Solucionar problemas de login no ASR5500 devido a sessões de ociosidade no TTY

Contents

Introduction

Este documento descreve como solucionar problemas de cenários quando a conectividade do Secure Shell (SSH) é perdida para os IPs de gerenciamento do Roteador de Serviços de Agregação (ASR5500/ASR 5000).

Problemas de login nos nós ASR5500

Você não pode fazer login nos nós do núcleo do pacote ASR5500. A conexão SSH é terminada imediatamente sem o prompt de login. As conexões Telnet exibem comportamento semelhante.

Etapas para solucionar problemas

Etapa 1. Tente fazer login no nó através da conexão do console.

Etapa 2. Na maioria dos casos, não são emitidas armadilhas específicas do Protocolo de Gerenciamento de Rede Simples (SNMP - Simple Network Management Protocol) que possam apontar para a causa da falha de conexão.

Etapa 3. Os registros relacionados ao login, constantemente presentes nos syslogs são:

```
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
```

Etapa 4. O comando **show crash list all** exibe os travamentos recentes, observe que os relacionados ao **vpnmgr** são especialmente importantes.

Etapa 5. O comando **show task resources todos** garantem que os processos **vpnmgr** e **sshd** não devem estar em estado exagerado. **o vpnmgr** é responsável pelo gerenciamento do pool de endereços IP e executa todas as operações específicas do contexto. **sshd** suporta login seguro na CLI do StarOS.

Etapa 6. A reinicialização da instância 1 **do vpnmgr**. ajuda a recuperar a conexão SSH com impacto mínimo em alguns casos. No entanto, a conexão pode terminar depois de um tempo.

Passo 7. O switchover MIO resolve o problema. Observe que em cenários em que um processo pode alcançar um valor de limite ou um estado de sobrecarga, a devolução MIO pode ajudar a limpá-lo.

A solução alternativa no local é o switchover de MIO. A próxima seção fala sobre as etapas para a análise da causa raiz.

Análise da causa raiz

1. Use o comando **show administrator** para determinar o número de conexões ativas no nó. No entanto, a saída pode não exibir um número excessivo de sessões ativas que podem ter bloqueado as conexões ao nó.

Saída de exemplo:

```
[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. Além disso, execute esses comandos e examine o problema. Navegue até o shell de depuração através do modo oculto.

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

Execute estes comandos no shell de depuração:

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

ps - listar processos. O comando **ps** permite visualizar informações técnicas sobre os processos atuais em um sistema, bem como verificar seu status.

-e - mostrar todos os processos, independentemente do usuário.

-f - mostrar processos em formato detalhado.

O comando **netstat** é uma das opções de linha de comando mais convenientes usadas para exibir todas as conexões de soquete presentes no nó. Ele possui a capacidade de listar todas as conexões de soquete tcp e udp, bem como as conexões unix. Essa CLI também pode ser usada para listar os possíveis soquetes de escuta que ainda podem esperar o estabelecimento de uma conexão.

Saída de exemplo:

```
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
```

De acordo com o relatório mencionado anteriormente, os servidores executaram scripts que geraram conexões com a caixa ASR55K. Esses servidores abriram muitas dessas conexões que estavam em um estado de bloqueio ou ocioso, mas nunca foram fechadas.

Mesmo depois que a conexão TTY (TeleTypeWriter) foi encerrada, a conexão TCP permaneceu ativa em nossos gateways.

Como resultado dessas conexões, o ASR5500 atingiu o número máximo de conexões SSH permitidas, obstruindo a conexão com a caixa. Assim que você tentar fazer login nos servidores e eliminar os processos pai, todas as conexões serão liberadas instantaneamente e o SSH será restaurado imediatamente.

Essas conexões SSH ociosas são estabelecidas como nenhuma conexão TeleTypeWriter (noTTY). Essas conexões noTTY são usadas por programas conectados de tal forma que a saída não seja exibida.

Comandos como SSH admin@asr55k hostname "display version" estabelece uma conexão noTTY na maioria dos casos.

Da mesma forma, instruções como SSH: *@notty indica que há logins SSH para nossos Gateways (GWs) que não receberam um terminal visual, como shell ou pseudo-terminal. Isso pode ocorrer durante uma variedade de operações relacionadas a scripts, especialmente quando se usa conexões FTP/Secure Copy (SCP).

Solução proposta

1. Implemente um tempo limite nos scripts que podem ser usados para os servidores de API. Várias conexões SSH que executam várias CLIs podem gerar congestionamento de mensagens e uso significativo da CPU em todos os processos do sessmgr.
2. Para facilitar a solução de problemas, configure esta opção:

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

3. Aplique essa configuração ao nó. Esse comando é usado para terminar sessões SSH ociosas após 5 minutos. Isso é usado como um mecanismo de proteção contra sessões obsoletas causadas pelo servidor:

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

Informações Relacionadas

- informação CLI
- Guias de configuração do Cisco ASR 5000 Series
- **Suporte Técnico e Documentação - Cisco Systems**