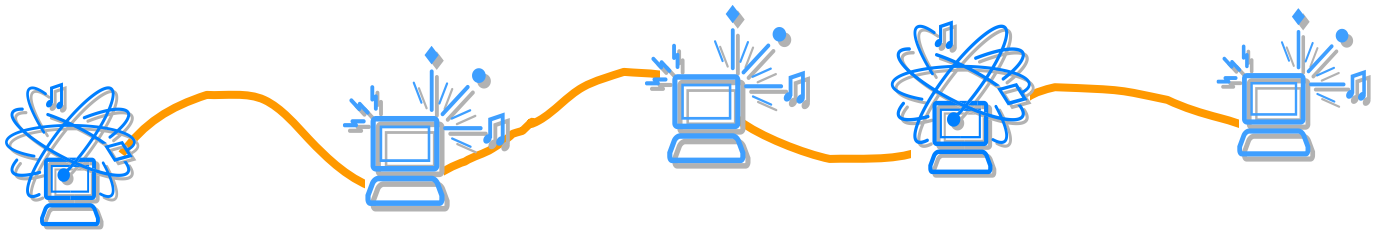
Segurança em Redes de Computadores
RADIUS – *Remote Access Dial In User Service*



Redes de Comunicação

Departamento de Engenharia da Electrónica e Telecomunicações e de Computadores

Instituto Superior de Engenharia de Lisboa

Objectivo

O RADIUS é um protocolo de controlo de acessos que permite a autorização, a autenticação e a contabilização dos utilizadores.

Objectivo

Embora sendo anterior ao aparecimento formal do modelo AAA (*Authentication, Authorization, Accounting*) do IETF, foi no entanto o primeiro protocolo baseado no AAA a exibir as funcionalidades AAA e a ganhar a aceitação e o uso generalizado por parte do mercado/indústria.

RADIUS – Segurança na camada de aplicação

- A confiança é estabelecida entre os clientes RADIUS e o servidor via segredo partilhado
- Suporta integridade e autenticação por pacote
  - Campos Request e Response Authenticator
  - Atributo Message-Authenticator
- Suporte para tornar confidenciais determinados atributos
  - Atributos normalizados: *User-Password, Tunnel-Password*
  - Atributos *Microsoft Vendor Specific Attributes* (VSA)
- Sem suporte generalizado de confidencialidade
- Sem proteção contra repetições
  - Campo *Authenticator* do *Authentication Request* com 128 bits pseudo-aleatórios e imprevisíveis
    - Não é um contador, os servidores RADIUS não testam a reutilização

Propriedades do RADIUS

- É um protocolo baseado em UDP que não utiliza suporte à ligação
- Utiliza o modelo de segurança *hop-by-hop*
- É *stateless*
- Suporta autenticação PAP e CHAP via PPP e outros métodos via EAP
- Utiliza MD5 nos algoritmos que lidam com as *passwords*
- Fornece dezenas de pares de atributos/valores com capacidade de suportar atributos específicos de vendedores
- Suporta o modelo autenticação-autorização-contabilização (AAA).

Limitações do RADIUS

- **Primeiro**, a segurança é um obstáculo em algumas implementações. Numa implementação onde existam vários servidores *proxy* de RADIUS todas as ligações devem ver, efectuar operações lógicas, e passar todos os dados no pedido, escondidos ou não. Isto significa que os dados ficam disponíveis em todos os saltos, o que não representa o ambiente mais seguro no qual colocar dados sensíveis como certificados e *passwords*.

- **Segundo**, pelo menos nas implementações mais comuns, não tem suporte para recolher ou deixar de atribuir recursos após ter sido realizada um autenticação.

- **Terceiro**, O RADIUS não guarda o estado (*stateless*) o que quer dizer que ele não se mantém informação sobre configurações, informação sobre transacções, ou qualquer outro dado para a próxima sessão.

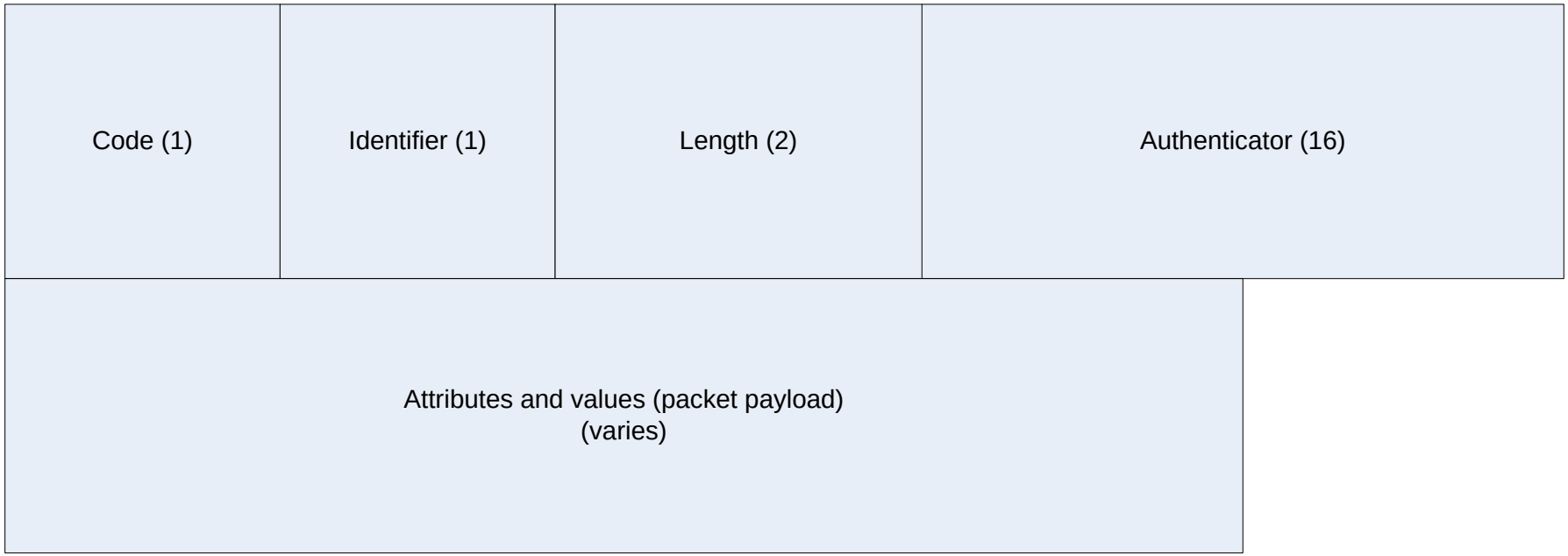
- Por **último**, os utilizadores de RADIUS têm verificado alguns problemas de escalabilidade.

Protocolo de transporte utilizado pelo RADIUS

- O RADIUS utiliza o **UDP** em detrimento do TCP.

- Utiliza o **porto 1812** (anteriormente era especificado o porto 1645 mas foi alterado dado entrar em conflito com o serviço Datametrics).

Formato das mensagens de RADIUS



As mensagens RADIUS são transportadas pelo UDP

Formato das mensagens de RADIUS

Code

- Serve para distinguir o tipo de mensagem RADIUS. As mensagens com *code* não válido são deitados fora sem notificação.
  - Códigos das mensagens:
    - 1 – Access-Request
    - 2 – Access-Accepted
    - 3 – Access-Rejected
    - 11 - Access-Challenge
    - 4 – Accouting-Request
    - 5 – Accounting-Response
    - 12 – Status-Server
    - 13 – Status-Client
    - 255 - Reservado

Formato das mensagens de RADIUS

Identifier

- É utilizado para dar suporte a multitarefa permitindo associar pedidos a respostas. Permite aos servidores RADIUS detectarem mensagens duplicadas.

Length

- É utilizado para especificar qual o comprimento da mensagem RADIUS. Os servidores RADIUS testam este campo para verificarem a integridade da mensagem. Se a dimensão da mensagem for superior ao indicado o excesso é deitado fora, se for menos a mensagem é descartada.

- O valor pode variar entre os 20 e 4096 bytes.

Formato das mensagens de RADIUS

Authenticator

- O campo *Authenticator* é o campo que dá suporte ao teste de integridade da carga. Existem dois tipos de específicos de valores de autenticação: Os valores de *request* e os de *response*.
- Os autenticadores de *request* são utilizados nos pacotes de *Access-Request* e no *Accounting-Request*. O valor do campo ***authenticator*** é a 128 bits e gerado de forma aleatória. O octeto de maior peso é transmitido primeiro.
- Nos autenticadores das outras mensagens que não a de request, o valor do ***authenticator*** é o MD5 protegido de todos os campos (***authenticator*** com o valor recebido no *request*).
- O NAS (autenticador) e o servidor RADIUS partilham um segredo. Desse segredo partilhado e do *request authenticator* é calculado o *hash* MD 5 para criar o valor a 128 bits, o qual é *xored* com a *password* introduzida pelo utilizador e o resultado colocado no atributo *user-password* do pacote *Access-Request* (ver adiante algoritmo utilizado).

Formato das mensagens de RADIUS

Response Authenticator

- O valor de *response* é utilizado nos pacotes *Access-Accepted*, *Access-Rejected* e *Access-Challenge*. O valor é calculado utilizando MD5 sobre os campos *code, identifier, length, request-authenticator,* carga do pacote (atributos) e segredo partilhado.

ResponseAuth = *MD5(Code+ID+Length+RequestAuth+Attributes+Secret)*

Formato das mensagens de RADIUS

Cálculo do valor a enviar no atributo *user-password*

- Designe-se o segredo partilhado por S e os 128 bits pseudo-aleatórios por *Request Authenticator* (RA). Parta-se a *password* em bocados de 128 bits p1, p2, etc., com o último bloco preenchido com nulos até ao limite de 128 bits . Chame-se aos blocos cifrados c(1), c(2), etc. Utilizando os valores intermédios b1, b2, etc. teremos:

$$b1 = MD5(S + RA); \qquad c(1) = p1 \text{ xor } b1;$$
$$b2 = MD5(S + c(1)); \qquad c(2) = p2 \text{ xor } b2;$$
$$\ldots \qquad .$$
$$bi = MD5(S + c(i-1)); \qquad c(i) = pi \text{ xor } bi$$

  o valor a colocar na *user-password* será c(1)+c(2)+...+c(i), onde + denota concatenação.

Segredos partilhados

- Para aumentar a segurança e aumentar a integridade nas transacções o RADIUS utiliza segredos partilhados.

- Segredos partilhados são valores gerados aleatoriamente que são dados a conhecer a ambos os lados, cliente e servidor.

- A única limitação é que os segredos partilhados devem ter dimensão superior a 0, embora o RFC aconselhe a que tenha pelo menos 128 bits. Os segredos partilhados devem ser únicos para cada par cliente-servidor do RADIUS.

Tipos de mensagens

- São quatro os tipos de mensagens RADIUS relevantes para as fases de autenticação e autorização:

1 – Access-Request

2 – Access-Accepted

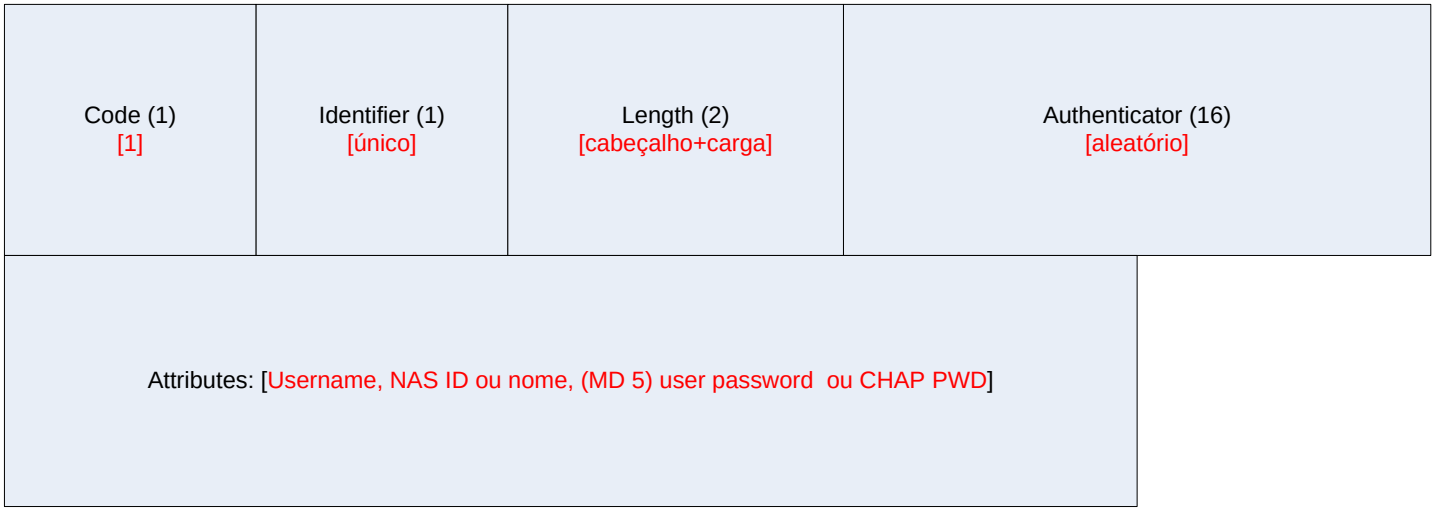
3 – Access-Rejected

11 - *Access-Challenge*

Tipos de mensagens

1 – Access-Request

- Envio de lista de serviços pretendidos incluindo o atributo *username* de quem está a tentar ganhar acesso, o endereço IP ou o nome canónico que está a enviar o pedido, deve conter a senha (PAP ou CHAP) *hashed* com MD5.
- Nas cadeias *proxy* de RADIUS é necessário criar novas mensagens sempre que for necessário alterar ou incluir outros atributos. Pode ser necessário decifrar e cifrar de novo (com a chave do próximo) antes de enviar para o servidor seguinte de RADIUS.
- Pode ser repetido na ausência de resposta por *timeout*. Pode ser escolhido outro servidor RADIUS alternativo.



Tipos de mensagens

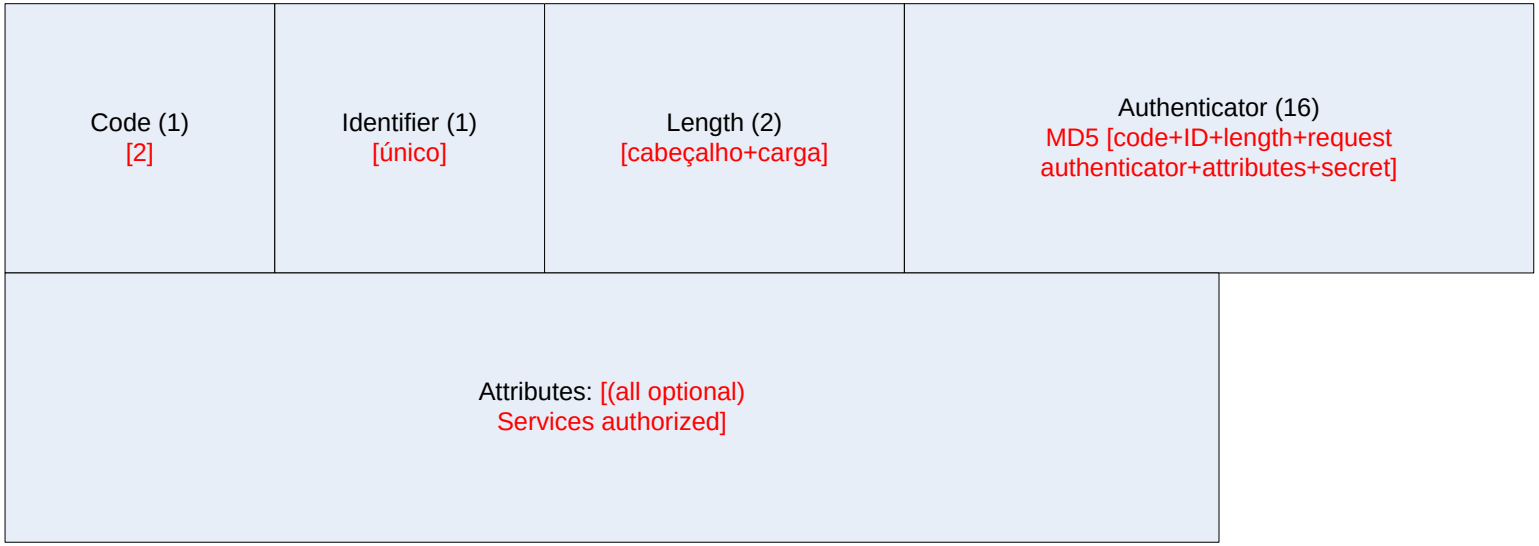
1 – Access-Request (cont.)

- Se não se verificar alguma das condições requeridas para a ligação do cliente, o servidor RADIUS envia um "*Access-Rejected*" que pode levar uma mensagem para ser mostrada ao utilizador (atributo *reply*).

Tipos de mensagens

2 – Access-Accepted

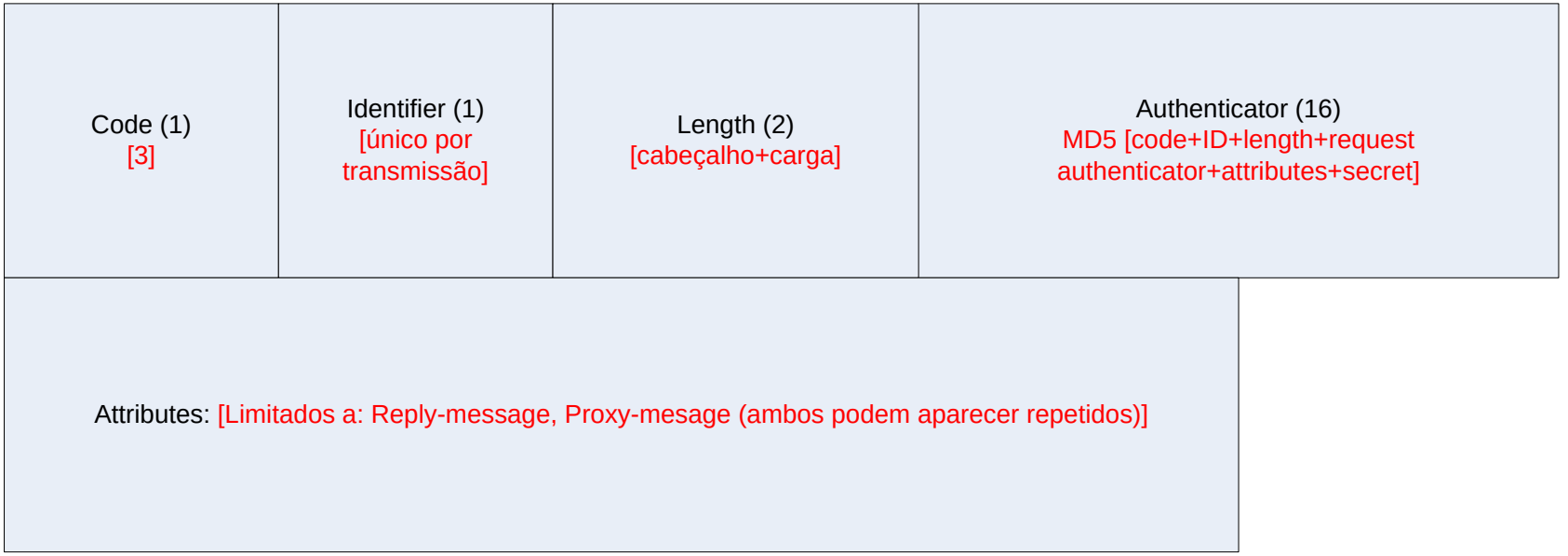
- Enviada pelo servidor de RADIUS para o cliente. O campo *Identifier* (ID) deve ser idêntico à da mensagem de *Access-Request* que deu origem a esta, caso contrário a mensagem de resposta é descartada.
- Pode conter atributos. Se não contiver assume que os atributos da mensagem de pedido foram todos aceites. Caso contenha atributos estes complementam e sobrepõem-se aos do pedido.



Tipos de mensagens

3 – Access-Rejected

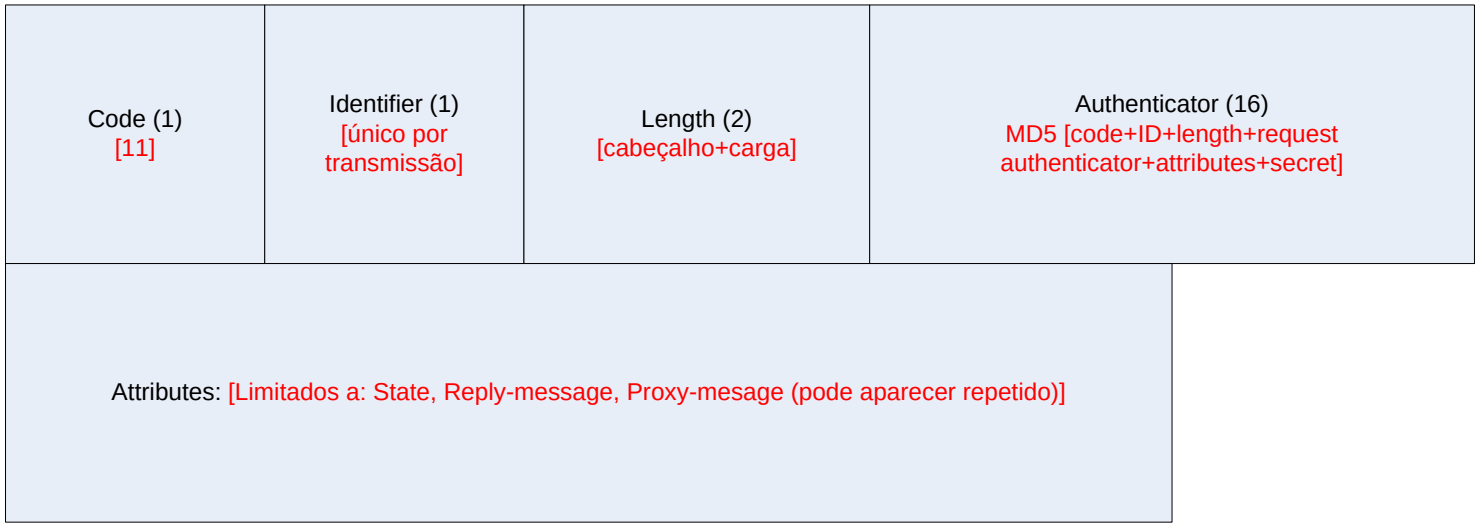
- Se algum dos serviços pedidos for rejeitado o servidor de RADIUS deve enviar esta mensagem ao cliente.
- Pode ser enviada em qualquer altura durante a sessão.
- A carga desta mensagem está limitada, para além de atributos específicos de vendedor, a dois atributos específicos: *Reply-Message* e *Proxy-State.*



Tipos de mensagens

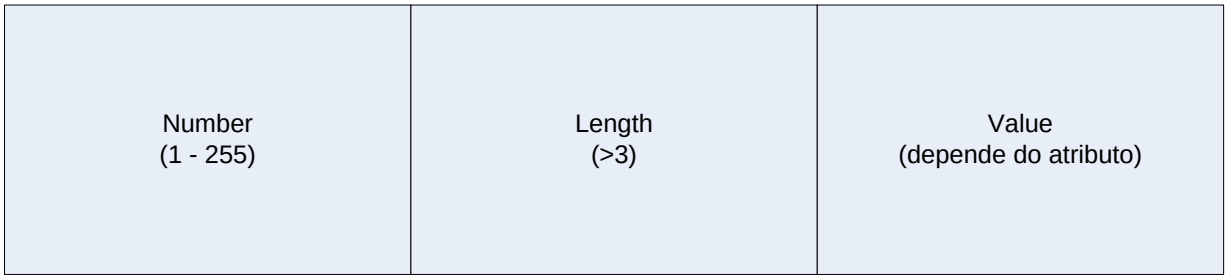
11 - Access-Challenge

- Se um servidor receber informação conflituosa e necessitar mais informação pode enviar este tipo de mensagem ao cliente. O cliente após receber este tipo de mensagem deve enviar um novo *Access-Request* com a informação apropriada incluída.
- Existem apenas dois atributos que podem ser incluídos neste tipo de mensagem: *State* e *Reply*, assim como atributos de vendedores. O atributo de *State* pode aparecer apenas uma vez, os outros podem aparecer várias. O atributo *State* é copiado inalterado para o *Access-Request* que é retornado ao servidor desafiante.



Atributos e valores

- A transacção RADIUS entre servidor e cliente é construída em torno da passagem de pares atributo-valor (AVP) entre o cliente e o servidor, os quais contêm virtualmente todas as propriedades e características da transacção AAA.
- Para aumentar a segurança o RFC restringe certos atributos de serem enviados em certas mensagens. Por exemplo, o atributo *User-Password* nunca é enviado num pacote de *reply* do servidor para o cliente.
- Os atributos nos pacotes têm um formato especifico, tipo TLV (*type, length, value*) composto por:
  - *Attribute number* (1 – 255)
  - *Attribute length* (>3)
  - *Value*



Atributos e valores

Request	Accept	Reject	Challenge	#	Attribute
0-1	0-1	0	0	1	User-Name
0-1	0	0	0	2	User-Password [Note 1]
0-1	0	0	0	3	CHAP-Password [Note 1]
0-1	0	0	0	4	NAS-IP-Address [Note 2]
0-1	0	0	0	5	NAS-Port
0-1	0-1	0	0	6	Service-Type
0-1	0-1	0	0	7	Framed-Protocol
0-1	0-1	0	0	8	Framed-IP-Address
0-1	0-1	0	0	9	Framed-IP-Netmask
0	0-1	0	0	10	Framed-Routing
0	0+	0	0	11	Filter-Id
0-1	0-1	0	0	12	Framed-MTU
0+	0+	0	0	13	Framed-Compression
0+	0+	0	0	14	Login-IP-Host
0	0-1	0	0	15	Login-Service
0	0-1	0	0	16	Login-TCP-Port

Atributos e valores

Request	Accept	Reject	Challenge	#	Attribute
0	0+	0+	0+	18	Reply-Message
0-1	0-1	0	0	19	Callback-Number
0	0-1	0	0	20	Callback-Id
0	0+	0	0	22	Framed-Route
0	0-1	0	0	23	Framed-IPX-Network
0-1	0-1	0	0-1	24	State [Note 1]
0	0+	0	0	25	Class
0+	0+	0	0+	26	Vendor-Specific
0	0-1	0	0-1	27	Session-Timeout
0	0-1	0	0-1	28	Idle-Timeout

Atributos e valores

Request	Accept	Reject	Challenge	#	Attribute
0	0-1	0	0	29	Termination-Action
0-1	0	0	0	30	Called-Station-Id
0-1	0	0	0	31	Calling-Station-Id
0-1	0	0	0	32	NAS-Identifier [Note 2]
0+	0+	0+	0+	33	Proxy-State
0-1	0-1	0	0	34	Login-LAT-Service
0-1	0-1	0	0	35	Login-LAT-Node

Atributos e valores (RFC 2869 RADIUS Extensions Junho 2000)

- 1-39 (RFC 2865, "RADIUS")
- 40-51 (RFC 2866, "RADIUS Accounting")
- 52 Acct-Input-Gigawords
- 53 Acct-Output-Gigawords
- 54 Unused 55 Event-Timestamp
- 56-59 Unused
- 60-63 (ver RFC 2865, "RADIUS")
- 64-67 (ver RFC 2868, "RADIUS Attributes for Tunnel Protocol Support")
- 68 (ver RFC 2867, "RADIUS Accounting Modifications for Tunnel Protocol Support")
- 69 (ver RFC 2868, "RADIUS Attributes for Tunnel Protocol Support")
- 70 ARAP-Password
- 71 ARAP-Features
- 72 ARAP-Zone-Access
- 73 ARAP-Security
- 74 ARAP-Security-Data
- 75 Password-Retry
- 76 Prompt
- 77 Connect-Info
- 78 Configuration-Token
- 79 EAP-Message
- 80 Message-Authenticator
- 81-83 (refer to [6])
- 84 ARAP-Challenge-Response
- 85 Acct-Interim-Interval
- 86 (refer to [7])
- 87 NAS-Port-Id
- 88 Framed-Pool
- 89 Unused
- 90-91 (refer to [6])
- 92-191 Unused

Suporte de EAP pelo RADIUS (RFC 2869)

- **Novos atributos relacionados com EAP:**
  - *EAP-Message* (79)
    - Pode transportar no atributo uma mensagem EAP
  - *Message-Authenticator* (80)
    - Serve para assinar as mensagens de *Access-request, Access-Accept, Access-Reject* ou *Access-Challenge* que contenham o atributo *EAP-Message*

Segurança da camada de aplicação RADIUS

- Confiança estabelecida entre clientes RADIUS e servidores através de segredo partilhado
- Suporte de autenticação e integridade por mensagem
  - Campos de autenticação do *Request* e do *Response*
  - Atributo *Message-Authenticator*
- Suporte para atributos específicos
  - Atributos normalizados: *User-Password, Tunnel-Password*
  - Atributos específicos de vendedores (VSAs)
- Sem suporte de confidencialidade
- Sem suporte para protecção contra repetições
  - O campo "*Request Authenticator*" é a 128 bits, pseudo-aleatório e não predizível.
    - Não é um contador.

Autenticação e integridade por mensagem

- Pacotes de autenticação sem o atributo *EAP-Message* (RFC 2865)
  - Sem autenticação por mensagem para as mensagens *Access-Request*
    - A mensagem *Access-Request* contém 128 bits pseudo-aleatórios no *Request Authenticator* (RA)
    - Nas mensagens *Access-Request* o RA é um *nonce*, não um *Authenticator*
    - É usado o *nonce* RA do *Access-Request* para esconder a *password* enviada nos *Access-Requests*

Resultado: As mensagens *Access-Request* não são autenticadas

Autenticação e integridade por mensagem

- Autenticação por mensagem para as mensagens *Access-Challenge, Access-Reject, Access-Accept*

  Response Authenticator *de* 128 bits =

  MD5(*Code* + *Identifier* + *Length* + *Request Authenticator* + *Attributes* + *Shared Secret*)

  – Nota: Os atributos *NAS-IP-Address* ou *NAS-Identifier* não DEVEM ser incluídos neste calculo dado não poderem ser incluídos em mensagens *Access-Challenge*, *Access-Reject* e *Access-Accept*.

Autenticação e integridade por mensagem

- **Autenticação de mensagens com o atributo *EAP-Message* (RFC 2869)**
  - Autenticação por mensagem para todas as mensagens
    - O RFC 2869 requer a inclusão do atributo *Message-Authenticator* nas mensagens contendo o atributo *EAP-Message (Access-Request, Access-Accept, Access-Reject, Access-Challenge)*
    - O atributo *Message-Authenticator* fornece a autenticação por mensagem.

Message-Authenticator = **HMAC-MD5** (*Type, Identifier, Length, Request Authenticator, Attributes*)

  - Para a mensagem *Access-Request* utiliza-se o *Request Authenticator* com 128 bits a 0
  - Para as mensagens *Access-Accept*, *Access-Reject*, *Access-Challenge* utiliza-se o *Message-Authenticator (Request Authenticator) do Access-Request* correspondente.

Esconder atributos

- ***User-Password*** (RFC 2865)
  - Utilizado na autenticação PAP do PPP (a cair em desuso)
    - O PAP é utilizado agora mais frequentemente quando se utiliza autenticação com *token card*
  - Também é utilizado na autenticação base HTTP
  - A autenticação com texto em claro não é suportada no EAP, desta forma os atributos *User-Password* nunca são enviados na autenticação IEEE 802.1x sobre RADIUS
  - A sequência da chave (*key stream*) é gerada a partir do segredo partilhado no RADIUS e dos 128 bits do *request authenticator*
    - B1 = MD5(Secret + *request authenticator*)
    - Bi = MD5(Secret + c(i-1))
  - O texto cifrado é o xor da sequência da chave com a *password* em claro
    - Ci = Pi xor Bi
    - Pi = i-nésimo bloco de 128 bits da *password*

Esconder atributos

- *Tunnel-Password* (RFC 2868)
  - Utiliza uma forma semelhante ao *User-Password* para esconder a *password*
    - B1 = MD5(Secret + *request authenticator* + Salt),
      - Salt = inteiro sem sinal de 16 bits. Deve ser único para cada Access-Accept, o bit mais à esquerda deve estar a 1.
- Microsoft VSAs (RFC 2548)
  - MS-CHAP-MPPE-Keys
    - Usado para transmitir as chaves do MS-CHAPv1
    - Os mesmos mecanismos que para o esquema *User-Password*
      - B1 = MD5(Secret + *request authenticator*)
  - MS-MPPE-Send-Key, MS-MPPE-Recv-Key
    - Pode ser usado para transmitir chaves EAP
    - Usa mecanismos semelhantes ao esquema do *Tunnel-Password*
      - B1 = MD5(Secret + *request authenticator* + Salt)

Vulnerabilidades do RADIUS

- Detalhes disponíveis em: http://www.untruth.org/~josh/security/radius/radius-auth.html
- Ataque *offline* de dicionário ao segredo partilhado do RADIUS via *Response Authenticator* do RFC 2865 ou dos *Request* ou *Response Authenticators* do RFC 2866
  - Muitas implementações só permitem segredos partilhados que contenham apenas caracteres ASCII, sejam menores que 16 caracteres tendo como resultado segredos partilhados do RADIUS com pouca entropia.
  - Um atacante pode capturar mensagens de *Access-Request/Response* ou *Accounting-Request* ou *Accounting-Response* para um ataque *offline* de dicionário
  - O estado do MD5 pode ser pré-calculado pelo que um ataque de dicionário pode ser eficiente
- Ataque *offline* de dicionário ao segredo partilhado do RADIUS via atributo EAP-Message
  - O atacante pode tentar um ataque *offline* a qualquer mensagem com atributo *EAP-Message*
  - A utilização do HMAC-MD5 no atributo *EAP-Message* torna o ataque mais difícil tornando o *Response Authenticator* o elo mais fraco.

Vulnerabilidades do RADIUS

- Detalhes disponíveis em: http://www.untruth.org/~josh/security/radius/radius-auth.html
- Decifração em tempo real de atributos escondidos
  - Um atacante autenticando-se via PAP pode, coleccionando mensagens RADIUS *Access-Request*, determinar a sequência usada para proteger o atributo *User-Password* .
  - Permite a um atacante coleccionar *Request Authenticators/Ids* às sequências correspondentes.
  - Para cada sequência capturada o atacante pode gerar novas sequências para cada valor do *Salt*.
  - Conforme a tabela de valores RA/ID/Salt aumenta a decifração de atributos *User-Password, Tunnel-Password* e *MPPE-Key* torna-se possível.
  - Nota: Onde o PAP não for usado (tal como na autenticação EAP), o ataque contra *User-Password* não é possível.
- Ataque "*Known plaintext*" contra atributo *Tunnel-Password*
  - Um atacante a quebrar uma *User-Password* pode enviar um *Access-Request* forjado e receber como resposta um *Access-Response* contendo um atributo *Tunnel-Password* e *Salt*
  - Dado que MD5(Secret+RA) é conhecido, tal como o *Salt*, é possível calcular imediatamente MD5(Secret+RA+Salt)
  - O atributo *Tunnel-Password* fica imediatamente comprometido!

Vulnerabilidades do RADIUS

- Ataque de dicionário *online* contra a *password* do PAP
  - Funciona com os servidores RADIUS permitindo a repetição dos campos de *Request Authenticator* (128 bits) e *Identifier* (um octeto)
  - Autenticando com o PAP e capturando o atributo *User-Password* o atacante pode derivar a sequência para um *request authenticator* e o ID
  - O atacante pode então tentar um ataque de dicionário *online* contra a *password* do utilizador de 16 caracteres ou menos, usando o mesmo *Request authenticator*, *Identifier* e sequência.
  - Nota: Este ataque não é possível se for requerido o atributo *Message-Authenticator* (tal como em mensagens EAP).

- Falsificação
  - Um atacante pode forjar mensagens RADIUS *Access-Request* (dado que estas mensagens não são autenticadas)
  - Nota: Este ataque não é possível se o atributo *Message-Authenticator* estiver presente (por exemplo, *EAP Access-Request*).

Vulnerabilidades do RADIUS

- Repetição de mensagens *Access-Accept/Reject*
  - O *Request Authenticator* é um valor a 128 bits pseudo-aleatório e imprevisível
  - No entanto nem todas as aplicações usam geradores de números pseudo-aleatórios credíveis
  - Por vezes o segredo é partilhado entre vários NASs – implica que o *Request Authenticator* deve ser global e temporalmente único em toda a rede.
  - Se o *Request Authenticator* e o *Identifier* forem reutilizados por vários NAS então o atacante pode repetir o *Access-Response* (possivelmente de outro NAS!)
  - A repetição não é limitada ao mesmo NAS dado que os atributos *NAS-Identifier* ou *NAS-IP-Address* **NÃO DEVEM** ser incluídos em mensagens *Access-Response* .

São possíveis os ataques de dicionário *offline* ao segredo partilhado do RADIUS?

- Numa resposta simples: Sim
- Um ataque deste tipo requer apenas a captura de um par *Request/Response Authenticator*
- Os administradores frequentemente escolhem segredos partilhados propícios a ataques de dicionário:
  - As implementações de RADIUS por vezes apenas aceitam *passwords* alfanuméricas
  - Por exemplo as palavras inglesas no dicionário têm uma entropia de apenas 1,2 bits por carácter
- Alguns segredos partilhados são usados múltiplas vezes em vários NAS
- Uma vez comprometido o segredo partilhado o RADIUS torna-se ineficaz
  - Atributos escondidos podem ser decifrados em tempo real
  - Todos os tipos de mensagens podem ser forjadas
  - Mas …
    - Ainda é necessário montar ataques de dicionário *offline* para o CHAP e o EAP-MD5
    - Não ajudam métodos menos vulneráveis a ataques como o EAP-TLS ou SRP (***S**ecure **R**emote **P**assword*) [http://srp.stanford.edu/]

É mesmo possível a decifração em tempo real?

- Se o *Request-Authenticator* for aleatório e único, global e temporalmente (como requerido pela RFC 2865), então este ataque não é possível.
  - Exemplo
    - A 10 Gbps, 1 milhão de NAS podem enviar um máximo de 2 biliões de mensagens RADIUS *Access-Request*/segundo ou 73.54 quadriliões de *Access-Requests*/ano
    - Analisar os *request authenticator* de 128 bits levaria mais de um trilião de anos!
- No entanto, se um *Request Authenticator* não for gerado aleatoriamente, pode ser repetido.
  - Usando o mesmo segredo partilhado em cada NAS torna isto mais provável.

Sumário - Vulnerabilidades

	PPP	**PPP**	**PPP/802**	**PPP/802**
Attack	**PAP**	**CHAP**	**EAP-MD5**	**EAP-TLS/SRP**
Ataque de dicionário *offline* ao segredo partilhado do RA	X	X	X	X
Decifracção *online* aos atributos escondidos	?			
Ataque de dicionário ao CHAP *Response*		X	X	
Ataque *online* contra a *pasword*	X	X		
Mensagens forjadas de *Access-Request*	X	X		
Repetição de mensagens *Access-Accept/Reject*	?	?	?	?

Remendos aconselhados

- Não permitir o PAP
  - A autenticação EAP já requer isto
- Uso de um gerador credível para o *Request Authenticator* (verRFC 1750)
- Uso do RADIUS sobre IPsec ESP (RFC 3162)
- Inclusão de um atributo *Message-Authenticator* em todas as mensagens
  - O RFC 2869 já requer a autenticação EAP
- Usar um segredo partilhado para o RADIUS de elevada entropia
  - Não limitar o segredo partilhado a 16 caracteres
  - Utilizar segredos partilhados gerados aleatoriamente
- Uso de um segredo partilhado diferente para cada par de cliente-servidor RADIUS.

Requisitos gerais de autenticação para acesso a redes

- Identificação única dos utilizadores nos limites da rede
- A falsificação de identidade deve ser impossível
- Fácil de utilizar para o utilizador final
- Manutenção dos utilizadores de cada instituição numa base de dados da rede da instituição
- Baixa manutenção
- Facilidade de utilização por convidados
- Possibilidade de vários mecanismos de autenticação
- Requisitos adicionais para o acesso a redes:
  - Atribuição de VLAN por grupos de utilizadores
  - Acesso *wireless* cifrado

RADIUS sobre IPsec

- Suporte de integridade por mensagem, autenticação, confidencialidade e protecção contra repetições, quer para mensagens de autenticação, quer de contabilidade.

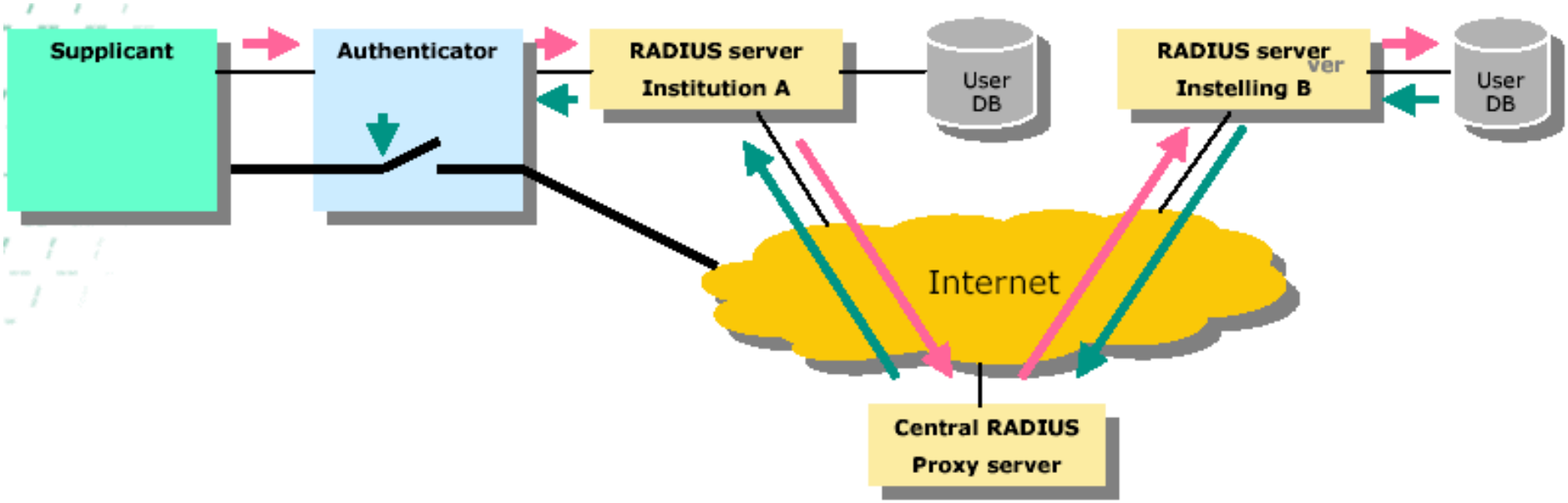
- Utilização descrita no RFC 3162.

RADIUS-*proxy*

- A instituição A apenas conhece os seus próprios utilizadores (utilizador_A@deetc.isel.ipl.pt), mas confia noutras instituições (por ex.: comunidade e-U)

- Para permitir a utilização por utilizadores pertencentes à comunidade a que pertence (por ex.: e-U), a instituição envia, de forma transparente para o utilizador, mensagens "RADIUS-request" relativas aos utilizadores que não constem na base de dados (utilizador_B@instituição_B.pt) para um *proxy* RADIUS central, o qual reenvia a mensagem para a instituição correcta. Qualquer que seja a forma de autenticação utilizada na instituição B pode ser usada na rede da instituição A.

Como funciona o *proxy* do RADIUS



Mais informação

- 802.1x http://standards.ieee.org/reading/ieee/std/lanman/802.1X-2001.pdf
- RFC's: see http://www.ietf-editor.org
- EAP RFC 2284
- EAP-MD5 RFC 1994, RFC 2284
- EAP-TLS RFC 2716
- EAP-TTLS http://www.funk.com/NIdx/draft-ietf-pppext-eap-ttls-01.txt
- PEAP http://www.globecom.net/ietf/draft/draft-josefsson-pppext-eap-tls-eap-02.html
- RADIUS RFC 2865, 2866, 2867, 2868, 2869 (I/w EAP)