PDMv2 Protocol and Registration

N. Elkins, M. Ackermann, A. Deshpande, T. Pecorella

draft-ietf-ippm-encrypted-pdmv2-10

draft-elkins-pdmv2-registration-00

PDM Implementation

Diagnosing IPv6 in Nanoseconds: RFC 8250
https://portal.aiori.in/rfc8250-ietf122/

"As modern networks scale, the boundary between application performance and network behavior grows blurry. Network performance issues are notoriously hard to pinpoint. A web page loads slowly, but is it the DNS resolution, the server, the network?"

PDM Implementation

Diagnosing IPv6 in Nanoseconds: RFC 8250
https://portal.aiori.in/rfc8250-ietf122/

"As modern networks scale, the boundary between application performance and network behavior grows blurry. Network performance issues are notoriously hard to pinpoint. A web page loads slowly, but is it the DNS resolution, the server, the network? RFC 8250, however, proposes something far more strategic, i.e. embedding timing diagnostics directly inside IPv6 packets."

PDMv2 Principles

The design of PDMv2 adheres to a set of foundational principles:

1. Offline Decryption: All decryption of data occurs offline, eliminating the computational overhead of real-time decryption on network devices.

2. Speed of Handshake Processing: The goal of PDMv2 is to have as little time spent in handshake processing as possible.

3. Handshake at IP Layer: The establishment of session keys is at the IP layer not at the transport or session layers. However, keys will be changed when there is a change in the 5-tuple. For ICMP and IPsec, sender-destination IP pair defines the session for key rotation purposes.

PDMv2 Principles

4. Separation of Encryption Layers: Encryption at the extension header level is designed to be independent of encryption in higher-layer protocols (e.g., TLS, QUIC). This avoids bootstrapping problems where key negotiation at one layer (IP) is dependent on information from another layer (TCP / TLS).

5. Key Reuse Avoidance: Keys are not reused between sessions. Each session uses a freshly derived key to enhance security and forward secrecy.

6. Base Key Registration: Master keys and device authentication are established through a registration process with an Authentication Server. A complementary draft will detail the full registration procedure and the operation of the Decryption Server that handles offline decryption.

PDMv2 Principles

7. Sequential Field for Key Derivation: Each packet may include a sequential field (in cleartext), which serves as input to a key derivation function (KDF). This supports dynamic keying mechanisms such as those used in Hybrid Public Key Encryption (HPKE).

8. Sample Key Derivation Implementation: A sample implementation using HPKE will be included to illustrate how these principles can be applied in practice. Alternative KDFs may be used, based on implementation needs.

9. Optional Sequential Field Usage: A field which may be used as a nonce and is sent in the clear will be provided. Usage of this sequential field is optional and can be omitted if not required for the cryptographic scheme in use. It is required for HPKE but the implementor may choose another scheme.

Why is registration needed?

- Registration is done ahead of time to save time in the actual session setup

- We also cannot just re-use TLS keys because PDMv2 is at the IP layer not the TCP. We also handle all other transport protocols (UDP, ICMP, etc)

Participants

- Client,
- Server,
- Analyzer,
- Authentication Server.

To start with …

- A TLS connection is made between the participating entity (Client or Server) and the Authentication Server (AS).
- All traffic will flow over this encrypted connection.
- The AS will present its certificate during the TLS flow. This will authenticate the AS.
- Question: how does the Client know of an AS's IP address?
- AS needs to have a TLS server running and get an X509 certificate.

Commands: From Participant

1. Register Organization
2. RegisterClient
3. RegisterServer
4. RegisterConnection
5. RegisterAnalyzer

More commands needed in the registration draft.

Is anyone doing something like this already?

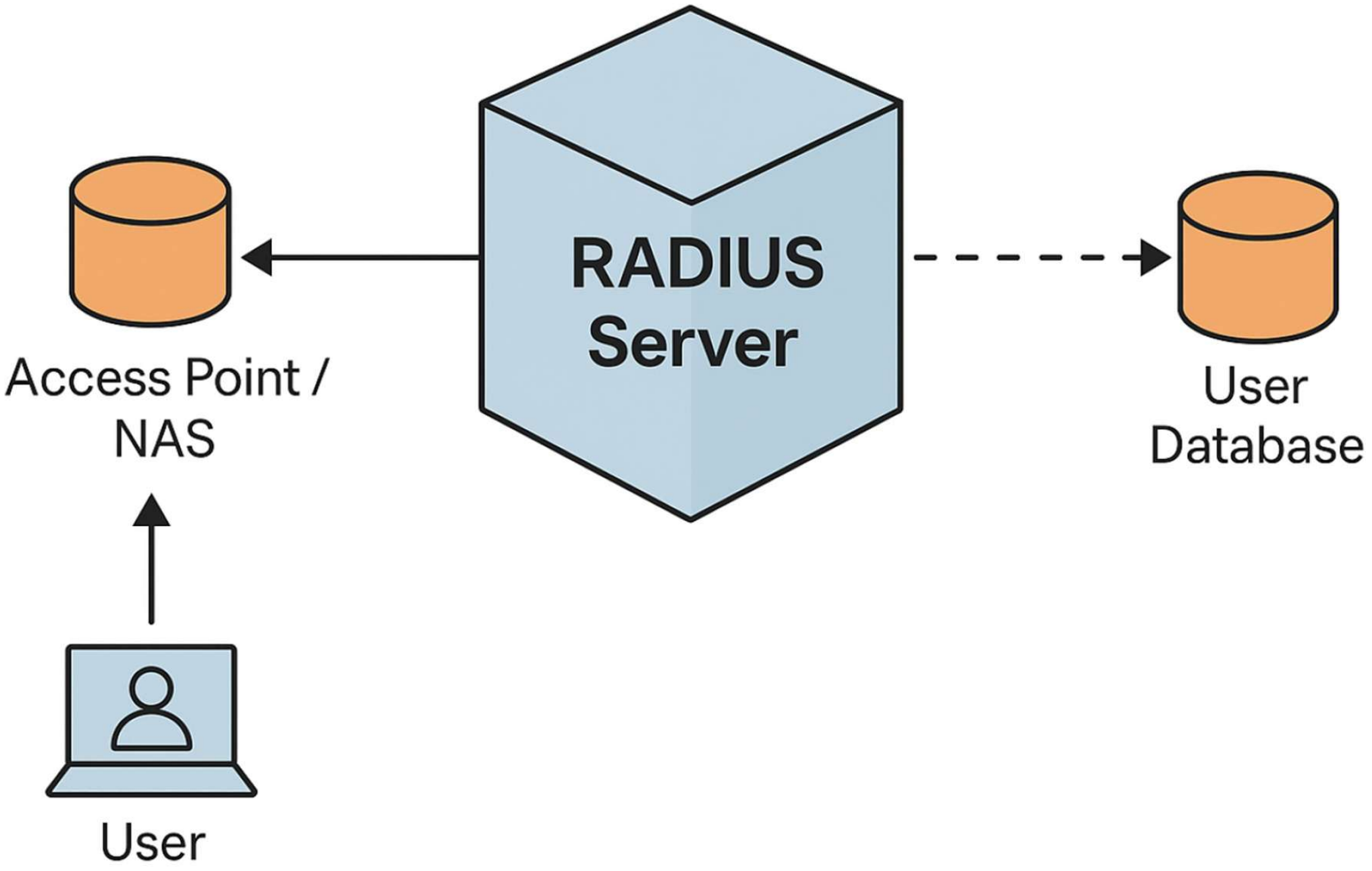
- eduroam (education roaming) is a global Wi-Fi roaming service for users in research and education. It relies on RADIUS (Remote Authentication Dial-In User Service) for authentication, authorization, and accounting across multiple administrative domains.

- How about RADIUS?

What is RADIUS?

RADIUS (Remote Authentication Dial-In User Service) is a client-server protocol that provides centralized Authentication, Authorization, and Accounting (AAA) for users who connect and use a network service. Originally designed for dial-up access, it is now widely used for managing access to Wi-Fi networks, VPNs, and enterprise authentication systems. A RADIUS client (such as a network access server or Wi-Fi controller) forwards user credentials to a RADIUS server, which authenticates them against a backend directory (like LDAP or Active Directory), determines access policies, and logs usage data for accounting purposes. RADIUS operates primarily over UDP and supports features such as user role enforcement, VLAN assignment, and multi-entity federated authentication, as seen in systems like eduroam.

Enterprise RADIUS



Similar Use of RADIUS

- eduroam (education roaming) is a global Wi-Fi roaming service for users in research and education. It relies on RADIUS (Remote Authentication Dial-In User Service) for authentication, authorization, and accounting across multiple administrative domains.

🔐 Overview of eduroam with RADIUS

eduroam enables users from one institution to connect securely to Wi-Fi networks at other participating institutions using their home institution credentials. This requires:

- Cross-institutional authentication
- Hierarchical RADIUS infrastructure
- Federated identity verification

Better to use Diameter?

RFC 6733, titled "Diameter Base Protocol," defines a robust framework for Authentication, Authorization, and Accounting (AAA) services, serving as a successor to the RADIUS protocol. Designed to address the scalability, reliability, and security limitations of its predecessor, Diameter operates over reliable transport protocols like TCP and SCTP, and supports optional security mechanisms such as TLS and IPsec. The protocol facilitates peer-to-peer communication between Diameter nodes, which can function as clients, servers, or agents (e.g., proxies, relays, redirect agents). A key feature is its extensibility through Diameter applications, which introduce specific command codes and Attribute-Value Pairs (AVPs) to cater to various services, including network access, IP mobility, and policy control.

More research

- Looking at SPIFFE and the new Web-Bot authorization draft to see if that is more in line with what we need.

- More to follow.

- Thoughts? Comments?