Departamento de Ingeniería de la Información y las Comunicaciones

Mejora del protocolo RADIUS para soportar la fragmentación de datos de autorización

Alejandro Pérez (alex@um.es), Fernando Pereñíguez (pereniguez@um.es),
Rafael Marín (rafa@um.es), Gabriel López (gabilm@um.es), Diego R. López (diego@tid.es)

JITEL 2013, Granada
Martes 29 de Octubre, 2013

Universidad de Murcia
Telefónica I+D

UNIVERSIDAD DE MURCIA

Departamento de Ingeniería de la Información y las Comunicaciones

Tabla de contenidos

1. Introducción

- **RADIUS** (*Remote Access Dial-In User Server*)
  - Uno de los protocolos AAA (*Authentication, Authorization, and Accounting*) más utilizados
  - Principalmente usado para el control de acceso a la red (p.ej. eduroam)
  - Intercambio de datos entre un cliente (p.ej. punto de acceso) y un Servidor
- Los paquetes RADIUS tienen un **límite de 4096 octetos**
- Recientemente ha surgido la necesidad de transportar datos de gran tamaño, como:
  - Sentencias SAML
  - Grandes políticas de filtrado de paquete

1. Introducción

- **Caso de uso:** Se quiere que los usuarios que se conecten a la red *eduroam* obtengan diferentes QoS en función de su rol (profesor o alumno).
  - Organización origen envía una sentencia SAML a la organización visitada con atributos de usuario tras la autenticación
  - Esta sentencia puede tener un tamaño considerable.
- **Propuesta**: definir una solución de fragmentación para RADIUS que permita el intercambio de grandes cantidades de datos de autorización:
  - Sin requerir la modificación de las infraestructuras existentes
  - Sin cambiar el formato de paquetes o atributos

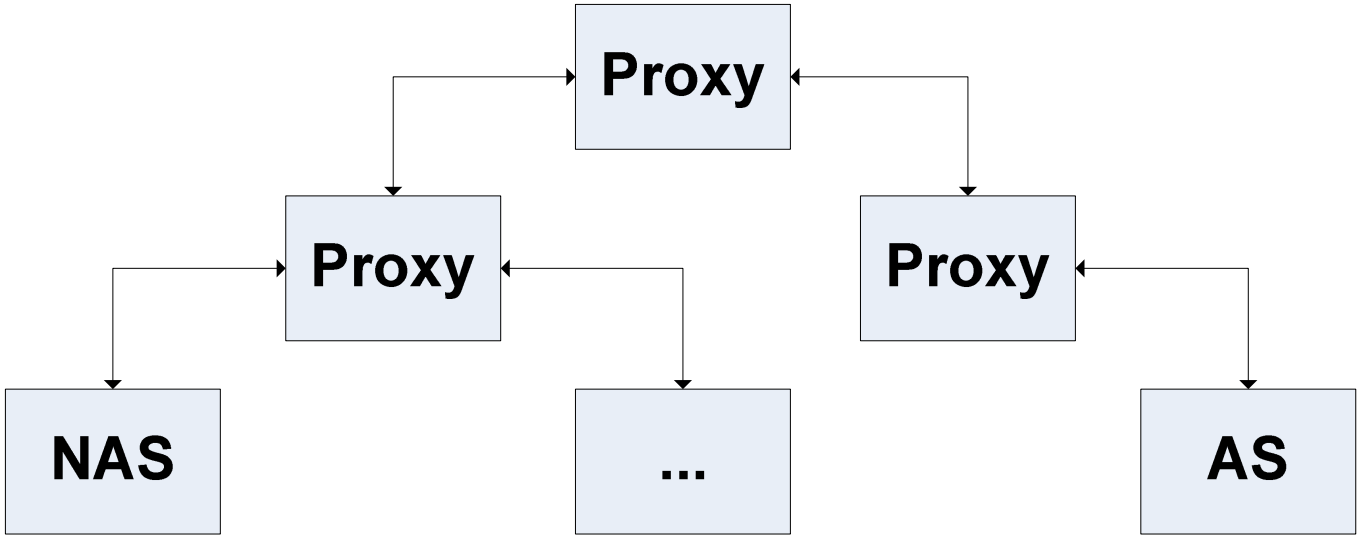
2. RADIUS

- Un paquete RADIUS se compone de
  - Cabecera
  - Atributos
    - Longitud: hasta 255 octetos
    - Los atributos > 255 se representan como la concatenación de varios atributos del mismo tipo

Cabecera	Attr1	Attr2	Attr2	Attr2	Attr3

2. RADIUS

- Comunicación inter-dominio a través de proxies
  - Permite crear federaciones de identidad jerárquicas
    - Un nodo sólo conoce a sus vecinos (proxies) más inmediatos



3. Trabajo relacionado

- RFC 6158 → esquemas para tratar con datos grandes
  1. Secuencia de Access-Request/Access-Challenge
     - Muchos atributos no pueden transportarse en Access-Challenge
  2. Envío de nombres en lugar de valores
     - Los nombres referencian un valor. Por ejemplo:
       - "regla1" → "filtrar los paquetes entrantes al puerto 80"
     - Sólo válido para datos de naturaleza estática
       - SAML es dinámico → cada sentencia es diferente
     - Se pueden pasar URLs apuntando al dato
       - Requiere habilitar servidores Web, distribuir certificados, configurar firewalls…

3. Trabajo relacionado

- O simplemente podemos eliminar la limitación de 4096 octetos
  - Campo de longitud permite hasta 64 KB
  - Se limitó para evitar la fragmentación UDP
    - Se podría usar TCP (RFC 6613)
- Implica un cambio en el protocolo RADIUS
  - Requiere la actualización de todas las entidades intermedias (proxies)

4. Mecanismo de fragmentación

- Permita el intercambio de grandes cantidades de datos
- Centrado en datos de autorización
  - Los datos de autenticación y accounting no tienen este problema
- No requiera la modificación de las infraestructuras existentes
  - Sólo de los nodos implicados

4. Mecanismo de fragmentación

- **Operación para el emisor**
  - Los paquetes grandes se dividen en *chunks*
    - Son paquetes RADIUS estándar
    - Del mismo tipo que el original
    - Transportan subconjunto de los atributos
    - Señalizados mediante dos atributos (excepto el último chunk)
      - Frag-Status = More-Data-Pending
      - Service-Type = Additional-Authorization
- **Operación para el receptor**
  - Pide más chunks mediante un paquete que contiene:
    - Frag-Status = More-Data-Request
    - Service-Type = Additonal-Authorization
  - Una vez recibidos todos los *chunks* se reconstruye el paquete original y se procesa

4. Mecanismo de fragmentación

- Distinguimos tres fases:
  1. Pre-autorización. NAS → AS
     - Ej. SAML-Request
  2. Autenticación. NAS → AS (sin fragmentación)
     - Ej. RADIUS-EAP
  3. Post-autorización. NAS ← AS
     - Ej. SAML-Response

4. Mecanismo de fragmentación

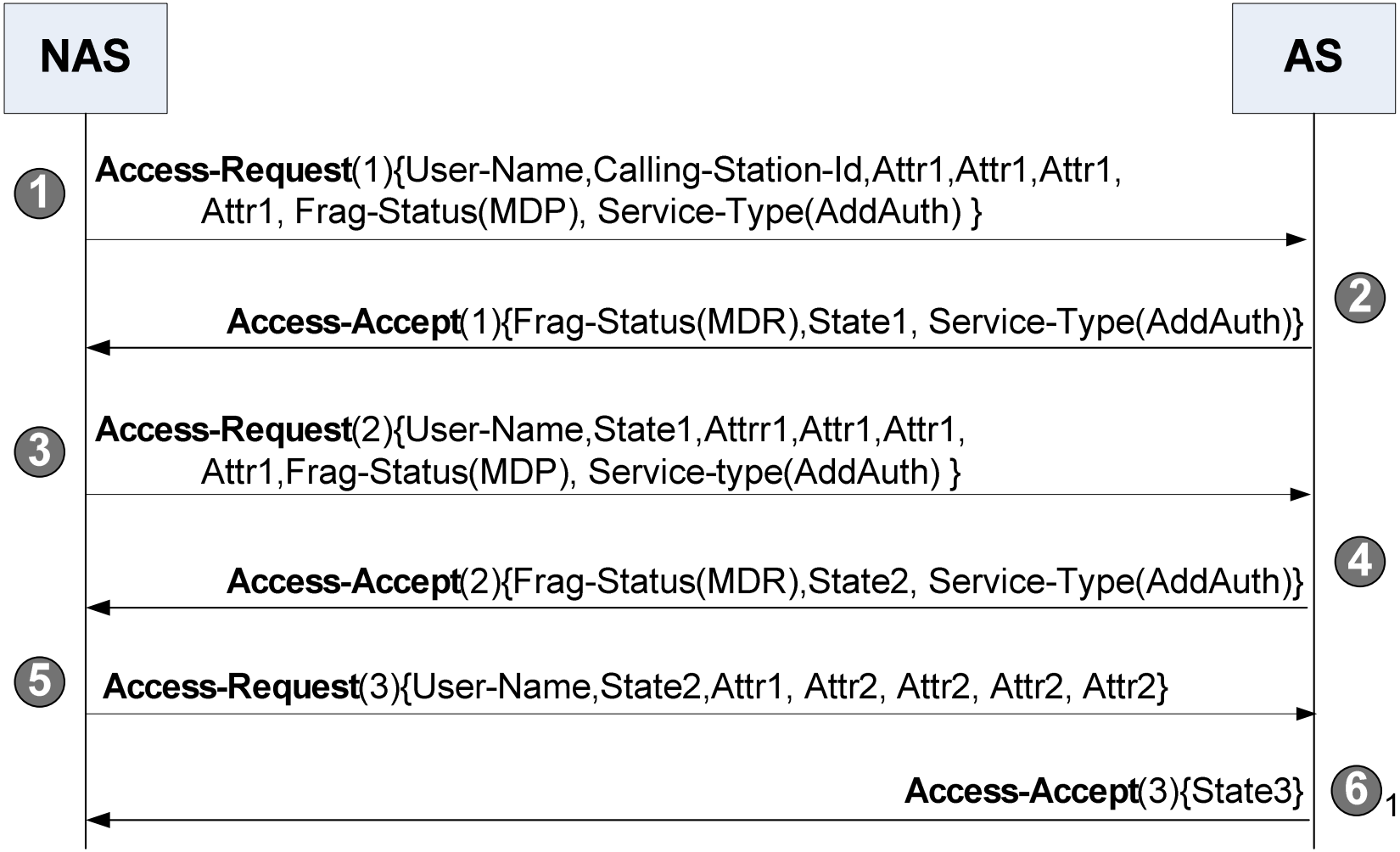
Ejemplo de pre-autorización

- Tamaño máximo de paquete = 8 atributos
- Supongamos que el NAS quiere enviar un *Access-Request* grande al AS (15 atributos):

```
User-Name, Calling-Station-Id, Attr1,
Attr1, Attr1, Attr1, Attr1, Attr1, Attr1,
Attr1, Attr1, Attr2, Attr2, Attr2, Attr2
```

4. Mecanismo de fragmentación

Ejemplo de pre-autorización



4. Mecanismo de fragmentación

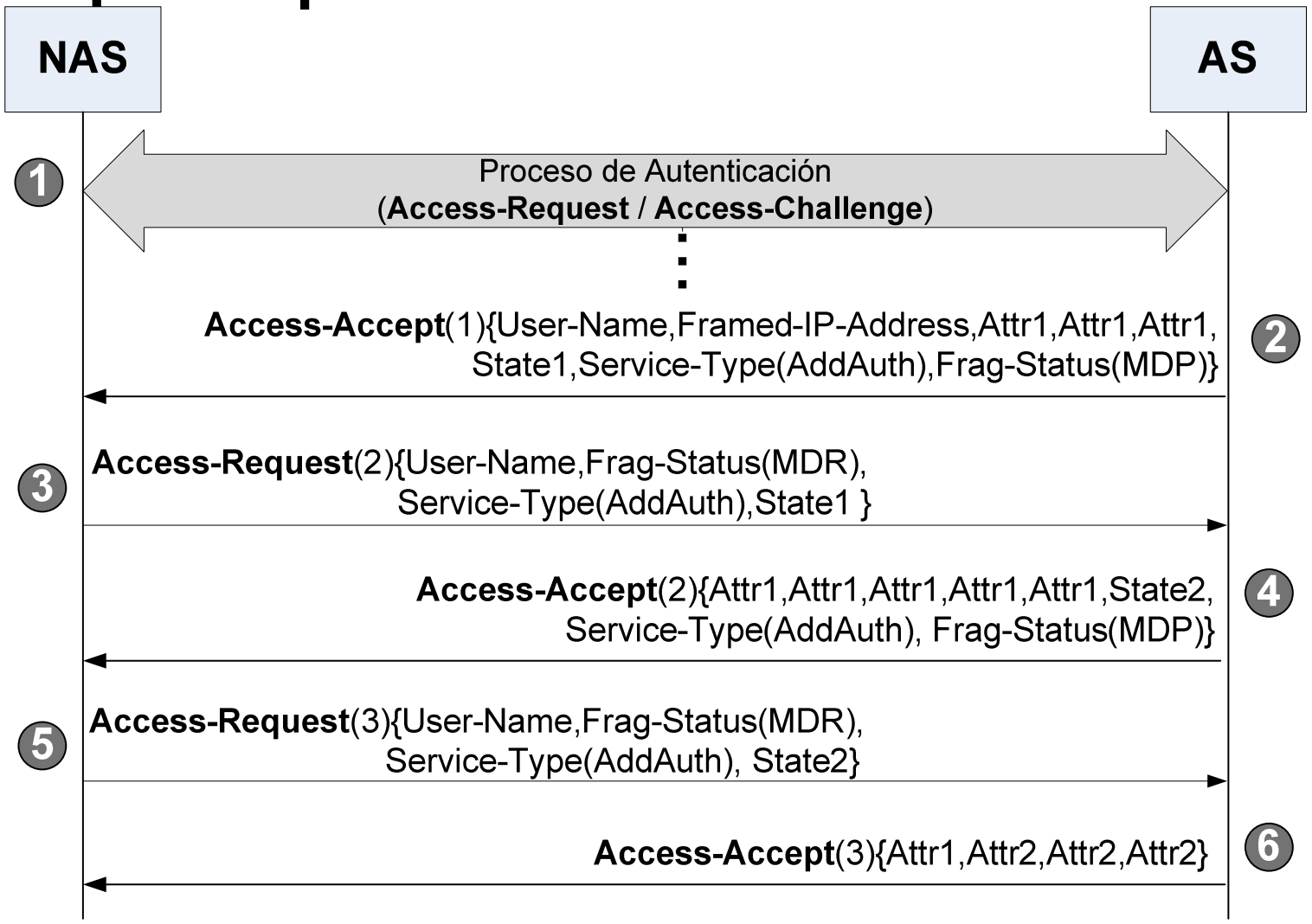
Ejemplo de post-autorización

- Tamaño máximo de paquete = 8 atributos
- Supongamos que el AS quiere enviar un *Access-Accept* grande al NAS (14 atributos)

```
User-Name, Framed-IP-Address, Attr1,
Attr1, Attr1, Attr1, Attr1, Attr1, Attr1,
Attr1, Attr1, Attr2, Attr2, Attr2
```

4. Mecanismo de fragmentación

Ejemplo de post-autorización



5. Discusión

- **Operación con proxies**
  - Legacy
    - Ven los *chunks* como paquetes normales
    - Pueden añadir *Proxy-State,* pero no pueden modificar, añadir o eliminar otros atributos del paquete original
  - Updated
    - Pueden añadir, modificar y eliminar atributos del paquete original

5. Discusión

- **Tamaño útil del chunk** afectado por:
  - Atributos de señalización (*Frag-Status y Service-Type)*
  - Proxies (*Proxy-State)*
- **Consideraciones de seguridad**
  - Este mecanismo no cambia nada a nivel de seguridad de RADIUS
  - Se requiere que los chunks vayan protegidos mediante el atributo *Message-Authenticator*
  - Se recomienda limitar el tamaño máximo de datos a recibir
    - Máximo 64 KB
    - No más de 20 chunks

6. Aplicabilidad

- Grupo de trabajo IETF ABFAB
  - Mecanismo de control de acceso a servicios basado en EAP, GSS-API, AAA y SAML
  - Necesitan transportar una sentencia SAML mediante AAA hacia el servicio tras la autenticación
- Reglas de filtrado de acceso a la red
  - Atributo *NAS-Filter-Rule* define una regla de forma explícita

7. Conclusiones y vías futuras

- En proceso de estandarización
  - Reciente adopción como documento de trabajo por el IETF RADEXT WG
    - draft-ietf-radext-radius-fragmentation-01
  - Colaboración activa con Alan DeKok (FreeRADIUS)
- Se ha implementado un prototipo en colaboración con TID que demuestra su viabilidad

¡ Gracias por su atención !

¿Alguna pregunta?