Dépannage des problèmes de connexion à ASR5500 en raison de sessions inactives noTTY

Contenu

Introduction

Ce document décrit comment dépanner des scénarios lorsque la connectivité Secure Shell (SSH) est perdue par les adresses IP de gestion du routeur de services d'agrégation (ASR5500/ASR 5000).

Problèmes de connexion aux noeuds ASR5500

Vous ne pouvez pas vous connecter aux noeuds de coeur de paquet ASR5500. La connexion SSH se termine immédiatement sans l'invite de connexion. Les connexions Telnet présentent un comportement similaire.

Étapes de dépannage

Étape 1. Essayez de vous connecter au noeud via la connexion console.

Étape 2. Dans la plupart des cas, aucun déroutement SNMP (Simple Network Management Protocol) spécifique n'est émis qui pourrait indiquer la cause de l'échec de la connexion.

Étape 3. Les journaux relatifs à la connexion, constamment présents dans les Syslogs, sont les suivants :

```
evlogd: [local-60sec55.607] [tacacs+ 37201 error] [5/0/8908 <vpnmgr:1> authen_events.c:598]
[software internal system critical-info syslog] Authentication failed for user **** on tty
/dev/pts/0, application ssh, remote IP address XX.XX.XX.XX
evlogd: [local-60sec55.623] [cli 30028 debug] [5/0/8908 <vpnmgr:1> luser_auth.c:1448] [context:
local, contextID: 1]  [software internal system syslog] Login attempt failure for user **** IP
address XX.XX.XX.XX - Access type ssh/sftp
evlogd: [local-60sec53.652] [tacacs+ 37201 error] [5/0/8908 <vpnmgr:1> authen_events.c:598]
[software internal system critical-info syslog] Authentication failed for user **** on tty
/dev/pts/0, application ssh, remote IP address XX.XX.XX.XX
evlogd: [local-60sec53.679] [cli 30028 debug] [5/0/8908 <vpnmgr:1> luser_auth.c:1448] [context:
local, contextID: 1]  [software internal system syslog] Login attempt failure for user **** IP
address XX.XX.XX.XX - Access type ssh/sftp#####
evlogd: [local-60sec2.942] [tacacs+ 37201 error] [5/0/8908 <vpnmgr:1> authen_events.c:598]
[software internal system critical-info syslog] Authentication failed for user epcats on tty
```

```
/dev/pts/0, application ssh, remote IP address YY.YY.YY.YY
```
Étape 4. La commande **show crash list all** affiche les incidents récents, notez que ceux liés à **vpnmgr** sont particulièrement importants.

Étape 5. La commande **show task resources all** garantit que les processus **vpnmgr** et **sshd** ne doivent pas être en surétat. **vpnmgr** est responsable de la gestion du pool d'adresses IP et effectue toutes les opérations spécifiques au contexte. **sshd** prend en charge la connexion sécurisée à l'interface de ligne de commande StarOS.

Étape 6. Redémarrage de l'instance 1 **vpnmgr**. permet de rétablir la connexion SSH avec un impact minimal dans certains cas. Cependant, la connexion peut se terminer au bout d'un certain temps.

Étape 7. La commutation MIO résout le problème. Veuillez noter que dans les scénarios où un processus peut atteindre une valeur de seuil ou un état de surcharge, le renvoi MIO peut aider à l'effacer.

La solution de contournement en place est la commutation MIO. La section suivante décrit les étapes de l'analyse des causes premières.

Analyse des causes premières

1. Utilisez la commande **show administrateurs** afin de déterminer le nombre de connexions actives sur le noeud. Cependant, le résultat peut ne pas présenter un nombre excessif de sessions actives qui auraient pu bloquer les connexions au noeud.

Exemple de sortie :

```
[local]ASR5500-2# show administrators
Monday September 06 13:15:07 CDT 2021
Administrator/Operator Name    M Type    TTY            Start Time               Mode
Idle

------------------------------ - ------- ------------- ---------------------- ------------ --
--

admin                            admin   /dev/pts/4     Mon Sep 06 13:14:38 2021 Context User 29

admin                            admin   /dev/pts/3     Mon Sep 06 12:21:13 2021 Context User
749

admin                            admin   /dev/pts/2     Thu Sep 02 11:03:57 2021 Context User
342206
[local]ASR5500-2#
```

2. En outre, exécutez ces commandes et analysez le problème. Accédez au shell de débogage via le mode masqué.

```
cli test-command pass <password>
debug shell
```

Exécutez ces commandes dans le shell de débogage :

```
ps -ef
setvr 1 bash
netstat -n
```

ps - processus de liste. La commande **ps** vous permet d'afficher des informations techniques sur les processus en cours sur un système et de vérifier leur état.

-e - affiche tous les processus, quel que soit l'utilisateur.

-f - afficher les processus en format détaillé.

La commande **netstat** est l'une des options de ligne de commande les plus pratiques qui est utilisée pour afficher toutes les connexions de socket présentes sur le noeud. Il possède la capacité d'énumérer toutes les connexions de socket tcp et udp, ainsi que les connexions unix. Cette interface de ligne de commande peut également être utilisée pour répertorier les connecteurs d'écoute qui peuvent encore attendre l'établissement d'une connexion.

Exemple de sortie :

```
ASR5500-2:card5-cpu0# ps -eF

UID        PID  PPID  C    SZ   RSS PSR STIME TTY          TIME CMD

root         1     0  0   511   640   4 Aug20 ?        00:00:13 init [5]

root         2     0  0     0     0   2 Aug20 ?        00:00:00 [kthreadd]

root         3     2  0     0     0   0 Aug20 ?        00:00:00 [ksoftirqd/0]

root         6     2  0     0     0   0 Aug20 ?        00:00:00 [migration/0]

root         7     2  0     0     0   0 Aug20 ?        00:00:01 [watchdog/0]

root         8     2  0     0     0   1 Aug20 ?        00:00:00 [migration/1]

root        10     2  0     0     0   1 Aug20 ?        00:00:00 [ksoftirqd/1]

root        11     2  0     0     0   0 Aug20 ?        00:00:31 [kworker/0:1]

root        12     2  0     0     0   1 Aug20 ?        00:00:00 [watchdog/1]

root        13     2  0     0     0   2 Aug20 ?        00:00:00 [migration/2]

root        15     2  0     0     0   2 Aug20 ?        00:00:00 [ksoftirqd/2]

root        16     2  0     0     0   2 Aug20 ?        00:00:00 [watchdog/2]

root        17     2  0     0     0   3 Aug20 ?        00:00:00 [migration/3]

root        19     2  0     0     0   3 Aug20 ?        00:00:00 [ksoftirqd/3]

root        20     2  0     0     0   3 Aug20 ?        00:00:00 [watchdog/3]

root        21     2  0     0     0   4 Aug20 ?        00:00:00 [migration/4]

root        22     2  0     0     0   4 Aug20 ?        00:00:00 [kworker/4:0]

root        23     2  0     0     0   4 Aug20 ?        00:00:00 [ksoftirqd/4]
```

……

```
ASR5500-2:card5-cpu0# setvr 1 bash
bash-2.05b# netstat -n

Active Internet connections (w/o servers)

Proto Recv-Q Send-Q Local Address           Foreign Address         State

tcp        0      0 10.201.211.23:22        10.227.230.222:51781    ESTABLISHED

tcp        0      0 10.201.211.23:22        10.24.28.55:49918       ESTABLISHED

tcp        0      0 10.201.211.23:22        10.99.10.148:54915      ESTABLISHED

tcp        0      0 10.201.211.23:22        10.227.230.222:51783    ESTABLISHED

Active UNIX domain sockets (w/o servers)

Proto RefCnt Flags       Type       State         I-Node Path

unix  2      [ ]         DGRAM                    39221385

unix  2      [ ]         DGRAM                     27056

bash-2.05b# exit
```

Selon le rapport mentionné précédemment, les serveurs exécutaient des scripts qui généraient des connexions à la zone ASR55K. Ces serveurs ont ouvert beaucoup de ces connexions qui étaient soit bloquées, soit inactives, mais qui n'ont jamais été fermées.

Même après la fin de la connexion TeleTypeWriter (TTY), la connexion TCP est restée active sur nos passerelles.

En conséquence de ces connexions, l'ASR5500 a atteint le nombre maximal de connexions SSH autorisées, obstruant la connexion au boîtier. Dès que vous essayez de vous connecter aux serveurs et de supprimer les processus parents, toutes les connexions sont immédiatement libérées et le SSH est immédiatement restauré.

Ces connexions SSH inactives sont établies en tant que connexions TeleTypeWriter (noTTY). Ces connexions noTTY sont utilisées par les programmes qui sont connectés de telle manière que leur sortie ne s'affiche pas.

Dans la plupart des cas, des commandes telles que SSH admin@asr55k hostname « display version » établissent une connexion noTTY.

De même, les instructions sous forme de SSH : *@notty indique qu'il existe des connexions SSH à nos passerelles (GW) qui n'ont pas été affectées à un terminal visuel, tel qu'un shell ou un pseudo-terminal. Cela peut se produire lors de diverses opérations liées aux scripts, en particulier lors de l'utilisation de connexions FTP/Secure Copy (SCP).

Solution proposée

1. Implémentez un délai d'attente sur les scripts qui peuvent être utilisés pour les serveurs API. Plusieurs connexions SSH qui exécutent plusieurs CLI peuvent générer un encombrement des messageries et une utilisation importante du CPU sur tous les processus sessmgr.

2. Afin de faciliter le dépannage, configurez cette option :

```
logging filter runtime facility cli level debug critical-info
```

3. Appliquez cette configuration au noeud. Cette commande permet de terminer les sessions SSH inactives après 5 minutes. Ceci est utilisé comme mécanisme de protection contre les sessions obsolètes causées par le serveur :

```
Exec > Global Configuration > Context Configuration
configure > context context_name
administrator encrypted password timeout-min-absolute 300 timeout-min-idle 300
```

Informations connexes

- Informations CLI
- Guides de configuration de la gamme Cisco ASR 5000
- **Support et documentation techniques - Cisco Systems**