

Problemen oplossen met AAA draaien

Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Achtergrondinformatie](#)

[Werkingsmechanisme](#)

[AAMGR-wachtrijen](#)

[Beperkingen](#)

[Gerelateerde Cisco Support Community-discussies](#)

Inleiding

In dit document wordt de Throttling of AAA (RADIUS)-registratiefunctie beschreven, die ondersteuning biedt voor het bundelen van toegang (authenticatie en autorisatie) en boekhoudgegevens die naar de RADIUS-server worden verzonden.

Deze functie stelt een gebruiker in staat om de juiste stempo te configureren om netwerkcongestie en instabiliteit te voorkomen wanneer er onvoldoende bandbreedte is om een plotselinge burst van records te ontvangen die van de Cisco-router naar de RADIUS-server zijn gegenereerd.

Voorwaarden

Vereisten

Er zijn geen specifieke vereisten van toepassing op dit document.

Gebruikte componenten

De informatie in dit document is gebaseerd op ASR5k-platform.

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u de potentiële impact van elke opdracht begrijpen.

Achtergrondinformatie

Wanneer een RADIUS-bericht met een hoog tempo naar een RADIUS-server wordt verzonden (bv. wanneer een groot aantal sessies tegelijkertijd afneemt, worden boekhoudingsstop-berichten voor alle sessies tegelijkertijd gegenereerd), kan de RADIUS-server de berichten niet tegen zo hoge prijzen ontvangen. Om deze conditie aan te kunnen hebben hebben we een effectief snelheidscontrole mechanisme nodig bij Aamgr, zodat Aamgr berichten met een optimaal tempo

verstuurt zodat de RADIUS-server alle berichten kan ontvangen en ervoor zorgt dat geen bericht komt te vallen als gevolg van overbelasting op de RADIUS-server.

Werkingsmechanisme

Wanneer amgr berichten met de ingestelde snelheid naar de RADIUS-server stuurt, worden er gelijkmatig berichten verstuurd elke seconde in plaats van alle berichten te verzenden in één keer. Afhankelijk van de configuratie, wordt elke seconde verdeeld in verschillende gelijke tijdsleuven (met een specifieke tijdsperiode per sleuf). De minimumperiode van een sleuf kan 50 milliseconden zijn.

Bij het instellen van het tarief moet rekening worden gehouden met deze

- Het aantal inkomende oproepen,
- Aantal aaamgr-gevallen
- Het tempo waarmee de RADIUS-server de berichten en
- Interims-interfaces (configuratie van de boekhouding)
- Algoritme voor serverselectie

Als de ingestelde waarde voor de authenticatieservers te laag is, zal er een flessenhals ontstaan die leidt tot

congestie, wat kan leiden tot oproepen die vallen vanwege de tijdelijke installatie van de sessie. Als een lage waarde is ingesteld voor boekhoudservers, zal een hoop wegzuivering van boekhoudkundige berichten worden waargenomen als gevolg van overflow in de wachtrij.

Wanneer de functie is ingesteld, wordt het aantal tijdsleuven in een tweede periode en de tijdsperiode van een seconde berekend en opgeslagen op het niveau van de straal. Wanneer een bericht klaar is om naar een RADIUS-server te worden verstuurd, wordt gecontroleerd of het quotum (aantal berichten voor deze tijdsleuf) is bereikt. Als de limiet niet wordt bereikt, wordt het bericht verstuurd, als het zo is, dan wordt het bericht in de wachtrij voor serverniveau geplaatst om in toekomstige tijdsleuven te worden verstuurd. Elke RADIUS-server bevat informatie over het aantal berichten dat in de huidige tijdsleuf wordt verzonden en het tijdstip waarop de tijdsleuf vervalst. Wanneer de in de wachtrij staande berichten uit de wachtrij op serverniveau worden geselecteerd, worden ze in het hoofd van de wachtrij op het niveau van de instantie geplaatst, waarbij de voorkeur wordt gegeven aan oudere berichten dan een ander nieuw bericht. Berichten uit de wachtrij voor een instantie worden geselecteerd voor onderhoud.

AAMGR-wachtrijen

Er zijn twee soorten wachtrijen bij AAMGR voor berichten:

1. Wachtrijen voor instanties
2. Wachtrijen voor serverniveau

Wanneer een bericht wordt gegenereerd, wordt het aanvankelijk in de wachtrij van het installatieniveau geplaatst voor onderhoud.

De preek van het installatieniveau wordt voor elke 50 milliseconden 25 milliseconden verwerkt. Elk bericht dat wordt verwijderd uit de wachtrij voor een voorbeeldniveau wordt naar de RADIUS-

server verzonden. Onder bepaalde omstandigheden kunnen we de berichten niet verzenden (geen beschikbare bandbreedte-width of geen beschikbare ID's). In dergelijke gevallen worden de berichten die de poging mislukt, in de wachtrijen op het serverniveau geplaatst. Voor elke 50 milliseconde kiest u evenveel berichten die ID's beschikbaar hebben en ook bandbreedte-width beschikbaar zijn, en zet ze in het hoofd van de wachtrij voor instantie-niveau (deze berichten zijn ouder dan elk ander bericht dat in de wachtrij voor instantie-niveau aanwezig is).

Wanneer er een snelheidscontrole is voor boekhoudkundige berichten, en als er veel boekhoudkundige berichten zijn in de wachtrij op het niveau van de instantie, dan gaat elke nieuwe authenticatieboodschap naar de staart van de wachtrij op het niveau van de instantie. Om verwerkt te worden moet het wachten tot al het boekhoudingsbericht (vóór het nieuwe autebericht) naar een RADIUS-server wordt verzonden of naar een server-level-wachtrij wordt verplaatst. Het is een bestaand gedrag en het wordt niet gewijzigd. Het kan dus een kleine vertraging veroorzaken voor het verwerken van het nieuwe geluidsbericht.

Voorbeeld

Gebaseerd op max-rate met waarde 5, kunt u vijf berichten in 1 seconde verzenden en 256 onopgeloste authenticatieberichten (standaard max-uitstaande configuratie) per aaamgr naar Radius authenticatieserver hebben. Indien er meer dan 5 berichten zijn, worden de berichten in de wachtrij geplaatst totdat de AAA-server op de bestaande verzoeken reageert.

Indien u de 256 Radius-authenticatieberichten haalt die van een aapag naar een server worden verstuurd, worden de resterende verzoeken in de wachtrij geplaatst totdat de AAA-server reageert op de bestaande verzoeken. Het zal opnieuw in dezelfde rij gaan als de max-ratio. Het bericht wordt alleen uit de wachtrij gehaald wanneer u een vrije sleuf hebt. De vrije sleuf komt binnen wanneer u een antwoord voor het bericht ontvangt of wanneer het uitkomt.

Beperkingen

Aangezien Cisco ASR5K een gedistribueerd systeem is met onafhankelijke sessmgr/aaamgr-paren die de oproepen verwerken, kan de snelheidscontrole alleen voor onafhankelijke aamaanvoorstellingen worden geïmplementeerd. Het is theoretisch om het tempo van één exemplaar uit te breiden naar het gehele Cisco ASR5K-vakje door het totale aantal gevallen alleen te vermenigvuldigen met het max-tarief van elke instantie.

Dit aantal is slechts de absolute bovengrens bij een zonnig dagscenario. U kunt Cisco ASR5K niet als een zwarte doos behandelen en u kunt niet veronderstellen dat alle oproepen zouden moeten slagen als de berekende waarde die in het systeem is gezien de bovenste limiet niet overschrijdt.

De maximale snelheid van de straal is gekoppeld aan andere interne en externe parameters die betrekking hebben op het systeem. Zie het verwachte effect als niet aan een van de voorwaarden is voldaan.

Voorwaarden

Eenvoudige verdeling van oproepen vanaf demuxmgr naar alle sessmgrs

Uniforme verdeling van IMSI's (dit is alleen

Gevolgen indien niet met

Als de calldistributie niet uniform is, kunnen de Straal-berichten in de wachtrij staan voor bepaalde gevallen. Dus zelfs al v de theoretische max-rate limiet niet bereikt, zal de aanroep worden teruggebracht voor gevallen waar de berichten in de wachtrij staan.

Bemiddeling-accounting-ronde-robin is gebaseerd op op l

voor het geval dat
van de "round robin"-
bemiddelingsboekhouding)

Geen plotselinge uitbarstingen van oproepen
die binnenkomen

Radius-servers moeten op tijd reageren

gebaseerde routing.

In dit geval kan, op basis van de IMSI-distributie, een aantal servers de voorkeur genieten boven andere, op basis van routinglogica, dan kan de wachtrij voor die servers worden aangelegd, hetgeen tot een call-down leidt.

Als er een breuk van nieuwe vraag is, dan zullen opnieuw nieuw gegenereerde Straalberichten in de wachtrij van het systeem worden geplaatst. Tegen de tijd dat de nieuwe straalverzoeken worden verwerkt. De setup-tijd van de sessie kan zijn verlopen en kan worden teruggebeld.

Wanneer straal om tijden-out vraagt vanwege serverprobleem zal er opnieuw een rijopbouw zijn, omdat nieuwe verzoeken niet zullen worden verzonden tenzij de huidige die een antwoord verwacht van het systeem wordt verwijderd. Het tempo waarmee de getimed berichten uit het systeem zullen worden verwijderd is ook afhankelijk van de maximaal uitstaande en de time-out configuraties.

In veel gevallen kunnen we zien dat verzoeken om toegang niet door alle actieve taken worden verwerkt. Dat betekent dat we ongelijke gespreksverdeling hebben binnen de sessgmh-taken en verder, niet alle andere belangrijke instanties zijn betrokken bij de gespreksverwerking.

De distributie van gesprekken is niet gebaseerd op het strikte round robin mechanisme dat is als er 10 inkomende oproepen zijn, dan zullen ze naar 10 sessmgrs gaan in een monotonisch algoritme.

De verdeling van de oproepen is gebaseerd op deze vier belangrijke factoren

- **active_sessie_count**
- **cpu_load**
- **round_trip_delar** (demuxmgr - sessmgr - demuxmgr)
- **uitstaande_add_request** (demux naar sessmgr)

Dit is de huidige implementatie. De max-snelheid is slechts een bovengrens, maar vanwege de gedistribueerde aard van onze architectuur kunt u deze niet rechtstreeks extrapoleren naar de lading van het chassis. Het gedrag hangt af van de lading op een gegeven AAAMh op een bepaald moment.

De rij van de hoogste snelheden van de straal zou moeten worden gebruikt om de **status** van het systeem **te controleren**. Als er een **wachtrij** is **aangelegd**, dan het betekent dat aan een van deze 4 voorwaarden (zie de tabel) niet is voldaan en u moet de oorzaak van hetzelfde vaststellen.

**De maximale wachtrijdrempel kan worden geïmplementeerd en constant worden gecontroleerd.