# Research Report

Network traffic

This chapter describes the observed network traffic between the ssh client, identity card application, ssh server and radius server on the Moonshot Pilot DVD.

The ssh client first sets up a TCP connection with the ssh server, after which they complete the SSH key exchange. Seemingly directly after the key exchange, the ssh client opens the moonshot identity card application in which the user selects an identity to “send”. At this point, either the identity card application or the ssh client itself, connects to the freeradius server and initiates an EAP-TTLS inner tunnel. Once the secure tunnel is set up, authentication against the radius server is performed and an Access-Accept is returned with a vendor-specific attribute containing a SAML assertion. This is used to finalize the authentication and the ssh server gives the user (in this case steve) the generic “moonshot” account on the server.

Application legend:

   [SSH] OpenSSH-Moonshot client

   [SSHD] OpenSSH-Moonshot server

   [IDC] Identity Cards application

   [RAD] FreeRADIUS server

Step 1: TCP Handshack

   1. [SSH] sends SYN to [SSHD]

   2. [SSHD] sends SYN, ACK to [SSH]

   3. [SSH] sends ACK to [SSHD]

Step 2: SSH Key Exchange

   1. [SSHD] sends SSH version identification (SSH-2.0-OpenSSH\_5.9p1 Debian-5+moonshot1\x0d\x0a) to [SSH]

   2. [SSH] sends SSH version identification (SSH-2.0-OpenSSH\_5.9p1 Debian-5+moonshot1\x0d\x0a) to [SSHD]

   3. [SSH] sends the "Client Key Exchange Init" message containing cookie and a list of all supported encryption and key exchange algorithms to [SSHD]

   4. [SSHD] sends the "Server Key Exchange Init" message containing cookie and a list of all supported encryption and key exchange algorithms to [SSH]

   5. [SSH] sends a message specifying that the "Diffie-Hellman" key exchange algorithm will be used to [SSHD]

   6. [SSHD] sends a "Server: New Keys" message containing key information to [SSH]

   7. [SSH] sends a "Client: New Keys" message containing only the message message code 0x15 (new keys) to [SSHD]

Step 3: RADIUS authentication

   1. [IDC] Access-Request with AVPs to [RAD]:

   -User-Name="@<realm>" (@local in this case)

   -0x4a 0x06="host"

   -0x5a 0x08="<hostname>"

   -Eap-Message=EAP with type=identity

   -Message-Authenticator with its value

   2. [RAD] Access-Challange with aVPs to [IDC]

   -Eap-Message=EAP with type=EAP-TTLS with start bit set

   -Message-Authenticator=<MA>

   -State=<state>

   3. [IDC] Access-Request with AVPs to [RAD]:

   -User-Name="@<realm>"

   -0x4a 0x06="host"

   -0x5a 0x08="<hostname>"

   -Eap-Message=EAP with type=EAP-TTLS

   -TLS Client Hello

   -State=<state>

   -Message-Authenticator=<MA>

   4. [RAD] Access-Challange with AVPs to [IDC]:

   -EAP-Message (consists of 4 AVPs with a total of 813 bytes)

   -Server Hello

   -gmt\_unix\_time

   -random\_bytes

   -Cipher Suites=TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA

   -Certificate

   -Containing the certificate

   -Server Hello Done

   -Message-Authenticator=<MA>

   -State=<state>

    5. [IDC] Access-Request with AVPs to [RAD]:

   -User-Name="@<realm>"

   -0x4a 0x06="host"

   -0x5a 0x08="<hostname>"

   -EAP-Message

   -Client Key Exchange (RSA encrypted premaster)

   -Change Cipher Spec

   -Encrypted Handshake Message

   -State=<state>

   -Message-Authenticator=<MA>

   6. [RAD] Access-Challange with AVPs to [IDC]:

   -EAP-Message

   -Change Cipher Spec

   -Encrypted Handshake Message

   -Message-Authenticator=<MA>

   -State=<state>

   7. [IDC] Access-Request with AVPs to [RAD]:

   -User-Name="@<realm>"

   -0x4a 0x06="host"

   -0x5a 0x08="<hostname>"

   -EAP-Message

   -Encrypted data

   -State=<state>

   -Message-Authenticator=<MA>

   8. [RAD] Access-Challange with AVPs to [IDC]:

   -EAP-Message

   -Encrypted data

   -State=<state>

   -Message-Authenticator=<MA>

   9. [IDC] Access-Request with AVPs to [RAD]:

   -User-Name="@<realm>"

   -0x4a 0x06="host"

   -0x5a 0x08="<hostname>"

   -EAP-Message

   -Encrypted data

   -State=<state>

   -Message-Authenticator=<MA>

   10. [RAD] Access-Challange with AVPs to [IDC]:

   -EAP-Message

   -Encrypted data

   -State=<state>

   -Message-Authenticator=<MA>

   11. [IDC] Access-Request with AVPs to [RAD]:

   -User-Name="@<realm>"

   -0x4a 0x06="host"

   -0x5a 0x08="<hostname>"

   -EAP-Message

   -Encrypted data

   -State=<state>

   -Message-Authenticator=<MA>

   12. [RAD] Access-Accept with AVPs to [IDC]:

   -User-Name="<username>@<realm>"

   -MS-MPPE-Recv-Key=<key>

   -MS-MPPE-Send-Key=<key>

   -EAP-Message

   -Success

   -Message-Authenticator=<MA>

   -Janet Custom Attribute

   -SAML assertion

EAP/TTLS GSS

**Scenario 1**

In this chapter we will describe how an connection is made using gss-server and gss-client according to the Janet online how-to (<https://community.ja.net/groups/moonshot/article/moonshot-pilot-release-1-dvd>). We will check if this solution uses proven technologies like GSS and EAP-TTLS to make secure authentication possible.

Test scenario 1:

Janet moonshot pilot DVD

Hostname: debian

Used commands:

gss-server -verbose gss@debian (in pts/0)

gss-client -mech ‘{1.3.6.1.5.5.15.1.1.18}’ debian gss@debian “message” (in pts/1)

freeradius -X (in pts/2, freeradius must be halted if this isn’t the case you can execute /etc/init.d/freeradius stop)

Resulted output:

pts/0:

root@debian:~# gss-server -verbose gss@debian

starting...

reading token flags: 0 bytes read

reading token flags: 0 bytes read

reading token flags: 0 bytes read

Received token (size=33):

60 1f 06 09 2b 06 01 05 05 0f 01 01 12 06 01 00

00 00 02 00 00 00 0a 67 73 73 2f 64 65 62 69 61

6e

Sending accept\_sec\_context token (size=46):

60 2c 06 09 2b 06 01 05 05 0f 01 01 12 06 02 00

00 00 03 00 00 00 0a 67 73 73 2f 64 65 62 69 61

6e 80 00 00 05 00 00 00 05 01 00 00 05 01

continue needed...

Received token (size=34):

60 20 06 09 2b 06 01 05 05 0f 01 01 12 06 01 80

00 00 04 00 00 00 0b 02 00 00 0b 01 40 6c 6f 63

61 6c

Sending accept\_sec\_context token (size=29):

60 1b 06 09 2b 06 01 05 05 0f 01 01 12 06 02 80

00 00 05 00 00 00 06 01 01 00 06 15 20

continue needed...

Received token (size=85):

60 53 06 09 2b 06 01 05 05 0f 01 01 12 06 01 80

00 00 04 00 00 00 3e 02 01 00 3e 15 00 16 03 01

00 33 01 00 00 2f 03 01 51 5c 9a e7 7a 45 e2 64

73 ac cf bb 03 f3 bf 12 15 a7 55 2b 5c 73 11 12

da 87 b5 e6 6f 62 d0 a5 00 00 08 00 2f 00 0a 00

05 00 04 01 00

Sending accept\_sec\_context token (size=830):

60 82 03 3a 06 09 2b 06 01 05 05 0f 01 01 12 06

02 80 00 00 05 00 00 03 25 01 02 03 25 15 80 00

00 03 1b 16 03 01 00 2a 02 00 00 26 03 01 51 5c

9a e7 3c b9 26 9c 67 3c c1 71 51 b5 ee ed ee 5e

ae 31 fe e8 19 94 25 f6 ee 7b d6 f6 84 08 00 00

2f 00 16 03 01 02 de 0b 00 02 da 00 02 d7 00 02

d4 30 82 02 d0 30 82 01 b8 a0 03 02 01 02 02 09

00 fa 6d 53 3f 4d 87 a8 fb 30 0d 06 09 2a 86 48

86 f7 0d 01 01 05 05 00 30 20 31 1e 30 1c 06 03

55 04 03 13 15 6c 6f 63 61 6c 68 6f 73 74 2e 6c

6f 63 61 6c 64 6f 6d 61 69 6e 30 1e 17 0d 31 33

30 33 30 37 32 32 33 38 35 33 5a 17 0d 32 33 30

33 30 35 32 32 33 38 35 33 5a 30 20 31 1e 30 1c

06 03 55 04 03 13 15 6c 6f 63 61 6c 68 6f 73 74

2e 6c 6f 63 61 6c 64 6f 6d 61 69 6e 30 82 01 22

30 0d 06 09 2a 86 48 86 f7 0d 01 01 01 05 00 03

82 01 0f 00 30 82 01 0a 02 82 01 01 00 c0 d7 ef

09 ef 97 1a df b2 f7 bb e7 36 eb 04 56 28 dd df

be 91 47 75 37 ff d1 52 76 cc 50 5e 4b af f8 eb

aa b6 aa 63 c8 6a f5 92 49 48 b0 97 29 45 63 21

4e 3e d9 8e 8a 37 b3 c3 a5 9c 4b 21 56 7a 86 5b

f0 fb 59 c0 23 63 95 09 91 f8 86 e8 ad d9 12 33

9e c1 a6 b0 f8 33 8f 71 a1 8b d6 ee 8a 4e 94 8b

c2 80 81 88 e3 a9 72 18 b6 8e 72 99 60 b0 8b 45

c6 8d ed 7f 91 f7 78 02 1a 90 9a ea 84 41 24 55

f3 06 9f ef cf 55 46 31 83 c6 45 d6 90 7b e6 e8

1e ad fb ee 7c d6 91 b3 f0 70 60 9c 22 7f 9c 8e

d2 a0 6a 07 f3 28 4a 1b 57 1a 59 83 c0 d5 62 45

a0 d7 f7 b3 ad 5d e9 50 a2 af b6 6f 36 54 29 19

49 cf 55 8d f8 72 ed 66 20 a6 34 f0 01 ab f6 04

5a 3c 33 5b ed e1 60 2d 1b bd 7d dc 1f 5c b3 23

6c 0e 80 6d 8a 7c 82 d1 f0 a0 9c e0 8c cf 69 f0

cd d9 38 18 0e 51 1c ca 50 db cb b7 c9 02 03 01

00 01 a3 0d 30 0b 30 09 06 03 55 1d 13 04 02 30

00 30 0d 06 09 2a 86 48 86 f7 0d 01 01 05 05 00

03 82 01 01 00 5e b1 53 fb fc 94 23 9a 2b 5e eb

80 38 49 48 b1 da 54 71 85 f5 8a 98 2f 10 ca c6

07 32 a8 12 23 82 0d ed 2c 8e 79 93 bd ef c5 16

d4 cc 61 66 08 61 33 7a ac c9 38 90 7e c3 89 0f

bd 59 57 20 1f 8f 6d 6e f2 46 34 72 d1 44 0f 24

08 32 af 8b 43 17 8e 38 66 b3 9f cb 79 be 79 2f

96 41 e7 e1 47 8c da fc 48 de 51 0a 24 eb 52 01

9e f4 8b c4 9d c2 7c 50 67 f1 59 84 c7 57 35 9a

c8 3c bc 19 9a 54 9b 70 b3 5c d9 17 4b e0 41 12

c4 1b da c8 d8 6a eb 5b da ee 29 65 5b a8 88 f9

d9 b7 63 47 1e 7d 14 3c bf 32 77 4b 18 2a 20 24

ea 95 6f 24 ef 3a 61 12 7d 1b 29 e2 8b b8 bf 8d

a1 9c ca e5 4b 89 0d f9 63 9e 98 83 8b 2d f0 df

71 9f 37 57 23 4a fc 0f 23 5f 13 1e 32 55 69 d6

1b 27 36 ec a6 35 fc f9 d2 1a 5e dc 04 9a 29 da

25 90 8f c7 01 d8 04 40 97 0c 3f 7a 7c 7f 72 cb

9f 62 f3 03 02 16 03 01 00 04 0e 00 00 00

continue needed...

Received token (size=357):

60 82 01 61 06 09 2b 06 01 05 05 0f 01 01 12 06

01 80 00 00 04 00 00 01 4c 02 02 01 4c 15 00 16

03 01 01 06 10 00 01 02 01 00 8b 1b 9b ff 9b d2

f0 9b a8 4c de de 62 4f 58 b3 03 ce 18 a6 6d 4b

f6 c9 61 96 57 85 dd ee 48 a9 3f 6f aa d8 d4 73

32 d4 8a 3c cb fd 76 83 6d d7 ca 4d a7 7a 1d 3d

fb 93 41 fe 95 f9 c8 82 47 e5 4c 0b cf 67 73 1d

47 d5 0e 7f 41 88 f8 d3 91 53 01 33 6f 37 81 e3

5b 07 67 f3 56 e3 2f 0e e6 e6 c8 bb c3 51 90 65

ee 16 56 f2 d2 9f 18 11 83 26 9d 49 9b 60 dc 84

e1 e4 bd a5 0e 69 ae 92 e9 60 da 04 eb b0 f1 9d

82 11 e9 d1 52 58 8c a3 7b d7 19 96 b5 a0 e1 ea

55 29 27 a3 35 92 ad a1 54 84 0f d1 83 88 ba 23

c1 ae a4 a1 e9 9d b6 9f c5 d9 90 b5 c8 e4 4a 70

46 4b ed c4 9d 87 2a af a9 26 30 25 19 22 25 06

f0 06 66 b8 de f3 66 a0 78 31 c7 72 5f ae 5b 91

c5 ef b8 d0 a9 ea 76 e3 69 c9 7e e6 03 e3 8b 2e

ec 2f 32 16 50 13 e2 94 d3 fd 4b a2 ee 9b ac 80

e0 d6 19 19 8d d0 46 e1 ab f9 14 03 01 00 01 01

16 03 01 00 30 29 72 8c 61 6a 94 5f 37 20 99 ef

73 83 8c 81 6f d3 2e 87 73 3b 2e 59 d3 2d f8 8a

84 9b ab b5 82 ee 72 9b 2f 2c 14 4b 9e d3 01 05

33 0d 25 22 20

Sending accept\_sec\_context token (size=92):

60 5a 06 09 2b 06 01 05 05 0f 01 01 12 06 02 80

00 00 05 00 00 00 45 01 03 00 45 15 80 00 00 00

3b 14 03 01 00 01 01 16 03 01 00 30 2b 2d d9 e9

e8 5d e6 45 1c 1a 7a 53 d1 ce ee 41 10 40 c7 8c

bf 0f f3 d6 d7 1b 1c 3d cf 11 af 43 15 86 e5 dc

ad 75 2a 8f e6 d4 cd 12 b3 cb 1a 62

continue needed...

Received token (size=82):

60 50 06 09 2b 06 01 05 05 0f 01 01 12 06 01 80

00 00 04 00 00 00 3b 02 03 00 3b 15 00 17 03 01

00 30 5e af 17 35 5b 2f 6d 9c 37 ea 44 ef 15 c0

2f 7e 2c d3 d3 24 74 d8 46 33 86 25 dd 76 79 49

4a 6e c4 f2 63 21 0a ee 86 64 fd ee dc 1e 2a 84

8d 53

Sending accept\_sec\_context token (size=118):

60 74 06 09 2b 06 01 05 05 0f 01 01 12 06 02 80

00 00 05 00 00 00 5f 01 04 00 5f 15 80 00 00 00

55 17 03 01 00 50 ea a4 1d 0a 70 ed ee a7 b6 5b

9c 09 19 d0 82 b8 84 b6 14 cb 39 6a 51 38 7c 3d

b7 10 e3 72 de 89 a3 fd 07 e9 af c2 c3 a7 c7 c0

7c b5 c6 5a 30 1e f6 d1 6b f6 25 d8 7d 63 a9 3d

69 ae d5 e1 54 fa b2 5f d5 7f 46 cf 03 f1 1f 33

9e b6 2c 98 7d 93

continue needed...

Received token (size=147):

60 81 90 06 09 2b 06 01 05 05 0f 01 01 12 06 01

80 00 00 04 00 00 00 7b 02 04 00 7b 15 00 17 03

01 00 70 ad 08 a5 07 8d ee 19 ff 08 fa d4 f0 31

3f b1 b1 a6 46 c7 94 69 29 34 0b e2 ab 9e a9 52

5b d5 b6 18 7f 11 18 09 97 37 99 5a d2 57 9c 63

02 07 a8 62 70 cf 66 96 a0 5f a1 3d 33 6e 7f 58

30 10 0d 74 c4 ab e3 cf c1 46 31 1b 6d da 3f 82

39 d6 8f e1 a1 2f bd 40 39 91 a9 82 54 2c eb be

f3 ed 2a 7e 78 af 64 22 97 a5 73 6e c6 d4 1b a4

bd de 3e

Sending accept\_sec\_context token (size=135):

60 81 84 06 09 2b 06 01 05 05 0f 01 01 12 06 02

80 00 00 05 00 00 00 6f 01 05 00 6f 15 80 00 00

00 65 17 03 01 00 60 db af d0 c6 c5 63 ee 70 d0

eb aa 33 78 8e 96 86 58 2a a0 28 72 1c 6e 2c 15

a6 f5 58 8f 12 21 59 fd 5e 85 4e 80 e8 e1 dc d4

05 1a 69 8a 21 70 5a f1 68 88 d2 6a 76 75 0b 3c

10 76 20 42 42 8b d8 d9 02 85 0a 88 68 93 6d a9

48 cc 69 9e cf b6 d3 41 90 4d 5c 9c 1e 3a 1e a0

d5 35 45 8d 09 c6 a0

continue needed...

Received token (size=82):

60 50 06 09 2b 06 01 05 05 0f 01 01 12 06 01 80

00 00 04 00 00 00 3b 02 05 00 3b 15 00 17 03 01

00 30 c5 a6 fa c3 c8 af 2e 35 d7 ec 7e 7c 35 76

a5 75 28 49 33 02 ab 99 71 18 ec f7 37 50 18 44

74 58 58 66 25 3a f1 12 d1 24 ae c6 c1 2d d8 fa

83 cf

2013-04-03 23:11:03 WARN Shibboleth.Application : empty/missing cookieProps setting, set to "https" for SSL/TLS-only usage

2013-04-03 23:11:03 WARN Shibboleth.Application : handlerSSL should be enabled for SSL/TLS-enabled web sites

2013-04-03 23:11:03 WARN Shibboleth.Application : no MetadataProvider available, configure at least one for standard SSO usage

2013-04-03 23:11:03 WARN Shibboleth.AttributeExtractor.XML : attribute mappings are reloadable; be sure to restart web server when adding new attribute IDs

2013-04-03 23:11:03 WARN Shibboleth.AttributeFilter : removed value at position (0) of attribute (eppn) from (unknown source)

2013-04-03 23:11:03 WARN Shibboleth.AttributeFilter : no values left, removing attribute (eppn) from (unknown source)

Sending accept\_sec\_context token (size=27):

60 19 06 09 2b 06 01 05 05 0f 01 01 12 06 02 80

00 00 05 00 00 00 04 03 05 00 04

continue needed...

Received token (size=47):

60 2d 06 09 2b 06 01 05 05 0f 01 01 12 06 01 00

00 00 0c 00 00 00 04 00 00 00 02 80 00 00 0d 00

00 00 0c cc 78 0a e0 cf b3 ae 2f d8 94 0a 88

Sending accept\_sec\_context token (size=53):

60 33 06 09 2b 06 01 05 05 0f 01 01 12 06 02 00

00 00 03 00 00 00 0a 67 73 73 2f 64 65 62 69 61

6e 80 00 00 0e 00 00 00 0c bb 0a 82 db b9 2a d5

4e 2d 75 b1 56

context flag: GSS\_C\_MUTUAL\_FLAG

context flag: GSS\_C\_REPLAY\_FLAG

context flag: GSS\_C\_SEQUENCE\_FLAG

context flag: GSS\_C\_CONF\_FLAG

context flag: GSS\_C\_INTEG\_FLAG

Accepted connection using mechanism OID { 1 3 6 1 5 5 15 1 1 18 }.

Attribute urn:ietf:params:gss:radius-attribute 1 Authenticated Complete

steve@local

7374657665406c6f63616c

Attribute urn:ietf:params:gss:radius-attribute 79 Authenticated Complete

03050004

Attribute urn:ietf:params:gss:radius-attribute 80 Authenticated Complete

1983980fd722a2d1ae6705ba40128f7f

Attribute urn:ietf:params:gss:federated-saml-assertion Authenticated Complete

<saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="foo" IssueInstant="2011-03-19T08:30:00Z" Version="2.0"><saml:Issuer>urn:mace:incommon:osu.edu</saml:Issuer><saml:AttributeStatement><saml:Attribute Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"><saml:AttributeValue>cantor.2@osu.edu</saml:AttributeValue></saml:Attribute><saml:Attribute Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"><saml:AttributeValue>moonshot</saml:AttributeValue></saml:Attribute></saml:AttributeStatement></saml:Assertion>

3c73616d6c3a417373657274696f6e20786d6c6e733a73616d6c3d2275726e3a

6f617369733a6e616d65733a74633a53414d4c3a322e303a617373657274696f

6e222049443d22666f6f22204973737565496e7374616e743d22323031312d30

332d31395430383a33303a30305a222056657273696f6e3d22322e30223e3c73

616d6c3a4973737565723e75726e3a6d6163653a696e636f6d6d6f6e3a6f7375

2e6564753c2f73616d6c3a4973737565723e3c73616d6c3a4174747269627574

6553746174656d656e743e3c73616d6c3a417474726962757465204e616d653d

2275726e3a6f69643a312e332e362e312e342e312e353932332e312e312e312e

3622204e616d65466f726d61743d2275726e3a6f617369733a6e616d65733a74

633a53414d4c3a322e303a617474726e616d652d666f726d61743a757269223e

3c73616d6c3a41747472696275746556616c75653e63616e746f722e32406f73

752e6564753c2f73616d6c3a41747472696275746556616c75653e3c2f73616d

6c3a4174747269627574653e3c73616d6c3a417474726962757465204e616d65

3d2275726e3a6f69643a312e332e362e312e342e312e353932332e312e312e31

2e3722204e616d65466f726d61743d2275726e3a6f617369733a6e616d65733a

74633a53414d4c3a322e303a617474726e616d652d666f726d61743a75726922

3e3c73616d6c3a41747472696275746556616c75653e6d6f6f6e73686f743c2f

73616d6c3a41747472696275746556616c75653e3c2f73616d6c3a4174747269

627574653e3c2f73616d6c3a41747472696275746553746174656d656e743e3c

2f73616d6c3a417373657274696f6e3e

Attribute urn:ietf:params:gss:federated-saml-attribute urn:oasis:names:tc:SAML:2.0:attrname-format:uri urn:oid:1.3.6.1.4.1.5923.1.1.1.6 Authenticated Complete

cantor.2@osu.edu

63616e746f722e32406f73752e656475

Attribute urn:ietf:params:gss:federated-saml-attribute urn:oasis:names:tc:SAML:2.0:attrname-format:uri urn:oid:1.3.6.1.4.1.5923.1.1.1.7 Authenticated Complete

moonshot

6d6f6f6e73686f74

Attribute local-login-user Authenticated Complete

moonshot

6d6f6f6e73686f74

localname: moonshot

Accepted connection: "steve@local"

Message token (flags=228):

05 04 02 ff 00 00 00 00 00 00 00 00 00 00 00 00

18 0c 77 3e dd 1e 50 26 b7 b4 a3 75 7c 3a 0a 62

1a 7a a8 24 3c 2f 39 3f 26 bd af e3 8d ac 3e 95

94 3b 02 45 ca db 2b 22 bf 7f 1d 0d 4c 66 ae a1

18 38 dc

Received message: "message"

NOOP token

pts/1:

user@debian:~$ gss-client -mech '{1.3.6.1.5.5.15.1.1.18}' debian gss@debian "message"

Sending init\_sec\_context token (size=33)...continue needed...

CTRL-EVENT-EAP-STARTED EAP authentication started

Sending init\_sec\_context token (size=34)...continue needed...

CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=21

CTRL-EVENT-EAP-METHOD EAP vendor 0 method 21 (TTLS) selected

Sending init\_sec\_context token (size=85)...continue needed...

Sending init\_sec\_context token (size=357)...continue needed...

Sending init\_sec\_context token (size=82)...continue needed...

Sending init\_sec\_context token (size=147)...continue needed...

EAP-MSCHAPV2: Authentication succeeded

Sending init\_sec\_context token (size=82)...continue needed...

CTRL-EVENT-EAP-SUCCESS EAP authentication completed successfully

Sending init\_sec\_context token (size=47)...continue needed...

context flag: GSS\_C\_MUTUAL\_FLAG

context flag: GSS\_C\_REPLAY\_FLAG

context flag: GSS\_C\_SEQUENCE\_FLAG

context flag: GSS\_C\_CONF\_FLAG

context flag: GSS\_C\_INTEG\_FLAG

"steve@local" to "gss/debian", lifetime -1, flags 13e, locally initiated, open

Name type of source name is { 1 2 840 113554 1 2 1 1 }.

Mechanism { 1 3 6 1 5 5 15 1 1 18 } supports 5 names

 0: { 1 2 840 113554 1 2 1 1 }

 1: { 1 2 840 113554 1 2 1 4 }

 2: { 1 3 6 1 5 6 4 }

 3: { 1 3 6 1 5 5 15 2 1 }

 4: { 1 3 6 1 5 6 3 }

Signature verified.

pts/2:

Some information in the beginning is omitted because it only loads configurations and modules.

Ready to process requests.

rad\_recv: Access-Request packet from host 127.0.0.1 port 50993, id=0, length=72

User-Name = "@local"

Attr-164 = 0x677373

Attr-165 = 0x64656269616e

EAP-Message = 0x0200000b01406c6f63616c

Message-Authenticator = 0x2f39437ea249714480431df489e8a975

# Executing section authorize from file /etc/freeradius/sites-enabled/default

+- entering group authorize {...}

++[preprocess] returns ok

++[chap] returns noop

++[mschap] returns noop

++[digest] returns noop

[suffix] Looking up realm "local" for User-Name = "@local"

[suffix] Found realm "LOCAL"

[suffix] Adding Stripped-User-Name = ""

[suffix] Adding Realm = "LOCAL"

[suffix] Authentication realm is LOCAL.

++[suffix] returns ok

[eap] EAP packet type response id 0 length 11

[eap] No EAP Start, assuming it's an on-going EAP conversation

++[eap] returns updated

++[files] returns noop

++[expiration] returns noop

++[logintime] returns noop

[pap] WARNING! No "known good" password found for the user.  Authentication may fail because of this.

++[pap] returns noop

Found Auth-Type = EAP

# Executing group from file /etc/freeradius/sites-enabled/default

+- entering group authenticate {...}

[eap] EAP Identity

[eap] processing type tls

[tls] Initiate

[tls] Start returned 1

++[eap] returns handled

Sending Access-Challenge of id 0 to 127.0.0.1 port 50993

EAP-Message = 0x010100061520

Message-Authenticator = 0x00000000000000000000000000000000

State = 0xdaa826a7daa9330c1c88d9b715c71e63

Finished request 0.

Going to the next request

Waking up in 4.9 seconds.

rad\_recv: Access-Request packet from host 127.0.0.1 port 50993, id=0, length=141

Cleaning up request 0 ID 0 with timestamp +175

User-Name = "@local"

Attr-164 = 0x677373

Attr-165 = 0x64656269616e

EAP-Message = 0x0201003e150016030100330100002f0301515c9ae77a45e26473accfbb03f3bf1215a7552b5c731112da87b5e66f62d0a5000008002f000a000500040100

State = 0xdaa826a7daa9330c1c88d9b715c71e63

Message-Authenticator = 0xaab6e4f157c06b7e4f5195d731b56cce

# Executing section authorize from file /etc/freeradius/sites-enabled/default

+- entering group authorize {...}

++[preprocess] returns ok

++[chap] returns noop

++[mschap] returns noop

++[digest] returns noop

[suffix] Looking up realm "local" for User-Name = "@local"

[suffix] Found realm "LOCAL"

[suffix] Adding Stripped-User-Name = ""

[suffix] Adding Realm = "LOCAL"

[suffix] Authentication realm is LOCAL.

++[suffix] returns ok

[eap] EAP packet type response id 1 length 62

[eap] Continuing tunnel setup.

++[eap] returns ok

Found Auth-Type = EAP

# Executing group from file /etc/freeradius/sites-enabled/default

+- entering group authenticate {...}

[eap] Request found, released from the list

[eap] EAP/ttls

[eap] processing type ttls

[ttls] Authenticate

[ttls] processing EAP-TLS

[ttls] eaptls\_verify returned 7

[ttls] Done initial handshake

[ttls]     (other): before/accept initialization

[ttls]     TLS\_accept: before/accept initialization

[ttls] <<< TLS 1.0 Handshake [length 0033], ClientHello

[ttls]     TLS\_accept: SSLv3 read client hello A

[ttls] >>> TLS 1.0 Handshake [length 002a], ServerHello

[ttls]     TLS\_accept: SSLv3 write server hello A

[ttls] >>> TLS 1.0 Handshake [length 02de], Certificate

[ttls]     TLS\_accept: SSLv3 write certificate A

[ttls] >>> TLS 1.0 Handshake [length 0004], ServerHelloDone

[ttls]     TLS\_accept: SSLv3 write server done A

[ttls]     TLS\_accept: SSLv3 flush data

[ttls]     TLS\_accept: Need to read more data: SSLv3 read client certificate A

In SSL Handshake Phase

In SSL Accept mode

[ttls] eaptls\_process returned 13

++[eap] returns handled

Sending Access-Challenge of id 0 to 127.0.0.1 port 50993

EAP-Message = 

EAP-Message = 

EAP-Message = 

EAP-Message = 0x1b2736eca635fcf9d21a5edc049a29da25908fc701d80440970c3f7a7c7f72cb9f62f3030216030100040e000000

Message-Authenticator = 0x00000000000000000000000000000000

State = 0xdaa826a7dbaa330c1c88d9b715c71e63

Finished request 1.

Going to the next request

Waking up in 4.9 seconds.

rad\_recv: Access-Request packet from host 127.0.0.1 port 50993, id=0, length=413

Cleaning up request 1 ID 0 with timestamp +175

User-Name = "@local"

Attr-164 = 0x677373

Attr-165 = 0x64656269616e

EAP-Message = 

EAP-Message = 0xe294d3fd4ba2ee9bac80e0d619198dd046e1abf9140301000101160301003029728c616a945f372099ef73838c816fd32e87733b2e59d32df88a849babb582ee729b2f2c144b9ed30105330d252220

State = 0xdaa826a7dbaa330c1c88d9b715c71e63

Message-Authenticator = 0x28f95b320e305a92ef49493d9a7ca59e

# Executing section authorize from file /etc/freeradius/sites-enabled/default

+- entering group authorize {...}

++[preprocess] returns ok

++[chap] returns noop

++[mschap] returns noop

++[digest] returns noop

[suffix] Looking up realm "local" for User-Name = "@local"

[suffix] Found realm "LOCAL"

[suffix] Adding Stripped-User-Name = ""

[suffix] Adding Realm = "LOCAL"

[suffix] Authentication realm is LOCAL.

++[suffix] returns ok

[eap] EAP packet type response id 2 length 253

[eap] Continuing tunnel setup.

++[eap] returns ok

Found Auth-Type = EAP

# Executing group from file /etc/freeradius/sites-enabled/default

+- entering group authenticate {...}

[eap] Request found, released from the list

[eap] EAP/ttls

[eap] processing type ttls

[ttls] Authenticate

[ttls] processing EAP-TLS

[ttls] eaptls\_verify returned 7

[ttls] Done initial handshake

[ttls] <<< TLS 1.0 Handshake [length 0106], ClientKeyExchange

[ttls]     TLS\_accept: SSLv3 read client key exchange A

[ttls] <<< TLS 1.0 ChangeCipherSpec [length 0001]

[ttls] <<< TLS 1.0 Handshake [length 0010], Finished

[ttls]     TLS\_accept: SSLv3 read finished A

[ttls] >>> TLS 1.0 ChangeCipherSpec [length 0001]

[ttls]     TLS\_accept: SSLv3 write change cipher spec A

[ttls] >>> TLS 1.0 Handshake [length 0010], Finished

[ttls]     TLS\_accept: SSLv3 write finished A

[ttls]     TLS\_accept: SSLv3 flush data

[ttls]     (other): SSL negotiation finished successfully

SSL Connection Established

[ttls] eaptls\_process returned 13

++[eap] returns handled

Sending Access-Challenge of id 0 to 127.0.0.1 port 50993

EAP-Message = 0x0103004515800000003b14030100010116030100302b2dd9e9e85de6451c1a7a53d1ceee411040c78cbf0ff3d6d71b1c3dcf11af431586e5dcad752a8fe6d4cd12b3cb1a62

Message-Authenticator = 0x00000000000000000000000000000000

State = 0xdaa826a7d8ab330c1c88d9b715c71e63

Finished request 2.

Going to the next request

Waking up in 4.9 seconds.

rad\_recv: Access-Request packet from host 127.0.0.1 port 50993, id=0, length=138

Cleaning up request 2 ID 0 with timestamp +175

User-Name = "@local"

Attr-164 = 0x677373

Attr-165 = 0x64656269616e

EAP-Message = 0x0203003b150017030100305eaf17355b2f6d9c37ea44ef15c02f7e2cd3d32474d846338625dd7679494a6ec4f263210aee8664fdeedc1e2a848d53

State = 0xdaa826a7d8ab330c1c88d9b715c71e63

Message-Authenticator = 0x4f97d56aae2f68c6d93daf88c49b3ba4

# Executing section authorize from file /etc/freeradius/sites-enabled/default

+- entering group authorize {...}

++[preprocess] returns ok

++[chap] returns noop

++[mschap] returns noop

++[digest] returns noop

[suffix] Looking up realm "local" for User-Name = "@local"

[suffix] Found realm "LOCAL"

[suffix] Adding Stripped-User-Name = ""

[suffix] Adding Realm = "LOCAL"

[suffix] Authentication realm is LOCAL.

++[suffix] returns ok

[eap] EAP packet type response id 3 length 59

[eap] Continuing tunnel setup.

++[eap] returns ok

Found Auth-Type = EAP

# Executing group from file /etc/freeradius/sites-enabled/default

+- entering group authenticate {...}

[eap] Request found, released from the list

[eap] EAP/ttls

[eap] processing type ttls

[ttls] Authenticate

[ttls] processing EAP-TLS

[ttls] eaptls\_verify returned 7

[ttls] Done initial handshake

[ttls] eaptls\_process returned 7

[ttls] Session established.  Proceeding to decode tunneled attributes.

[ttls] Got tunneled request

EAP-Message = 0x02000010017374657665406c6f63616c

FreeRADIUS-Proxied-To = 127.0.0.1

[ttls] Got tunneled identity of steve@local

[ttls] Setting default EAP type for tunneled EAP session.

[ttls] Sending tunneled request

EAP-Message = 0x02000010017374657665406c6f63616c

FreeRADIUS-Proxied-To = 127.0.0.1

User-Name = "steve@local"

server inner-tunnel {

# Executing section authorize from file /etc/freeradius/sites-enabled/inner-tunnel

+- entering group authorize {...}

++[chap] returns noop

++[mschap] returns noop

[suffix] Looking up realm "local" for User-Name = "steve@local"

[suffix] Found realm "LOCAL"

[suffix] Adding Stripped-User-Name = "steve"

[suffix] Adding Realm = "LOCAL"

[suffix] Authentication realm is LOCAL.

++[suffix] returns ok

++[control] returns ok

[eap] EAP packet type response id 0 length 16

[eap] No EAP Start, assuming it's an on-going EAP conversation

++[eap] returns updated

[files] users: Matched entry steve at line 76

++[files] returns ok

++[expiration] returns noop

++[logintime] returns noop

[pap] WARNING: Auth-Type already set.  Not setting to PAP

++[pap] returns noop

Found Auth-Type = EAP

# Executing group from file /etc/freeradius/sites-enabled/inner-tunnel

+- entering group authenticate {...}

[eap] EAP Identity

[eap] processing type mschapv2

rlm\_eap\_mschapv2: Issuing Challenge

++[eap] returns handled

} # server inner-tunnel

[ttls] Got tunneled reply code 11

EAP-Message = 0x010100251a0101002010f45ff541d2777b64cad69e1f5c741dd77374657665406c6f63616c

Message-Authenticator = 0x00000000000000000000000000000000

State = 0x42e0e63342e1fcbfb3b798393b59f1fa

[ttls] Got tunneled Access-Challenge

++[eap] returns handled

Sending Access-Challenge of id 0 to 127.0.0.1 port 50993

EAP-Message = 0x0104005f1580000000551703010050eaa41d0a70edeea7b65b9c0919d082b884b614cb396a51387c3db710e372de89a3fd07e9afc2c3a7c7c07cb5c65a301ef6d16bf625d87d63a93d69aed5e154fab25fd57f46cf03f11f339eb62c987d93

Message-Authenticator = 0x00000000000000000000000000000000

State = 0xdaa826a7d9ac330c1c88d9b715c71e63

Finished request 3.

Going to the next request

Waking up in 4.9 seconds.

rad\_recv: Access-Request packet from host 127.0.0.1 port 50993, id=0, length=202

Cleaning up request 3 ID 0 with timestamp +175

User-Name = "@local"

Attr-164 = 0x677373

Attr-165 = 0x64656269616e

EAP-Message = 0x0204007b15001703010070ad08a5078dee19ff08fad4f0313fb1b1a646c7946929340be2ab9ea9525bd5b6187f1118099737995ad2579c630207a86270cf6696a05fa13d336e7f5830100d74c4abe3cfc146311b6dda3f8239d68fe1a12fbd403991a982542cebbef3ed2a7e78af642297a5736ec6d41ba4bdde3e

State = 0xdaa826a7d9ac330c1c88d9b715c71e63

Message-Authenticator = 0x9e4b6300bb2594b50cebef3bb343ad23

# Executing section authorize from file /etc/freeradius/sites-enabled/default

+- entering group authorize {...}

++[preprocess] returns ok

++[chap] returns noop

++[mschap] returns noop

++[digest] returns noop

[suffix] Looking up realm "local" for User-Name = "@local"

[suffix] Found realm "LOCAL"

[suffix] Adding Stripped-User-Name = ""

[suffix] Adding Realm = "LOCAL"

[suffix] Authentication realm is LOCAL.

++[suffix] returns ok

[eap] EAP packet type response id 4 length 123

[eap] Continuing tunnel setup.

++[eap] returns ok

Found Auth-Type = EAP

# Executing group from file /etc/freeradius/sites-enabled/default

+- entering group authenticate {...}

[eap] Request found, released from the list

[eap] EAP/ttls

[eap] processing type ttls

[ttls] Authenticate

[ttls] processing EAP-TLS

[ttls] eaptls\_verify returned 7

[ttls] Done initial handshake

[ttls] eaptls\_process returned 7

[ttls] Session established.  Proceeding to decode tunneled attributes.

[ttls] Got tunneled request

EAP-Message = 0x020100461a020100413199989e21adc4b4a807caa1b55054f3cd000000000000000085ab820cef865e745a1c809bbd96ef955dad57c2d6a9146f007374657665406c6f63616c

FreeRADIUS-Proxied-To = 127.0.0.1

[ttls] Sending tunneled request

EAP-Message = 0x020100461a020100413199989e21adc4b4a807caa1b55054f3cd000000000000000085ab820cef865e745a1c809bbd96ef955dad57c2d6a9146f007374657665406c6f63616c

FreeRADIUS-Proxied-To = 127.0.0.1

User-Name = "steve@local"

State = 0x42e0e63342e1fcbfb3b798393b59f1fa

server inner-tunnel {

# Executing section authorize from file /etc/freeradius/sites-enabled/inner-tunnel

+- entering group authorize {...}

++[chap] returns noop

++[mschap] returns noop

[suffix] Looking up realm "local" for User-Name = "steve@local"

[suffix] Found realm "LOCAL"

[suffix] Adding Stripped-User-Name = "steve"

[suffix] Adding Realm = "LOCAL"

[suffix] Authentication realm is LOCAL.

++[suffix] returns ok

++[control] returns ok

[eap] EAP packet type response id 1 length 70

[eap] No EAP Start, assuming it's an on-going EAP conversation

++[eap] returns updated

[files] users: Matched entry steve at line 76

++[files] returns ok

++[expiration] returns noop

++[logintime] returns noop

[pap] WARNING: Auth-Type already set.  Not setting to PAP

++[pap] returns noop

Found Auth-Type = EAP

# Executing group from file /etc/freeradius/sites-enabled/inner-tunnel

+- entering group authenticate {...}

[eap] Request found, released from the list

[eap] EAP/mschapv2

[eap] processing type mschapv2

[mschapv2] # Executing group from file /etc/freeradius/sites-enabled/inner-tunnel

[mschapv2] +- entering group MS-CHAP {...}

[mschap] Creating challenge hash with username: steve@local

[mschap] Told to do MS-CHAPv2 for steve@local with NT-Password

[mschap] adding MS-CHAPv2 MPPE keys

++[mschap] returns ok

MSCHAP Success

++[eap] returns handled

} # server inner-tunnel

[ttls] Got tunneled reply code 11

EAP-Message = 0x010200331a0301002e533d31444334463633453232303745394530454543464144363944384241413938463842453639353439

Message-Authenticator = 0x00000000000000000000000000000000

State = 0x42e0e63343e2fcbfb3b798393b59f1fa

[ttls] Got tunneled Access-Challenge

++[eap] returns handled

Sending Access-Challenge of id 0 to 127.0.0.1 port 50993

EAP-Message = 0x0105006f1580000000651703010060dbafd0c6c563ee70d0ebaa33788e9686582aa028721c6e2c15a6f5588f122159fd5e854e80e8e1dcd4051a698a21705af16888d26a76750b3c10762042428bd8d902850a8868936da948cc699ecfb6d341904d5c9c1e3a1ea0d535458d09c6a0

Message-Authenticator = 0x00000000000000000000000000000000

State = 0xdaa826a7dead330c1c88d9b715c71e63

Finished request 4.

Going to the next request

Waking up in 4.9 seconds.

rad\_recv: Access-Request packet from host 127.0.0.1 port 50993, id=0, length=138

Cleaning up request 4 ID 0 with timestamp +175

User-Name = "@local"

Attr-164 = 0x677373

Attr-165 = 0x64656269616e

EAP-Message = 0x0205003b15001703010030c5a6fac3c8af2e35d7ec7e7c3576a57528493302ab997118ecf73750184474585866253af112d124aec6c12dd8fa83cf

State = 0xdaa826a7dead330c1c88d9b715c71e63

Message-Authenticator = 0x51b3ecc5b084be5d06a75fe81044e63f

# Executing section authorize from file /etc/freeradius/sites-enabled/default

+- entering group authorize {...}

++[preprocess] returns ok

++[chap] returns noop

++[mschap] returns noop

++[digest] returns noop

[suffix] Looking up realm "local" for User-Name = "@local"

[suffix] Found realm "LOCAL"

[suffix] Adding Stripped-User-Name = ""

[suffix] Adding Realm = "LOCAL"

[suffix] Authentication realm is LOCAL.

++[suffix] returns ok

[eap] EAP packet type response id 5 length 59

[eap] Continuing tunnel setup.

++[eap] returns ok

Found Auth-Type = EAP

# Executing group from file /etc/freeradius/sites-enabled/default

+- entering group authenticate {...}

[eap] Request found, released from the list

[eap] EAP/ttls

[eap] processing type ttls

[ttls] Authenticate

[ttls] processing EAP-TLS

[ttls] eaptls\_verify returned 7

[ttls] Done initial handshake

[ttls] eaptls\_process returned 7

[ttls] Session established.  Proceeding to decode tunneled attributes.

[ttls] Got tunneled request

EAP-Message = 0x020200061a03

FreeRADIUS-Proxied-To = 127.0.0.1

[ttls] Sending tunneled request

EAP-Message = 0x020200061a03

FreeRADIUS-Proxied-To = 127.0.0.1

User-Name = "steve@local"

State = 0x42e0e63343e2fcbfb3b798393b59f1fa

server inner-tunnel {

# Executing section authorize from file /etc/freeradius/sites-enabled/inner-tunnel

+- entering group authorize {...}

++[chap] returns noop

++[mschap] returns noop

[suffix] Looking up realm "local" for User-Name = "steve@local"

[suffix] Found realm "LOCAL"

[suffix] Adding Stripped-User-Name = "steve"

[suffix] Adding Realm = "LOCAL"

[suffix] Authentication realm is LOCAL.

++[suffix] returns ok

++[control] returns ok

[eap] EAP packet type response id 2 length 6

[eap] No EAP Start, assuming it's an on-going EAP conversation

++[eap] returns updated

[files] users: Matched entry steve at line 76

++[files] returns ok

++[expiration] returns noop

++[logintime] returns noop

[pap] WARNING: Auth-Type already set.  Not setting to PAP

++[pap] returns noop

Found Auth-Type = EAP

# Executing group from file /etc/freeradius/sites-enabled/inner-tunnel

+- entering group authenticate {...}

[eap] Request found, released from the list

[eap] EAP/mschapv2

[eap] processing type mschapv2

[eap] Freeing handler

++[eap] returns ok

# Executing section post-auth from file /etc/freeradius/sites-enabled/inner-tunnel

+- entering group post-auth {...}

expand: %{request:User-Name} -> steve@local

++[outer.reply] returns noop

} # server inner-tunnel

[ttls] Got tunneled reply code 2

MS-MPPE-Encryption-Policy = 0x00000001

MS-MPPE-Encryption-Types = 0x00000006

MS-MPPE-Send-Key = 0x0a022e5b366f9c6921b3af46ca97c899

MS-MPPE-Recv-Key = 0x5d9b835962c153b0a2cc14fea8cbbc46

EAP-Message = 0x03020004

Message-Authenticator = 0x00000000000000000000000000000000

User-Name = "steve"

[ttls] Got tunneled Access-Accept

[eap] Freeing handler

rlm\_eap\_ttls: Freeing handler for user steve@local

++[eap] returns ok

# Executing section post-auth from file /etc/freeradius/sites-enabled/default

+- entering group post-auth {...}

++[exec] returns noop

++[reply] returns noop

Sending Access-Accept of id 0 to 127.0.0.1 port 50993

User-Name = "steve@local"

MS-MPPE-Recv-Key = 0xd1a49dfa9099df6ff96d8837c3326fdb159df9cf6c81ad9d2dcabcdcc1c23015

MS-MPPE-Send-Key = 0x7f5afbac4d72f7e2f5a2891b86157e5e11304469489d93b98c68a952c7ee7989

EAP-Message = 0x03050004

Message-Authenticator = 0x00000000000000000000000000000000

SAML-AAA-Assertion = "<saml:Assertion xmlns:saml=\"urn:oasis:names:tc:SAML:2.0:assertion\" IssueInstant=\"2011-03-19T08:30:00Z\" ID=\"foo\" Version=\"2.0\">"

SAML-AAA-Assertion = "<saml:Issuer>urn:mace:incommon:osu.edu</saml:Issuer>"

SAML-AAA-Assertion = "<saml:AttributeStatement>"

SAML-AAA-Assertion = "<saml:Attribute NameFormat=\"urn:oasis:names:tc:SAML:2.0:attrname-format:uri\" Name=\"urn:oid:1.3.6.1.4.1.5923.1.1.1.6\"><saml:AttributeValue>cantor.2@osu.edu</saml:AttributeValue></saml:Attribute>"

SAML-AAA-Assertion = "<saml:Attribute NameFormat=\"urn:oasis:names:tc:SAML:2.0:attrname-format:uri\" Name=\"urn:oid:1.3.6.1.4.1.5923.1.1.1.7\"><saml:AttributeValue>moonshot</saml:AttributeValue></saml:Attribute>"

SAML-AAA-Assertion = "</saml:AttributeStatement>"

SAML-AAA-Assertion = "</saml:Assertion>"

Finished request 5.

Going to the next request

Waking up in 4.9 seconds.

Cleaning up request 5 ID 0 with timestamp +175

Ready to process requests.

Conclusion:

In the radius output you will see that there is an EAP-TTLS tunnel where all information is securely transported and handled. In the output that radius produces you can deduce that the local realm is first transported without an username. Radius will then check if the realm exists in his configuration and if so, it will set-up an EAP-TTLS connection with the client. When the connection is set-up the username will be send from the client to the radius server and thereon it will authenticate the user.

The radius log shows what information is send in the outer and inner tunnel. In the outer tunnel the realm will only be transported to the, from then, un-authenticated server (no EAP-TTLS connection exists from then). In the inner tunnel, which the radius server can see when the EAP-TTLS tunnel is set-up, the username will be transported.

When the authentication proces is completed a local account will be mapped to the user, named moonshot.

**Scenario 2:**

Janet pilot dvd (virtuele machine)

Root radius server

Radius server moonshot.nl

OpenLDAP server moonshot.nl

Radtest (program to test radius network connectivity)

Changing the configuration on the JaNET pilot made it possible for succesfily authenticating against another radius server in a radius chain. However, it is not possible at the moment to securely authenticate against an radius server in a radius chain. One sidenote, we didn’t make use of the GSS integrated in the JaNET pilot. Further research is needed if a default radius server can be used in a radius chain using GSS technology inside OpenSSH.

When using radtest and when you run all software in debug mode it’s possible to see the user information that wants to login. For security and privacy it’s not desirable that all information is viewable in cleartext.

A possible solution to address this problem is using GSS, further research will provide the necessary information to determine if Radius or OpenSSH needs to be changed in order to accomplish this goal.

De mogelijkheid om je te authenticeren werkt, maar de mapping wordt dan niet automatisch gemaakt (nog beter uit te zoeken). Wel kan er een mapping gemaakt worden door een script te laten draaien die een user automatisch koppelt aan een room-account.

Differences between Janet and FreeRADIUS stable release

This chapter describes.

Janet SSH

The British organisation JaNET is currently working on a Moonshot project themselves. It is of interest to our project team to take a look at their code to see what they have done in order to make it work.

**Difference in code**

JaNET needed to make some changes in the source code of the SSH daemon, in order to make federated login possible. They created some extra files in the source code in order to make it work, and they did make some adjustment to the current source code. In the rest of the chapter we will highlight some of those changes and some of the new files.

We tried to compare the different source codes using the source version of JaNET’s SSH source code and the official OpenSSH 5.9p1 source code from OpenBSD. Therefore we were able to compare the two sources and see what the differences are.

One of the interesting differences in the source code is located in the “auth2-gss.c” file. The file itself exists on every OpenSSH installation, but JaNET did some changes. They added two extra classes of code for GSS key exchange, named: “userauth\_gsskeyex”, “gssapi\_set\_username(Authctxt \*authctxt)”, and a new authentication method: “Authmethod method\_gsskeyex”. The first new part of code is the new userauth mechanism for the GSSAPI key exchange. When a user wants to login, it will eventually return “authenticated”. This piece of code will eventually call the gssapi\_set\_username. This function will set the GSSAPI context for the username and password to the PAM module authctxt, if the option USE\_PAM is enabled.

Another file where some major adjustments have been made is the file “ssh-gss.h”. The original file contains a part for MIT Kerberos. In the version of JaNET’s source code, this part is gone. A lot of new code has been added to the file. This is especially for the SSH / GSSAPI key exchange.

**Interesting files**

In this paragraph is a list with interesting files, which we could find, and of which we think they are interesting:

* auth2-gss.c
  + (Include) monitor\_wrap.h
    - Line 40 deleted
    - Added: OM\_uint32 mm\_ssh\_gssapi\_localname / OM\_uint32 mm\_ssh\_gssapi\_sign
  + (Include) packet.h
    - Line added: “packet\_connection\_is\_ipv4(void)”
* mod\_auth\_kerb
* mod\_auth\_gssapi.c
* ssh-gss.h
* radius\_example.c
* eap\_example
* kexgsss.c

**Unlocatable files**

Here you’ll find a list of files which are not present on the JaNET live CD installation but are mentioned in either the original installations or referred to in the code.

* Security/AuthSession.h

**Points of attention**

A list of things that caught our attention while looking through the JaNET live CD installation.

* /moonshot/moonshot/libeap/radiusexample (radius\_example.c)

Radius configuration build in c. Not yet analysed.

* /moonshot/moonshot/libeap/eap\_example (certificates (.pem))

They’ve experimented with certificates as well, unknown how far they’ve got or what the results are.

**Overview changes Openssh**

|  |  |  |
| --- | --- | --- |
| **Removed files** | **Changed files** | **Added files** |
| **/moonshot/openssh** | **/moonshot/openssh** | **/moonshot/openssh** |
| gss-serv-krb5.c | auth2.c | changeLog.gssapi |
| sandbox-darwin.c | auth2-gss.c | kexgssc.c |
| sandbox-null.c | authfile.c | kexgsss.c |
| sandbox-rlimit.c | changelog | ssh\_prng\_cmds.in |
| sandbox-systrace.c | clientloop.c | ssh-rand-helper.0 |
| ssh-sandbox.h | configure | ssh-rand-helper.8 |
|  | configure.ac | ssh-rand-helper.c |
|  | entropy.c | WARNING.RNG |
|  | gss-genr.c |  |
|  | gss-serv-krb5.c |  |
|  | kex.c |  |
|  | kex.h |  |
|  | makefile.in |  |
|  | monitor.h |  |
|  | mux.c |  |
|  | PROTOCOL.mux |  |
|  | servconf.c |  |
|  | servconf.h |  |
|  | sftp.1 |  |
|  | ssh.c |  |
|  | ssh\_config.0 |  |
|  | ssh\_prng\_cmds.in |  |
|  | ssh-agent.c |  |
|  | sshconnect.c |  |
|  | sshconnect2.c |  |
|  | sshd.c |  |
|  | ssh-gss.h |  |
|  | ssh-keygen.0 |  |
|  | ssh-keygen.1 |  |
|  | ssh-keygen.c |  |
|  |  |  |
|  | **/moonshot/openssh/contrib/cygwin** |  |
|  | **ssh-host-config** |  |
|  | ssh-user-config |  |
|  |  |  |
|  | **/moonshot/openssh/contrib./redhat** |  |
|  | sshd.init |  |
|  |  |  |
|  | **/moonshot/openssh/contrib./suse** |  |
|  | openssh.spec |  |
|  |  |  |
|  | **/moonshot/openssh/openbsd-compat** |  |
|  | bsd-cygwin\_util.c |  |
|  |  |  |
|  | **/moonshot/openssh/regress** |  |
|  | cfgmatch.sh |  |
|  |  |  |

**strace & ltrace**

**Conclusion**

Due to the amount of changes JaNET did to their Test Environment and without their documentation, it is hard to pinpoint the essence of their work. We could use some of their work as an example but it takes quite some time to implementate it to our purpose. This is why we need to consider if it is smart to put quite some time in it analysing their work, if not focus on our own solution.