

非暗号化ハンドシェイクを行う SSL Web サーバとの接続または接続解除を行うキープアライブスクリプトの例

目次

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[サンプル スクリプト](#)

[関連情報](#)

概要

このスクリプトは、SSL バージョン 3.0 を実行する Secure Socket Layer (SSL) Web サーバに接続します。サーバに接続し、非暗号化ハンドシェイクを実行し、接続解除します。このドキュメントでは、スクリプト キープアライブの実装についても説明します。このスクリプトティング方法は、リモート アクセス サーバ (RAS) のダイヤルアップ クライアント、ターミナル プログラム、および一般的なスクリプト ユーティリティに存在する機能と密接に関連しています。この機能は、WebNS の豊富なスクリプト言語を使用します。

簡単なソケット Application Program Interface (API) と (接続して下さい/接続解除/送信/受信)、スクリプト キープアライブ与えますユーザに自身のプロトコルを合わせる機能を完了して下さいまたはサービスの信頼できる ALIVE DOWN 提供するためにステップの自身のシーケンスを書いて下さい。スクリプト キープアライブ 機能性なしで、FTP、HTTP、ICMP および TCP に現在制限されます。しかしスクリプト キープアライブを使うと現在のプロトコルの上あなた自身のスクリプトを書くことによって残ることができます。たとえば、WebNS が KeepAlive型 POP3 を構築するように要求しないで POP3 サーバに接続するためにとりわけある調子を与えられるスクリプトを開発できます。この機能は顧客が特定の必要条件に適するために自身のカスタム キープアライブを作成することを可能にします。これが Content Services Switch (CSS) のコンポーネントであるが、カスタムスクリプトは Cisco Technical Assistance Center (Cisco TAC) によってサポートされません。

スクリプト キープアライブは下記の TAC によって公式のサポートを提供されないし、テストされ、あなた自身の思慮分別で利用可能です。

前提条件

要件

WebNS リッチ スクリプト言語の習熟度。

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づくものです。

- WebNS バージョン 3.x 以上
- CSS 11x00 シリーズ

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。このドキュメントで使用するすべてのデバイスは、クリアな (デフォルト) 設定で作業を開始しています。ネットワークが稼働中の場合は、コマンドが及ぼす潜在的な影響を十分に理解しておく必要があります。

サンプル スクリプト

下記のスクリプトが暗号化されていないハンドシェイクと動作する SSL Webサーバに接続し、切らのに使用することができます。

```
!--- No echo. !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! !--- Filename:
ap-kal-ssl-port !--- Parameters: HostName ! !--- Description: !--- This script will connect to
an SSL Web server running SSL !--- version 3.0. Connect to the server, do the non-encrypted !---
handshake, and disconnect. ! !--- Parameters: !--- SSL-IP: Address of the SSL Accelerator !---
SSL-Port: Port for the SSL Accelerator ! !--- Failure Upon: !--- 1. Not establishing a
connection with the host. !--- 2. Not receiving a positive authentication. ! !--- Author: KGS !-
-- Last Tested: 9/27/01 ! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! if
${ARGS} [#] "==" "0" echo "Usage: ap-kal-ssl-port \'SSL-IP [SSL-Port]\'" exit script 1 endbranch
!--- Defines. set HostName "${ARGS}[1]" set SSL-Port "443" if ${ARGS} [#] "GT" "1" set SSL-Port
"${ARGS}[2]" endbranch !--- Connect to the remote host. set EXIT_MSG "Connection Failure for
${HostName}:${SSL-Port}" socket connect host ${HostName} port ${SSL-Port} tcp 2000 !--- Send the
GET request for the Web page. set EXIT_MSG "Send: Failed" !--- Send over the hex for the fields:
!--- [Handshake: 0x16] [Version: 0x03 0x00] [Length: 0x00 0x59] !--- [Client Hello: 0x01]
[Length: 0x00 0x00 0x55] [Version: 0x03 0x00] !--- [Random (32bit) #: 0x39 -> 0xff] [Session
Length: 0x20] !--- [Session ID (32bit): 0x3a -> 0x5d] [Cipher Length: 0x00 0x0e] !--- [Cipher
Suite: 0x00 -> 0x00 (Last Byte in stream)] !--- Break the request into two send requests, as
there is a 128 byte !--- max on quoted text parameters. socket send ${SOCKET}
"1603000059010000550300392ae5530da35d89041b4beaa42891470e49 351c3bfef7631296139928dd7fff203a"
raw socket send ${SOCKET} "9a0ed92a4e4f66d75ecce24c3a361efc26ab86310c4b9e7271a1317d9
7635d000e0004ffe0000a00640062000300060100" raw !--- Wait for a good status code. set EXIT_MSG
"Waitfor: Failed" !--- Wait for a handshake message (0x16), paired with the version !--- of SSL
(0x03 0x00). socket waitfor ${SOCKET} "160300" 2000 raw !--- Wait for the specific server hello
(0x02). socket waitfor ${SOCKET} "02" 2000 raw !--- Wait for the version again (it appears
twice: 0x03 0x00). socket waitfor ${SOCKET} "0300" 2000 raw no set EXIT_MSG socket disconnect
${SOCKET} exit script 0
```

関連情報

- [CSS 11000 シリーズ コンテント サービス スイッチのハードウェア サポート \(英語\)](#)
- [CSS 11500](#)
- [CSS11500 \(登録ユーザのみ\) のためのソフトウェアダウンロード](#)
- [テクニカルサポートとドキュメント - Cisco Systems](#)