

排除AAA限制功能故障

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[背景資訊](#)

[工作機制](#)

[AAAMGR隊列](#)

[限制](#)

[相關思科支援社群討論](#)

簡介

本檔案介紹AAA限制(RADIUS)記錄功能，此功能支援限制傳送到RADIUS伺服器的存取 (驗證和授權) 和記帳記錄。

此功能允許使用者配置適當的限制速率，以避免當頻寬不足以容納從Cisco路由器到RADIUS伺服器生成的突發記錄時，出現網路擁塞和不穩定性。

必要條件

需求

本文件沒有特定需求。

採用元件

本文檔中的資訊基於ASR5k平台。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除 (預設) 的組態來啟動。如果您的網路正在作用，請確保您已瞭解任何指令可能造成的影響。

背景資訊

當aamgr以較高速率將radius訊息傳送到RADIUS伺服器時 (例如當大量作業階段同時關閉時，系統會同時產生所有作業階段的記帳停止訊息)，RADIUS伺服器可能無法以如此高速率接收訊息。為了處理這種情況，我們需要在aamgr處建立有效的速率控制機制，以便aamgr以最佳速率傳送消息，從而使RADIUS伺服器能夠接收所有消息，並確保沒有消息由於RADIUS伺服器的過載而被丟棄。

工作機制

當aaamgr以配置的速率向RADIUS伺服器傳送消息時，它會均勻傳送消息而不是以單個突發傳送所有消息。根據配置，每秒被分成幾個相等的時隙（每個時隙具有特定時段）。插槽的最小時間週期可能是50毫秒。

必須在考慮這些因素的情況下配置費率

- 傳入呼叫的速率，
- aaamgr例項數
- RADIUS伺服器可以接收訊息和
- 臨時間隔（用於記帳配置）
- 用於伺服器選擇的演算法

如果為身份驗證伺服器配置的值太低，則會出現瓶頸，導致擁塞，可能導致由於會話設定超時而導致呼叫被丟棄。如果為記帳伺服器配置了較小的值，則由於隊列溢位，將發現大量清除記帳消息。

當配置該功能時，計算第二的時隙數和第二的時段的時隙數，並將其儲存在半徑級別。當準備好向RADIUS伺服器傳送消息時，將檢查是否達到配額（此時隙的消息數）。如果未達到該限制，則傳送消息；如果是，則消息在伺服器級隊列中排隊，以在將來的時隙中傳送。每個RADIUS伺服器都保留有關在當前時間槽中傳送的訊息數量以及該時間槽到期時間的詳細資訊。當從伺服器級隊列中挑選排隊消息時，這些消息將被置於例項級隊列的頭部，確保較舊的消息優先於任何其他新消息。從例項級隊列中選取消息以進行維護。

AAAMGR隊列

AAAMGR中有兩種型別的消息隊列：

1. 例項級隊列
2. 伺服器級隊列

生成消息時，消息最初會在例項級隊列中排隊進行維護。

每50毫秒處理例項級隊列25毫秒。從例項級隊列中出隊的任何消息都將嘗試傳送到RADIUS伺服器。在某些情況下，我們可能無法傳送消息（沒有可用的頻寬或沒有可用的ID）。在這種情況下，嘗試失敗的消息將在伺服器級隊列中排隊。每50毫秒，您會選擇多少封具有ID可用、頻寬可用的郵件，並將其放在例項級別隊列的頭部（這些郵件比例項級別隊列中的任何其它郵件都舊）。

當存在對記帳消息的速率控制時，並且如果例項級別隊列中有大量記帳消息，則任何新的身份驗證消息都將進入例項級別隊列的末尾。要進行處理，必須等待所有記帳消息（在新身份驗證消息之前）傳送到RADIUS伺服器或移動到伺服器級隊列。這是現有行為，不會對其進行修改。因此，這可能會導致處理新身份驗證消息的小延遲。

範例

根據值為5的max-rate，您可以在1秒內向Radius身份驗證伺服器傳送五條消息，並且每個警報有256條未應答的radius身份驗證消息（預設最大未應答配置）。如果消息超過5條，在1秒內消息將排入隊列，直到AAA伺服器響應現有的請求。

如果從一台伺服器向伺服器傳送了256條Radius身份驗證消息，則剩餘請求將排入隊列，直到AAA伺服器響應現有請求。它將再次進入與max-rate相同的隊列。只有當您有空閒插槽時，才會從隊列中取出消息。當您收到消息響應或消息超時時，會進入空閒插槽。

限制

由於Cisco ASR5K是一個分散式系統，具有處理呼叫的獨立sessmgr/aaamgr對，因此速率限制只能針對獨立aamgr例項實施。理論上，只需將例項總數乘以最大速率，即可將單個例項的速率擴展到整個Cisco ASR5K盒每個例項的。

此數字只是晴天場景中的絕對上限。您不能將Cisco ASR5K視為黑匣子，也不能假定在系統中所見的計算值未超過上限時所有呼叫都應成功。

Radius max-rate與與系統相關的其他內部和外部引數關聯。如果其中一個條件未滿足，請檢視預期影響。

狀況

從demuxmgr到所有會話的呼叫的統一分配

IMSI的均勻分佈 (這只是一種情況)
輪詢調解記帳)

不會突然接到電話進來

Radius伺服器應及時回應

如果不滿足以下條件的影響

如果呼叫分佈不均勻，則radius消息可以

對某些例項進行排隊。因此，即使沒有達到理論上的最大速率限制，消息排隊的例項也會丟棄呼叫。

中介記帳輪詢基於基於IMSI的路由。

在這種情況下，根據IMSI分佈，根據路由邏輯，某些伺服器組可能比其它伺服器組更優先，針對這些伺服器建立隊列，導致消息丟棄。

如果有大量新呼叫，則新生成的radius消息將再次在系統中排隊。處理新的radius請求時。會話設定時間可能已過期，導致呼叫中斷。

當radius請求由於伺服器問題而超時時，將再次出現隊列建立，因為除非從系統中刪除當前期望響應的請求，否則將不會處理新請求。從系統中刪除超時消息的速率也取決於最大未處理消息和超時配置。

在許多情況下，我們可以看到並非所有活動的aamgr任務都處理訪問請求。這意味著sessmgr任務中的呼叫分配不均，而且並非所有aaamgr例項都參與呼叫處理。

呼叫分配不基於嚴格的循環排程機制，即如果有10個來電，它們將以單調演算法轉到10個會話。

呼叫分配基於這四個主要因素

- active_session_count
- cpu_load
- Round_trip_delay(demuxmgr - sessmgr - demuxmgr)
- outstanding_add_request (demux到sessmgr)

這是當前的實現。最大速率只是一個上限，但由於我們架構的分散式性質，您無法直接將其外推到機箱負載。此行為取決於給定AAAmgr上的負載在特定的時間。

應使用RADIUS max-rate queue監控系統的狀態。如果存在隊列建立，則這表示不符合以下4個條件之一 (請參閱表)，您必須確定導致此問題的根本原因。

**max-rate queue threshold可以實施和持續監控。