Risoluzione dei problemi di accesso ad ASR5500 a causa di sessioni TTY inattive

Sommario

Introduzione

In questo documento viene descritto come risolvere i problemi relativi alla perdita della connettività SSH (Secure Shell) sugli IP di gestione del router di servizi di aggregazione (ASR5500/ASR 5000).

Problemi di login ai nodi ASR550

Impossibile accedere ai nodi principali del pacchetto ASR5500. La connessione SSH viene terminata immediatamente senza la richiesta di accesso. Le connessioni Telnet hanno un comportamento simile.

Procedure per la risoluzione dei problemi

Passaggio 1. Tentare di accedere al nodo tramite la connessione alla console.

Passaggio 2. Nella maggior parte dei casi, non vengono emesse trap SNMP (Simple Network Management Protocol) specifiche che potrebbero indicare la causa dell'errore di connessione.

Passaggio 3. I log relativi all'accesso, presenti costantemente nei syslog, sono:

```
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
```

Passaggio 4. Il comando **show crash list all** visualizza gli arresti anomali recenti. Tenere presente che quelli relativi a **vpnmgr** sono particolarmente importanti.

Passaggio 5. Il comando **show task resources all** assicura che i processi **vpnmgr** e **sshd** non siano in stato eccessivo. **vpnmgr** è responsabile della gestione del pool di indirizzi IP ed esegue tutte le operazioni specifiche del contesto. **sshd** supporta l'accesso sicuro alla CLI di StarOS.

Passaggio 6. Il riavvio dell'istanza 1 di **vpnmgr** consente di ripristinare la connessione SSH con un impatto minimo in alcuni casi. Tuttavia, la connessione potrebbe terminare dopo qualche istante.

Passaggio 7. Lo switchover MIO risolve il problema. Si noti che negli scenari in cui un processo potrebbe raggiungere un valore di soglia o uno stato di sovraccarico, il rimbalzo dell'MIO può essere utile per cancellarlo.

La soluzione implementata è lo switchover MIO. Nella sezione successiva vengono illustrati i passaggi per l'analisi della causa principale.

Analisi della causa principale

1. Per determinare il numero di connessioni attive sul nodo, usare il comando **show administrators**. Tuttavia, l'output potrebbe non mostrare un numero eccessivo di sessioni attive che potrebbero aver bloccato le connessioni al nodo.

Output di esempio:

```
[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. Inoltre, eseguire questi comandi e analizzare il problema. Passare alla shell di debug attraverso la modalità nascosta.

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

Eseguire questi comandi nella shell di debug:

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

ps - processi di elenco. Il comando **ps** consente di visualizzare informazioni tecniche sui processi

correnti di un sistema e di verificarne lo stato.

-e - visualizza tutti i processi, indipendentemente dall'utente.

-f - visualizza i processi in formato dettagliato.

Il comando **netstat** è una delle opzioni della riga di comando più pratiche utilizzate per visualizzare tutte le connessioni socket presenti al nodo. Possiede la capacità di elencare tutte le connessioni socket tcp e udp, nonché le connessioni unix. Questa CLI può anche essere usata per elencare i possibili socket di ascolto che potrebbero ancora attendere la connessione.

Output di esempio:

```
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
```

In base al rapporto menzionato in precedenza, i server eseguivano script che generavano connessioni alla confezione di ASR55K. Questi server hanno aperto molte di queste connessioni che si trovavano in uno stato di blocco o inattività, ma non sono mai state chiuse.

Anche dopo l'interruzione della connessione TTY (TeleTypeWriter), la connessione TCP è rimasta attiva sui nostri gateway.

Grazie a queste connessioni, l'ASR5500 ha raggiunto il numero massimo di connessioni SSH consentite, ostacolando la connessione alla scatola. Non appena si cerca di accedere ai server e di terminare i processi padre, tutte le connessioni vengono rilasciate immediatamente e il protocollo SSH viene ripristinato immediatamente.

Queste connessioni SSH inattive vengono stabilite come nessuna connessione TeleTypeWriter (noTTY). Tali connessioni noTTY vengono utilizzate da programmi connessi in modo tale che il relativo output non venga visualizzato.

Comandi come SSH admin@asr55k hostname "display version" stabiliscono una connessione noTTY nella maggior parte dei casi.

Analogamente, le affermazioni di SSH: *@notty indica che sono presenti accessi SSH ai nostri gateway (GW) a cui non è stato assegnato un terminale visivo, come una shell o uno pseudo-terminale. Questa situazione può verificarsi durante una serie di operazioni relative agli script, in particolare quando si utilizzano connessioni FTP/Secure Copy (SCP).

Soluzione proposta

1. Implementare un timeout sugli script che possono essere utilizzati per i server API. Connessioni SSH multiple che eseguono più CLI possono generare una congestione di messenger e un utilizzo significativo della CPU su tutti i processi di sessmgr.
2. Per semplificare la risoluzione dei problemi, configurare questa opzione:

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

3. Applica la configurazione al nodo. Questo comando è usato per terminare le sessioni SSH inattive dopo 5 minuti. Viene utilizzato come meccanismo di protezione contro le sessioni non aggiornate causate dal server:

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

Informazioni correlate

- Informazioni CLI
- Guide alla configurazione di Cisco ASR serie 5000
- **Documentazione e supporto tecnico – Cisco Systems**