Abstract

This document describes how in PE-based VPNs a PE of a VPN can use Radius [1-2] to authenticate its CEs and discover the other PEs of the VPN.

1. Introduction

This document describes how in PE-based VPNs a PE of a VPN can use Radius [1-2] to authenticate its CEs and discover the other PEs of the VPN. In Radius terms, the CEs are users and PEs are Network Access Servers (NAS) implementing Radius client function.

A VPN can span multiple Autonomous Systems (AS) and multiple providers. Each PE, however, only needs to be a Radius client to Radius of the "local" provider. In case of a CE belongs to a "foreign" VPN, Radius of the local provider act as a proxy client to Radius of the foreign provider.
A companion document will describe how Radius can be used by the CEs of a CE-based VPN to discover the other CEs of the VPN.

2. Site Identification

Each CE (a VPN site) is identified by a "user name" of the form

[realm/]user@domain

"realm", if present, denotes a provider that is the administrative owner of the VPN. It is needed only if a CE connects to a VPN at a "foreign" PE. It is then used by the local Radius of the PE to proxy requests to the correct foreign Radius.

"user" identifies a site in a VPN identified by "domain". As an example,

providerX/atlanta@vpnY.domainZ.net

could identify a CE called "atlanta" in a VPN called "vpnY.domainZ.net", which belongs to an administratively owner providerX.

3. Radius Configuration

Each realm has a single Radius that stores all information regarding VPNs that belong to the realm. For reliable operation of this protocol, each Radius should consist of more than one physical Radius server. For correct operation of this protocol, all these physical servers MUST at all times share the same database content.

For each VPN, Radius of the realm to which the VPN belongs to MUST be configured with a set of "users" that correspond to the CEs of the VPN. User information includes user name, password, and VPN identifier:

<CE user name, password, VPN identifier>

User information MAY also include a list of PEs, to which the CE is allowed to connect to, and QoS information regarding the CE’s connection to the VPN.

In addition to the above static information, Radius keeps dynamically track of the PEs of the VPN as described below (Protocol Operation). The PEs MAY also have pre-configured attributes telling, for example, that a PE is a hub of a VPN.

If dynamic PE discovery capability of this protocol is not used,
Radius MUST be configured for each VPN with a list of its PEs. Such a degenerate use of this protocol does is not discussed further in this memo.

In order to allow local PEs and Radius servers in foreign realms to make CE related queries to Radius of the local realm, potential local PEs and Radius servers of foreign realms MUST be configured in local Radius as clients.

4. PE Configuration

Each PE MUST be configured with the information about the Radius servers of local Radius to which to send requests to. For reliability reasons, each PE SHOULD have available more than one physical Radius server.

5. Protocol Operation

Radius keeps track of the PEs and CEs of a VPN in a database table that has the following fields:

<VPN identifier, PE IP address, CE user name>

In addition, Radius records for each active PE the time when it has last received from the PE any Radius request:

<PE IP address, timestamp>

This information is used by Radius to detect if a PE has failed for a longer period of time or has been taken improperly out of use, and if so, to clean up its database.

When a CE wants to join a VPN, the PE to which the CE is attached to issues a Radius Access-Request using the user name and password of the CE. The PE has either learned this information from the CE via an authentication protocol, for example, 802.1x/EAP, or it has been configured in the PE.

If authentication succeeds, Radius responds with Access-Accept that tells to the PE the VPN to which the CE belongs to and possibly other CE specific information, e.g., QoS parameters. If authentication fails or if some other preconditions are not met, e.g. the CE is not allowed to connect to the particular PE, Radius responds with Access-Reject.

After receiving an Access-Accept response, the PE issues a Start Accounting-Request. When Radius receives the request, it inserts <VPN identifier, PE IP address, CE user name> record in its database.
replacing a possible earlier <VPN identifier, *, CE user name> record. Then it responds with an Accounting-Response, which includes an attribute that contains all unique PE IP addresses in the set <VPN identifier, *, *>. The PE considers the CE as connected to the VPN after it has received the Accounting-Response.

Whenever N minutes has elapsed from the last Radius request that the PE has sent for any CE in a realm, it issues an Interim-Update Accounting-Request for any one of its CEs that belong to the realm. In order to keep this keep-alive mechanism as light weight as possible, Radius’ response to the Interim-Update Accounting-Request does not contain any VPN related attributes.

If a PE wants for some reason to get from Radius an up-to-date list of PEs in a particular VPN, it can at any time issue a new Start Accounting request for any one of its CEs that belongs to the VPN. Similarly, the PE can at any time issue a new Access-Request for any one of its CEs.

If Radius doesn’t receive from a PE any request during a period of M * N minutes, Radius considers that the PE is not anymore operational and removes from its database all <*, PE IP address, *> records and the <PE IP address, *> record.

When a CE is to be disconnected from the VPN, the PE issues a Stop Accounting-Request. After receiving the request, Radius removes the <VPN identifier, PE IP address, CE user name> record from its database and responds with an Accounting-Response that does not contain any VPN related attributes. The PE considers the CE as disconnected from the VPN when it has received the Accounting-Response.

6. PE Failure Recovery

If a PE fails and the failure has lasted longer than M * N minutes, Radius has removed all <*, PE IP address, *> records from its database. No matter how long the failure has lasted, upon recovery, the PE re-authenticates all CEs connected to it in all VPNs and thus re-discovers all other PEs in all those VPNs.

7. Scaling Limits

Since Radius protocol operates over UDP, the maximum UDP payload size available for Radius attributes is limited to about 1500 - 40 = 1480 octets assuming that UDP fragmentation is not supported. The most space consuming message is Accounting-Response to Start Accounting-Request, which contains a list of IP addresses of the PEs of a VPN. This limits the number of PEs in a VPN to about 370, which is large
enough for a fully meshed VPN.

Larger VPNs can be easily supported by configuring some of the PEs as hubs, since only the hubs of a VPN need to be advertised in the Accounting-Response. This provides a way to increase the maximum number of PEs in a VPN to thousands.

Besides the packet size, another factor limiting scalability of this protocol might be the keep-alive mechanism implemented by Interim-Update Accounting-Request. However, since a PE needs to send only one keep-alive message per N minutes per realm in which it has at least one VPN site, scalability of keep-alives it is not an issue.

8. Security Considerations

Security of Radius based VPN discovery depends on the security of Radius, that is covered in [1] and [2].

9. Further Work

Protocol details of Radius based VPN discovery will be specified in a future version of this memo, provided that there is enough IETF interest in Radius based discovery.

Acknowledgements

I would like to thank Mark Duffy and Joel Halpern for their constructive comments on earlier versions of this memo.

References


Author’s Address

Juha Heinanen
Song Networks, Inc.
Hallituskatu 16
33200 Tampere, Finland
Email: jh@song.fi

Full Copyright

Copyright (C) The Internet Society (2000). All Rights Reserved.