SSL セッション ツール

バージョン 1.6.2
初版発行日 : 3/1/16

SSL セッション ツールは、Cisco SSL アプライアンスによって生成されたエクスポート済みセッション ログ内の SSL セッション ログ情報を解析するために使用します。このガイドでは、SSL セッション ツールとそのインストール方法について概要を説明します。このツールでは、バイナリの SSL セッション ログ ファイルを解析したり、分類されたフィールドでフィルタリングしたり、他のアプリケーションで使用するために CSV 形式でデータを保存したりすることができます。SSL セッション ツールは、コマンド ラインまたは GUI モードで使用できます。

すべてのコマンドについての説明は、ツールによって生成されたドキュメントの「SSL セッション ツールの使用、5 ページ」に記載されています。ツールおよびツールによるドキュメントは `cisco.com` で入手できます。

このバージョンの SSL セッション ツールは、SSL 3.5 以降をサポートしています。

SSL セッション ツールは、デフォルトでは SSL 3.8.x セッションのログ ファイルで動作します。別のバージョンの SSL アプライアンス ソフトウェアでエクスポートされたファイルを使用するには、ツールで SSL のバージョンを設定します(コマンド ラインと GUI のどちらでも使用できます)。たとえば、SSL 3.5.x で出力されたセッション ファイルを使用する場合は 3.5、3.6.x の場合は 3.6、3.7.x の場合は 3.7 というようにバージョンを設定します。

このバージョンの SSL セッション ツールは、Blue Coat Reporter で使用できるように、スペース区切りでのデータ エクスポートをサポートしています。Reporter 用の形式で `.csv` ファイルを出力するには、コマンドラインで `-R` オプションを使用します。構文の詳細については、SSL セッション ツールの使用、5 ページを参照してください。

Cisco SSL アプライアンス ソフトウェアは、シスコおよびサード パーティ製ソフトウェアのプロバイダーが定めるライセンス契約条件に従います。

詳細については、次の項を参照してください。

Cisco Systems, Inc.
www.cisco.com

要件

- Windows または Linux を実行している x86 または x86-64 のパーソナル コンピュータ。システムは Python 2.6.x または 2.7.x を実行している必要があります。Python 3.x はサポートされていません。このシステムは、このドキュメントでは「ホスト」システムと呼んでいます。
- Python バージョン 2.6.x または 2.7.x
- pyOpenSSL バージョン 0.12
- wxPython 2.9（UI モードに必要）
- ツールを実行する Cisco SSL アプライアンス sslsessions 出力ファイル。

SSL セッション ツールのインストール

SSL セッション ツールは、ホスト システムにインストールする必要がある zip 形式のパッケージとして提供されます。パッケージのファイル名は `sslsessions-n.n.n.zip` で、`n.n.n` にはパッケージのバージョン番号が入ります。インストール後は、コマンド ラインまたは GUI アプリケーション（wxPython 2.9 が必要）からツールを実行できます。

Linux 環境でのパッケージのインストール

ステップ 1 ホスト システムにファイルをコピーします。

ステップ 2 コマンド ラインで次を入力します。

```
unzip sslsessions-n.n.n.zip
cd sslsessions-n.n.n
python setup.py install
```

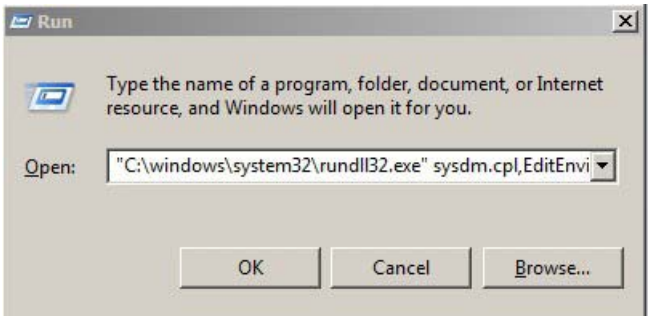
ヒント ソフトウェアのインストール権限を持つユーザとしてログオンする必要があります。

Windows 環境でのパッケージのインストール

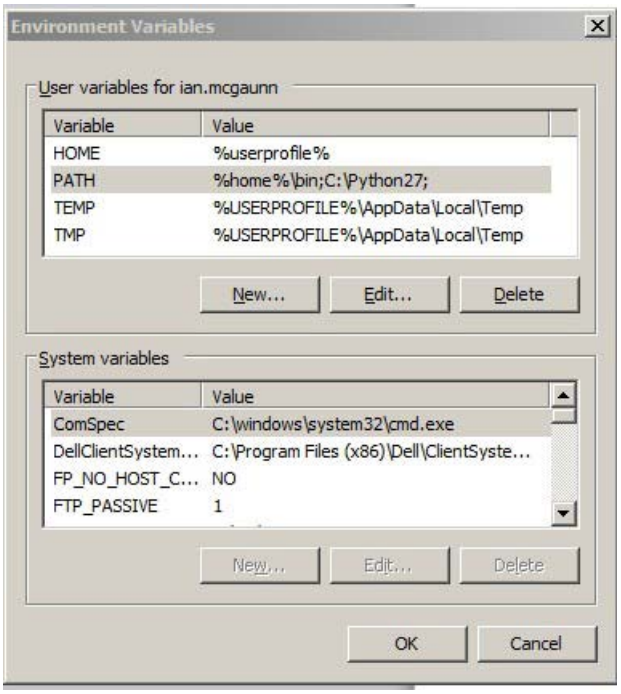
Windows システムの場合、インストールを完了するには追加の手順が必要になります。スクリプトをインストールしたディレクトリをシステム パスに追加します。これを行うには、[Run] ウィンドウを開きます（[Start] をクリックして「run」と入力し、[Enter] をクリック）。

ステップ 1 [Run] ウィンドウで、次を入力します。

```
C:\windows\system32\rundll32.exe" sysdm.cpl,EditEnvironmentVariables
```



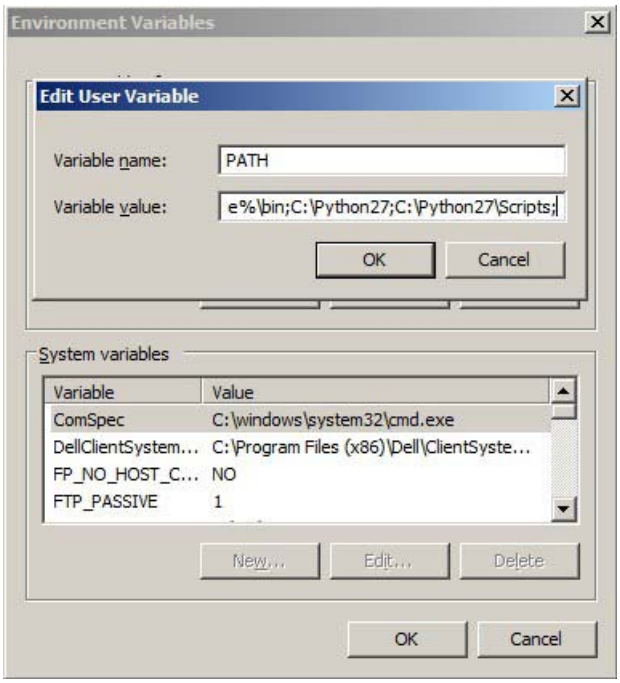
ステップ 2 [OK] をクリックします。[Environment Variables] ウィンドウが表示されます。



ステップ 3 [User variables for <user name>] というラベルがある上部のパネルで、[PATH] エントリを選択して [Edit] をクリックします。[Edit User Variable] ウィンドウが表示されます。

ステップ 4 [Variable value] フィールドで、解凍したファイルのパスをセミコロンの後に入力します。次に例を示します。

（注） この例では Python 2.7 を使用しています。SSL セッション ツールには、Python 2.6.x または 2.7.x が必要です（Python 3.x はサポートされていません）。



ステップ 5 入力内容が正しければ、[OK] をクリックします。[Environment Variables] ウィンドウを閉じます。

インストールの確認

SSL 診断ツールが正しくインストールされていることを確認するには、コマンド プロンプトを開き、次を入力します。

```
sslsessions.py --version
```

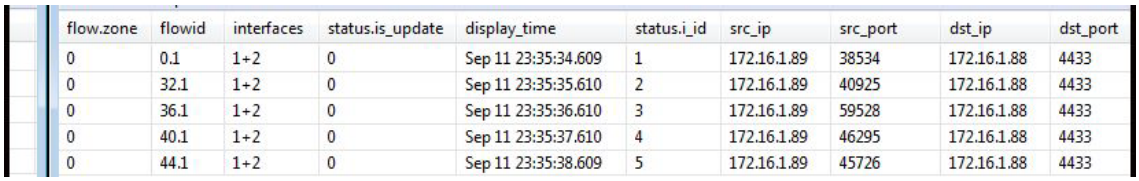
次の内容が表示されます。

```
host:<path>/sslsessions/$ sslsessions.py --version
Usage: sslsessions.py [options] [output-path]
Export csv data from filtered ssl session log files. [output-path] is required.
If no command line arguments are provided, the tool starts up in ui mode.
Set the mode (-M or --mode) to csv or report when using the tool from command line.
Version: n.n.n
```

SSL セッション ツールの使用

SSL セッション ログ ツールは、バイナリの SSL セッション ログ ファイルの解析、各種フィールドでのフィルタリング、および CSV 形式でのデータの保存が可能なアプリケーションです。このツールは、コマンド ライン モードまたは UI モードで使用できます。

UI モードでは、アプリケーションは次のように表示されます。



flow.zone	flowid	interfaces	status.is_update	display_time	status.i_id	src_ip	src_port	dst_ip	dst_port
0	0.1	1+2	0	Sep 11 23:35:34.609	1	172.16.1.89	38534	172.16.1.88	4433
0	32.1	1+2	0	Sep 11 23:35:35.610	2	172.16.1.89	40925	172.16.1.88	4433
0	36.1	1+2	0	Sep 11 23:35:36.610	3	172.16.1.89	59528	172.16.1.88	4433
0	40.1	1+2	0	Sep 11 23:35:37.610	4	172.16.1.89	46295	172.16.1.88	4433
0	44.1	1+2	0	Sep 11 23:35:38.609	5	172.16.1.89	45726	172.16.1.88	4433

コマンド ライン モードの例は次のとおりです。

```
sslsessions.py -M csv -S ../sessions_logs/ssl_session_log-20150115T173613
-d *.google.com ../session_log_output
```

コマンド ラインでは、次のオプションを使用できます。

使用方法:

```
sslsessions.py [options]
```

オプション:

オプション	結果
`-h, --help`	このヘルプ メッセージを表示して終了
`-s SRC, --src=SRC`	送信元アドレスのワイルドカード フィルタ
`-d DST, --dst=DST`	送信先アドレスのワイルドカード フィルタ
`-S SESSLOG_PATH, --sesslog-path=SESSLOG_PATH`	SSL セッション ログのパス
`-p SESSLOG_PATTERN, --sesslog-pattern=SESSLOG_PATTERN`	SSL セッション ログのファイル名のワイルドカード フィルタ
`-x, --dump-state dump`	SSL のフロー状態
`-m, --dump-messages dump`	SSL のメッセージ
`-5, --dump-x509-fp`	X.509 フィンガープリントのダンプ
`-c DUMP_X509_FIELD, --dump-x509-field=DUMP_X509_FIELD`	X.509 フィールドのダンプ(例:CN)
`-e ERR_FLOWS, --err-flows=ERR_FLOWS`	エラー コードのフィルタ
`-M MODE, --mode=MODE`	出力モード:CSV、レポートまたは UI(デフォルト)
`-t TIMESTAMP, --timestamp=TIMESTAMP`	タイムスタンプのフィルタ
`-B BEGIN_TIMESTAMP, --begin-timestamp=BEGIN_TIMESTAMP`	タイムスタンプ フィルタの開始

オプション	結果
-E END_TIMESTAMP, --end-timestamp=END_TIMESTAMP	タイムスタンプ フィルタの終了
-l LINES, --lines=LINES	CSV ファイルごとの行数
-f PRE_DECISION, --pre-decision=PRE_DECISION	事前決定(誤検出)フローのダンプ
-r, --dump-hsm-resigning-ca	hsm 再署名 ca 情報のダンプ
-V SSLNG_VERSION, --sslng-version=SSLNG_VERSION	使用する sslv アプライアンス エラー コードのバージョン 3.5、3.6、3.7、または 3.8
v, --version	バージョン番号を表示して終了
-R, --reporter Output	Reporter 形式の .csv ファイル(スペース区切り)。

API

ツールのイベント読み取りライブラリは、Python のインストールからもアクセスできます。

以下は、コマンド ラインのパス引数から 1 つ目のログ ファイルを読み取り、いくつかのフィールドに出力するコマンドの例です。

```
import os, sys
from ssl_sessions import readevents
log_path = sys.argv[1]
for event in readevents.iter_session_log_events(
os.path.join(log_path, 'ssl_session_log.1.bin'), '3.8',
readevents.read_cert_store(log_path),
readevents.read_hsm_resigning_cas(log_path)):
print event.flowid.ID, event.flow, event.subject
if not event.match_result('OK'):
print 'ERROR:', event.result
```

ssl_sessions.readevents

```
class ssl_sessions.readevents.SessionLogEvent (data, sslng_version, cert_store=None,
hsm_resigning_cas=None)
```

単一のセッション ログ イベントを表すクラス。

```
ssl_sessions.readevents.asn_parse_san_entry (data)
```

ASN エントリを解析して SAN IP および DNS 名のエントリを抽出します。

パラメータ:`data (str)` -- SAN エントリの文字列データ。

戻り値:(`'san_ip', str`)または(`'dns_name, str`)のタプルのリスト

`ssl_sessions.readevents.inet_ntoa_6 (address)`

ネットワーク形式の IPv6 アドレスをテキストに変換します。

パラメータ:`address (str)` -- バイナリ アドレス。

戻り値:文字列形式の IP アドレス。

`ssl_sessions.readevents.iter_all_filtered_events (options, progress_notify=None)`

オプションによってフィルタリングされたイベントのみを返すイテレータ

`ssl_sessions.readevents.read_cert_store (path)`

証明書ストアの pem ファイルを、証明書名の値の dicts への dict マッピング フィンガープリントに読み込みます。

パラメータ:`path (str)` -- セッション ログ ディレクトリのパス。

戻り値:証明書情報ディクショナリへのフィンガープリント。

`ssl_sessions.readevents.read_hsm_resigning_cas (path)`

json 形式から HSM 再署名 CA 情報を読み取ります。

パラメータ:`path (str)` -- セッション ログ ディレクトリのパス。

戻り値:HSM 情報ディクショナリへのフィンガープリント。

支援が必要な場合

シスコ サポート

ドキュメントの入手、Cisco Bug Search Tool(BST)の使用、サービス リクエストの提出、および Cisco SSL アプライアンスの詳細情報については、http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html の『**What's New in Cisco Product Documentation**』を参照してください。

『**What's New in Cisco Product Documentation**』は、シスコの新規および改訂版の技術マニュアルの一覧も示し、RSS フィードとして購読できます。また、リーダー アプリケーションを使用してコンテンツをデスクトップに配信することもできます。RSS フィードは無料のサービスです。

ご質問がある場合、またはシスコ SSL アプライアンスに関するサポートが必要な場合は、シスコ サポートにお問い合わせください。

- シスコ サポート サイト:http://support.cisco.com/。
- シスコ サポート(tac@cisco.com)に電子メールをお送りください。
- シスコ サポートの電話番号:1-408-526-7209 または 1-800-553-2447。