

Script del keepalive de la muestra para marcar la página web para el Webstring

Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Secuencia de comandos de ejemplo](#)

[Información Relacionada](#)

[Introducción](#)

Este script marca la página web para la cadena de la red. Si la cadena falta, marque el servicio como abajo. Este script se utiliza con cualquier clase de página, especialmente las dinámicas que se generan vía los scripts, ColdFusion, y así sucesivamente. Este documento también dirige la implementación de los keepalives en secuencia de comandos. Este método de scripting está el más estrechamente vinculado a las funciones, que está presente en los clientes de marcación manual del Remote Access Server (RAS), los programas para terminal, y las utilidades generales del scripting. Esta característica utiliza el lenguaje de la secuenciación de comandos rico de WebNS.

Complete con un Application Program Interface del socket simple (API) (conecte/desconexión/envían/reciben), un dará al usuario del keepalive en secuencia de comandos la capacidad de adaptar su propio protocolo, o escribió su propia la secuencia de para proporcionar un estado ALIVE o DOWN confiable de algún servicio de los pasos. Sin las funciones del keepalive en secuencia de comandos, le limitan actualmente al FTP, al HTTP, al ICMP, y al TCP. Con los keepalives en secuencia de comandos, sin embargo, usted puede permanecer encima de los protocolos actuales escribiendo sus propios scripts. Por ejemplo, usted puede desarrollar un script entonado específicamente para conectar con un servidor POP3 sin requerir WebNS construir un tipo de keepalive POP3. Esta característica permite que los clientes creen su propio Keepalives de encargo para adaptarse a sus requisitos específicos. Aunque éste sea un componente del Content Services Switch (CSS), las secuencias de comandos personalizadas no son soportadas por el Centro de Asistencia Técnica de Cisco (TAC de Cisco).

Los keepalives en secuencia de comandos abajo no son soportados oficialmente por TAC, sino se han probado, y están disponibles para el uso en su propia discreción.

[prerrequisitos](#)

[Requisitos](#)

Familiaridad con el lenguaje de la secuenciación de comandos de los ricos de WebNS.

Componentes Utilizados

La información que contiene este documento se basa en las versiones de software y hardware.

- WebNS versión 3.x y posteriores
- 11x00 Series CSS

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener cualquier comando.

Secuencia de comandos de ejemplo

El script abajo se puede utilizar para marcar la página web para webstring.

```
!--- No echo. !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! !--- Filename:
ap-kal-httpstring !--- Parameters: WebsiteIP WebPage WebString [Port] !--- Requirements:
WebNS4.x or higher ! !--- Uses: !--- Checks the Web page for the Web string. If the string is
missing, !--- mark the service as down. Used with any sort of page, especially dynamic !--- ones
that are generated via scripts, ColdFusion, and so on. ! !--- Logic: !--- The script connects to
a Web server on port 80 by default. !--- It performs a GET on the specified page. !--- If the
Web string is returned, the service stays up. !--- If anything fails, the service is marked
down. ! !--- Notes: !--- The Web string is case-sensitive. !--- Only the first 10Kb of the
response is inspected. ! ! !--- Tested: 04/12/01-KGS !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! no set CONTINUE_ON_ERROR no
set EXIT_MSG !--- Make sure the user has the proper number of arguments. if ${ARGS}[#] "LT" "3"
echo "Usage: ap-kal-httpcheck \'WebsiteIP WebPage WebString [Port]\'" exit script 1 endbranch !-
-- Set variables corresponding to the args. set WebSite "${ARGS}[1]" set WebPage "${ARGS}[2]"
set WebString "${ARGS}[3]" set WebPort "80" if ${ARGS}[#] "GT" "3" set WebPort "${ARGS}[4]"
endbranch echo "Requesting ${WebPage} from ${WebSite} on port ${WebPort}." !--- Connect to the
remote server. set EXIT_MSG "Connect: Failed. Could not connect to ${WebSite} on port
${WebPort}" set CONTINUE_ON_ERROR "1" socket connect host ${WebSite} port ${WebPort} tcp if
${STATUS} "NEQ" "0" exit script 1 endbranch no set CONTINUE_ON_ERROR !--- Request the desired
Web page. set EXIT_MSG "Send: Failed. Could not send to ${WebSite}:${WebPort}" socket send
${SOCKET} "GET ${WebPage} HTTP/1.0\n\nHost: ${WebSite}:${WebPort}\n" !--- Look for the Web
string. set EXIT_MSG "Waitfor: Failed. Did not find [${WebString}]" set CONTINUE_ON_ERROR "1"
socket waitfor ${SOCKET} "${WebString}" case-sensitive if ${STATUS} "NEQ" "0" exit script 1
endbranch no set CONTINUE_ON_ERROR !--- Disconnect from the server. no set EXIT_MSG socket
disconnect ${SOCKET} graceful exit script 0
```

Información Relacionada

- [Soporte del hardware de los CSS 11000 Series Content Services Switch](#)
- [Soporte del hardware de los CSS 11500 Series Content Services Switch](#)
- [Descarga del software para CSS11500 \(clientes registrados solamente\)](#)
- [Soporte Técnico y Documentación - Cisco Systems](#)