EPICS Security Technical Plan

Osprey DCS
George McIntyre

- A presentation of the implementation plan for EPIC Security, being carried out by SLAC, Osprey DCS and ORNL

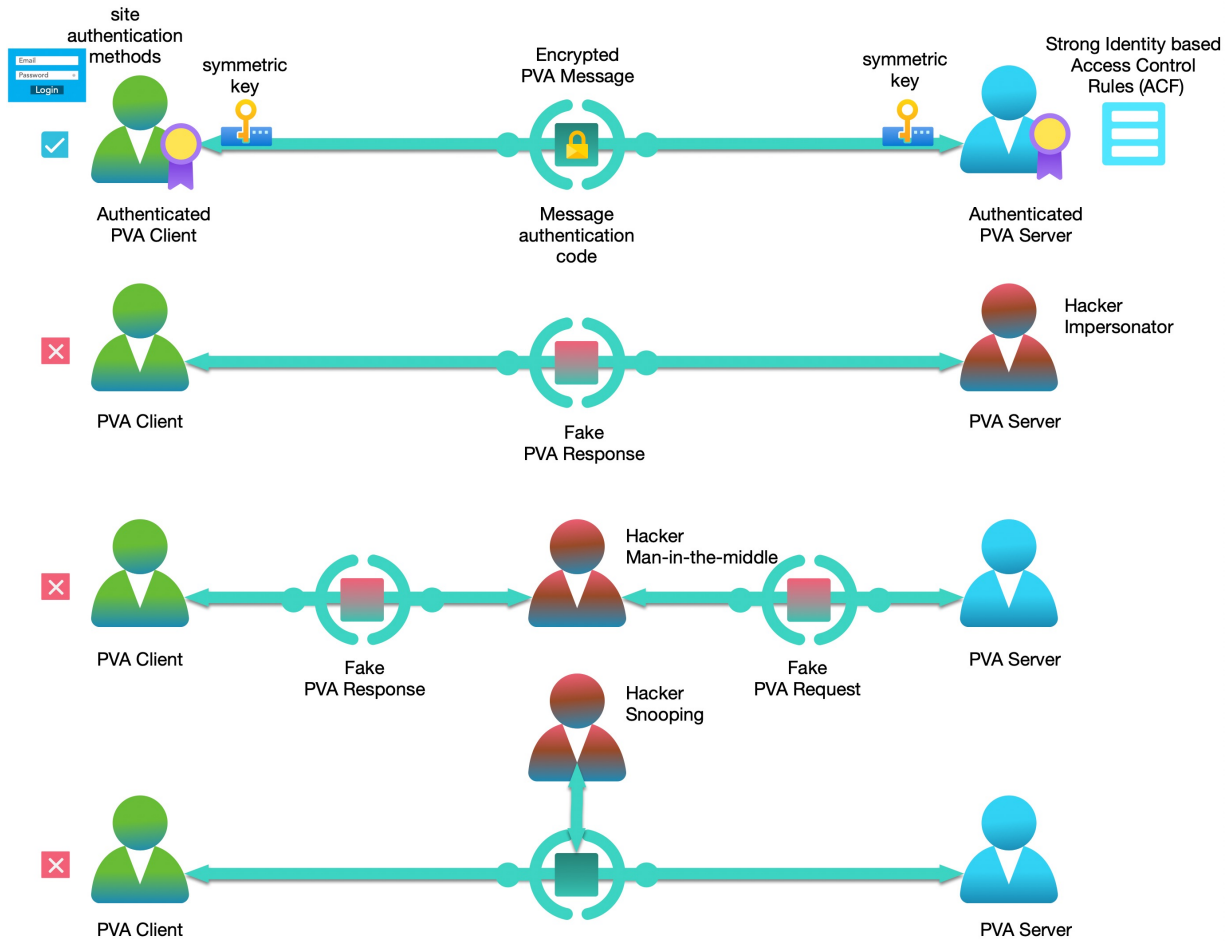
Agenda

- **Planned Features**
  - Implementation of basic TLS
  - Certificate Management
  - TLS Session Status & Management
  - Enhanced Client Authorization
- **TLS Demo**
  - Manually Create Certificates
  - Configure Wireshark
  - Non-secure PVA communications
  - Configure secure PVA communications
  - Demo Secure PVA communications

What this implementation will get us?

Benefits

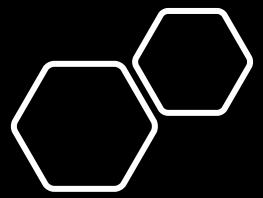
- Server Certificates ➔
  - **Authentication**:
    - Prevent **Service Impersonation**
    - Prevent **Man-in-the-Middle** attacks
- PVA Message Encryption ➔
  - Cipher suite Message Authentication Codes ➔
    - Guarantee **Data Integrity**
  - Securely shared Symmetric Session Keys ➔
    - Prevent **Packet Snooping**
- Client Certificates & Site Authentication integration ➔
  - **Authorization**:
    - Provide a mechanism for **Service Access Control**
    - **Protect Data** by allowing Services to Restrict Access



What this implementation will get us?

Benefits	Will Not

• Server Certificates ➔
  • Authentication:
  • Prevent **Service Impersonation**
  • Prevent **Man-in-the-Middle** attacks
• PVA Message Encryption ➔
  • Cipher suite Message Authentication Codes ➔
  • Guarantee **Data Integrity**
  • Securely shared Symmetric Session Keys ➔
  • Prevent **Packet Snooping**
• Client Certificates & Site Authentication integration ➔
  • Authorization:
  • Provide a mechanism for **Service Access Control**
  • **Protect Data** by allowing Services to Restrict Access
	• Prevent **PV Impersonation** in a mixed TLS/TCP network
	• Prevent discovery of **Service Endpoint** or **PV name**
	• Prevent discovery of **Encryption Type**
	• Prevent discovery of **Data Transmission Frequency**
	• Prevent discovery of approximate **Amount of data transmitted**
	• Change **Site Security Policies** – you need to implement these on top of the technical solution presented here



Basic TLS Implementation

Basic TLS Implementation

- Allow “tls” in search request
- Allow “tls” in search response
- Initiate TLS handshake if server certificate is configured
- Pass client certificate in handshake if client certificate is configured
- Encapsulate and encrypt PVA protocol messages
- Maintain backwards compatibility

TLS in EPICS

Programmatic Interface changes

Possibility of specifying “tls” as a protocol

New PV Access Server configuration options for TLS

Network Management Impact

Install Server Certificates

***Note**: TLS and legacy clients/servers use the same ports and may interoperate on the same network without any changes to legacy clients and servers*

New EPICS Features

Verified identity of EPICS channel servers

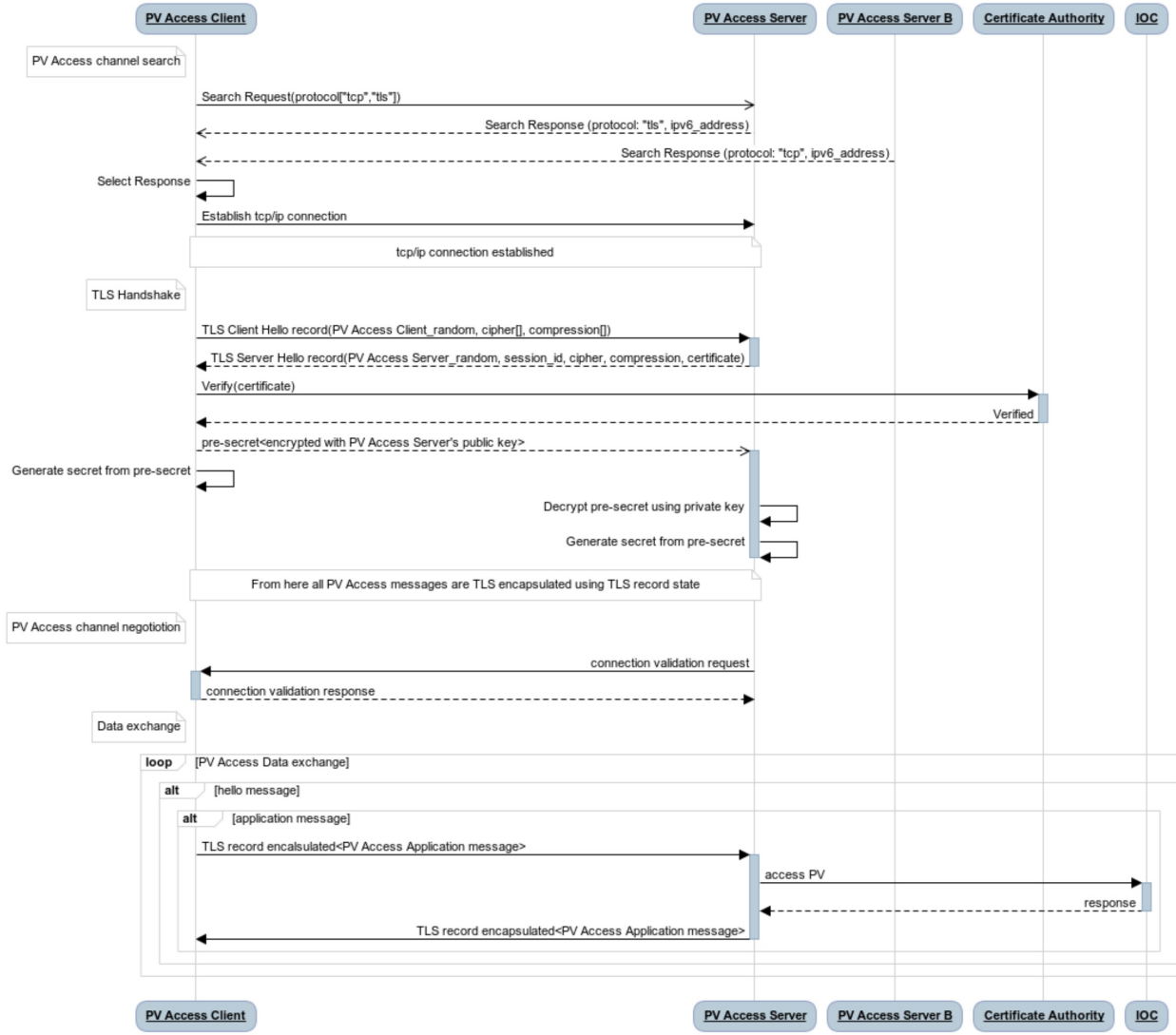
Guaranteed data integrity for EPICS services/channels

Encrypted EPICS data packets

Protocol Changes

New client-initiated TLS handshake phase before connection validation

Encapsulation of PV Access Messages



Implementation repositories

Java implementation – *maintainer Kay Kasemir*

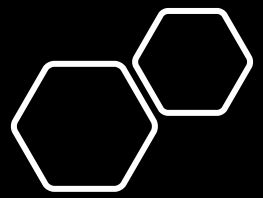
- https://github.com/ControlSystemStudio/phoebus/tree/master/core/pva

C++ implementation – *maintainer Michael Davidsaver*

- https://mdavidsaver.github.io/pvxs branch TLS

Documentation

- https://github.com/epics-base/pvAccessCPP/wiki/protocol



Out of scope

Features

- Add TLS to Channel Access
- UDP Broadcast search
- UDP response
- Beacon messages
- Add additional TLS beacon messages for servers supporting both TLS and TCP
- Any changes to support TLS in Gateways
- Any changes to support TLS in EPICS Python (pvaPy)
- Any changes to support TLS in PV Database

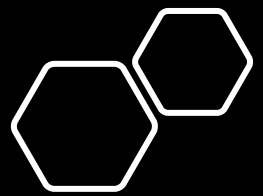
Repositories

EPICS base Java

- https://github.com/epics-base/epicsCoreJava
- https://github.com/epics-base/pvaClientJava

EPICS base C++

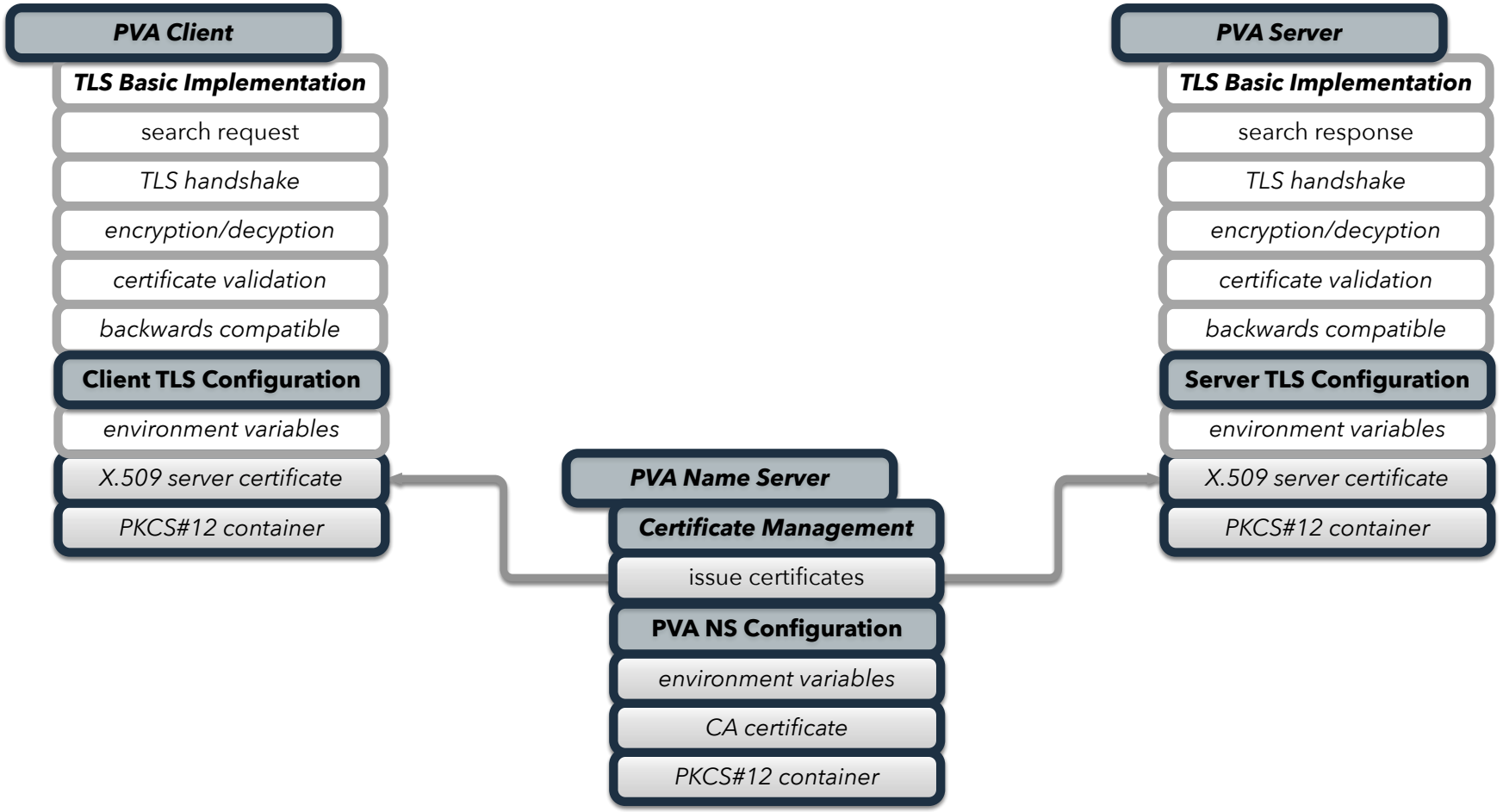
- https://github.com/epics-base/pvAccessCPP
- https://github.com/epics-base/pvaClientCPP



Certificate Management

Certificate Management

- Update PVA Name Server to issue client and server certificates
- Manage Site Certificate Authorities



Certificate Management

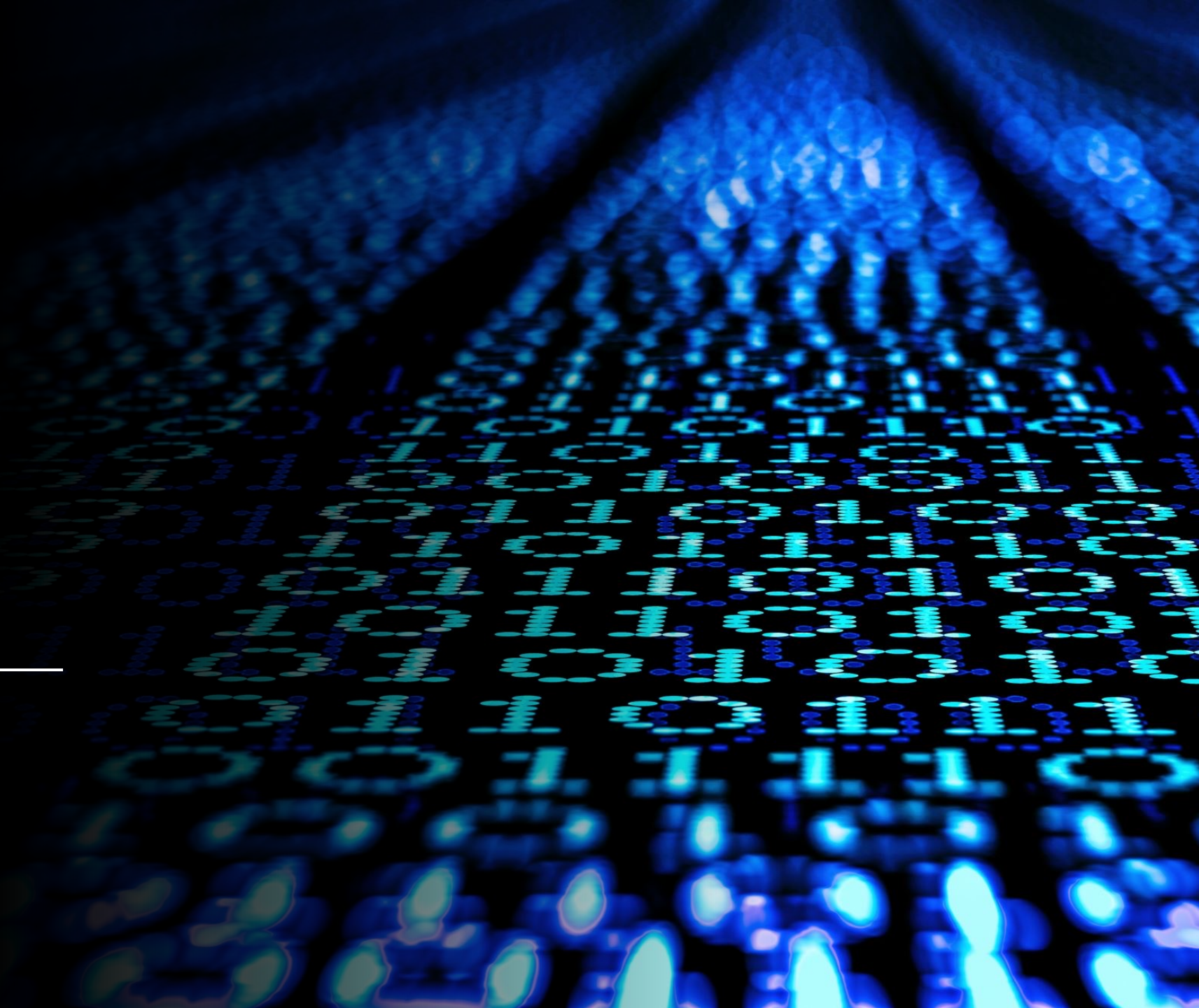
Manage Site Certificate Authority

- Secure Storage for Site CA
- Management of Private Key

Issue and Distribute Server and Client Certificates

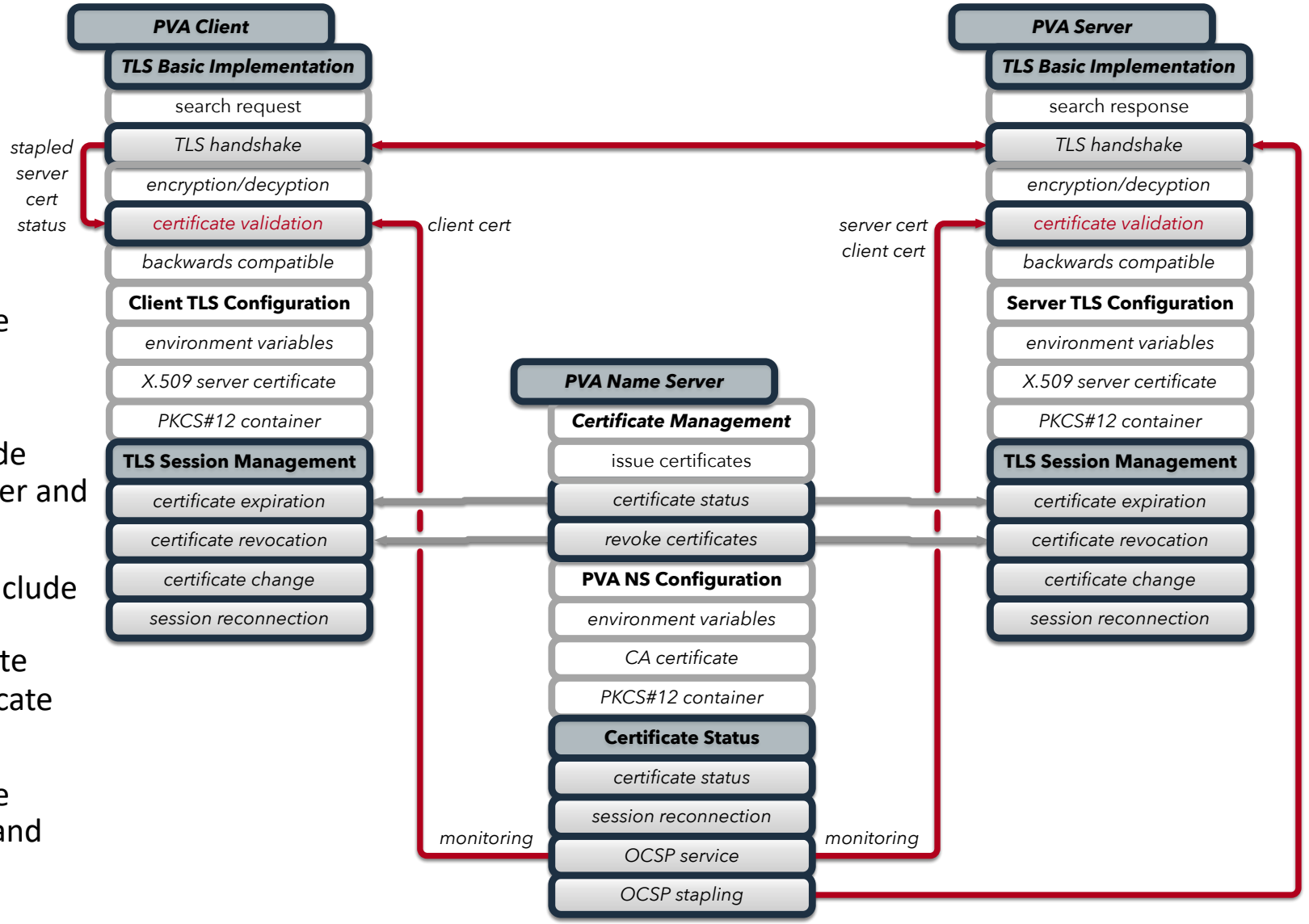
- Create Certificate Key Pairs
- Create Signing Request
- Sign Certificates with CA
- Deliver server and client certificates

Session Status and Management



Session Status and Management

- Update Client and Server to handle reconnections after session disconnections
- Update PVA Name Server to provide certificate status via OCSP for Server and Client's certificate validation
- Update server TLS handshake to include server certificate status with OCSP stapling and update client certificate validation to use this server certificate status
- Update Client and Server to handle certificate expiration, revocation, and change



Session Status & Management

Certificate Status

Implement OCSP server to deliver certificate status

Allow pub/sub event model for server status monitoring

Client and Server use this service to monitor for their own status changes

Revocation

Implements a function that revokes a certificate by setting the appropriate status

Clients and servers respond appropriately to revoked status

Expiration

Recognizes certificate expiration

Alerts listeners to upcoming expirations

Clients and Servers respond appropriately when certificates expire

Rotation

Allow orderly rotation of valid certificates with new valid certificates

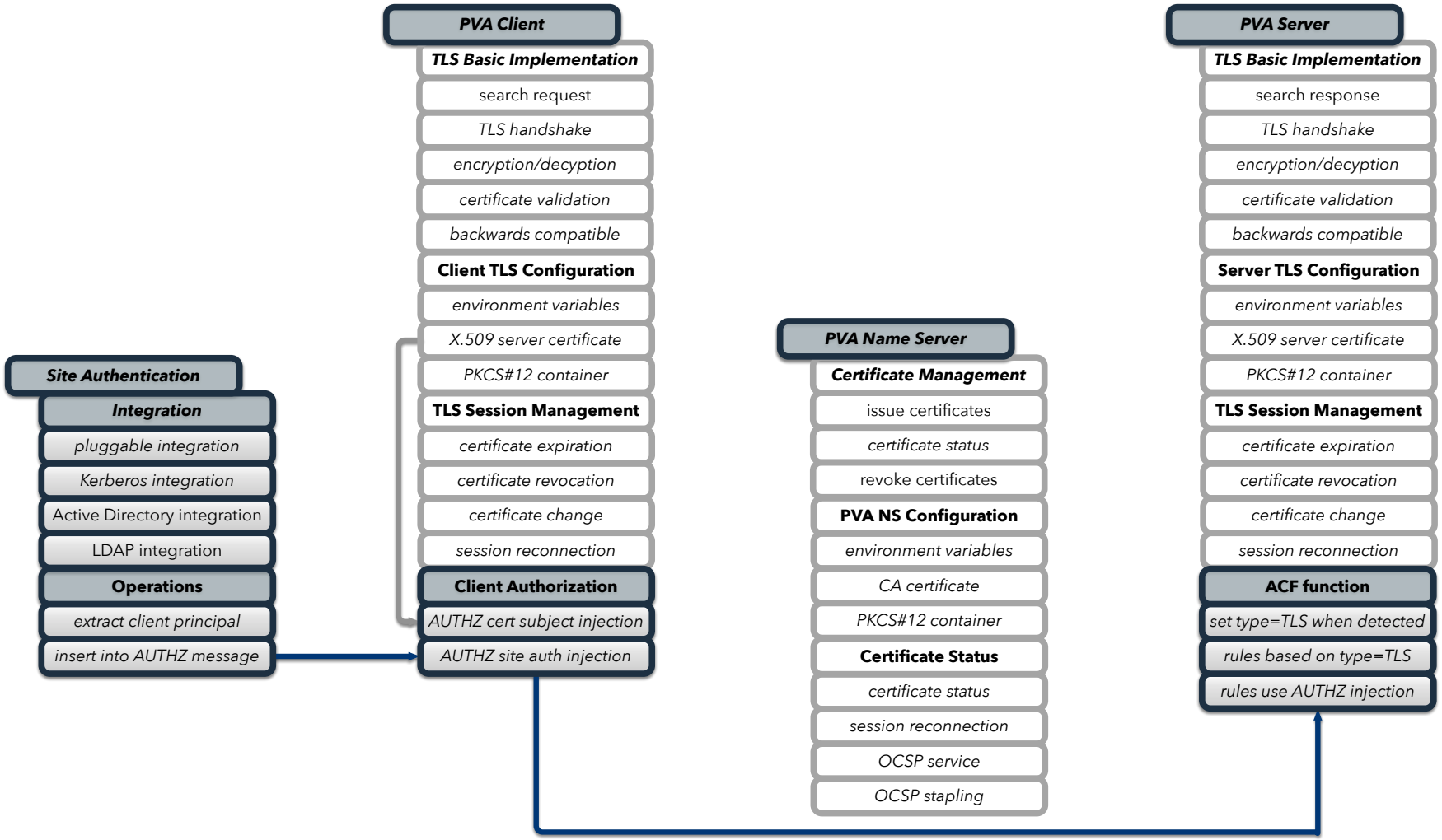
Stapling

Append server certificate status to TLS handshake with the OCSP stapling extension so that clients don't have to verify the status

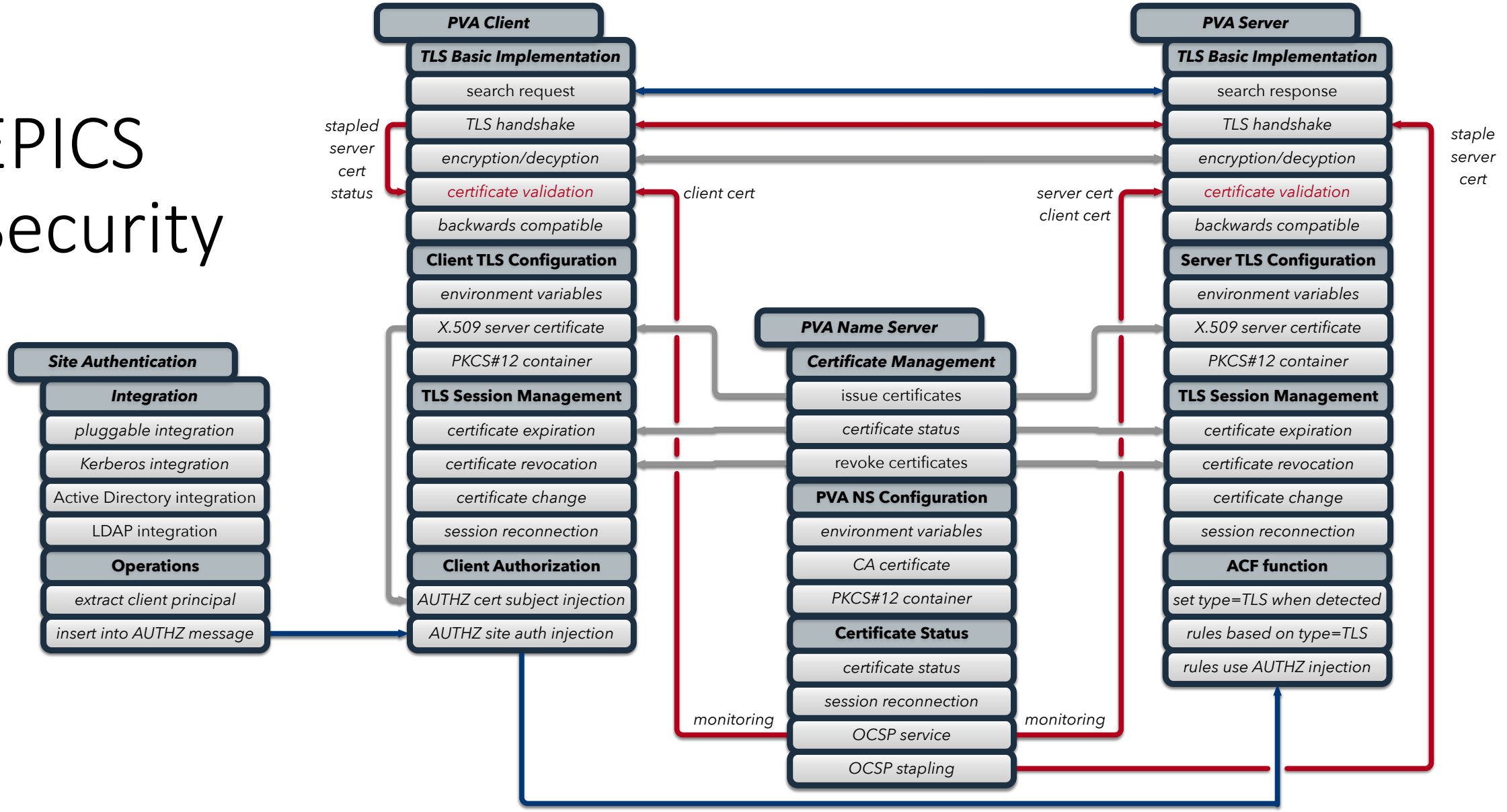
Enhanced Client Authorization

Enhanced Client Authorization

- Integrate with common site authentication methods
  - Kerberos
  - LDAP
  - Active Directory
- In PVA Connection validation message inject into the AUTHZ name field either the certificate subject or principal from site authentication integration
- Update ACF function that controls authorization so that it can use TLS status to control access



EPICS Security



Recap

1. **Basic TLS** to authenticate and encrypt
2. **Certificate Management** to issue and distribute certificates
3. **Session Management** to handle expiration, revocation, recycling, and reconnection
4. **Authorization** using site authentication or client certificates

TLS Demo

Manually create certificates

CA Certificate

```
openssl genpkey -algorithm RSA -out ca.key

# create ca_config.cnf

openssl req -x509 -new -nodes -key ca.key -sha256 -days 3650 -out ca.crt -config ca_config.cnf
```

```
ca_config.cnf ...
keyUsage = digitalSignature, cRLSign, keyCertSign
authorityKeyIdentifier = keyid:always,issuer
basicConstraints = CA:TRUE
```

Server Certificate

```
openssl genpkey -algorithm RSA -out server.key

# create server_config.cnf

openssl req -new -key server.key -out server.csr -config server_config.cnf

openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt -days 365 -sha256
-extfile server_config.cnf -extensions v3_req

openssl pkcs12 -export -out server.p12 -inkey server.key -in server.crt -certfile ca.crt
```

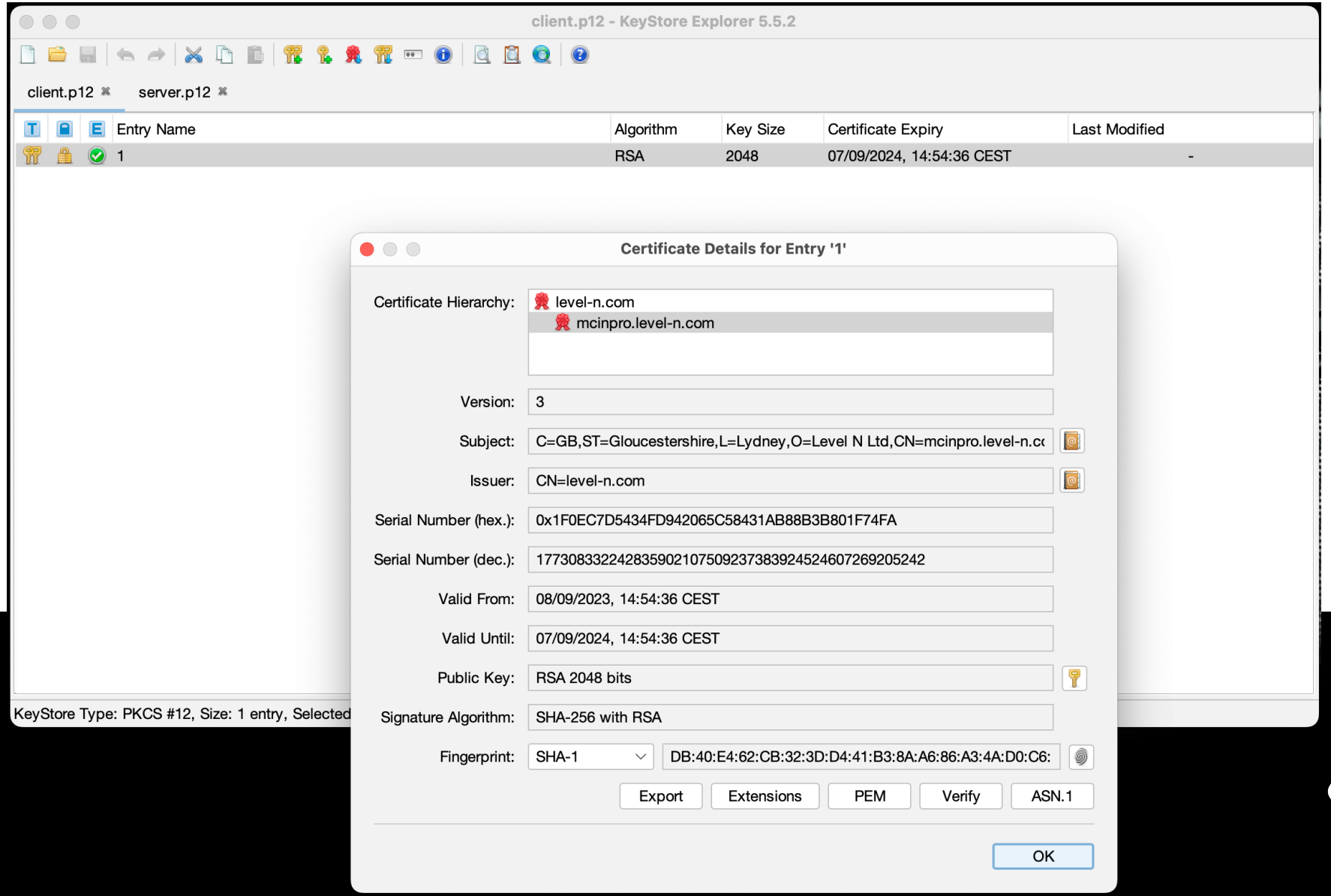
```
server_config.cnf ...
keyUsage = digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth
```

Client Certificate

```
openssl genpkey -algorithm RSA -out client.key

# create client_config.cnf

openssl req -new -key client.key -out client.csr -config client_config.cnf

openssl x509 -req -in client.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out client.crt -days 365 -sha256
 -extfile client_config.cnf -extensions v3_req

openssl pkcs12 -export -out client.p12 -inkey client.key -in client.crt -certfile ca.crt
```

```
client_config.cnf ...
keyUsage = digitalSignature, keyEncipherment
extendedKeyUsage = clientAuth
```

Use KeyStore explorer to view certificates



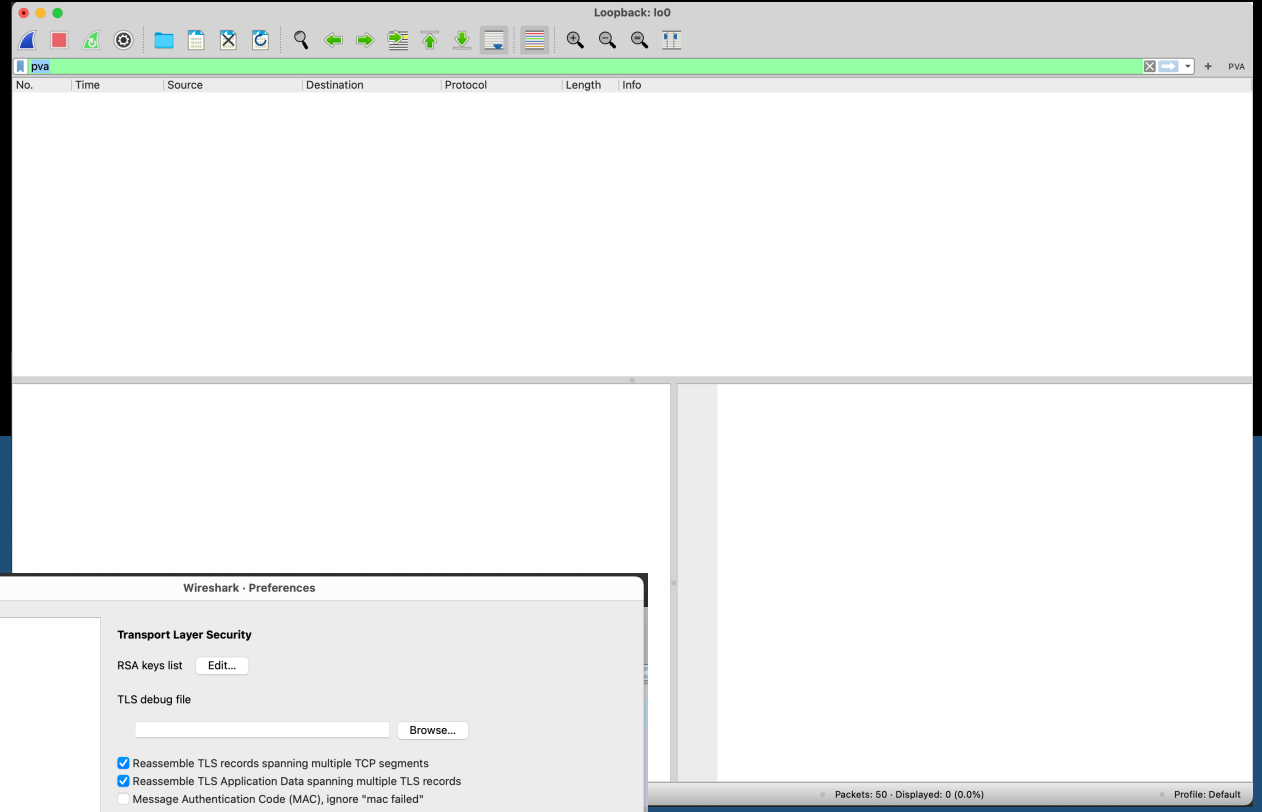
Configure Wireshark

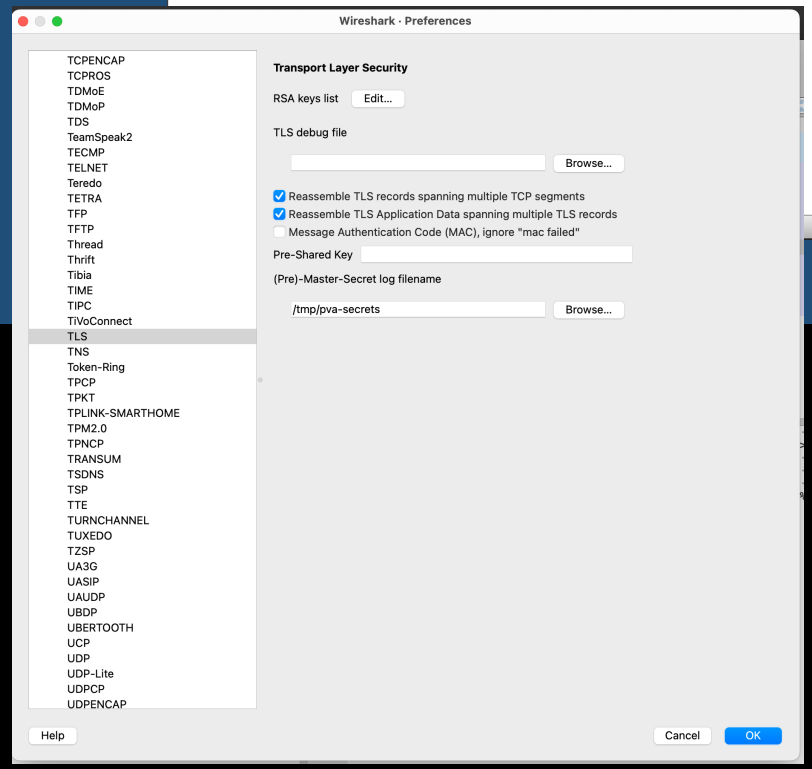
Set up LUA scripts

```
cd ~/.config/wireshark/plugins/
ln -s ~/Projects/com/osprey-dcs/cashark/*.lua
```

Set up key log file to capture session keys

```
export SSLKEYLOGFILE=/tmp/pva-secrets
```





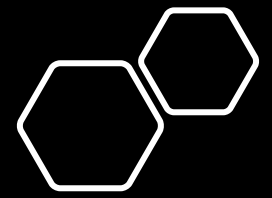
Set display filter

```
pva
```

Non-Secure PVA Communications

Start IOC server

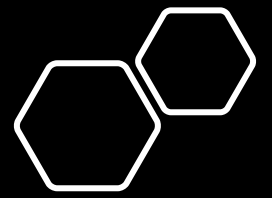
```
softIocPVX -v -m user=test,N=tst,P=tst -d test/testioc.db -d test/testiocg.db  -a test/testioc.acf
dbLoadDatabase("/Users/george/Projects/com/osprey-dcs/pvxs/bin/darwin-aarch64/../../dbd/softIocPVX.dbd")
softIocPVX_registerRecordDeviceDriver(pdbbase)
dbLoadRecords("test/testioc.db", "user=test,N=tst,P=tst")
dbLoadRecords("test/testiocg.db", "user=test,N=tst,P=tst")
asSetSubstitutions("user=test,N=tst,P=tst")
asSetFilename("test/testioc.acf")
asSetFilename: Warning - relative paths won't usually work
iocInit()
INFO: PVXS QSRV2 is loaded and ENABLED.
Starting iocInit
############################################################################
## EPICS R7.0.7.1-DEV
## Rev. R7.0.7-56-g718da5c9be96b7eccd7c
## Rev. Date Git: 2023-02-04 22:56:19 -0600
############################################################################
epics>
```

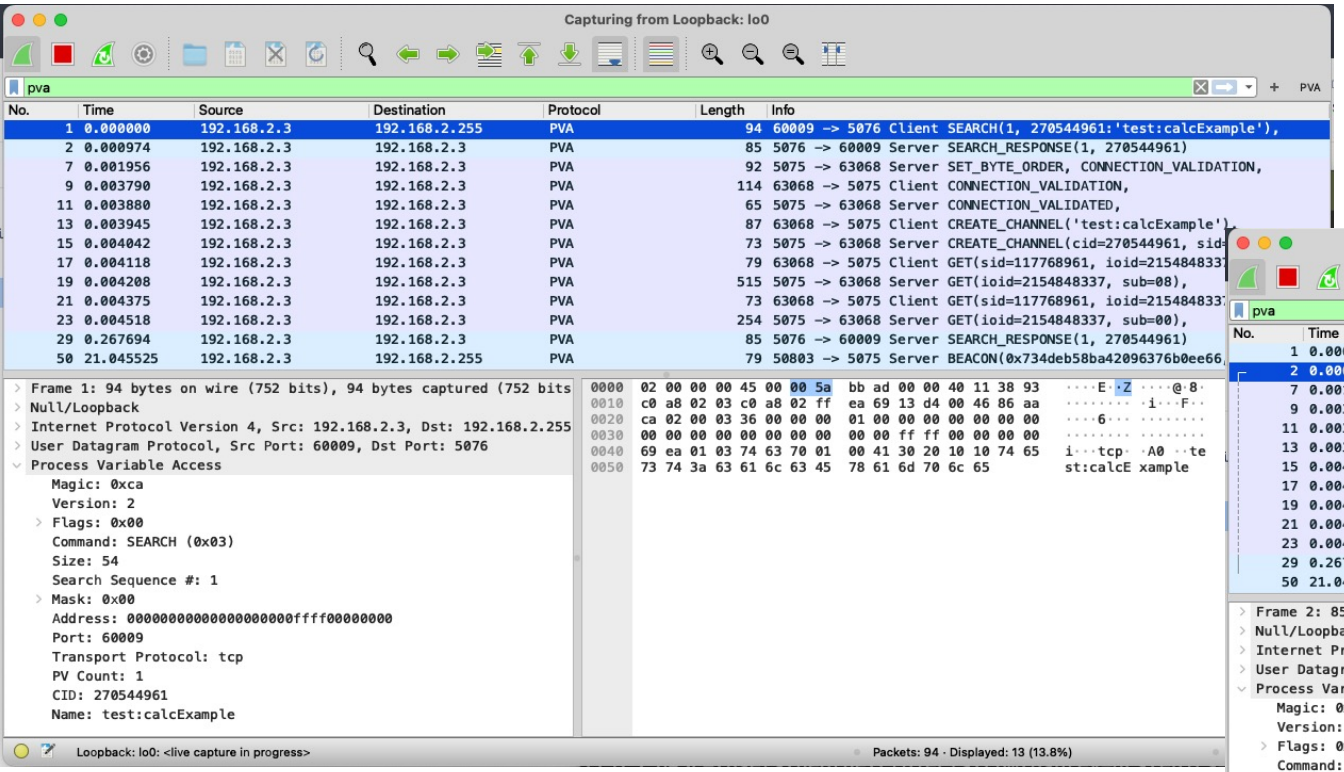


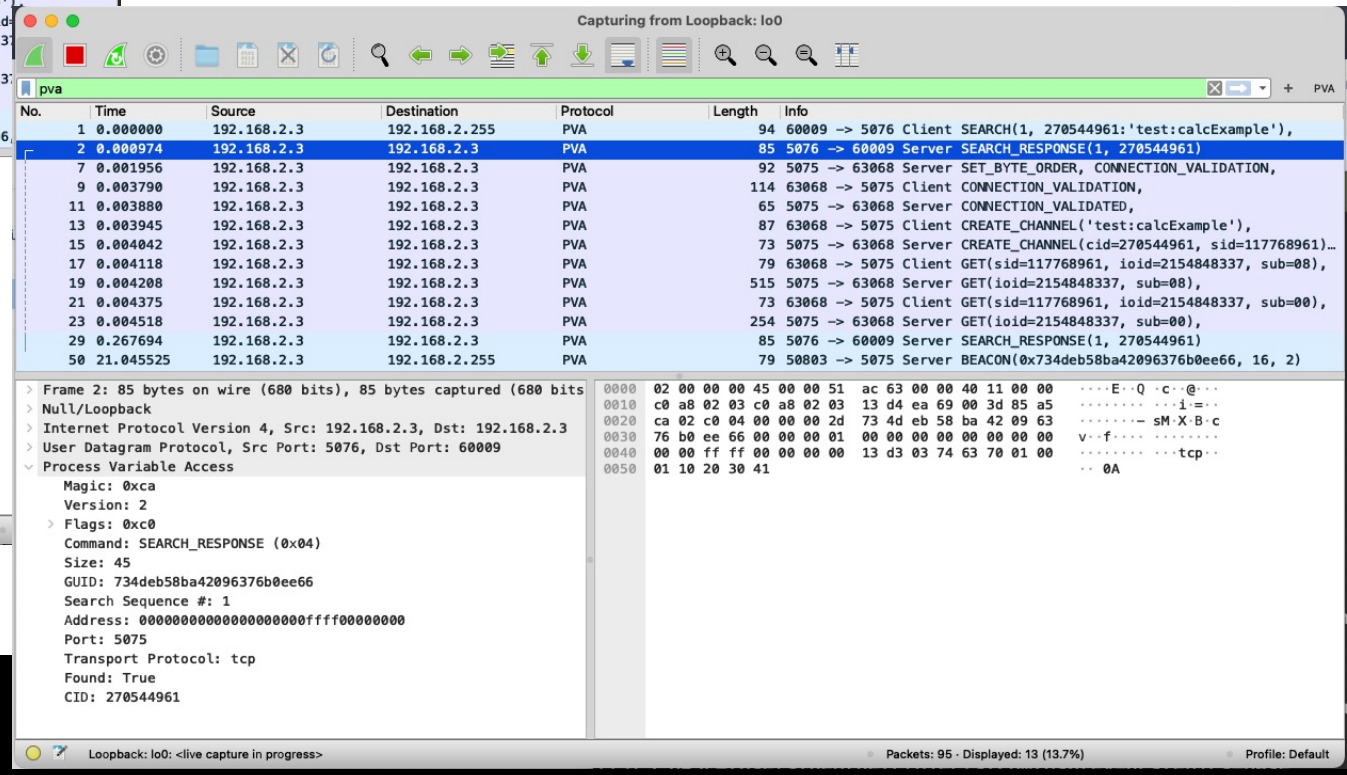
Non-Secure PVA Communications

Get a PV

```
pvxget test:calcExample
test:calcExample
    value double = 4
    alarm.severity int32_t = 1
    alarm.status int32_t = 1
    alarm.message string = "LOW"
    timeStamp.secondsPastEpoch int64_t = 1695756172
    timeStamp.nanoseconds int32_t = 258063000
    timeStamp.userTag int32_t = 0
    display.limitLow double = 0
    display.limitHigh double = 10
    display.description string = "Counter"
    display.units string = "Counts"
    display.form.index int32_t = 0
    display.form.choices string[] = {7}["Default", "String", "Binary", "Decimal", "Hex", "Exponential", "Engineering"]
    control.limitLow double = 0
    control.limitHigh double = 10
    valueAlarm.lowAlarmLimit double = 2
    valueAlarm.lowWarningLimit double = 4
    valueAlarm.highWarningLimit double = 6
    valueAlarm.highAlarmLimit double = 8
```

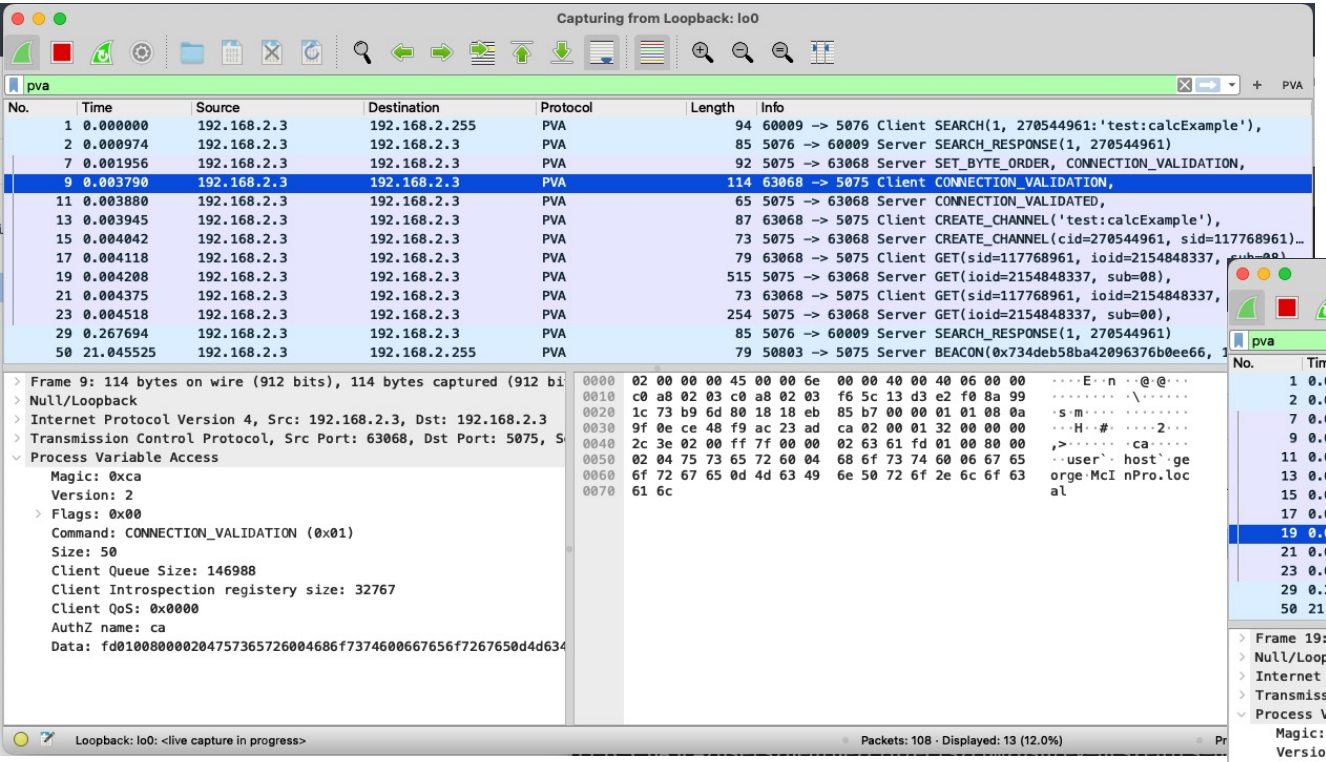


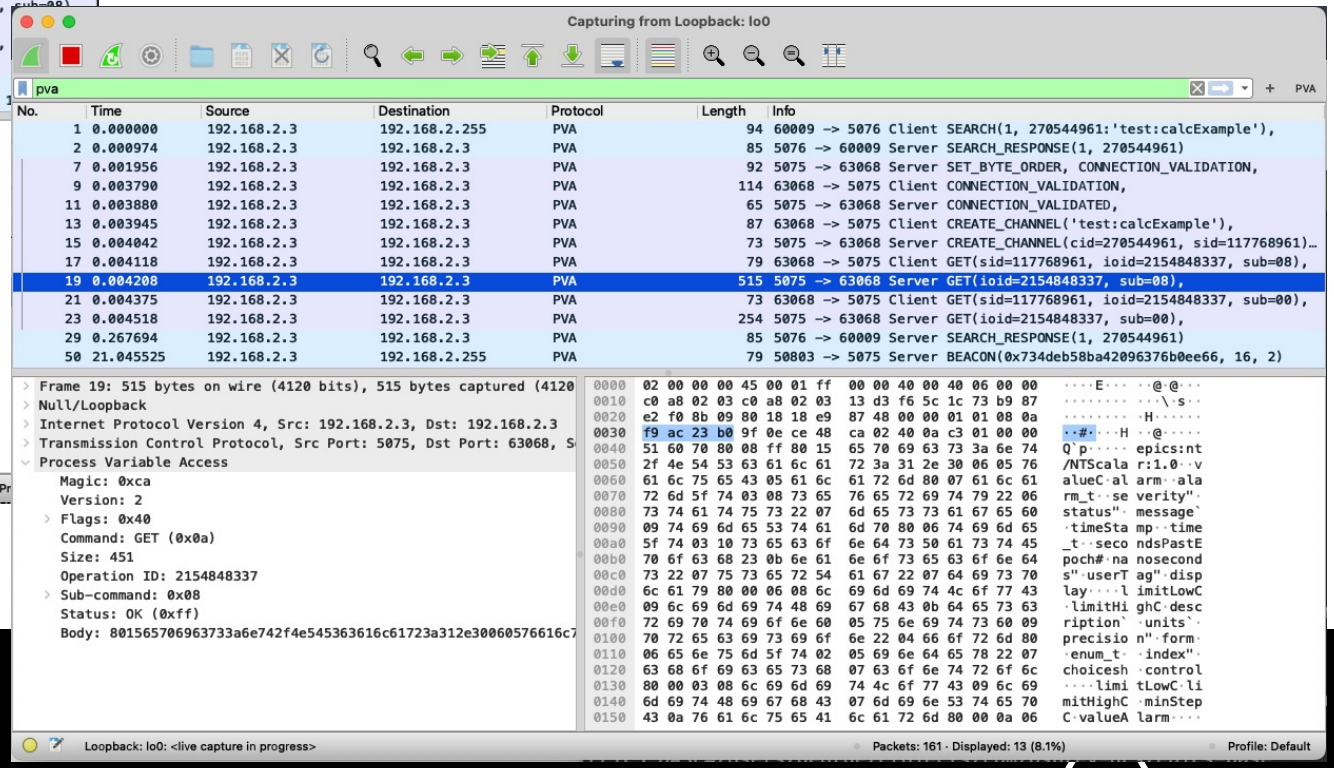




View Non-Secure Results in Wireshark

- Only **tcp** is included in search message
- No **TLS Handshake** takes place





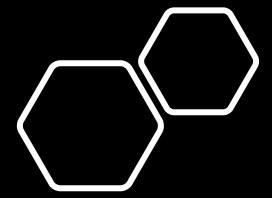
View Non-Secure Results in Wireshark

- Only **tcp** is included in search message
- No **TLS Handshake** takes place

Secure PVA Communications

Configure TLS Client and Server environments

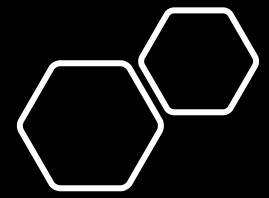
```
export EPICS_PVAS_TLS_KEYCHAIN=”~/Projects/com/osprey-dcs/certificates/server.p12"
export EPICS_PVA_TLS_KEYCHAIN=”~/Projects/com/osprey-dcs/certificates/client.p12"
```



Secure PVA Communications

Start IOC server

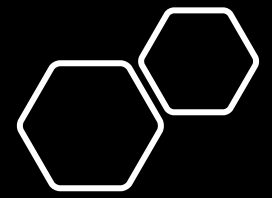
```
softIocPVX -v -m user=test,N=tst,P=tst -d test/testioc.db -d test/testiocg.db  -a test/testioc.acf
dbLoadDatabase("/Users/george/Projects/com/osprey-dcs/pvxs/bin/darwin-aarch64/../../dbd/softIocPVX.dbd")
softIocPVX_registerRecordDeviceDriver(pdbbase)
NOTICE: debug logging TLS SECRETS to SSLKEYLOGFILE=/tmp/pva-secrets
dbLoadRecords("test/testioc.db", "user=test,N=tst,P=tst")
dbLoadRecords("test/testiocg.db", "user=test,N=tst,P=tst")
asSetSubstitutions("user=test,N=tst,P=tst")
asSetFilename("test/testioc.acf")
asSetFilename: Warning - relative paths won't usually work
iocInit()
INFO: PVXS QSRV2 is loaded and ENABLED.
Starting iocInit
############################################################################
## EPICS R7.0.7.1-DEV
## Rev. R7.0.7-56-g718da5c9be96b7eccd7c
## Rev. Date Git: 2023-02-04 22:56:19 -0600
############################################################################
epics>
```

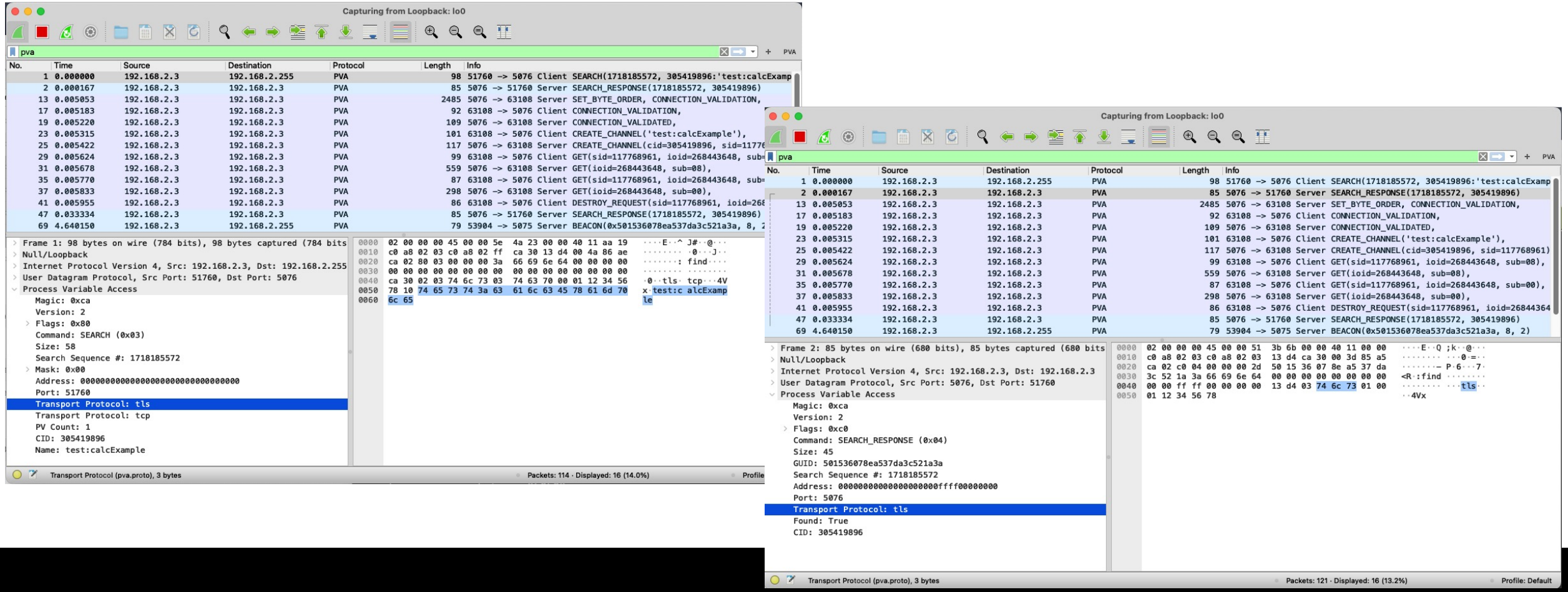


Secure PVA Communications

Get a PV

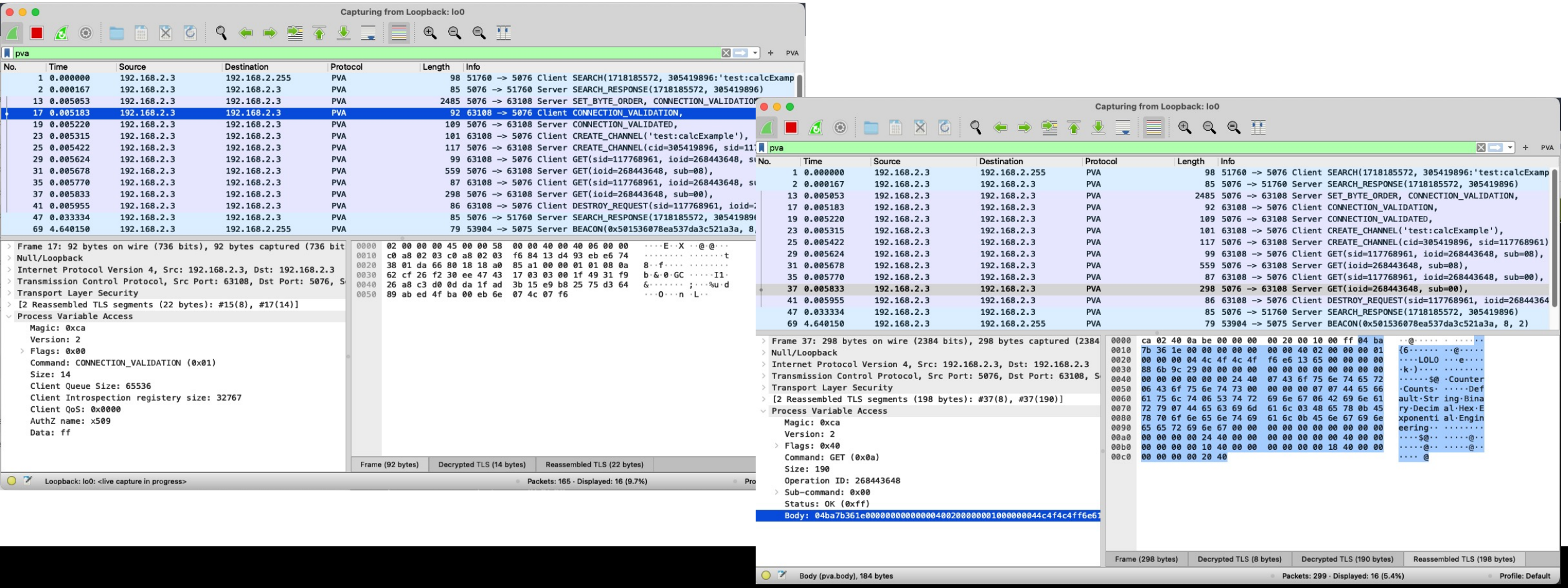
```
pvxget test:calcExample
NOTICE: debug logging TLS SECRETS to SSLKEYLOGFILE=/tmp/pva-secrets
test:calcExample
    value double = 4
    alarm.severity int32_t = 1
    alarm.status int32_t = 1
    alarm.message string = "LOW"
    timeStamp.secondsPastEpoch int64_t = 1695756172
    timeStamp.nanoseconds int32_t = 258063000
    timeStamp.userTag int32_t = 0
    display.limitLow double = 0
    display.limitHigh double = 10
    display.description string = "Counter"
    display.units string = "Counts"
    display.form.index int32_t = 0
    display.form.choices string[] = {7}["Default", "String", "Binary", "Decimal", "Hex", "Exponential", "Engineering"]
    control.limitLow double = 0
    control.limitHigh double = 10
    valueAlarm.lowAlarmLimit double = 2
    valueAlarm.lowWarningLimit double = 4
    valueAlarm.highWarningLimit double = 6
    valueAlarm.highAlarmLimit double = 8
```





View Secure Results in Wireshark

- Both **tcp** and **tls** are included in search message
- **TLS Handshake** takes place
- PVA session is encrypted, and we see the decrypted message thanks to the key log.



View Secure Results in Wireshark

- Both **tcp** and **tls** are included in search message
- **TLS Handshake** takes place
- PVA session is encrypted, and we see the decrypted message thanks to the key log.

EPICS Security Technical Plan
Osprey DCS

- Thank You

George McIntyre
george@level-n.com