Comprensione e risoluzione dei problemi - TTL Gatekeeper e processo di aging out

Sommario

Introduzione

Questo documento descrive come Cisco Gatekeeper analizza gli endpoint utilizzando il valore Time to Live (TTL) in diversi casi. I debug e i comandi **show** vengono usati per mostrare come funziona il TTL in diversi modi.

Prerequisiti

Requisiti

I lettori di questo documento devono essere a conoscenza dei seguenti argomenti:

- Implementazione di Cisco H.323, con Cisco Gatekeeper incluso. Per una descrizione di base dei gatekeeper H.323, fare riferimento alla sezione Descrizione dei gatekeeper H.323.

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware.

- Ai fini del presente documento, per raccogliere le informazioni viene usato il software Cisco IOS® versione 12.3(9).Accertarsi di utilizzare Cisco IOS con la funzionalità Gatekeeper H.323. Il nome dell'immagine Cisco IOS è contrassegnato da una **x**. Ad esempio, un Cisco IOS valido per Cisco 3640 in modo che agisca come gatekeeper è c3640-**ix**-mz.123-9.bin.
- Cisco Gatekeeper (tutte le piattaforme).**Nota:** NetMeeting è stato utilizzato come endpoint H.323 nell'esempio di questo documento perché non fornisce un valore TTL.Per informazioni su come configurare NetMeeting con i gateway Cisco IOS, consultare il documento sulla configurazione di Microsoft NetMeeting con i gateway Cisco IOS.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Convenzioni

Per ulteriori informazioni sulle convenzioni usate, consultare il documento Cisco sulle convenzioni nei suggerimenti tecnici.

Suggerimenti su Time to Live

Di seguito vengono forniti alcuni suggerimenti sul funzionamento del comando TTL su un Gatekeeper Cisco e sul funzionamento del processo di aging per gli endpoint in diversi casi.

- Il Gatekeeper Cisco si aspetta di ricevere periodicamente informazioni dall'endpoint tramite i messaggi di richiesta di registrazione (RQ) (leggeri o completi).
- Per i timeout dell'endpoint, il Gatekeeper Cisco presta attenzione al valore TTL fornito dall'endpoint nel messaggio RQ. Per il timeout dell'endpoint, è presente un'impostazione predefinita hardcoded di 1800 secondi (trenta minuti). Questo valore può essere modificato con il comando **endpoint ttl** ***<time_value> della*** Cisco Gatekeeper CLI. In questo modo viene modificato il comportamento di tutti gli endpoint H.323 v1 o H.323 v2 e degli endpoint successivi che non includono il valore TTL nel messaggio di richiesta di offerta.
- Cisco Gatekeeper esegue periodicamente un "processo di aging dell'endpoint". Questo processo varia in base al carico della CPU corrente da ogni minuto a ogni cinque minuti. Per ogni venti percento di utilizzo della CPU, l'intervallo di aging aumenta di un minuto, fino a un massimo di cinque minuti. Per non sovraccaricare la CPU quando sono presenti molti endpoint, il processo di misurazione durata viene eseguito solo fino a cinquanta endpoint per passaggio. Se ce ne sono altri, vengono rinviati al successivo pop del timer. Può durare da uno a cinque minuti.
- Se il campo timeToLive è incluso nel messaggio RRQ Registration, Admission and Status (RAS), il gatekeeper utilizza il valore di tale campo per sostituire il valore predefinito del sistema o il valore configurato utilizzando il comando **endpoint ttl** ***<time_value>*** gatekeeper CLI. Una volta trascorso tale periodo di tempo senza che l'endpoint sia in grado di ascoltare, il successivo pop-up del timer viene sottoposto al processo di pulizia dell'endpoint. Il caso peggiore è il valore TTL inviato dall'endpoint, più cinque minuti (se il Gatekeeper Cisco è costantemente sottoposto a un carico elevato della CPU). Uno scenario peggiore è quello del timeout TTL più un minuto.
- Quando l'endpoint non include il campo timeToLive nel messaggio RQ, Cisco Gatekeeper tratta l'endpoint come se non supportasse il valore TTL. In questo caso, quando il gatekeeper non riceve più le richieste di preventivo dall'endpoint, calcola il timeout TTL (il valore predefinito è 1800 secondi o il valore specificato nel comando **endpoint ttl**). Quindi invia tre richieste di informazioni (IRQ) con intervalli di tempo che variano da uno a cinque minuti ciascuna (in base al carico della CPU del gatekeeper). Dopo aver inviato tre IRQ e non aver ricevuto risposta, Cisco Gatekeeper rimuove infine l'endpoint.
- Se l'endpoint è in una chiamata attiva, non viene sottoposto a aging fino al termine della chiamata.
- Il Gatekeeper Cisco si aspetta di ricevere informazioni dagli endpoint coinvolti in una chiamata

con il messaggio di risposta alle informazioni (IRR). Se il Gatekeeper Cisco non riceve messaggi IRR periodici contenenti riferimenti al "guid" per una chiamata, attende quattro minuti, quindi invia un IRQ agli endpoint a cui la chiamata è destinata. Se, dopo altri otto minuti, il Gatekeeper Cisco non ha ancora sentito nulla riguardo a tale chiamata, questa viene pulita e il gatekeeper invia le richieste di disinnesto (DRQ) agli endpoint. Sono trascorsi circa dodici minuti prima che una chiamata "penzolante" venga cancellata (e che la larghezza di banda venga liberata). Il timer di chiamata non è configurabile.
- Gli endpoint di proprietà di un Gatekeeper Cisco alternativo non vengono sottoposti a aging direttamente (in quanto il gatekeeper non è il proprietario effettivo dell'endpoint).
- Gli endpoint statici (creati con l'alias di comando di configurazione static **xxxxx**) non sono obsoleti.

Debug Cisco Gatekeeper

Di seguito sono riportati alcuni dei comandi **show** e **debug** che è possibile utilizzare per verificare il funzionamento del TTL:

- mostra endpoint gatekeeper
- debug ras
- debug h225 asn1

In queste due sezioni vengono illustrati due casi in cui Cisco Gatekeeper analizza gli endpoint utilizzando valori TTL diversi.

Caso 1

Questo output viene generato dai comandi debug ras e debug h225 asn1 e viene generato da un Gatekeeper Cisco. Nel debug, il valore TTL del gateway è 60 secondi. Il Gatekeeper Cisco conferma e accetta questa condizione nel messaggio RCF (Registration Confirmation), a prescindere dal valore TTL dell'endpoint predefinito o configurato. Infatti l'endpoint include un valore TTL.

```
Mar  2 23:52:50.797: RAS INCOMING ENCODE BUFFER::= 0E 400FD206 0008914A
00030000 0100AC10 0D2AE26A 00040067 006B0062 0
02D0032 00B50000 12128F00 02003B01 80211E00 36003100 36004600 32004400
43004300 30003000 30003000 30003000 30003101 00
0180
Mar  2 23:52:50.797:
Mar  2 23:52:50.797: RAS INCOMING PDU ::=

value RasMessage ::= registrationRequest :
    {
      requestSeqNum 4051
      protocolIdentifier { 0 0 8 2250 0 3 }
      discoveryComplete FALSE
      callSignalAddress
      {
      }
      rasAddress
      {
        ipAddress :
        {
          ip 'AC100D2A'H
```

```
          port 57962
        }
      }
      terminalType
      {
        mc FALSE
        undefinedNode FALSE
      }
      gatekeeperIdentifier {"gkb-2"}
      endpointVendor
      {
        vendor
        {
          t35CountryCode 181
          t35Extension 0
          manufacturerCode 18
        }
      }
      timeToLive 60
!--- TTL value. keepAlive TRUE endpointIdentifier {"616F2DCC00000001"} willSupplyUUIEs FALSE
maintainConnection TRUE } Mar 2 23:52:50.805: RRQ (seq# 4051) rcvd
Mar  2 23:52:50.805: RAS OUTGOING PDU ::=

value RasMessage ::= registrationConfirm :
    {
      requestSeqNum 4051
      protocolIdentifier { 0 0 8 2250 0 3 }
      callSignalAddress
      {
      }
      gatekeeperIdentifier {"gkb-2"}
      endpointIdentifier {"616F2DCC00000001"}
      alternateGatekeeper
      {

        {
          rasAddress ipAddress :
          {
            ip 'AC100D29'H
            port 1719
          }
          gatekeeperIdentifier {"gkb-1"}
          needToRegister TRUE
          priority 0
        }
      }
      timeToLive 60
      willRespondToIRR FALSE
      maintainConnection TRUE
    }



Mar  2 23:52:50.813: RAS OUTGOING ENCODE BUFFER::= 12 400FD206 0008914A
00030008 0067006B 0062002D 00321E00 36003100 3
6004600 32004400 43004300 30003000 30003000 30003000 3000310F 8A140140
AC100D29 06B70800 67006B00 62002D00 31800200 3B
010001 80
Mar 2 23:52:50.813:
Mar 2 23:52:50.817: IPSOCK_RAS_sendto: msg length 86 from
                    172.16.13.16:1719 to 172.16.13.42: 57962
Mar 2 23:52:50.817: RASLib::RASSendRCF: RCF (seq# 4051) sent to 172.16.13.42
```

Caso 2

In questo esempio, un endpoint che non ha inviato un valore TTL nel messaggio RQ viene avvisato di inviare una RQ lightweight prima di 120 secondi, ossia il valore configurato sul gatekeeper. Come si può notare in questo output, il Gatekeeper Cisco non elimina l'endpoint finché non vengono ricevuti tre messaggi IRQ senza risposta, anche se è stato ricevuto un messaggio URQ (Unregistration Request). Le ore comprese tra l'IRQ sono comprese tra uno e cinque minuti.

```
gka-1#show logging
Syslog logging: enabled (0 messages dropped, 0 messages rate-limited, 0 flushes, 0 overruns)
    Console logging: disabled
    Monitor logging: level debugging, 1076 messages logged
    Buffer logging: level debugging, 4257 messages logged
    Logging Exception size (4096 bytes)
    Trap logging: level informational, 60 message lines logged

Log Buffer (9999999 bytes):

Mar 14 06:28:31.771: RAS INCOMING ENCODE BUFFER::= 0C 80000006 0008914A
00020001 00AB4555 BF06B801 00AB4555 BF05C502 00014007 006B0065 00740070
00610074 0065006C 60B50053 4C164D69 63726F73 6F6674AE 204E6574 4D656574
696E67AE 0003332E 3000
Mar 14 06:28:31.783:
Mar 14 06:28:31.787: RAS INCOMING PDU ::=

value RasMessage ::= registrationRequest :
    {
      requestSeqNum 1
      protocolIdentifier { 0 0 8 2250 0 2 }
      discoveryComplete FALSE
      callSignalAddress
      {
        ipAddress :
        {
          ip 'AB4555BF'H
          port 1720
        }
      }
      rasAddress
      {
        ipAddress :
        {
          ip 'AB4555BF'H
          port 1477
        }
      }
      terminalType
      {
        terminal
        {
        }
        mc FALSE
        undefinedNode FALSE
      }
      terminalAlias
      {
        h323-ID : {"ketpatel"}
      }
      endpointVendor
      {
        vendor
        {
```

```
          t35CountryCode 181
          t35Extension 0
          manufacturerCode 21324
        }
        productId '4D6963726F736F6674AE204E65744D656574696E...'H
        versionId '332E3000'H
      }
    }



Mar 14 06:28:31.811: RAS OUTGOING PDU ::=

value RasMessage ::= registrationConfirm :
    {
      requestSeqNum 1
      protocolIdentifier { 0 0 8 2250 0 3 }
      callSignalAddress
      {
      }
      terminalAlias
      {
        h323-ID : {"ketpatel"}
      }
      gatekeeperIdentifier {"gka-1"}
      endpointIdentifier {"81F6A89800000001"}
      alternateGatekeeper
      {
      }
      timeToLive 120
      willRespondToIRR FALSE
      maintainConnection FALSE
    }



Mar 14 06:28:31.823: RAS OUTGOING ENCODE BUFFER::= 12 C0000006 0008914A
00030001 4007006B 00650074 00700061 00740065 006C0800 67006B00 61002D00
311E0038 00310046 00360041 00380039 00380030 00300030 00300030 00300030
00310F8A 01000200 77010001 00


gka-1#show gatekeeper endpoints
                   GATEKEEPER ENDPOINT REGISTRATION
                   ================================
CallSignalAddr  Port  RASSignalAddr   Port  Zone Name         Type    Flags
--------------- ----- --------------- ----- ---------         ----    -----
171.69.85.191   1720  171.69.85.191   1477  gka-1             TERM
    H323-ID: ketpatel
Total number of active registrations = 1


Mar 14 06:28:31.835:
Mar 14 06:28:31.835: RAS OUTGOING PDU ::=

value RasMessage ::= infoRequest :
    {
      requestSeqNum 70
      callReferenceValue 0
      callIdentifier
      {
        guid '00000000000000000000000000000000'H
      }
    }
```

```
Mar 14 06:28:31.839: RAS OUTGOING ENCODE BUFFER::= 56 00004500 000B0011
00000000 00000000 00000000 00000000 00
Mar 14 06:28:31.843:
Mar 14 06:28:31.847: RAS INCOMING ENCODE BUFFER::= 58 80004502 03C00038
00310046 00360041 00380039 00380030 00300030 00300030 00300030 003100AB
4555BF05 C50100AB 4555BF06 B8024007 006B0065 00740070 00610074 0065006C
4007006B 00650074 00700061 00740065 006C
Mar 14 06:28:31.859:
Mar 14 06:28:31.859: RAS INCOMING PDU ::=

value RasMessage ::= infoRequestResponse :
    {
      requestSeqNum 70
      endpointType
      {
        terminal
        {
        }
        mc FALSE
        undefinedNode FALSE
      }
      endpointIdentifier {"81F6A89800000001"}
      rasAddress ipAddress :
      {
        ip 'AB4555BF'H
        port 1477
      }
      callSignalAddress
      {
        ipAddress :
        {
          ip 'AB4555BF'H
          port 1720
        }
      }
      endpointAlias
      {
        h323-ID : {"ketpatel"},
        h323-ID : {"ketpatel"}
      }
    }



Mar 14 06:30:42.208: RAS OUTGOING PDU ::=

value RasMessage ::= infoRequest :
    {
      requestSeqNum 71
      callReferenceValue 0
      callIdentifier
      {
        guid '00000000000000000000000000000000'H
      }
    }



Mar 14 06:30:42.212: RAS OUTGOING ENCODE BUFFER::= 56 00004600 000B0011
00000000 00000000 00000000 00000000 00
Mar 14 06:30:42.216:
```

```
Mar 14 06:30:42.216: RAS INCOMING ENCODE BUFFER::= 58 80004602 03C00038
00310046 00360041 00380039 00380030 00300030 00300030 00300030 003100AB
4555BF05 C50100AB 4555BF06 B8024007 006B0065 00740070 00610074 0065006C
4007006B 00650074 00700061 00740065 006C
Mar 14 06:30:42.228:
Mar 14 06:30:42.232: RAS INCOMING PDU ::=

value RasMessage ::= infoRequestResponse :
    {
      requestSeqNum 71
      endpointType
      {
        terminal
        {
        }
        mc FALSE
        undefinedNode FALSE
      }
      endpointIdentifier {"81F6A89800000001"}
      rasAddress ipAddress :
      {
        ip 'AB4555BF'H
        port 1477
      }
      callSignalAddress
      {
        ipAddress :
        {
          ip 'AB4555BF'H
          port 1720
        }
      }
      endpointAlias
      {
        h323-ID : {"ketpatel"},
        h323-ID : {"ketpatel"}
      }
    }



Mar 14 06:32:05.938: RAS INCOMING ENCODE BUFFER::= 19 40000101 00AB4555
BF06B802 4007006B 00650074 00700061 00740065 006C4007 006B0065 00740070
00610074 0065006C 1E003800 31004600 36004100 38003900 38003000 30003000
30003000 30003000 31
Mar 14 06:32:05.950:
Mar 14 06:32:05.950: RAS INCOMING PDU ::=

value RasMessage ::= unregistrationRequest :
    {
      requestSeqNum 2
      callSignalAddress
      {
        ipAddress :
        {
          ip 'AB4555BF'H
          port 1720
        }
      }
      endpointAlias
      {
        h323-ID : {"ketpatel"},
        h323-ID : {"ketpatel"}
      }
```

```
      endpointIdentifier {"81F6A89800000001"}
    }



Mar 14 06:32:05.962: RAS OUTGOING PDU ::=

value RasMessage ::= unregistrationConfirm :
    {
      requestSeqNum 2
    }



Mar 14 06:32:05.962: RAS OUTGOING ENCODE BUFFER::= 1C 0001
Mar 14 06:32:05.966:


gka-1#show gatekeeper endpoints
                   GATEKEEPER ENDPOINT REGISTRATION
                   ================================
CallSignalAddr  Port  RASSignalAddr   Port  Zone Name         Type    Flags
--------------- ----- --------------- ----- ---------         ----    -----
171.69.85.191   1720  171.69.85.191   1477  gka-1             TERM
Total number of active registrations = 1

Mar 14 06:33:42.223: RAS OUTGOING PDU ::=

value RasMessage ::= infoRequest :
    {
      requestSeqNum 72
      callReferenceValue 0
      callIdentifier
      {
        guid '00000000000000000000000000000000'H
      }
    }



Mar 14 06:33:42.227: RAS OUTGOING ENCODE BUFFER::= 56 00004700 000B0011
00000000 00000000 00000000 00000000 00
Mar 14 06:33:42.231:
Mar 14 06:34:42.234: RAS OUTGOING PDU ::=

value RasMessage ::= infoRequest :
    {
      requestSeqNum 73
      callReferenceValue 0
      callIdentifier
      {
        guid '00000000000000000000000000000000'H
      }
    }



Mar 14 06:34:42.238: RAS OUTGOING ENCODE BUFFER::= 56 00004800 000B0011
00000000 00000000 00000000 00000000 00
Mar 14 06:34:42.242:
Mar 14 06:35:42.244: RAS OUTGOING PDU ::=

value RasMessage ::= infoRequest :
    {
```

```
      requestSeqNum 74
      callReferenceValue 0
      callIdentifier
      {
        guid '00000000000000000000000000000000'H
      }
    }



Mar 14 06:35:42.248: RAS OUTGOING ENCODE BUFFER::= 56 00004900 000B0011
00000000 00000000 00000000 00000000 00
Mar 14 06:35:42.252:

gka-1#
gka-1#show gatekeeper endpoints
                    GATEKEEPER ENDPOINT REGISTRATION
                    ================================
CallSignalAddr  Port  RASSignalAddr   Port  Zone Name         Type    Flags
--------------- ----- --------------- ----- ---------         ----    -----
Total number of active registrations = 0
```

Informazioni correlate

- Cisco High-Performance Gatekeeper
- Supporto H.323 versione 2
- Risoluzione dei problemi di registrazione di Gatekeeper
- Supporto alla tecnologia vocale
- Supporto ai prodotti voce e Unified Communications
- Risoluzione dei problemi di Cisco IP Telephony
- Supporto tecnico – Cisco Systems