Änderung mit C_10802

Änderung in gemSpec_Aktensystem

Es wird der neue Abschnitt "5.8. Tracing in Nichtproduktivumgebungen" hinzugefügt.

1.1 5.8. Tracing in Nichtproduktivumgebungen

Ein gewonnener Erfahrungswert ist, dass es für die Fehlersuche in Nichtproduktivumgebungen -- insbesondere bei IOP-Problemen zwischen Produkten verschiedener Hersteller in einer fortgeschrittenen Entwicklungsphase -- leistungsfähigere Mechanismen als zuvor geben muss. Gab es zunächst nur die Testschnittstelle ([gemKPT_Test#A_21193-*]) in den ePA-Clients, so wird mit ePA 2.0 ein Tracing im Aktensystem für Nichtproduktivumgebungen eingeführt.

Dieses Tracing kann man in zwei fachliche Teile untergliedern:

1. Innerhalb des AS werden an die für die Fehlersuche in Nichtproduktivumgebungen wichtigen Stellen Sensoren platziert. Diese Sensoren streamen die aktuell transportierten Daten an bestimmte TCP-Ports am ZGdV. Die Sensorpunkte liegen im AS immer hinter der TLS-Entschlüsselung. Fehlersuchende können sich zu diesen TCP-Ports am ZGdV verbinden und lesen dann im Read-Only-Modus den aktuellen Datenverkehr, der an den Sensorpunkten vorbei fließt, mit.
2. ePA-Clients müssen in Nichtproduktivumgebungen beim VAU-Protokoll durch [gemSpec_Krypt#A_21888-*] definiert fest vorgegebene ECDH-Schlüssel verwenden.

Damit wird es insbesondere möglich für die Fehlersuche in Nichtproduktivumgebungen den Datenverkehr zwischen ePA-Clients und VAU-Instanzen mitzulesen. Für ePA 2.0 konzentriert sich das Tracing auf genau diese Verbindungsstrecke, andere Sensorpunkte im AS sind optional.

Die durch die Spezifikation vorgegebene Architektur eines AS geht davon aus, dass das AS in Komponenten unterteilt ist, zwischen denen die Kommunikation auf TCP-Ebene stattfindet. An vielen Stellen ist diese Kommunikation über TLS gesichert, an einigen nicht. Die Dokumentenverwaltung hat eine HTTPS-Schnittstelle, die TLS-Sicherung endet jedoch vor den VAU-Instanzen. In den VAU-Instanzen möchte man die Trusted Computing Base (TCB) minimieren und setzt dort das VAU-Protokoll als extrem reduziertes TLS-Analogon ein. Ein Sensorpunkt MUSS auf der TCP-Strecke zwischen TLS-Terminierung in der Dokumentenverwaltung und den VAU-Instanzen liegen.

A_21887 - Tracing, Sensorpunkt nahe vor den VAU-Instanzen (Nichtproduktivumgebungen)
Ein Aktensystem MUSS sicherstellen, dass genau in Nichtproduktivumgebungen der Datenverkehr zur und von den VAU-Instanzen auf TCP-Ebene mitgeschnitten wird (Sensorpunkt). Dieser Mitschnitt MUSS in Form eines PCAP-Files kodiert werden (tcpdump). Der aktuell mitgeschnittene Datenverkehr MUSS auf einen TCP-Port im ZGdV gestreamt werden (vgl. [gemSpec_Zugangsgateway_Vers#A_21890-*]. D. h. wenn ein Client sich zu diesem TCP-Port verbindet, MUSS er im Read-Only-Modus die aktuell auf dem Interface durchlaufenden Daten im PCAP-Format gestreamt lesen können. Dieser

Sensorpunkt MUSS für Lasttests temporär per Konfiguration im Aktensystem abschaltbar sein.
[<=]

Hinweis: die gematik stellt auf Anfrage einen Docker-Container als Demonstrator zur Verfügung.

A_21891 - Tracing, alternative Sensordaten-Kodierung
Ein Aktensystem KANN bei der Umsetzung von A_21887-* mit der gematik ein anderes Kodierungsverfahren des Mitschnitts vereinbaren.**[<=]**

Änderung in gemSpec_Krypt

Es wird der neue Abschnitt "6.13. Tracing in Nichtproduktivumgebungen" hinzugefügt.

1.2 6.13. Tracing in Nichtproduktivumgebungen

Für die Fehlersuche -- insbesondere bei IOP-Problemen zwischen Produkten verschiedener Hersteller in einer fortgeschrittenen Entwicklungsphase -- hat es sich als notwendig erwiesen, dass ein Fehlersuchender den Klartext der Kommunikation zwischen ePA-Client und VAU-Instanz mitlesen kann. Vgl. [gemSpec_Aktensystem#5.8. Tracing in Nichtproduktivumgebungen].

A_21888 - VAU-Client, Nichtproduktivumgebung, vorgegebene ECDH-Schlüssel
Ein Client im VAU-Protokoll MUSS in einer Nichtproduktivumgebung von A_16883-* abweichen in dem er sein ephemeres ECDH-Schlüsselpaar nicht zufällig erzeugt, sondern folgendes ECDH-Schlüsselpaar (auf der Kurve BrainpoolP256r1) für jeden VAU-Protokollverbindungsaufbau verwendet:

- privater Schlüssel:
  d=0x78ece94904fc6bf9900cced4c5af4dcd4fbebc585b2054b636cc3f76333bdee
- öffentlicher
  Schlüssel: x=0x21fafedc8ba1ef5477995a28a9794f86355df305f1f58afc88f87e91c664353a
  y=0x27803e3dcf670bd305f3b923f915bfe119389d869c3565828c89bd422231f9f4

[<=]

Hinweis-1: SHA-256("gemSpec_Krypt: privater VAUClient-ECDH-Schluessel fuer Nichtproduktivumgebungen nach A_21888-*") ist als hexdump kodiert gleich 078ece94904fc6bf9900cced4c5af4dcd4fbebc585b2054b636cc3f76333bdee

Hinweis-2:

Folgend ist der private Schlüssel aus A_21888-* als PKCS8-Container PEM-kodiert aufgeführt:

```
-----BEGIN PRIVATE KEY-----
MIGIAgEAMBQGByqGSM49AgEGCSskAwMCCAEBBwRtMGsCAQEEIAeOzpSQT8a/mQDM
7Uxa9NzU++vFhbIFS2Nsw/djM73uoUQDQgAEIfr+3Iuh71R3mVooqXlPhjVd8wXx
9Yr8iPh+kcZkNTongD49z2cL0wXzuSP5Fb/hGTidhpw1ZYKMib1CIjH59A==
-----END PRIVATE KEY-----
```

Erläuterung: für einen Konnektorhersteller erleichtert die Konfigurierbarkeit die Sicherheitsevaluierung. A_21977-* ist eine KANN-Anforderung, d. h. ein Hersteller eines VAU-Client kann entscheiden, ob er A_21977-* umsetzen möchte oder nicht.

A_21977 - VAU-Client, Nichtproduktivumgebung, vorgegebene ECDH-Schlüssel, optionale Konfigurierbarkeit
Ein Client im VAU-Protokoll KANN in einer Nichtproduktivumgebung die Umsetzung von A_21888-* durch einen Nutzer (FdV) oder Administrator (Konnektor etc.) konfigurierbar machen, d. h. das Verhalten des Clients zwischen zufälliger Schlüsselerzeugung und festen Schlüssel aus A_21888-* per Konfiguration veränderlich machen.**[<=]**

A_21889 - VAU-Server, Produktivumgebung, Ablehnung der ECDH-Client-Schlüssel nach A_21888-*

Ein Server im VAU-Protokoll MUSS genau in der Produktivumgebung (PU) bei einem VAU-Handshake den Client-ECDH-Schlüssel aus A_21888-* ablehnen. D. h. er muss falls ein Client im VAUClientHello den öffentlichen Schlüssel aus A_21888-* präsentiert, den Handshake mit einem VAUServerError gemäß A_16851-* abbrechen.**[**<=**]**

Änderung in gemSpec_Zugangsgateway_Vers

Es wird der neue Abschnitt "4.8. Tracing in Nichtproduktivumgebungen" hinzugefügt.

1.3 4.8. Tracing in Nichtproduktivumgebungen

Für die Fehlersuche -- insbesondere bei IOP-Problemen zwischen Produkten verschiedener Hersteller in einer fortgeschrittenen Entwicklungsphase -- hat es sich als notwendig erwiesen, dass ein Fehlersuchender den Klartext der Kommunikation zwischen ePA-Client und VAU-Instanz mitlesen kann. Vgl. [gemSpec_Aktensystem#5.8. Tracing in Nichtproduktivumgebungen].

Das ZGdV stellt Informationen über die aktuell verfügbaren Sensorpunkte im AS bereit. Weiterhin exponiert das ZGdV die Daten der Sensorpunkte über TCP (i. s. v. nicht TLS-gesichert) bspw. über Port 8001,...,8009. Die dort von den Sensorpunkten ge-streamten Daten sind nur Testdaten, also keine Echtdaten, d. h. es sie haben keinen Schutzbedarf bez. Vertraulichkeit. Um die exponierten Sensordaten-Punkte vor DoS-Angriffen zu schützen, erlaubt das ZGdV im Fall der Fälle die TCP-Ports auf IP-Layer über Firewall-Regeln abzusichern.

A_21890 - Zugangsgateway des Versicherten, Sensorpunkt für Nichtproduktivumgebungen
Die Komponente Zugangsgateway des Versicherten (ZGdV) MUSS genau in Nichtproduktivumgebungen:

- die Daten der Sensorpunkte des Aktensystems auf TCP-Ports (bspw. ab Port 8000) öffentlich ohne TLS-Sicherung im Internet zur Verfügung stellen, indem die aktuell an den Sensorpunkten auflaufenden Daten auf dem TCP-Port am ZGdV öffentlich gestreamt werden.
- die Möglichkeit bieten, den Zugriff auf diese TCP-Ports durch Firewall-Einstellungen auf IP-Layer zu beschränken (Initial gibt es keine Beschränkungen).

Weiterhin MUSS das ZGdV über die URL /tracingpoints Informationen über die aktuell im AS verfügbaren und damit auch im ZGdV öffentlich exponierten Sensorpunkt-Daten als JSON-Array (=> Response-Type 'application/json) der folgenden Form bereitstellen:
[{ "name" : "direkt vor der VAU RZ1/B1", "port" : 8001 }, { "name" : "VAU RZ2/B1", "port" : 8002 }, ...]
Sollten keine Sensorpunkte aktuell im AS aktiviert sein (bspw. bei Lasttests) so ist das Array leer: [ ].
Die einzelnen Felder des Arrays sind associative array (oder auch maps oder dictionaries genannt). Diese KÖNNEN neben "name" und "port" beliebige vom AS definierbare, weitere key-value-Paare enthalten. Der Eintrag "port" gibt an, an welchem öffentlich erreichbaren TCP-Port des ZGdV die Sensordaten des entsprechenden Sensors abrufbar sind (gestreamt werden).**[<=]**

Änderung in gemKPT_Test

...

A_21193-01 - ePA-Frontend des Versicherten mit Testtreiber - grafische Logausgabe

Der Hersteller eines ePA-Frontend des Versicherten MUSS im Rahmen der Bereitstellung einer App mit Testtreiberschnittstelle nach [gemSpec_ePA_FdV] auf dem Testgerät eine grafische Logausgabe mit folgenden Informationen bereitstellen:

~~· TLS Secret~~
~~o Ein Logeintrag je Aufbau einer TLS-Verbindung zum Zugangsgateway des Versicherten~~
~~o Der Inhalt eines Logeintrags muss den Vorgaben in [NSS_Key_Log_Format] entsprechen.~~
~~· VAU Secret~~
~~o VAU Schlüsselmaterial bei Aufbau eines VAU-Kanals mit einer Dokumentenverwaltung~~
· TSL-Update
o Ergebnis (success/failed)
o Element TSLSequenceNumber der aktiven TSL (nach der Aktualitätsprüfung / dem Update)
o Erstes Element TSLLocation der aktiven TSL (nach der Aktualitätsprüfung / dem Update)
· Zertifikatsprüfung (z.B. bei TLS-Handshakes und VAU-Verbindungsaufbauten)
o Seriennummer des Zertifikats
o Ergebnis der Zertifikatsprüfung (success/failed)
o OCSP-Status (good/revoked/unknown/unavailable)
· Operation
o <Interface>::<Operationsname>
o Ergebnis der Operation (success/failed)

Ein Eintrag "Operation" enthält jeweils eine technische Operation der Komponenten Authentisierung, Autorisierung, Dokumentenverwaltung, SigD, SGD1, SGD2 oder VZD. Jeder Logeintrag MUSS mit einem lesbaren Zeitstempel (Datum und Uhrzeit) beginnen.
<=

Hinweis:
Der OCSP-Status "unavailable" soll genutzt werden, falls aus irgendeinem Grund kein OCSP-Status ermittelt werden konnte, z.B. Nicht-Erreichbarkeit des OCSP. Die anderen drei Statuswerte entsprechen dem, was ein OCSP-Responder zurückliefern kann.
<=

~~**A_21194 - ePA-Frontend des Versicherten mit Testtreiber - Format einer Logausgabe für ein VAU Secret**~~
~~Die Ausgabe eines Logeintrags für ein VAU Secret nach A_21193 MUSS wie folgt aussehen:~~
~~In eine Zeile werden die drei aus einer HKDF-Schlüsselableitung erzeugten AES-Schlüssel für die KeyId, AES-256-GCM-Key-Client-to-Server und AES-256-GCM-Key-Server-to-Client als Hexadezimalstring geschrieben (AES-Schlüssel siehe [gemSpec_Krypt], A_16943-01). Die drei Werte sind getrennt durch ein Leerzeichen in dieser Reihenfolge abzulegen.~~
~~Je Aufbau einer VAU-Verbindung zwischen Client und Dokumentenverwaltung ist entsprechend eine Log-Zeile auszugeben.~~
~~<=~~

Änderungen in gemProdT_..._PTVx.y.z-n

Anmerkung: Die Anforderungen der folgenden Tabelle stellen einen Auszug dar und verteilen sich innerhalb der Tabelle des Originaldokuments [gemProdT_...]. Alle Anforderungen der Tabelle des Originaldokuments, die in der folgenden Tabelle nicht ausgewiesen sind, bleiben unverändert bestehenden.

Tabelle 1: Anforderungen zur funktionalen Eignung "Produkttest/Produktübergreifender Test"

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
	[…]	