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 12 - ルックアップの作成

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
このラボ実習では、Buttercup Games プロダクトの補足情報を提供する自動ルックアップを新規に作成します。

シナリオ: ウェブアプリケーションデータには販売されたプロダクトの名称および価格情報は含まれません。レポートのユーザーはプロダクト ID だけではなく、レポートで使用されるプロダクト名も参照したいと思っています。

タスク 1:ルックアップファイルをダウンロードして検証します。

1. 新しいブラウザウィンドウを開き、http://splk.it/productdata に移動します。
2. **products.zip** ファイルがシステムにダウンロードされます。
3. アーカイブツールを使用してファイルを解凍します。
4. 解凍したら、products.csv ファイルが確認できます。
5. Splunk Web のインスタンスのブラウザウィンドウに戻るか、新しいウィンドウを開きます。

6. 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。）

タスク 2: ルックアップファイルを追加してルックアップ定義を作成します。

7. **設定 > ルックアップ > ルックアップテーブルファイル**に移動します。
8. **新しいルックアップ テーブル ファイル**をクリックします。
9. これらの値でルックアップテーブルを保存します:
   - 宛先 App: *search*
   - ファイル: products.csv ファイル
   - 宛先ファイル名**:** *products.csv*
10. **設定 > ルックアップ > ルックアップ定義**に移動します。
11. **Search & Reporting** が **App コンテキスト**用に選択されていることを確認し、**新しいルックアップ定義**をクリックします。
12. これらの値でルックアップテーブルを保存します:
    - 宛先 App: *search*
    - 名前: products_lookup
    - タイプ**:** ファイルベース
    - ルックアップファイル: *products.csv*
13. サーチビューに戻ります。
14. 「`inputlookup`」 コマンドを使用し、ルックアップ定義が正しく作成されているかを検証します。

結果例:

Code	categoryId	price	productId	product_name
A	STRATEGY	24.99	DB-SG-G01	Mediocre Kingdoms
B	STRATEGY	39.99	DC-SG-G02	Dream Crusher
C	STRATEGY	24.99	FS-SG-G03	Final Sequel
D	SHOOTER	24.99	WC-SH-G04	World of Cheese
E	TEE	9.99	WC-SH-T02	World of Cheese Tee
F	STRATEGY	4.99	PZ-SG-G05	Puppies vs. Zombies

タスク 3: サーチにルックアップを使用します。

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。**

15. ユーザーがプロダクトを正常に購入したすべてのイベントのウェブアプリケーションデータをサーチします。
16. 「ルックアップ」コマンドを使用し、先ほど作成したルックアップテーブルを参照します。 ルックアップ「`productId`」をイベントデータの「`productId`」フィールドに一致させます。 OUTPUT 関数を使用し、「`ProductName`」フィールドに「プロダクト_名前」ルックアップテーブルデータを出力します。
17. これで「`ProductName`」フィールドがフィルードリストに表示されました。

例:

other 100+
a productId 15
a ProductName 15
a punct 2
a referer 15
a referer_domain 1

18. 「Stats Count」関数を使用して、サーチを「ProductName」別イベントでのカウントに変更します。

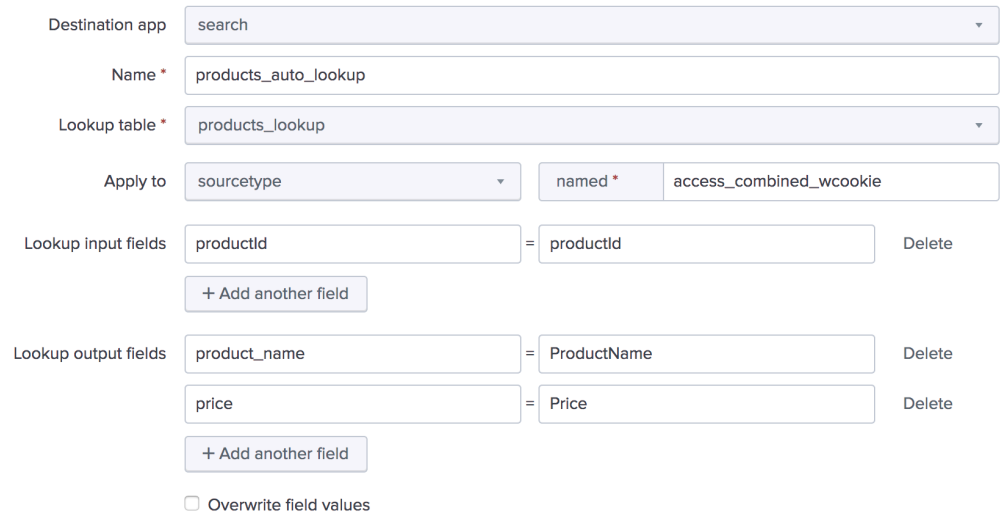
結果例:

ProductName	count
Benign Space Debris	935
Curling 2014	935
Dream Crusher	1308
Final Sequel	1155
Fire Resistance Suit of Provolone	1192
Grand Theft Scooter	61

タスク 4: 自動ルックアップ定義を作成します。

19. **設定 > ルックアップ > 自動ルックアップ**に移動します。
20. これらの値で自動ルックアップを保存します:
    - 宛先 App: *search*
    - 名前: *products_auto_lookup*
    - ルックアップテーブル: *products_lookup*
    - 適用先: *sourcetype*
    - 名前: *access_combined_wcookie*
    - ルックアップ入力フィールド: *productId = productId*
    - ルックアップ出力フィールド: *product_name = ProductName*
      price = Price

例:



タスク 5: 自動ルックアップが機能しているかを検証します。

21. サーチビューに戻ります。
22. ユーザーがプロダクトを正常に購入したすべてのイベントのウェブアプリケーションデータをサーチします。`stats sum` 関数を使用して、`ProductName` 別で `Price` フィールドを合計します。「`Revenue`」フィールドに名前をつけます。
23. 「`Sort`」コマンドを使用して、最も収益を上げたプロダクトを検索します。 モジュールクイズで思い出すよう促される場合があるため「`ProductName`」に留意してください。
24. レポートを「セールスダッシュボード」にダッシュボードパネルとして保存します。

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 12 - ルックアップの作成 (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
このラボ実習では、Buttercup Games プロダクトの補足情報を提供する自動ルックアップを新規に作成します。

シナリオ: ウェブアプリケーションデータには販売されたプロダクトの名称および価格情報は含まれません。レポートのユーザーはプロダクト ID だけではなく、レポートで使用されるプロダクト名も参照したいと思っています。

タスク 1: ルックアップファイルをダウンロードして検証します。

1. 新しいブラウザウィンドウを開き、http://splk.it/productdata に移動します。
2. **products.zip** ファイルがシステムにダウンロードされます。
3. アーカイブツールを使用してファイルを解凍します。
4. 解凍したら、products.csv ファイルが確認できます。
5. Splunk Web のインスタンスのブラウザウィンドウに戻るか、新しいウィンドウを開きます。
6. 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。）

タスク 2: ルックアップファイルを追加してルックアップ定義を作成します。

7. **設定 > ルックアップ > ルックアップテーブルファイル**に移動します。
8. **新しいルックアップ テーブル ファイル**をクリックします。
9. これらの値でルックアップテーブルを保存します:
   - 宛先 App: *search*
   - ファイル: products.csv ファイル
   - 宛先ファイル名**:** *products.csv*
10. **設定 > ルックアップ > ルックアップ定義**に移動します。
11. **Search & Reporting** が **App コンテキスト**用に選択されていることを確認し、**新しいルックアップ定義**をクリックします。
12. これらの値でルックアップテーブルを保存します:
    - 宛先 App: *search*
    - 名前: products_lookup
    - タイプ**:** ファイルベース
    - ルックアップファイル: *products.csv*
13. サーチビューに戻ります。
14. 「`inputlookup`」 コマンドを使用し、ルックアップ定義が正しく作成されているかを検証します。

    (| inputlookup products_lookup)

結果例:

Code	categoryId	price	productId	product_name
A	STRATEGY	24.99	DB-SG-G01	Mediocre Kingdoms
B	STRATEGY	39.99	DC-SG-G02	Dream Crusher
C	STRATEGY	24.99	FS-SG-G03	Final Sequel
D	SHOOTER	24.99	WC-SH-G04	World of Cheese
E	TEE	9.99	WC-SH-T02	World of Cheese Tee
F	STRATEGY	4.99	PZ-SG-G05	Puppies vs. Zombies

タスク 3: サーチにルックアップを使用します。

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。**

15. ユーザーがプロダクトを正常に購入したすべてのイベントのウェブアプリケーションデータをサーチします。

    (index=main sourcetype=access_combined_wcookie status=200 file=success.do)
16. 「ルックアップ」コマンドを使用し、先ほど作成したルックアップテーブルを参照します。 ルックアップ「`productId`」をイベントデータの「`productId`」フィールドに一致させます。 OUTPUT 関数を使用し、「`ProductName`」フィールドに「プロダクト_名前」ルックアップテーブルデータを出力します。

    (index=main sourcetype=access_combined_wcookie status=200 file=success.do | lookup products_lookup productId as productId OUTPUT product_name as ProductName)
17. これで「`ProductName`」フィールドがフィールドリストに表示されました。

例:

other 100+
a productId 15
a ProductName 15
a punct 2
a referer 15
a referer_domain 1

18. 「Stats Count」関数を使用して、サーチを「ProductName」別イベントでのカウントに変更します。

    (index=main sourcetype=access_combined_wcookie status=200 file=success.do | lookup products_lookup productId as productId OUTPUT product_name as ProductName | stats count by ProductName)

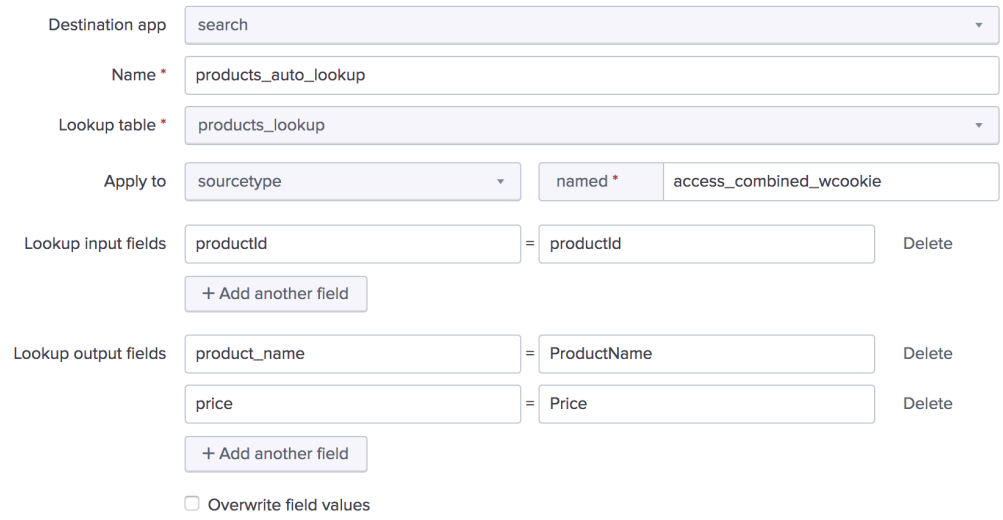
結果例:

ProductName	count
Benign Space Debris	935
Curling 2014	935
Dream Crusher	1308
Final Sequel	1155
Fire Resistance Suit of Provolone	1192
Grand Theft Scooter	61

タスク 4: 自動ルックアップ定義を作成します。

19. **設定 > ルックアップ > 自動ルックアップ**に移動します。
20. これらの値で自動ルックアップを保存します:
    - 宛先 App: *search*
    - 名前: *products_auto_lookup*
    - ルックアップテーブル: *products_lookup*
    - 適用先: *sourcetype*
    - 名前: *access_combined_wcookie*
    - ルックアップ入力フィールド: *productId = productId*
    - ルックアップ出力フィールド: *product_name = ProductName*
      price = Price

例:



タスク 5: 自動ルックアップが機能しているかを検証します。

21. サーチビューに戻ります。
22. ユーザーがプロダクトを正常に購入したすべてのイベントのウェブアプリケーションデータをサーチします。`stats sum` 関数を使用して、`ProductName` 別で `Price` フィールドを合計します。「`Revenue`」フィールドに名前をつけます。

    (index=main sourcetype="access_combined_wcookie" file=success.do status=200 | stats sum(Price) as Revenue by ProductName)
23. 「`Sort`」コマンドを使用して、最も収益を上げたプロダクトを検索します。 モジュールクイズで思い出すよう促される場合があるため「`ProductName`」に留意してください。

    (index=main sourcetype="access_combined_wcookie" file=success.do status=200 | stats sum(Price) as Revenue by ProductName | sort -Revenue)

    (Dream Crusher)
24. レポートを「セールスダッシュボード」にダッシュボードパネルとして保存します。