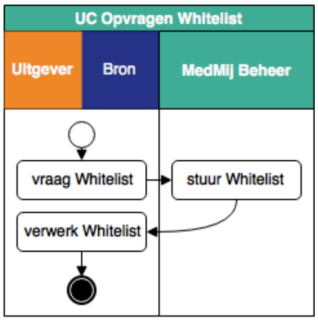
UCI Parsen Whitelist



Programmering talen: C#, Java, Node.js, Python, PHP, Ruby,
Complexiteit: *) ++
Verwachte implementatie tijd: 2 – 4 u
Oplevering: Coding Standard volgens Blauwdruk Design Template (zie Appendix)
Deployment: Library in een van Programmering talen
Testing: XML direct access

Opdracht context:

MedMij Beheer beheert een *Whitelist*. De *Whitelist* beschrijft welke *Nodes* (zie de Applicatielaag) MedMij-verkeer mogen afhandelen.

Rollen

1. In het MedMij-netwerk *functioneert:*

- *elke* PGO Server, *met inbegrip van zijn* OAuth-*rol*, *op één of meerdere* PGO Nodes. *Voor frontchannel-verkeer gebruikt elke* PGO Server *één* PGO Node, *en wel met een hostname die voor die* PGO Server *voorkomt op de* OAuth Clientlist.
- *elke* Authorization Server, *met inbegrip van zijn* OAuth-*rol, op één* ZA Node*;*
- *elke* Resource Server, *met inbegrip van zijn* OAuth-*rol, op één* ZA Node*;*
- *precies één* MedMij Stelselnode, *waarop* MedMij Registratie *functioneert.*

2. Op één:

PGO Node *functioneert één* PGO Server*;*

ZA Node *kunnen meer dan één* Authorization Server *en/of meer dan één* Resource Server *functioneren.*

3. Een of meerdere PKIoverheid TSP*'s treden op als* PKIoverheid TSP.

Toelichting

De getalsverhouding tussen *Servers* en *Nodes* is gespiegeld tussen het persoonsdomein (één-op-meer) en het zorgaanbiedersdomein (meer-op-één). Dat komt doordat er twee lijsten aan de orde zijn die in in tegengestelde richting een vertaling maken: de *OAuth Clientlist* vertaalt **van** hostnames, de *Zorgaanbiederslijst* juist **naar** hostnames. Om deze vertalingen te kunnen laten slagen moet er bij elke *PGO Node* één *PGO Server* horen, en (andersom) bij één *Authorization Server* of één *Resource Server* dus één *ZA Node*.

Het is dus mogelijk voor een *PGO Server* om verschillende certificaten te hanteren voor frontchannel- en backchannel-verkeer, zolang op de *OAuth Clientlist* maar de hostname in het certificaat voor frontchannelverkeer voorkomt die tevens voorkomt in de redirect URI inzake OAuth. Want laatstgenoemde wordt gebruikt door de Authorization Server ten behoeve van de toestemmingsvraag (in UCI Verzamelen) en de bevestigingsvraag (in UCI Delen).

Zie tevens verantwoordelijkheid 5 op de pagina Gegevens en performance in UCI Verzamelen en UCI Delen.

Er is precies één *MedMij Stelselnode* in het *MedMij-netwerk*. Zonder die *MedMij Stelselnode* is er geen *MedMij-netwerk*.

- *In lijn met keuzes op de Proces- en Informatielaag, treden in het zorgaanbiedersdomein alleen de* ZA Nodes *op in het* MedMij-netwerk. *Dat wil zeggen dat bijvoorbeeld achterliggende xIS'en niet over het* MedMij-netwerk *communiceren met de* ZA Node. *Dat verkeer is verborgen achter de* ZA Node. *Alle daarvoor benodigde routering wordt afgehandeld door de server-implementaties en speelt zich buiten het zicht van het MedMij Afsprakenstelsel af.*

Functie Server Authorization

Verspreiding van de *Whitelist*

1. De MedMij Stelselnode *biedt aan* PGO Node *en* ZA Node *een use case-implementatie (*UCI Opvragen Whitelist*) om de actuele versie van die* Whitelist *op te vragen. Betrokken rollen gebruiken hiervoor het betreffende stroomdiagram.*

2. Het aandeel van de MedMij Stelselnode *in* UCI Opvragen Whitelist *is voor minstens 99,9% van de tijd beschikbaar.* MedMij Registratie *laat, na het niet beschikbaar raken van het aandeel van* MedMij Stelselnode *in de use case, maximaal acht uren (4800 minuten) verstrijken voordat het weer beschikbaar is.*

3. PGO Nodes *en* ZA Nodes *betrekken minstens elke vijftien minuten (900 seconden) de meest recente* Whitelist *van* MedMij Stelselnode.

4. De MedMij Stelselnode *heeft de hostname PLACEHOLDER. De* MedMij Stelselnode *staat niet* op *de* Whitelist*, maar wordt er voor de controle tegen de* Whitelist *wel geacht op te staan.*

Toelichting

Door op deze manier de MedMij Stelselnode te autoriseren voor MedMij-verkeer wordt ervoor gezorgd dat ook in foutsituaties of bootstrap-situaties een PGO Node of ZA Node de MedMij Stelselnode kan aanspreken om een Whitelist op te halen.

5. PGO Nodes en ZA Nodes *valideren elke nieuw verkregen* Whitelist *tegen het XML-schema van de Whitelist. Dit XML-schema is een technische implementatie van het MedMij-metamodel.*

6. Ten behoeve van de technische beveiliging van het gegevensverkeer dat zich voltrekt in het kader van UCI Opvragen Whitelist *maakt deze gebruik van* Versleuteling, Server Authentication *en* Server Authorization*, volgens het bepaalde op deze Netwerk-laag.*

Gebruik van de whitelist

7. ZA Node, PGO Node en MedMij Stelselnode *laten, elk hunnerzijds, backchannel-verkeer over het* MedMij-netwerk *dan en alleen dan doorgang vinden, nadat zij hebben vastgesteld dat de hostname van de andere* Node, *waarmee verbinding gemaakt zou worden, op de meest actuele* Whitelist *voorkomt.*

Toelichting

In geval van frontchannel-verkeer vindt er geen Server Authorization plaats.

8. De Node *die*

- de TLS-client zou worden voert de in verantwoordelijkheid 13 bedoelde controle tegen de *Whitelist* uit voorafgaand *aan de start van de TLS-handshake. Indien die controle niet kan worden uitgevoerd, of een negatief resultaat oplevert, wordt de TLS-handshake niet gestart.*

- de TLS-server is, voert de in verantwoordelijkheid 13 bedoelde controle tegen de Whitelist *uit tijdens de TLS-handshake, en wel onmiddellijk voorafgaand aan de voorziene verzending van de Finished message. Indien die controle niet kan worden uitgevoerd, of een negatief resultaat oplevert, wordt in plaats van de Finished message de uitzondering access_denied verzonden. In dit geval slaagt de controle tegen de* Whitelist *dan en slechts dan als op de* Whitelist *tenminste een van de volgende namen uit het de door de TLS-client aangeboden certificaat voorkomen: de Common Name of een van de eventuele Subject Alternative Names.*

Toelichting

In geval van uitgaand verkeer kan de voorziene TLS-client de controle tegen de Whitelist al uitvoeren voordat hij de TLS-handshake initieert, omdat hij de voorziene TLS-server al heeft geïdentificeerd, om te weten wie hij überhaupt moet aanspreken. In geval van inkomend verkeer echter, kan de TLS-server de zich aandienende TLS-client pas identificeren gedurende de TLS-handshake, aan de hand van het certificaat dat hij, conform verantwoordelijkheid 1b, moet ontvangen. Daarop moet een hostname voorkomen die op de Whitelist is terug te vinden. Door toe te staan dat niet alleen de Common Name de voor MedMij geautoriseerde hostname mag bevatten, maar ook een Subject Alternative Name, biedt het MedMij Afsprakenstelsel aan deelnemers de mogelijkheid tot hergebruik van certificaten voor meerdere MedMij-nodes, of voor meerdere doelen dan alleen deelname in MedMij.

Wanneer de Whitelist wordt geraadpleegd gedurende de TLS-handshake, vraagt dat in de implementatie van de TLS-handshake mogelijk een extra stap ten opzichte van sommige standaard-implementaties. Daarom zijn alternatieven overwogen voor de Whitelist-controle in geval van inkomend verkeer. Eén alternatief is om de Whitelist-controle te laten plaatsvinden na afloop van een (succesvolle) TLS-handshake, maar dat introduceert een beveiligingsrisico, omdat na een succesvolle TLS-handshake ook al inhoudelijk gegevensverkeer kan plaatsvinden, mogelijk dus on-geautoriseerd. Bovendien zou deze variant een MedMij-specifiek autorisatieprotocol introduceren, terwijl de internationale en open TLS-standaard, door middel van de foutmelding access_denied, deze functionaliteit al biedt. Een andere overweging zou nog zijn de Whitelist-controle te verplaatsen naar de Applicatie-laag, maar

dat zou weinig mogelijkheden bieden tot hergebruik en tot extra complexiteit leiden in zowel implementatie als onderhoud van het MedMij Afsprakenstelsel.

De foutmelding access_denied wordt besproken in sectie 7.2.2 van de TLS-specificatie.

15. Indien een Whitelist*-controle, in het kader van verantwoordelijkheid 14, niet kan worden uitgevoerd, of een negatief resultaat oplevert, breekt dit de voortgang af van de uitvoering van de use case-implementatie en wordt deze uitzondering behandeld als ware het de eerstvolgende inhoudelijke uitzondering conform de tabellen met uitzonderingen op UCI Verzamelen, respectievelijk UCI Delen, met dien verstande dat de betrokken Applicatie-rollen elkaar hiervan niet op de hoogte stellen.*

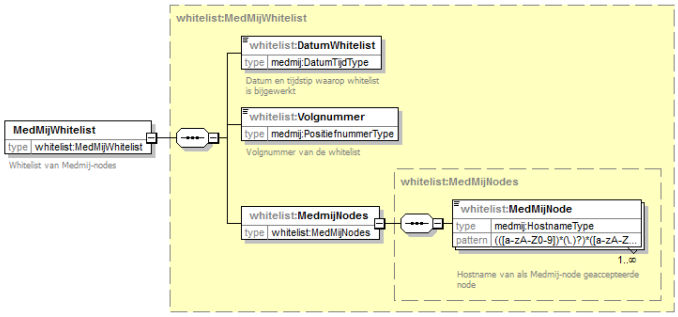
Toelichting

Zo krijgt een uitzondering op Netwerk-niveau ook betekenis op Applicatie-niveau. Omdat het niet slagen van de Whitelist*-controle duidt op een niet te vertrouwen tegenpartij, wordt deze daarvan niet op de hoogte gesteld.*

Domain Name System

16. Elke Dienstverlener Persoon*, elke* Dienstverlener Zorgaanbieder *en* MedMij Beheer *dragen ervoor zorg, in zijn rol als DNS Server, of cliënt daarvan, in het publieke Domain Name System, inzake de hostnames van de* MedMij Nodes*, respectievelijk* MedMij Stelselnode*, waarvoor hij verantwoordelijk is, dat de name records behorende bij die hostname zijn ondertekend volgens DNSSEC.*

17. De MedMij Stelselnode *en elke* MedMij Node*, in zijn rol als DNS resolver in het Domain Name System, controleert of de ontvangen name records zijn voorzien van ondertekening volgens DNSSEC en valideert deze volgens DNSSEC. Indien deze controle en validatie niet beide slagen, ziet hij af van verbinding met de betreffende hostname.*



MedMij Beheer zal de *Whitelist* steeds aanpassen wanneer:

- een *Dienstverlener* zijn deelname aan het MedMij Afsprakenstelsel aangaat of beëindigt;

- er gedurende haar deelname veranderingen optreden in de *Whitelist*-gegevens die op haar betrekking hebben.

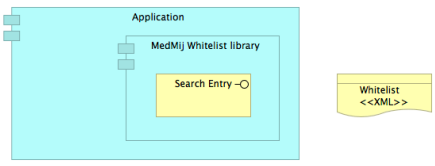
Opdracht beschrijving:

MedMij Beheer biedtaan Uitgever een use case (UC Opvragen Whitelist) om de actuele versie van die Whitelist op te vragen: Opvragen Whitelist.

De trusted party list van deelnemers op het netwerk van MedMij die opgenomen is in het metadatabestand met een XML (en gerelateerde XSD) formaat.

Het verzoek is om deze lijst te kunnen benaderen. De validatie houdt in dat er is mogelijk om een API te definiëren om de XML metadatabestand te benaderen voor zoeken (valideren) naar een specifiek delenemer.

Opdracht 1:
Definieer de API functies nodig om de Whitelist te benaderen
Opdracht 2:
Implementeer de API functies nodig om binnen de Whitelist te zoeken



Opdracht resources:

XML en XSD Files:

Opdracht acceptatie:

Test case 1: Functionele API coverage

Test case 2: Unit Test executie om API validatie te testen

- verschillende XML files
- error handling
- performance voor verschillende type XML files

Test case 3:

- Source code evaluatie