

## Splunk 基本 1 ラボ実習

ラボ表記規則:

[sourcetype=db\_audit] または [cs\_mime\_type] はソースタイプまたはフィールド名を指します。

**備考**    ラボ作業が個人のコンピュータまたはバーチャルマシンで実施された場合、ラボ環境は提供されません。 運用環境でのラボ作業は決して実施しないでください。

ラボマニュアルは示されるデータタイプ別にソースタイプを参照しています:

タイプ	ソースタイプ	関連のフィールド
ウェブアプリケーション	access_combined_wcookie	action、bytes、categoryId、clientip、itemId、JSESSIONID、productId、referer、referer_domain、status、useragent、file
データベース	db_audit	Command、Duration、Type
Web サーバー	linux_secure	COMMAND、PWD、pid、process

## ラボモジュール 9 - 変換コマンド

**備考:**    このラボ文書には2つのセクションがあります。  
最初のセクションには解答の記載がない指示が含まれます。  
次のセクションには予想されるサーチ文字列 (解答) が赤で記載された指示が含まれます。

### 説明

このラボでは、いくつかの Splunk 変換コマンドを使用してデータから統計を引き出します。

### 手順

シナリオ: セールスグループはプロダクト ID 別に最もよく売れているプロダクトのレポートを望んでいます。

**タスク 1:** top コマンドを使用して、最もよく売れているプロダクトのリストを入手します。

1. サーチビューに移動します。(ホーム App にいる場合は、画面左側にある列からサーチ & レポートをクリックしてください。サーチビューへは、画面一番上の緑のバーにあるサーチメニューをクリックしてもアクセスすることができます。)

**備考**    このコースでは、常時メインインデックスを使用してサーチすることになります。  
:        これは運用環境で最適な実例ではありませんが、データセット制限の性質により、これらのラボに必要となります。

2. アイテムが正常に購入されたすべてのウェブアプリケーションイベントを返す検索を入力します。  
「success.do」ファイルがユーザーに正常に返される場合、購入が行われたことになります。

結果例:

i	Time	Event
>	5/21/18 11:57:14.000 PM	109.169.32.135 - - [21/May/2018:23:57:14] "POST /cart/success.do?JSESSIONID=SD1SL7FF6ADFF89341&productId=FI-AG-G08 HTTP 1.1" 200 3767 "http://www.buttercupgames.com/cart.do?action=purchase&" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)" 986 host = web_application   source = access_30DAY.log   sourcetype = access_combined_wcookie
>	5/21/18 11:57:13.000 PM	109.169.32.135 - - [21/May/2018:23:57:13] "POST /success.do?action=purchase&categoryId=SHOOTER&productId=WC-SH-G04&JSESSIONID=SD1SL7FF6ADFF89341 HTTP 1.1" 200 268 "http://www.buttercupgames.com/cart.do?action=addtocart&categoryId=SHOOTER&productId=WC-SH-G04" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)" 448 host = web_application   source = access_30DAY.log   sourcetype = access_combined_wcookie
>	5/21/18 11:53:43.000 PM	198.35.3.23 - - [21/May/2018:23:53:43] "POST /success.do?action=purchase&categoryId=ARCADE&productId=MB-AG-G07&JSESSIONID=SD8SL8FF6ADFF4957 HTTP 1.1" 200 2915 "http://www.buttercupgames.com/cart.do?action=addtocart&categoryId=ARCADE&productId=MB-AG-G07" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)" 448 host = web_application   source = access_30DAY.log   sourcetype = access_combined_wcookie

3. top コマンドを使用して最もよく売れているプロダクト ID を常時で検索します。

結果例:

productId	count	percent
WC-SH-G04	1360	8.426792
DB-SG-G01	1319	8.172749
DC-SG-G02	1308	8.104591
MB-AG-T01	1205	7.466386
MB-AG-G07	1204	7.460190
WC-SH-A02	1192	7.385836
FS-SG-G03	1155	7.156577
WC-SH-A01	1100	6.815788
WC-SH-T02	1076	6.667080
PZ-SG-G05	1012	6.270525

4. 10 行が返されています。 トップ 5 のみを返すように依頼されていました。
5. 「limit」引数を使用してリクエストされた行数のみを返すようにします。

結果例:

productId	count	percent
WC-SH-G04	1360	8.426792
DB-SG-G01	1319	8.172749
DC-SG-G02	1308	8.104591
MB-AG-T01	1205	7.466386
MB-AG-G07	1204	7.460190

6. top の showperc オプションを使用して、表示から percent を削除します。

結果例:

productId	count
WC-SH-G04	1360
DB-SG-G01	1319
DC-SG-G02	1308
MB-AG-T01	1205
MB-AG-G07	1204

7. 最もよく売れているプロダクトのプロダクト ID は何ですか？ クイズにでる可能性がありますので覚えておいてください。

シナリオ: まれなイベントが最も貴重な情報を提供してくれる場合があります。例えばほとんどアクセスされることのないファイルで、会社内部の誰かが公開したファイルや、わずかな人しか知らないバックドアなどが考えられます。セキュリティチームはあなたにオンラインで入手できるがほとんどアクセスがないファイルを検索するように依頼します。

**タスク 2: rare コマンド**を使用してウェブアプリケーションで最もアクセス数が少ないファイルを確認します。

8. ファイルが正常に使用されたすべてのウェブアプリケーションイベントを返す検索を入力します。  
9. rare コマンドを使用して、イベントで最も表示回数が少ないファイルを検索します。

結果例:

file	count	percent
api	1	0.000858
account	2	0.001716
userlist	10	0.008582
passwords.pdf	235	0.201670
error.do	1795	1.540416
success.do	16139	13.850009
oldlink	19642	16.856179
category.screen	19958	17.127361
cart.do	29328	25.168416
product.screen	29417	25.244793

10. セキュリティチームの懸念となるような何かが確認できますか？ 「by」を使用して生じた月別 (date\_month) に「rare」イベントを分割し、より粒度の細かいレポートにします。

シナリオ: セールスチームはカートに追加されたアイテム数と購入されたアイテム数の比較を知りたいと思っています。

**タスク 3: stats コマンド**の count 関数を使用して、カートに追加されたアイテム数と購入数を割り出します。

11. アイテムが正常にカートに追加された、または購入されたすべてのウェブアプリケーションイベントを返す検索を入力します。アイテムがカートに追加されると「cart.do」ファイルが使用され、アイテムが購入されると「success.do」が使用されることを覚えておいてください。

12. stats count 関数を by 句で使用し、使用されたファイルごとにイベントをカウントします。

結果例:

file	count
cart.do	29328
success.do	16139

13. カウント列はデフォルトで「count」と名づけられています。「as」を使用し、その列の名前を「Transactions」に変更します。

結果例:

file	Transactions
cart.do	29328
success.do	16139

14. 「rename」コマンドを使用し、「file」フィールドの名前を「Function」に変更します。

結果例:

Function	Transactions
cart.do	29328
success.do	16139

シナリオ: マーケティングマネージャーはユーザーがウェブアプリケーションにログインした回数を知りたいと思っています。

- タスク 4: distinct count stats** 関数を使用して、システムでユーザー用に作成されたセッションの回数をカウントします。

15. stats dc 関数でサーチ単語を使用し、ウェブアプリケーションデータで使用されたすべてのセッション (JSESSIONID) をカウントします。

結果例:

dc(JSESSIONID)
11455

16. as 句を使用し、そのセッションの名前を「Logins」に変更します。

シナリオ: ログイン数は確認できますが、ログインしている人物に関する、もう少し詳しい情報をマネージャーは求めており、あなたに IP 別ログインのレポートを依頼します。

17. by 句を使用し、その「Logins」を「clientip」別に分割します。

結果例:

clientip	Logins
107.3.146.207	186
108.65.113.83	83
109.169.32.135	172
110.138.30.229	82
110.159.208.78	95

18. 「sort」コマンドを使用して「Logins」をソートし、一番多い「Logins」の「clientip」がリストの一番上に表示されるようにします。トップの「clientip」についてはクイズで質問される可能性があるするので覚えておいてください。

シナリオ: IT チームは当年のインフラ予算を処理しています。最も多く帯域幅コストが費やされているのはどこかを把握しようとしています。

**タスク 5: stats sum 関数を使用して、ウェブアプリケーションに使用された総バイトを検索します。**

19. ファイルが正常にユーザーに使用されたすべてのイベントを返す検索単語を作ります。
20. stats コマンドの sum 関数を使用し、「TotalBytes」という名前のフィールドを作成します。

結果例:

TotalBytes ◆

255334688

21. by 句を使用して file フィールド別に結果を分割します。

結果例:

file ◆	TotalBytes ◆
account	4238
api	1456
cart.do	61311724
category.screen	42250130
error.do	3747647
oldlink	41349801
passwords.pdf	11103985
product.screen	61672339
success.do	33862909
userlist	27690

22. sort コマンドを使用してアルファベット順にファイル名をソートします。
23. 帯域幅の使用量が最も少ないファイル名は何ですか？ クイズで質問される可能性があります。

シナリオ: 非常事態！ ウェブサイトが遅くなり、ユーザーから苦情が寄せられています。アプリケーション開発者によれば、更新を最近実施しており、トラブルを引き起こすデータベースクエリの存在が懸念されるということです。

**タスク 6: stats コマンドの average 関数を使用し、各データベースクエリの平均実行時間を検索します。**

24. すべてのデータベースイベントを検索し、stats コマンドの平均 (avg) 関数を使用して、全クエリの平均「Duration」を得ます。

結果例:

avg(Duration) ◆

239.4764303178484

25. 「as」および「by」を使用して平均フィールドの名前を「time to complete」に変更し、「Command」別に分割します。

## 結果例:

Command ↕	time to complete ↕
SELECT * FROM users INNER JOIN creditcards ON users.userid=creditcards.userid INNER JOIN contactinfo ON users.userid=contactinfo.userid INNER JOIN usertracking ON users.userid=usertracking.userid WHERE users.userid = 2208	9988
SELECT * FROM users INNER JOIN creditcards ON users.userid=creditcards.userid INNER JOIN contactinfo ON users.userid=contactinfo.userid INNER JOIN usertracking ON users.userid=usertracking.userid WHERE users.userid = 1021	9985
SELECT * FROM users INNER JOIN creditcards ON users.userid=creditcards.userid INNER JOIN contactinfo ON users.userid=contactinfo.userid INNER JOIN usertracking ON users.userid=usertracking.userid WHERE users.userid = 4011	9985
SELECT * FROM users INNER JOIN creditcards ON users.userid=creditcards.userid INNER JOIN contactinfo ON users.userid=contactinfo.userid INNER JOIN usertracking ON users.userid=usertracking.userid WHERE users.userid = 6186	9980
SELECT * FROM users INNER JOIN creditcards ON users.userid=creditcards.userid INNER JOIN contactinfo ON users.userid=contactinfo.userid INNER JOIN usertracking ON users.userid=usertracking.userid WHERE users.userid = 2429	9971

26. time to complete をソートし、Command 値の一番長いものが最初に表示されるようにします。

27. 完了までに最も長い時間がかかっている「コマンド」値について何か気付くことがありますか？

シナリオ: ウェブ開発チームはウェブアプリケーションへのアクセスに使用されているブラウザのリストを依頼しました。

タスク 7: stats コマンドのリストおよび値関数を使用して、ブラウザユーザーがウェブアプリケーションへのアクセスに使用しているブラウザのレポートを実行します。

28. stats list 関数を使用してウェブアプリケーションにアクセスしたすべての useragent 値のリストを作成します。

## 結果例:

```
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.2.28) Gecko/20120306 YFF3 Firefox/3.6.28 (.NET CLR 3.5.30729; .NET4.0C)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)
```

29. ほとんどの「useragent」値は結果に複数回表示されます。

30. 「stats values」関数を使用して各「useragent」の1つのインスタンスのみを返すようにします。「as」を追加して結果の名前を「Agents used」に変更します。

## 結果例:

```
Agents used ↕
Googlebot/2.1 ( http://www.googlebot.com/bot.html)
Googlebot/2.1 (http://www.googlebot.com/bot.html)
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)
Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; .NET4.0C; .NET4.0E; MS-RTC LM 8; InfoPath.1)
Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; InfoPath.1; .NET4.0C; .NET4.0E; MS-RTC LM 8)
Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; MS-RTC LM 8; InfoPath.2)
Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.2; .NET CLR 1.1.4322; InfoPath.1; MS-RTC LM 8)
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8) AppleWebKit/534.55.3 (KHTML, like Gecko) Version/5.1.5 Safari/534.55.3
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5
```

31. このレポートは各「useragent」の使用回数を知ることができればはるかに有益なものとなります。  
「stats」コマンドに「count」関数を追加して「Times used」として「useragent」別にイベントをカウントします。

結果例:

useragent ⇅	Agents used ↕	Times used ⇅
Opera/9.20 (Windows NT 6.0; U; en)	Opera/9.20 (Windows NT 6.0; U; en)	1557
Opera/9.01 (Windows NT 5.1; U; en)	Opera/9.01 (Windows NT 5.1; U; en)	1890
Mozilla/5.0 (iPad; U; CPU OS 4_3_5 like Mac OS X; en-us) AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8L1 Safari/6533.18.5	Mozilla/5.0 (iPad; U; CPU OS 4_3_5 like Mac OS X; en-us) AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8L1 Safari/6533.18.5	5300
Mozilla/5.0 (iPad; CPU OS 5_1_1 like Mac OS X) AppleWebKit/534.46 (KHTML, like Gecko) Version/5.1 Mobile/9B206 Safari/7534.48.3	Mozilla/5.0 (iPad; CPU OS 5_1_1 like Mac OS X) AppleWebKit/534.46 (KHTML, like Gecko) Version/5.1 Mobile/9B206 Safari/7534.48.3	5013
Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots)	Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots)	1231
Mozilla/5.0 (compatible; NetcraftSurveyAgent/1.0/cc-prepass-https; +info@netcraft.com)	Mozilla/5.0 (compatible; NetcraftSurveyAgent/1.0/cc-prepass-https; +info@netcraft.com)	1483
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)	11754
Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)	1575

32. 「使用されたエージェント」と 「使用回数」の結果をテーブルに入れます。

結果例:

Agents used ⇅	Times used ⇅
Googlebot/2.1 ( http://www.googlebot.com/bot.html)	1277
Googlebot/2.1 (http://www.googlebot.com/bot.html)	1327
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)	2155
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)	2170
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)	1825

## Splunk 基本 1 ラボ実習

ラボ表記規則:

[sourcetype=db\_audit] または [cs\_mime\_type] はソースタイプまたはフィールド名を指します。

**備考** ラボ作業が個人のコンピュータまたはバーチャルマシンで実施された場合、ラボ環境は提供されません。運用環境でのラボ作業は決して実施しないでください。

ラボマニュアルは示されるデータタイプ別にソースタイプを参照しています:

タイプ	ソースタイプ	関連のフィールド
ウェブアプリケーション	access_combined_wcookie	action、bytes、categoryId、clientip、itemId、JSESSIONID、productId、referer、referer_domain、status、useragent、file
データベース	db_audit	Command、Duration、Type
Web サーバー	linux_secure	COMMAND、PWD、pid、process

## ラボモジュール 9 - 変換コマンド (ソリューション付)

**備考:** このラボ文書には2つのセクションがあります。  
最初のセクションには解答の記載がない指示が含まれます。  
次のセクションには予想されるサーチ文字列 (解答) が赤で記載された指示が含まれます。

### 説明

このラボでは、いくつかの Splunk 変換コマンドを使用してデータから統計を引き出します。

### 手順

シナリオ: セールスグループはプロダクト ID 別に最もよく売れているプロダクトのレポートを望んでいます。

**タスク 1:** top コマンドを使用して、最もよく売れているプロダクトのリストを入手します。

1. サーチビューに移動します。(ホーム App にいる場合は、画面左側にある列からサーチ & レポートをクリックしてください。サーチビューへは、画面一番上の緑のバーにあるサーチメニューをクリックしてもアクセスすることができます。)

**備考** このコースでは、常時メインインデックスを使用してサーチすることになります。  
これは運用環境で最適な実例ではありませんが、データセット制限の性質により、これらのラボに必要となります。



2. アイテムが正常に購入されたすべてのウェブアプリケーションイベントを返す検索を入力します。  
「success.do」ファイルがユーザーに正常に返される場合、購入が行われたことになります。

(index=main sourcetype=access\_combined\_wcookie status=200 file=success.do)

結果例:

i	Time	Event
>	5/21/18 11:57:14.000 PM	109.169.32.135 - - [21/May/2018:23:57:14] "POST /cart/success.do?JSESSIONID=SD1SL7FF6ADFF89341&productId=FI-AG-G08 HTTP 1.1" 200 3767 "http://www.buttercupgames.com/cart.do?action=purchase&" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)" 986 host = web_application   source = access_30DAY.log   sourcetype = access_combined_wcookie
>	5/21/18 11:57:13.000 PM	109.169.32.135 - - [21/May/2018:23:57:13] "POST /success.do?action=purchase&categoryId=SHOOTER&productId=WC-SH-G04&JSESSIONID=SD1SL7FF6ADFF89341 HTTP 1.1" 200 268 "http://www.buttercupgames.com/cart.do?action=addtocart&categoryId=SHOOTER&productId=WC-SH-G04" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)" 448 host = web_application   source = access_30DAY.log   sourcetype = access_combined_wcookie
>	5/21/18 11:53:43.000 PM	198.35.3.23 - - [21/May/2018:23:53:43] "POST /success.do?action=purchase&categoryId=ARCADE&productId=MB-AG-G07&JSESSIONID=SD8SL8FF6ADFF4957 HTTP 1.1" 200 2915 "http://www.buttercupgames.com/cart.do?action=addtocart&categoryId=ARCADE&productId=MB-AG-G07" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)" 448 host = web_application   source = access_30DAY.log   sourcetype = access_combined_wcookie

3. top コマンドを使用して最もよく売れているプロダクト ID を常時で検索します。

(index=main sourcetype=access\_combined\_wcookie status=200 file=success.do | top productId)

結果例:

productId	count	percent
WC-SH-G04	1360	8.426792
DB-SG-G01	1319	8.172749
DC-SG-G02	1308	8.104591
MB-AG-T01	1205	7.466386
MB-AG-G07	1204	7.460190
WC-SH-A02	1192	7.385836
FS-SG-G03	1155	7.156577
WC-SH-A01	1100	6.815788
WC-SH-T02	1076	6.667080
PZ-SG-G05	1012	6.270525

4. 10 行が返されています。 トップ 5 のみを返すように依頼されていました。

5. 「limit」引数を使用してリクエストされた行数のみを返すようにします。

(index=main sourcetype=access\_combined\_wcookie status=200 file=success.do | top productId limit=5)

結果例:

productId	count	percent
WC-SH-G04	1360	8.426792
DB-SG-G01	1319	8.172749
DC-SG-G02	1308	8.104591
MB-AG-T01	1205	7.466386
MB-AG-G07	1204	7.460190

6. top の showperc オプションを使用して、表示から percent を削除します。

(index=main sourcetype=access\_combined\_wcookie status=200 file=success.do | top productId limit=5 showperc=false)

結果例:

productId	count
WC-SH-G04	1360
DB-SG-G01	1319
DC-SG-G02	1308
MB-AG-T01	1205
MB-AG-G07	1204

7. 最もよく売れているプロダクトのプロダクト ID は何ですか？ クイズにでる可能性がありますので覚えておいてください。 (WC-SH-G04)

シナリオ: まれなイベントが最も貴重な情報を提供してくれる場合があります。例えばほとんどアクセスされることのないファイルで、会社内部の誰かが公開したファイルや、わずかな人しか知らないバックドアなどが考えられます。セキュリティチームはあなたにオンラインで入手できるがほとんどアクセスがないファイルを検索するように依頼します。

**タスク 2: rare コマンドを使用してウェブアプリケーションで最もアクセス数が少ないファイルを確認します。**

8. ファイルが正常に使用されたすべてのウェブアプリケーションイベントを返すサーチを入力します。  
(index=main sourcetype=access\_combined\_wcookie status=200)
9. rare コマンドを使用して、イベントで最も表示回数が少ないファイルを検索します。  
(index=main sourcetype=access\_combined\_wcookie status=200 | rare file)

結果例:

file	count	percent
api	1	0.000858
account	2	0.001716
userlist	10	0.008582
passwords.pdf	235	0.201670
error.do	1795	1.540416
success.do	16139	13.850009
oldlink	19642	16.856179
category.screen	19958	17.127361
cart.do	29328	25.168416
product.screen	29417	25.244793

10. セキュリティチームの懸念となるような何かが確認できますか？ 「by」を使用して生じた月別 (date\_month) に「rare」イベントを分割し、より粒度の細かいレポートにします。  
(index=main sourcetype=access\_combined\_wcookie status=200 | rare file by date\_month)

シナリオ: セールスチームはカートに追加されたアイテム数と購入されたアイテム数の比較を知りたいと思っています。

**タスク 3: stats コマンドの count 関数を使用して、カートに追加されたアイテム数と購入数を割り出します。**

11. アイテムが正常にカートに追加された、または購入されたすべてのウェブアプリケーションイベントを返すサーチを入力します。アイテムがカートに追加されると「cart.do」ファイルが使用され、アイテムが購入されると「success.do」が使用されることを覚えておいてください。  
(index=main sourcetype=access\_combined\_wcookie file=success.do OR file=cart.do status=200)

12. `stats count` 関数を `by` 句で使用し、使用されたファイルごとにイベントをカウントします。

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

結果例:

file	count
cart.do	29328
success.do	16139

13. カウント列はデフォルトで「count」と名づけられています。「as」を使用し、その列の名前を「Transactions」に変更します。

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

結果例:

file	Transactions
cart.do	29328
success.do	16139

14. 「rename」コマンドを使用し、「file」フィールドの名前を「Function」に変更します。

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

結果例:

Function	Transactions
cart.do	29328
success.do	16139

シナリオ: マーケティングマネージャーはユーザーがウェブアプリケーションにログインした回数を知りたいと思っています。

**タスク 4: distinct count stats** 関数を使用して、システムでユーザー用に作成されたセッションの回数をカウントします。

15. `stats dc` 関数で検索単語を使用し、ウェブアプリケーションデータで使用されたすべてのセッション (JSESSIONID) をカウントします。

(`index=main sourcetype=access_combined_wcookie | stats dc(JSESSIONID)`)

結果例:

dc(JSESSIONID)
11455

16. `as` 句を使用し、そのセッションの名前を「Logins」に変更します。

シナリオ: ログイン数は確認できますが、ログインしている人物に関する、もう少し詳しい情報をマネージャーは求めており、あなたに IP 別ログインのレポートを依頼します。

17. `by` 句を使用し、その「Logins」を「clientip」別に分割します。

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

結果例:

clientip	Logins
107.3.146.207	186
108.65.113.83	83
109.169.32.135	172
110.138.30.229	82
110.159.208.78	95

18. 「sort」コマンドを使用して「Logins」をソートし、一番多い「Logins」の「clientip」がリストの一番上に表示されるようにします。トップの「clientip」についてはクイズで質問される可能性があるなので覚えておいてください。

```
(index=main sourcetype=access_combined_wcookie | stats dc(JSESSIONID) as Logins by clientip | sort -Logins)
(87.194.216.51)
```

シナリオ: IT チームは当年のインフラ予算を処理しています。最も多く帯域幅コストが費やされているのはどこかを把握しようとしています。

**タスク 5: stats sum 関数を使用して、ウェブアプリケーションに使用された総バイトを検索します。**

19. ファイルが正常にユーザーに使用されたすべてのイベントを返す検索単語を作ります。

```
(index=main sourcetype=access_combined_wcookie status=200)
```

20. stats コマンドの sum 関数を使用し、「TotalBytes」という名前のフィールドを作成します。

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

結果例:

TotalBytes
255334688

21. by 句を使用して file フィールド別に結果を分割します。

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

結果例:

file	TotalBytes
account	4238
api	1456
cart.do	61311724
category.screen	42250130
error.do	3747647
oldlink	41349801
passwords.pdf	11103985
product.screen	61672339
success.do	33862909
userlist	27690

22. sort コマンドを使用してアルファベット順にファイル名をソートします。

(index=main sourcetype=access\_combined\_wcookie status=200 | stats sum(bytes) as TotalBytes by file | sort file)

23. 帯域幅の使用量が最も少ないファイル名は何ですか？ クイズで質問される可能性があります。(api)

シナリオ: 非常事態！ ウェブサイトが遅くなり、ユーザーから苦情が寄せられています。アプリケーション開発者によれば、更新を最近実施しており、トラブルを引き起こすデータベースクエリの存在が懸念されるとのことです。

**タスク 6: stats コマンドの average 関数を使用し、各データベースクエリの平均実行時間を検索します。**

24. すべてのデータベースイベントをサーチし、stats コマンドの平均 (avg) 関数を使用して、全クエリの平均「Duration」を得ます。

(index=main sourcetype=db\_audit | stats avg(Duration))

結果例:

avg(Duration) ◆

239.4764303178484

25. 「as」および「by」を使用して平均フィールドの名前を「time to complete」に変更し、「Command」別に分割します。

(index=main sourcetype=db\_audit | stats avg(Duration) as "time to complete" by Command)

結果例:

Command ◆	time to complete ◆
SELECT * FROM users INNER JOIN creditcards ON users.userid=creditcards.userid INNER JOIN contactinfo ON users.userid=contactinfo.userid INNER JOIN usertracking ON users.userid=usertracking.userid WHERE users.userid = 2208	9988
SELECT * FROM users INNER JOIN creditcards ON users.userid=creditcards.userid INNER JOIN contactinfo ON users.userid=contactinfo.userid INNER JOIN usertracking ON users.userid=usertracking.userid WHERE users.userid = 1021	9985
SELECT * FROM users INNER JOIN creditcards ON users.userid=creditcards.userid INNER JOIN contactinfo ON users.userid=contactinfo.userid INNER JOIN usertracking ON users.userid=usertracking.userid WHERE users.userid = 4011	9985
SELECT * FROM users INNER JOIN creditcards ON users.userid=creditcards.userid INNER JOIN contactinfo ON users.userid=contactinfo.userid INNER JOIN usertracking ON users.userid=usertracking.userid WHERE users.userid = 6186	9980
SELECT * FROM users INNER JOIN creditcards ON users.userid=creditcards.userid INNER JOIN contactinfo ON users.userid=contactinfo.userid INNER JOIN usertracking ON users.userid=usertracking.userid WHERE users.userid = 2429	9971

26. time to complete をソートし、Command 値の一番長いものが最初に表示されるようにします。

(index=main sourcetype=db\_audit | stats avg(Duration) as "time to complete" by Command | sort - "time to complete")

27. 完了までに最も長い時間がかかっている「コマンド」値について何か気付くことがありますか？

シナリオ: ウェブ開発チームはウェブアプリケーションへのアクセスに使用されているブラウザのリストを依頼しました。

**タスク 7: stats コマンドのリストおよび値関数を使用して、ブラウザユーザーがウェブアプリケーションへのアクセスに使用しているブラウザのレポートを実行します。**

28. stats list 関数を使用してウェブアプリケーションにアクセスしたすべての useragent 値のリストを作成します。

```
(index=main sourcetype=access_combined_wcookie | stats list(useragent))
```

結果例:

```
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.2.28) Gecko/20120306 YFF3 Firefox/3.6.28 (.NET CLR 3.5.30729; .NET4.0C)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)
```

29. ほとんどの「useragent」値は結果に複数回表示されます。

30. 「stats values」関数を使用して各「useragent」の1つのインスタンスのみを返すようにします。「as」を追加して結果の名前を「Agents used」に変更します。

```
(index=main sourcetype=access_combined_wcookie | stats values(useragent) as "Agents used")
```

結果例:

```
Agents used
Googlebot/2.1 ( http://www.googlebot.com/bot.html)
Googlebot/2.1 (http://www.googlebot.com/bot.html)
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)
Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; .NET4.0C; .NET4.0E; MS-RTC LM 8; InfoPath.1)
Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; InfoPath.1; .NET4.0C; .NET4.0E; MS-RTC LM 8)
Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; MS-RTC LM 8; InfoPath.2)
Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.2; .NET CLR 1.1.4322; InfoPath.1; MS-RTC LM 8)
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8) AppleWebKit/534.55.3 (KHTML, like Gecko) Version/5.1.5 Safari/534.55.3
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.46 Safari/536.5
```

31. このレポートは各「useragent」の使用回数を知ることができればはるかに有益なものとなります。

「stats」コマンドに「count」関数を追加して「Times used」として「useragent」別にイベントをカウントします。

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

結果例:

useragent	Agents used	Times used
Opera/9.20 (Windows NT 6.0; U; en)	Opera/9.20 (Windows NT 6.0; U; en)	1557
Opera/9.01 (Windows NT 5.1; U; en)	Opera/9.01 (Windows NT 5.1; U; en)	1890
Mozilla/5.0 (iPad; U; CPU OS 4_3_5 like Mac OS X; en-us) AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8L1 Safari/6533.18.5	Mozilla/5.0 (iPad; U; CPU OS 4_3_5 like Mac OS X; en-us) AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8L1 Safari/6533.18.5	5300
Mozilla/5.0 (iPad; CPU OS 5_1_1 like Mac OS X) AppleWebKit/534.46 (KHTML, like Gecko) Version/5.1 Mobile/9B206 Safari/7534.48.3	Mozilla/5.0 (iPad; CPU OS 5_1_1 like Mac OS X) AppleWebKit/534.46 (KHTML, like Gecko) Version/5.1 Mobile/9B206 Safari/7534.48.3	5013
Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots)	Mozilla/5.0 (compatible; YandexBot/3.0; +http://yandex.com/bots)	1231
Mozilla/5.0 (compatible; NetcraftSurveyAgent/1.0/cc-prepass-https; +info@netcraft.com)	Mozilla/5.0 (compatible; NetcraftSurveyAgent/1.0/cc-prepass-https; +info@netcraft.com)	1483
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)	Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0; BOIE9;ENUS)	11754
Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)	1575

32. 「使用されたエージェント」と「使用回数」の結果をテーブルに入れます。

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

結果例:

Agents used ↕	Times used ↕
Googlebot/2.1 ( http://www.googlebot.com/bot.html)	1277
Googlebot/2.1 (http://www.googlebot.com/bot.html)	1327
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)	2155
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)	2170
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)	1825