

Conectar do script do Keepalive da amostra/disconexão a um servidor de Web SSL que é executado com um aperto de mão NON-cifrado

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Exemplo de script](#)

[Informações Relacionadas](#)

Introdução

Este script conectará a um 3.0 running da versão de SSL do servidor de Web do Secure Socket Layer (SSL). Conecte ao server, faça o aperto de mão NON-cifrado, e desligue-o. Este documento igualmente endereça a aplicação dos keepalive de script. Este método do script é o mais estreitamente relacionado à funcionalidade, que esta presente em clientes dialup do servidor de acesso remoto (RAS), em programas terminal, e em utilidades gerais do script. Esta característica utiliza o linguagem de script rico de WebNS.

Termine com um Application Program Interface do soquete simples (API) (conecte/disconexão/enviam/recebem), um keepalive de script dará ao usuário a capacidade para costurar seu próprio protocolo, ou escreva sua própria sequência das etapas para fornecer um `VIVO` ou um `estado inativo` seguro de um serviço. Sem a funcionalidade do keepalive de script, você é limitado atualmente ao FTP, ao HTTP, ao ICMP, e ao TCP. Com keepalive de script, contudo, você pode permanecer sobre os protocolos atual escrevendo seus próprios scripts. Por exemplo, você pode desenvolver um script tonificado especificamente para conectar a um server POP3 sem exigir WebNS construir um tipo keepalive POP3. Esta característica permite que os clientes criem seu próprio Keepalives feito sob encomenda para serir suas exigências específicas. Embora este seja um componente do Content Services Switch (CSS), os scripts personalizados não são apoiados pelo centro de assistência técnica da Cisco (tac Cisco).

Os keepalive de script abaixo não são apoiados oficialmente pelo TAC, mas foram testados, e estão disponíveis para o uso em sua própria discrição.

Pré-requisitos

Requisitos

Familiaridade com o linguagem de script dos ricos de WebNS.

Componentes Utilizados

As informações neste documento são baseadas nas versões de software e hardware:

- WebNS versões 3.x e superiores
- 11x00 Series CSS

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

Exemplo de script

O script abaixo pode ser usado para conectar e desligar a um servidor de Web SSL que é executado com um apertado de mão NON-cifrado.

```
!--- 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 351c3bfeb7631296139928dd7fff203a"
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
```

Informações Relacionadas

- [Suporte a hardware dos CSS 11000 Series Content Services Switch](#)
- [Suporte a hardware dos CSS 11500 Series Content Services Switch](#)
- [Download do software para CSS11500 \(clientes registrados somente\)](#)
- [Suporte Técnico e Documentação - Cisco Systems](#)