# Bouwen van een PKI voor het Securitylab van Fontys

*Onderzoeken van de werking van een PKI en expertise op doen m.b.t. digitale certificaten*

| | |
|---|---|
| Datum: | 27 mei 2012 |
| Auteur: | Richard Bovens |
| Versie: | 1.9 |

| | |
|---|---|
| Studentnummer: | 2163051 |
| Afstudeerbedrijf: | Fontys Hogeschool |
| Adres: | Rachelsmolen 1, 5612MA, Eindhoven |
| Telefoonnummer: | 088-5080000 |
| Opdrachtgever: | Casper Schellekens |

Wijzigingen

| Datum | Wijzigingen |
|---|---|
| 5 mei 2012 | Hoofdstuk 5 t/m paragraaf 5.1 geschreven |
| 6 mei 2012 | Paragraaf 5.2 geschreven |
| 9 mei 2012 | Hoofdstuk 6 indeling gemaakt |
| 11 mei 2012 | Hoofdstuk 6 geschreven |
| 12 mei 2012 | Hoofdstuk 7 en 8 geschreven |
| 13 mei 2012 | Conclusies en aanbevelingen geschreven, Literatuurlijst en bijlagen geschreven, voorwoord en inleiding geschreven, samenvatting geschreven, spelling en grammatica controle. |
| 22 mei 2012 | Aanpassing in 4.2 gemaakt |
| 24 mei 2012 | Hoofdstuk 6 verwijderd en dit geplaatst onder hoofdstuk 4, controle van certificaten toegevoegd in hoofdstuk 4, bij 4.4 stukje single pki architectuur aangepast, 4.5 aangepast. |
| 25 mei 2012 | 4.6 aangepast, 5.1 aangepast, 5.2 aangepast, hoofdstuk Diginotar verwijderd en kleine toelichting gegeven hierover bij 4.4, hoofdstuk over eindgebruiker verwijderd en dit opgenomen in hoofdstuk 6. Hoofdstuk 6 geschreven. |
| 27 mei 2012 | Hoofdstuk over eindgebruiker toch weer toegevoegd, samenvatting aangepast, hoofdstuk 2 ingekort tot beschrijving Fontys en opdrachtgever, andere aanpassingen verwerkt n.a.v. commentaar dhr. A. Lak, aanbevelingen in hoofdstuk 8 , summary geschreven, verklarende woordenlijst geschreven. Spelling en grammatica check uitgevoerd. Titel document aangepast. |
| 1 juni 2012 | 4.2 aangepast n.a.v. comment dhr. Lak |

# Voorwoord

De afstudeeropdracht is bij Fontys Hogeschool uitgevoerd. De opdracht was om het certificaten probleem van de servers in het Seclab op te lossen en meer expertise op te doen m.b.t. een PKI en certificaten.

Tijdens het afstudeerproject is Casper Schellekens de bedrijfsbegeleider geweest en is Albert Lak mijn docentbegeleider geweest. Beiden wil ik bij deze bedanken voor hun inzet en begeleiding tijdens het afstudeerproject.

Bij het uitvoeren van de afstudeeropdracht zijn er ook momenten bij geweest waar ik vast liep en heb ik ondersteuning gekregen van Dennis Cools. Dennis is een ICT medewerker bij Fontys Hogeschool en bij deze wil ik Dennis bedanken voor zijn ondersteuning en inzet.

Richard Bovens

27 mei 2012

# Inhoudsopgave

# Samenvatting

Het afstudeerproject is uitgevoerd bij Fontys Hogeschool. Fontys Hogeschool is een stichting die onderwijs en cursussen verzorgd voor HBO en Master opleidingen.

Fontys Hogeschool heeft een Security lab waar de servers een certificaat hebben. Dit certificaat wordt niet vertrouwd, wat de nodige meldingen veroorzaakt als studenten op de servers inloggen. De opdracht was om er voor te zorgen dat de meldingen m.b.t. het certificaat niet meer voorkomen. Daarnaast was het ook de bedoeling om meer expertise op te bouwen m.b.t. een PKI en de certificaten die een CA uitgeeft.

Als eerst ben ik begonnen met het opstellen van een plan van aanpak. Ik vond het belangrijk om eerst de juiste kennis te hebben, voor ik aan de opdracht kon gaan beginnen. Daarna ben ik begonnen om de opdracht uit te voeren in fasen. Uiteindelijk is het mij gelukt om ervoor te zorgen dat de meldingen m.b.t. het certificaat wat niet vertrouwd werd niet meer voor te laten komen. Dit heb ik gedaan door zelf certificaten aan te maken, deze om te zetten naar het bestandsformaat geschikt voor het platform waar de servers op draaien en toen de certificaten op de servers te vervangen.

Tijdens het afstudeerproject heb ik veel geleerd over PKI's en certificaten. Zo blijkt dat het ontwerp van een PKI heel belangrijk is om fouten of beveiligingsproblemen te voorkomen. Ook bij het bouwen en implementeren van een PKI is het belangrijk om zoveel mogelijk te documenteren hoe een PKI gebouwd of geïmplementeerd moet worden.

Bij het opzetten van een PKI is het beter om er voor te zorgen dat certificaat uitgifte door de CA goed geregeld is zodat eindgebruikers hier geen hinder van hebben en niet blijven zitten met de perikelen van het installeren van een certificaat op hun besturingssysteem. Voor publieke CA's of organisaties die certificaten verstrekken is het verstandig om daar goede instructies voor de eindgebruiker die het certificaat moet installeren bij te leveren. Voor eindgebruikers die zelf een certificaat moeten installeren is de hele materie nogal lastig. Veel eindgebruikers hebben er dan ook moeite mee en krijgen dit soms niet voor elkaar. Als de gebruikers het wel voor elkaar krijgen een certificaat te installeren, dan is het certificaat geïnstalleerd en zijn er geen problemen.

Commerciële CA's moeten ook zorgen dat ze goede basis- en netwerkbeveiliging hebben. Bij het onderzoek naar Diginotar is gebleken dat het bij Diginotar aan elementaire beveiliging ontbrak en dat bepaalde apparatuur voor netwerkbeveiliging waarschijnlijk niet goed was geconfigureerd.

Kortom, de PKI als veilig systeem om ervoor te zorgen dat certificaten uitgegeven kunnen worden is nog volledig betrouwbaar, maar het gaat er net als het bouwen van een huis om dat het ontwerp, de bouw en implementatie goed gemaakt, gepland en uitgevoerd wordt.

# Summary

The assignment to get my bachelors degree was an assignment for Fontys Hogeschool. Fontys Hogeschool is an organisation which gives courses for bachelor and master degrees.

Within Fontys Hogeschool there is a security lab where students can work on their projects. In this lab are servers that have a certificate which is not trusted. The certificates are installed by default when installing the servers. When someone logs onto the servers, he or she will get a message that the certificate is not trusted. Besides the fact that is very annoying to get that message several times, it also depicts a security issue, since the certificate is not trusted.

My assignment was to make sure this message doesn't show up anymore and to get more knowledge and expertise when it comes to PKI's and certificates that are issued by PKI's.

The approach I had in mind was to start with a plan how to work this project. First of all I thought it was necessary to gain knowledge about the subject first before actually starting to build a PKI.

In the end I managed to build a PKI, create, convert and issue my own certificates and make sure they were trusted in the whole domain. I installed the certificates on the server and the message about the certificate that was not trusted is gone now.

During the project I learned a lot about PKI's and certificates. Actually a lot of things can go wrong before the PKI is even build. There are still organisations who make mistakes in the design, building and implementation phase which can have very unpleasant surprises in the end. A good example for this is Diginotar.

When building a PKI it is important to make sure that the issuing of certificates is done properly and if possible make sure that end users shouldn't have to install a certificate on their systems. In Windows a lot of this can be configured to be done automatically, so the user doesn't have to install certificates anymore. This was also the case with the certificates I created and installed on the server; the end user doesn't have to install the certificate anymore and he or she can log on to the server without any messages about untrusted certificates popping up again.

The PKI as a secure system to issue certificates had a bit of a popularity issue during the whole incident that happened at Diginotar. I can honestly say that the PKI as a system itself is still reliable but building a PKI is similar to building a house; First you need to design the house before you can build it and implement the rest. That way you can be sure you have a safe house or in this case, a safe PKI system.

# Verklarende woordenlijst

**PKI**

Public Key Infrastructure. Een veilig systeem om certificaten uit te geven.

**CA**

Certification Authority. Het onderdeel van een PKI die binnen een PKI de certificaten uitgeeft, intrekt en waar het beheer van de certificaten plaats vindt. Een CA kan betrekking hebben op een publieke CA of een interne CA in een organisatie.

**RA**

Registration Authority. Het onderdeel van een PKI die de aanvrager van een certificaat verifieert.

**VA**

Validation Authority. Het onderdeel van een PKI wat de certificaten controleert.

**TCO**

Total Cost of Owernship. Hiermee worden vaak de totale kosten bedoeld om iets (een systeem, network, PKI etcetera) te ontwerpen, bouwen, implementeren en beheren.

**Root CA**

De CA die de certificaten uitgeeft binnen een PKI.

**Trusted root for in house PKI**

Oplossing om voor een interne PKI een publieke CA aan te stellen zodat de intern uitgegeven certificaten betrouwbaarder zijn.

**Vcenter server**

Product van Vmware om een datacenter op te zetten en ESX hosts te beheren.

**ESX server**

Een product van Vmware. Op een server wordt ESX als platform geïnstalleerd, waarna er virtuele machines op kunnen worden geïnstalleerd.

# 1 Inleiding

Voor u ligt mijn afstudeerscriptie, een scriptie met een onderzoek naar een Public Key Infrastructure en het uitgeven van certificaten.

De afstudeeropdracht is uitgevoerd voor Fontys Hogeschool. Het probleem bij Fontys Hogeschool is dat zij een Security lab hebben met een aantal servers die meldingen geven over het geïnstalleerde certificaat zodra studenten er verbinding mee maken. Het certificaat op die servers wordt niet vertrouwd, waardoor deze melding steeds verschijnt. Het was de opdracht om te zorgen dat deze melding niet meer voor zou komen en tegelijkertijd expertise op te doen over PKI's en certificaten.

In hoofdstuk 2 kunt u lezen over Fontys hogeschool en krijgt u meer informatie over de organisatie, in hoofdstuk 3 wordt de beginsituatie bij Fontys hogeschool, de opdracht en de opdrachtomschrijving beschreven.

Hoofdstuk 4 beschrijft wat een PKI is en hoe een PKI werkt, in hoofdstuk 5 wordt meer ingegaan op PKI architecturen in de praktijk en waar in de praktijk rekening mee moet worden gehouden wanneer er een PKI wordt opgezet. In hoofdstuk 6 wordt de PKI behandeld vanuit het oogpunt van de eindgebruiker, in hoofdstuk 7 besteed ik aandacht aan de uitgevoerde afstudeeropdracht.

Hoofdstuk 8 bevat de uiteindelijke conclusies en aanbevelingen voor het opzetten van een PKI, beveiliging van de PKI, de certificaten die uitgegeven worden en de eindgebruikers.

# 2 Het bedrijf

Fontys is 1 van de grootste hoger onderwijsinstellingen in Nederland. Fontys Hogeschool heeft zo'n 40.000 studenten, 4.000 personeelsleden en het biedt onderwijs aan in bijna alle sectoren.

In Nederland zijn er 31 Fontys instituten die de opleidingen verzorgen. Er zijn ongeveer 90 bachelor opleidingen in verschillende onderwijsvormen zoals voltijd, duaal, deeltijd en in-service. Ook zijn er 23 mastervarianten en 12 Associate degree opleidingen die door Fontys worden aangeboden.

Ook worden er verschillende cursussen en trainingen verzorgd in de stedelijke centra, voornamelijk in het Zuiden van Nederland.

Naast het aanbieden van opleidingen en cursussen legt Fontys zich ook toe op onderzoek, kennis innovatie en contract activiteiten.

Fontys biedt opleidingen op het gebied van:

- Bedrijfsmanagement en Logistiek
- Economie- Marketing- Rechten
- Engineering
- Gezondheidszorg
- ICT
- Kunsten
- Lerarenopleidingen
- Media- Communicatie
- Mens en Maatschappij
- Natuurwetenschappen
- Sport

## *2.1 Organisatie en bestuur*

Fontys heeft een juridische en interne organisatiestructuur. Net als elke andere organisatie heeft ook Fontys een aantal processen om ervoor te zorgen dat eindproducten (onderwijs) kunnen worden aangeboden en door klanten kunnen worden afgenomen. De organisatiestructuur is opgebouwd rond 3 processen:

- Bestuurlijk proces
- Primair proces
- Ondersteunend proces

**Bestuurlijk proces**
Binnen een bedrijf moeten er dingen geregeld worden, processen moeten aangestuurd worden, er moet controle zijn dat de processen goed worden uitgevoerd. Dit gebeurt allemaal in het bestuurlijk proces.

In het bestuurlijk proces bevindt zich het bevoegd gezag. Het Collega van Bestuur vormt het bevoegd gezag en het bestuur heeft de eind verantwoordelijkheid voor de resultaten.

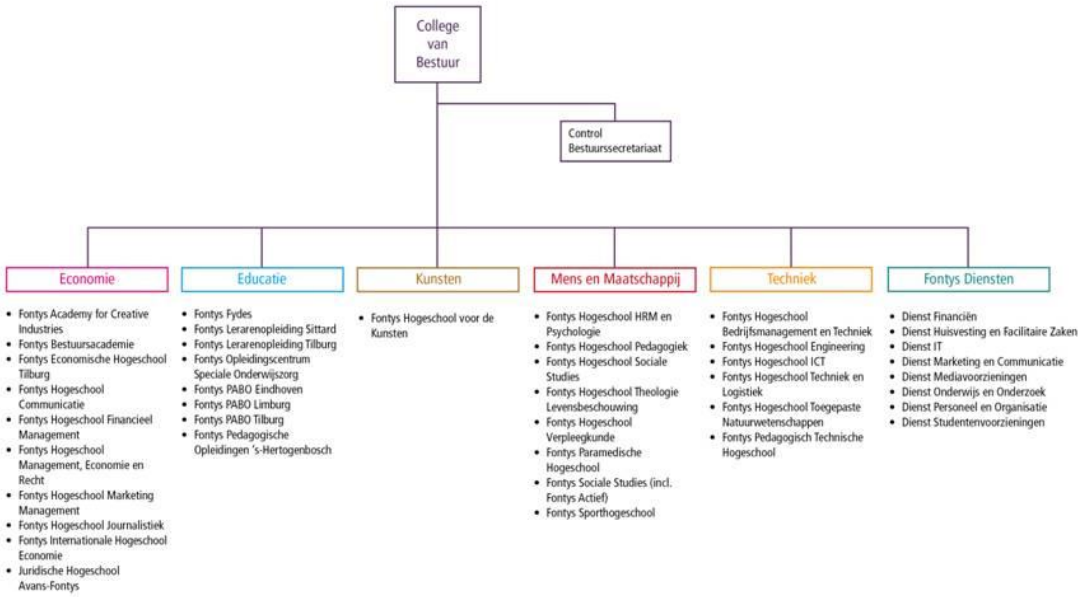
Toezicht en leiding van de Fontys Hogeschool is belegd bij de Raad van Toezicht en het College van Bestuur.

**Primair proces**
In het primaire proces bevinden zich de activiteiten waarmee inkomsten worden verworven. Het primaire proces vindt plaats bij de instituten. De kernactiviteiten in het primaire proces zijn onderwijs, onderzoek en contractactiviteiten.

**Ondersteunend proces**
Met ondersteunend proces worden alle activiteiten bedoeld die niet onder het bestuurlijk of primair proces vallen. Hierbij kan gedacht worden aan boekhouding, administratie, schoonmaak, marketing, personeelszaken et cetera. Niet alleen de instituten, maar ook het College van Bestuur maken gebruik van de Fontys diensten die onder dit proces vallen. Dit zijn de volgende diensten:

- Financiën
- Huisvesting en Facilitaire Zaken
- IT
- Marketing en Communicatie
- Onderwijs en Onderzoek
  Met onderdeel: Juridische Zaken
- Personeel en Organisatie
- Studentenvoorzieningen

**Organogram**
Het onderstaande organogram laat zien hoe de organisatie van Fontys in elkaar steekt. Hier wordt ook duidelijk hoe per gebied de opleidingen/cursussen zijn ingedeeld en waar deze worden aangeboden.



Bron: Fontys Organogram http://www.fontys.nl/over.fontys/organogram.aspx

## *2.2 Opdrachtgever*

De opdrachtgever voor de afstudeeropdracht is de heer Casper Schellekens. Casper Schellekens is een docent en doceert aan de Fontys Hogeschool in Eindhoven.

Fontys Hogeschool heeft een heel mooi securitylab waar studenten hun eigen projecten kunnen uitvoeren. Deze projecten komen voort vanuit de vakken die op Fontys Hogeschool worden gegeven. Hierbij kan men denken aan het installeren van servers, het opzetten van een netwerk of bepaalde testen uitvoeren in het securitylab.

De afstudeeropdracht heb ik uitgevoerd in dit securitylab. Het grote voordeel van het securitylab is dat de studenten er hun projecten in kunnen uitvoeren en dat opdrachten, zoals deze afstudeeropdracht, daar veilig kunnen worden uitgevoerd zonder problemen te kunnen veroorzaken op het eigen netwerk van Fontys.

# 3 Opdracht

## 3.1 Beginsituatie

Het securitylab is een intern, beveiligd netwerk en er worden producten van VMware gebruikt om dit netwerk te beheren. Met Vcenter server is het mogelijk om een netwerk zoals het securitylab te beheren en dit wordt voor het securitylab gebruikt.

In het netwerk zijn 3 ESX servers geïnstalleerd die met Vcenter server beheerd worden. Op de machines kunnen dan virtuele machines worden geïnstalleerd. Studenten kunnen bijvoorbeeld eigen netwerkjes maken en installeren dan bijvoorbeeld Windows servers en clients.

Zodra studenten een verbinding opzetten met Vcenter server krijgen zij een melding dat het certificaat niet is vertrouwd. Wanneer een verbinding opgezet zou worden naar een ESX server door een beheerder van het securitylab, dan krijgt men dezelfde melding.

Op dit moment is er bij Fontys in het securitylab geen eigen Public Key Infrastructure (PKI) die de certificaten kan uitgeven. Met een eigen interne PKI en daarin een root CA kunnen namelijk certificaten zelf worden ondertekend en uitgegeven. Binnen het netwerk kan dan gecontroleerd worden of de certificaten geldig zijn, waardoor de melding die de gebruikers nu krijgen m.b.t. een niet te vertrouwen certificaat verleden tijd is.

## 3.2 Doel van het project

Het doel van het project is om expertise op te bouwen met betrekking tot praktische aspecten van digitale certificaten en onderliggende mechanismen. Daarbij kan gedacht worden aan de werking van uitgifte van digitale certificaten, maar ook de certificaten op zich; hoe zitten deze in elkaar, zijn er standaarden of protocollen? Daarnaast was het doel om ervaring op te doen m.b.t. het vervangen van certificaten en welke eventuele praktische uitdagingen dit met zich meebrengt.

## 3.3 Opdrachtomschrijving

Voor Fontys hogeschool was het de opdracht om te onderzoeken wat een PKI is, wat het precies doet en hoe het werkt. Ook moest er onderzoek gedaan worden naar de hack bij Diginotar en waarom het daar zo goed mis kon gaan.

Voor het Securitylab moest een PKI worden gebouwd. Op die manier heeft Fontys een eigen PKI en kunnen er binnen het eigen netwerk vertrouwde certificaten worden uitgegeven. Zo krijgen de gebruikers geen meldingen meer dat de certificaten niet te vertrouwen zijn.

# 4 Wat is een PKI en hoe werkt het?

## 4.1 PKI en cryptografie

Voordat dieper ingegaan wordt op wat een PKI is, wordt eerst aandacht aan het onderliggend cryptografiesysteem besteed. PKI staat voor Public Key Infrastructure en het maakt gebruik van een publieke en privé sleutel. Dit wordt ook wel een asymmetrisch cryptosysteem genoemd. De privé sleutel blijft altijd geheim en de publieke sleutel wordt gepubliceerd zodat deze gebruikt kan worden.

Het PKI systeem werkt goed, zolang de PKI voldoende beschermd is. Het is belangrijk dat niemand de privé sleutel in handen krijgt, omdat het dan mogelijk is om uit naam van de partij waar het sleutelpaar van is, valse certificaten uit te geven. De gebruiker moet ook zeker er van zijn dat hij/zij de juiste publieke sleutel heeft.

## 4.2 Basiswerking van een PKI

**Basiswerking van een PKI**

Een PKI is ontstaan uit de behoefte aan vertrouwen en integriteit. Voor er PKI's waren, was het moeilijk om vast te stellen of een bepaalde persoon of partij wel diegene was waarvan hij of zij zegt/zeggen dat die het zijn.

Binnen een PKI worden certificaten gemaakt en uitgegeven om het vertrouwen in een persoon of partij en de integriteit van de data te waarborgen. Het uitgeven van certificaten wordt door de Certificate Authority (CA) gedaan. Aan de hand van het certificaat kan worden gecontroleerd of een bepaalde persoon of partij te vertrouwen is. Dat is namelijk waar certificaten voor dienen; de publieke sleutel die in het certificaat is opgeslagen maakt de certificaatbezitter bekend. De certificaatbezitter is bij de CA geregistreerd als zijn de bezitter en zo kan het certificaat vertrouwd worden.

Om ervoor te zorgen dat een PKI betrouwbaar blijft, worden certificaten ook ingetrokken. Hiervoor kan een bepaalde reden ten grondslag liggen, waarna het certificaat door de CA wordt ingetrokken. De basiswerking van een PKI is dan ook gebaseerd op het uitgeven en intrekken van certificaten.

**Uitgeven van certificaten**

Het uitgeven van een certificaat wordt door de CA gedaan. Bij goedkeuring van een aanvraag voor een certificaat wordt door de CA een certificaat gegenereerd en wordt het certificaat uitgegeven. Het uitgegeven certificaat wordt dan naar de gebruiker opgestuurd en in de openbare database geregistreerd. De gebruiker kan nu het certificaat installeren en gebruiken. De uitgegeven certificaten worden ook in Authority Information Access (AIA, zie voetnoot 35) locaties opgeslagen, zodat de certificaten beschikbaar zijn voor clients.

Het certificaat kan nu ook door andere personen of partijen worden gecontroleerd aan de hand van de publieke sleutel die in het certificaat staat aangegeven. Als het certificaat en het certificering pad klopt (het pad wat terugvoert naar de uitgevende CA, de root CA genoemd), dan is het certificaat te vertrouwen.

**Intrekken van certificaten**

Het kan voorkomen dat een certificaat moet worden ingetrokken, hiervoor kan een bepaalde reden ten grondslag liggen. Een paar redenen zouden kunnen zijn:

- De CA stopt er mee (zoals bij Diginotar het geval is geweest)
- Er zijn wijzigingen m.b.t. gegevens van de certificaathouder
- De certificaathouder heeft zich met een valse identificatie geïdentificeerd
- Er is door de certificaathouder een verzoek ingediend om het certificaat in te trekken

De ingetrokken certificaten worden in een database geregistreerd en deze informatie wordt ook publiekelijk verspreid. Zo is iedereen op de hoogte van de ingetrokken certificaten en kan men altijd nog zelf beslissen of men een bepaald persoon of een partij wil vertrouwen. Een goed voorbeeld hiervan is een website waarbij een niet vertrouwd certificaat is geïnstalleerd; men kan de website dan links laten liggen of de website toch vertrouwen en doorgaan met het bezoeken van de website.

*CRL's en Delta CRL's*

Ingetrokken certificaten komen op een Certificate Revocation List (CRL) te staan. In eerste instantie is er 1 basis versie van een CRL. Wanneer er meerdere certificaten worden ingetrokken, worden er updates aangeboden. Een gewone CRL is al snel heel erg groot in het aantal MB's, dit is dan ook de reden waarom de Delta CRL's in het leven zijn geroepen. Dit zijn kleinere CRL's die de updates bevatten van certificaten die na de publicatie van de gewone CRL zijn ingetrokken.

De CRL's moeten ook worden aangeboden aan de gebruikers van een PKI. De CRL's worden verspreid via Certificate Distribution Points (CDP). Via ingestelde CDP's worden deze via het http of LDAP protocol verspreid naar de gebruikers. Het voordeel van het verspreiden van de CRL's via het http protocol, is dat het besturingssysteem onafhankelijk is. Of de gebruiker nu een Windows, Linux of Unix machine gebruikt, de updates kunnen altijd worden binnengehaald met het http protocol. Ook het LDAP protocol heeft voordelen; zo kan m.b.v. LDAP met certificaat beveiligde computers onderling communiceren maar kunnen de CRL's ook heel gemakkelijk aan computers en gebruikers binnen Active Directory worden verspreid.

*OCSP*

Het nadeel aan CRL's is dat het lang kan duren voordat iedereen op de hoogte is als er certificaten zijn ingetrokken. Base CRL's zijn behoorlijk groot, daarom zijn Delta CRL's in het leven geroepen voor de updates van ingetrokken certificaten. Deze CRL's moeten door clients gedownload worden, wat soms erg lang kan duren en een nadelig effect heeft op de bandbreedte van het netwerk. Om deze redenen is het Online Certificate Status Protocol (OCSP) bedacht.

Met OCSP[1] wordt er een verbinding gemaakt met de database waar het certificaat staat vermeld en kan realtime worden gecontroleerd of dit certificaat nog geldig is. OCSP wordt veel gebruikt; zo

[1] IETF, RFC voor het OCSP protocol http://www.ietf.org/rfc/rfc2560.txt

wordt het namelijk door een aantal van de grootste CA's op het internet gebruikt zoals VeriSign, Globalsign en Thawte. Voor het besturingssysteem Windows heeft het tot 2008 geduurd voordat Microsoft OCSP implementeerde. In 2008 implementeerde Microsoft OCSP in het Windows Server 2008 product. Microsoft heeft uiteindelijk OCSP geïmplementeerd omdat er veel vraag naar was en met die implementatie zal het verwachte gebruik van OCSP waarschijnlijk gestegen zijn.

Diginotar maakte ook gebruik van OCSP, maar dit heeft niet het gewenste effect gehad dat van OCSP verwacht wordt. Met OCSP wordt realtime gecontroleerd of certificaten geldig zijn en in het geval van Diginotar heeft de OCSP responder wel zijn werk gedaan, maar werden valse certificaten gewoon als geldig gezien door de OCSP responder. Dit komt omdat de OCSP responder zijn informatie m.b.t. validiteit van het certificaat direct uit de database haalt en de valse certificaten door de root CA van Diginotar zelf waren ondertekend en in de database voor uitgegeven certificaten waren opgeslagen, dus werden vertrouwd. In een filmpje op Youtube[2] is te zien hoeveel OCSP verzoeken er werden gedaan aan de OCSP responder voor het vals uitgegeven certificaat voor het gehele Google domein.

Om PKI's nog veiliger te maken en het checken van ingetrokken certificaten en de geldigheid van certificaten te verbeteren, wordt er gewerkt aan een nieuw protocol. Dit protocol heet DANE.

*DANE*

Om ervoor te zorgen dat een dergelijke fout als bij Diginotar niet meer gebeurt, kan DNS-based Authentication of Named Entities (DANE) een goede aanvulling zijn op het bestaande Domain Name System Security Extensions (DNSsec) protocol.[3] Het DNSsec protocol voegt beveiliging toe aan het bestaande Domain Name System (DNS) protocol.[4] DNS wordt gebruikt om namen om te zetten in IP adressen en vice versa. Hier is later DNSsec aan toegevoegd, omdat er bij DNS op zichzelf geen beveiliging was ingebouwd.

DANE staat namelijk los van de betreffende website en het zou een betere controle kunnen zijn of de host een geldig certificaat heeft of niet. Naast de informatie over het certificaat krijgt de gebruiker ook extra informatie over de hostname en eventuele andere certificaten die er voor zijn uitgegeven. Op dit moment is een werkgroep bezig met het ontwikkelen van DANE. DANE is nog in ontwikkeling en een verwachting over wanneer DANE wordt geïmplementeerd is nu niet te schetsen. De laatste draft versie van DANE is gemaakt op 17 mei 2012. Over DANE kan meer informatie worden gevonden op de pagina van het Internet Engineering Task Force.[5]

[2] Youtube, OCSP serial requests for rogue *.google.com certificate (part of Operation Black Tulip of Fox-It)

[3] Wikipedia, DNSsec protocol http://en.wikipedia.org/wiki/Dnssec

[4] Wikipedia, DNS protocol http://en.wikipedia.org/wiki/Domain_Name_System

[5] IETF, beschrijving van het DANE protocol http://tools.ietf.org/html/draft-ietf-dane-protocol-21

**Controle van certificaten**

Uitgegeven certificaten kunnen worden ingetrokken, waarna ze op een CRL komen te staan. De Validation Authority (VA) controleert of certificaten nog geldig zijn of moeten worden ingetrokken. Als een certificaat moet worden ingetrokken, wordt dit door de CA gedaan.

Het betreffende certificaat wordt dan ingetrokken en het certificaat komt dan met het serienummer van het certificaat op de CRL of Delta CRL te staan. Vervolgens worden de CRL en/of Delta CRL aangeboden ter download aan clients. De client controleert de CRL of Delta CRL zodra er wordt gecommuniceerd en het blijkt dat er bij de communicatie tussen de client en de andere partij (bijvoorbeeld server) een certificaat vereist is. Een ingetrokken certificaat wordt geblokkeerd en wordt verder niet geladen door de client. Certificaten die niet op de CRL of Delta CRL staan, worden gewoon toegelaten en vertrouwd.

In een PKI in Windows Server 2008 kan een certificaat ingetrokken worden, maar kan de intrekking weer 'ongedaan' worden gemaakt. Het ingetrokken certificaat wordt dan weer bij uitgegeven certificaten teruggeplaatst en verdwijnt dan van de CRL of Delta CRL.

Als de CA over een OCSP responder beschikt en de client over een besturingssysteem beschikt waarin OCSP wordt ondersteund (Windows Vista, Windows 7, Windows Server 2008), dan kan via het OCSP protocol direct een check worden gedaan of het certificaat nog geldig is of niet. Er wordt dan direct in de database van de ingetrokken certificaten gecontroleerd of het betreffende certificaat wat benodigd is voor de communicatie tussen client en de andere partij daar in staat, zo niet, dan is het certificaat vertrouwd en wordt de communicatie verder afgehandeld.

**X.509 standaard voor certificaten**

De X.509 werd voor het eerst uitgegeven op 3 juli 1988 en deze standaard had toen een relatie met de X.500 standaard. De X.500 standaard gaat uit van een strikt hiërarchisch systeem van CA's die de certificaten uitgeven. Dit is in deze tijd niet meer denkbaar, omdat er nu zoveel CA's zijn dat er geen hiërarchisch systeem van CA's meer is. Het hiërarchisch systeem is overgegaan naar het webmodel (mesh architectuur)zoals we dat vandaag de dag kennen.

Op dit moment kennen we versie 3 van de X.509 standaard. Er zijn 2 eerdere versies geweest; de eerste versie dateert van 1993 en is te vinden in RFC 1422[6]. De tweede versie van de standaard heeft het uiteindelijk niet gehaald om dé standaard voor het internet te worden voor CA's, omdat het mogelijk was om dezelfde naam van een opgeheven CA na een tijdje te gebruiken. Met versie 2 was het dus bijvoorbeeld mogelijk om na het faillissement van Diginotar een nieuwe CA op te starten met dezelfde naam en dan weer certificaten uit te geven. Het IETF raadt het af om uitgever en subject namen te hergebruiken. Versie 3 is de huidige standaard en staat beschreven in RFC 5280[7].

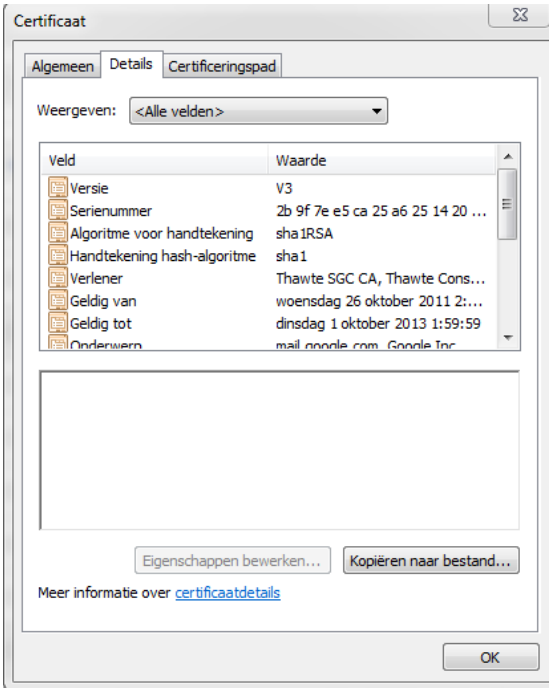
Met de X.509 standaard kan een CA een certificaat uitgeven welke is verbonden met een distinguished name zoals bij de X.500 standaard of het certificaat kan worden verbonden aan een

[6] Internet Engineering Taskforce, RFC1422 http://www.ietf.org/rfc/rfc1422

[7] Internet Engineering Taskforce, RFC 5280 http://tools.ietf.org/html/rfc5280

alternatieve naam zoals een e-mail adres of een DNS entry. Met name de binding van de publieke sleutel met een DNS entry is wat wordt gedaan door de CA's op het internet. Zo wordt de echtheid van een website aangetoond met het certificaat.

Een certificaat die wordt uitgegeven met de X.509 standaard heeft een bepaalde structuur. Deze structuur bestaat uit een aantal velden in het certificaat zoals de versie, het serienummer van het certificaat, de naam van degene die het certificaat uitgeeft, de geldigheidsperiode, het certificaat ondertekening algoritme en de digitale certificaat ondertekening. Hieronder is een voorbeeld van de X.509 structuur te zien van een Google certificaat:




In versie 3 zijn ook de extensions geïntroduceerd. De extensions geven aan hoe het certificaat gebruikt moet worden. Een extension kan critical of non critical zijn. Dit wil zeggen dat een bepaalde extension vereist is (critical) en een andere extension niet vereist (non critical). Voor het verwerken van de certificaten door het systeem is dit beter i.v.m. beveiliging; als een bepaald type certificaat een critical extension heeft wat niet door een systeem verwerkt kan worden, dan wordt het certificaat verworpen. De extensions voor versie 3 zijn vastgelegd in RFC 5280 in subparagraaf 4.2.1 (zie voetnoot 12).

Binnen de X.509 standaard worden de volgende PKI standaarden gebruikt:

- PKCS7 (Cryptografie Message Syntax Standard voor het ondertekenen en/of versleutelen van berichten binnen een PKI)
- Secure Sockets Layer (SSL) (Protocol voor beveiligde comunnicatie over het internet)
- Online Certificate Status Protocol (OCSP)/Certificate Revocation List (CRL) (voor controle van certificaten of deze al dan niet zijn ingetrokken)

- PKCS12 (Wordt gebruikt voor het opslaan van een privé sleutel met de bijbehorende public key certificaten)

De PKCS7 en 12 standaard hebben een relatie met elkaar, omdat de ene standaard de wijze van ondertekening/versleuteling regelt en de andere standaard de opslag van de privé sleutel en het certificaat voorschrijft.

Er zijn ook andere protocollen en standaarden die X.509 certificaten ondersteunen. De protocollen worden gebruikt om bijvoorbeeld beveiligde verbindingen op te zetten zoals met SSH en HTTPS kan. Andere protocollen hebben weer andere toepassingen m.b.t. ondersteuning van het X.509 certificaat. Deze protocollen en standaarden zijn bijvoorbeeld:

- TLS/SSL (Transport Layer Security[8])
- S/MIME (Secure Multipurpose Internet Mail Extensions ofwel secure e-mail[9])
- IPsec (Beveiliging toegevoegd aan DNS[10])
- SSH (Secure Shell, wordt veel gebruikt in beheer omgevingen, bijvoorbeeld beveiligd inloggen op netwerkapparatuur[11])
- Smart Card (wordt gebruikt voor bijvoorbeeld voor toegangsbeveiliging of zoals bij gemeente Sluis om in te loggen in het systeem)
- HTTPS (voor veilige webpagina's)
- EAP (Extensible Authentication Protocol[12])
- LDAP (Lightweight Directory Access Protocol, wordt ook voor Active Directory gebruikt[13])
- XMPP

Hoewel de standaard beter is geworden met versie 3 en het systeem van een PKI veiliger heeft gemaakt, zijn er toch ook minpunten aan de standaard.

Zo is het ontwerp opgezet dat er aan blacklisting wordt gedaan in plaats van whitelisting. Het gebruik van CRL's en OCSP is namelijk blacklisting, er wordt gekeken of het certificaat nog wel geldig is,

[8] IETF, TLS protocol http://tools.ietf.org/html/rfc6101

[9] IETF, S/MIME protocol http://tools.ietf.org/html/rfc5751

[10] Wikipedia, overzicht met RFC lijsten m.b.t. IPsec http://en.wikipedia.org/wiki/IPsec#Standards_Track_RFCs

[11] IETF, SSH protocol http://tools.ietf.org/html/rfc4253

[12] IETF, EAP protocol http://tools.ietf.org/html/rfc3748

[13] Wikipedia, lijst met RFC m.b.t. LDAP
http://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol#RFCs

terwijl het misschien makkelijker is om een whitelist van te vertrouwen certificaten aan te maken. Vaak zijn de CRL's behoorlijk groot en moeten deze via het netwerk of het internet worden verspreid naar de clients. Een ander punt is dat het niet te achterhalen is of het root certificaat ooit is ingetrokken. Het kan belangrijk zijn om zulke informatie te hebben, als Diginotar nog had bestaan, had bijvoorbeeld nagezocht kunnen worden of het root certificaat ooit is ingetrokken. Met het bestaande webmodel zijn de relaties tussen de CA's niet altijd even duidelijk en complex om in kaart te brengen. Bij een ramp zoals Diginotargate is het een hels karwei om de complexe relaties uit te zoeken en te bepalen welke certificaten niet meer te vertrouwen zijn.

De commerciële CA's die op het internet aanwezig zijn, zijn niet allemaal sterke CA's. Er zijn zoveel CA's die op het internet actief zijn, dat de relaties heel complex zijn en het is niet bekend welke CA's zwak zijn. Met zwak of sterk wordt bedoeld dat een CA qua beveiliging bestand is tegen bijvoorbeeld hacken. Omdat de CA's met elkaar verbonden zijn is de keten zo sterk als de zwakste schakel; zit er een CA zoals Diginotar bij, dan kan de hele keten daardoor beïnvloed worden.

Bij implementatie van PKI's worden, al dan bewust of niet, ook fouten gemaakt. De meeste fouten hebben betrekking op het ontwerp, bugs of andere interpretaties van de standaarden. Een voorbeeld is het uitzetten van het checken of een certificaat is ingetrokken omdat het als een obstakel wordt gezien. Als het checken van ingetrokken certificaten standaard aan staat in de browsers, dan kan de gehele infrastructuur daarop crashen.

Qua beveiliging zijn er ook een aantal minpunten aan de X.509 v3 standaard. Zo zijn er bepaalde exploits gevonden. In 2008 is aangetoond dat er valse certificaten konden worden uitgegeven omdat een bepaalde CA nog gebruik maakte van de MD5 versleutelings methode.

Er wordt actief aan de standaard gewerkt door het IETF om de standaard te verbeteren. Zij hebben een speciale werkgroep genaamd Public Key Infrastructure (X.509) PKIX. Wat de werkgroep doet en de bijbehorende documenten zoals ontwerpen en RFC's is te vinden op hun website[14].

*Bestandsformaten van certificaten*

De certificaten kunnen opgeslagen worden in bepaalde bestandsformaten. Tussen de bestandsformaten kunnen verschillen zijn, omdat het ene formaat het certificaat en de sleutel in 1 bestand opslaat terwijl het andere formaat het certificaat en de sleutel in 2 bestanden opslaat. Dit heeft te maken met het platform waar het certificaat op gebruikt wordt. De platformen verschillen allemaal met elkaar qua bestandsformaten en het bestandsformaat van certificaten is daar 1 voorbeeld van.

Zo zijn er bestandsformaten voor Windows platformen en Linux (gebaseerde) platform. Een voorbeeld van zo'n Linux platform is ESX van Vmware. Hieronder worden de volgende bestandsformaten beschreven.

[14] IETF PKIX werkgroep http://datatracker.ietf.org/wg/pkix/

*PEM bestandsformaat*

PEM staat voor Privacy Enhanced Mail en het is 1 van de meest voorkomende formaten die CA's gebruiken om certificaten uit te geven. Het zijn Base64[15] gecodeerde ASCII bestanden die beginnen met ----Begin certificate------- en eindigen met -------end certificate-------- statements. Base64 is een encoderings schema wat veel gebruikt wordt om data te versleutelen en te transporteren over media (netwerk, het internet). Met het PEM formaat kunnen server certificaten, tussenliggende certificaten en privé sleutels worden gemaakt. Het certificaat en de sleutel kan bij PEM in 1 bestand worden opgeslagen, maar sommige platforms zoals Apache hebben de vereiste dat het certificaat en de privé sleutel in aparte bestanden moeten worden opgeslagen.

*CER, CRT en DER bestandsformaat*

Deze certificaten worden in het DER formaat (Distinguished Encoding Rules[16]) opgeslagen, maar ze kunnen ook als Base64 worden opgeslagen. Het verschil tussen DER en Base64 is dat DER een binaire vorm van een certificaat is en Base64 is in de vorm van ASCII. De enige manier om te zien of het een DER of Base64 bestand is, is door het te openen in een tekst editor zoals Notepad en te kijken naar de Begin/End statements in het bestand. DER bestanden worden veelal op het Java platform gebruikt.

*P7B en P7C bestandsformaat*

Certificaten die in p7b of p7c formaat worden opgeslagen zijn Base64 ASCII bestanden. Deze bestanden beginnen met de ---------Begin PKCS7---------- en ---------END PKCS7---------- statements. Dit formaat heeft een relatie met de PKCS#7 standaard[17]. Een P7b en P7c bestand bevatten alleen certificaten en certificaten ketting. De privé sleutel wordt nooit in deze bestanden opgeslagen. P7b en P7c bestandsformaten worden vooral door Windows en Java Tomcat gebruikt.

*P12/PFX bestandsformaat*

Deze bestandsformaten hebben een binair formaat en slaan de server certificaten, tussenliggende certificaten en de privé sleutel op in 1 encrypted bestand. De formaten zijn .pfx en .p12. .pfx bestanden worden veelal op Windows machines gebruikt om de certificaten en privé sleutels te importeren en exporteren. Het p12/PFX formaat heeft een relatie met de PKCS#12 standaard[18], waarin beschreven wordt hoe privé sleutels en certificaten worden opgeslagen in dit formaat.

*Certificaten en interoperabiliteit*

Certificaten kunnen dus in bepaalde bestandsformaten worden opgeslagen. Dit heeft te maken met het platform waar de certificaten op worden gebruikt.

---

[15] Wikipedia, Base64 http://en.wikipedia.org/wiki/Base64

[16] Wikipedia, DER http://en.wikipedia.org/wiki/Distinguished_Encoding_Rules

[17] Wikipedia, PKCS#7 http://en.wikipedia.org/wiki/PKCS

[18] Wikipedia, PKCS#12 http://en.wikipedia.org/wiki/PKCS

Voor bijvoorbeeld Apache, Java en Tomcat kunnen hele andere bestandsformaten worden gebruikt dan voor de certificaten in Windows.

Bij een PKI die op een Windows systeem draait, is het probleem dat gemaakte certificaten die bestemd zijn om op een ander platform te worden geplaatst een bestandsformaat moeten hebben voor het betreffende platform. Die certificaten zullen dus omgezet moeten worden naar een bestandsformaat wat geschikt is voor het platform waar de certificaten op komen te staan.

Het omzetten kan worden gedaan met een open source programma genaamd OpenSSL. Ook zijn er verschillende converters op het internet te vinden die de certificaten kunnen omzetten naar het gewenste bestandsformaat. Een voorbeeld van zo'n converter is die van SSLshopper[19].

Het belangrijkste om rekening mee te houden bij het omzetten van certificaten is dat van tevoren bekend moet zijn voor welk platform het bedoeld is. Elk platform heeft een bepaald bestandsformaat voor certificaten zodat het als een certificaat herkend kan worden. Ook is het bij elk platform anders hoe wordt omgegaan met de opslag van het certificaat en de privé sleutel. Zo kan het certificaat en de privé sleutel in 1 bestand worden opgeslagen, maar kan het ook zo zijn dat het certificaat en de privé sleutel beide in aparte bestanden moeten worden opgeslagen. Dit is bijvoorbeeld bij ESX en Vcenter Server van Vmware het geval.

Het omzetten van certificaten met OpenSSL kan lastig zijn vanwege de commando's die ingevoerd moeten worden. Gelukkig zijn er goede instructies om dit een stuk makkelijker te maken. Zo is er de instructie van Adrian Costea[20] waar het vervangen van een standaard ESX certificaat goed wordt beschreven en waar de commando's die in OpenSSL gebruikt moeten worden goed staan beschreven.

Op de website SSL certificaten zijn ook goede handleidingen te vinden om certificaten te installeren aan te maken en om te zetten met OpenSSL. Er kan op de website van SSL certificaten zelfs per platform of applicatie worden bekeken hoe een certificaat moet worden geïnstalleerd, gemaakt of omgezet[21].

## *4.3 Componenten van een PKI*

Een PKI bestaat uit een aantal belangrijke componenten. Elke component heeft zijn eigen taken. Deze componenten zijn:

- Certification Authority (CA)
- Registration Authority (RA)
- Validation Authority (VA)

[19] SSLshopper, https://www.sslshopper.com/ssl-converter.html

[20] Blog Adrian Costea, http://www.vkernel.ro/blog/replacing-vmware-esx-server-default-self-signed-certificate

[21] SSLcertificaten.nl, Handleidingen per platform of applicatie https://www.sslcertificaten.nl/knowledgebase/SSL_Certificaten/

- Repository/Directory
- Eindentiteiten

**Certification Authority**

De Certification Authority heeft als voornaamste taak om een certificaat aan te maken, uit te geven en te distribueren. Ook heeft de Certification Authority de taak om certificaten te beheren. Hiermee wordt dan het bewaren van de certificaten bedoeld, maar ook het bijhouden van een openbare lijst van certificaten die ingetrokken zijn (CRL). De taken van een Certification Authority op een rijtje:

- Genereren van certificaten
- Uitgeven en distribueren van certificaten
- Intrekken van certificaten
- Onderhouden van de directory (de openbare lijst met certificaten)
- Publiceren van certificaten en Certificate Revocation Lists (CRL's)
- Eventuele cross certificatie
- Sleutel back-up (optioneel)

**Registration Authority**

De Registration Authority is de component waar een certificaat aangevraagd kan worden. De Registration Authority heeft 2 taken:

- Registratie van gebruikers
- Het vaststellen van identiteit van de gebruikers

**Validation Authority**

De Validation Authority heeft als voornaamste taak te controleren of het betreffende certificaat niet is ingetrokken of opgeschort.

**Repository/Directory**

De Repository/Directory is de opslagplaats voor certificaten en de CRL's.

**Eindentiteiten**

De eindentiteiten zorgen voor het aanvragen van certificaten en het aanmaken van privé en publieke sleutels.

## 4.4 PKI Architecturen

De PKI architectuur zorgt er voor dat de relatie tussen openbare sleutels, certificaten en autoriteiten te vertrouwen is. Daarom is het belangrijk dat bij de opzet van een PKI goed wordt nagedacht over de architectuur en vragen als 'worden de certificaten alleen intern uitgegeven of gaan we later ook extern, buiten de eigen organisatie, ook certificaten uitgeven?'. Ook moet de organisatie zich de vraag stellen of zij een interne PKI willen opzetten of de gehele PKI uit willen besteden aan een derde partij.

**De PKI gedeeltelijk of geheel uit laten besteden aan een derde partij**

Als een organisatie een eigen PKI heeft, kunnen ze ervoor kiezen om een publieke CA te kiezen om te zorgen dat de interne PKI nog betrouwbaarder wordt. Dit wordt een Trusted root for inhouse PKI/CA [22]genoemd; ofwel het product wat voor PKI's/CA's beschikbaar is die intern bij bedrijven zijn opgezet.

Redenen om dit te doen zijn bijvoorbeeld beter imago voor het bedrijf, het beheer van een PKI uit handen geven zodat intern in de organisatie daar geen aandacht meer aan besteed hoeft te worden, compliancy¸en ook een lagere Total Cost of Owernship (TCO).

Het is ook mogelijk om de PKI geheel uit te besteden aan een publieke CA. De CA zorgt dan binnen hun eigen netwerk voor een private PKI. Dit scheelt de organisatie kosten voor ontwerp, bouw en implementatie, maar ook kosten voor hardware, licenties en personeel die de PKI moeten beheren/managen.

Het gedeeltelijk of geheel uitbesteden van de PKI aan een publieke CA is duur. Vaak is er al een setup fee die enkele tien duizenden dollars is, maar daarnaast moeten ook licentiekosten per gebruiker worden betaald, wordt er een service fee in rekening gebracht en wordt er een fee berekend voor de inzet van IT personeel bij de publieke CA. Deze optie is dan ook aangeraden voor grotere organisaties. Om een goed inzicht te krijgen in de totale kosten per jaar, is er op de website van Arx[23] een duidelijke tabel te vinden met kosten voor een interne en uitbesteedde PKI. De tabel laat zien dat voor een geheel uitbesteedde PKI de kosten op ongeveer 300.000 dollar liggen voor het eerste jaar.

**Een interne PKI opzetten**

Voor organisaties die zelf een PKI willen opzetten, kan er gekozen worden uit 3 architecturen. De single CA architectuur, hiërarchische architectuur en de mesh architectuur.

*Single PKI architectuur*

Dit is de meest basic structuur die er is en bestaat uit 1 CA. Alle gebruikers vertrouwen op deze CA voor de communicatie en aanvragen die zij doen. Omdat er 1 CA is, begint het certificatie pad met

[22] Globalsign, Trusted root for inhouse PKI http://www.globalsign.com/certificate-authority-root-signing/internal-pki/

[23] Arx.com, white paper digital signatures and the hidden costs of PKI http://www.arx.com/resources/white-papers/digital-signatures-and-the-hidden-costs-of-pki.htm

het certificaat van de root CA. Bij een single PKI architectuur bestaat het certificatie pad uit 1 niveau waar de certificaten onder worden uitgegeven. Het certificatie pad kan worden achterhaald door de eigenschappen van het certificaat te bekijken.

*Hiërarchische architectuur*

In de praktijk is dit de meest gebruikte PKI architectuur. De CA's zijn in een structuur opgebouwd en bovenaan is er 1 CA, namelijk de Root CA. Alle certificaten worden door de root CA ondertekend en komen via de root CA en tussenliggende CA's die ook certificaten uitgeven (in een hierarchische structuur), bij de eindgebruikers terecht.

*Mesh architectuur*

De mesh structuur wordt ook wel het web model genoemd. Bij deze structuur is er geen root CA. De CA's in de structuur zijn allen gelijk en hebben een gelijkwaardige relatie met elkaar. Het voornaamste voordeel van de mesh structuur is dat deze kan groeien zonder tussenkomst van een root CA. Hiermee is gelijk ook een nadeel te benoemen, namelijk dat de groei zo hard kan gaan dat er geen duidelijk overzicht meer is van de CA's. Een voorbeeld van een mesh architectuur is de PKI architectuur op het internet. Er zijn vele CA's die met elkaar samenwerken en op het moment van schrijven zijn er +- 650 CA's op het internet aangesloten die digitale certificaten uitgeven.

*Crosscertificering*

CA's kunnen er voor kiezen om samen te werken. De samenwerking wordt dan aangegaan op basis van een vertrouwensrelatie. Door crosscertificering kan een gesloten omgeving gebruik maken van een andere CA om certificaten uit te geven in een open omgeving. Het nadeel is als de root CA van een externe CA gecompromitteerd is en niet meer te vertrouwen. Dan kan het zijn dat certificaten die door de externe CA worden uitgegeven aan de gesloten omgeving niet veilig en betrouwbaar zijn.

Het vertrouwen in de PKI is gestoeld op de architectuur van de PKI. Er kan voor een hiërarchisch model worden gekozen waarbij er maar 1 root CA is, of er kan voor een mesh model worden gekozen waarbij de CA's op basis van vertrouwen met elkaar samenwerken en geen root CA aanwezig is.

Van beide vormen zijn nadelen te benoemen. Bij een PKI met slechts 1 root CA moet worden opgelet dat de root CA en/of de privé sleutel nooit in verkeerde handen terecht komt.

Bij een infrastructuur waarbij meerdere CA's met elkaar samenwerken, moeten alle CA's zich aan de regels houden en is er 1 CA bij die het qua beveiliging minder voor elkaar heeft, dan is de hele keten in gevaar. Een geval van ernstige inbreuk op een PKI is vorig jaar aan de hand geweest met Diginotar. Bij Diginotar was er een inbraak geweest in het computersysteem, wat uiteindelijk heeft geleid tot het valselijk kunnen uitgeven van certificaten door de root CA van Diginotar.

Na onderzoek bleek dat er veel mis was bij Diginotar. Achteraf bleek dat er op gebied van het ontwerp van de PKI,  technisch en organisatorisch dingen niet klopten en problemen waren, wat heeft kunnen leiden tot de misschien wel meest besproken hack van 2011.

De gevolgen waren desastreus; duizenden certificaten waren vals uitgegeven en moesten worden ingetrokken, het vertrouwen in de PKI was dramatisch laag op dat moment, Iraanse burgers waren

de dupe van afluisterpraktijken omdat een valselijk uitgegeven Google certificaat was uitgegeven op Iraanse servers en uiteindelijk is Diginotar failliet gegaan.

## *4.5 PKCS standaarden*

Binnen een PKI worden standaarden gebruikt. Deze standaarden zorgen er ook voor dat CA's die ieder hun eigen PKI hebben, met elkaar kunnen samenwerken. De betreffende CA's mogen onderling afspraken maken m.b.t. de gebruikte cryptografische technieken, maar de compatibiliteit van de certificaten moet gewaarborgd worden door het toepassen van standaarden.

Om deze reden zijn de Public Key Cryptographic Standards (PKCS) ontwikkeld. De volgende PKCS standaarden zijn erg belangrijk:

- PKCS 1 RSA cryptography standard[24]
- PKCS 3 Diffie-Hellman Key Agreement standard
- PKCS 5 Password based cryptography standard[25]
- PKCS 6 Extended certificate syntax standard
- PKCS 7 Cryptographic message syntax standard[26]
- PKCS 8 Private key information syntax standard[27]
- PKCS 9 Selected object classes and attribute types[28]
- PKCS 10 Certification request syntax standard[29]
- PKCS 11 Cryptographic token interface standard
- PKCS 12 Personal information exchange syntax
- PKCS 13 Elliptic curve cryptography standard
- PKCS 15 Cryptographic token information format standard

[24] IETF, PKCS1 standaard http://tools.ietf.org/html/rfc3447

[25] IETF, PKCS5 standaard http://tools.ietf.org/html/rfc2898

[26] IETF, PKCS7 standaard http://tools.ietf.org/html/rfc2315

[27] IETF, PKCS8 standaard http://tools.ietf.org/html/rfc5208

[28] IETF, PKCS9 standaard http://tools.ietf.org/html/rfc2985

[29] IETF, PKCS10 standaard http://tools.ietf.org/html/rfc2986

In de PKCS standaarden komen de PKCS7 en PKCS12 standaard terug die binnen een PKI worden gebruikt.De standaarden hebben betrekking op cryptografie en hoe bepaalde dingen geregeld zijn qua programmering, beveiliging en opslag van de privé sleutel en certificaat.

## *4.6 Toepassingen van een PKI*

Nu duidelijk is wat een PKI is en waar het voor dient, kunnen er een aantal voorbeelden van toepassingen van een PKI worden gegeven.

In het dagelijks leven heeft een ieder wel met 1 toepassing te maken. Een heel goed voorbeeld is e-commerce. Met het HTTPS protocol wordt een webpagina veilig aangeboden en kan de consument veilig online shoppen.

Een andere toepassing is secure mail. Heel veel bedrijven maar ook ISP's bieden webmail aan. Dit is vaak beveiligd om ervoor te zorgen dat de communicatie over een veilige verbinding loopt. Voor secure e-mail kan gebruik worden gemaakt van bijvoorbeeld S/MIME, SSL en TLS.

Een VPN verbinding komt in deze tijd heel veel voor. Er komen steeds meer flexwerkers bij die ook thuis kunnen werken. Vaak wordt er dan een VPN verbinding met het bedrijfsnetwerk gemaakt waardoor de gebruiker vanuit huis over een veilige verbinding kan inloggen op het bedrijfsnetwerk. Voor VPN verbindingen wordt dan een VPN certificaat gebruikt.

Met een smartcard is het mogelijk om authenticatie te regelen. Zo kan een smartcard bijvoorbeeld worden gebruikt voor toegang tot een gebouw, maar kan het ook worden gebruikt voor toegang tot een systeem. Bij de gemeente Sluis gebruikt men de smartcard om in te loggen in het systeem. De smartcard wordt dan in het toetsenbord gestoken, waarna inloggen in het systeem mogelijk is.

Binnen een interne PKI kunnen aan gebruikers certificaten worden gegeven, zodat men erop kan vertrouwen dat de juiste gebruiker inlogt op het netwerk en gebruik maakt van de netwerkservices. Zodra een gebruiker inlogt krijgt deze een certificaat toegewezen en die wordt dan aan de gebruiker gekoppeld, totdat het certificaat verloopt. Daarna moet een nieuw certificaat worden uitgegeven voor de gebruiker.

Ook servers kunnen binnen een interne PKI een certificaat krijgen. In de afstudeeropdracht heb ik aan 2 servers (ESX host en Vcenter server) certificaten uitgegeven zodat deze servers vertrouwd zijn binnen het netwerk van mijn lab opstelling.

# 5 PKI architecturen in de praktijk

## *5.1 Kiezen van een PKI architectuur*

In de praktijk zal er voor een organisatie die een PKI op wilt zetten eerst gekeken moeten worden naar de wensen en eisen alvorens een PKI te gaan bouwen en te implementeren. Op basis van de wensen en eisen kunnen er selectiecriteria worden gesteld waarna het makkelijk is om de juiste PKI architectuur te kiezen.

Eerst zal een organisatie een keuze moeten maken of de PKI intern wordt gebouwd en helemaal zelf wordt beheerd, het beheer van de interne PKI wordt uitbesteed aan een derde partij of dat de keus wordt gemaakt om een PKI helemaal uit te besteden.

In het artikel van Computer weekly[30] wordt beschreven waar men bij de opzet van een PKI rekening moet houden. Bij de bouw van een PKI komen o.a. licentiekosten en hardware kosten om de hoek kijken, in het artikel staan een aantal vragen die een organisatie zich kan stellen. Deze vragen hebben betrekking op kosten, maar ook op het al dan niet hebben van gekwalificeerd personeel die een PKI kunnen beheren en de taken zoals beschreven bij de componenten van een PKI kunnen uitvoeren.

Echte best practices zoals die er voor bijvoorbeeld ITIL zijn, zijn voor het bouwen van een PKI niet beschikbaar. De keus om een PKI intern te bouwen en te beheren of dit geheel of gedeeltelijk uit te besteden ligt bij de organisatie die dit wilt gaan doen. De keus zal uiteindelijk afhangen van de te maken kosten, inzet van personeel om een PKI te managen en de beveiliging en imago van het bedrijf. Bedrijven die bijvoorbeeld bij CA Diginotar waren aangesloten, hebben achteraf mogelijk imago schade geleden. Als een bedrijf totale controle over de beveiliging en het imago van het bedrijf wilt, dan is uitbesteding geen goede optie.

Voor organisaties die wel zelf een PKI op willen zetten en beheren, zijn er een aantal selectiecriteria waar men rekening mee moet houden om de juiste PKI architectuur te kunnen kiezen die het beste past bij hun organisatie. Deze criteria worden hieronder beschreven.

**Selectiecriteria voor het kiezen van een PKI architectuur**

*Selectie op basis van budget*

Budget kan een selectiecriterium zijn voor het kiezen van een PKI architectuur. Een single PKI structuur vereist maar 1 machine, waar een hiërarchische of mesh structuur al meerdere machines vereisen. Daar komen dan ook nog de kosten voor software en licenties bij. Ook qua gebruik zal dit meer kosten m.b.t. stroom en dataverkeer.

[30] Computer Weekly, In house or out, how to start building a PKI http://www.computerweekly.com/feature/In-house-or-out-how-to-start-building-a-PKI

*Selectie op basis van de grootte van de organisatie*

De grootte van de organisatie kan invloed hebben op de keuzen van een PKI architectuur. Als een kleine organisatie een PKI wil bouwen en implementeren dan kan volstaan worden met een single PKI architectuur. Het opzetten, beheer en onderhoud van de PKI is gemakkelijk en kost weinig. Vaak is het in de praktijk dat 1 of 2 mensen dit bij hun dagelijkse werkzaamheden doen.

In het geval van grotere organisaties zal er waarschijnlijk voor een hiërarchische of mesh architectuur worden gekozen. Bij grotere organisaties kunnen de diverse taken die bij de CA componenten horen door meerdere medewerkers worden uitgevoerd.

*Selectie op basis van de organisatiestructuur*

Een kleine organisatie kan het beste kiezen voor een single PKI architectuur. Een single PKI architectuur is gemakkelijk uit te rollen en het beheer en onderhoud kan door een aantal personeelsleden worden uitgevoerd. De single PKI architectuur bedient ook niet zoveel gebruikers, vaak is het maar een kleine groep gebruikers.

Grotere organisaties moeten kiezen tussen een hiërarchische en mesh architectuur, omdat dit architecturen zijn die geschikt zijn voor grotere organisaties. Een hiërarchische architectuur past goed bij hiërarchische organisaties zoals bijvoorbeeld de overheid. Bij een hiërarchische organisatiestructuur zijn de trust relaties namelijk gemakkelijker in kaart te brengen. Het certificatie pad is niet zo complex als bij een mesh architectuur. Vooral als er binnen hiërarchische organisaties niet veel wordt gecommuniceerd tussen de afdelingen, is een hiërarchische architectuur een zeer goede keuze.

Als een grote, niet hiërarchische organisatie wel veel communiceert, dan is een mesh architectuur een goede keuze. Zo is er bijvoorbeeld een groot verschil tussen een overheidsorganisatie die weinig onderling communiceert en een dynamische organisatie zoals een bank met miljoenen transacties per dag. Bij een dynamische organisatie past een mesh architectuur het beste vanwege de hoeveelheid aan communicatie, waarbij het een goede zaak is dat er meerdere trust points in de architectuur zijn. Bij een mesh architectuur liggen de trust relaties wel complexer waardoor het certificatie pad complexer wordt, maar door crosscertificering is een mesh architectuur veel flexibeler dan een hiërarchische architectuur.

*Selectie op basis van beveiliging en vertrouwen*

Er kan ook voor een PKI architectuur worden gekozen met beveiliging en vertrouwen in het achterhoofd. Bij een hiërarchische architectuur is er namelijk maar 1 root CA en zijn er eventueel tussenliggende CA's. Dit betekent wanneer de root CA gecompromitteerd is, de hele PKI keten gecompromitteerd is. Het kost dan veel tijd en moeite om de PKI keten weer vertrouwd te maken en alle gebruikers binnen de PKI hebben er dan last van.

Bij een mesh architectuur zijn er meerdere trust points. Als er 1 CA gecompromitteerd is, dan heeft dit geen ernstige gevolgen voor de rest van de PKI keten. Zodra er een nieuwe publieke sleutel is voor de gecompromitteerde CA dan zal deze veilig via de andere CA's naar de eindgebruikers worden verstuurd. Hierbij zijn er dus minder gebruikers die er last van hebben als 1 CA gecompromitteerd is.

## 5.2 Praktische punten bij het opzetten van een PKI

Bij het opzetten van een PKI zijn er een aantal praktische punten qua configuratie en beveiliging waar rekening mee gehouden moet worden. Deze punten zijn hieronder beschreven.

**Configuratie van de PKI**

*Certificaat geldigheid*

Het is belangrijk om rekening te houden met de levensduur van een certificaat. De levensduur van een certificaat is nooit langer dan de levensduur van een CA die het uitgeeft. Dit betekent dat wanneer de geldigheidsdatum van de CA verstreken is, er geen certificaten meer worden uitgegeven. Het is belangrijk om in het ontwerp de juiste geldigheidsdata per CA en voor elk certificaat te bepalen, zodat er geen nare verrassingen komen met certificaten die na een aantal jaar verlopen.

*Sleutellengte van de privé sleutel voor het certificaat*

De sleutellengte van de privé sleutel is belangrijk om van te voren te bepalen. Wordt er nog een sleutel van 1024 bits gebruikt of wordt er een 2048 bits sleutel gebruikt? De laatste is een betere keuze omdat dit een veiligere sleutellengte is die moeilijker is te kraken. Het is natuurlijk wel zo dat de systemen en de applicaties de 2048 bits sleutel moeten kunnen verwerken. Daarom is het belangrijk dit in het ontwerp al vast te leggen.

Het is aanbevolen een 2048 bits sleutel te genereren omdat dit veiliger is dan een 1024 bits sleutel. Het is namelijk gebleken dat door de steeds krachtigere computers die op de markt verschijnen, het steeds gemakkelijker is om een 1024 bits sleutel te kraken. Deze aanbeveling komt van het National Institute of Standards and Technology (NIST)[31].

In de praktijk worden 4096 bit sleutels niet veel gebruikt, omdat de verwerking ervan te langzaam is. de 2048 bit sleutel wordt het meest gebruikt en wordt breed ondersteund. 4096 bit certificaten kunnen wel worden gebruikt, maar dit wordt dan vaak gebruikt om te voldoen aan de FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION FIPS 140-2 standaard om aan te tonen dat de organisatie een goed beveiligde PKI heeft, maar voor SSL is het niet geschikt vanwege de te langzame verwerking.[32] Daarnaast kan het ook performance problemen geven, omdat niet ieder systeem of applicatie is gemaakt om met 4096 bits certificaten om te kunnen gaan.

*Authority Information Access (AIA) locaties*

De Authority Information Access (AIA) locaties zijn erg belangrijk om certificaten in op te slaan. Vanuit de AIA locaties kunnen de clients of de applicaties de certificaten downloaden. De AIA locatie is de database waar de certificaten worden opgeslagen. In Windows is dit beter bekend als de certificate store. In RFC 3280[33] is meer informatie te lezen over AIA locaties.

[31] National Institute of Standards and Technology, http://www.nist.gov/index.html

[32] NIST, FIPS 140-2 standaard http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

[33] IETF, RFC 3280 http://www.ietf.org/rfc/rfc3280.txt

*Bepalen van de publication points voor CRL's en de te gebruiken protocollen*

Om CRL's en Delta CRL's te publiceren zijn CRL Distribution Points (CDP) nodig waar clients en applicaties de laatste CRL's en Delta CRL's vandaan kunnen halen om zo de geldigheid van een certificaat te kunnen verifiëren. Het is belangrijk om hierbij rekening te houden met welk type clients men te maken heeft. Afhankelijk van het type clients kan voor een bepaald protocol of juist beide protocollen worden gekozen.

Zo kan bijvoorbeeld worden gekozen voor het LDAP protocol om CRL's en Delta CRL's te publiceren. Op die manier werkt het publiceren van de CRL's en Delta CRL's via Active Directory die gebruik maakt van het LDAP protocol. Wanneer er clients zijn die geen Windows draaien, kan worden gekozen voor het http protocol. Dan worden alle CRL's en Delta CRL's via http gepubliceerd zodat de clients het via het http protocol kunnen downloaden.

Wanneer een Windows client een CRL of Delta CRL download, wordt dit geplaatst in de cache in Windows. Met simpele commando's in de DOS prompt kan de cache worden bekeken of geleegd worden, waarna de machine weer nieuwe CRL's en Delta CRL's download.

*CRL geldigheid en overlap periodes*

De geldigheid van de CRL en de overlap periode is belangrijk, omdat tussen de updates voor ingetrokken certificaten geen gat mag zijn. De geldigheid van een CRL is afhankelijk van de hoeveelheid certificaten die een CA uitgeeft en intrekt. Een CA die veel certificaten uitgeeft, zal een kortere geldigheidsduur hebben van een CRL, bijvoorbeeld een paar dagen. Een CA die heel weinig certificaten uitgeeft kan een CRL hebben met een geldigheidsduur van 6 maanden tot een jaar.

Wanneer er certificaten worden ingetrokken en dit valt net in een overlap periode, dan zijn de huidige en de nieuwe CRL tegelijk geldig om problemen te voorkomen. Zo is het namelijk zeker dat de huidige CRL pas echt verloopt op de aangegeven datum en op deze manier blijft de administratie qua ingetrokken certificaten kloppen.

*OCSP URL's*

Naast de CRL's en Delta CRL's kan ook gekozen worden om een OCSP responder in te zetten zodat clients real time de certificaten kunnen verifiëren op geldigheid. Wanneer een OCSP responder wordt ingezet is het belangrijk om de URL van de responder aan te geven zodat clients weten waar de responder zich bevindt. Op dit moment kunnen Windows Vista, Windows 7, Windows 2008 en Windows 2008 R2 informatie opvragen van OCSP responders.

**Beveiliging van de PKI**

*Beveiliging van de privé sleutel van de root CA*

Het is heel belangrijk om de privé sleutel van de root CA te beveiligen en dit geheim te houden. Als de privé sleutel in verkeerde handen valt, kunnen er certificaten worden ondertekend die niet uitgegeven mogen worden; ofwel, valse certificaten.

Er zijn verschillende opties qua beveiliging van de privé sleutel van de root CA. Van het kopiëren van de privé sleutel op een smartcard, het offline halen van een root CA, het achter slot en grendel zetten van een root CA tot een Hardware Security Module (HSM). De keuze van beveiliging zal voor een organisatie afhangen van kosten, maar ook hoe hun PKI is ingericht en het daarbij geldende beveiligingsniveau.

De beste, maar hele dure optie is een Hardware Security Module (HSM). De privé sleutel kan hierop worden opgeslagen en de HSM kan aan de root CA worden gekoppeld. Dit lijkt qua beveiliging niet logisch, maar een HSM is er speciaal voor gemaakt om de privé sleutel en andere cryptografische bestanden te bewaren. HSM's kosten al gauw een paar duizend dollar, zo is er bijvoorbeeld de HSM van Globalscape met[34] of zonder[35] onderhoud en deze zijn te vinden bij A03, een Australische verkoopsite.

De privé sleutel moet dus goed worden beveiligd, als de privé sleutel in verkeerde handen valt of kwijt raakt, kunnen anderen certificaten uitgeven die niet uitgegeven mogen worden. Dit compromitteert de hele PKI, waarna alle certificaten moeten worden ingetrokken, een nieuw sleutelpaar voor de root CA moet worden aangemaakt en weer nieuwe certificaten moeten worden uitgegeven.

*Rollenscheiding*

Als een organisatie FIPS 140-2 compliant wil zijn, dan kan er rollenscheiding worden ingevoerd. In Windows Server is dit geregeld met administrator groepen en verschillende administrator accounts. Zo kan worden voorkomen dat er per ongeluk iets gewijzigd of verwijderd wordt. Windows Server 2008 is FIPS 140-2 compliant.

*Beleid en beleidsdocumenten*

Beveiliging is ook geregeld op beleidsniveau en dit wordt vastgelegd in een Certificate Policy (CP) en een Certificate Practice Statement (CPS).

In de CP wordt vastgelegd welke methode er gebruikt wordt om de identiteit van een persoon/object vast te stellen voor er een certificaat wordt uitgegeven.

In de CPS wordt beschreven hoe de PKI werkt vanuit het oogpunt van beveiliging. Hier wordt beschreven hoe de PKI en de CA worden gemanaged om zo te laten zien welke beveiliging er wordt toegepast. In een CPS kan bijvoorbeeld bij een mesh architectuur ook beschreven worden welke samenwerkingsverbanden er zijn met andere CA's (crosscertificering).

[34] A03 HSM met onderhoud erbij http://www.ao3.com.au/product.asp?ManufPartNo=GLOB324

[35] AO3 HSM zonder onderhoud http://www.ao3.com.au/product.asp?ManufPartNo=GLOB320

# 6 PKI gezien vanuit de eindgebruiker

Om de PKI vanuit de eindgebruiker te kunnen zien moet er een onderscheid worden gemaakt tussen een PKI die in een organisatie is opgezet waar de certificaten van de interne root CA worden uitgegeven en de publieke PKI waar de commerciële PKI's hun certificaten uitgeven.

Een organisatie die een PKI opzet zal er naar streven om de eindgebruiker zo min mogelijk te belasten met de uitgifte van certificaten. Een organisatie kan certificaten op verschillende manieren gebruiken zoals bijvoorbeeld toegang tot het gebouw of het systeem met een smartcard, authenticatie voor het netwerk, secure e-mail of authenticatie voor bepaalde applicaties. Voor een organisatie is het belangrijk dat een PKI goed ingeregeld wordt en dat certificaten bij voorkeur automatisch aan de eindgebruikers worden uitgedeeld zodat de eindgebruiker zo min mogelijk met installatie van certificaten op hun systeem belast worden.

De CA's die op het internet actief zijn verstrekken voornamelijk certificaten aan websites. Op deze manier wordt vastgesteld dat de website ook echt van degene is waar het certificaat op geregistreerd staat. De certificaten die CA's afgeven worden in de browser geladen en kunnen worden getoetst op geldigheid door een CRL of door OCSP. Dit gaat allemaal min of meer automatisch, het enige waar de eindgebruiker op moet letten is dat er bij bepaalde sites https:// voorafgaand aan de URL in de adresbalk staat, zodat het certificaat te vertrouwen is.

Het kan in de praktijk ook zo zijn dat een eindgebruiker toch een certificaat moet aanvragen en nog moet installeren. Een goed voorbeeld hiervan is het Amice certificaat van het Landelijk Meldpunt Afvalstoffen[36]. Om de applicatie te kunnen gebruiken heeft de eindgebruiker een certificaat nodig om te kunnen inloggen. Zo wordt getoetst dat de gebruiker die het certificaat heeft ook degene is wie hij/zij zegt dat die is. Het downloaden en installeren van een certificaat is lastig voor de eindgebruiker. Afhankelijk van het gebruikte platform en browser moeten instellingen worden gewijzigd, aanvullende software (Java en ActiveX componenten) worden geïnstalleerd om de website of applicatie te kunnen gebruiken met een certificaat. Het installeren van een certificaat met de import wizard voor certificaten van Windows is vrij gemakkelijk, maar in de praktijk wijst het uit dat een simpele import van het certificaat in het Windows systeem soms niet werkt.

Concluderend kan gesteld worden dat de eindgebruiker die certificaten voor een bepaald doel op het internet gebruikt en deze zelf moet installeren op zijn/haar systeem, dit als lastige materie ondervindt. Wanneer certificaten niet geïnstalleerd kunnen worden, kan dit een beveiligingsrisico zijn, omdat niet gecontroleerd worden of de website inderdaad de website is die men denkt te moeten gebruiken. Bij het niet kunnen installeren van een certificaat kan het ook zo zijn dat een eindgebruiker zijn/haar werk niet kan uitvoeren zoals bij het LMA het geval is. Kan het LMA certificaat niet geïnstalleerd worden, dan kan de applicatie niet gebruikt worden.

[36] LMA, Amice certificaat http://www.lma.nl/faq/technisch/benaderen_website_Amice/index.htm

# 7 De afstudeeropdracht: Opzetten van een eigen PKI en het vervangen van certificaten

## 7.1 Beginsituatie

Fontys heeft een securitylab wat bedoeld is voor studenten zodat zij hun projecten daarop kunnen uitvoeren. Het securitylab heeft 3 ESX servers die voorzien zijn van het standaard Vmware certificaat wat bij de ESX servers horen. Daarnaast is er nog een Vcenter Server waarmee het beheer van de 3 ESX servers wordt geregeld.

Op deze servers komt bij het aanloggen op de server met de Vsphere client steeds een foutmelding dat het geïnstalleerde certificaat niet vertrouwd wordt. Dit wordt vooral als lastig ervaren, omdat de melding soms een aantal keer achter elkaar komt en steeds weg geklikt moet worden. Het was de opdracht om er voor te zorgen dat de meldingen niet meer voorkomen en dat de certificaten op de servers vertrouwd worden.

## 7.2 Aanpak

Als eerst is in de labopstelling een Windows domein opgezet bestaande uit 1 domain controller. Vervolgens is een Windows Server geïnstalleerd met daarop Certificate Services om de PKI op te zetten. Bij de Certificate Services is ook de IIS server geïnstalleerd, zodat het aanvragen van certificaten via de website van de root CA kan lopen.

Daarna is een Windows Server met Vcenter Server geïnstalleerd, zodat het mogelijk was om een virtueel datacenter op te zetten. Later is er 1 ESX host geïnstalleerd.

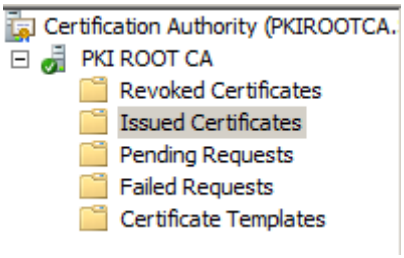
De PKI is toen ingericht en afgeconfigureerd voor automatische certificaat uitgifte, waarna de certificaten op de ESX server en Vcenter server zijn vervangen .

## 7.3 Werking van PKI

De PKI zoals die is opgezet in de labopstelling beschikt over alle functies die in een PKI thuis horen. Certificaten kunnen worden uitgegeven. Dit kan handmatig maar ook automatisch op basis van sjablonen.

Certificaten kunnen ook ingetrokken worden, waarna de certificaten op de CRL of Delta CRL komen te staan. Een certificaat kan ook van de CRL of Delta CRL gehaald worden en weer uitgegeven worden.  Het uitgeven en intrekken van certificaten werkt in Windows Server gemakkelijk.

Onderstaande afbeelding geeft aan dat er mappen zijn waar de certificaten in zijn opgeslagen.




Bij revoked certificates staan de ingetrokken certificaten en bij issued certificates staan de uitgegeven certificaten. Door in de map issued certificates op een certificaat te klikken met de rechter muisknop, kan in het menu wat verschijnt worden gekozen voor 'revoke certificate' waarna een reden opgegeven moet worden. Als dit is gedaan, dan is het certificaat direct ingetrokken en komt het op de CRL of Delta CRL te staan.

Met het intrekken van de certificaten kan verder niets mis gaan. De ingetrokken certificaten worden netjes op de CRL of Delta CRL gezet en omdat de CRL en/of Delta CRL toegankelijk zijn voor de clients, zijn alle clients in het netwerk heel snel op de hoogte van het ingetrokken certificaat. Zodra er wijzigingen zijn in de CRL of Delta CRL worden clients hiervan automatisch op de hoogte gesteld en wordt door de clients een CRL of Delta CRL gedownload.

In het Windows systeem kan aangegeven worden of gebruikers automatisch een certificaat kunnen krijgen van de root CA. Door dit zo in te stellen, hoeft de eindgebruiker niks meer te doen en dus geen certificaten zelf te installeren.

## *7.4 Vervangen van certificaten*

Vervolgens moesten de certificaten van de ESX server en Vcenter server worden vervangen door eigen ondertekende, vertrouwde certificaten.

*Vervangen Vcenter server certificaat*

Met OpenSSL is een privé sleutel aangemaakt en is een certificaat aangemaakt. Deze werden opgeslagen in 2 bestanden. Uit 1 bestand gegenereerd door OpenSSL stond een code waarmee via de website van de root CA een certificaat kon worden aangevraagd.

Via de root CA is het certificaat aangemaakt en deze is toen gedownload. Dit certificaat is toen met OpenSSL zo omgezet dat het geschikt is voor Vcenter server en daar als certificaat herkend wordt. Vervolgend zijn de bestanden gekopieerd naar de Vcenter server. Het kopiëren was een kwestie van kopiëren en plakken op de Windows server, omdat deze bestanden naar de map van Vcenter server moeten worden gekopieerd.

*Vervangen ESX server certificaat*

Het aanmaken en omzetten van het ESX server certificaat is op dezelfde wijze met OpenSSL en aanvragen aan de root CA. Het enige verschil met Vcenter server is dat de bestanden op de ESX server moeten worden geplaatst. Dit is geen kwestie van kopiëren en plakken. De bestanden moeten via een FTP verbinding op de juiste locatie op de ESX server worden gezet.

Daarna moeten de certificaten in de Trusted Root store van Windows Server worden gezet. Dit moet op de domain controller worden gedaan, zodat de certificaten daarna in het gehele domein vertrouwd zijn.

## 7.5 Knelpunten tijdens de opdracht

In eerste instantie wilde ik een PKI opzetten aan de hand van het boek van Brian Komar. Dit boek beschrijft goed hoe een PKI in Windows server 2008 kan worden opgezet. In de praktijk bleek dat het boek van Brian Komar een aantal fouten bevat en dat een aantal dingen niet goed zijn uitgelegd in het boek, waardoor ik al snel vast liep. Omdat een PKI met een eigen vooraf gedefinieerde configuratie geen harde eis was, heb ik besloten om de PKI in de simpelste vorm in Windows te installeren.

Het installeren van een PKI in Windows Server is geen probleem en dit is heel eenvoudig en het is verbazingwekkend hoe snel dit gaat. Het opzetten van een PKI is letterlijk in een mum van tijd gebeurd in Windows Server 2008.

Een groot knelpunt tijdens de opdracht was het instellen van automatische uitgifte van certificaten. Dit kan ingesteld worden voor bijvoorbeeld gebruikers en het heeft veel tijd gekost om uit te zoeken hoe dit precies werkt. Dit bleek uiteindelijk een domain policy te zijn die ingesteld moest worden, maar dit was in het boek van Brian Komar en op diverse blogs op het internet niet terug te vinden. In een handleiding van Windows Server 2003 ben ik uiteindelijk erachter gekomen dat de domain policy het laatst ontbrekende stukje was om de PKI zo te laten werken dat er automatisch een certificaat wordt uitgegeven aan bijvoorbeeld servers in hetzelfde domein of gebruikers die inloggen.

## 7.6 Resultaat

De certificaten van de ESX server en Vcenter server zijn vervangen door zelf ondertekende certificaten die vertrouwd zijn. Omdat de certificaten op de domain controller ook in de Trusted Root store staan, hoeft de eindgebruiker niets meer te doen.

De eindgebruiker logt in op een client op hetzelfde netwerk en kan vervolgens met de Vsphere client verbinding maken met de ESX of Vcenter server. Wanneer de verbinding wordt gelegd, krijgt de eindgebruiker geen melding meer, omdat het certificaat op domein niveau wordt vertrouwd. De eindgebruiker hoeft dus ook geen certificaat meer te installeren.

# 8 Conclusies en aanbevelingen

## *8.1 Conclusies*

Het idee achter een PKI is goed. De eerste versie ging uit van een strikt hiërarchisch model waar in deze tijd geen sprake meer van is. Dat ligt niet aan degenen die de standaard hebben ontwikkeld, maar meer aan de ontwikkelingen op het internet, de ontwikkelingen in het uitgeven van certificaten en de groei van het aantal CA's op het internet. Een PKI kan worden opgezet zodat de root CA binnen de eigen organisatie certificaten uitgeeft aan bijvoorbeeld gebruikers, machines of applicaties, maar er zijn ook commerciële CA's die met hun PKI certificaten uitgeven aan publieke websites.

Door een bepaald proces in te richten met de bijbehorende PKI componenten en taken die binnen deze componenten moeten worden uitgevoerd, wordt de identiteit van een persoon of een organisatie vastgesteld, waarna een certificaat wordt uitgegeven. Men weet dus van wie een bepaalde website is, dus is de website te vertrouwen. Binnen een PKI in een organisatie kunnen certificaten aan machines, gebruikers of applicaties worden uitgegeven, zodat men weet dat deze te vertrouwen zijn en er kan op vertrouwelijke wijze onderling worden gecommuniceerd.

Aan het opzetten van een PKI gaan een aantal zeer belangrijke stappen vooraf. Dit zijn het ontwerp, de bouw en de implementatie (de PKI implementeren in de productieomgeving). Het blijkt dat de meeste fouten en problemen in een PKI voortkomen uit een slecht ontwerp of gemaakte fouten tijdens de bouw en implementatie. Diginotar is hier een goed voorbeeld van omdat hun root CA op hetzelfde Windows domein als het interne netwerk zat.

Organisaties kunnen er voor kiezen om een eigen PKI op te zetten. Afhankelijk van hun eisen en wensen kunnen ze een PKI architectuur kiezen en dit kan bijvoorbeeld een single, hiërarchisch of webmodel zijn. De grotere bedrijven hebben locaties over de gehele wereld en zullen meer geneigd zijn om voor een webmodel te kiezen, vooral als er onder de afdelingen veel gecommuniceerd wordt. Er worden dan meer certificaten gebruikt en daarom is een webmodel een goede keuze voor grotere organisaties.

De CA's op het internet worden gezien als veilige, betrouwbare partners om certificaten te kopen mocht een organisatie behoefte hebben om hun eigen PKI veiliger en betrouwbaarder te maken door het aanschaffen van een certificaat. Helaas is gebleken dat dit niet bij elke CA op gaat en daar is Diginotar een voorbeeld van. De commerciële CA wordt dan boven de root CA van de organisatie gezet, waardoor vanuit de commerciële CA certificaten worden uitgegeven in de organisatie. Redenen om dit te doen zijn het management van de PKI wat uit handen genomen wordt, compliancy, overbodig maken van wachtwoorden en compatibiliteit met besturingssystemen en webbrowsers.

Organisaties die een eigen PKI hebben, geven zelf de certificaten uit. Het uitgeven van certificaten in een PKI is makkelijk, vooral als de PKI is opgezet in Windows. Het wordt lastiger als er certificaten gemaakt moeten worden door een Windows systeem die voor een ander platform bedoeld zijn. De certificaten moeten dan namelijk omgezet worden in een ander bestandsformaat waar het certificaat

uiteindelijk op komt te staan. Het omzetten kan worden uitgevoerd met open source programma OpenSSL of met een converter die op verschillende website te vinden zijn.

Qua eindgebruikers is er een groot verschil tussen een interne PKI in een organisatie of een commerciële CA die de certificaten uitgeeft. Binnen een organisatie zal vaak een PKI zo worden opgezet dat de eindgebruiker hier zo min mogelijk mee belast wordt. Hiermee wordt bedoeld dat certificaten vaak geautomatiseerd worden uitgegeven.

Bij certificaten die voor een bepaald doel van een CA op het internet moeten worden gedownload is dit een ander verhaal. Het certificaat moet dan nog geïnstalleerd worden zodat het vertrouwd wordt. Certificaten kunnen geïmporteerd of geëxporteerd worden. In Windows is dit heel gemakkelijk, maar het kan voorkomen dat het certificaat niet succesvol is geïnstalleerd op het systeem. Vaak weet de eindgebruiker dan ook niet het certificaat wel goed geïnstalleerd moet worden. Kortom, voor de eindgebruiker blijft het gebruiken van een certificaat nog lastige materie.

## *8.2 Aanbevelingen*

Het is aan te bevelen om in het begin goed na te denken over de PKI, waar de PKI voor bedoeld is en wat het doel van de PKI is. Op basis hiervan kan een PKI ontworpen, gebouwd en geïmplementeerd worden als men besluit om de PKI intern op te zetten.

Buiten het opzetten van een interne PKI zijn er meerdere keuzes, zo kan een publieke CA boven de eigen internet PKI worden aangesteld, zodat de interne PKI nog betrouwbaarder is. Ook kan er een keus gemaakt worden om het gehele beheer van de PKI bij een publieke CA onder te brengen of een private PKI bij een publieke CA afnemen.

Beveiliging hangt af van de configuratie van de PKI, de beveiliging voor de root CA zelf, netwerkbeveiliging en elementaire beveiliging op de systemen. Indien dit intern wordt opgezet en in het geval van uitbesteding hangt beveiliging af van de publieke CA. Zoals bij Diginotar is gebleken is het niet vanzelfsprekend dat een publieke CA een veilige PKI heeft. Ook moet voorkomen worden dat eindgebruikers problemen hebben met het gebruik of installatie van certificaten op hun systemen om beveiligingsrisico's te voorkomen.

Er zijn veel voordelen voor het geheel of gedeeltelijk uitbesteden van het beheer van een PKI of uitbesteding van de gehele PKI. Redenen hiervoor kunnen zijn: compliancy, vermindering van TCO en het uit handen nemen van het beheer van een PKI.

Een groot nadeel bij gedeeltelijke of gehele uitbesteding aan een publieke CA voor het beheer of de gehele PKI, is eventuele imagoschade als er een incident plaats vindt bij de publieke CA in kwestie. Daarom is het advies bij uitbesteding van een PKI goed na te gaan met welke publieke CA de organisatie in zee moet gaan en hoe de publieke CA hun PKI hebben ingericht en hoe zij omgaan met beveiliging.

Het gedeeltelijk uitbesteden van het beheer van een PKI of geheel uitbesteden van een PKI aan een publieke CA is behoorlijk prijzig. De organisatie in kwestie zal de opties moeten wegen tegen de kosten en de mogelijkheden in een organisatie om een PKI draaiend te kunnen houden en zo moeten kiezen tussen een interne PKI of een uitbesteedde PKI.

# Evaluatie

“We transform our dreams into the truth”. Slechts 1 zin van de opening van de America: the history of US series. Deze zin beschrijft mijn vastberadenheid, de wil om voor een betere toekomst te werken en studeren, het doel om mijn afstudeeropdracht tot een goed resultaat te brengen.

De afstudeeropdracht is voor mij een echte uitdaging geweest. Ik heb een hele sterke interesse in ICT beveiliging en deze afstudeeropdracht paste precies bij mij.

Ik had geen ervaring op het gebied van certificaten, PKI’s en ik had ook nog niet eerder een domein zelf ingericht. Dit alles heb ik tijdens mijn afstudeeropdracht kunnen doen en ik ben heel erg blij dat ik dit heb mogen doen.

Tijdens de opdracht heb ik heel veel ervaring opgedaan die in mijn beroepspraktijk nog steeds veel voordelen oplevert.

Er waren tijdens de opdracht momenten bij waar ik echt vastliep, maar door mijn vasthoudendheid en de wil om de afstudeeropdracht tot een goed einde te brengen heb ik alle hobbels kunnen overwinnen.

Het eindresultaat is dan ook precies wat ik voor ogen had; een totaaloplossing voor het certificaten probleem bij Fontys Hogeschool en ik heb meer expertise opgedaan m.b.t. een PKI en certificaten.

Voor u ligt mijn afstudeerscriptie en kan ik zeggen:  I transformed my dreams into the truth.

# Literatuurlijst


1. Brian Komar, *Windows Server 2008 PKI and security,* uitgever Microsoft Press, 2008

2. Said el Aoufi, *Cryptografie en ICT. Theorie en praktijk.*, tweede druk, uitgever SDU uitgevers

3. Webwereld website

   Webwereld is 1 van de grootste ICT nieuws websites van Nederland. Regelmatig komt Webwereld met nieuws wat niet alleen de ICT wereld raakt, maar alle lagen van de bevolking. Tijdens de Diginotar crisis was Webwereld dé website waar het meest recente nieuws over Diginotar was te vinden. De website wordt regelmatig door hackers benaderd om gewezen te worden op lekker in websites, netwerken of computersystemen. Zo was de maand oktober in 2011 uitgeroepen tot Lektober en kwam Webwereld elke dag in oktober met een lek naar buiten.

   URL: http://www.webwereld.nl

   URL Diginotar dossier: http://webwereld.nl/tags/diginotarcrisis.html

4. Microsoft Technet website

   De Microsoft Technet website is de website met kennis, handleidingen, blogs en trainingen voor Microsoft producten. Technet is er voor ICT professionals om hun kennis uit te breiden.

   URL: http://blogs.technet.com/b/askds/archive/2009/09/01/designing-and-implementing-a-pki-part-i-design-and-planning.aspx

# Bijlagen

I. Plan van aanpak

II. Onderzoeksverslag