Exemple de script Keepalive pour vérifier la page Web de la chaîne Web

Contenu

Introduction

Ce script vérifie la page Web pour la chaîne de Web. Si la chaîne manque, marquez le service en tant que vers le bas. Ce script est utilisé avec n'importe quel tri de page, particulièrement la dynamique qui sont générées par l'intermédiaire des scripts, ColdFusion, et ainsi de suite. Ce document adresse également l'implémentation des keepalives à base de script. Cette méthode de script le plus étroitement est liée à la fonctionnalité, qui est présente dans des clients distants du Remote Access Server (RAS), des programmes de terminal, et des utilitaires généraux de script. Cette caractéristique utilise le langage de script riche de WebNS.

Terminez-vous avec une interface de programmation simple de socket (API) (connectez/débranchement/send/receive), une keepalive à base de script donnera à l'utilisateur la capacité de travailler leur propre protocole, ou écrivez leur propre ordre des étapes pour fournir un `ACTIF` ou un état d'indisponibilité fiable d'un service. Sans fonctionnalité de keepalive à base de script, vous êtes actuellement limité au FTP, au HTTP, à l'ICMP, et au TCP. Avec des keepalives à base de script, cependant, vous pouvez rester sur les protocoles en cours à côté d'écrire vos propres scripts. Par exemple, vous pouvez développer un script spécifiquement modifié la tonalité pour se connecter à un serveur POP3 sans exiger de WebNS d'établir un type de keepalive POP3. Cette caractéristique permet à des clients pour créer leur propre Keepalives fait sur commande pour satisfaire à leurs exigences spécifiques. Bien que ce soit un composant du Commutateur de services de contenu (CSS), des scripts personnalisés ne sont pas pris en charge par le centre d'assistance technique Cisco (Cisco TAC).

Les keepalives à base de script ci-dessous ne sont pas officiellement prises en charge par TAC, mais ont été testées, et sont disponibles pour l'usage à votre propre discrétion.

Conditions préalables

Conditions requises

Connaissance de langage de script de riches de WebNS.

Composants utilisés

Les informations de ce document sont basées sur les versions de logiciel et matériel suivantes :

- Versions 3.x et ultérieures de WebNS
- Gamme 11x00 CSS

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

Exemple de script

Le script ci-dessous peut être utilisé pour vérifier la page Web pour 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
```

Informations connexes

- Support matériel de Commutateurs de services satisfaits de gamme 11000 CSS
- Support matériel pour les commutateurs de services de contenu de la gamme CSS 11500
- Téléchargement logiciel pour CSS11500 (clients enregistrés seulement)
- Support et documentation techniques - Cisco Systems