splunk>

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 8 - 基本コマンド

備考: Splunk におけるサーチの基本を理解したところで、もう少し難しいラボにチャレンジしてみましょう。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、フィールド、テーブル、リネーム、重複排除を含む一般的な Splunk コマンドをいくつか使用します。

手順

シナリオ: マーケティングチームはマーケティングキャンペーンに関連するすべてのユーザーセッションを追跡しています。 購入アクションを含むすべてのユーザーセッションのレポートのようなもので、それにより、実行中の別のキャンペーンで評価できるようにします。

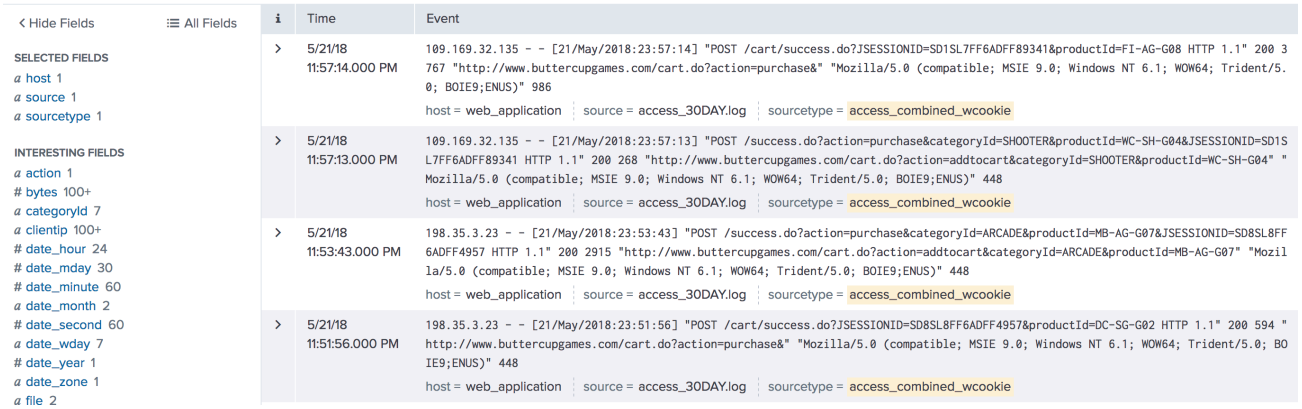
タスク 1: リクエストされたデータをサーチします。

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. ウェブステータスが 200 の購入アクションを含む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。

結果例:



3. **関連のフィールド**リストで `file` フィールドを選択します。

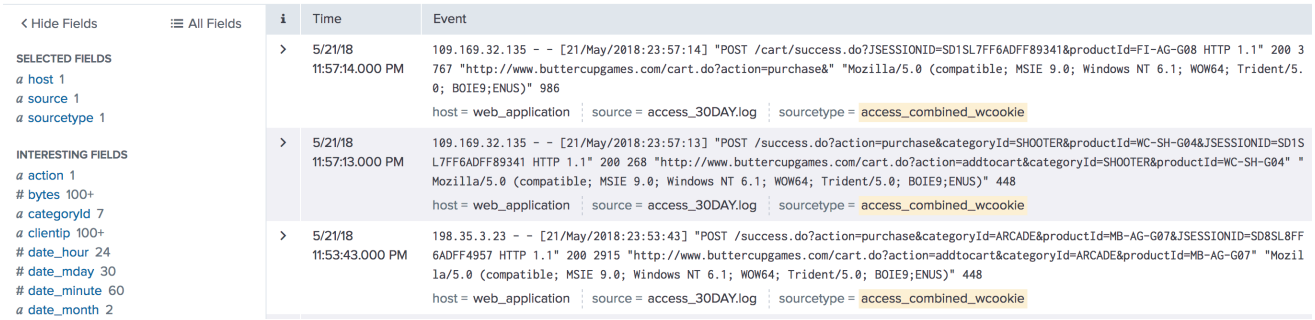
結果例:

Values	Count	%
success.do	16,139	89.991%
error.do	1,795	10.009%

4. Web サーバーから返されるファイルは 2 種類あります。 その 2 つとは、`error.do` と `success.do` です。ウェブ開発チームによると、`success.do` は注文処理時に、`error.do` は処理中の情報にエラーがある場合に使用されます。

5. チームは正常処理された購入のみを検索していますので、サーチを変更してそれらのみ返すようにします。

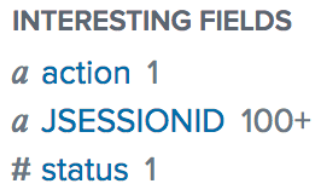
結果例:



6. チームに関係のないフィールドを確認します。 `fields` コマンドを使用し、`action`、`JSESSIONID`、`status` フィールドのみを返すようにします。サーチはコマンドを使用してより速く実行されますか？

7.

結果例:



7. フィールドリストは整理されているように見えますが、こういったイベントが見えるとチームが混乱する場合があります。

タスク 2: テーブルが読みやすいようにデータを配置します。

8. `fields` コマンドを `table` コマンドに置き換えて、データをテーブルとして表示します。

結果例:

20 Per Page ▾ | Format | Preview ▾ | < Prev | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | Next >

action	JSESSIONID	status
purchase	SD6SL5FF6ADFF89354	200
purchase	SD6SL5FF6ADFF89354	200
purchase	SD6SL5FF6ADFF89354	200
purchase	SD6SL5FF6ADFF89354	200
purchase	SD6SL5FF6ADFF89354	200

9. フィールドの順序を変更し、`JSESSIONID` が最初の列にくるようにします。

結果例:

20 Per Page ▾ | Format | Preview ▾ | < Prev | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... | Next >

JSESSIONID	action	status
SD6SL5FF6ADFF89354	purchase	200
SD6SL5FF6ADFF89354	purchase	200
SD6SL5FF6ADFF89354	purchase	200
SD6SL5FF6ADFF89354	purchase	200
SD6SL5FF6ADFF89354	purchase	200

10. セッション ID はマーケティングデータでは「User Sessions」とよばれます。 `JSESSIONID` の名前を変更し、レポートとマーケティングデータを一致させます。

結果例:

UserSessions	action	status
SD6SL5FF6ADFF89354	purchase	200
SD6SL5FF6ADFF89354	purchase	200
SD6SL5FF6ADFF89354	purchase	200
SD6SL5FF6ADFF89354	purchase	200

11. `sort` コマンドを使用して `UserSessions` をソートします。
12. `UserSessions` 値の中には複数回表示されるものもあります。 また、**統計**タブで返されるイベント数にも注意します。
13. `sort` コマンドを解除し、`dedup` を使用して同一のセッション値を削除します。

結果例:

UserSessions	action	status
SD1SL7FF6ADFF89341	purchase	200
SD8SL8FF6ADFF4957	purchase	200
SD2SL10FF6ADFF4955	purchase	200

14. いくつのイベントが**統計**タブに記載されていますか？

備考: ベストな成果を得るため、dedup はサーチのできるだけ早い段階で行ってください。

15. `action` および `status` フィールドの表示はデータのサニティーチェックに適していましたが、マーケティングチームはこれらを表示させる必要はありません。 これらをテーブル表示から削除します。

結果例:

UserSessions
SD1SL7FF6ADFF89341
SD8SL8FF6ADFF4957
SD2SL10FF6ADFF4955
SD3SL5FF3ADFF89564

splunk>

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 8 - 基本コマンド (ソリューション付)

備考: Splunk におけるサーチの基本を理解したところで、もう少し難しいラボにチャレンジしてみましょう。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、フィールド、テーブル、リネーム、重複排除を含む一般的な Splunk コマンドをいくつか使用します。

手順

シナリオ: マーケティングチームはマーケティングキャンペーンに関連するすべてのユーザーセッションを追跡しています。 購入アクションを含むすべてのユーザーセッションのレポートのようなもので、それにより、実行中の別のキャンペーンで評価できるようにします。

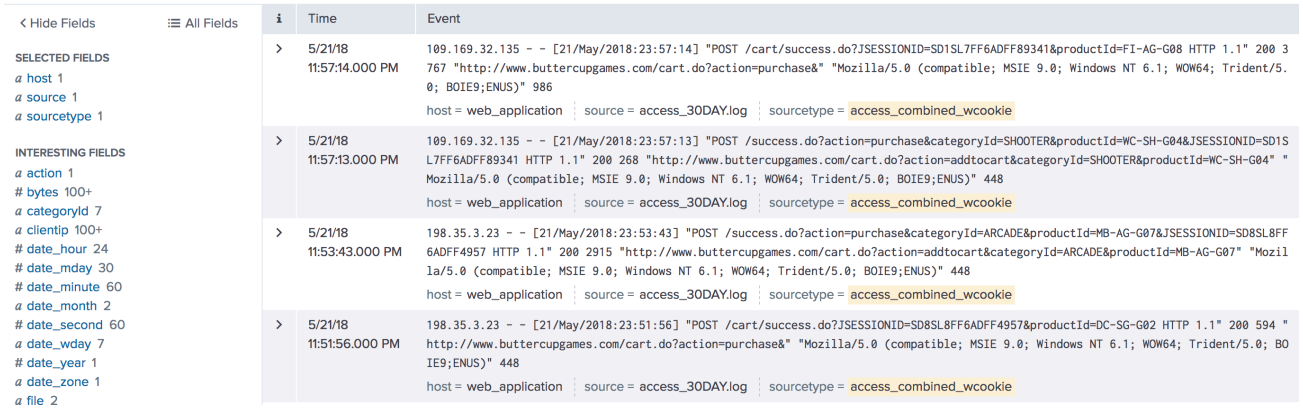
タスク 1: リクエストされたデータをサーチします。

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. ウェブステータスが 200 の購入アクションを含む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 action=purchase status=200)

結果例:



3. **関連のフィールド**リストで `file` フィールドを選択します。

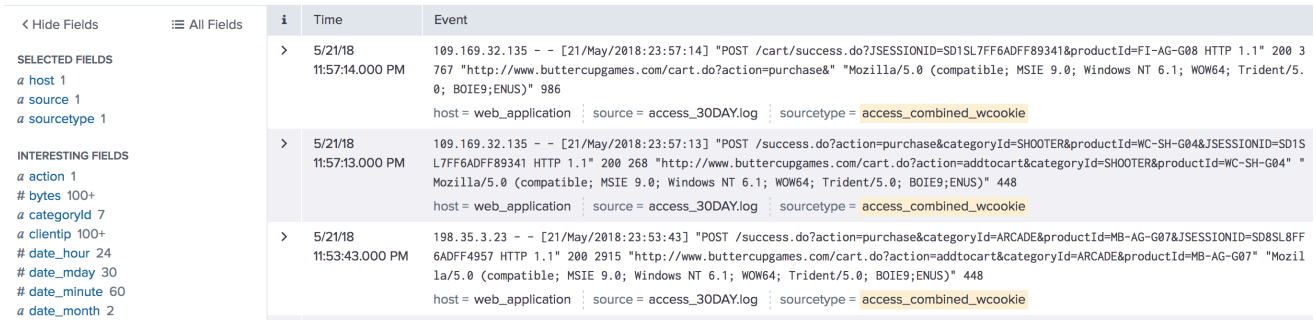
結果例:

Values	Count	%
success.do	16,139	89.991%
error.do	1,795	10.009%

4. Web サーバーから返されるファイルは 2 種類あります。 その 2 つとは、`error.do` と `success.do` です。ウェブ開発チームによると、`success.do` は注文処理時に、`error.do` は処理中の情報にエラーがある場合に使用されます。

5. チームは正常処理された購入のみを検索していますので、サーチを変更してそれらのみ返すようにします。(index=main sourcetype=access_combined_wcookie action=purchase status=200 file=success.do)

結果例:



6. チームに関係のないフィールドを確認します。 `fields` コマンドを使用し、`action`、`JSESSIONID`、`status` フィールドのみを返すようにします。サーチはコマンドを使用してより速く実行されますか？

(index=main sourcetype=access_combined_wcookie action=purchase status=200 file=success.do | fields action, JSESSIONID, status)

結果例:

7. フィールドリストは整理されているように見えますが、こういったイベントが見えるとチームが混乱する場合があります。

タスク 2: テーブルが読みやすいようにデータを配置します。

8. `fields` コマンドを `table` コマンドに置き換えて、データをテーブルとして表示します。(index=main sourcetype=access_combined_wcookie action=purchase status=200 file=success.do | table action, JSESSIONID, status)

結果例:

20 Per Page ▾ Format Preview ▾ < Prev 1 2 3 4 5 6 7 8 ... Next >

action	JSESSIONID	status
purchase	SD6SL5FF6ADFF89354	200
purchase	SD6SL5FF6ADFF89354	200
purchase	SD6SL5FF6ADFF89354	200
purchase	SD6SL5FF6ADFF89354	200
purchase	SD6SL5FF6ADFF89354	200

9. フィールドの順序を変更し、`JSESSIONID` が最初の列にくるようにします。(index=main sourcetype=access_combined_wcookie action=purchase status=200 file=success.do | table JSESSIONID, action, status)

結果例:

20 Per Page ▾ Format Preview ▾ < Prev 1 2 3 4 5 6 7 8 ... Next >

JSESSIONID	action	status
SD6SL5FF6ADFF89354	purchase	200
SD6SL5FF6ADFF89354	purchase	200
SD6SL5FF6ADFF89354	purchase	200
SD6SL5FF6ADFF89354	purchase	200
SD6SL5FF6ADFF89354	purchase	200

10. セッション ID はマーケティングデータでは「User Sessions」とよばれます。 `JSESSIONID` の名前を変更し、レポートとマーケティングデータを一致させます。(index=main sourcetype=access_combined_wcookie action=purchase status=200 file=success.do | table JSESSIONID, action, status | rename JSESSIONID as UserSessions)

結果例:

UserSessions	action	status
SD6SL5FF6ADFF89354	purchase	200
SD6SL5FF6ADFF89354	purchase	200
SD6SL5FF6ADFF89354	purchase	200
SD6SL5FF6ADFF89354	purchase	200

11. sort コマンドを使用して UserSessions をソートします。(index=main sourcetype=access_combined_wcookie action=purchase status=200 file=success.do | table JSESSIONID, action, status | rename JSESSIONID as UserSessions | sort UserSessions)
12. UserSessions 値の中には複数回表示されるものもあります。 また、**統計**タブで返されるイベント数にも注意します。
13. sort コマンドを解除し、dedup を使用して同一のセッション値を削除します。(index=main sourcetype=access_combined_wcookie action=purchase status=200 file=success.do | dedup JSESSIONID | table JSESSIONID, action, status | rename JSESSIONID as UserSessions)

結果例:

UserSessions	action	status
SD1SL7FF6ADFF89341	purchase	200
SD8SL8FF6ADFF4957	purchase	200
SD2SL10FF6ADFF4955	purchase	200

14. いくつのイベントが**統計**タブに記載されていますか？

備考: ベストな成果を得るため、dedup はサーチのできるだけ早い段階で行ってください。

15. action および status フィールドの表示はデータのサニティーチェックに適していましたが、マーケティングチームはこれらを表示させる必要はありません。 これらをテーブル表示から削除します。(index=main sourcetype=access_combined_wcookie action=purchase status=200 file=success.do | dedup JSESSIONID | table JSESSIONID | rename JSESSIONID as UserSessions)

結果例:

UserSessions
SD1SL7FF6ADFF89341
SD8SL8FF6ADFF4957
SD2SL10FF6ADFF4955
SD3SL5FF3ADFF89564