DAY 1:

1. Enter a search that returns all web application events that include a purchase action with a web status of 200
   index=test sourcetype=access_combined_wcookie action=purchase status=200

2. Enter a search that returns all web application events that include a purchase action where product ID is WC-SH-G04

   index=test sourcetype=access_combined_wcookie action=purchase productId=WC-SH-G04

3. case 3:The team is only looking for successful purchases, so change your search to only return those.

index=test sourcetype=access_combined_wcookie action=purchase (status>=200 AND status<400) file=success.do

4. The team is only looking for unsuccessful purchases, so change your search to only return those.

index=test sourcetype=access_combined_wcookie action=purchase status=200 file=error.do

5. Status not equal to 200 and got error

index=test sourcetype=access_combined_wcookie action=purchase status>400 file="error.do"

6. You will see fields that do not matter to the team. Use the fields command to only return the action, JSESSIONID and status fields. Does your search run faster using the command?

Ans: index=test sourcetype=access_combined_wcookie action=purchase status=200 file=success.do | fields action, JSESSIONID, status

7. Now tell the difference:

index=test sourcetype=access_combined_wcookie action=purchase status=200 file=success.do | fields - action, JSESSIONID, status

index=test sourcetype=access_combined_wcookie action=purchase status=200 file=success.do | fields- action, JSESSIONID, status

8. Write a query to provide a statistics of the above activity by JSESSIONID

index=test sourcetype=access_combined_wcookie | stats count by JSESSIONID

9. Write a query to provide a statistics of the products purchased by productId. Use the fields command to only return the action, productid and status fields

index=test sourcetype=access_combined_wcookie | fields action, productId, status | stats count by productId

10. Find the statistics of the productid and clientip of all the successful purchases

    index=test sourcetype=access_combined_wcookie action=purchase (status>200 AND status<400)file=success.do | fields action, productId, status, clientip | stats count by productId, clientip

11. Let us assume, the CISO reports that buttercupgames.com is a malicious URL and it is not blocked by the firewall. Please find the list of clientips who have visited the URL "Buttercupgames.com

index=test sourcetype=access_combined_wcookie status=200 referer_domain="http://www.buttercupgames.com" | stats count by clientip | rename count as events | sort –events.

DAY-2

12. Create a table where columns will be action, Jsessionid, status and it will return all the products which were successfully processed and action=purchase.

index=test sourcetype=access_combined_wcookie action=purchase status=200 file=success.do | table action, JSESSIONID, status

13. Change the order of the fields so that JSESSIONID is the first column. (index=test

sourcetype=access_combined_wcookie action=purchase status=200 file=success.do | table JSESSIONID, action, status.

14. Write a query to search, which returns all web application events that include a purchase action and visualize the products statistics with the help of Splunk
15. Write a query to search, which returns all web application events that include a purchase action and create a pie-chart as per the product ID obtained

index=test sourcetype=access_combined_wcookie action=purchase | stats count by productId.

16. Session IDs are called "UserSessions" in the marketing data. Rename JSESSIONID so that your report matches the marketing data

index=test sourcetype=access_combined_wcookie action=purchase status=200 file=success.do | table JSESSIONID, action, status | rename JSESSIONID as UserSessions.

17. Remove the duplicate Usersessions

index=test

sourcetype=access_combined_wcookie action=purchase status=200 file=success.do | dedup

JSESSIONID | table JSESSIONID, action, status | rename JSESSIONID as UserSessions

18. While having action and status fields displayed was nice for a sanity check of the data, the marketing team will not need to have these displayed. Remove them from your table display

index=test

sourcetype=access_combined_wcookie action=purchase status=200 file=success.do | dedup

JSESSIONID | table JSESSIONID | rename JSESSIONID as UserSessions

19. Enter a search that returns all web application events where an item was successfully purchased. Remember that if the success.do file is successfully returned to a user, a purchase was made

index=test sourcetype=access_combined_wcookie status=200 file=success.do

20. Enter a search that returns all web application events where an item was successfully purchased. Remember that if the success.do file is successfully returned to a user, a purchase was made and create a table with unique productid

index=test sourcetype=access_combined_wcookie action=purchase status=200 file=success.do | dedup productId | table productId

Day -3:

1. Use the top command to find the best-selling productIds for all time.

index=test sourcetype=access_combined_wcookie status=200 file=success.do | top productId

2. Use a command to find worst selling procductIDs

index=test sourcetype=access_combined_wcookie status=200 file=success.do | rare productId

3. Notice that ten rows were returned. You were asked to only return the top 5.

index=test sourcetype=access_combined_wcookie status=200 file=success.do | top limit=5 productId

4. Use the showperc option of top to remove percent from the display

   index=test sourcetype=access_combined_wcookie status=200 file=success.do | top productId limit=5 showperc=false

5. What is the productId of the best-selling product?

   WC-SHG04

6. Enter a search that returns all web application events where a file was successfully served to the user.

(index=test sourcetype=access_combined_wcookie status=200)

7. Use the rare command to find the files that show up the least amount of times in our events.

   index=test sourcetype=access_combined_wcookie status=200 | rare file

8. Enter a search that returns all web application events where a file was successfully served to the user. Use the rare command to find the products that show up the least amount of times in our events.

   index=test sourcetype=access_combined_wcookie status=200 | rare productId

9. Do you see anything that might be of a concern for the security team? Make the report even more granular using the by clause to split the rare events by the month in which they happened (date_month)

   index=test sourcetype=access_combined_wcookie status=200 | rare file by date_month

10. Now sort the data.

index=test sourcetype=access_combined_wcookie status=200 | rare file by date_month | sort –count

11. Enter a search that returns all web application events where an item was successfully added to a cart, or purchased. Remember, when an item is added to the cart the cart.do file is served; the success.do is served when the item is purchased.

index=test sourcetype=access_combined_wcookie file=success.do OR file=cart.do status=200

12. Use the stats count function with a by clause to count events by the file that was served.

(index=test sourcetype=access_combined_wcookie file=success.do OR file=cart.do status=200 | stats count by file)

13. Enter a search that returns all web application events where the connection failed because of client side error

index=test sourcetype=access_combined_wcookie (status>=400 AND status<500)

14. Notice that the count column is labeled count by default. Use an as clause to rename the column to Transactions

index=test sourcetype=access_combined_wcookie file=success.do OR file=cart.do status=200 | stats count as Transactions by file

15. Using the rename command, change the name of the file field to Function.

(index=test sourcetype=access_combined_wcookie file=success.do OR file=cart.do status=200 | stats count as Transactions by file | rename file as Function)

Day -4

16. Use search terms with the stats dc function to count all sessions (JSESSIONID)that have been used in our web application data. ----- returns the total values of the field

index=test sourcetype=access_combined_wcookie | stats dc(JSESSIONID)

17. Write a query to return the total values of Products

index=test sourcetype=access_combined_wcookie | stats dc(productsId)

Difference between dedup and distinct count: Counts of values vs Unique Values of the field

18. Use the as clause to rename the sessions as Logins.

Scenario: While being able to see the number of logins, the manager would like a little more information on who is logging in and asked you to report log-ins by IP.

19. Using the by clause, split the Logins by clientip.

(index=test sourcetype=access_combined_wcookie | stats dc(JSESSIONID) as Logins by clientip)

20. Use the sort command to sort Logins so that the clientip with the most Logins is displayed at the top of the list.

(index=test sourcetype=access_combined_wcookie | stats dc(JSESSIONID) as Logins by clientip | sort -Logins)

21. Craft search terms that return all events where a file was successfully served to a user.

(index=test sourcetype=access_combined_wcookie status=200)

22. Create a field named TotalBytes by using the sum function of the stats command.

(index=test sourcetype=access_combined_wcookie status=200 | stats sum(bytes) as TotalBytes)

23. Create a field named AverageBytes by using the avg function of the stats command

    index=test sourcetype=access_combined_wcookie status=200 | stats avg(bytes) as AverageBytes

24. Create a field named MaxBytes by using the max function of the stats command Split the results by the file field

index=test sourcetype=access_combined_wcookie | stats max(bytes) as MaxBytes by file

25. index=test sourcetype=access_combined_wcookie status=200 | stats sum(bytes) as TotalBytes | eval TotalMB=(TotalBytes/1024/1024) --- Conversion of Bytes to Megabyte

26. Create a field named MinBytes by using the min function of the stats command Split the results by the clientip field

index=test sourcetype=access_combined_wcookie | stats min(bytes) as MinBytes by clientip

27. Write a query to find the max value of all the bytes, convert it to kilobyte and sort by productid

index=test sourcetype=access_combined_wcookie status=200 | stats max(bytes) as MaxBytes by productId | eval KiloByte=(MaxBytes/1024) | sort – KiloByte

Write a query to find the min value of all the bytes, convert it to Megabyte and sort by file

28. Use the sort command to sort the file names into alphabetical order.

(index=test sourcetype=access_combined_wcookie status=200 | stats sum(bytes) as TotalBytes by file | sort file)

29. Search all database events and use the average (avg) function of the stats command to get an average duration of all queries.

(index=test sourcetype=db_audit | stats avg(Duration))

30. Use as and by clauses to rename the average field to time to complete and split by the Command.

(index=test sourcetype=db_audit | stats avg(Duration) as "time to complete" by Command

31. Sort the time to complete so that Command values that take the longest are shown first.

(index=test sourcetype=db_audit | stats avg(Duration) as "time to complete" by Command | sort - "time to complete")

Notice anything about the Command values that are taking the longest to complete?

32. Scenario: The Web Development team has asked for a list of the browsers that are being used to access the web application.

Task 7: Use the lists and values functions of the stats command to run a report of the browsers users are using to access the web application from. Use the stats list function to generate a list of all useragent values that have accessed the web application.

This report would be much more useful if we knew the number of times each useragent was used. Add a count function to the stats command that counts the events by useragent as Times used.

(index=test sourcetype=access_combined_wcookie | stats values(useragent) as "Agents used" count as "Times used" by useragent)

33. Put the results of Agents used and Times used into a table.

index=test sourcetype=access_combined_wcookie | stats values(useragent) as "Agents used" count as "Times used" by useragent | table "Agents used", "Times used"

34. Find the amount of data used in Megabytes by each clientip

index=test sourcetype=access_combined_wcookie status=200 | stats sum(bytes) as TotalBytes by clientip | eval TotalMB=(TotalBytes/1024/1024)

35. Enter a search that returns all web application events with a forbidden status greater than 400 and less than 599

index=test sourcetype="access_combined_wcookie" status>400 AND status<599 | stats count by status

DAY-5

1. Enter a search that returns all web application events with a forbidden status (403).

(index=test sourcetype=access_combined_wcookie status=403)

2. Use the stats count function to count the events by clientip and rename the count to attempts.

(index=test sourcetype=access_combined_wcookie status=403 | stats count as attempts by clientip)

3. Use the sort command to display the results so that the clientip with the highest attempts appears first.

index=test sourcetype=access_combined_wcookie status=403 OR status=404| stats count as attempts by clientip | sort –attempts

4. Enter a search that returns all web application events for all time where an item was successfully purchased. Remember, when an item is successfully purchased a success.do file is

served and a 200 status is returned. Use the stats count function with a by clause to count events by the productId.

(index=test sourcetype=access_combined_wcookie file=success.do status=200 | stats count by productId)

5. Remove the by clause from the search to return the total count of products sold

index=test sourcetype=access_combined_wcookie file=success.do status=200 | stats count

6. Write a query to search, which returns all web application events that include a purchase action or view and the product is in cart and then create a pie-chart as per the product ID obtained

index = test sourcetype="access_combined_wcookie" status=200 (action=purchase OR action=view) AND file=cart.do | stats count by productId

7. Find the list of clientips and the products bought who have accessed through google chrome and rename the count as events
   index = test sourcetype="access_combined_wcookie" referer_domain="http://www.google.com" |stats count by productId, clientip |rename count as events

AFTER LOOKUP

8. Enter a search that returns all web application events with a forbidden status (403) and display the corresponding productId

index=test sourcetype=access_combined_wcookie status=403 | stats count by productId

9. Find the list of clientips who have visited the URL "Buttercupgames.com" and sort the data by clientip and productnames

index=test sourcetype=access_combined_wcookie action=purchase referer_domain="http://www.buttercupgames.com" | stats count by clientip, ProductName

10. Enter a search that returns all web application events where an item was successfully purchased. Remember that if the success.do file is successfully returned to a user, a purchase was made and create a table with unique productid, Price and corresponding productname

index=test sourcetype=access_combined_wcookie file=success.do action=purchase | table productId,Price,ProductName