THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

University of Waikato
Network Research Group

openLI

Adding TLS to OpenLI -- Part 2

OpenLI Training: Chapter Nineteen

Shane Alcock
University of Waikato
New Zealand
shane.alcock@waikato.ac.nz

Previously on OpenLI...

- These error messages don't look great

```
openliprovisioner[567]: OpenLI: SSL Handshake failed for collector 172.19.0.4-51862
openliprovisioner[567]: OpenLI: SSL handshake for mediator 172.19.0.3-58354 is pending...
openliprovisioner[567]: OpenLI: Pending SSL Handshake for mediator 172.19.0.3-58354 failed
```

Enabling TLS

- TLS must be enabled for ALL components
  - Repeat configuration changes for collector and mediator
  - Don’t forget to restart components afterwards

Enabling TLS

- TLS configuration for the collector

```
tlscert: /etc/openli/ssl/collector-crt.pem
tlskey: /etc/openli/ssl/collector-key.pem
tlsca: /etc/openli/ssl/ca-crt.pem
```

Enabling TLS

- TLS configuration for the mediator

```
tlscert: /etc/openli/ssl/mediator-crt.pem
tlskey: /etc/openli/ssl/mediator-key.pem
tlsca: /etc/openli/ssl/ca-crt.pem
```

Success!

- After restarting, check the provisioner logs again

```
openliprovisioner[567]: OpenLI: SSL handshake for collector 172.19.0.4-51866 is pending...
openliprovisioner[567]: OpenLI: Pending SSL handshake for collector 172.19.0.4-51866
accepted
openliprovisioner[567]: OpenLI provisioner: collector 172.19.0.4-51866 is now active
openliprovisioner[567]: OpenLI: SSL handshake for mediator 172.19.0.3-58358 is pending...
openliprovisioner[567]: OpenLI: Pending SSL handshake for mediator 172.19.0.3-58358
accepted
openliprovisioner[567]: OpenLI: mediator 172.19.0.3-58358 on fd 12 auth success.
```

Confirming Encryption

- Repeat the tracepktdump experiment from earlier
  - Packet payload is now encrypted

```
unknown protocol tcp/8080
 Unknown Protocol: 8080
  17 03 03 03 0d 30 7b c6 6b c9 74 7a 78 48 8d 13....0{.k.tzxH..
  d9 5e 8c e5 8f af 45 15 fc 97 21 76 e3 d9 15 32.^....E...!v...2
  a5 fe 4c 43 1a 90 0c 11 0a 1e 10 71 13 8b 8d a9..LC.......q....
  3a b7 54 2c 08 de fc 64 a0 37 76 48 d2 68 27 fd:.T,...d.7vH.h'.
  16 c3 44 fc b0 98 a3 42 1a 57 db 2e 8c 77 63 76..D....B.W...wcv
  aa 58 b4 87 ae 6a 46 58 85 0a 57 98 a4 6d 69 59.X...jFX..W..miY
  ff ec 08 3c 44 2b 73 6f f3 93 76 1b e8 08 30 2a...<D+so..v...0*
  a6 f4 5c 56 c0 6a 6e 56 bc 9c 98 62 de f7 50 17..\V.jnV...b..P.
  9a 5c 2c 4d 02 61 2e dd 94 53 89 b7 5e 01 54 fb.\,M.a...S..^.T.
  f9 71 2e 06 55 e3 b7 69 dc 9f 58 ee 5c fd 21 e2.q..U..i..X.\.!.
  1f 7a 08 f7 01 c8 17 08 6f 35 7f 61 25 b9 33 54.z......o5.a%.3T
  d6 82 ac 4a 8f 58 dd 3c e5 7f 45 fb 48 51 39 2c...J.X.<..E.HQ9,
  6f bb 08 f0 20 dd 4b 8e b6 c6 36 d6 e4 0b 08 25o... .K...6....%
  07 c8 3d 01 69 3f f3 e4 29 ee 4e 3c b3 e3 a6 db..=.i?..).N<....
  59 0b 40 2e dc cb e1 14 0e 2a 62 05 18 b0 6b 08Y.@......*b...k.
  9f f5 b1 19 99 18 1f 66 bd 4d fc c5 24 a2 05 2a.......f.M..$..*
  f6 00 95 e7 94 8a e1 fd ab dd 69 48 c1 10 82 18..........iH....
```

More Encryption

- What is encrypted
  - Provisioner < -- > Collector
  - Provisioner < -- > Mediator

- What is NOT encrypted
  - Collector < -- > Mediator
  - Mediator < -- > Agency
    - Requires IPSec tunnel or VPN to the agency

Why not Encrypt Collector Output?

- Intercepted traffic is potentially high throughput
  - Interception of heavy traffic users
  - Multiple concurrent interceptions

- Encryption adds overhead
  - Reduces maximum interception rate
  - Could lead to loss of intercepted packets

Encrypting Collector Output

- Configuration option to enable encryption
  - Must be set in both collector AND mediator config

```
/home/openli-coll# vim /etc/openli/collector-config.yaml

...

etsitls: yes        # Set to yes to ENABLE encrypted output

...
```

Encrypting Collector Output

- Configuration option to enable encryption
  - Must be set in both collector AND mediator config

```
/home/openli-med# vim /etc/openli/mediator-config.yaml

...

etsitls: yes        # Set to yes to REQUIRE collectors to send encrypted records

...
```

Encrypting Collector Output

- Remember to stop and restart both components
  - Be aware of possible performance degradation

Encrypting Collector Output

- You can use tracepktdump to confirm encryption is enabled
  - Run it on the collector component, interface eth1
  - Replay one of the earlier example pcaps
    - tcpsip_voip.pcap is a good one to use
  - There should be no identifiable text in the packet payloads

Encrypting Collector Output

- Unencrypted example

```
...
unknown protocol tcp/12009
 Unknown Protocol: 12009
  5c 4c 6c 5c 02 fb 00 11 00 00 00 00 00 00 00 00  \Ll\............
  00 0b 54 45 53 54 56 4f 49 50 30 30 31 30 82 02  ..TESTVOIP0010..
  ea a1 56 80 08 04 00 02 02 05 01 11 00 81 0b 54  ..V............T
  45 53 54 56 4f 49 50 30 30 31 82 02 4e 5a a3 1e  ESTVOIP001..NZ..
  a0 12 80 04 57 41 4e 44 81 0a 6f 70 65 6e 6c 69  ....WAND..openli
  2d 6c 61 62 81 04 6b 11 e9 75 82 02 4e 5a 84 01  -lab..k..u..NZ..
  00 86 06 63 6f 6c 30 30 31 a7 0b 80 04 61 79 ca  ...col001....ay.
  ac 81 03 03 20 a2 88 01 01 a2 82 02 8e a0 82 02  .... ...........
  8a 30 82 02 86 80 01 04 a2 82 02 7f ab 82 02 7b  .0.............{
  80 04 05 05 06 01 a1 82 02 71 a1 82 02 6d a0 14  .........q...m..
  81 01 00 a2 06 81 04 c0 a8 01 49 83 01 03 85 04  ..........I.....
  ff ff ff ff a1 14 81 01 00 a2 06 81 04 0a 64 32  ..............d2
  41 83 01 03 85 04 ff ff ff ff 82 82 02 3d 52 45  A.............=RE
  47 49 53 54 45 52 20 73 69 70 3a 74 6c 73 2e 65  GISTER sip:tls.e
  78 61 6d 70 6c 65 2e 63 6f 6d 3b 74 72 61 6e 73  xample.com;trans
  70 6f 72 74 3d 74 63 70 20 53 49 50 2f 32 2e 30  port=tcp SIP/2.0
  0d 0a 43 61 6c 6c 2d 49 44 3a 20 39 30 36 62 37  ..Call-ID: 906b7
  37 31 38 62 37 32 62 64 34 34 32 65 66 30 36 32  718b72bd442ef062
...
```

Encrypting Collector Output

- Encrypted example

```
...
unknown protocol tcp/12009
 Unknown Protocol: 12009
  17 03 03 03 1c cd 0a a8 ea 9e b2 8a f1 af 67 1d  ..............g.
  d8 5e 9c 0f 9a 56 f2 65 d5 c4 1a 95 f8 21 53 44  .^...V.e.....!SD
  ea 87 c7 0f 1a be 8b 59 14 81 24 b7 2f b8 37 66  .......Y..$./.7f
  03 7d 6f 9d cd b4 8b b4 0b 1e 0b 0b d1 d1 f3 a9  .}o.............
  bd d1 f3 5a c1 7d 54 81 1b 6d 72 0a 7e 6a c1 02  ...Z.}T..mr.~j..
  9a 23 6a e6 91 61 37 f7 cc c3 79 9f 7f d4 8a 0b  .#j..a7...y.....
  86 65 bf 35 92 68 1c d1 5b ae 70 65 bb 9f ea 00  .e.5.h..[.pe....
  c8 39 1d f6 03 12 c5 af 38 74 a9 7c 15 15 af 05  .9......8t.|....
  5e a5 d0 ea fa aa 19 50 a0 43 63 0e a5 f9 45 c3  ^......P.Cc...E.
  66 08 3e 7c a8 d0 16 0b 1f 95 c4 af 17 94 f1 1f  f.>|............
  e3 08 6a 32 d3 9e d3 2e 1f 3f 04 b5 8f f3 6c ab  ..j2.....?....l.
  6d f4 25 86 bf 32 ba d1 c7 0f 53 5d 3f 4e 41 6a  m.%..2....S]?NAj
  97 7f bd c2 ce 05 10 a9 51 9f 1c 27 15 91 f1 9e  ........Q..'....
  fc 7b c7 a4 42 1c a1 ff 91 2c 5e 29 52 7f 5e 27  .{..B....,^)R.^'
  06 02 41 d9 14 24 e5 22 b5 0a c7 09 1d fa 3b b7  ..A..$."......;.
  97 dc 71 a6 7d c5 28 42 17 9d 67 b0 19 04 99 d9  ..q.}.(B..g.....
  a0 19 42 b3 49 ee 65 c0 67 94 d0 6e f6 05 d5 da  ..B.I.e.g..n....
  72 66 4b b6 50 41 d5 61 47 5c 8a 45 b2 4c ea b8  rfK.PA.aG\.E.L..
...
```

Secure REST API

- REST API is now using HTTPS
  - Requests and responses are now encrypted
  - Request URL must now begin with `https://`

Secure REST API

- curl does not like servers running self-signed certificates

```
/home/openli-prov# curl -X GET https://172.19.0.2:8080/ipintercept

curl: (60) SSL certificate problem: unable to get local issuer certificate
More details here: https://curl.haxx.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.
```

Secure REST API

- curl does not like servers running self-signed certificates
  - Add -k option to tell curl to ignore the problem
  - NOT a good idea for production, just for the lab!
    - Use a proper signed certificate when you deploy

```
/home/openli-prov# curl -k -X GET https://172.19.0.2:8080/ipintercept

[ { "liid": "STATIC002", "authcc": "NZ", "delivcc": "NZ", "agencyid": "mocklea",
"mediator": 1, "user": "salcock", "accesstype": "fiber", "radiusident": "any", "staticips":
[ { "iprange": "10.1.18.217\/32", "sessionid": 101 } ] }, { "liid": "RADIUS003", "authcc":
"NZ", "delivcc": "NZ", "agencyid": "mocklea", "mediator": 1, "user": "b4CPidYn7u8Vesbo",
"accesstype": "xDSL", "radiusident": "user" }, { "liid": "NZP_20211010", "authcc": "NZ",
"delivcc": "NZ", "agencyid": "mocklea", "mediator": 1, "user": "2On5uRWxvQDeBBepKBu",
"accesstype": "wifi", "radiusident": "any", "vendmirrorid": 500 } ]
```

TLS in the Real World

- In the lab, we've taken a few shortcuts to make life easier
  - Self-signed certificates
  - Certificates were already installed on the components
  - We ignore certificate issues when using the REST API

TLS in the Real World

- For a real deployment, you'll need to:
  - Create and sign your own certificates
    - Use a real CA -- don't self-sign!
  - Copy them onto your component hosts
  - Set appropriate permissions to secure them
  - Pay attention to security warnings

Next Time

- Adding authentication to the REST API
  - Prevent random users from adding or inspecting intercepts