RADIUS 1.1

ALAN DEKOK, MAY 2023
ALPN

Client

radius/1.1

Server

radius/1.1

Access-Request ...

... Access-Accept
NEGOTIATION

- Each end sends "radius 1.1" to signal capabilities
- Falls back to RFC6614 RADIUS/TLS if not sent or received
- Suggested configuration is a flag: “RADIUS 1.1”
  - forbid
  - allow
  - require
PACKET HEADER

- Code, Length unchanged
- Identifier is now reserved, and is replaced by Token
- Authenticator field is now:
  - 4 octet Token
  - 12 octet reserved
- Token allows for $2^{32}$ packets per connection
ATTRIBUTES

- User-Password etc. are encoded as “text”, protected by TLS.
  - Message-Authenticator is ignored
  - CHAP, MS-CHAP, etc. can still be transported
  - Original-Packet-Code is no longer used or needed
  - No changes to other attribute encoding
  - “Instead of doing weird MD5 stuff, just encode the data as-is”
AUTHENTICATION METHODS

- PAP, CHAP, MS-CHAP, EAP, etc. are entirely unaffected by this

- Even in RADIUS with proxies, the transport does not affect authentication
  - proxies encrypt / decrypt the fields, but the output field data is unchanged

- A site can choose to not use CHAP or MS-CHAP, and gain full FIPS compliance
  - should we recommend that vendors ship a “FIPS” version of their software?
IMPLEMENTATION

- Shipping in FreeRADIUS 3.2.3 for interoperability testing
  - not enabled in the default build
  - ./configure --with-radiusv11
- And then add flag to client / socket / home server definition
- “diff” is ~2,000 lines. Can be done in a very small amount of time, plus tests
CRYPTOGRAPHY AND RADIUS

- RADIUS 1.1 negotiation status must be cached for session resumption
  - TLS does not do this for us
- This document forbids further cryptographic primitives in RADIUS
  - Partly finalizing the research started in RFC 6421.
NEW RADIUS STANDARDS

‣ This document should require no changes to new RADIUS standards

‣ i.e. it defines how to transform any RADIUS into RADIUS 1.1

‣ new standards must use existing attribute data types, obfuscation, etc.

‣ in which case the transformation is already defined

‣ The only place this is not possible is packets like Protocol-Error, which do weird things with Code and Identifier.
QUESTIONS?

- Hopefully that is it