

Prove o script do Keepalive para abrir e fechar um soquete nas portas especificadas do usuário

Índice

[Introdução](#)

[Antes de Começar](#)

[Convenções](#)

[Pré-requisitos](#)

[Componentes Utilizados](#)

[Exemplo de script](#)

[Informações Relacionadas](#)

[Introdução](#)

Este documento endereça a aplicação dos keepalive de script. Este script abrirá e fechará um soquete nas portas especificadas do usuário. O fim será um FIN um pouco do que um RST. Se uma das portas falha, o serviço estará declarado para baixo. 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.

[Antes de Começar](#)

[Convenções](#)

Para obter mais informações sobre convenções de documento, consulte as [Convenções de dicas técnicas Cisco](#).

Pré-requisitos

Não existem requisitos específicos para este documento.

Componentes Utilizados

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

- WebNS versões 3.x e superiores
- CSS 11000 Series

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 você estiver trabalhando em uma rede ativa, certifique-se de que entende o impacto potencial de qualquer comando antes de utilizá-lo.

Exemplo de script

O script abaixo pode ser usado para abrir e fechar um soquete nas portas especificadas do usuário.

```
!--- No echo. !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! !--- Filename:
ap-kal-tcp-ports !--- Parameters: Service Address, TCP Port(s) ! !--- Description: !--- This
script will open and close a socket on the user specified ports. !--- The close will be a FIN
rather than a RST. If one of the ports fails, !--- the service will be declared down ! !---
Failure Upon: !--- 1. Not establishing a connection with the host on one of the specified ports.
! !--- Note: Does not use output. !--- Will handle out of sockets scenario. ! !--- Tested: KGS
12/18/01 ! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! set OUT-OF-SOCKETS
"785" set NO-CONNECT "774" !--- Make sure the user has a qualified number of arguments. if
${ARGS}[#] "LT" "2" echo "Usage: ap-kal-tcp-ports \'ipAddress tcpPort1 [tcpPort2 tcpPort3...]\'"
exit script 1 endbranch set SERVICE "${ARGS}[1]" !--- echo "SERVICE = ${ARGS}[1]" var-shift ARGS
while ${ARGS}[#] "GT" "0" set TCP-PORT "${ARGS}[1]" var-shift ARGS function SOCKET_CONNECT call
!--- If out of sockets, exit, and look for sockets on the next KAL interval. if RETURN "=="
"${OUT-OF-SOCKETS}" set EXIT_MSG "Exceeded number of available sockets, skipping until next
interval." exit script 0 endbranch !--- Valid connection, look to see if it was good. if RETURN
"==" "${NO-CONNECT}" set EXIT_MSG "Connect: Failed to connect to ${SERVICE}:${TCP-PORT}" exit
script 1 endbranch endbranch no set EXIT_MSG exit script 0 function SOCKET_CONNECT begin set
CONTINUE_ON_ERROR "1" socket connect host ${SERVICE} port ${TCP-PORT} tcp 2000 set SOCKET-STAT
"${STATUS}" set CONTINUE_ON_ERROR "0" socket disconnect ${SOCKET} graceful function
SOCKET_CONNECT return "${SOCKET-STAT}" function SOCKET_CONNECT end
```

Informações Relacionadas

- [Sustentação do produto dos CSS 11000 Series Content Services Switch](#)
- [Sustentação do produto dos CSS 11500 Series Content Services Switch](#)
- [Software da transferência CSS11000 \(clientes registrados somente\)](#)
- [Software da transferência CSS11500 \(clientes registrados somente\)](#)
- [Suporte Técnico e Documentação - Cisco Systems](#)