Projeto 5 (1ªParte) Diogo Antunes 2018016615

1)

```
root@AAA-1:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.100.18  netmask 255.255.255.0  broadcast 0.0.0.0
        inet6 fe80::42:90ff:fe39:c100  prefixlen 64  scopeid 0x20<link>
        ether 06:23:66:15:aa:04  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 656 (656.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

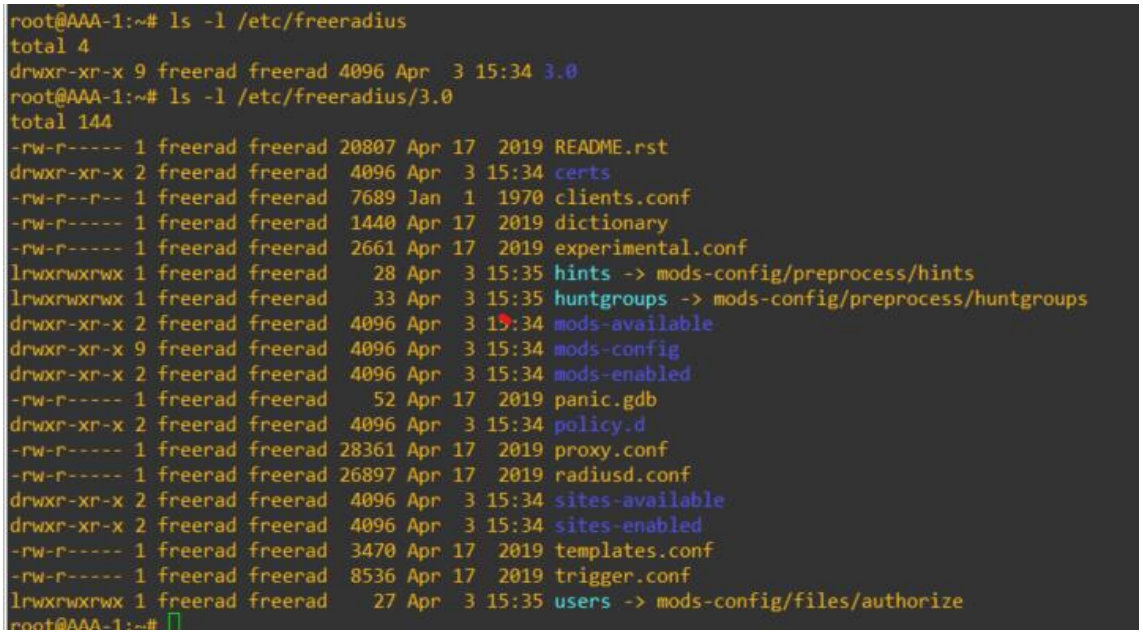
2)

```
root@AAA-1:~# netstat -anu
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 127.0.0.1:18120         0.0.0.0:*
udp        0      0 0.0.0.0:1812            0.0.0.0:*
udp        0      0 0.0.0.0:1813            0.0.0.0:*
udp        0      0 0.0.0.0:56089           0.0.0.0:*
udp6       0      0 :::54648                :::*
udp6       0      0 :::1812                 :::*
udp6       0      0 :::1813                 :::*
root@AAA-1:~# netstat -anu | grep -E "1812|1813"
udp        0      0 127.0.0.1:18120         0.0.0.0:*
udp        0      0 0.0.0.0:1812            0.0.0.0:*
udp        0      0 0.0.0.0:1813            0.0.0.0:*
udp6       0      0 :::1812                 :::*
udp6       0      0 :::1813                 :::*
root@AAA-1:~#
```

Na imagem em cima pode-se visualizar os Sockets UDP ativos e depois filtrei pelos serviços Radius (porta 1812) e Radius-acct (porta 1813).

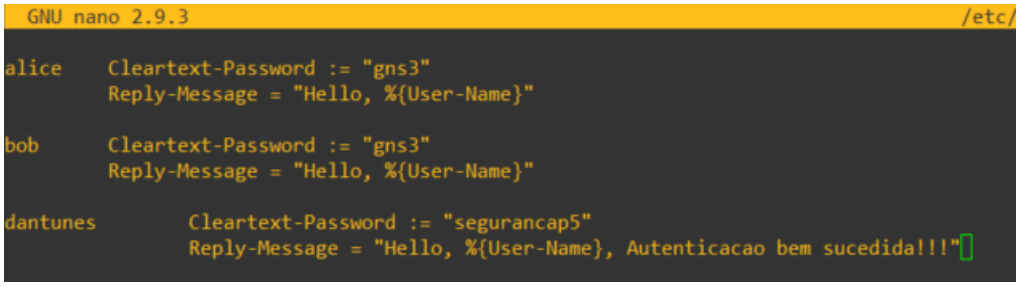
3)

```
root@AAA-1:~# service --status-all
 [ + ]  freeradius
 [ ? ]  hwclock.sh
 [ - ]  procps
 [ + ]  tacacs_plus
root@AAA-1:~# uname -a
Linux AAA-1 5.15.0-130-generic #140~20.04.1-Ubuntu SMP Wed Dec 18 21:35:34 UTC 2024 x86_64
x86_64 x86_64 GNU/Linux
```

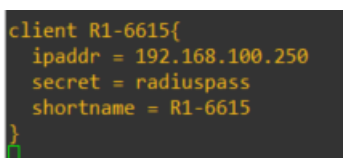



Como se pode verificar na imagem acima, está presente o ficheiro users, ou seja, onde estão guardadas as contas dos utilizadores autenticados pelo Radius, como é o caso do bob e da alice.

https://lycansec.wordpress.com/2020/05/15/instalando-um-servidor-radius-v3-no-linux-para-testes-simples-de-redes-enterprise/




4)



5)

```
R1-6615#debug aaa authentication
AAA Authentication debugging is on
R1-6615#debug aaa authorization
AAA Authorization debugging is on
R1-6615#debug aaa accounting
AAA Accounting debugging is on
R1-6615#terminal monitor
% Console already monitors
R1-6615#
```

6)

```
R1-6615(config)#username diogoa secret utilizadorlocal
R1-6615(config)#
```

```
R1-6615(config)#radius server radius1
R1-6615(config-radius-server)#$4 192.168.100.18 auth-port 1812 acct-port 1813
R1-6615(config-radius-server)#key radiuspass
R1-6615(config-radius-server)#exit
R1-6615(config)#aaa authentication login default group radius local
R1-6615(config)#
```

```
R1-6615(config-line)#line con 0
R1-6615(config-line)#privilege level 15
R1-6615(config-line)#
```

```
line vty 0 4
 password 7 095F4B0E0C17161C080D053F27253B
 transport input telnet ssh
!
```

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0.000000 | - | 06:23:66:15:aa:f1 | 06:23:66:15:aa:f1 | LOOP | 60 Reply |
| 2 | 1.334206 | - | 192.168.100.250 | 192.168.100.18 | RADIUS | 102 Access-Request id=5 |
| 3 | 2.335694 | - | 192.168.100.18 | 192.168.100.250 | RADIUS | 109 Access-Reject id=5 |
| 4 | 6.371715 | - | 192.168.100.250 | 192.168.100.18 | RADIUS | 102 Access-Request id=5, Duplicate Request |
| 5 | 7.372253 | - | 192.168.100.18 | 192.168.100.250 | RADIUS | 109 Access-Reject id=5, Duplicate Response |
| 6 | 7.557132 | - | 06:23:66:15:aa:04 | 06:23:66:15:aa:f1 | ARP | 42 Who has 192.168.100.250? Tell 192.168.100.18 |
| 7 | 7.564084 | - | 06:23:66:15:aa:f1 | 06:23:66:15:aa:04 | ARP | 60 192.168.100.250 is at 06:23:66:15:aa:f1 |
| 8 | 9.994338 | - | 06:23:66:15:aa:f1 | 06:23:66:15:aa:f1 | LOOP | 60 Reply |
| 9 | 11.416703 | - | 192.168.100.250 | 192.168.100.18 | RADIUS | 102 Access-Request id=5, Duplicate Request |
| 10 | 12.418127 | - | 192.168.100.18 | 192.168.100.250 | RADIUS | 109 Access-Reject id=5, Duplicate Response |
| 11 | 16.452450 | - | 192.168.100.250 | 192.168.100.18 | RADIUS | 102 Access-Request id=5, Duplicate Request |
| 12 | 17.453513 | - | 192.168.100.18 | 192.168.100.250 | RADIUS | 109 Access-Reject id=5, Duplicate Response |
| 13 | 19.996200 | - | 06:23:66:15:aa:f1 | 06:23:66:15:aa:f1 | LOOP | 60 Reply |
| 14 | 22.560571 | - | 06:23:66:15:aa:f1 | Broadcast | ARP | 60 Who has 192.168.100.100? Tell 192.168.100.250 |

```
R1-6615#test aaa group radius dantunes segurancap5 legacy
Attempting authentication test to server-group radius using radius
No authoritative response from any server.

R1-6615#
000066: *Apr  3 20:07:05.653 UTC: %SEC_LOGIN-1-QUIET_MODE_ON: Still timeleft for watching f
ailures is 2 secs, [user: dantunes] [Source: 192.168.100.15] [localport: 22] [Reason: Login
 Authentication Failed] [ACL: sl_def_acl] at 20:07:05 UTC Thu Apr 3 2025
R1-6615#
```

```
(dantunes@192.168.100.250) Password:
(dantunes@192.168.100.250) Password:
(dantunes@192.168.100.250) Password:
Connection closed by 192.168.100.250 port 22
root@A1:~#
```

Como se pode verificar, nas PDUS, existe o pedido Access-Request, no entanto é sempre rejeteitado.

7)

```
root@AAA-1:~# service freeradius reload
 * Checking FreeRADIUS daemon configuration...                              [ OK ]
 * FreeRADIUS daemon is running
 * Reloading FreeRADIUS daemon freeradius                                   [ OK ]
root@AAA-1:~#
```

O scrict freeradius pode ser encontrado ou visualizado com o comando /etc/init.d/freeradius.

```
reload)
    configtest || exit 150

    if status_of_proc -p "$PIDFILE" "$PROG" "$DESCR"; then
        log_daemon_msg "Reloading $DESCR" "$PROG"

        start-stop-daemon --stop --signal HUP --quiet --pidfile $PIDFILE || ret=$?
        log_end_msg $ret
    fi
    ;;
```

Na imagem acima pode-se verificar o script relativo ao comando reload. No caso deve ser usado o comando service freeradius reload.

```
  status)
      status_of_proc -p "$PIDFILE" "$PROGRAM" "$PROG" && exit 0 || exit $?
      ;;
```

Na imagem acima pode-se verificar o script relativo ao comando status. No caso deve ser usado o comando service freeradius status.

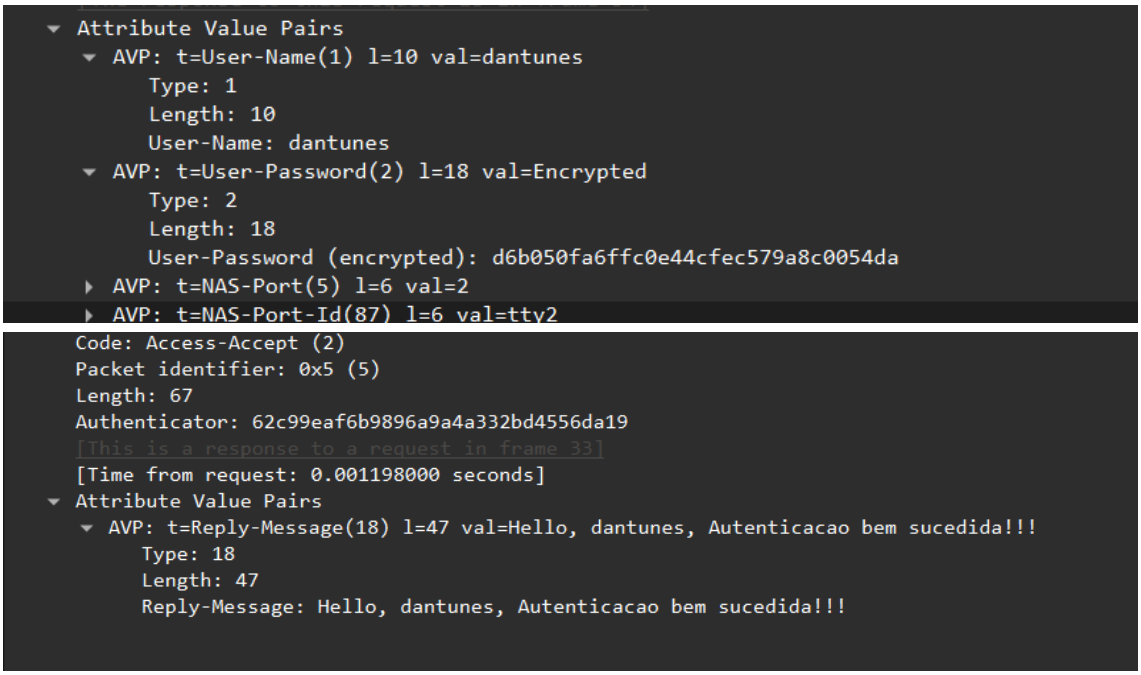
```
root@AAA-1:~# service freeradius status
 * freeradius is running
root@AAA-1:~#
```

```
root@A1:~# ssh dantunes@192.168.100.250




 __          __  _
 \ \        / / | |
  \ \  /\  / /__| | ___ ___  _ __ ___   ___
   \ \/  \/ / _ \ |/ __/ _ \| '_ ` _ \ / _ \
    \  /\  /  __/ | (_| (_) | | | | | |  __/
     \/  \/ \___|_|\___\___/|_| |_| |_|\___|

(dantunes@192.168.100.250) Password:
Hello, dantunes, Autenticacao bem sucedida!!!
 Acesso apenas a pessoas autorizadas!!!!



          Bem Vindo ao modo EXEC!!!!!
R1-6615>en
Password:
R1-6615#
```

Após executar o comando “service freeradius reload”, o serviço de Radius releu as novas configurações e assim as configurações do novo utilizador ficaram disponíveis. Consequentemente, consegui comunicar por ssh com esse utilizador.

```
33 4.260358      -          192.168.100.250      192.168.100.18       RADIUS    114 Access-Request id=5
34 4.261556      -          192.168.100.18       192.168.100.250      RADIUS    109 Access-Accept id=5
```

Como se pode verificar na captura do wireshark houve sucesso na autenticação Radius. Primeiramente o Router1 enviou um pedido de autenticação para o servidor Radius (Access-Request). Seguidamente, o servidor verificou se os dados de autenticação eram válidos comparando com as informações presentes nos ficheiros de configuração, tanto do cliente Router como do utilizador criado. Como estes eram válidos o servidor repondeu com um PDU Access-Accept.




A Ofuscação da Password é feita de modo a proteger a mesma a partir de MD5 e da operação XOR. Primeiramente o Request Authenticator é combinado com a shared secret, depois passando para um hash MD5. Depois a password é cifrada por blocos de 16 bytes com XOR utilizando os hashes MD5 iterativamente.

**Diagrama Temporal:**

**-** O utilizador inicia uma sessão via SSH enviando o seu username para a autenticação;

- O R1 envia um Access-Request para o servidor FreeRadius, contendo o username e a password já ofuscadace um campo Authenticator que é gerado aleatoreamente.

-O servidor FreeRadius valida os dados de configuração, comparando-os com as informações armazenadas na sua base de dados.

-Se for válido o servidor responde com um Access-Accept, enviando a resposta ao Router, que seguidamente reencaminha ao terminal onde está a ser feita a comunicação SSH, para informar ao utilizador que pode iniciar a sessão SSH.

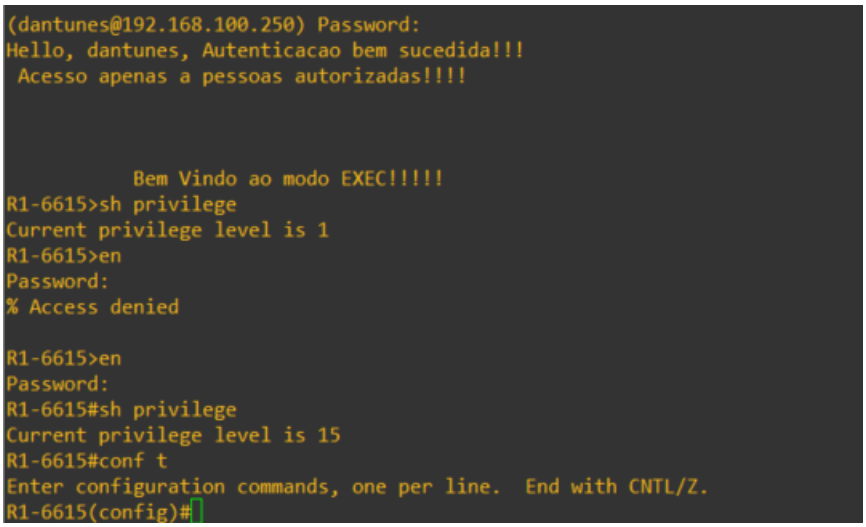
8)

**Como o Authenticator é calculado?**

Primeiramente, O Authenticator é um valor de 16 bytes gerado aleatoriamente, neste caso pelo R1.

Para o poder calcular faz-se a soma dos campos do Access Request, Code, Identifier, Lenght, Authenticator, Atributes (Parte dos dados Radius) e Sahred Secret. Após a soma ou concatenação destes campos aplica-se um hash MD5.

9)

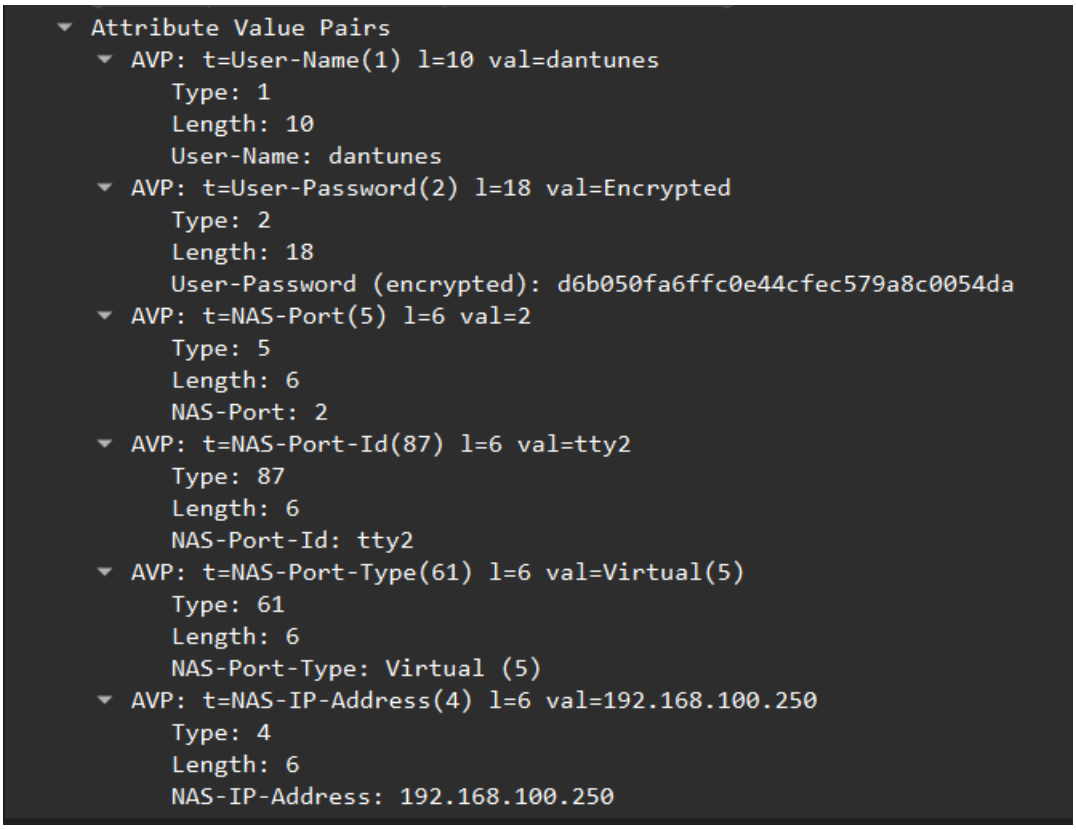



Após fazer a autenticação via SSH, pode-se verificar que o utilizador está com o nível de privilégio 1 (mdo EXEC de utilizador). Após aceder ao enable colocando a password, verifica-se que o utilizador está com o nível de privilégio 15. E consequentemente verificou-se que o utilizador pode entrar no modo de configuração.

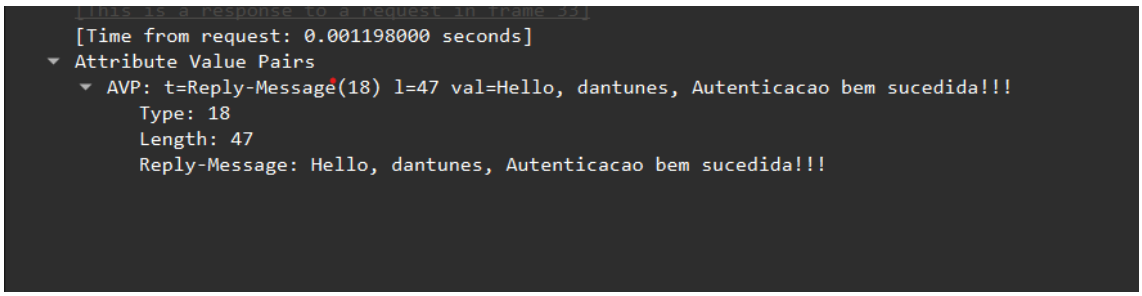
Após a autenticação RADIUS bem-sucedida, o usuário recebeu, por padrão, o nível de privilégio 1 no modo EXEC. Isso aconteceu porque o servidor RADIUS não enviou nenhum atributo de autorização específico instruindo o roteador (R1) a conceder um nível de privilégio diferente, como o nível 15.

A configuração aaa authorization exec default group radius local no R1 garantiu que a autorização para iniciar a sessão EXEC fosse aceita. No entanto, como não foi definida nenhuma instrução explícita para alterar o nível de acesso do usuário, o R1 aplicou automaticamente o nível padrão para usuários autenticados, que é o privilégio 1. Para obter acesso ao modo de configuração global (configure terminal), seria necessário elevar o privilégio para o nível 15 usando o comando enable.

AVP: Attribute-Value Pairs Access-Request




AVP: Attribute-Value Pairs Access-Accept



10)

```
R1-6615#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1-6615(config)#line vty 0 4
R1-6615(config-line)#privilege level 15
R1-6615(config-line)#
```

```
(dantunes@192.168.100.250) Password:
Hello, dantunes, Autenticacao bem sucedida!!!
 Acesso apenas a pessoas autorizadas!!!!



          Bem Vindo ao modo EXEC!!!!!
R1-6615#sh privilege
Current privilege level is 15
R1-6615#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1-6615(config)#line vty 0 4
R1-6615(config-line)#no privilege 15
                             ^
% Invalid input detected at '^' marker.

R1-6615(config-line)#no privilege level 15
R1-6615(config-line)#do wr
Building configuration...
[OK]
R1-6615(config-line)#
```

Ao adicionar o comando privilege level 15 à configuração das linhas VTY, estamos a instruir o roteador a atribuir automaticamente o nível de privilégio 15 a qualquer usuário que se autentique com sucesso através da linha SSH, sem considerar o que o servidor RADIUS ou a configuração local do usuário possam especificar em relação aos privilégios. Isso significa que essa configuração nas linhas VTY tem prioridade e será a primeira a ser aplicada para definir o nível de privilégio inicial.

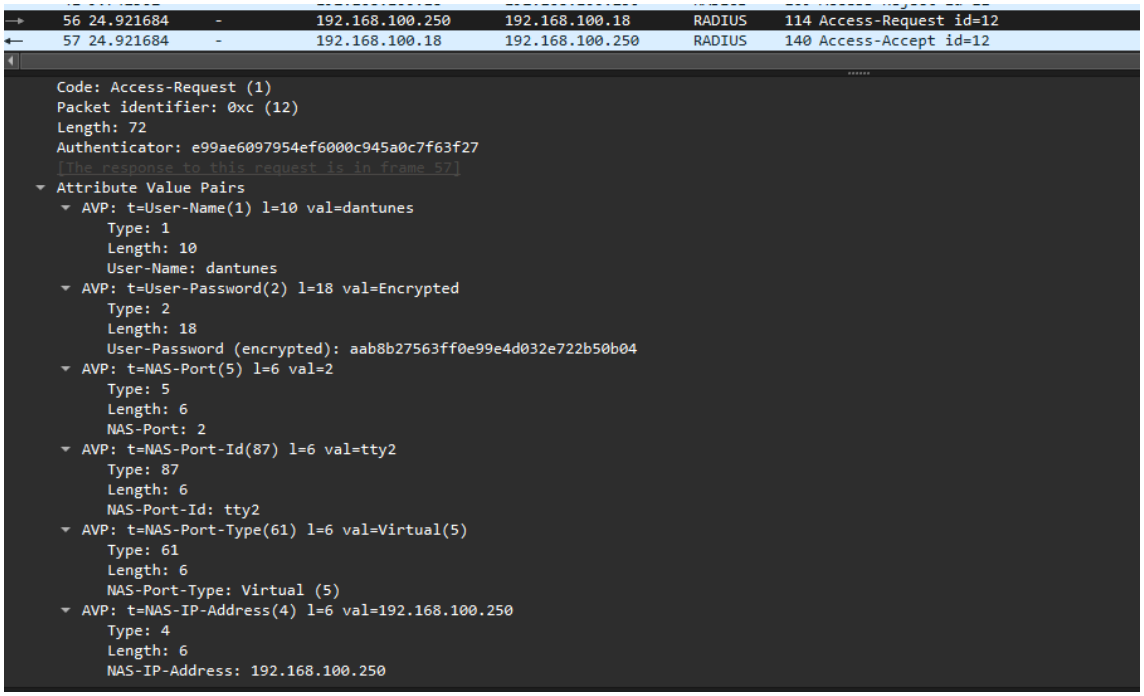
11)

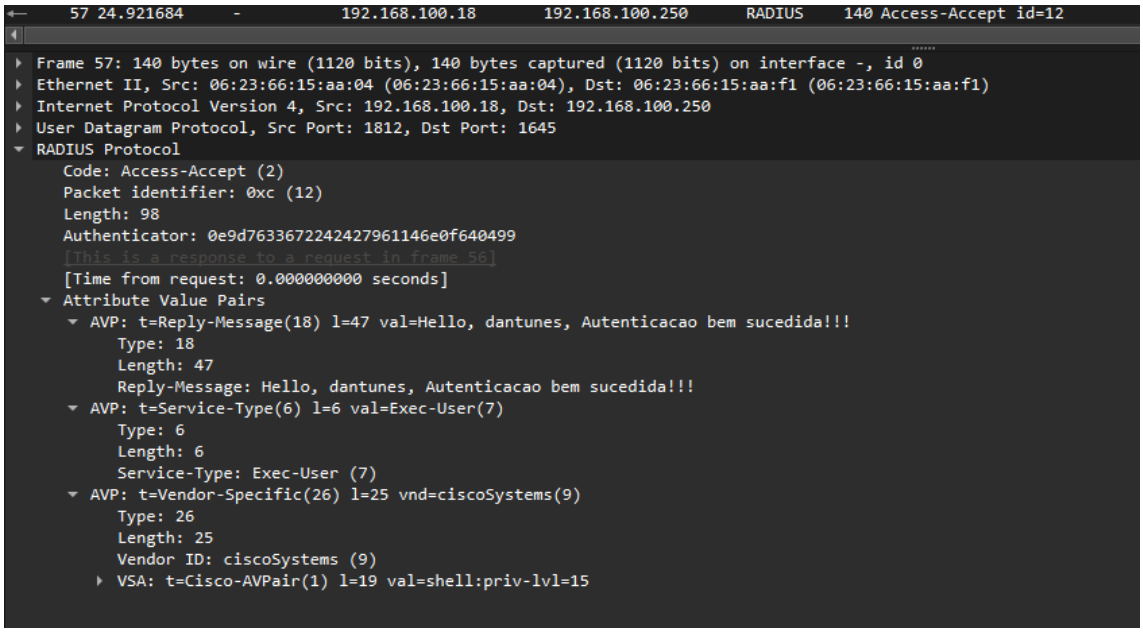
```
dantunes        Cleartext-Password := "segurancap5"
                Reply-Message = "Hello, %{User-Name}, Autenticacao bem sucedida!!!",
                Service-Type = NAS-Prompt-User,
                Cisco-AVPair = "shell:priv-lvl=15"
```

```
root@AAA-1:~# service freeradius restart
 * Checking FreeRADIUS daemon configuration...                          [ OK ]
 * Stopping FreeRADIUS daemon freeradius                                [ OK ]
 * Starting FreeRADIUS daemon freeradius                                [ OK ]
root@AAA-1:~# service freeradius status
 * freeradius is running
root@AAA-1:~#
```

```
R1-6615(config)#aaa authorization exec default group radius
R1-6615(config)#
```

```
(dantunes@192.168.100.250) Password:
Hello, dantunes, Autenticacao bem sucedida!!!
 Acesso apenas a pessoas autorizadas!!!!



          Bem Vindo ao modo EXEC!!!!!
R1-6615#
```




Na captura anterior a PDU Access-Accept apenas tinha o campo Reply-message. Agora pode-se verificar que tem também os atributos service-type e Vendor-Specific Cisco-AVPair.

O service-type, especifica o tipo de serviço autorizado para o utilizador.

O Cisco-AVPair é o atributo que define o nível de privilégio do utilizadador, neste caso 15 ("shell:priv-lvl=15").

**Vendor ID da Cisco para VSAs**

O Vendor ID da Cisco para os seus Vendor-Specific Attributes (VSAs) RADIUS é 9. Este ID é atribuído pela IETF (Internet Engineering Task Force), que é a entidade responsável pela regulamentação dos padrões da Internet.

Podemos encontrar uma lista atualizada dos IDs de fornecedores (Vendor IDs) e outros atributos no site do IANA (Internet Assigned Numbers Authority), que é responsável pela distribuição desses IDs.

https://www.iana.org/assignments/enterprise-numbers

12)


Ao tentar aceder por ssh, após parar o serviço Radius, verificou-se um atraso, pois o router estava a tentar comunicar com o servidor Radius, no entanto como este está “down”, não obtia resposta. Como o router tem um tempo de espera para obter a resposta obteve-se um atraso.

Como o utilizador apenas está configurado na base de dados do servidor Radius e não está presente na base de dados local, não é possível haver autenticação.