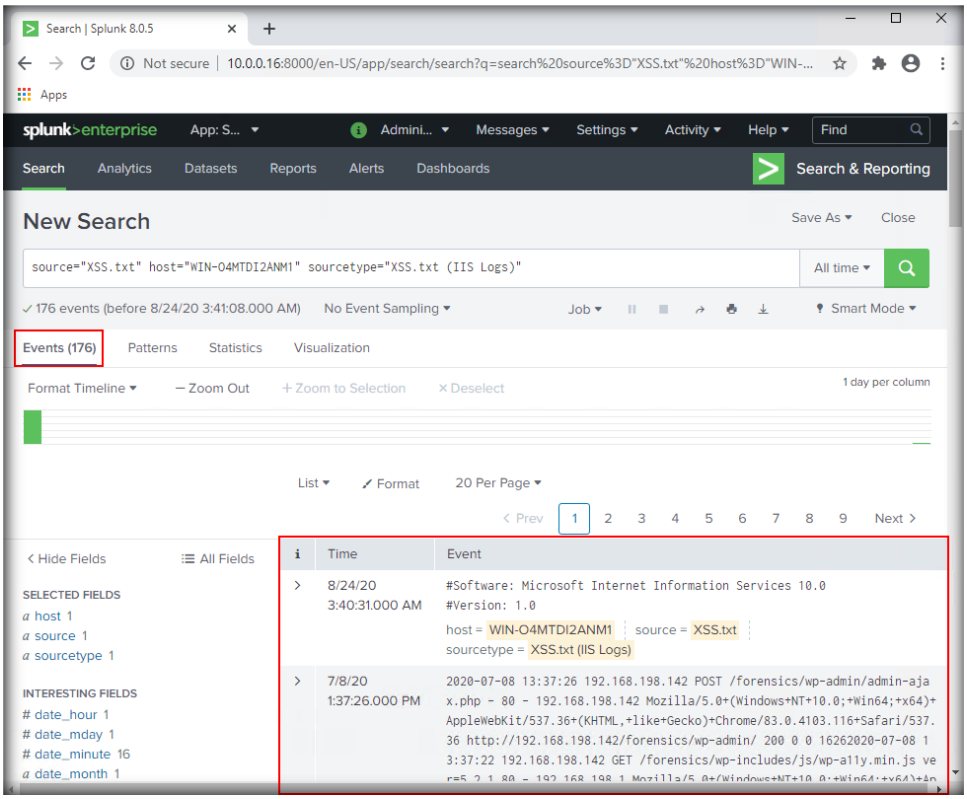

FIGURE 1.46: Log entries displayed in Splunk

55. To examine the log file for **XSS** attacks, we will first apply the plain-text filter **<script>**. However, upon applying the filter, we see that no results are generated.

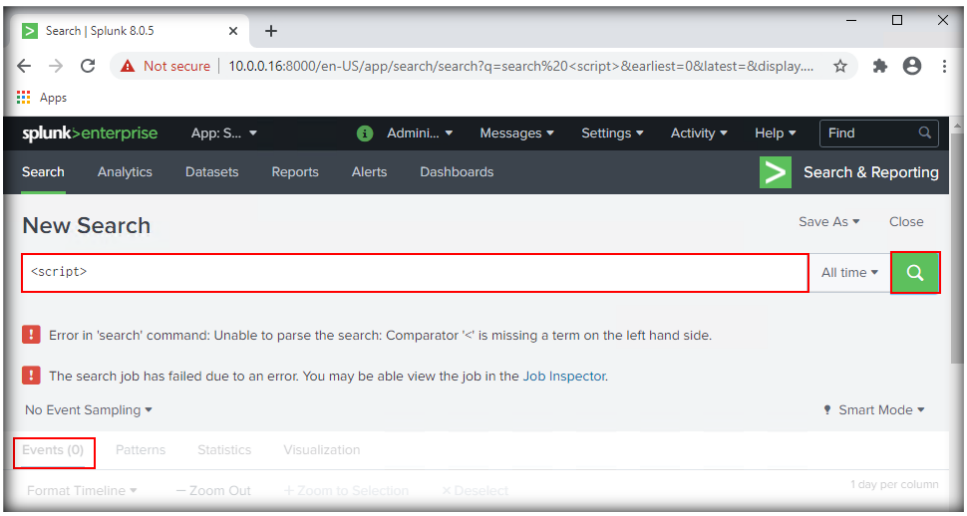

FIGURE 1.47: Use filter <script>

Note: Upon applying the above-mentioned filter, you might view entries from previously sourced log files. In such a case, you need to specify the current log

file along with the search filter, for example, **<script> sourcetype="XSS.txt (IIS Logs)"**, to fetch relevant results.

56. We will now apply an encoded variant of **<script>** filter to look for the results related to an **XSS** attack.

57. To apply the encoded filter, type **%3c** in the **New Search** box. Upon typing **%3c**, we can observe that a search term suggestion relating to **%3c**, which reads **%3c%2fscript%3e%3cbr**, in the drop-down. This is an encoded variant of the **<Script>** filter. Select the suggestion and then click the **Search** icon.

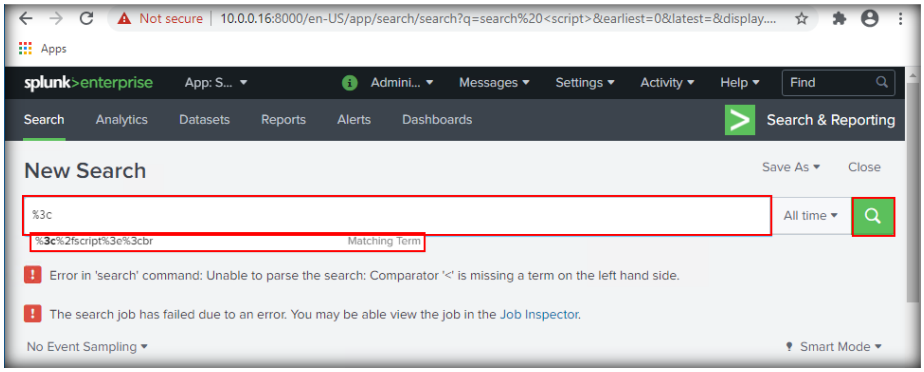

FIGURE 1.48: Select the encoded filter

Note: Upon applying the above-mentioned filter, you might view entries from previously sourced log files. In such a case, you need to specify the current log file along with the search filter, for example, **%3c%2fscript%3e%3cbr sourcetype="XSS.txt (IIS Logs)"**, to fetch relevant results.

58. Upon applying the encoded filter, the application displays several log entries as results. We need to examine these results for indicators of an **XSS** attack.

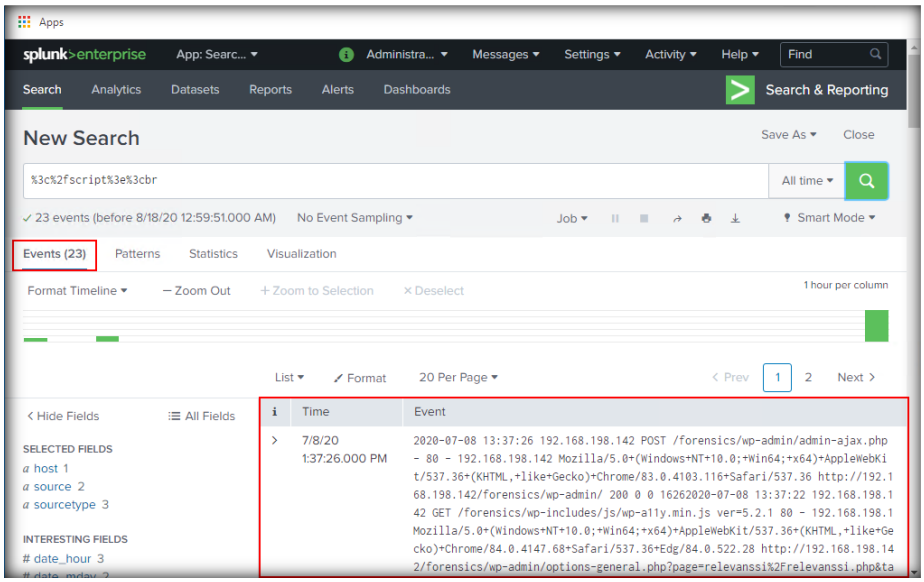

FIGURE 1.49: Results obtained using encoded filter

Note: For the ease of demonstration, we have used a log file containing entries specific only to recorded web attacks. In real-time scenarios, however, you might see several log entries upon applying a filter and you might have to scroll down the window to manually look for the log entry containing the signs of an **XSS** attack.

59. Upon examining the log entry at the top, we can see that its query string contains a malicious encoded script. The malicious encoded script reads as **%3CSCRIPT%3Evar+x+%3D+String(%2FXSS%2F)%3Bx+%3D+x.substring(1%2C+x.length-1)%3Balert(x)%3C%2FSCRIPT%3E**. Based on our previous observations, this is the same malicious script we had found when investigating the **XSS** log files from **Apache** server and **ModSecurity** firewall.

60. A detailed observation of the log entry provides us with the following findings:

    A. Date and time of the attack: **08th July 2020** and **01:37:26 PM**

    B. IP address of the attacker: **192.168.198.1**

    C. The webpage targeted in the attack: **http://192.168.198.142/forensics/wp-admin/options-general.php?** (this indicates it was an attack on the webpage of a **WordPress** site)

    D. Malicious script used in the attack: **%3CSCRIPT%3Evar+x+%3D+String(%2FXSS%2F)%3Bx+%3D+x.substring(1%2C+x.length-1)%3Balert(x)%3C%2FSCRIPT%3E**

    E. HTTP status code **200**: This means the request made to the server has been received and is being processed.

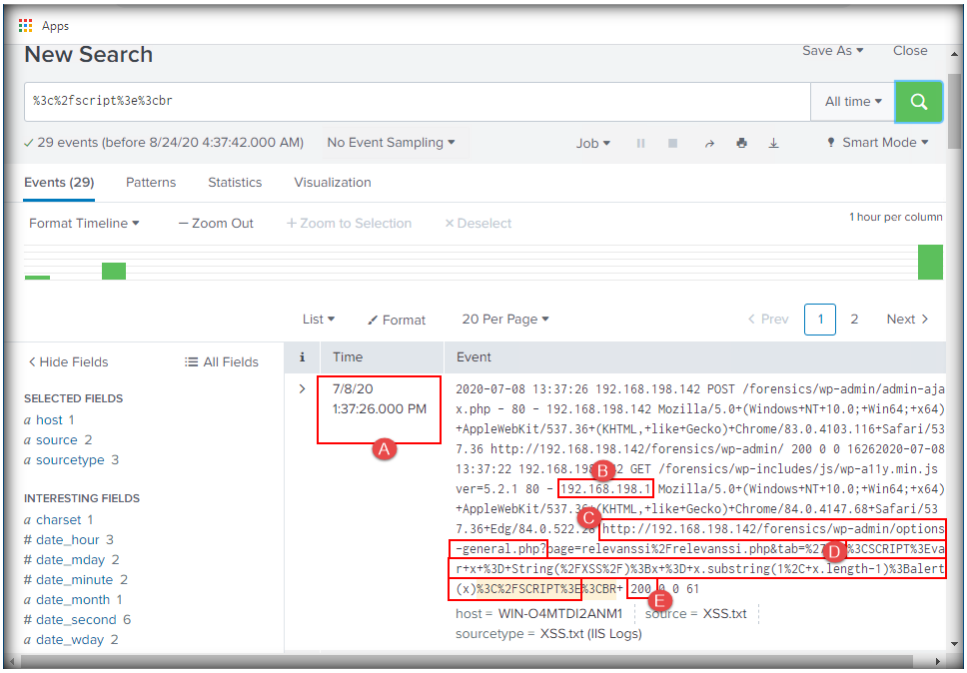

FIGURE 1.50: Detailed analysis of XSS attack (IIS)

TASK 3

Detect and Examine SQL Injection Attack via Log Files

61. We will now detect and analyze **SQL Injection** attack by examining log files.
62. Click **splunk>enterprise** at the top left of the webpage.

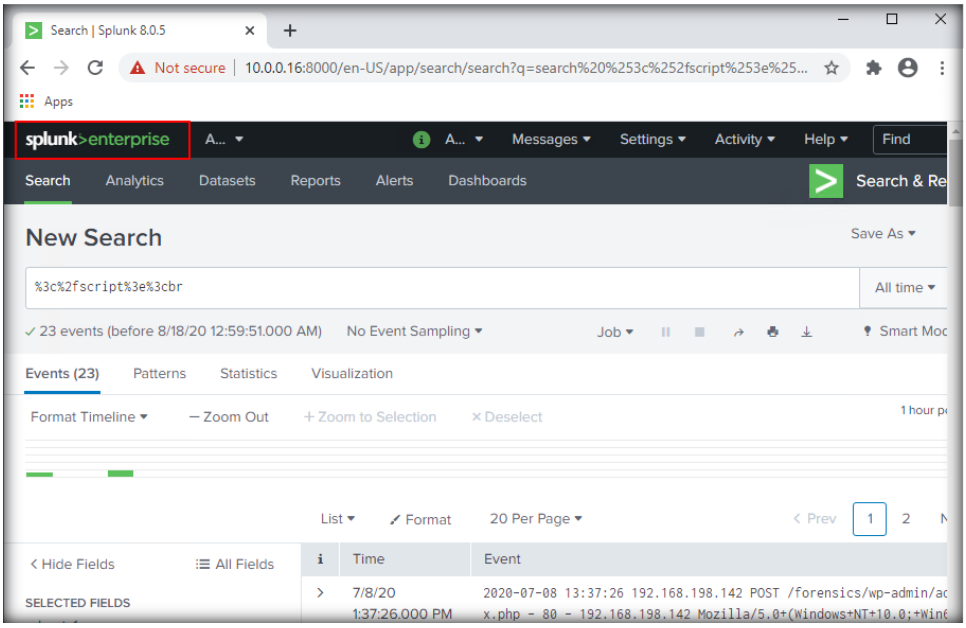

FIGURE 1.51: Click splunk>enterprise

63. You will now be taken to the **Splunk Enterprise** homepage. We now need to upload the **SQL Injection** log file. To be able to upload the evidence file, click on the **Add Data** icon on the homepage and then follow the steps for uploading it.
64. On the **Open** window that appears, navigate to **C:\CHFI-Tools\Evidence Files\Log Files\Apache Logs**, select **SQL Injection.log**, and click **Open** to upload the file.

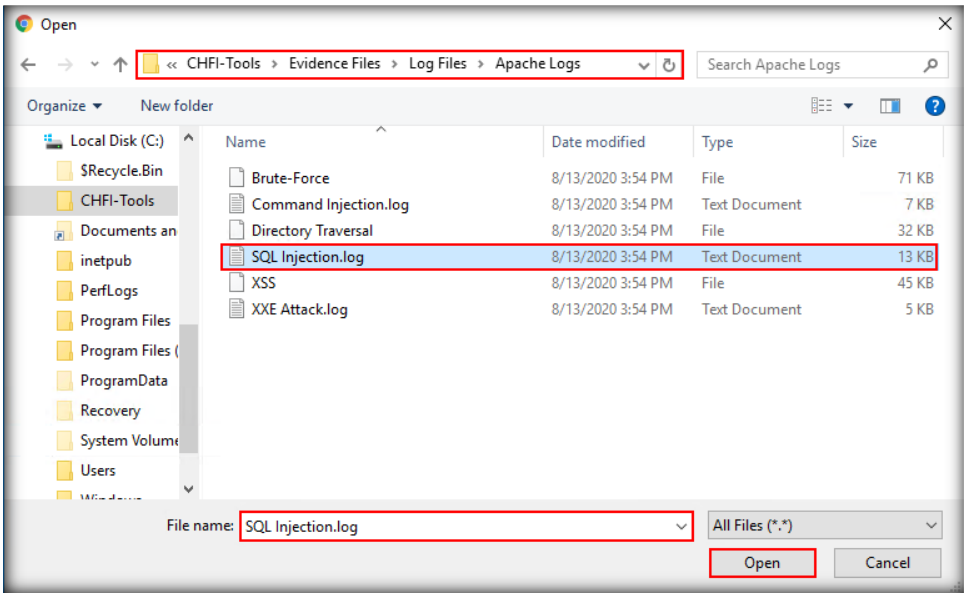

FIGURE 1.52: Select SQL Injection.log and click Open

65. The **SQL Injection.log** file will be uploaded successfully. Click **Next** to proceed further.

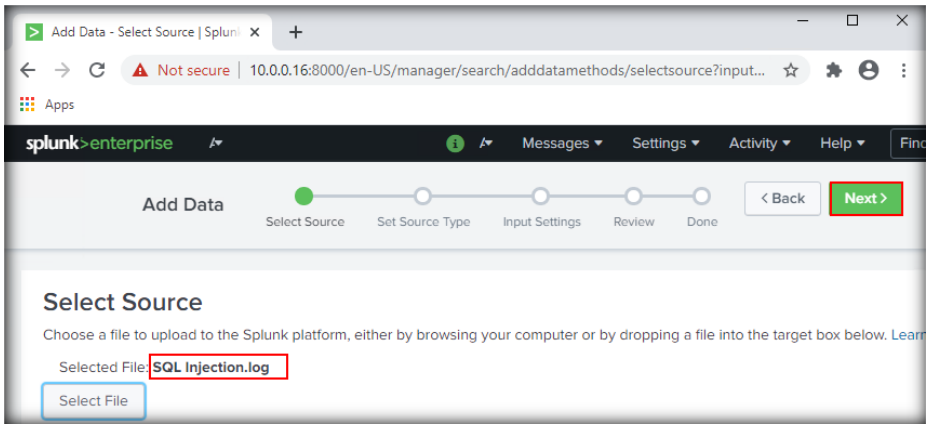

FIGURE 1.53: Click Next

66. In next step, the **Set Source Type** section will appear. Click the **Save As** button.

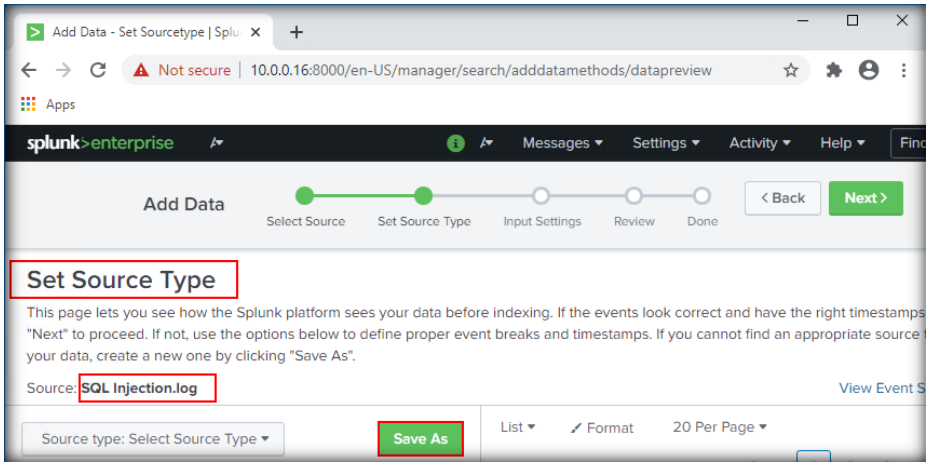

FIGURE 1.54: Click Save As in Set Source Type section

67. The **Save Source Type** window will now appear. In the **Name** field, enter **SQL Injection.log (Apache Logs)** as the source file name, as highlighted in the screenshot below. After entering the name, we need to click **Save** to proceed further.

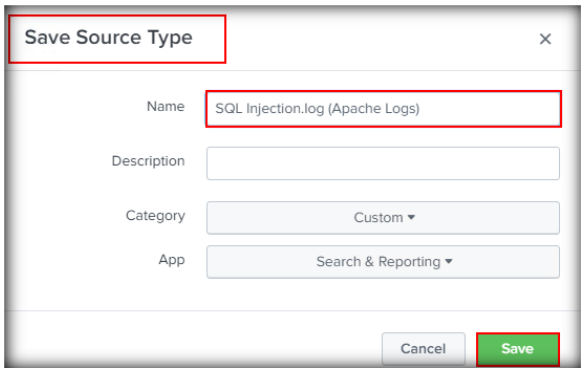

FIGURE 1.55: Enter details in Save Source Type window and click Save

68. You will now get back to the **Set Source Type** section again. Click **Next** at the top right of the page to proceed further.

69. In the next step, you will see the **Input Settings** section. Click **Review** at the top right of the page to proceed further.

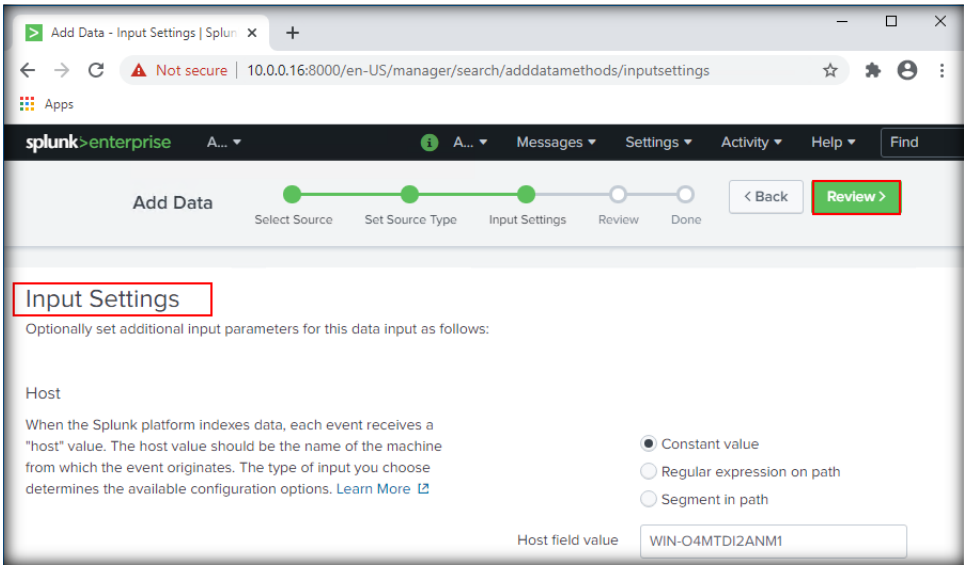

FIGURE 1.56: Click on Review in the Input Settings section

70. In the next step, you will see the **Review** section. Ensure that the details under the **Review** section are correct and then click **Submit** to proceed further. If necessary, apply corrections to the details by revisiting the previous sections by clicking on **Back**, and then come back again to the **Review** section and click **Submit** to proceed further.

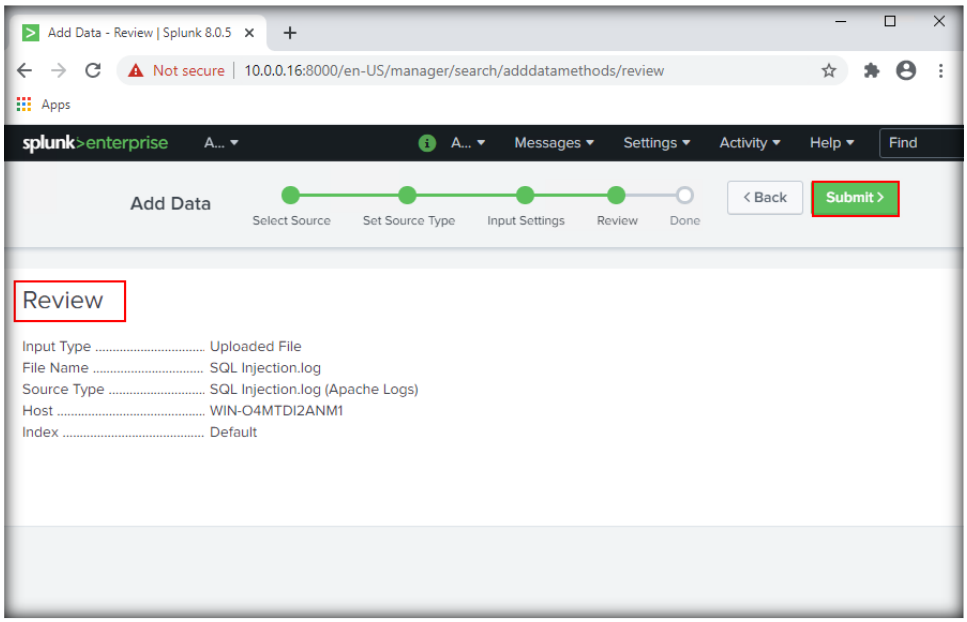

FIGURE 1.57: Review the details and click Submit

71. Upon clicking **Submit**, the application window will display the following message: **File has been uploaded successfully**. Now, click **Start Searching** to fetch the log entries from the **SQL Injection.log** file.

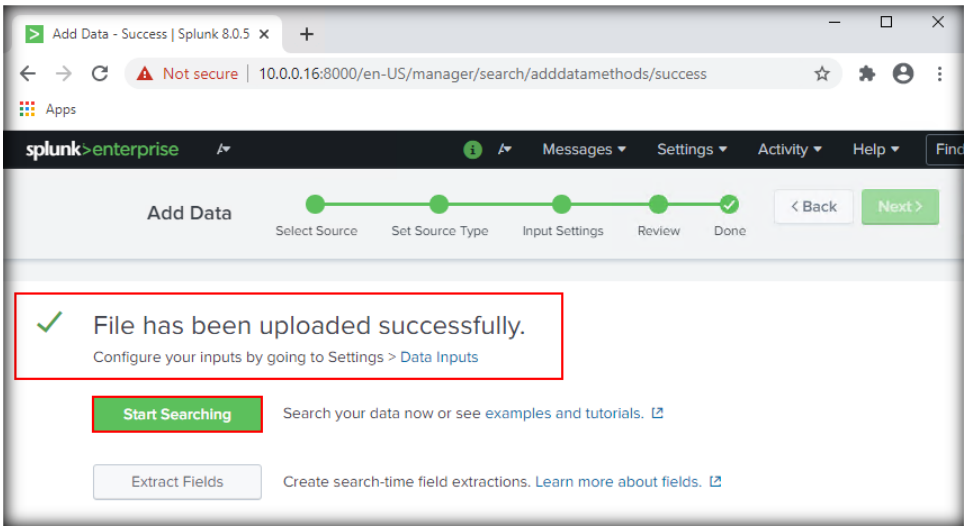

FIGURE 1.58: Click Start Searching

72. You will now see the entries from the uploaded log file displayed in **Splunk Enterprise** under the **Events** tab

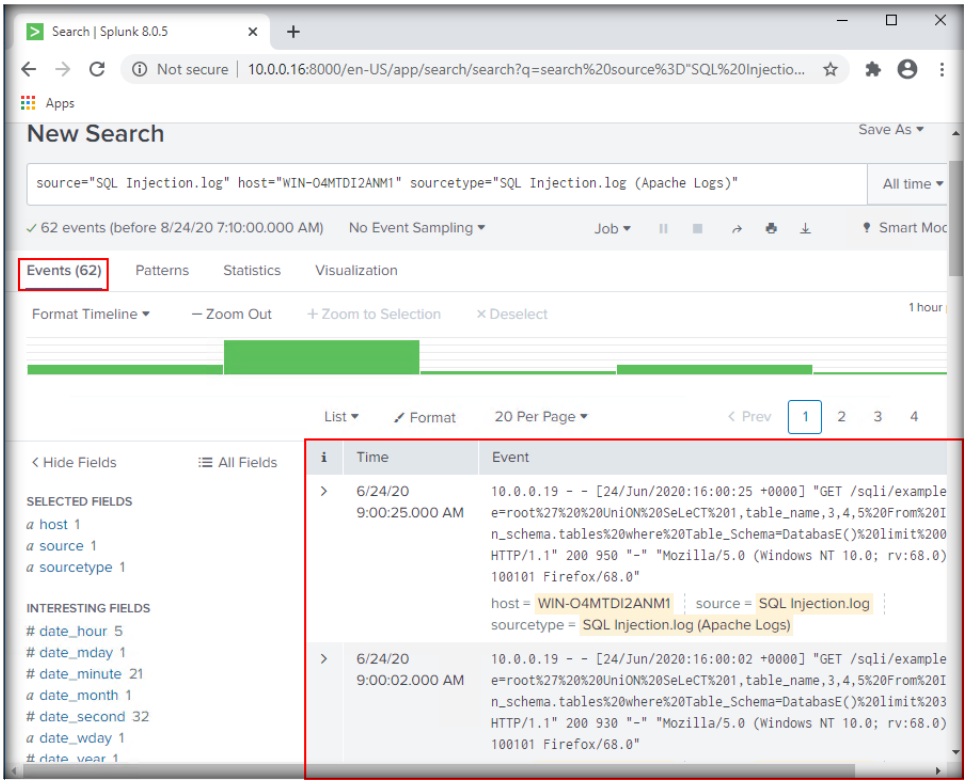

FIGURE 1.59: Log file entries displayed in Splunk

73. To look for an **SQL Injection** attack, we will first apply the plain-text filter **‘ or 1=1--**. To apply the filter, type **‘ or 1=1--** in the **New Search** box and then click on the **Search** icon, as shown in the screenshot below. Upon applying the filter, however, we find that no results are generated.

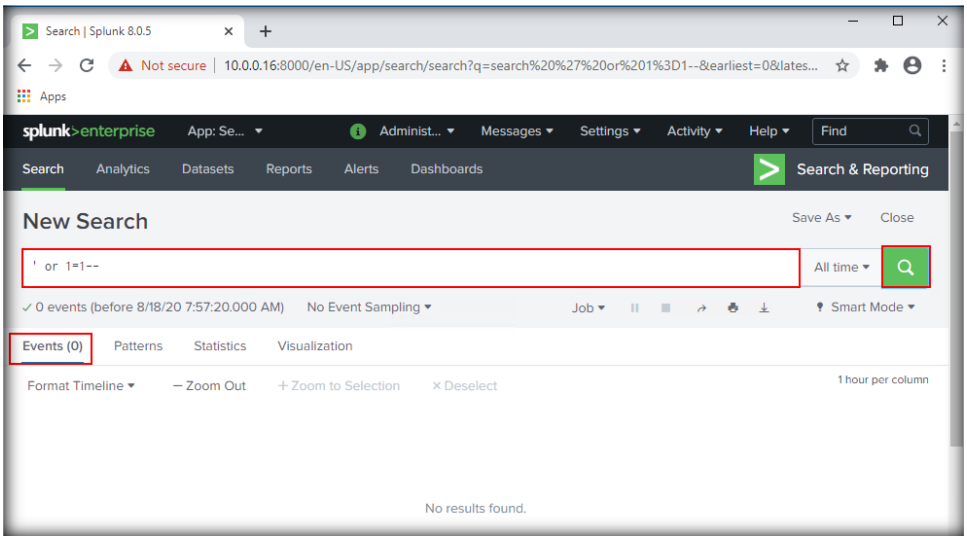

FIGURE 1.60: Use filter ‘ or 1=1--

Note: Upon applying the above-mentioned filter, you might view entries from previously sourced log files. In such a case, you need to specify the current log file along with the search filter, for example, **‘ or 1=1-- sourcetype="SQL Injection.log (Apache Logs)"**, to fetch relevant results.

74. We will now apply the encoded variant of the **‘ or 1=1--** filter. To apply the encoded variant of this filter, type **%27%20or%201%3D1%2D%2D** in the **New Search** box and then click the **Search** icon. We see that no results are obtained even after applying the encoded variant of the **‘ or 1=1--** filter.

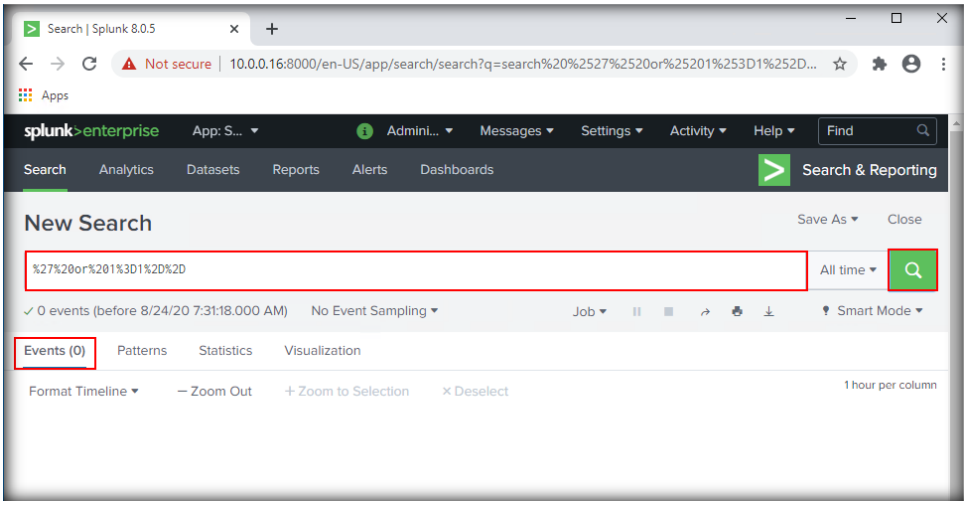

FIGURE 1.61: Apply encoded filter

Note: Upon applying the above-mentioned filter, you might view entries from previously sourced log files. In such a case, you need to specify the current log file along with the search filter, for example, **%27%20or%201%3D1%2D%2D sourcetype="SQL Injection.log (Apache Logs)"**, to fetch relevant results.

75. We will now apply the filter **root' union select**. We find that no results are generated upon the application of this filter.

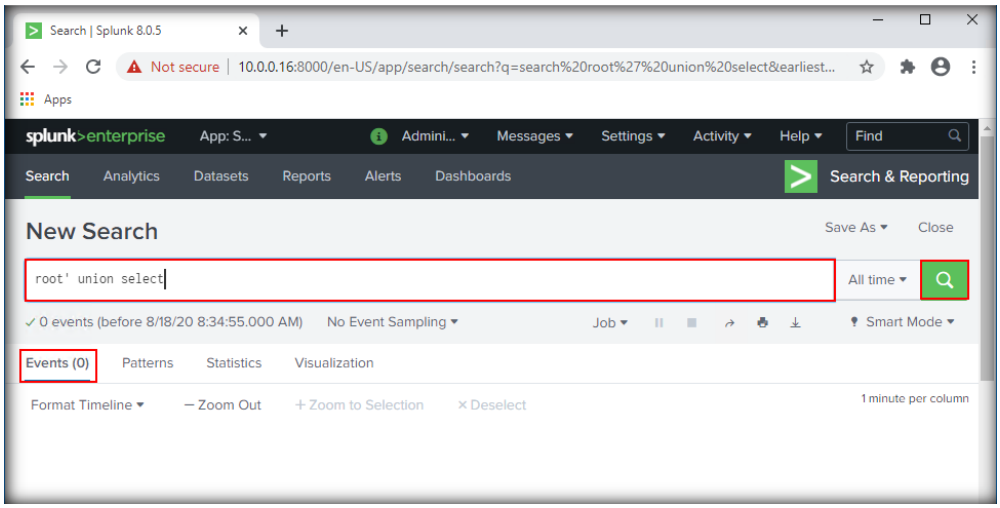

FIGURE 1.62: Use root' union select filter

Note: Upon applying the above-mentioned filter, you might view entries from previously sourced log files. In such a case, you need to specify the current log file along with the search filter, for example, **root' union select sourcetype="SQL Injection.log (Apache Logs)"**, to fetch relevant results.

76. Now, we will apply an encoded variant of the **root' Union Select** filter to look for the signs of an **SQL Injection** attack. To apply the encoded filter, type **root%27%20%20union%20select** in the **New Search** box and then click on the **Search** icon.

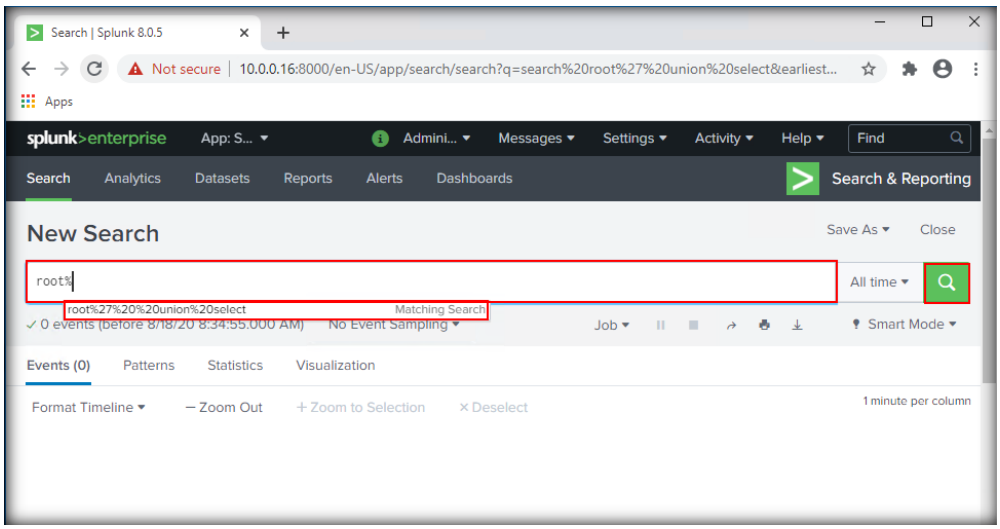

FIGURE 1.63: Enter encoded filter

Note: Upon applying the above-mentioned filter, you might view entries from previously sourced log files. In such a case, you need to specify the current log file along with the search filter, for example, **root%27%20%20union%20select sourcetype="SQL Injection.log (Apache Logs)"**, to fetch relevant results.

77. Upon applying the encoded filter, we see several results generated.

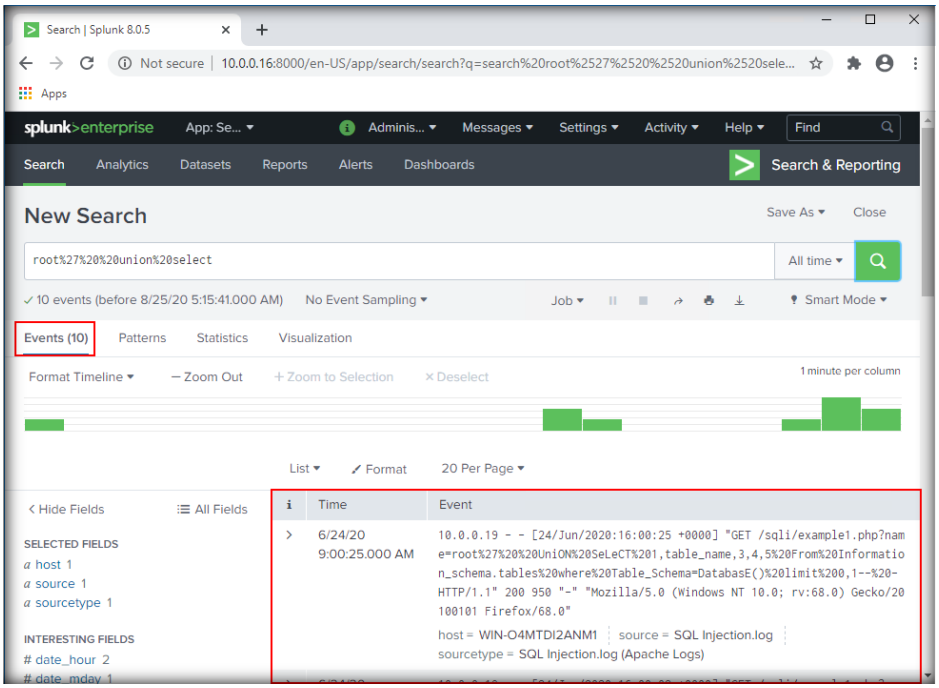

FIGURE 1.64: Results obtained using encoded filter

Note: For the ease of demonstration, we have used a log file containing entries specific only to recorded web attacks. In real-time scenarios, however, you might see several log entries upon applying a filter and you might have to scroll down the window to manually look for the log entry containing the signs of an **SQL Injection** attack.

78. Upon examining the log entry at the top, we find that it contains a **GET** request with a **UniON SeLeCT** query. Further, the **UniON SeLeCT** query is referencing parameters such as **1,table_name,3,4,5 From Information_schema.tables where Table_Schema=DatabasE() limit 0,1-- -** etc.

79. From this, we can infer that an attacker has tried to gain access to the details of tables present in the database and the contents stored within them by using the **UniON SeLeCT** query, which is an indicator of an **SQL Injection** attack.

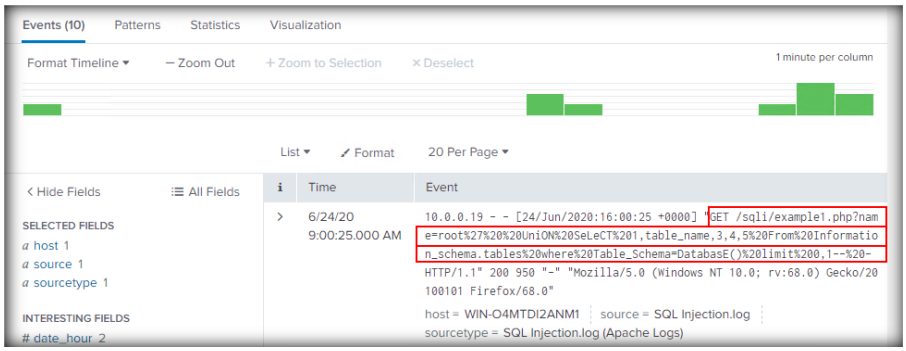

FIGURE 1.65: Malicious SQL query

80. A detailed observation of the log entry provides us with the following findings:

    A. Date and time of the attack: **24th June 2020** and **09:00:25 AM**

    B. IP address of the attacker: **10.0.0.19**

    C. Malicious part of the query used to carry out the attack: **GET /sqli/example1.php?name=root%27%20%20UniON%20SeLeCT%201,table_name,3,4,5%20From%20Information_schema.tables%20where%20Table_Schema=DatabasE()%20limit%200,1--%20-**

    D. HTTP status code **200**: This indicates that the request made to the server has been processed and the attack was successful

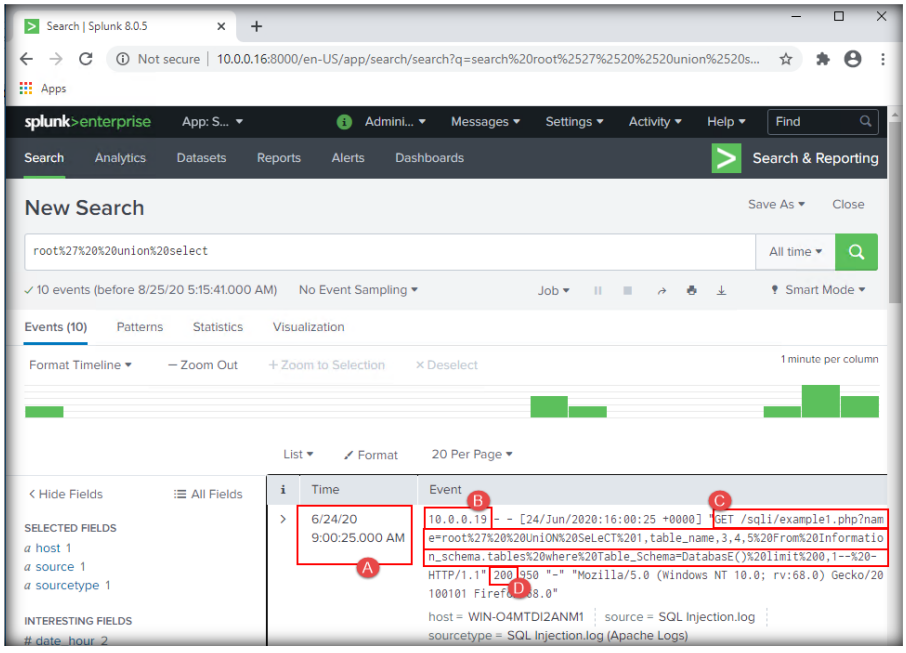

FIGURE 1.66: Detailed analysis of SQL injection attack (Apache logs)

TASK 4

Detect and Examine Directory Traversal Attack via Log Files

81. We will now examine the **Directory Traversal** attack in log files generated by **Apache** server
82. Click **splunk>enterprise** on the top left of the webpage

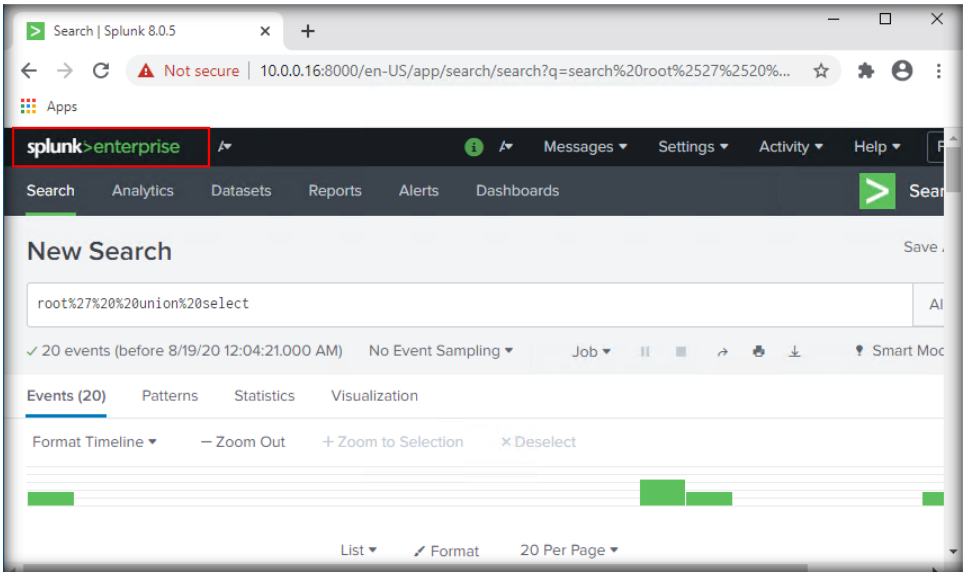

FIGURE 1.67: Click splunk>enterprise

83. You will be directed to the **Splunk Enterprise** homepage. We now need to upload the **Directory Traversal** log file from **Apache Logs**. To upload the evidence file, click on the **Add Data** icon on the homepage and then follow the steps for uploading it.
84. When the **Open** window appears, navigate to **C:\CHFI-Tools\Evidence Files\Log Files\Apache Logs**, select **Directory Traversal**, and click **Open** to upload the file

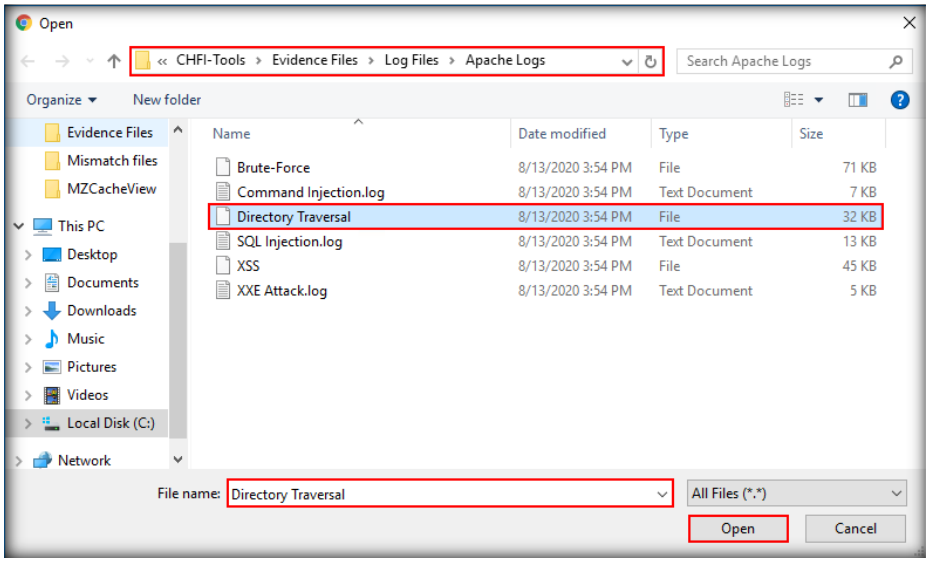

FIGURE 1.68: Select Directory Traversal file and click Open

85. The **Directory Traversal** file will be uploaded successfully. Click **Next** to proceed further.

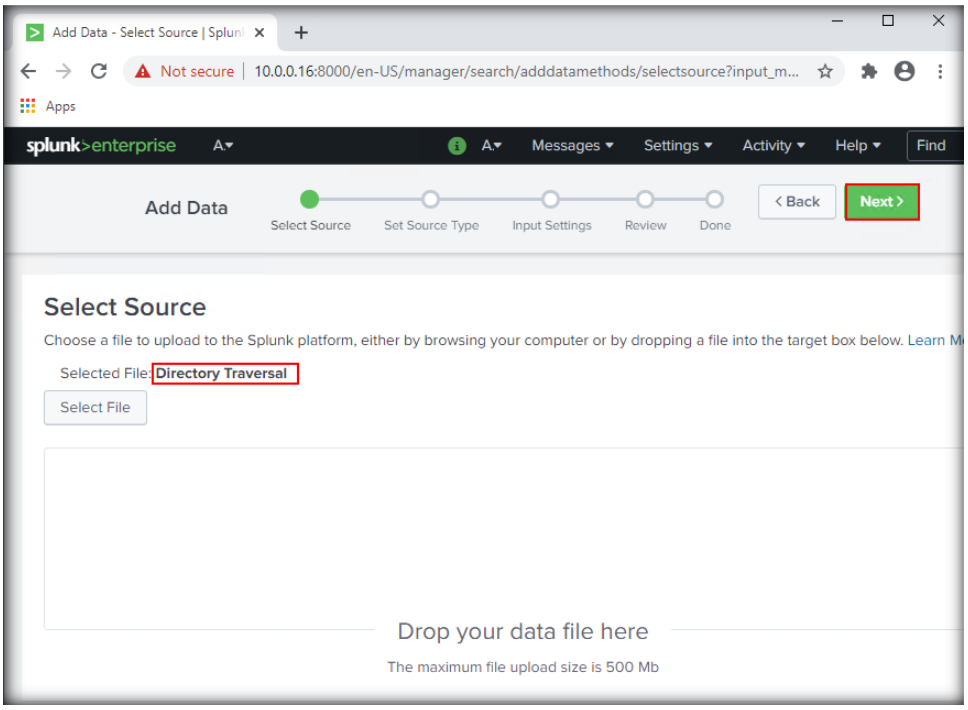

FIGURE 1.69: Click Next

86. In the next step, the **Set Source Type** section appears. Click **Save As**.

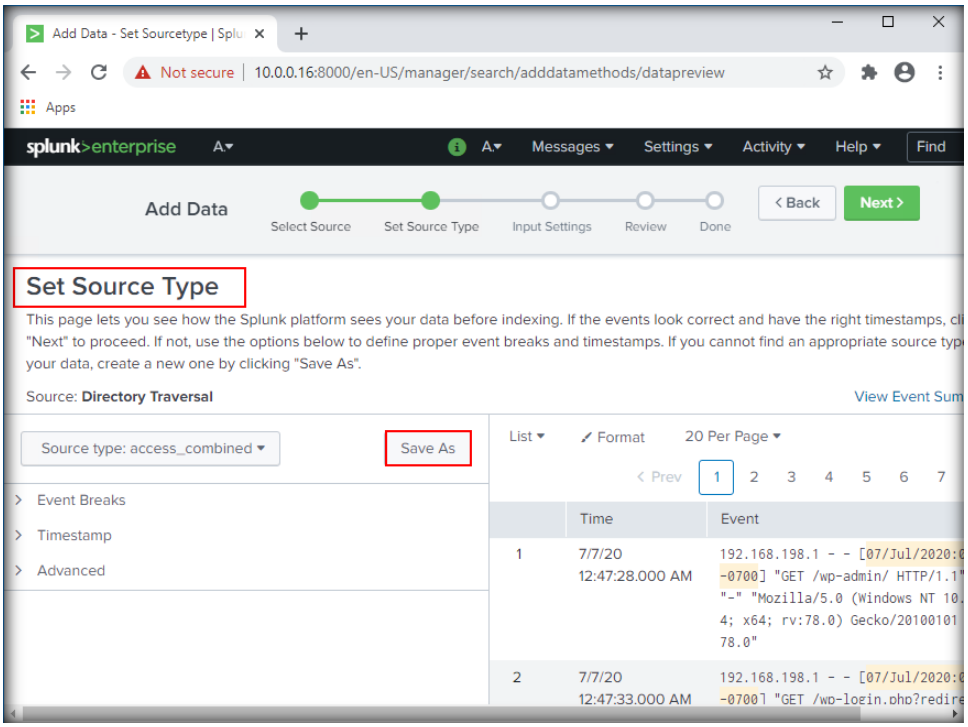

FIGURE 1.70: Click Save As in Set Source Type section

87. The **Save Source Type** window will appear. In the **Name** field, enter **Directory Traversal (Apache Logs)** to assign a name to the source file, and then click to **Save** to continue.

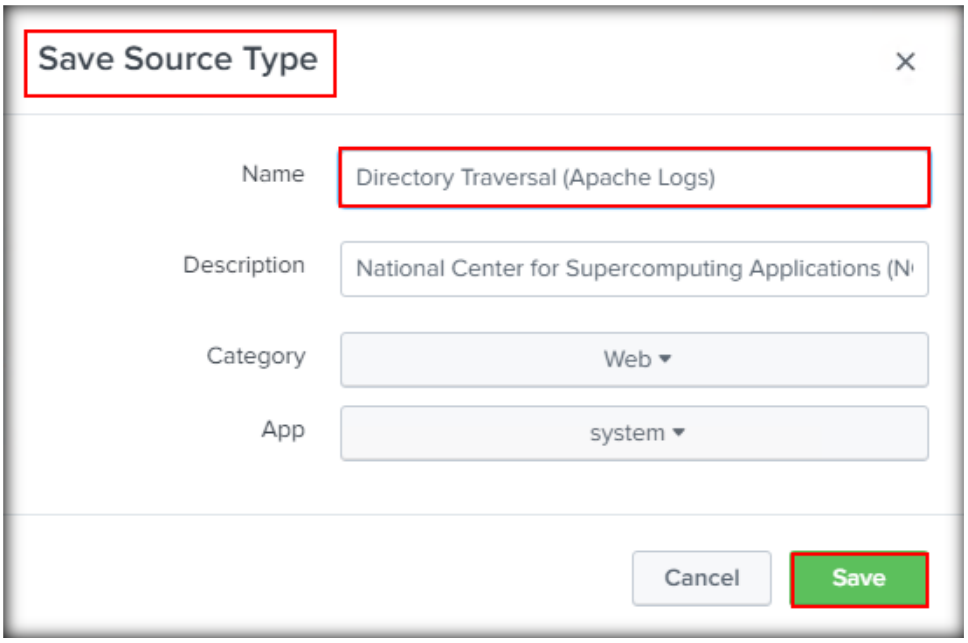

FIGURE 1.71: Enter details in Save Source Type window and click Save

88. You will now get back to the **Set Source Type** section again. Click **Next** at the top right of the page to proceed further.

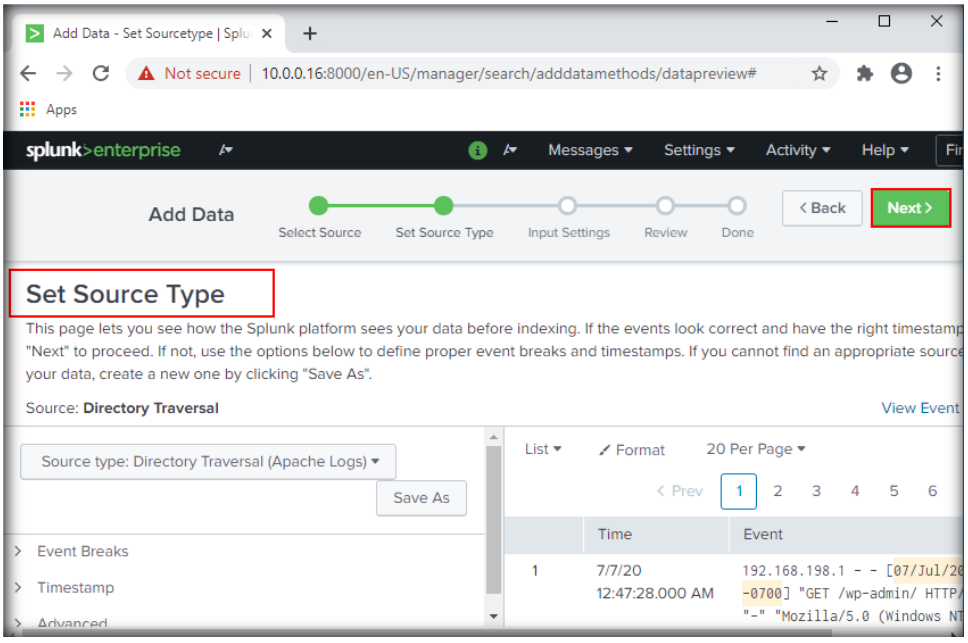

FIGURE 1.72: Click Next on Set Source Type section

89. In the next step, you will see the **Input Settings** section. Click **Review** to proceed further.

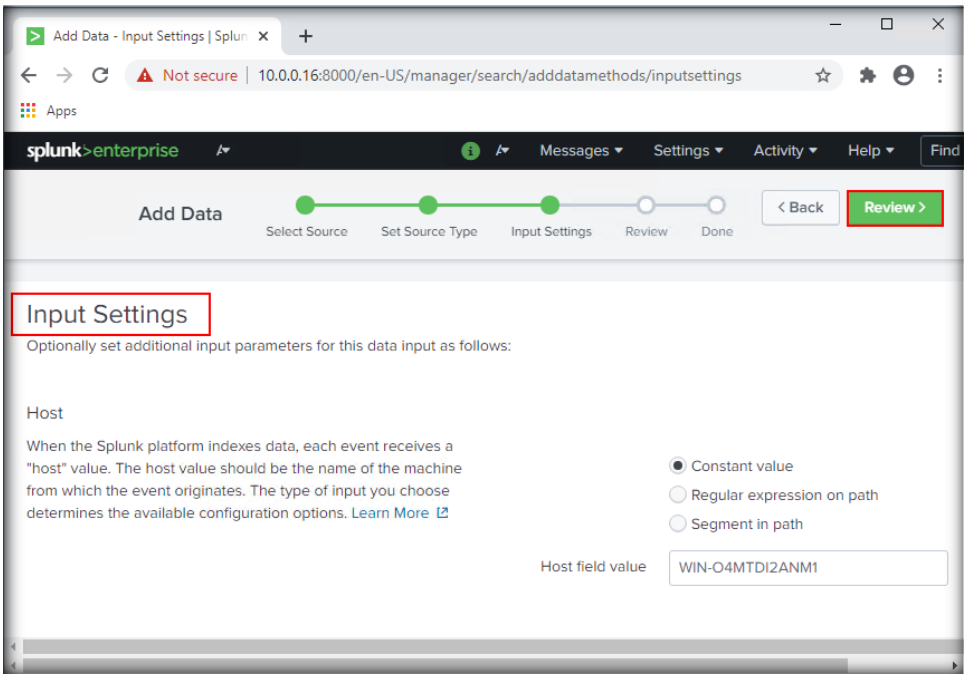

FIGURE 1.73: Click on Review in the Input Settings section

90. In the next step, you will see the **Review** section. Ensure that the details under the **Review** section are correct, and then click **Submit** to proceed further.

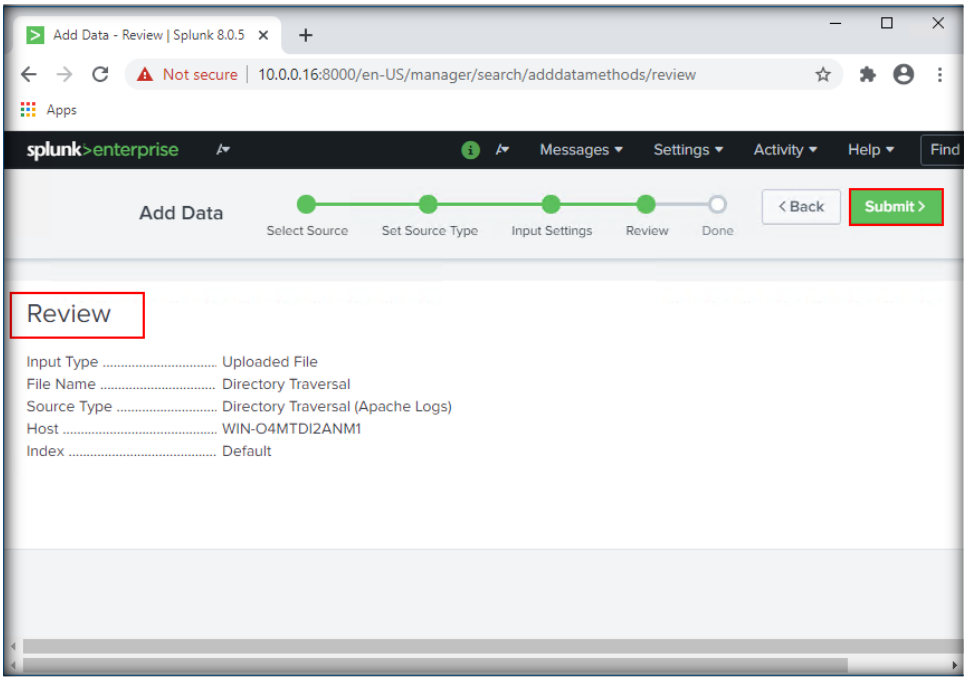

FIGURE 1.74: Review the details and click Submit

91. Upon clicking **Submit**, the application window will display the following message: **File has been uploaded successfully**. Now, click **Start Searching** to fetch the log entries to examine the **Directory Traversal** log file.

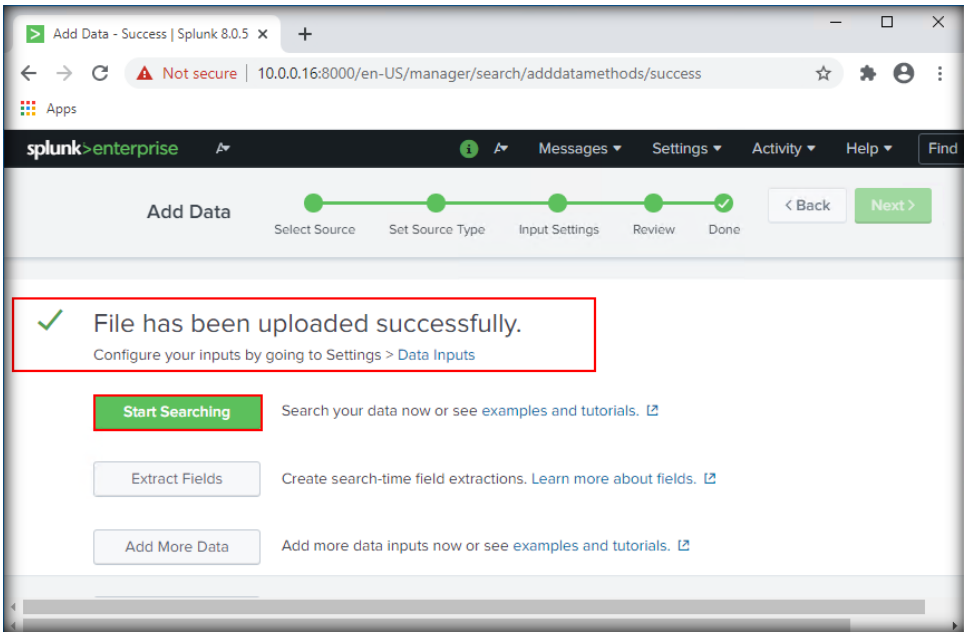

FIGURE 1.75: Click Start Searching

92. You will now see the entries from the uploaded log file displayed in **Splunk Enterprise** under the **Events** tab.

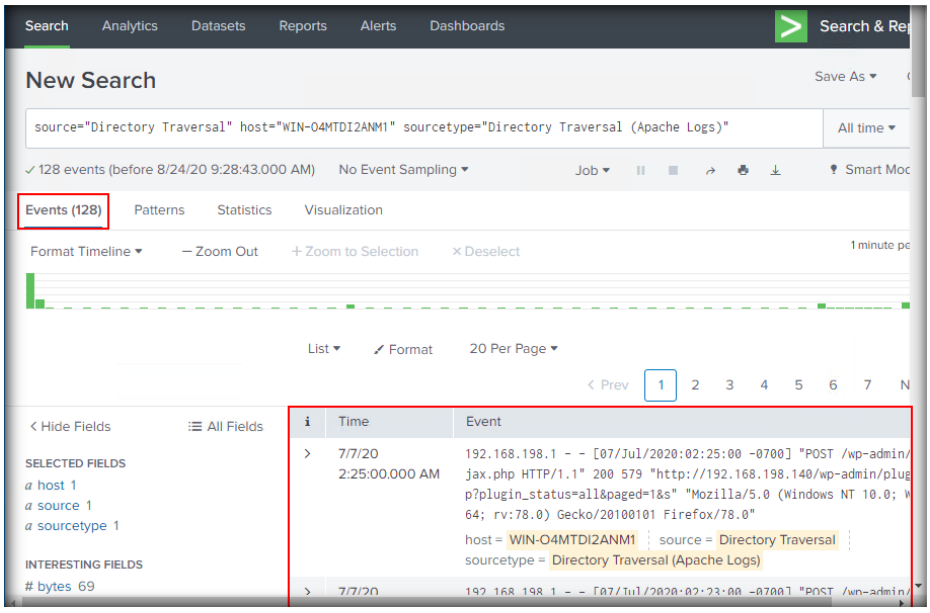

FIGURE 1.76: Log file entries displayed in Splunk

93. To find indicators pertaining to a **directory traversal** attack, we will first apply the plain-text filter **../** by typing it in the **New Search** box and then clicking on the **Search** icon. The purpose of applying this filter is that the presence of the **../ (dot-dot-slash)** sequence(s) in a log entry will help an investigator establish the occurrence of a **directory traversal** attack.

Note: Upon applying the above-mentioned filter, you might view entries from previously sourced log files. In such cases, you need to specify the current log file along with the search filter, for example, **../ sourcetype="Directory Traversal (Apache Logs)"**, to fetch relevant results.

94. Upon applying the filter, we see that a couple of log entries have been retrieved as results. We need to examine these entries to look for the signs of **directory traversal** attack.

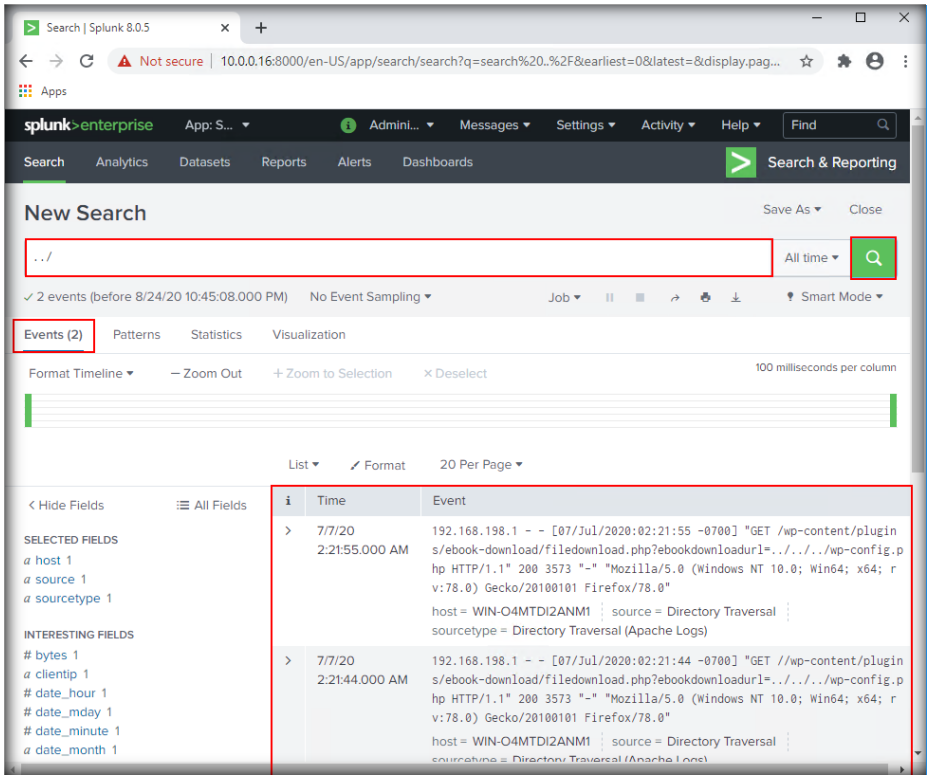

FIGURE 1.77: Log entries found after applying dot-dot-slash filter

Note: For the ease of demonstration, we have used a log file containing entries specific only to recorded web attacks. In real-time scenarios, however, you might see several log entries upon applying a filter and you might have to scroll down the window to manually look for the log entry containing indicators of a **directory traversal** attack.

95. Upon examining the first log entry, we notice that it contains a sequence of **dot-dot-slash (../../../)** in the query string, which matches with the filter we applied (**../**). Furthermore, we also notice that an attempt has been made to download the **wp-config.php** file using the dot-dot-slash **(../../../)**, which is an indicator of **directory traversal** attack. **wp-config.php** is a core **WordPress** file that stores information pertaining to the **WordPress** database such as host information, usernames, and passwords.

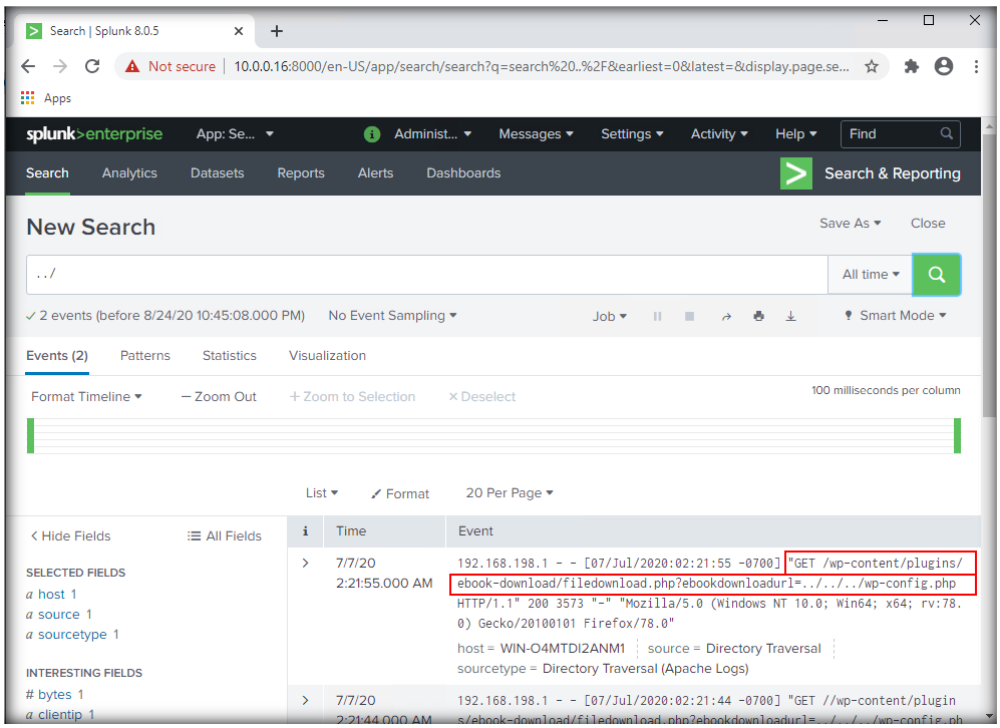

FIGURE 1.78: Malicious query

Note: In case the plain-text filter **../** fails to retrieve any log entries, you may apply different encoded variants of the **../** filter such as **%2E%2E%2F**, **%2E%2E/,** or **..%2F** to look for attempts of a directory traversal attack.

96. Detailed observation of the log entry provides us with the following findings:

A. Date and time of the attack: **07th July 2020**, **02:21:55 AM**

B. IP address of the attacker: **192.168.198.1**

C. Malicious query used in the **Directory Traversal** attack: **GET /wp-content/plugins/ebook-download/filedownload.php?ebookdownloadurl=../../../wp-config.php**

D. HTTP Status code **200**: This means the request made to the server has been processed and attack was successful.

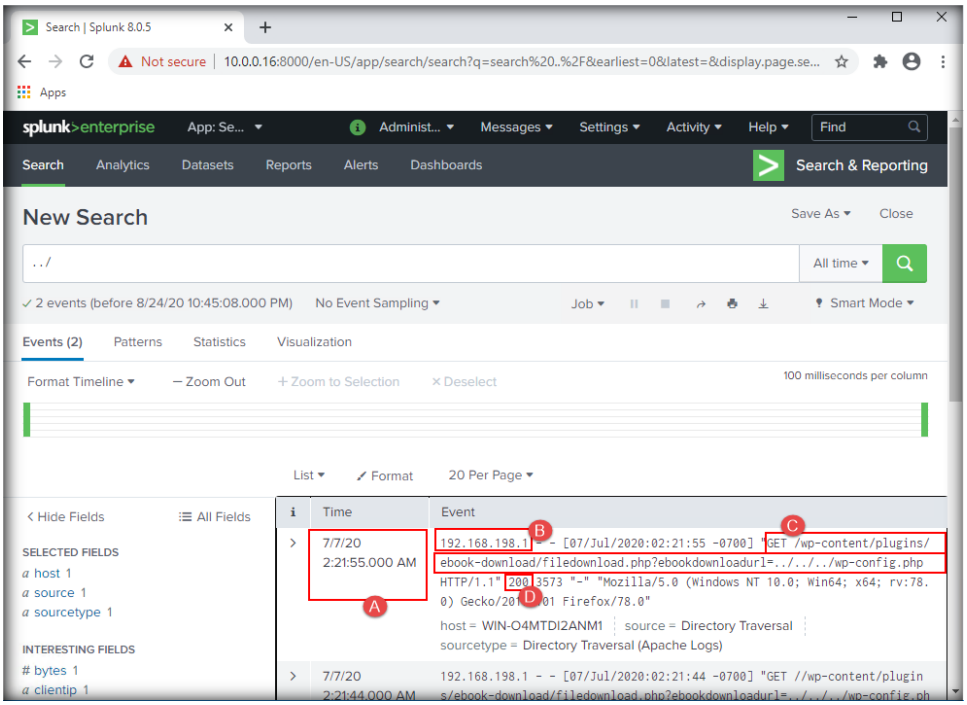

FIGURE 1.79: Detailed analysis of directory traversal attack

97. We will now examine the **Directory Traversal** log file generated by **ModSecurity** web application firewall.

98. Click **splunk>enterprise** at the top left of the webpage.

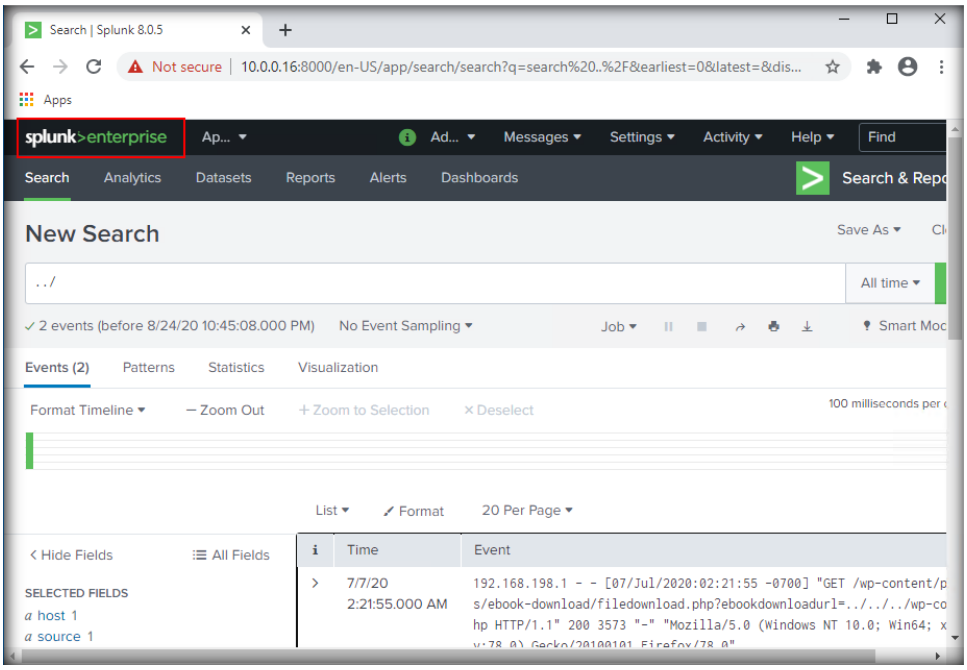

FIGURE 1.80: Click splunk>enterprise

99. When the homepage of **Splunk Enterprise** opens, click on the **Add Data** icon and then follow the steps meant for uploading the evidence file.

100. The **Open** window will appear. Navigate to **C:\CHFI-Tools\Evidence Files\Log Files\ModSecurity Logs** and select **Directory Traversal**. Click **Open** to upload the file.

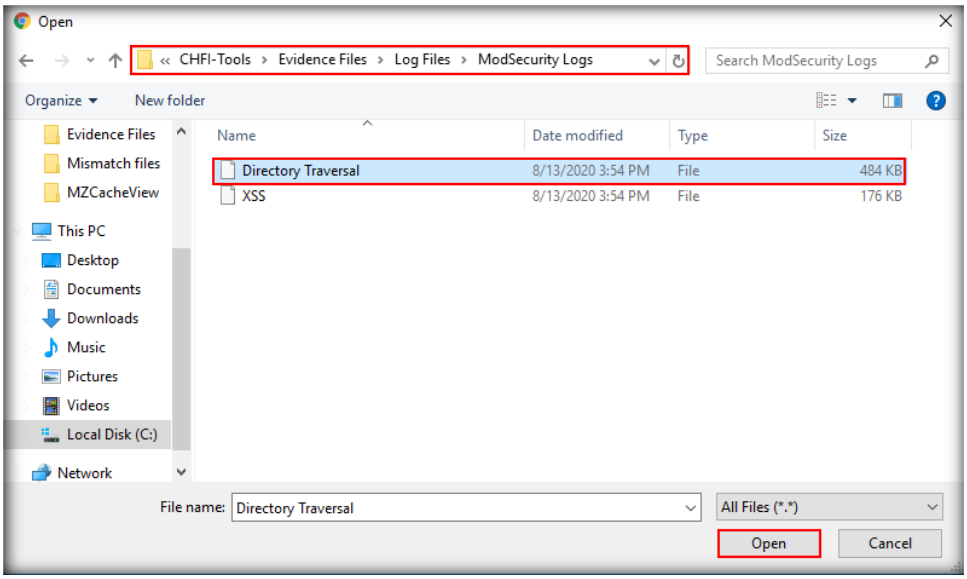

FIGURE 1.81: Select Directory Traversal file from ModSecurity Logs folder and click Open

101. The **Directory Traversal** file will be uploaded successfully. Click **Next** to continue.

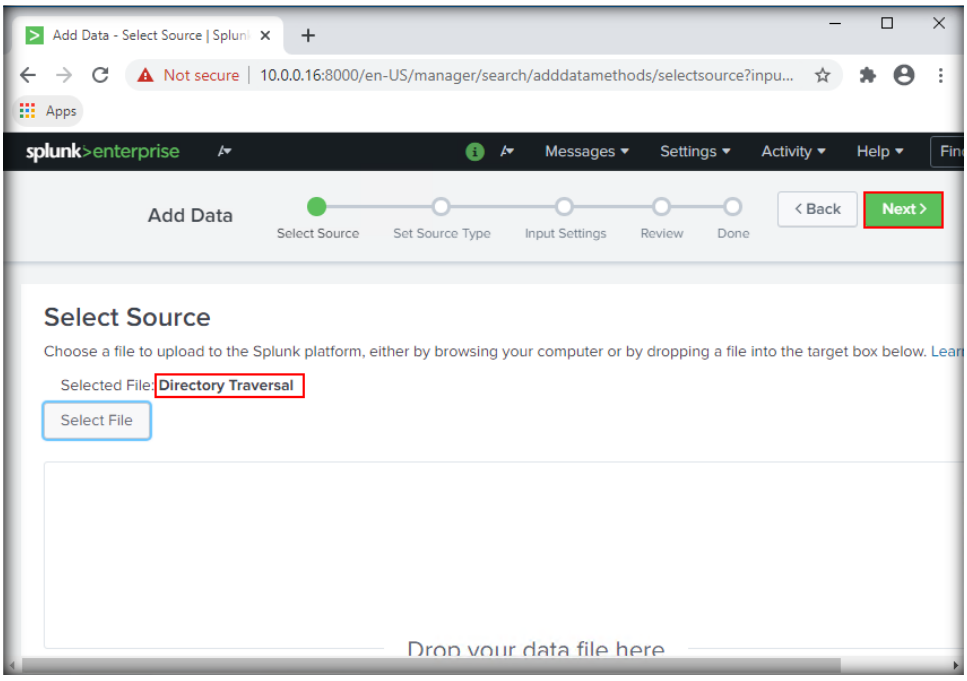

FIGURE 1.82: Click Next

102. In the next step, you will see the **Set Source Type** section. Click the **Save As** button.

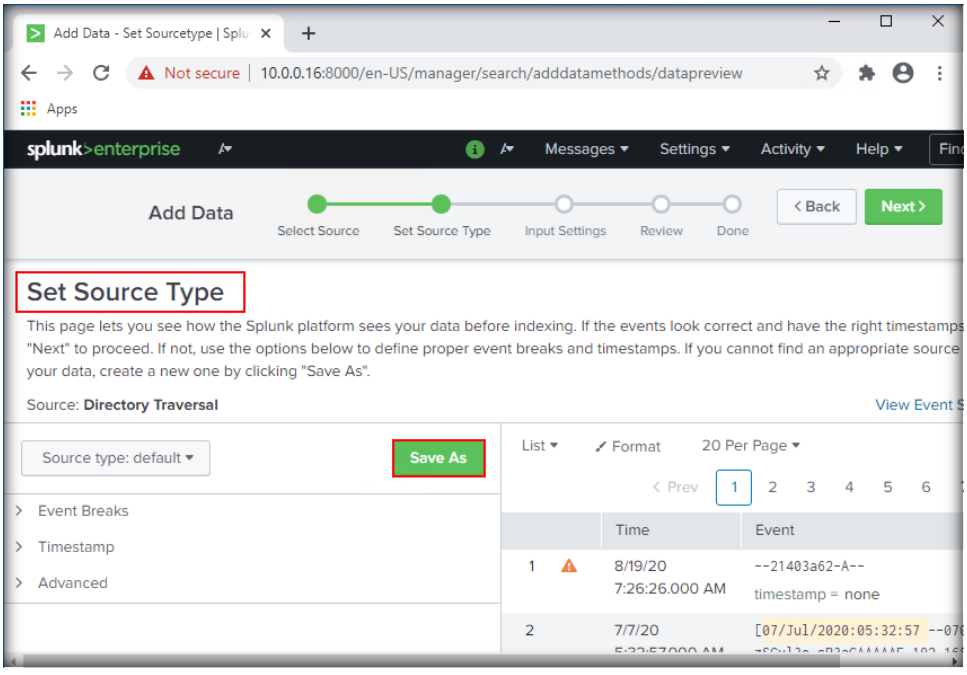

FIGURE 1.83: Click Save As in Set Source Type section

103. The **Save Source Type** window will now appear. In the **Name** field, we will enter **Directory Traversal (ModSecurity Logs)**. Click **Save** to continue.

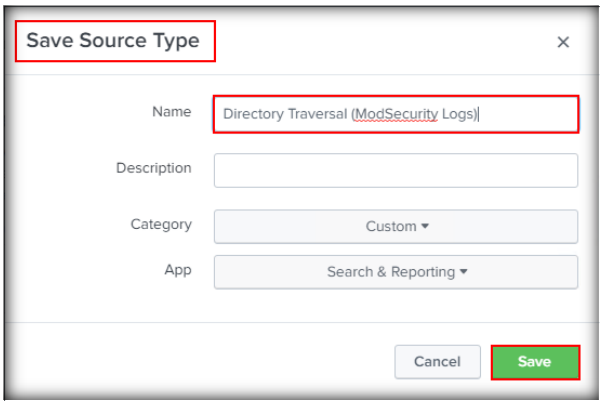

FIGURE 1.84: Enter details in Save Source Type window and click Save

104. You will now see the **Set Source Type** section again. Click **Next** to proceed further.

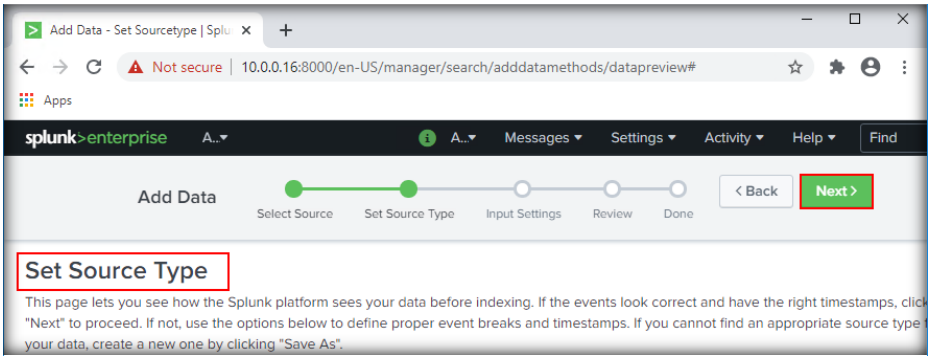

FIGURE 1.85: Click Next in Set Source Type section

105. In the next step, the **Input Settings** section will appear. Click **Review** to proceed further.

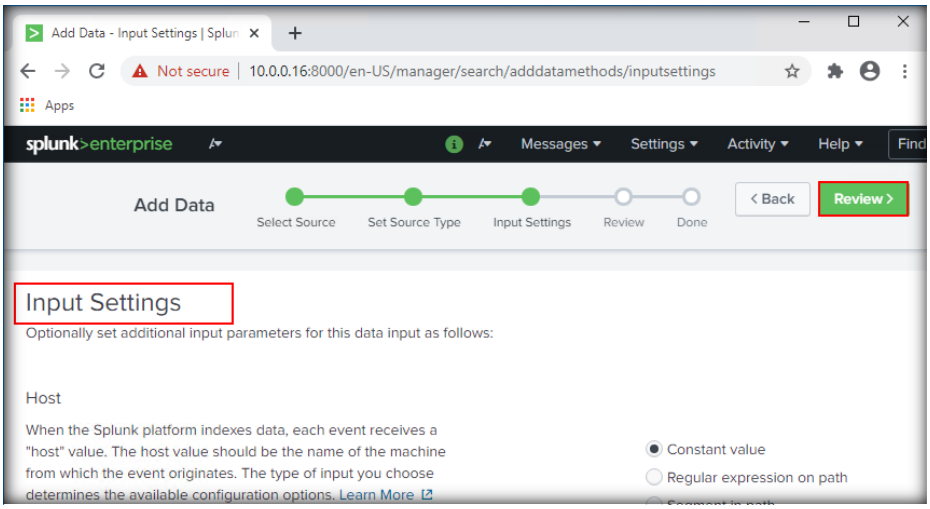

FIGURE 1.86: Click on Review in the Input Settings section

106. In the next step, the **Review** section will appear. Ensure that the details under the **Review** section are correct, and then click **Submit** to proceed further.

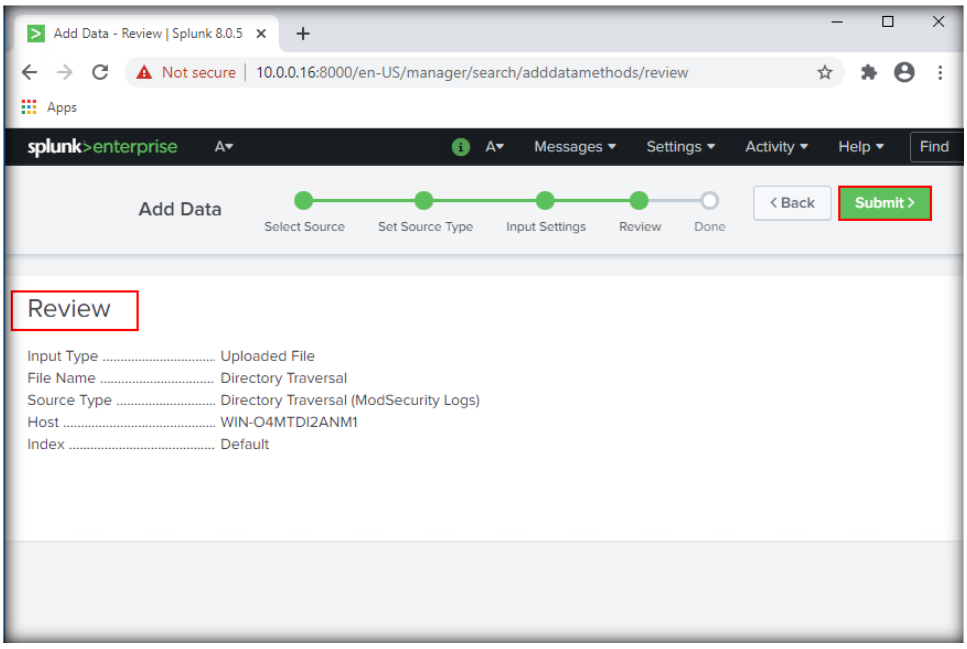

FIGURE 1.87: Review the details and click Submit

107. Upon clicking **Submit**, the application window will display the following message: **File has been uploaded successfully**. Now, click **Start Searching** to fetch the log entries from the **Directory Traversal** log file.

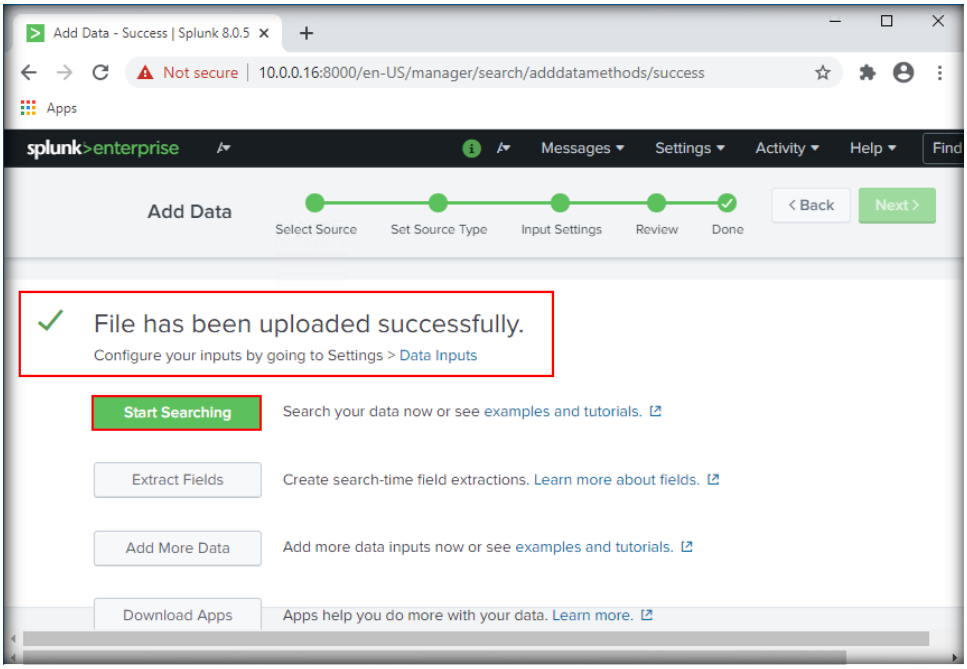

FIGURE 1.88: Click Start Searching

108. You will now see the log entries from the uploaded log file displayed in **Splunk Enterprise** under the **Events** tab.

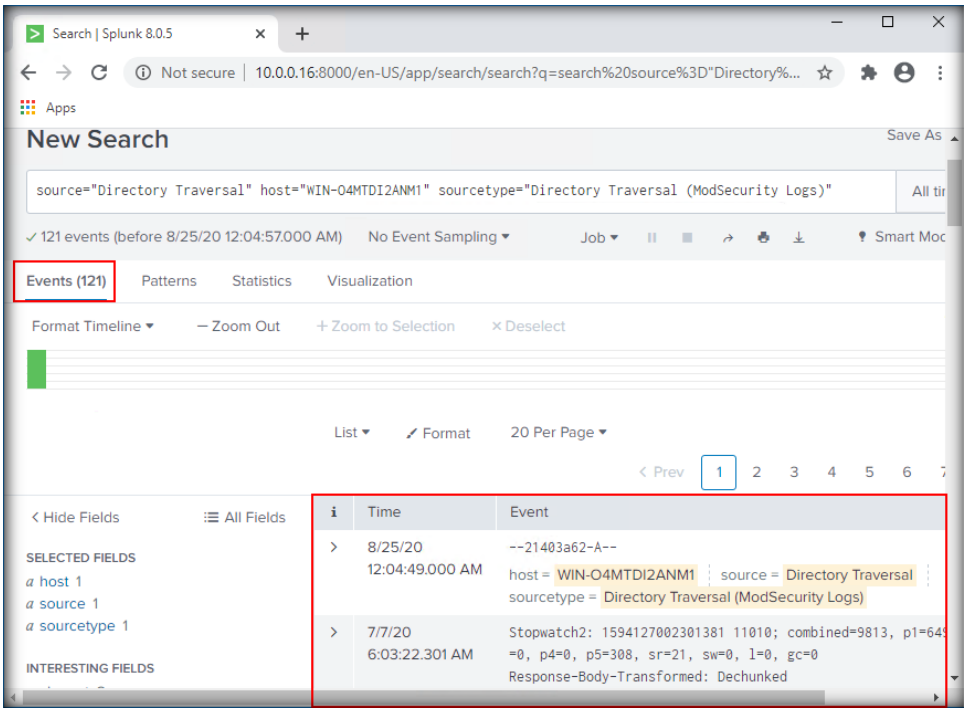

FIGURE 1.89: Log file entries displayed in Splunk

109. As in the previous case, we will now apply the plain-text filter **../** to check for indicators pertaining to a **Directory Traversal** attack. To apply the filter, type **../** in the **New Search** box and then click on the **Search** icon.

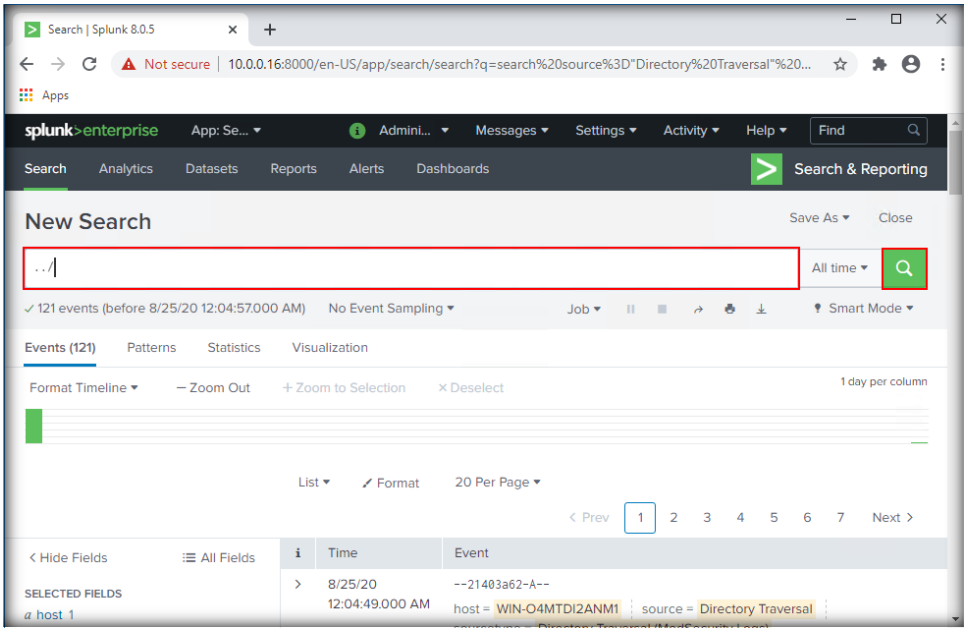

FIGURE 1.90: Use filter ../

Note: Upon applying the above-mentioned filter, you might view entries from previously sourced log files. In such a case, you need to specify the current log file along with the search filter, for example, **../ sourcetype="Directory Traversal (ModSecurity Logs)"**, to fetch relevant results.

110. Upon applying the filter as shown above, we see a few log entries displayed as results under the **Events** tab, as shown in the screenshot below:

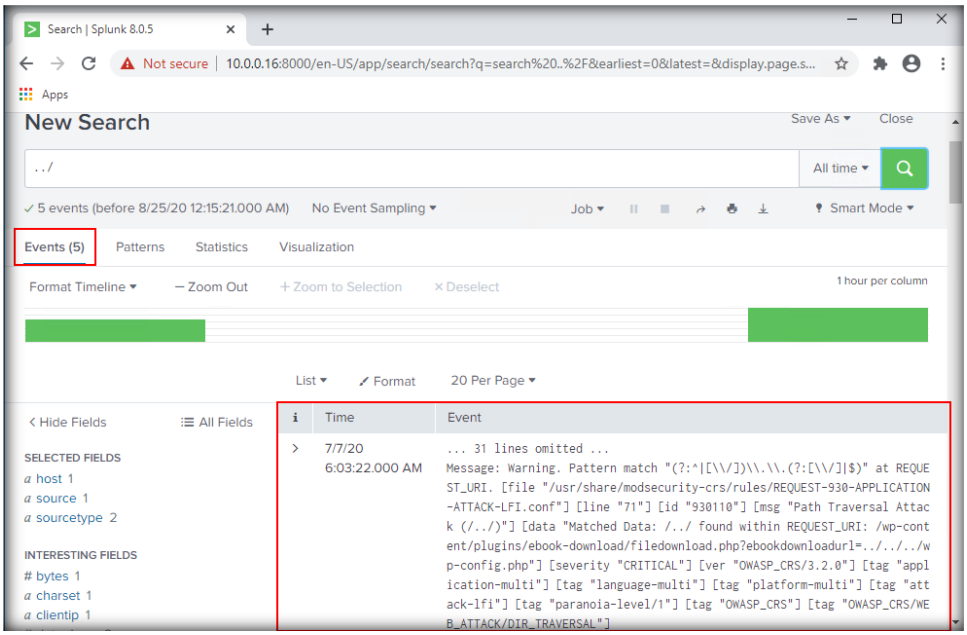

FIGURE 1.91: Results using encoded filter

Note: For the ease of demonstration, we have used a log file containing entries specific only to recorded web attacks. In real-time scenarios, however, you might see several log entries upon applying a filter and you might have to scroll down the window to manually look for the log entry containing indicators of a **Directory Traversal** attack.

111. As in the previous case, we need to look for the presence of **../ (dot-dot-slash)** sequence(s) in the query string to detect a **directory traversal** attack. Upon examining the log entry at the top in detail, we find that the query string contains a number of **../** sequences, as seen in the screenshot below:

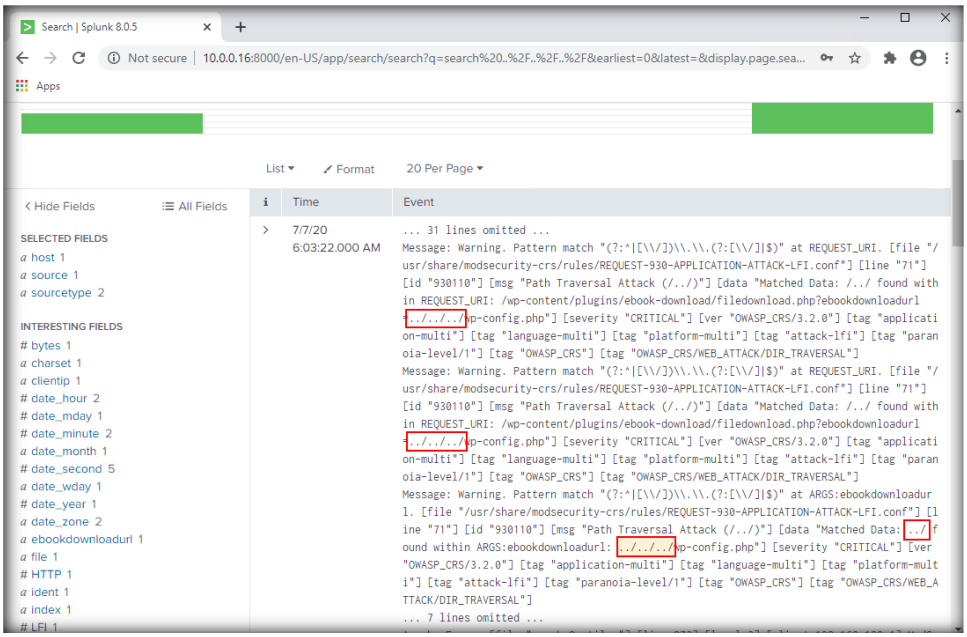

FIGURE 1.92: ../ sequences found

Note: In case the plain-text filter **../** fails to retrieve any log entries, you may apply different encoded variants of the **../** filter such as **%2E%2E%2F**, **%2E%2E/,** or **..%2F** to look for attempts of a **directory traversal** attack.

112. A detailed observation of the log entry provides us with the following findings (you must click on the **Show all 50 lines** option present below the log entry to be able to see the following details listed below):

A. Date and time of the attack: **07th July 2020** and **06:03:22 AM**

B. IP address of the attacker: **192.168.198.1**

C. Malicious query used for **directory traversal** attack: **GET /wp-content/plugins/ebook-download/filedownload.php?ebookdownloadurl=../../../wp-config.php** (as in the previous case of directory traversal attack logged by the Apache server, the file targeted in this case also is **wp-config.php**)

D. IP address of the Host: **192.168.198.140**

E. **HTTP/1.1 403 Forbidden message**: This means the access to requested resource has been blocked by the **ModSecurity** firewall

F. Message generated by Server in response to attacker's **GET** request: **You don't have permission to access this resource**

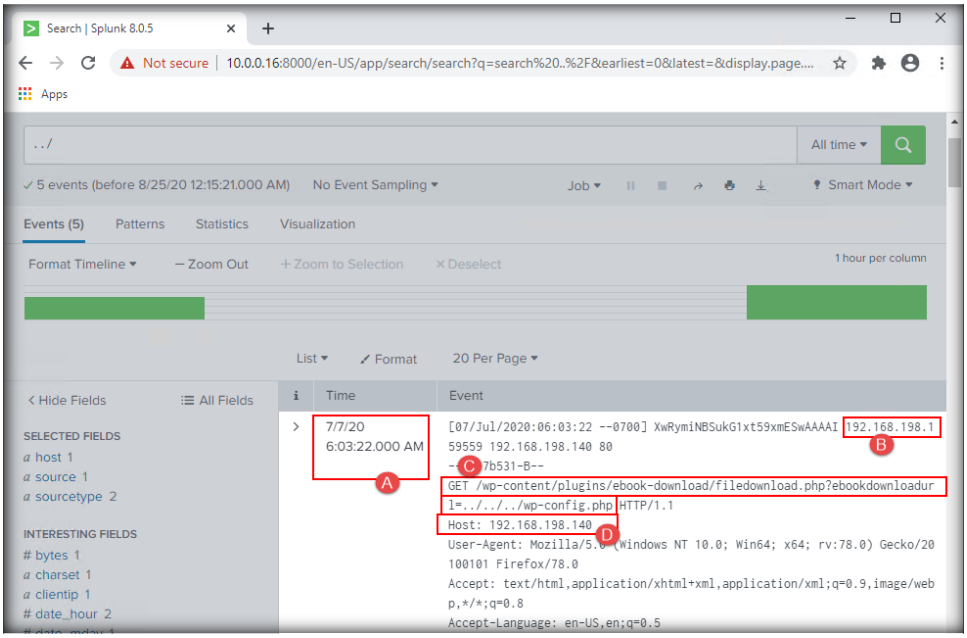

FIGURE 1.93: Detailed analysis of directory traversal attack (ModSecurity)

Note: Scroll down in the application window to find the observations **E** and **F**, as indicated in the screenshot below:

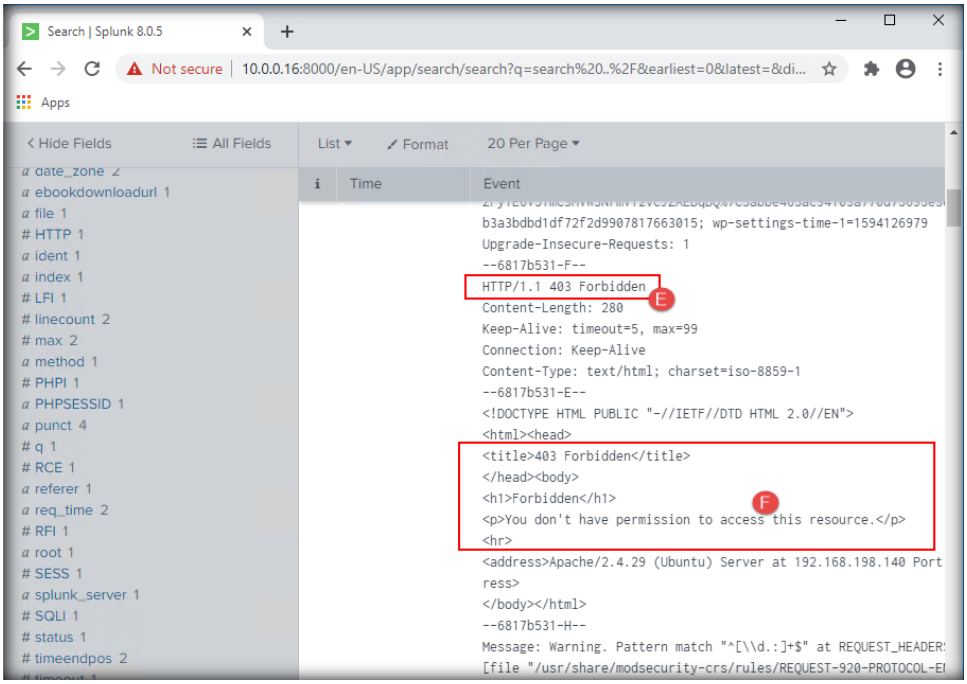

FIGURE 1.94: Access Forbidden message

113. We will now examine the **Directory Traversal.txt** log file generated by **IIS** server.

114. Click **splunk>enterprise** at the top left of the webpage.

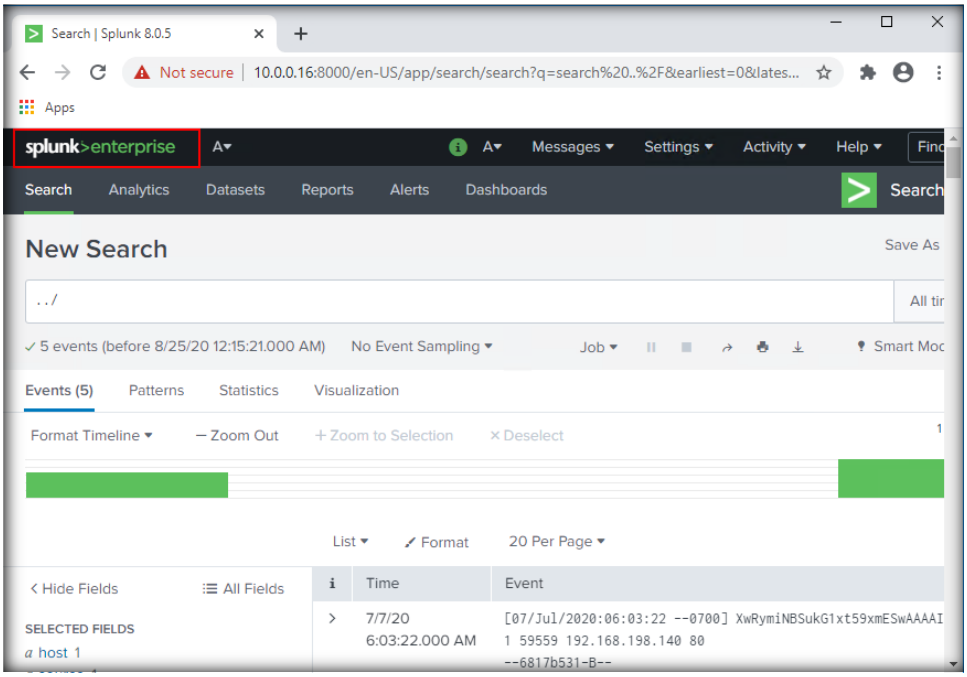

FIGURE 1.95: Click splunk>enterprise

115. You will be directed to the homepage of **Splunk Enterprise**. Click on the **Add Data** icon on the homepage and then follow the steps for uploading it.

116. The **Open** window will appear. Navigate to **C:\CHFI-Tools\Evidence Files\Log Files\IIS Logs** and select **Directory Traversal.txt**. Click **Open** to upload the file.

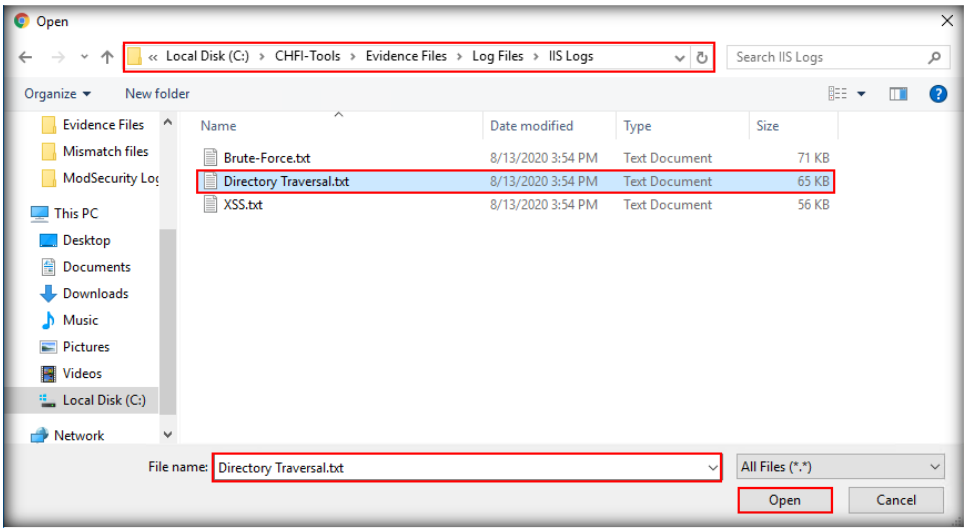

FIGURE 1.96: Select Directory Traversal.txt and click Open

117. The **Directory Traversal.txt** file will be uploaded successfully. Click **Next** to continue.

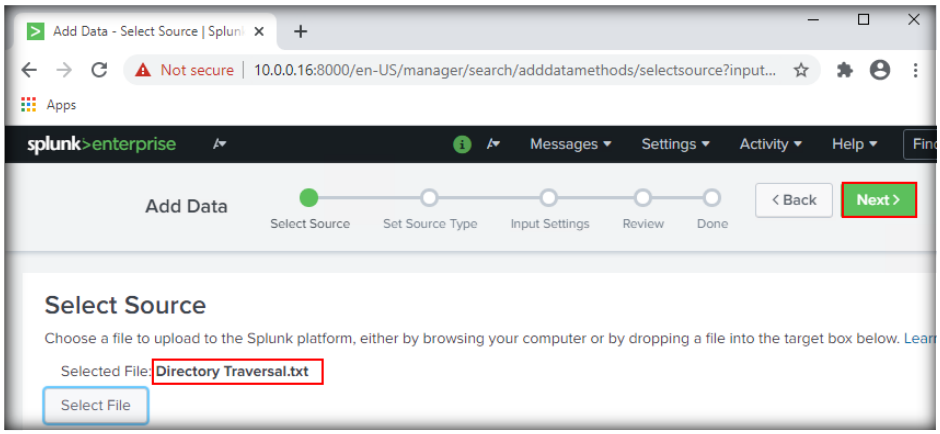

FIGURE 1.97: Click Next

118. In the next step, you will see the **Set Source Type** section. Click **Save As**.

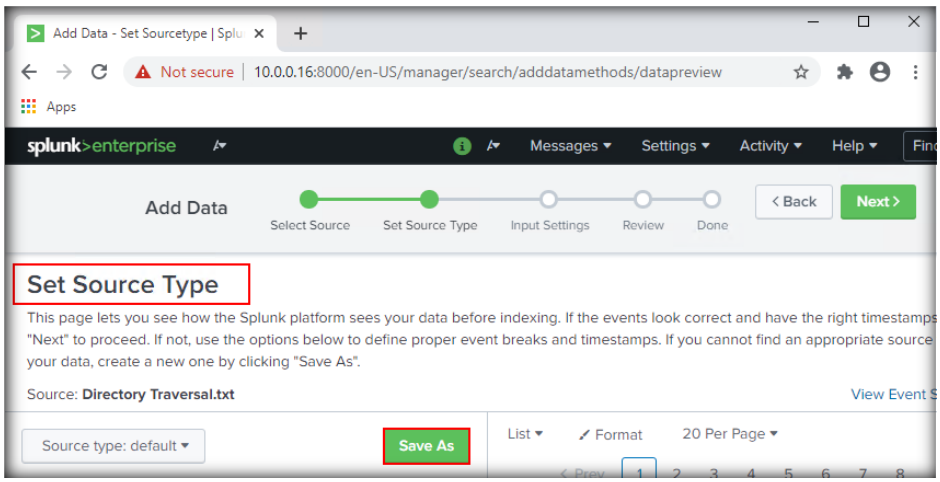

FIGURE 1.98: Click Save As in Set Source Type section

119. The **Save Source Type** window will now appear. In the **Name** field, we will enter **Directory Traversal.txt (IIS Logs)**. Click **Save** to continue.

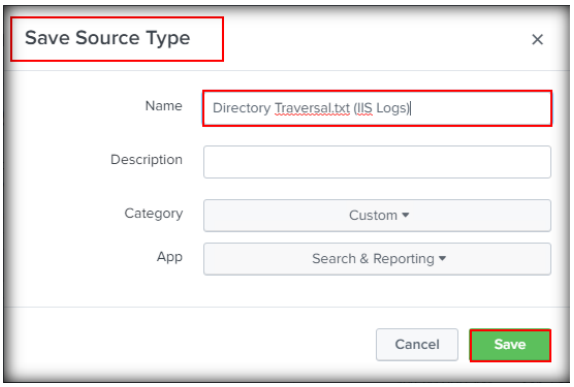

FIGURE 1.99: Enter details in Save Source Type window and click Save

120. You will now get back to the **Set Source Type** section again. Click **Next** to continue.

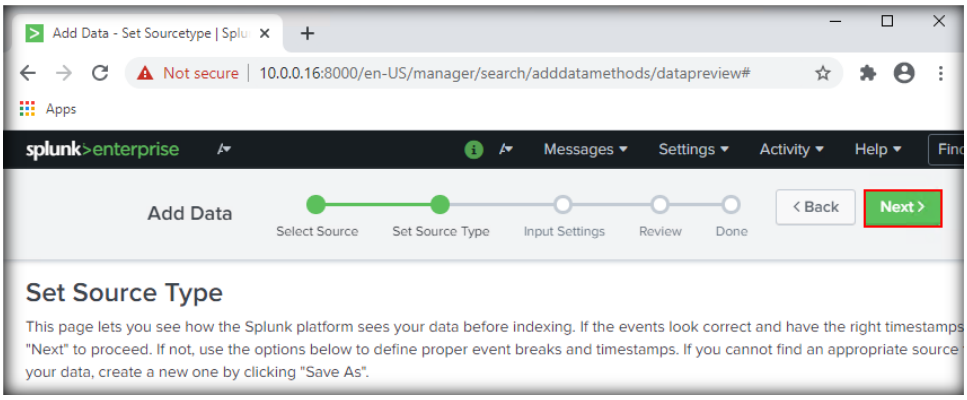

FIGURE 1.100: Click Next in Set Source Type section

121. In the next step, you will see the **Input Settings** section. Click **Review** to proceed further.

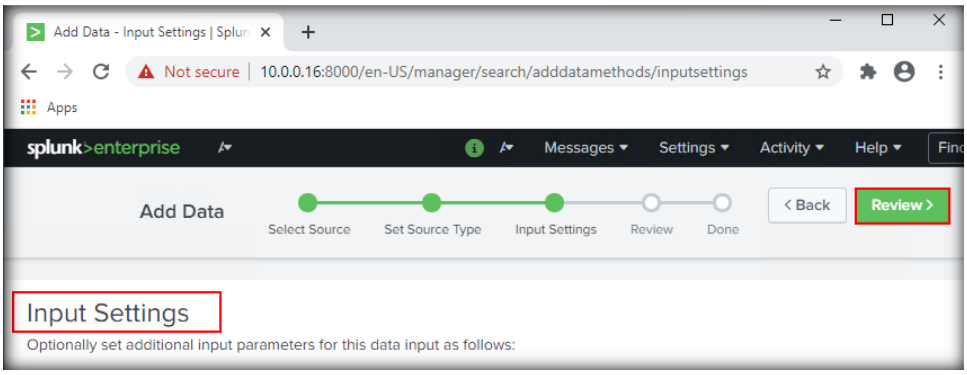

FIGURE 1.101: Click on Review in the Input Settings section

122. In the next step, you will see the **Review** section. Ensure that the details under the **Review** section are correct and then click **Submit** to proceed further.

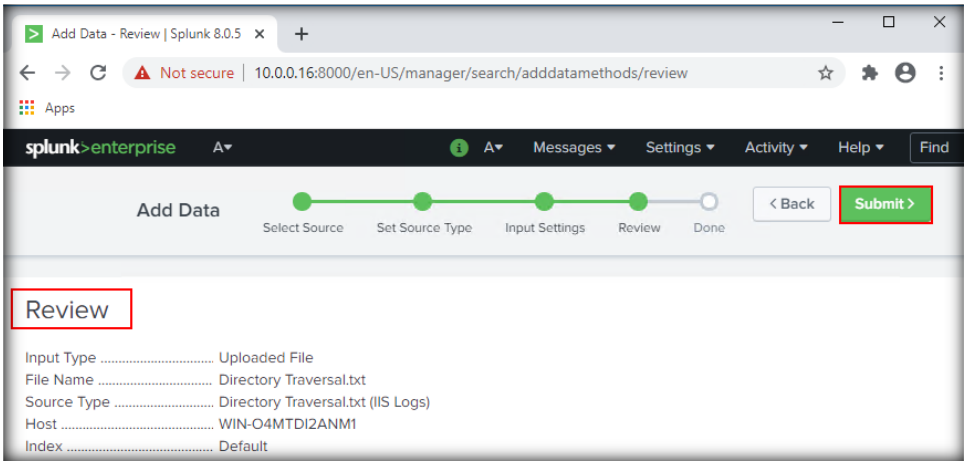

FIGURE 1.102: Review the details and click Submit

123. Upon clicking **Submit**, the application window will display the following message: **File has been uploaded successfully**. Now, click **Start Searching** to fetch the log entries from the uploaded log file.

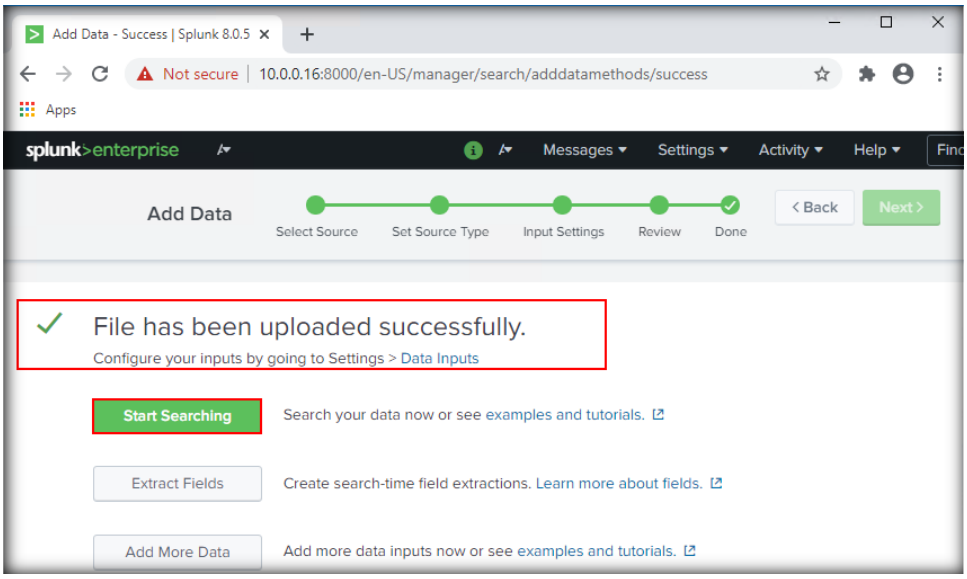

FIGURE 1.103: Click Start Searching

124. The log entries from **Directory Traversal.txt** log file will now be displayed in **Splunk Enterprise** under the **Events** tab.

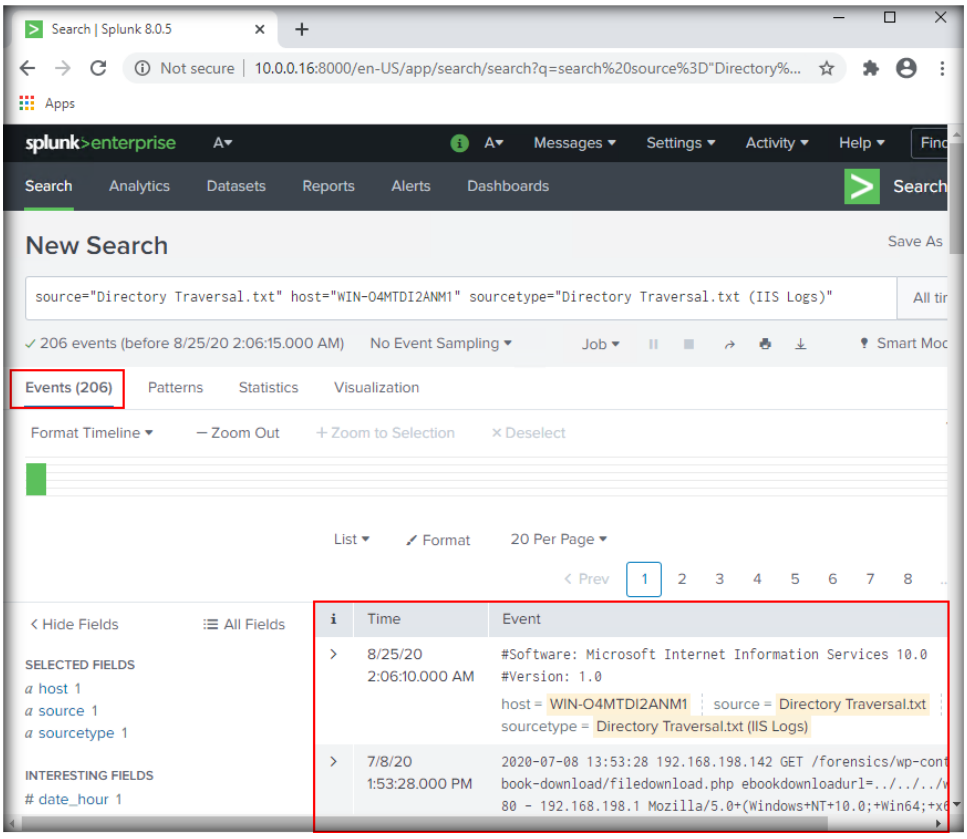

FIGURE 1.104: Log file entries displayed in Splunk

125. As in the previous two cases, we will apply the plain-text filter **../** here. To apply the filter, type **../** in the **New Search** box, and then click the **Search** icon.

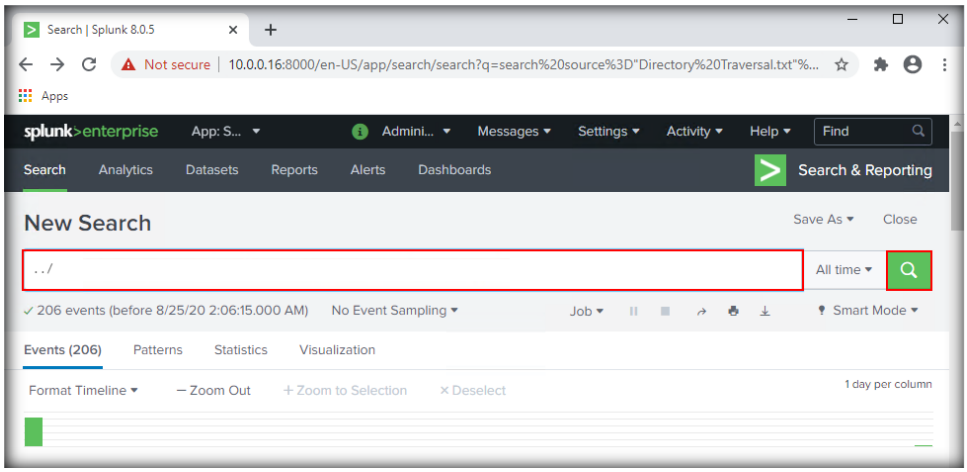

FIGURE 1.105: Use filter ../

Note: Upon applying the above-mentioned filter, you might view entries from previously sourced log files. In such a case, you need to specify the current log file along with the search filter, for example, **../ sourcetype="Directory Traversal.txt (IIS Logs)"**, to fetch relevant results.

126. Upon applying the filter as shown above, we can see a few log entries displayed as results under the **Events** tab.

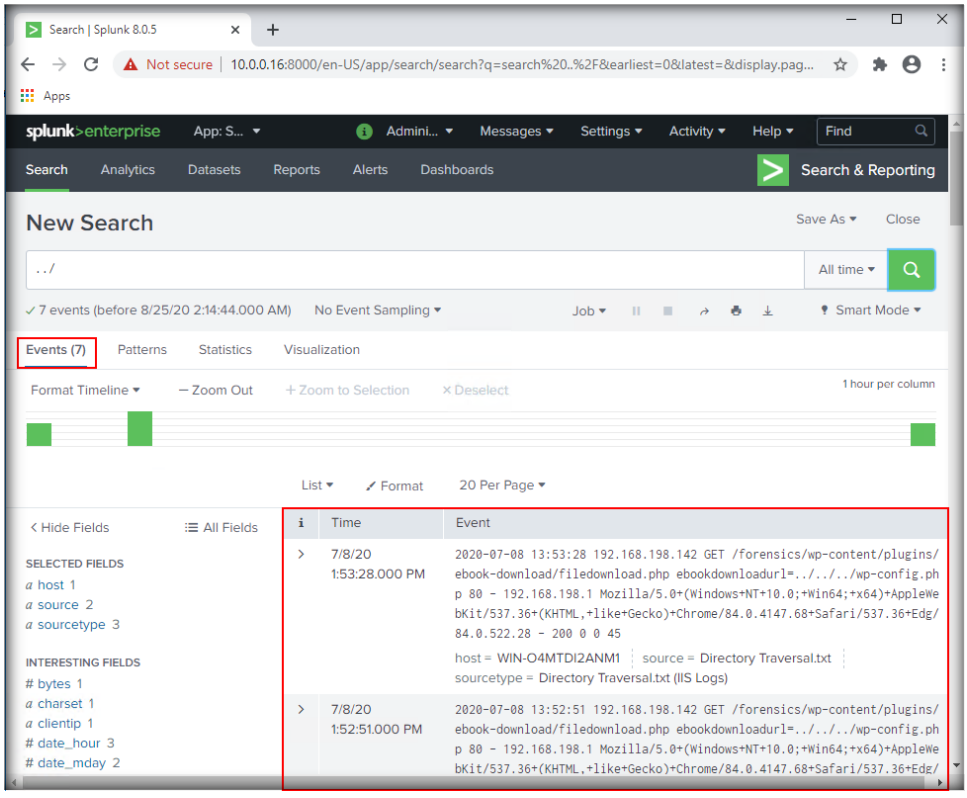

FIGURE 1.106: Results using ../ filter

Note: For the ease of demonstration, we have used a log file containing entries specific only to recorded web attacks. In real-time scenarios, however, you might see several log entries upon applying a filter and you might have to scroll down the window to manually look for the log entry containing indicators of a **Directory Traversal** attack.

127. As in the previous two cases, we need to look for the presence of the **../ (dot-dot-slash)** sequence(s) or its encoded variant in the query string to detect a **directory traversal** attack. Upon examining the log entry at the top, we find that the query string contains a sequence of **../**, as seen in the screenshot below:

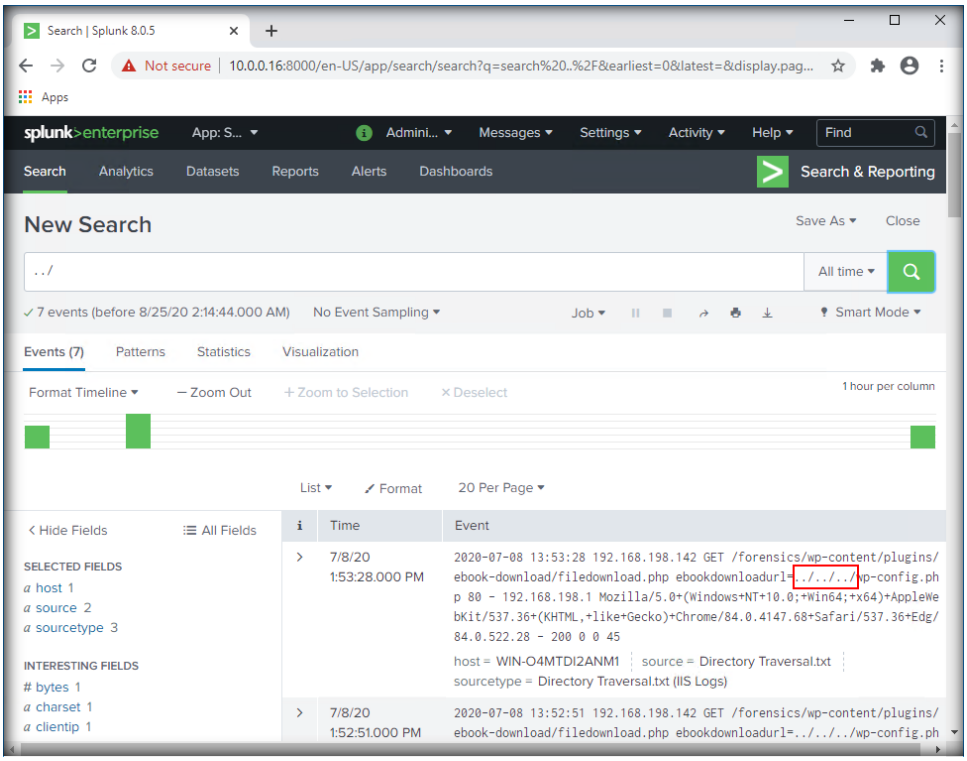

FIGURE 1.107: ../ sequence detected

Note: In case the plain-text filter **../** fails to retrieve any log entries, you may apply different encoded variants of the **../** filter such as **%2E%2E%2F**, **%2E%2E/**, or **..%2F** to look for attempts of directory traversal attack

128. A detailed observation of the log entry provides us with the following findings:

   A. Date and time of the attack: **08th July 2020** and **01:53:28 PM**

   B. IP address of the Host: **192.168.198.142**

   C. Malicious query used in the **directory traversal** attack: **GET /forensics/wp-content/plugins/ebook-download/filedownload.php ebookdownloadurl=../../../wp-config.php** (again, the file targeted by the attacker is **wp-config.php**)

   D. IP address of the attacker: **192.168.198.1**

   E. HTTP status code **200**: This indicates that the request made to the server has been received and is being processed

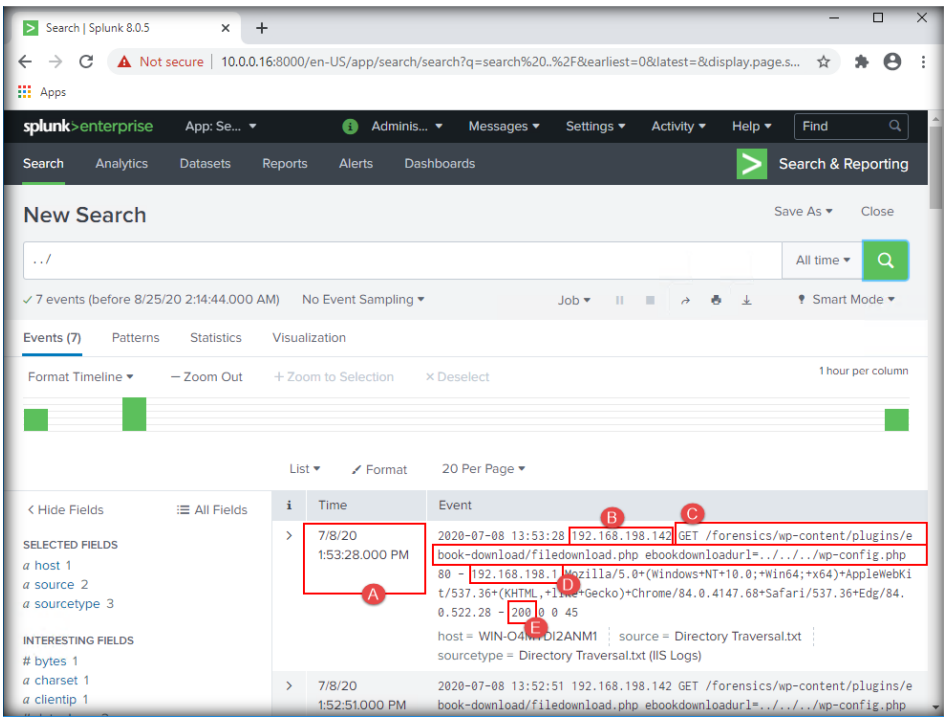

FIGURE 1.108: Detailed analysis of directory traversal attack

TASK 5

Detect and Examine Command Injection Attack via Log Files

129. We will now examine the **Command Injection** log file generated by **Apache** server

130. Click on **splunk>enterprise** at the top left of the page

131. You will be directed to the homepage of **Splunk Enterprise**. We now need to upload the **Command Injection.log** file. To be able to upload the evidence file, click the **Add Data** icon on the homepage and then follow the steps for uploading it.

132. **Open** window will now appear. Navigate to **C:\CHFI-Tools\Evidence Files\Log Files\Apache Logs** and select **Command Injection.log**. Click **Open** to upload the file.

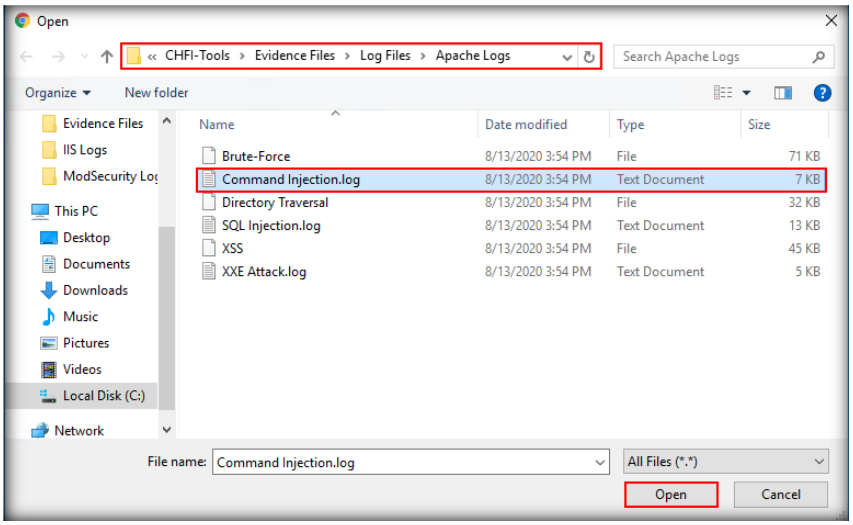

FIGURE 1.109: Select Command Injection.log and click Open

133. The **Command Injection.log** file will be uploaded successfully. Click **Next** to continue.

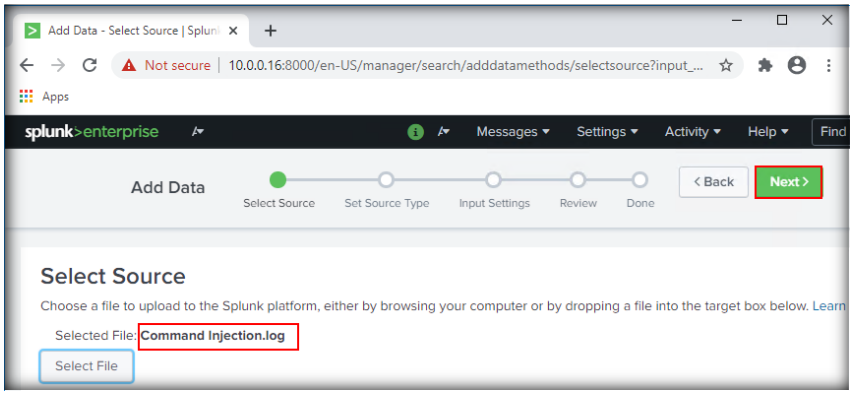

FIGURE 1.110: Click Next

134. In the next step, the **Set Source Type** section will appear. Click **Save As**.

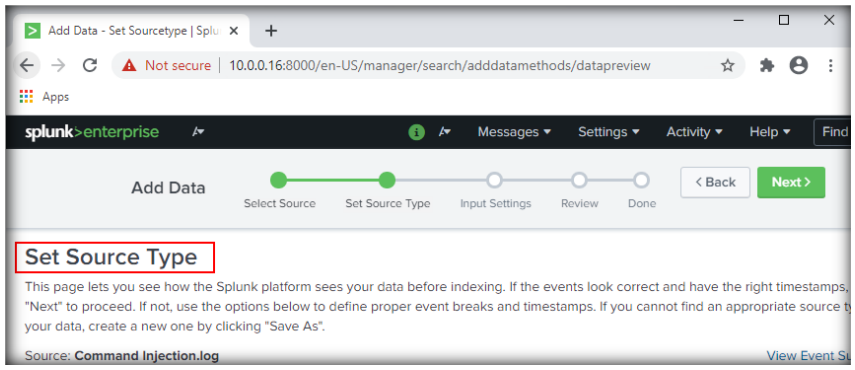

FIGURE 1.111: Click Save As in Set Source Type section

135. The **Save Source Type** window will now appear. In the **Name** field, we will enter **Command Injection.log (Apache Logs)**. Click **Save** to continue.

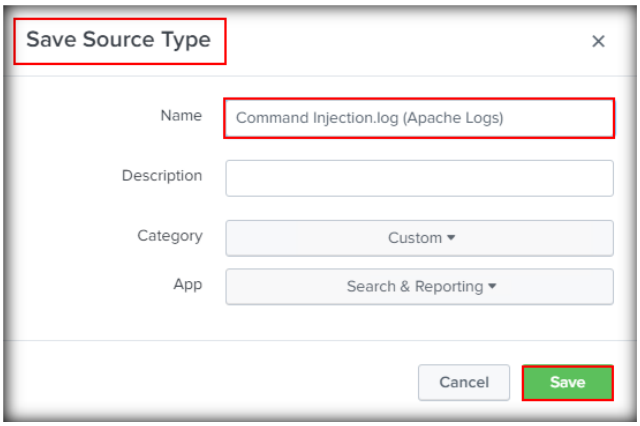

FIGURE 1.112: Enter details in Save Source Type window and click Save

136. You will now get back to the **Set Source Type** section again. Click **Next** to proceed further.

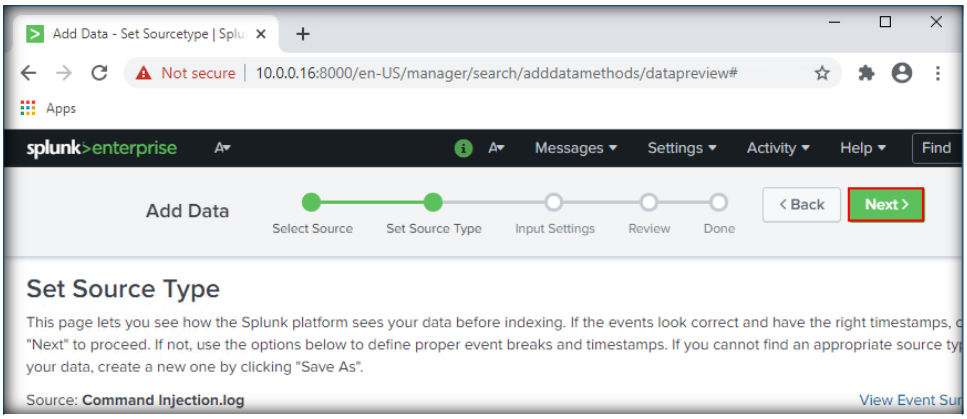

FIGURE 1.113: Click Next on Set Source Type section

137. In the next step, the **Input Settings** section will appear. Click **Review** to proceed further.

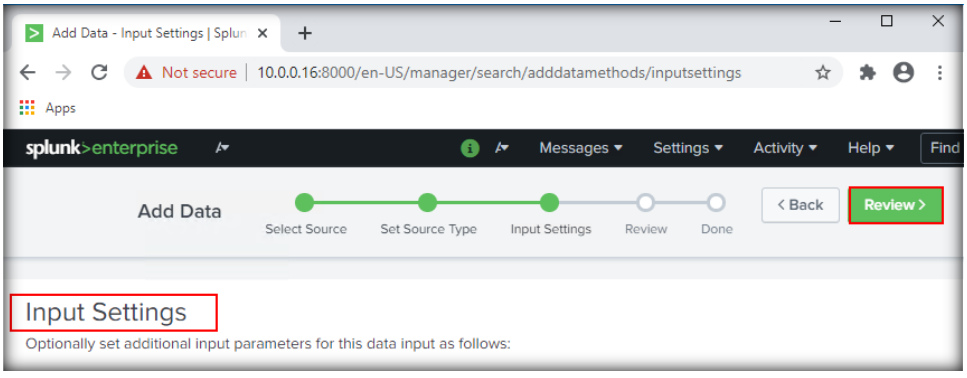

FIGURE 1.114: Click on Review in the Input Settings section

138. In the next step, you will see the **Review** section. Ensure that the details under the **Review** section are correct and then click **Submit** to proceed further.

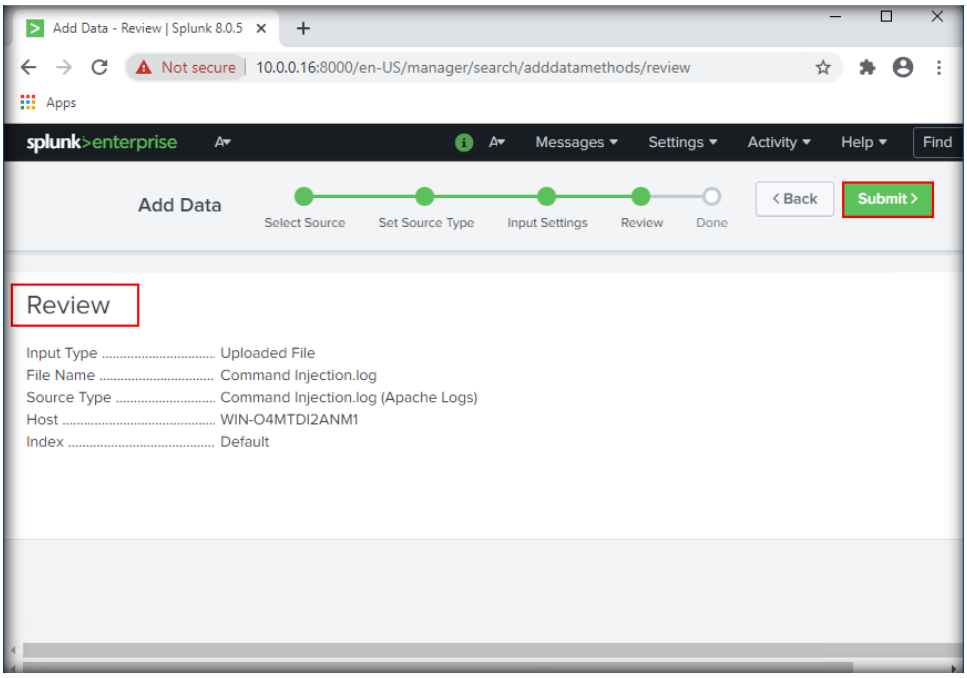

FIGURE 1.115: Review the details and click Submit

139. Upon clicking **Submit**, the application window will display the following message: **File has been uploaded successfully**. Now, click **Start Searching** to fetch the log entries from the uploaded log file.

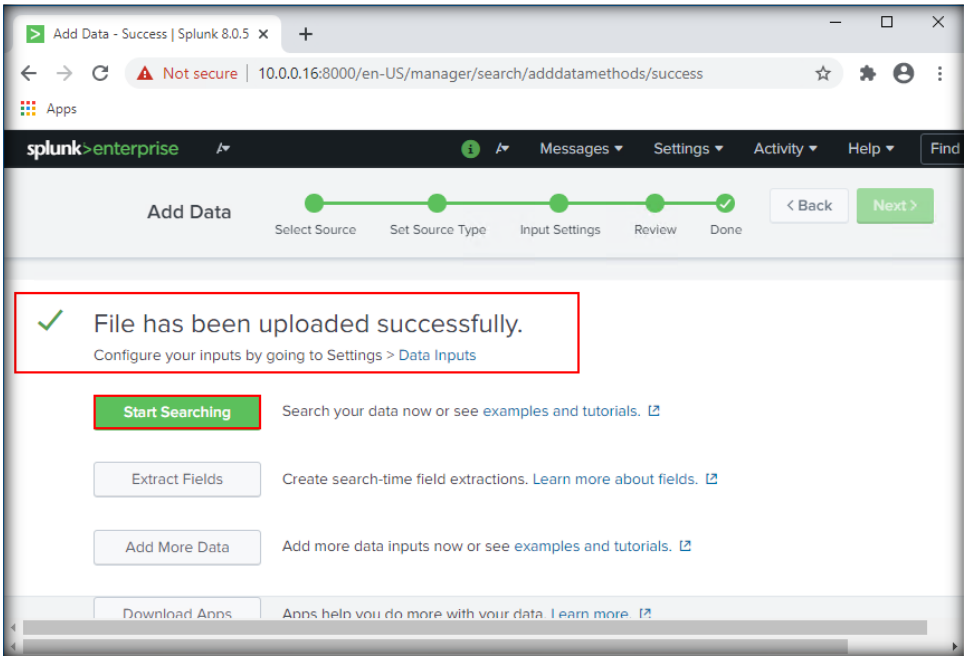

FIGURE 1.116: Click Start Searching

140. The log entries from the **Command Injection.log** file will now be displayed in the **Splunk Enterprise** window under the **Events** tab.

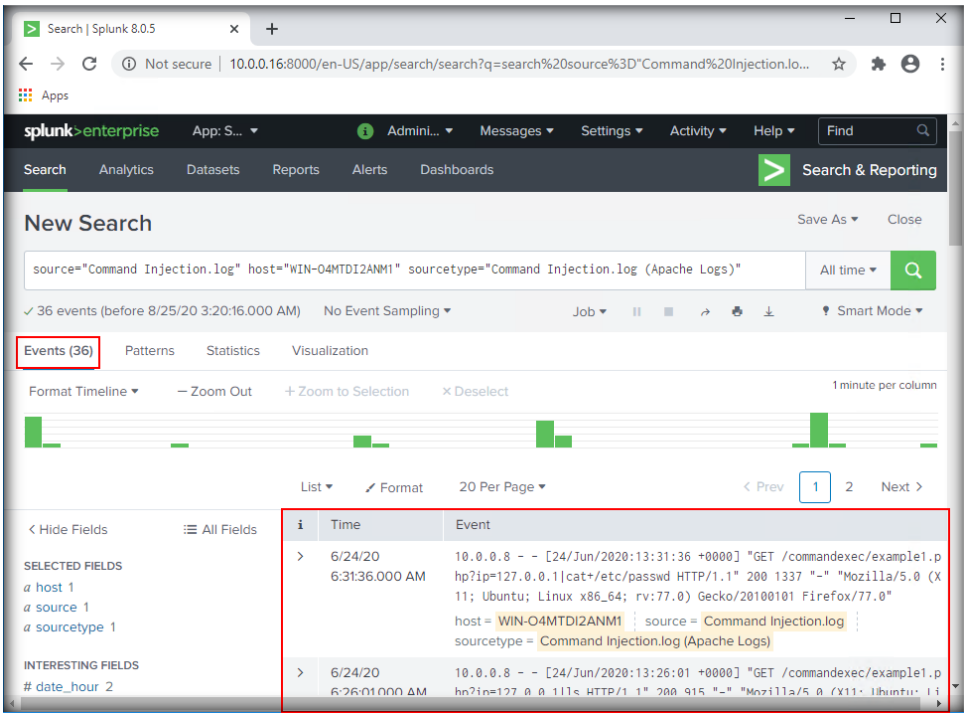

FIGURE 1.117: Log file entries displayed in Splunk

141. We will now apply the plain-text filters **|ifconfig** and **|cat** to find any indicators of a command injection attack. However, we see that no results are obtained after applying these filters, as shown in the screenshots below:

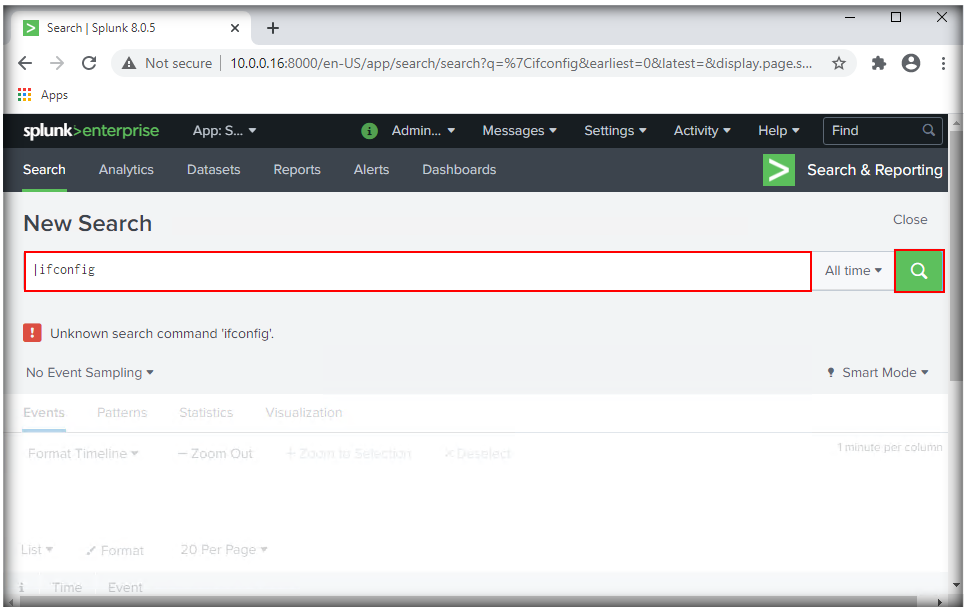

FIGURE 1.118: | ifconfig filter

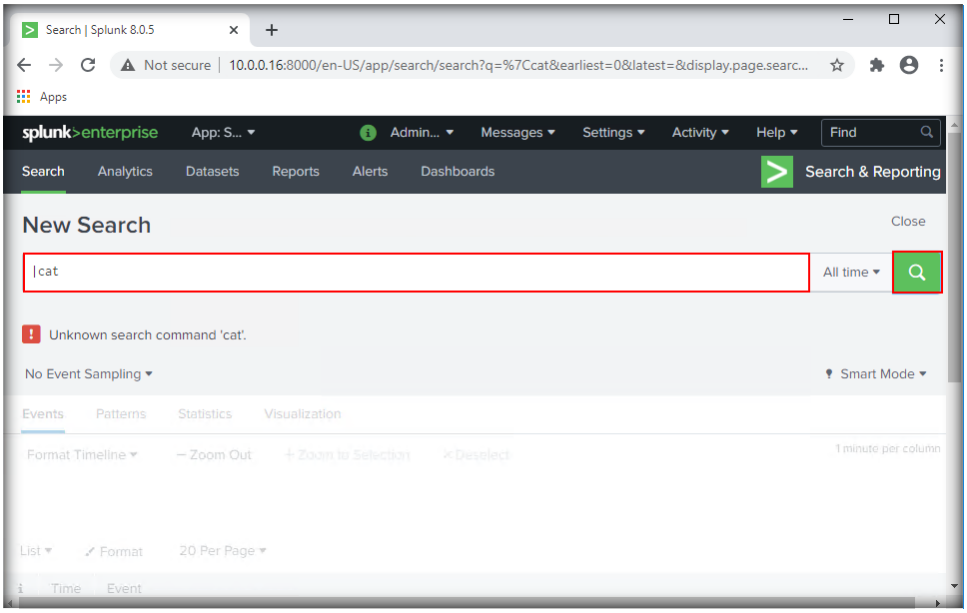

FIGURE 1.119: |cat filter

Note: Upon applying the above-mentioned filter, you might view entries from previously sourced log files. In such a case, you need to specify the current log file along with the search filter, for example, **|ifconfig sourcetype="Command Injection.log (Apache Logs)"** or **|cat sourcetype="Command Injection.log (Apache Logs)"**, to fetch relevant results.

Note: In the above scenario, we have used the **Pipe (|)** operator in combination with **Linux** specific commands as a means of applying filters in an attempt to identify indicators of command injection attacks, as the command injection log file was generated on a **Linux** machine. In case the log file was generated on a **Windows** machine, we would use the **Pipe (|)** operator in combination with **Windows** specific commands as a means of applying filters.

142. Since applying the above filters has not yielded any result, we now have to manually look for log entries containing signs of a **command injection** attack. For demonstrative ease, since this lab uses log files that are specific only to the recorded web attacks, we will examine the first/topmost log entry for signs of a **command injection** attack.

Note: In a real-time scenario, you might have to scroll down the application window to manually look for the log entries that reveal the indicators of a **command injection** attack.

143. So go back to the previous page that showed log entries under the events tab upon uploading the Command Injection.log file. Upon examining the topmost log entry, we see that its query string contains the **Pipe (|)** operator in combination with the **cat (concatenate)** command, which is a shell command used on Linux machines. A **Pipe (|)** operator used in combination with any of the common **Linux** shell commands acts as a vector for **command injection** attacks.

144. The query string also reveals that the **concatenate** command has been used to access the **/etc/passwd** file, as shown in the screenshot below. The **/etc/passwd** file stores passwords of user accounts on **Linux** systems. These findings are strong indicators of a **command injection** attack.

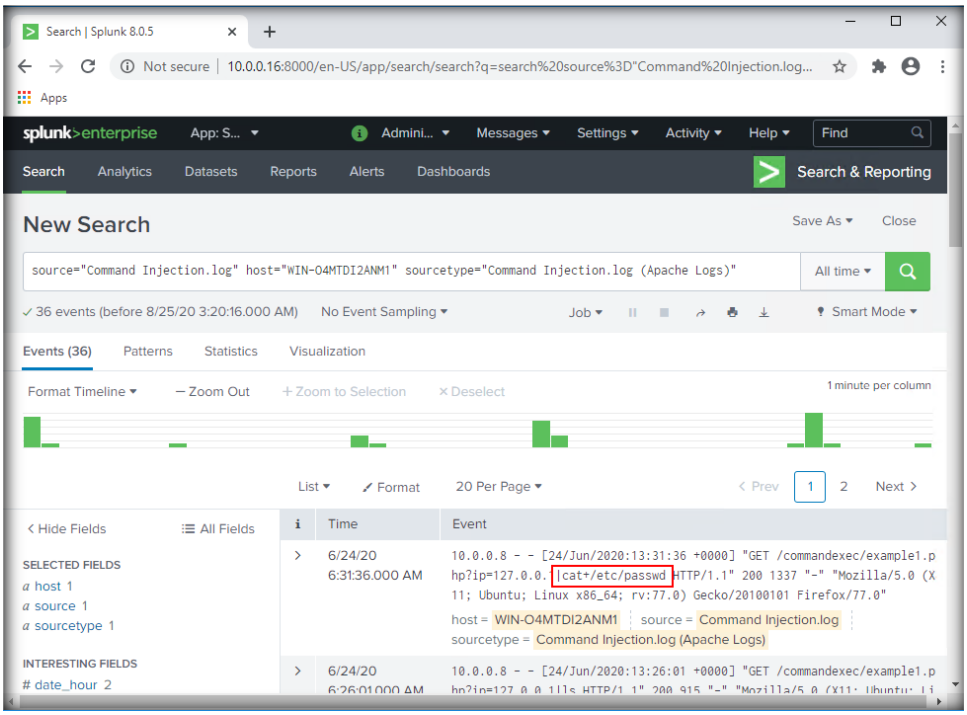

FIGURE 1.120: | cat+/etc/passwd detected

145. A detailed observation of the log entry provides us with the following findings:

   A. Date and time of the attack: **24th June 2020** and **06:31:36 AM**

   B. IP address of the attacker: **10.0.0.8**

   C. Malicious query used in the attack: **GET /commandexec/example1.php?ip=127.0.0.1|cat+/etc/passwd**

   D. HTTP status code **200**: This denotes that the request made to the server has been processed.

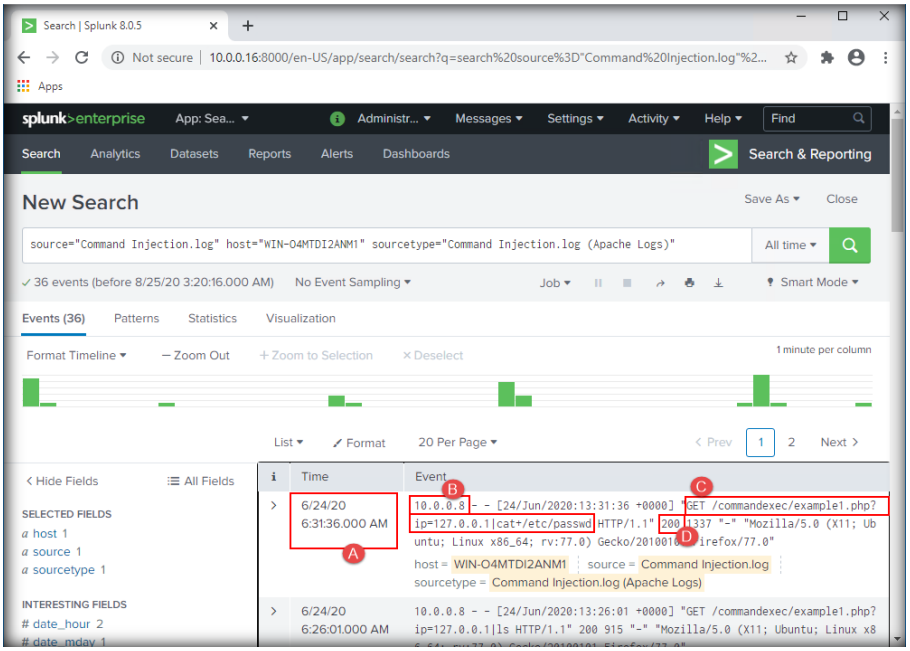

FIGURE 1.121: Detailed analysis of command injection attack

TASK 6

Detect and Examine XML External Entity (XXE) Attack via Log Files

146. We will now examine the **XML External Entity (XXE) Attack** log file generated by **Apache** server

147. Click **splunk>enterprise** at the top left of the webpage

148. You will now be directed to the homepage of **Splunk Enterprise**. We now need to upload the **XXE Attack.log** file from the **Apache Logs** folder. To be able to upload the evidence file, click on the **Add Data** icon on the homepage and then follow the steps for uploading it.

149. The **Open** window will now appear. Navigate to **C:\CHFI-Tools\Evidence Files\Log Files\Apache Logs** and select **XXE Attack.log**. Click **Open** to upload the file.

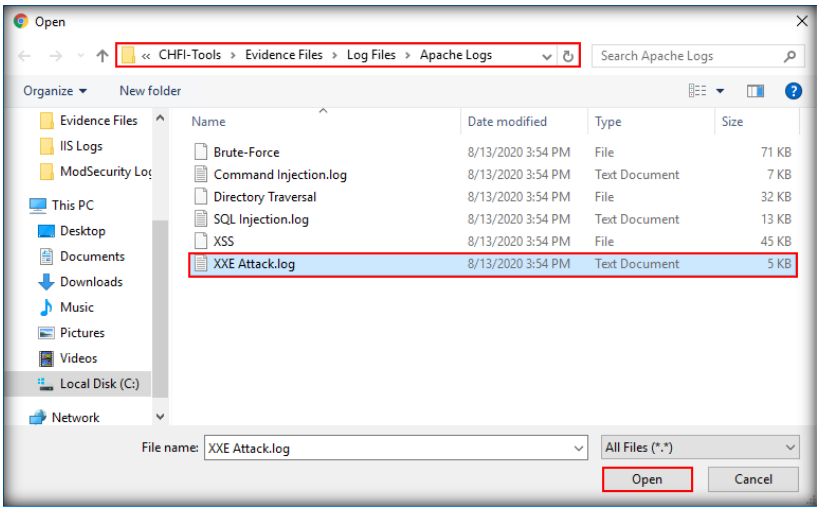

FIGURE 1.122: Select XXE Attack.log and click Open

150. The **XXE Attack.log** file will be uploaded successfully. Click **Next** to continue

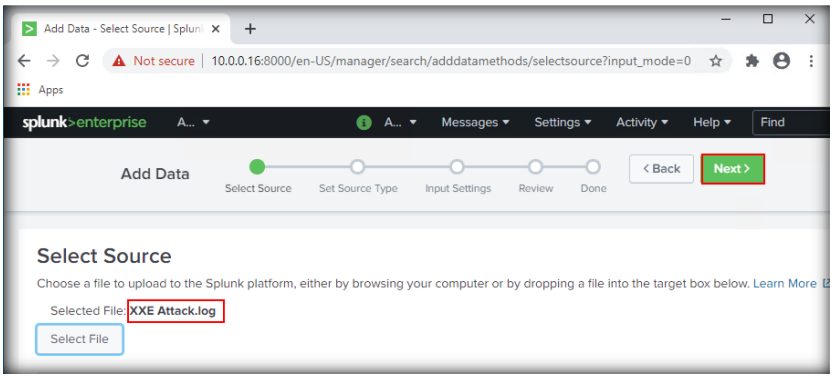

FIGURE 1.123: Click Next

151. In the next step, you will see the **Set Source Type** section. Click **Save As**.

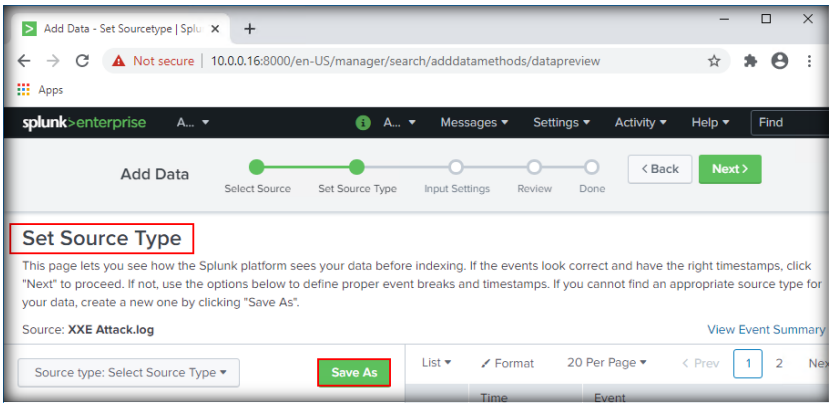

FIGURE 1.124: Click Next

152. The **Save Source Type** window will now appear. In the **Name** field, enter **XXE Attack.log (Apache Logs)** and then click **Save** to continue.

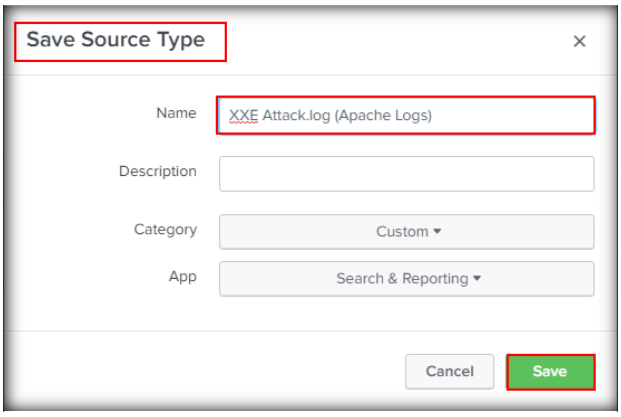

FIGURE 1.125: Enter details in Save Source Type window and click Save

153. You will now get back to the **Set Source Type** section again. Click **Next** to proceed further.

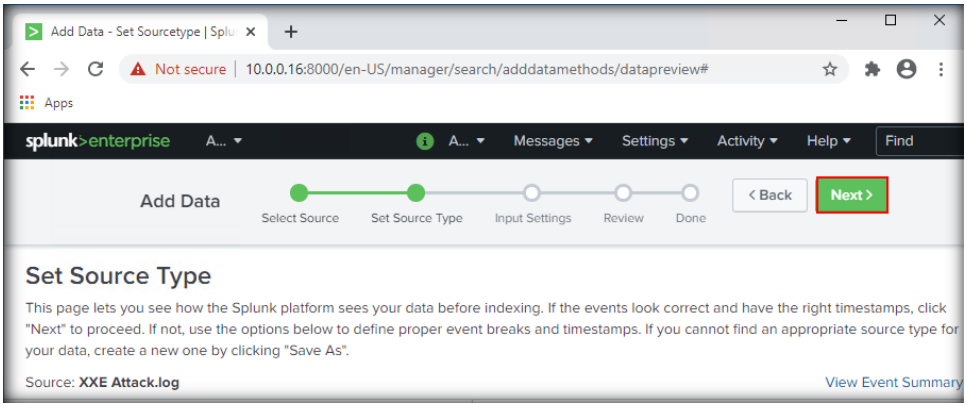

FIGURE 1.126: Click Next in Set Source Type section

154. In the next step, you will see the **Input Settings** section. Click **Review** to proceed further.

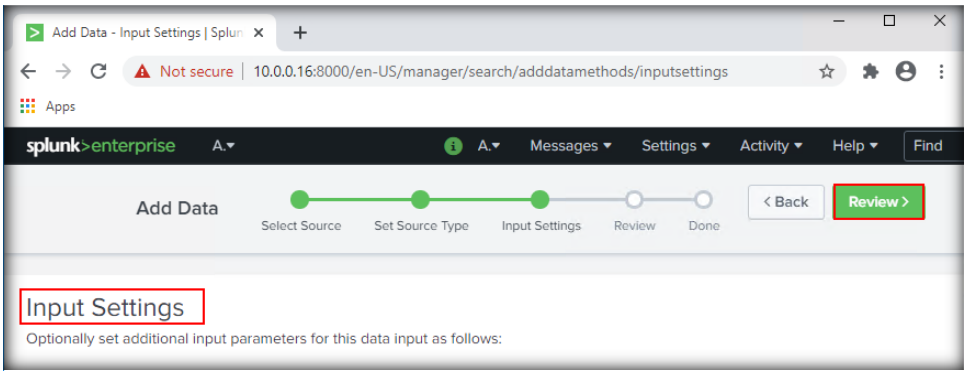

FIGURE 1.127: Click on Review in the Input Settings section

155. In the next step, you will see the **Review** section. Ensure that the details under the **Review** section are correct, and then click **Submit** to proceed further.

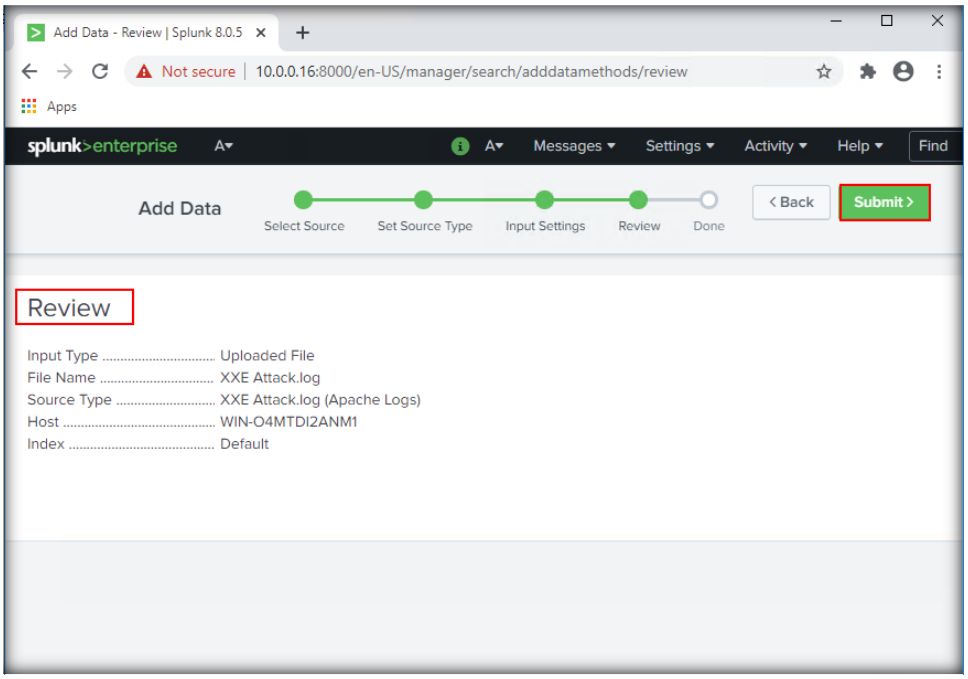

FIGURE 1.128: Review the details and click Submit

156. Upon clicking **Submit**, the application window will display the following message: **File has been uploaded successfully**. Now, click **Start Searching** to fetch all log entries from the uploaded log file.

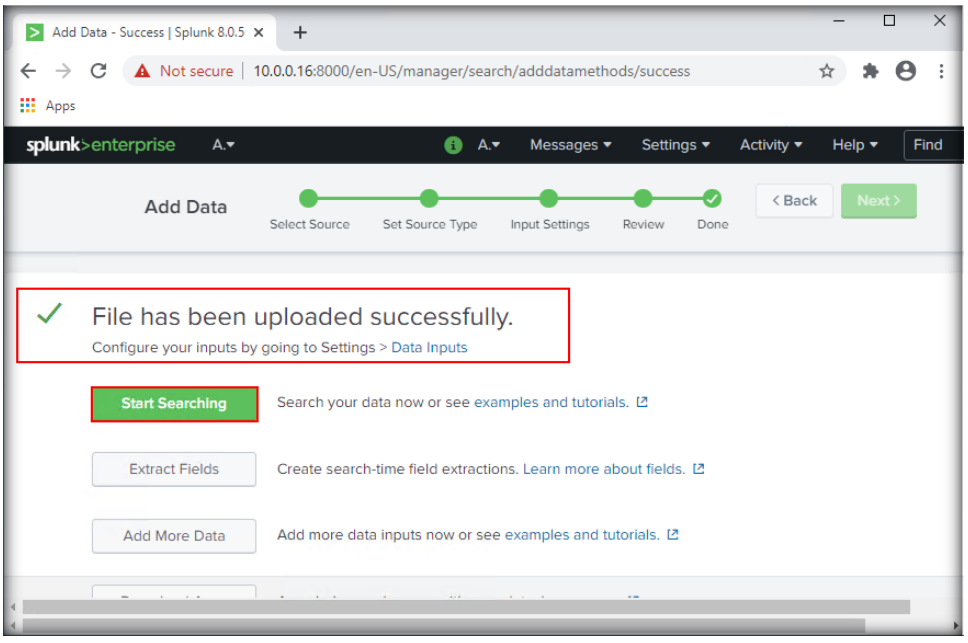

FIGURE 1.129: Click Start Searching

157. You will now see the log entries from the uploaded file displayed in **Splunk Enterprise** window under the **Events** tab, as shown in the screenshot below:

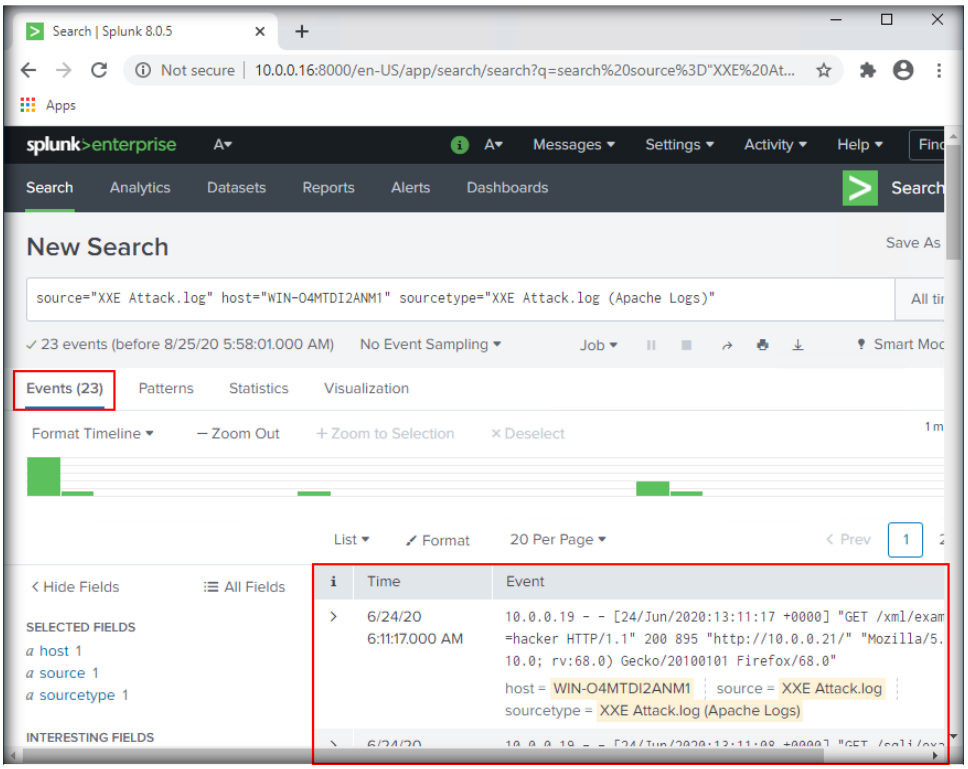

FIGURE 1.130: Log entries displayed in Splunk

158. To look for the occurrence an XXE attack, we will apply the plain-text filter **<!Doctype**. To do this, type **<!Doctype** in the **New Search** field and then click on the **Search** icon. However, as shown in the screenshot below, we find that no results are obtained after applying this filter.

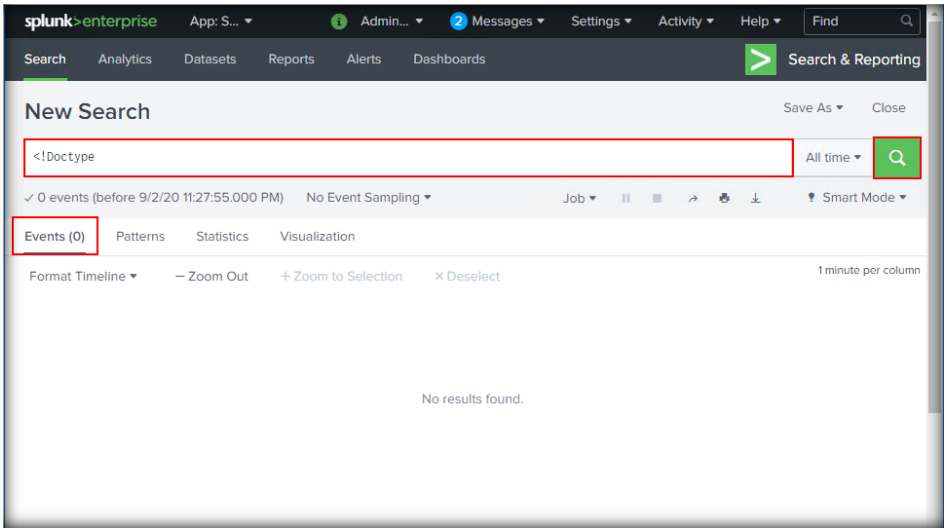

FIGURE 1.131: Use filter <!Doctype

Note: Upon applying the above-mentioned filter, you might view entries from previously sourced log files. In such a case, you need to specify the current log file along with the search filter, for example, **<!Doctype sourcetype="XXE Attack.log (Apache Logs)"**, to fetch relevant results.

159. Since the above plain-text filter did not yield any results, we will now apply its encoded variant. To apply the encoded variant of the filter, type **%3C!Doctype** in the **New Search** field and then click the **Search** icon. Upon applying the encoded filter, we see that a few log entries are retrieved and displayed under the **Events** tab, as shown in the screenshot below:

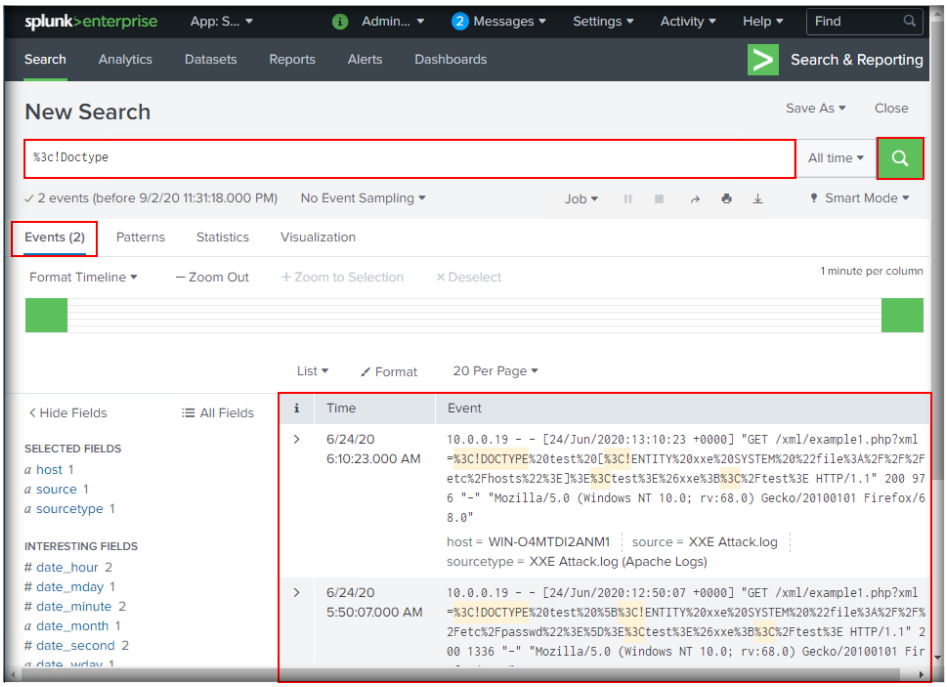

FIGURE 1.132: Results obtained using filter

Note: Upon applying the above-mentioned filter, you might view entries from previously sourced log files. In such a case, you need to specify the current log file along with the search filter, for example, **%3C!Doctype sourcetype="XXE Attack.log (Apache Logs)"**, to fetch relevant results.

Note: For the ease of demonstration, we have used a log file containing entries specific only to recorded web attacks. In real-time scenarios, however, you might see several log entries upon applying a filter and you might have to scroll down the window to manually look for the log entry containing the indicators of an **XXE** attack.

160. Upon examining the topmost log entry, we find that the query string in it contains an encoded XML input which reads **xml=%3C!DOCTYPE%20test%20[%3C!ENTITY%20xxe%20SYSTEM%20%22file%3A%2F%2F%2Fetc%2Fhosts%22%3E]%3E%3Ctest%3E%26xxe%3B%3C%2Ftest%3E**, as highlighted in the screenshot below:

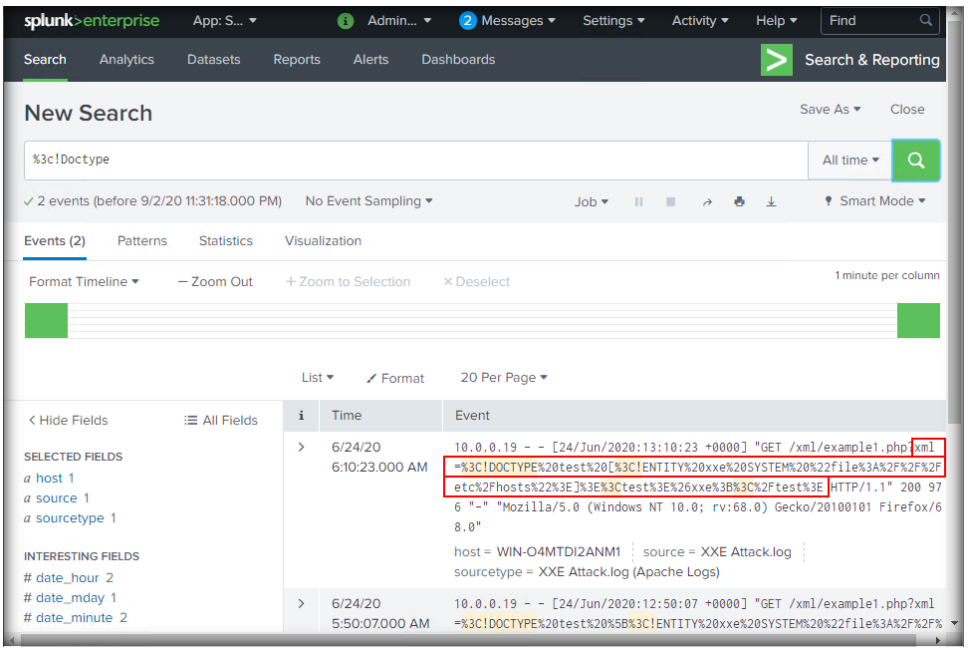

FIGURE 1.133: Malicious XML input

161. When we decode the above encoded XML input using the decoder table provided previously in the lab, it is interpreted as: **xml=<!DOCTYPE test [<!ENTITY xxe SYSTEM "file:///etc/hosts">]><test>&xxe;</test>**. The XML input shown in the query string above is designed to fetch the **/etc/hosts** file of a **Linux** system. The **/etc/hosts** file stores information pertaining to mapping of hostnames with their IP addresses on a network. From this, it can be inferred that an **XXE** attack has occurred and the attacker has attempted to gain access to the **/etc/hosts** file.

Note: You may also apply the above encoded filter as **%3c!Doctype %3c!Entity**

162. A detailed observation of the log entry provides us with the following findings:

    A. Date and time of the attack: **24th June 2020** and **06:10:23 AM**

    B. IP address of the attacker: **10.0.0.19**

C. Malicious query used to perform the attack: **"GET /xml/example1.php?xml=%3C!DOCTYPE%20test%20[%3C!ENTITY%20xxe%20SYSTEM%20%22file%3A%2F%2Fetc%2Fhosts%22%3E]%3E%3Ctest%3E%26xxe%3B%3C%2Ftest%3E**

D. HTTP status code **200**: This indicates that the request made to the server has been processed.

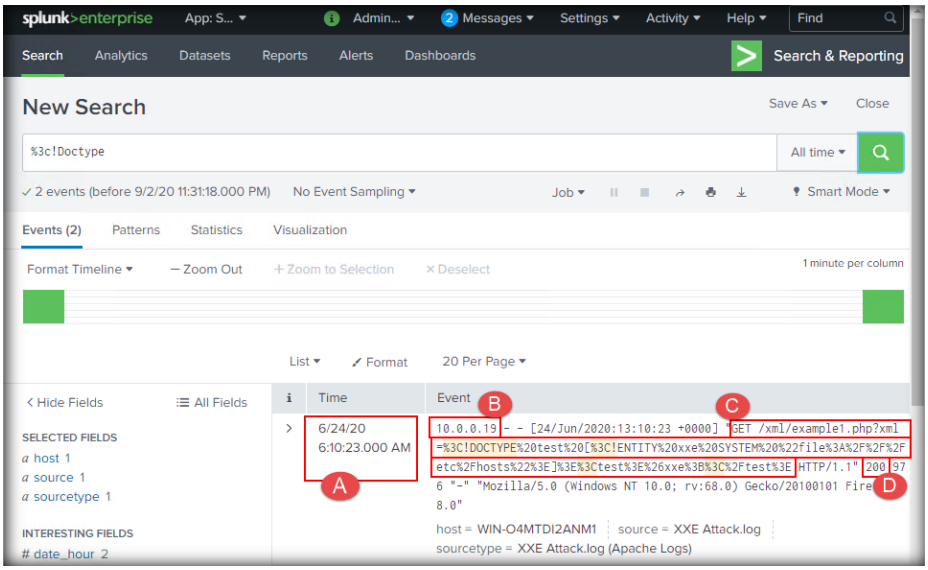

FIGURE 1.134: Detailed analysis of XXE attack

TASK 7

Detect and Examine Brute-force Attack via Log Files

163. We will now examine **brute-force** attack log files generated from the **Apache** server.

164. Click **splunk>enterprise** at the top left of the webpage to go to the homepage of **Splunk Enterprise**. We now need to upload the **Brute-Force** log file from the **Apache Logs** folder. Click the **Add Data** icon on the homepage and then follow the steps for uploading it.

165. On the **Open** window, navigate to **C:\CHFI-Tools\Evidence Files\Log Files\Apache Logs**, select **Brute-Force**, and click **Open**

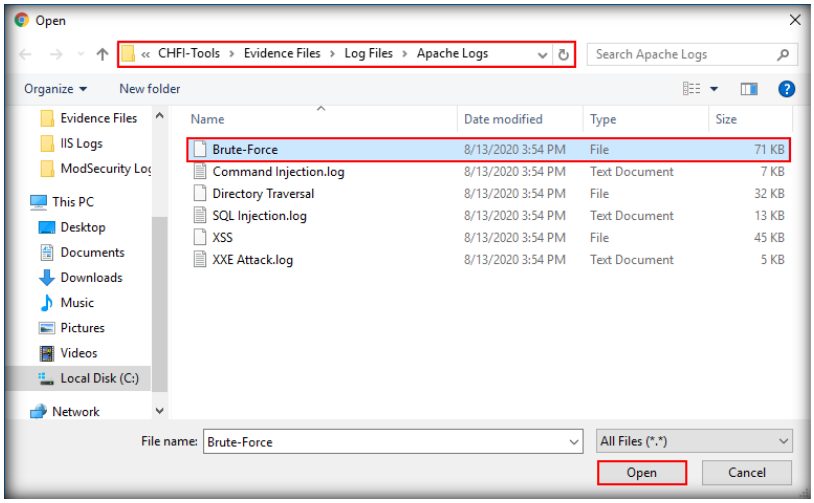

FIGURE 1.135: Select Brute-Force file and click Open

166. The **Brute-Force** file will be uploaded successfully. Click **Next** to proceed further.

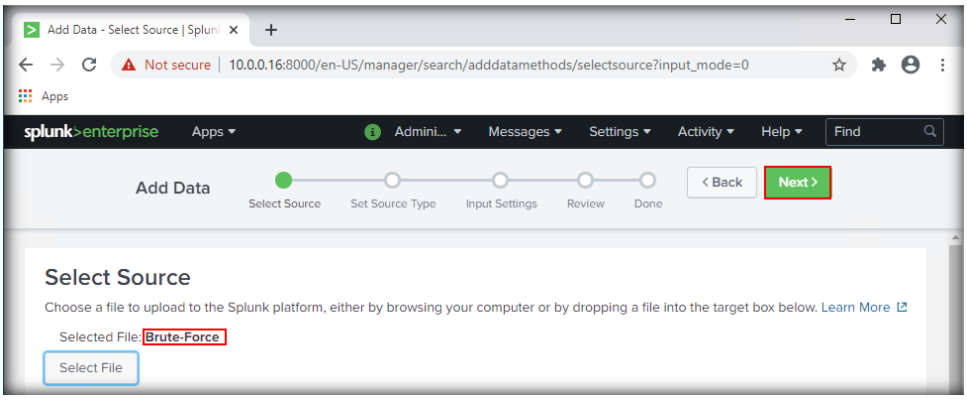

FIGURE 1.136: Click Next

167. In the next step, you will see the **Set Source Type** section. Click **Save As**.

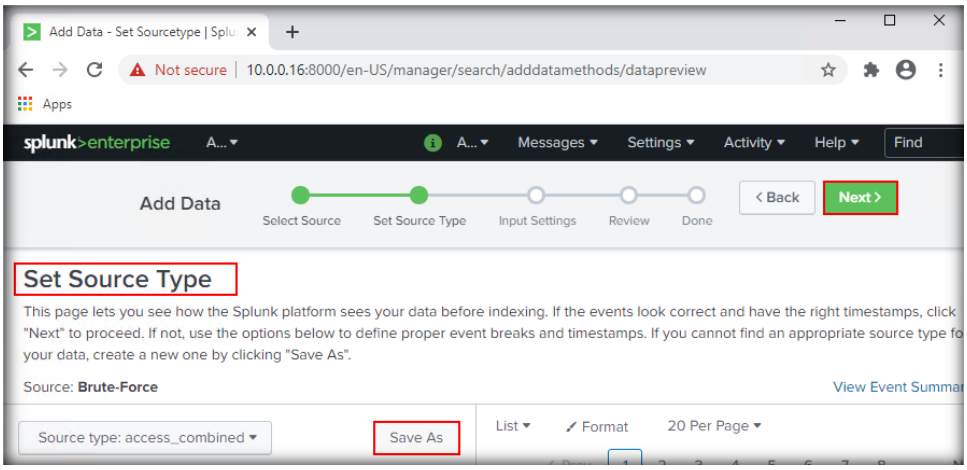

FIGURE 1.137: Click Save As in Set Source Type section

168. The **Save Source Type** window will now appear. In the **Name** field, enter **Brute-Force (Apache Logs)**. Click **Save** to continue.

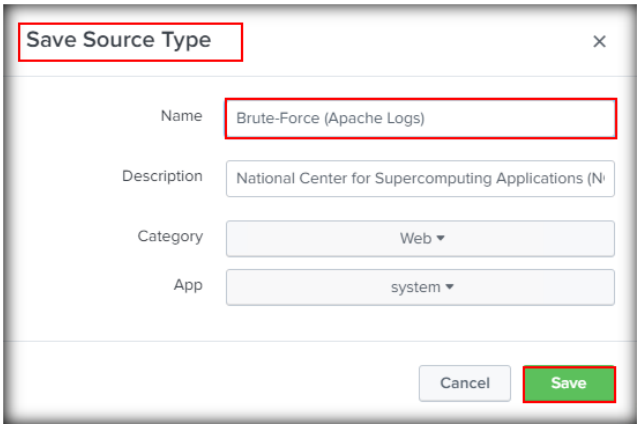

FIGURE 1.138: Enter details in Save Source Type window and click Save

169. You will now get back to the **Set Source Type** section again. Click **Next** to proceed further.

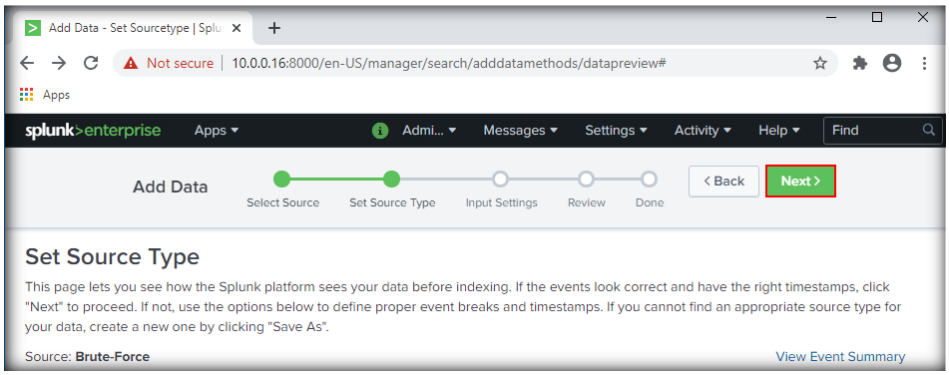

FIGURE 1.139: Click Next in Set Source Type section

170. In the next step, you will see the **Input Settings** section. Click **Review** to proceed further.

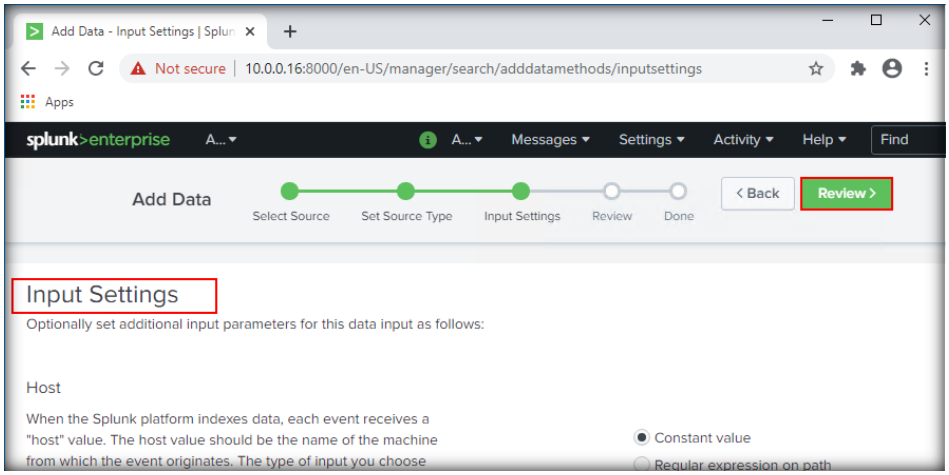

FIGURE 1.140: Click on Review in the Input Settings section

171. In the next step, you will see the **Review** section. Ensure that the details under the **Review** section are correct and then click **Submit** to proceed further.

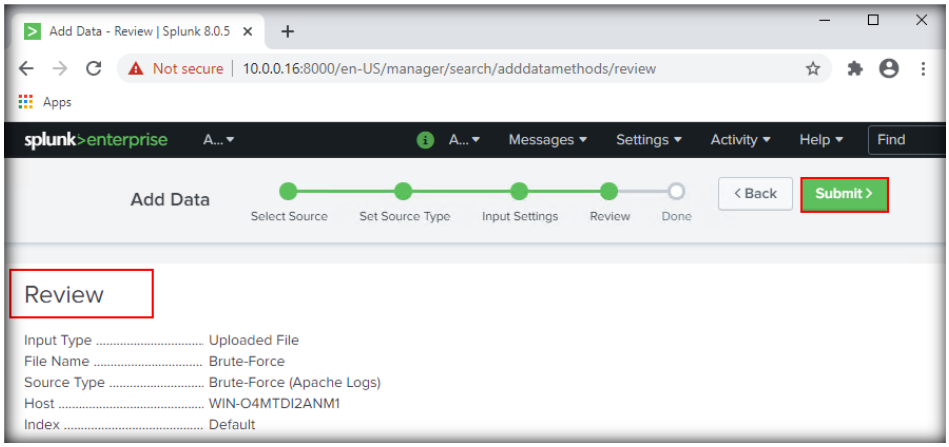

FIGURE 1.141: Review the details and click Submit

172. Upon clicking **Submit**, the application window will display the following message: **File has been uploaded successfully**. Now, click **Start Searching** to fetch the log entries from the uploaded log file.

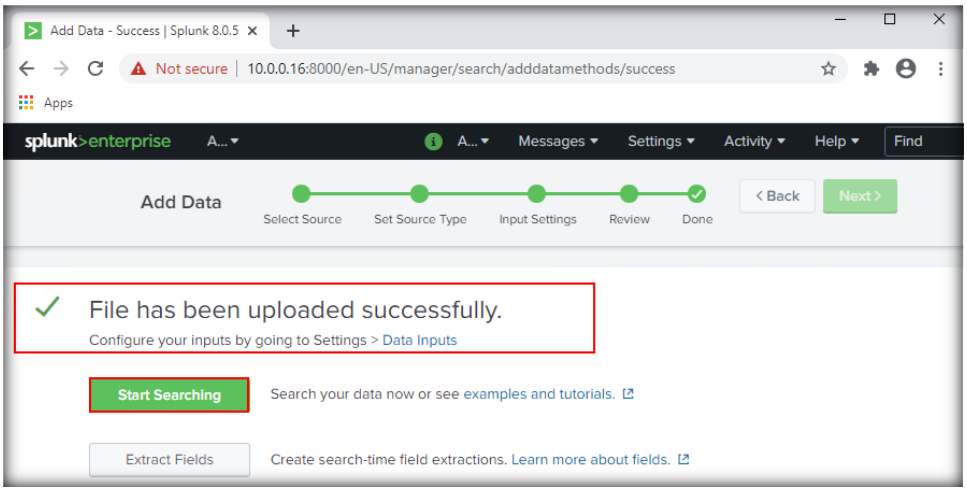

FIGURE 1.142: Click Start Searching

173. You will now see the log entries from the uploaded log file displayed in the **Splunk Enterprise** window under the **Events** tab, as shown in the screenshot below:

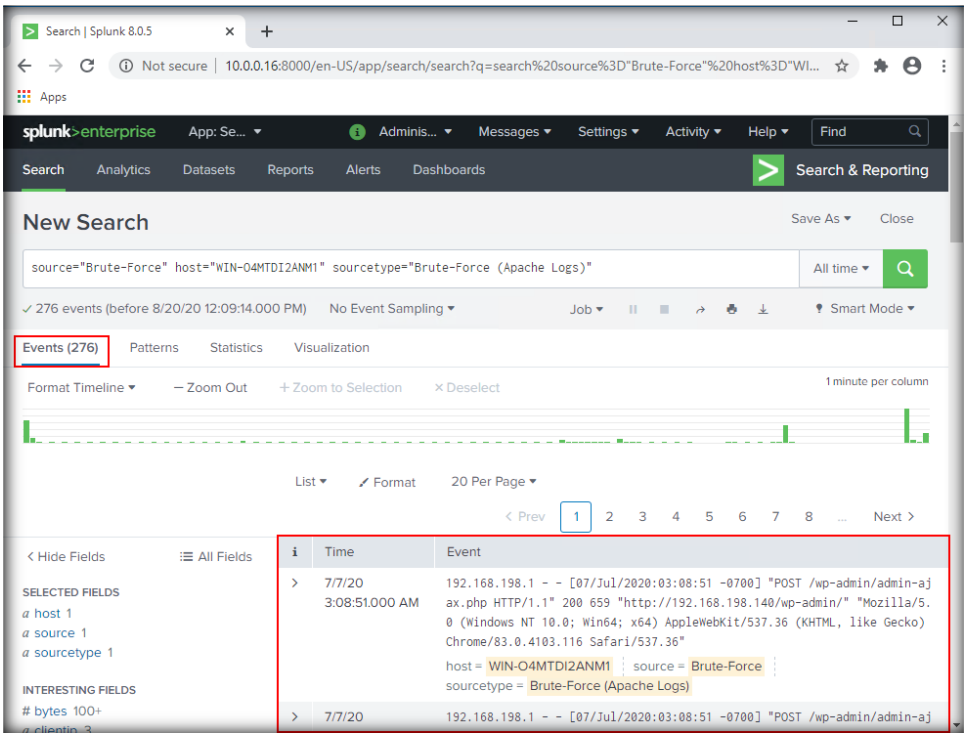

FIGURE 1.143: Log entries displayed in Splunk

174. The fetched log entries indicate that a **WordPress** website has been targeted. The screenshot below shows two highlighted areas marked as **1** and **2**.

175. The highlighted area marked by **1** represents the log entries that collectively reveal multiple/repeated login attempts from the IP address **192.168.198.1** on the webpage **/wp-login.php**, which is hosted on a server with the IP address **192.168.198.140**. All these login attempts have occurred within a very short timeframe.

176. In the highlighted area marked by **2**, we can see that the webpage identifier in the first line of the query string, which earlier reflected as **/wp-login.php** preceded by a **POST** request, is now changed to **/wp-admin/** preceded by a **GET** request, which indicates that an attacker has succeeded in gaining access to the **admin** account of the website after several login attempts. This is a strong indicator of a brute-force attack.

    Note: You need to scroll down the application window to find the multiple log entries described above.

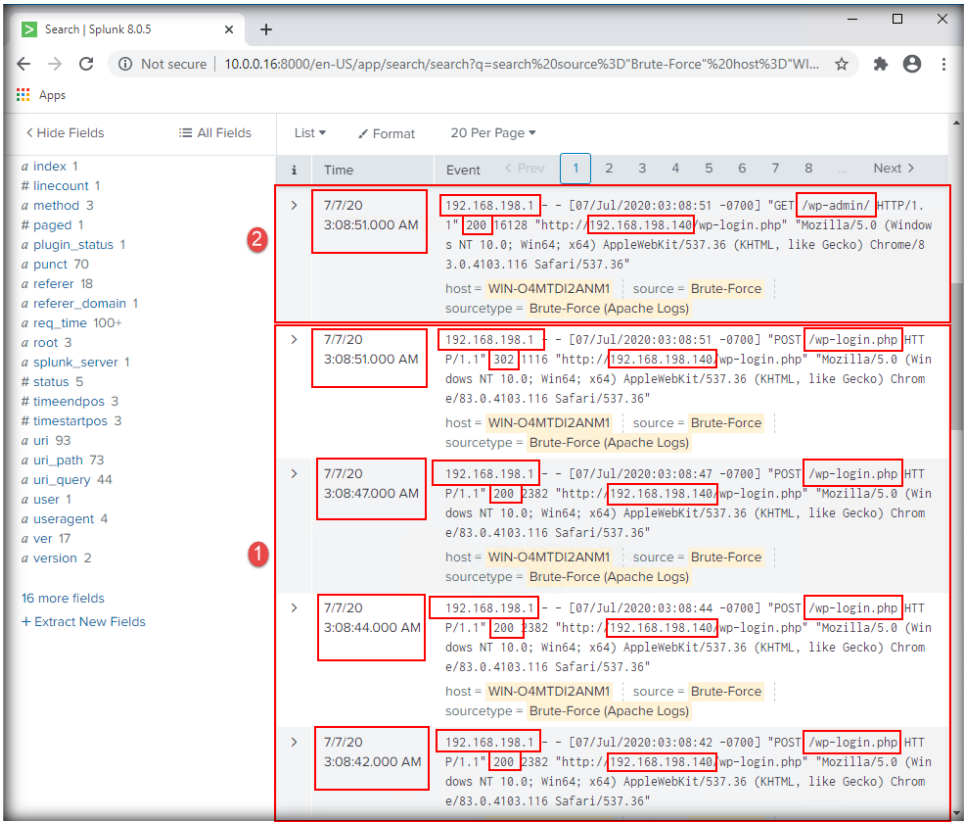

FIGURE 1.144: Detailed observation of brute-force attack

Note: For the ease of demonstration, since we have used a log file specific to recorded web attacks in this lab, we are able to find the indicators of a brute-force attack in the first few log entries from the top. In real-time, however, you might have to scroll down the application window and manually look for those multiple log entries that reveal the occurrence of a brute-force attack.

177. We can summarize our observations from the above as follows:

(i) Date and times at which the brute-force attack was attempted: **07th July 2020**, **03:08:42 AM**, **03:08:44 AM**, **03:08:47 AM**, and **03:08:51 AM**

(ii) Date and time at which the attacker successfully gained admin privileges: **07th July 2020**, **03:08:51 AM**

(iii) IP Address of the attacker: **192.168.198.1**

(iv) IP address of the Host: **192.168.198.140**

(v) HTTP status code **200**: This indicates that the request made to the server has been processed

(vi) HTTP status code **302**: This indicates URL redirection.

(vii) The webpage on which repeated logins have been attempted using **POST** request: **/wp-login.php**

(viii) The webpage identifier which indicates that the attacker has successfully gained admin privileges using **GET** request: **/wp-admin/**, in the first line of the query string at the topmost log entry marked by the highlighted area **2** in the above screenshot.

178. We will now examine the brute-force attack log file generated by **IIS** server.

179. Click **splunk>enterprise** at the top left of the webpage. You will be directed to the homepage of **Splunk Enterprise**. We now need to upload the **Brute-Force.txt** log file from **IIS Logs** folder. To upload the evidence file, click on the **Add Data** icon on the homepage and then follow the steps for uploading it.

180. The **Open** window will now appear. Navigate to **C:\CHFI-Tools\Evidence Files\Log Files\IIS Logs** and select **Brute-Force.txt**. Click **Open** to upload the file.

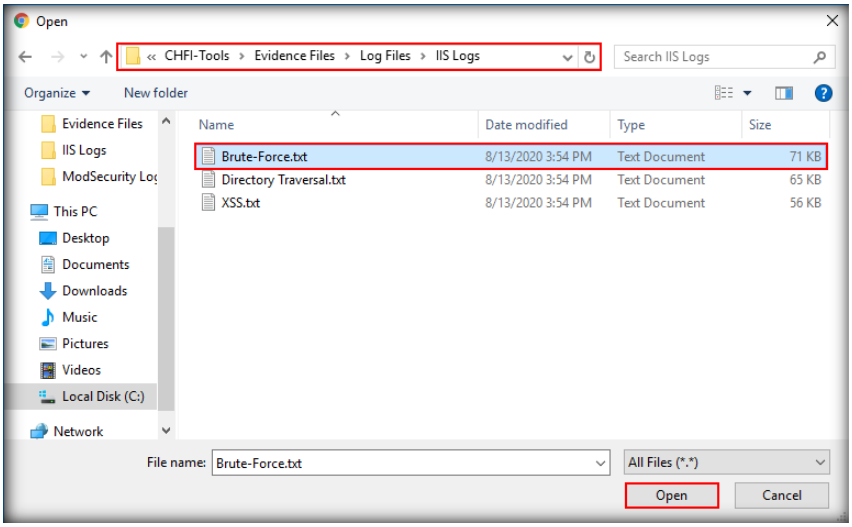

FIGURE 1.145: Select Brute-Force.txt and click Open

181. The **Brute-Force.txt** file will be uploaded successfully. Click **Next** to proceed further.

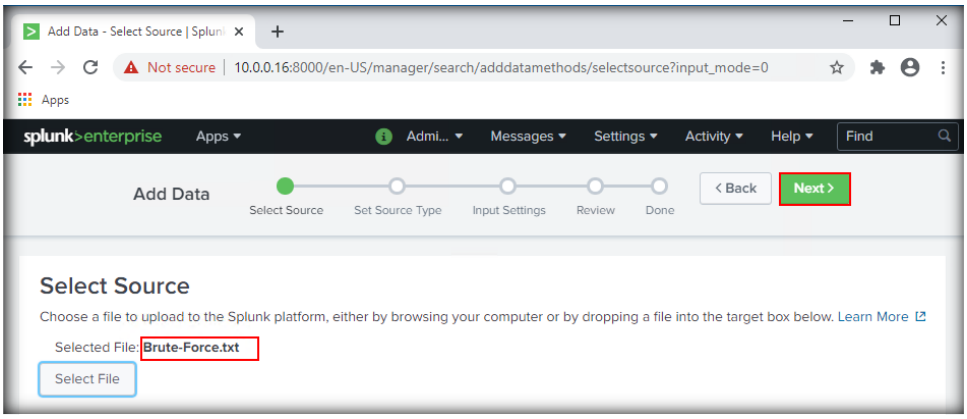

FIGURE 1.146: Click Next

182. In the next step, you will see the **Set Source Type** section. Click **Save As**.

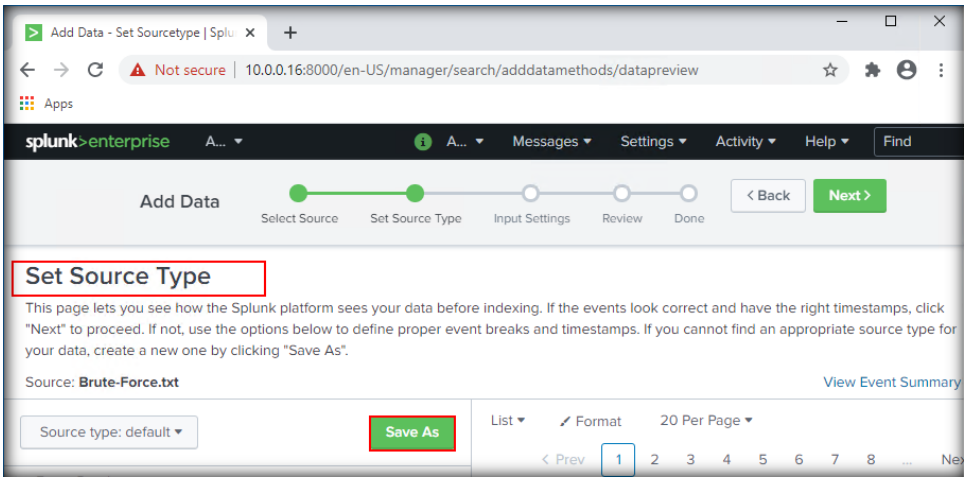

FIGURE 1.147: Click Next in Set Source Type section

183. The **Save Source Type** window will now appear. In the **Name** field, enter **Brute-Force.txt (IIS Logs)**. Click **Save** to continue.

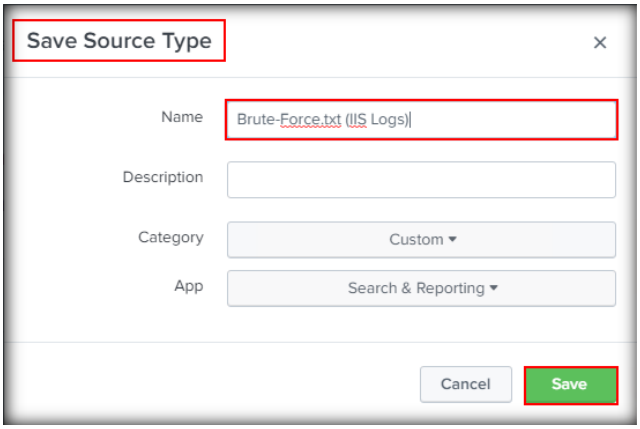

FIGURE 1.148: Enter details in Save Source Type window and click Save

184. You will now get back to the **Set Source Type** section again. Click **Next** to proceed further.

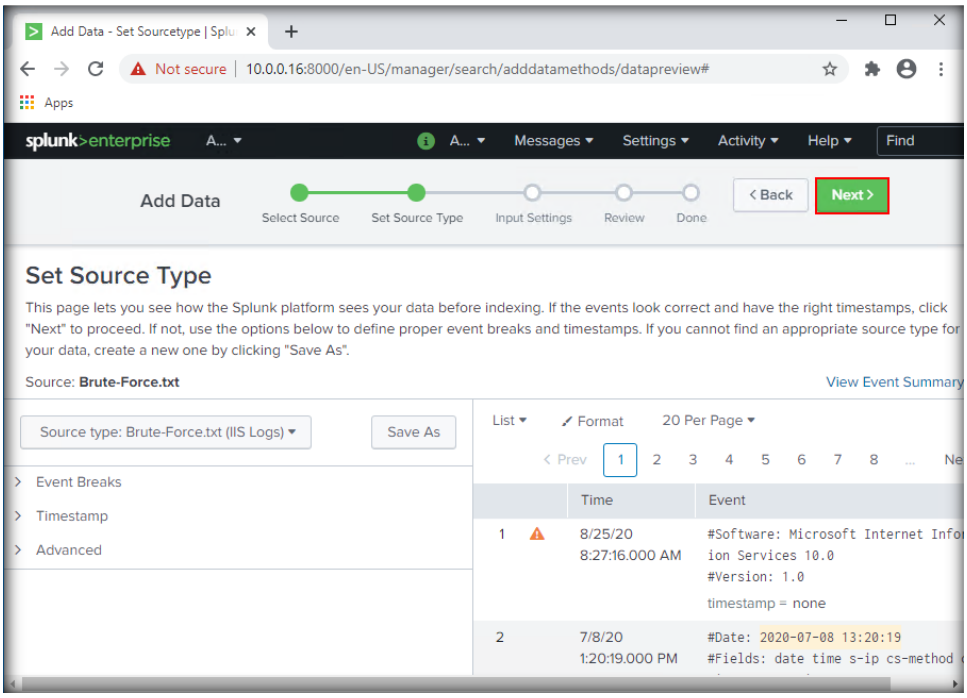

FIGURE 1.149: Click Next in Set Source Type section

185. In the next step, you will see the **Input Settings** section. Click **Review** to proceed further.

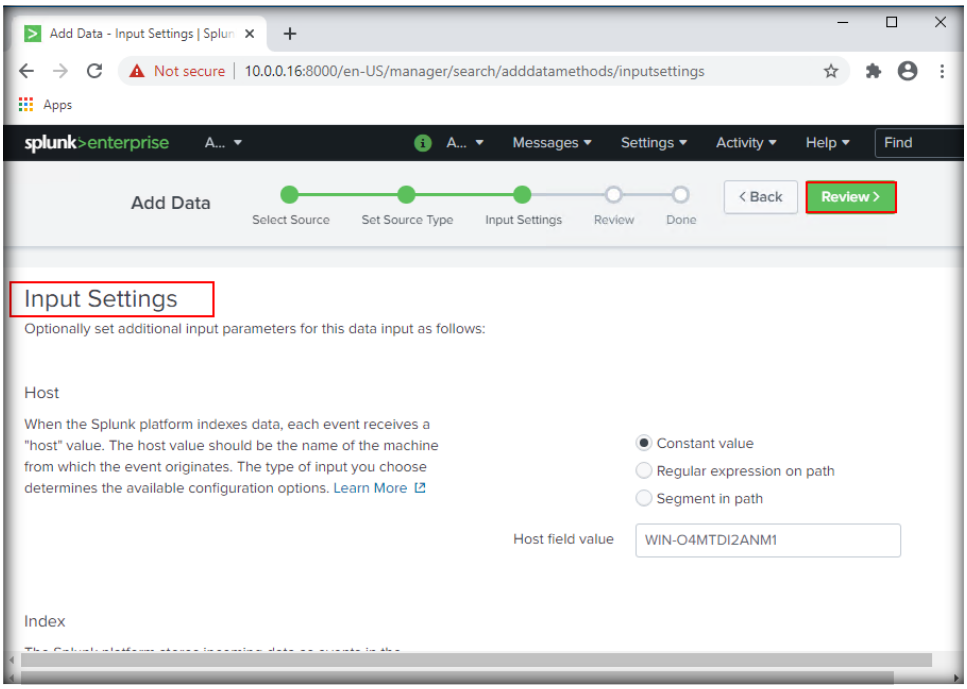

FIGURE 1.150: Click on Review in the Input Settings section

186. In the next step, you will see the **Review** section. Ensure that the details under the **Review** section are correct and then click **Submit** to proceed further.

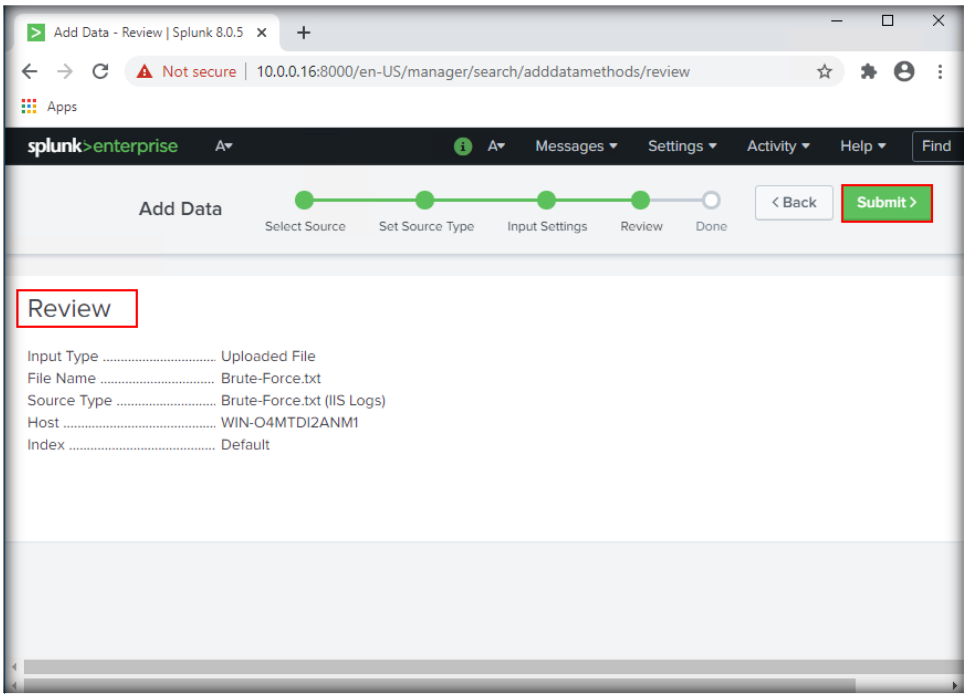

FIGURE 1.151: Review the details and click Submit

187. Upon clicking **Submit**, the application window will display the following message: **File has been uploaded successfully**. Now, click **Start Searching** to fetch the log entries from the uploaded log file.

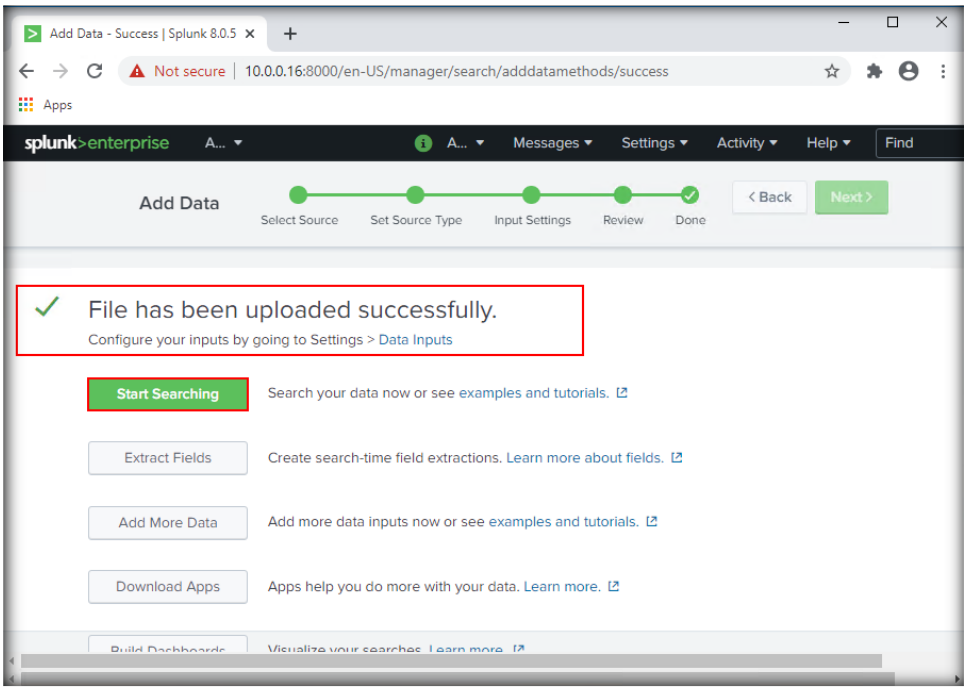

FIGURE 1.152: Click Start Searching

188. You will now see the log entries from the uploaded log file displayed in the **Splunk Enterprise** window under the **Events** tab, as shown in the following screenshot:

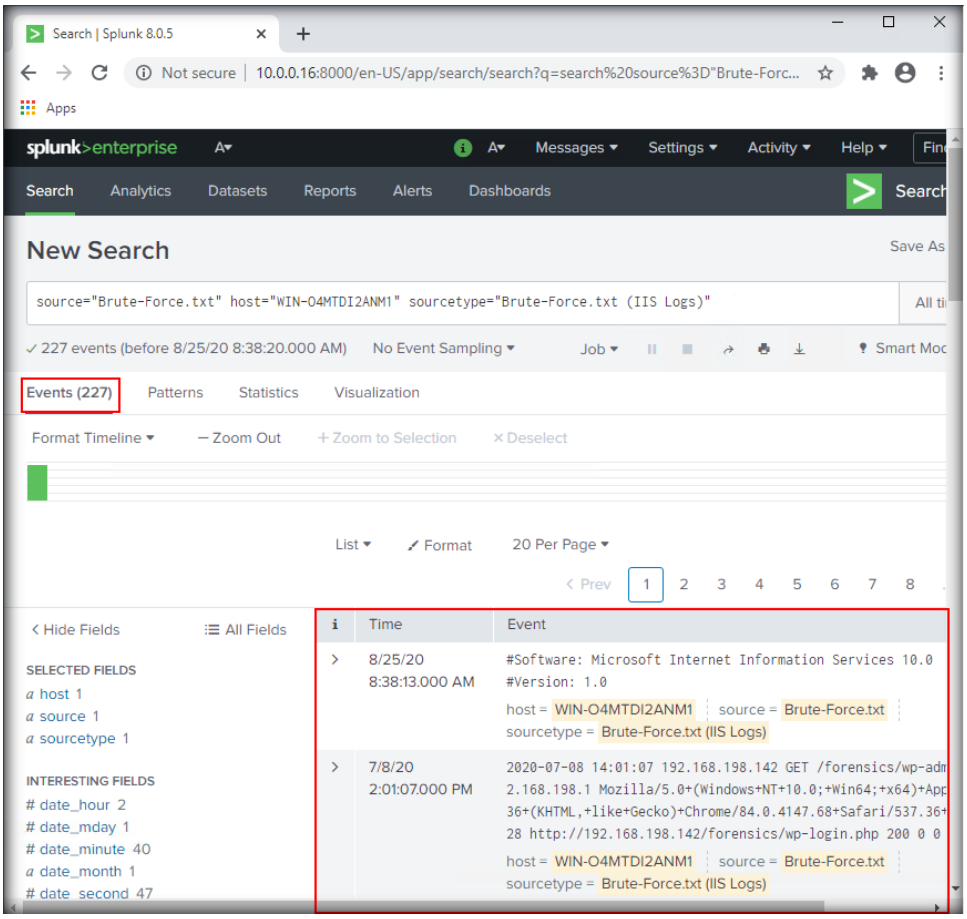

FIGURE 1.153: Log file entries displayed in Splunk

189. The fetched log entries indicate that a **WordPress** website has been targeted. Scroll down the results to find log entries that collectively reveal multiple login attempts, as indicated in the highlighted area **1** in the screenshot below. In all these entries, we see that multiple/repeated login attempts have been made from the IP address **192.168.198.1** on the webpage **/forensics/wp-login.php**, which is hosted on a server with IP address **192.168.198.142**.

190. Similarly, upon examining the log entry marked by highlighted area **2** at the top, we can see that the webpage identifier in the first line of the query string, which previously reflected as **/forensics/wp-login.php** preceded by a **POST** request in highlighted area **1**, is now changed to **/forensics/wp-admin/** preceded by a **GET** request.

191. These observations indicate that the IP address **192.168.198.1** belongs to a malicious user/attacker who has now gained **admin** privileges on the affected **WordPress** site. As seen in the screenshot below, these repeated login attempts have been made within a very short timeframe. Based on this, we can infer that a **brute-force** attack has been attempted.

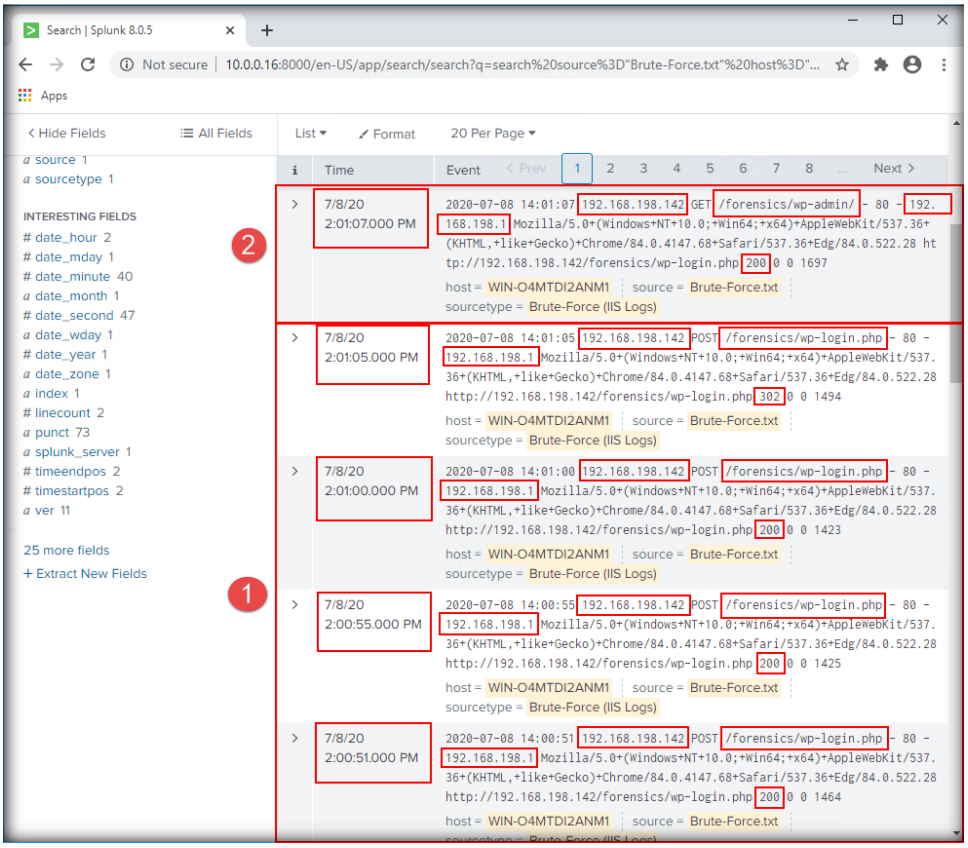

FIGURE 1.154: Detailed analysis of brute-force attack

Note: For the ease of demonstration, since we have used a log file specific to recorded web attacks in this lab, we are able to find the indicators of a **brute-force** attack in the first few log entries from the top. In real-time, however, you might have to scroll down the application window and manually look for those multiple log entries that reveal the occurrence of a **brute-force** attack.

192. We can summarize our observations from the above log entries as follows:

(i) Date and times at which the brute-force attack was attempted: **08th July 2020**, **02:00:51 PM**, **2:00:55 PM, 2:01:00 PM,** and **2:01:05 PM**

(ii) Date and time at which the attacker successfully gained **admin** privileges: **08th July 2020, 02:01:07 PM**

(iii) IP address of the Host Server: **192.168.198.142**

(iv) The webpage on which repeated login attempts were made using **POST** request: **/forensics/wp-login.php**

(v) IP address of the Attacker: **192.168.198.1**

(vi) HTTP status code **200**: This indicates the request made to the server has been processed.

(vii) HTTP status code **302**: This indicates URL redirection.

(viii) The webpage identifier which indicates that the attacker has successfully gained **admin** privileges using **GET** request: **/forensics/wp-admin/** in the first line of the query string at the topmost log entry marked by highlighted area **2** in the above screenshot.

193. In this manner, you can examine the log files from **Apache** and **IIS** servers as well as **ModSecurity** web application firewall for indicators pertaining to various web attacks.

Lab Analysis

Analyze and document the results related to the lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.