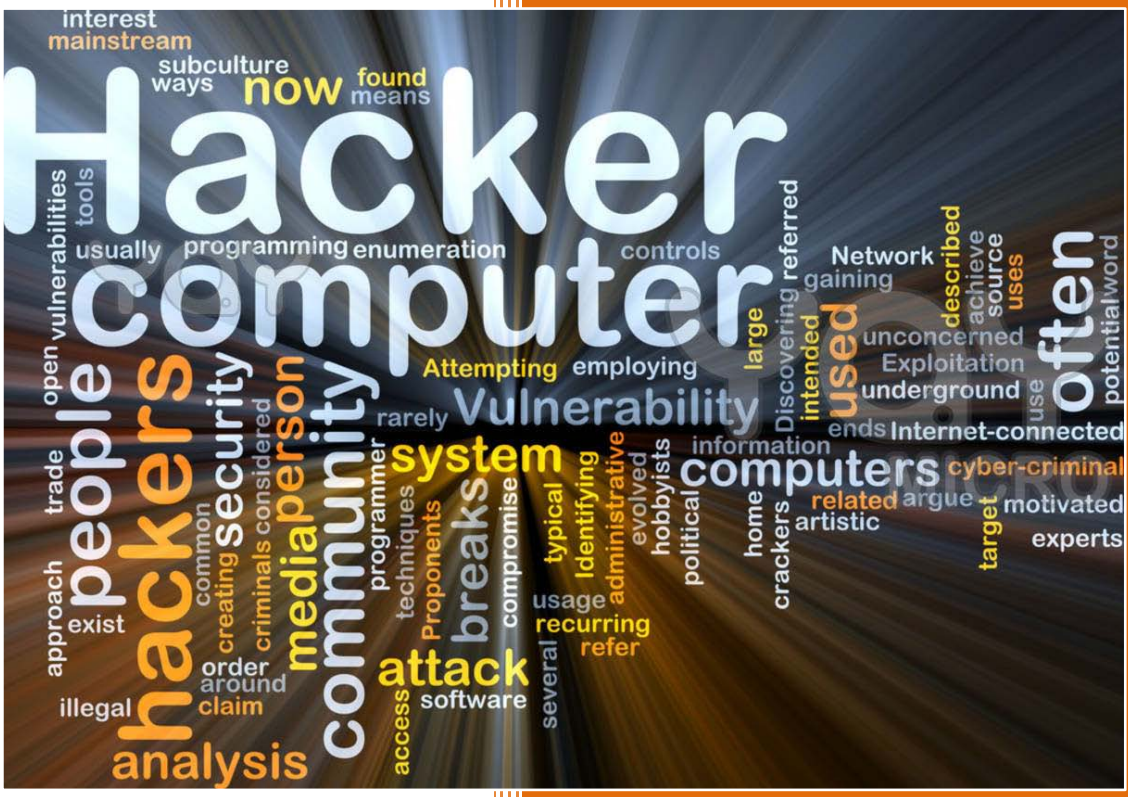
2011

Beveiliging VGO webomgeving



V • G • O
direct mail

Maikel Coolen
Afstudeerstage – VGO Direct Mail
Best, juni 2011

Gegevens:

Bedrijf :	**Naam:**	VGO Direct Mail
	Straat:	De Waal 2B
	Postcode:	5684 PH Best
	Tel:	+31 (0)499 - 33 11 88
	Fax:	+31 (0)499 - 33 11 89
	Email:	info@vgo.nl
	Website:	www.vgo.nl
Opdrachtgever / Stagebegeleider:	**Naam:**	**VGO Direct Mail** (Alex Slatman)
	Functie:	Coördinator Softwareontwikkeling
	Afdeling:	Softwareontwikkeling
	Tel:	+31 (0)499 331188
	Email:	info@vgo.nl
Opdrachtnemer:	**Naam:**	Maikel Coolen
	Functie:	Stagiair
	Afdeling:	Softwareontwikkeling
	Studierichting:	ICT Management & Security
	Student nr.:	200679
	Email:	maikel.coolen@student.fontys.nl
Procesbegeleider:	**Naam:**	Stefan Roijers
	Functie:	Docent
	Tel:	+31 (0) 877 877 348
	Email:	s.roijers@fontys.nl

Datum uitgifte: 9-06-2011

Getekend voor gezien door bedrijfbegeleider:

Datum:

Bedrijfsbegeleider: _____

Voorwoord

Deze scriptie is geschreven ter afsluiting van mijn HBO studie: ICT Management & Security aan de Fontys Hogeschool ICT te Eindhoven. Het afstudeerproject is uitgevoerd in opdracht van VGO Direct Mail, waar ik onderzoek heb gedaan naar de beveiliging van hun webomgevingen met name beperkt tot “ViaNota” en de “Mijn Omgeving”. In de periode van 21-02-2011 t/m 01-07-2011 ben ik werkzaam geweest bij VGO Direct Mail.

Tijdens deze periode had ik van tevoren al een beetje basiskennis opgedaan van web security door de opdrachten bij Ethical Hacking van het vak IMS, maar dit was weer een tijdje geleden. Deze kennis heb ik ververst en verder uitgebreid. Omdat de technieken snel veranderen moet deze kennis regelmatig vernieuwd worden. Daarnaast heb ik gekeken naar verschillende hulpmiddelen die mij ondersteunde tijdens het zoeken naar potentiële beveiligingslekken.

Uiteindelijk zijn er verschillende resultaten uit het onderzoek voortgekomen. Dit is onderbouwd in het adviesrapport op het gebied van beveiligingslekken om zo onderscheid te maken tussen de resultaten en de oplossingen voor een betere beveiliging van de omgevingen. Hierover zal mijn scriptie in grote lijnen gaan.

Graag wil ik iedereen bedanken die heeft bijgedragen aan het tot stand komen van het eindresultaat. Allereerst Alex Slatman, mijn bedrijfsbegeleider voor de fijne begeleiding, tips en ruimte die hij mij heeft geboden om mezelf in deze technieken te verdiepen en te ondersteunen. Ook wil ik Ulrich Höxtermann en Robert Marselje bedanken voor de fijne sfeer op de afdeling en de leuke discussies, Stefan Roijers mijn procesbegeleider wil ik bedanken voor de goede ondersteuning en het beantwoorden van mijn lastige vragen. Daarnaast wil ik het hele team van VGO Direct Mail bedanken voor de mooie tijd die ik heb beleefd tijdens mijn afstudeerperiode.

Best, juni 2011
Maikel Coolen

Inhoudsopgave

Samenvatting

Dit verslag is geschreven in opdracht van VGO Direct Mail. VGO Direct mail is een servicebedrijf dat zich richt op print, mailing, datamanagement en fulfillment. Deze verzorgen ze voor een aantal grote spelers waaronder bijv. Interpolis. Doordat er steeds meer digitaal gewerkt wordt heeft VGO een tweetal omgevingen opgericht die het voor hun klanten makkelijker maakt data aan te bieden voor verwerking en eventuele wijzigingen. Hiermee besparen ze tijd, geld en levert VGO een betere service. Dit moet dus veilig zijn omdat hier met privacygevoelige gegevens gewerkt wordt.

De opdracht vanuit VGO gericht op de twee omgevingen was om deze beveiliging te testen, in kaart te brengen en de eventuele oplossingen te documenteren. Het doel wat ze hiermee bereiken is een goede kwaliteit van de beveiliging voor deze klantomgeving(en).

Er zijn twee omgevingen genaamd ViaNota en de Mijn Omgeving. De eerste omgeving genaamd “ViaNota” dient als proces beheeromgeving voor directe klanten. Terwijl de tweede omgeving “Mijn Omgeving” dient als presentatieomgeving voor de eindklant om zijn gegevens te presenteren/wijzigen. De technieken waarmee de omgevingen ontwikkeld en draaien zijn: XML, ASPX, Silverlight, WCF en MSSQL.

Het onderzoek dat is uitgevoerd geeft meer inzicht in de wereld van websecurity. Wat tegenwoordig een hot topic is. Want als je nu kijkt naar het nieuws hoor je veel berichten over websites die zijn gehackt. Kijk bijvoorbeeld naar Sony met hun PSN netwerk of de websites die zijn gehackt. Hier zijn heel veel persoonlijke gegevens buit gemaakt. Waarschijnlijk is hier niet genoeg aandacht besteed op het gebied van security anders had dit grotendeels voorkomen kunnen worden. Er zal in het onderzoek een aantal technieken genoemd worden waaronder SQL injectie, Cross-Site Scripting en komen er Websecurity Statistieken (betrekking op 2010) aan bod. Hierdoor wordt bekend waar de meeste fouten voor komen. Daarnaast zijn de omgevingen (ViaNota & Mijn Omgeving) onderzocht en bekeken of deze daadwerkelijk veilig zijn. Dit onderzoek is verwerkt in een afzonderlijk document .

Uit het onderzoek zijn een aantal resultaten gekomen die zijn verwerkt in een adviesrapport dat als afzonderlijk document wordt aangeboden. Deze geeft een duidelijk beeld van de beveiligingslekken, hoe de risico`s zijn en geeft daarnaast de oplossingen om het te voorkomen. De implementatie van de oplossing wordt niet uitgevoerd maar kunnen aan de hand van het adviesdocument wel geïmplementeerd worden.

VGO kan over het algemeen met een gerust hart terug kijken naar dit onderzoek. Er zijn in totaal 24 meldingen gevonden verspreidt over 2 omgevingen waarvan 4 stuks de classificatie “hoog” hadden. Hier moet toch even naar gekeken worden. Deze meldingen zijn met een aantal maatregelen af te vangen. Een aantal oplossingen die kunnen helpen zijn door gebruik te maken van een SSL verbinding/certificaat, foutpagina`s, WAFs en het updaten van de software.

Summary

This paper was written on behalf of VGO Direct Mail, it is a company that focuses on print, mail, data management and fulfillment. They provide these services to several large clients such as Interpolis. Because the world is becoming more and more digital, VGO created two environments that allows its clients to submit their data for processing or modifications. This saves them time and money while VGO delivers a better service. However it is very important that those environments are as safe as possible, since sensitive data is being processed.

Based on the demands, VGO wants to analyze these environments so that they are safe enough.
After analyzing these environments the issues that were found will be documented and possible solutions will be included. The goal for this assignment is to be able to create and deliver a more secure environment for their customers.

There are two environments: ViaNota and "Mijn Omgeving". ViaNota serves as a process control environment while the "Mijn omgeving" serves as a presentation environment were the end customers can view/change their data. These environments are built using several techniques such as XML, ASP.NET, Silverlight, WCF and MSSQL.

In order to successfully run this project, a research in the world of websecurity was required. At the time of writing this, websecurity is a very hot topic. Sony Playstation Network and their websites are a very good example of how things can go terribly wrong regarding security. In this research I will include several techniques such as SQL injection, Cross Site Scripting and many more. Including some statistics about websecurity. This will help me to analyze the risk and impact of each technique on the provided environments to see if the security is in proper order. The results of this analysis are documented in a separated file named "research document".

As a result of the research an "advice report" was created for VGO. This document consist of a clear explanation on all security leaks, exploits and vulnerabilities found in these two environments used by VGO. It also includes the risks and possible solutions to fix the security issues. I was not responsible for fixing these security issues however this can be done by anyone using the information in the report.

VGO can generally look back with confidence. As only 24 security issues spread on two environments are found with 4 issues classified as high vulnerability. They seriously need to look at those. Some of those issues can be prevented by implementing a SSL connection/certificate, self-made error pages, WAFs and by updating software.

Verklarende woordenlijst

Naam:	**Omschrijving**
Authenticatie	Het verifiëren van je identiteit, locatie of gebruiker.
ASP.NET:	ASP.Net is een manier om websites te maken met behulp van de programmacode. Hiermee kan dus HTML code gekoppeld worden aan programmacode.
Abuse of Functionality	Een aanvalstechniek dat zijn eigen website functionaliteiten en features gebruikt om zichzelf aan te vallen of andere websites.
Brute Force	Methode die probeert om achter een onbekende waardes te komen d.m.v. een automatisch proces die een grote hoeveelheid waardes vergelijkt.
Buffer Overflow	Een fout dat voorkomt als er meer data wordt weggeschreven in het geheugen en/of buffer dan het eigenlijk kan bevatten.
Content Spoofing	Een techniek die gebruikt wordt om de gebruiker te laten denken dat de data op een valse website legitiem is.
Cross-Site Scripting (XSS)	Een beveiligingslek voortkomend bij webapplicaties die aanvallers de gelegenheid geeft om scripting code in te voeren die bekeken wordt door gebruikers en uitgevoerd.
Cross site Request Forgery	Een aanval die een eindgebruiker van een webapplicatie forceert om onwillige acties uit te voeren door gebruik te maken van een link waarop geklikt moet worden voor toegang.
Command injectie	Een aanval die gericht is op het injecteren en uitvoeren van specifieke commando`s op de applicatie.
Encrypt	Versleuten van gevoelige data met een bepaalde algoritmen waardoor het onleesbaar wordt.
Fingerprinting	Een techniek waarmee je gegevens probeert te achterhalen door bijvoorbeeld de server te onderzoeken op bijv. versie nummers, etc.
FTE	Full Time Equivalent is een rekeneenheid waarmee de omvang van een dienstverband of de personeelssterkte wordt uitgedrukt.
HTML	Hyper Text Markup Language is een opmaaktaal voor het creëren van documenten/websites voor het internet.
Information Leakage	Er wordt belangrijke informatie vrijgegeven zonder enige bescherming.
Live CDs	Speciale cd`s waarmee je een OS omgeving rechtstreeks vanaf een cd kunt laden en hiermee dan kunt werken zonder enige installatie.
(MS) SQL	(Microsoft) **S**tructured **Q**uery **L**anguage is taal die gebruikt kan worden voor taken zoals het bevragen en het aanpassen van gegevens in een relationele databank.

Open-Source	De programmatuur is beschermd met een licentie die er voor zorgt dat iedereen toegang heeft tot de broncode.
Path Traversal	Een aanval gericht op het verkrijgen van toegang tot bestanden en mappen die buiten de standaard map vallen.
Phising	Een manier van oplichting om bij mensen hun gegevens los te krijgen d.m.v. een namaak website/email van een bekend bedrijf/persoon om je vervolgens te vragen naar inlogcodes, etc.
RDP	Remote Desktop Protocolis een meerkanaals-protocol dat gebruikt wordt om verbinding te maken met Terminal Services of voor het overnemen van een op Microsoft Windows gebaseerd systeem.
Remote File Inclusion	Remote File Inclusion is een methode waarbij de aanvaller probeert een bestand met gevaarlijke code door een webapplicatie op de server van het slachtofer uit te laten voeren.
SaaS	Software as a Service is eigenlijk niks anders dan je software als een online dienst aanbieden.
Silverlight	Een ontwikkelingsplatform waarmee complexe grafische interfaces kunnen worden ontwikkeld voor in browsers, als desktopsoftware of in mobiele apparaten
WAF	Een Webapplicatie Firewall is een firewall die voor de webapplicatie de invoer controleert op bepaalde code van bekende technieken. Denk hierbij aan technieken zoals SQL, XSS etc. Deze worden dan voorkomen door de WAF.
WCF	Windows Communication Foundation is een verzameling van .Net technieken voor het bouwen en gebruiken van netwerksystemen door gebruik te maken van webservices.
Webbased	Geeft aan dat software via het internet/intranet werkt. De software draait op een andere pc en is dus te bereiken via de webbrowser
XML	Extensible Markup Language is een standaard voor het tonen van gestructureerde gegevens in de vorm van platte tekst.

1 Inleiding

1.1 Aanleiding van het onderzoek:

Deze opdracht is uitgevoerd in opdracht van VGO Direct Mail begeleid door dhr. A. Slatman manager op afdeling software ontwikkeling. VGO Direct Mail heeft een aantal klantomgevingen ontwikkeld waarop zowel zakelijk als particuliere gebruikers kunnen inloggen. Deze omgevingen worden op een aantal webservers gehost. De beveiliging van deze websites/webservers is nog niet getest of in kaart gebracht. Deze wil VGO Direct Mail dus in kaart brengen en indien nodig de beveiliging verbeteren. Hierdoor wordt er gegarandeerd veilig omgegaan met de data waarmee gewerkt wordt.

1.2 Opbouw van scriptie:

Dit afstudeerverslag is opgedeeld in de volgende hoofdstukken:

Hoofdstuk 1:	Inleiding Waarom is dit onderzoek uitgevoerd en wat is de opbouw van de scriptie.
Hoofdstuk 2:	Bedrijfsbeschrijving Algemene informatie over het bedrijf.
Hoofdstuk 3:	Opdrachtomschrijving Hier wordt de opdracht beschreven zodat het duidelijk is waaraan gewerkt is
Hoofdstuk 4:	Vooronderzoek: Hier wordt ingegaan op de omgevingen die zijn getest en de globale aanpak
Hoofdstuk 5:	Onderzoek: Algemene informatie over websecurity, hulpmiddelen bij testen/zoeken en de resultaten.
Hoofdstuk 6:	Advies: Overzicht van resultaten tegen elkaar op gezet, ernst van de meldingen en oplossingen voor de resultaten die waren gevonden.
Hoofdstuk 7:	Oplossing(en): Een aantal meldingen uitgewerkt met de daarbij horende oplossing.
Hoofdstuk 8:	Conclusies De bevindingen die naderhand zijn getrokken na het afronden van het project.
Evaluatie	Hoe het project is verlopen, hoe kijk ik na deze afstudeerperiode ertegenaan.
Literatuurlijst	Een opsomming van alle gebruikte bronnen.

<u>Bijlagen:</u>

Bijlage A:	PID
Bijlage B:	Onderzoeksdocument
Bijlage C:	Adviesrapport

2 Bedrijfsbeschrijving

In dit hoofdstuk wordt achtergrond informatie gegeven over het bedrijf: VGO Direct Mail. Hier zal de historie van het bedrijf aanbod komen. Daarna worden een aantal klanten en diensten beschreven. Als laatste is er een organigram opgenomen om een inzicht te geven in de organisatie binnen VGO Direct Mail.

2.1 Historie:

VGO Direct Mail is opgericht in september 1998 door Roger van Gestel en Dorothé van Halder onder de naam Gestel Operationeel, wat toen gevestigd was aan de Eindhovenseweg-Zuid 142a in Best. Met een toenmalige stagiair genaamd Dave Brands, wordt deze onderneming vanuit het niets opgestart.

In september 1999 met een totaal van 8 werknemers wordt er gekozen om een nieuwe huisvestiging te zoeken/kopen en met succes, waarna ze in april 2000 zijn verhuist naar de Waal 2B in Best.

In de periodes 2001, 2002 en 2004 wordt er uitgebreid en wordt er ruimte bijgekocht.

Anno 2008 bestaat het bedrijf uit een 70-tal werknemers waarvan er in totaal ongeveer 50 FTE zijn.

Intussen in maart 2010 is VGO samen met Deskservices gevestigd in Malgraten gefuseerd om zo de klant nog beter van dienst te zijn.

2.2 Klanten:

Hieronder een klein greep uit het klantenbestand met daarin enkele welbekende en grote spelers zoals.

2.3 Diensten:

Print

VGO direct mail zorgt voor de opmaak van mailstukken met logo, afzendergegevens, de adressering en port betaald-logo. De aangeleverde bestanden voegen ze samen met een database zodat elk individueel mailstuk volgens de wensen van de klant en huisstijl wordt gepersonaliseerd.
Er kan dan geprint worden op verschillende manieren zoals:

- Inkjet printen
- Laserprinten (losblad)
- Laserprinten (ketting)
- Full Colour (losblad)
- Printen on demand

Mail

VGO direct mail zorgt ervoor dat het poststuk wordt samengesteld tot één geheel en verzendklaar zijn. De vouw-, couverteer en sealmachines kunnen in korte tijd grote partijen verwerken. Enkelvoudig of meervoudig gepersonaliseerde documenten worden gevouwen en in een verpakking of envelop gestoken.
De optie zijn als volgt:

- Couverteren
- Sealen/ folieverpakken
- Nabewerking
- Handmatig

Fulfillment

Het uitleveren van materiaal vergt een goede logistieke begeleiding. VGO direct mail slaat artikelen en documentatie op en houdt de voorraad bij.
Op aanvraag verzamelen ze de benodigde materialen en zorgen dan voor een handmatige nabewerking.

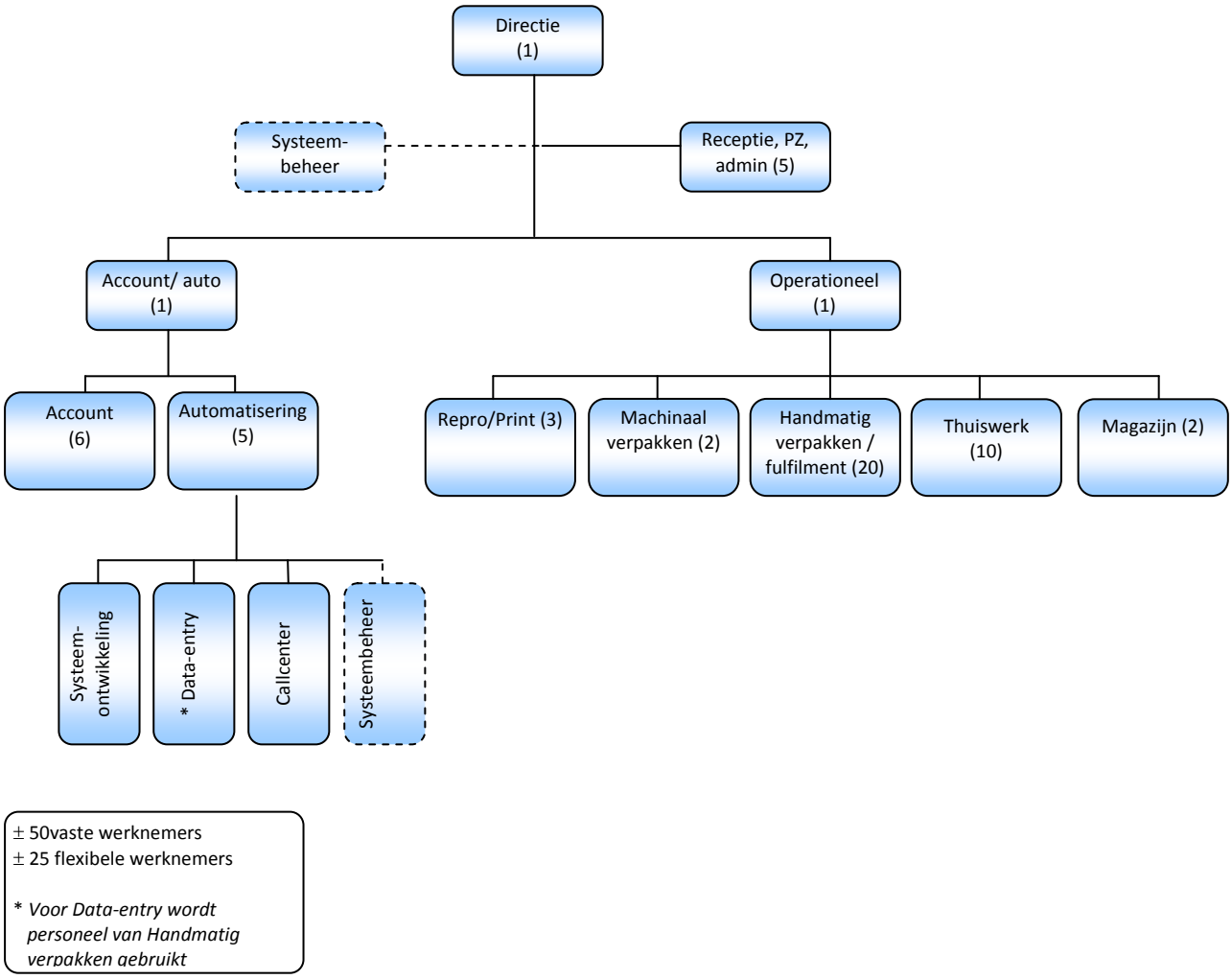
Datamanagment

VGO direct mail maakt een inventarisatie van uw informatiebehoefte. Zo biedt de structuur van de database ook op lange termijn voldoende ruimte voor de opslag van gegevens, bewerking en analyse ervan.

Door handmatige data entry, door een aangeleverd bestand of door automatisch gegenereerde data vanuit bijvoorbeeld een internetsite wordt dit in de database.
Dit bestand wordt gekoppeld aan de meest actuele postcodedata tabel.
Daarnaast wordt er goed gecontroleerd, ontdubbelt en goed verzorgt voor de verrijking van de klantgegevens.
Verloop van proces:

- Ontvangst response
- Verwerking
- Rapportage (output)
- Afwerking
- Overige

2.4 Organigram:



Bron: *http://www.vgo.nl*

3 Projectopdracht

In dit hoofdstuk wordt de opdracht die tijdens de stage is uitgevoerd verder uitgelegd. Zo wordt het probleem beschreven. Daarna zullen er een aantal doelstellingen geformuleerd zijn aan de hand van de opdracht. Uit deze twee zaken komen dan de deelvragen die als richtlijn dienen voor het onderzoek. Uiteindelijk wordt het verwachte resultaat gedocumenteerd.

3.1 Probleemstelling:

VGO Direct Mail heeft een aantal klantomgevingen waarop zowel zakelijk als particuliere gebruikers kunnen inloggen. Deze omgevingen worden op een aantal webservers gehost. De beveiliging van deze websites/webservers is nog niet getest of in kaart gebracht. Deze wil VGO Direct Mail dus in kaart brengen en indien nodig de beveiliging verbeteren.

Een van deze webomgeving genaamd ViaNota bevat namelijk miljoenen facturen van klanten. Deze dienen i.v.m. de privacy zo veilig mogelijk opgeslagen te worden volgens de wetgeving. Doordat deze niet door andere toegankelijk of ingezien kunnen worden. Er wordt dus o.a. gekeken hoe veilig dit wordt opgeslagen en/of niet andere onbevoegde(n) hierbij kunnen.

Hoofdvraag:
"Hoe goed is de beveiliging geregeld, wat zijn de resultaten van de uitgevoerde testen en wat kan er verbeterd wordt aan de webomgevingen?"

3.2 Doelstelling(en):

Het doel is om tijdens het onderzoek de kwaliteit van de beveiliging van deze klantomgevingen zo hoog mogelijk te hebben en in kaart te brengen. Naar aanleiding van het onderzoek zal gekeken worden welke verbeteringen geïmplementeerd kunnen worden.

In een tijdbestek van ongeveer 20 weken wordt er gekeken in hoeverre de beveiliging in orde is en wordt daarnaast in kaart gebracht. Daarnaast kan het zijn dat er verbeteringen worden doorgevoerd.

3.3 Deelvragen:

1. Welke technieken/methodes zijn er beschikbaar?
2. Wat zijn de meest voorkomende web security lekken?
3. Bevat de klantomgeving beveiligingslekken?
4. Wat zijn de risico`s van de beveiligingslekken?
5. Welke oplossing zijn er om dit te voorkomen?

3.4 Resultaat:

Aan het einde van deze stage zijn er aan het bedrijf een aantal producten opgeleverd die tevens als bijlage zijn toegevoegd. Het gaat hier met name om een onderzoeksdocument en een adviesdocument. Deze zullen hoogstwaarschijnlijk gaan dienen als ondersteuningsmateriaal voor verbetering van de webapplicaties/webservers.

4 Vooronderzoek

In dit hoofdstuk zal de omgeving besproken worden zodat het duidelijk is waar het onderzoek op plaats gevonden heeft. Er zal hier sprake zijn van een tweetal omgevingen waarvan de websecurity getest/onderzocht is beschikbaar gesteld door VGO. Ook komt er globaal de aanpak aanbod die ingedeeld is in fases.

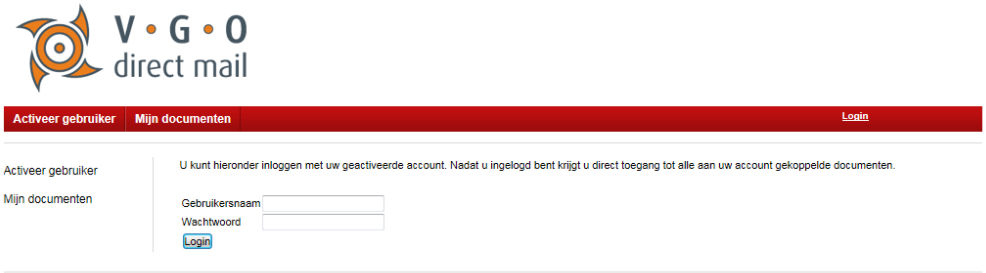
4.1 Omgeving(en)

4.1.1 ViaNota

ViaNota is een omgeving die wordt gebruikt voor het goedkeuren van processen voor verwerking. Denk hierbij aan het goedkeuren/afkeuren van het printen van brieven, facturen, etc. die daarna door VGO worden afgehandeld en geproduceerd.

4.1.2 Mijn Omgeving

Mijn omgeving is een omgeving waar klanten hun account en documenten kunnen beheren denk hierbij aan facturen en dergelijke. Deze is gebaseerd op een open source pakket die aangepast is door VGO om te voldoen aan de wensen en eisen.



4.2 Technieken

Een aantal gebruikte technieken waarmee de omgevingen zijn opgebouwd en worden gebruikt zijn:

- ASPX
- Silverlight
- WCF
- MS SQL
- XML

4.3 Aanpak:

Tijdens het uitvoeren van het project is globaal aan het tien-stappen plan van Kempen en Keizer[1] aangehouden en deze zijn opgenomen in de planning. Dit is ingedeeld in een aantal fases:

4.3.1 Oriëntatie:

In deze fase wordt er kennis gemaakt met het bedrijf. Er zal ook bij verschillende afdelingen meegelopen worden om kennis te maken met de mensen en wat de afdeling doet. Ook wordt er met de opdrachtgever afgesproken wat de opdracht inhoudt.

4.3.2 Analyse:

In deze fase wordt alles rond het project gedocumenteerd denk hierbij dus aan het afbakenen van het project door het opstellen van een PID, planning en de informatie verzamelen over het onderwerp.

4.3.3 Onderzoek/ontwikkeling:

In deze fase wordt het daadwerkelijk onderzoek gestart en zullen de hoofdvraag en deelvragen beantwoordt worden die zijn opgesteld. Ook wordt hier de omgeving onderzocht en. Waar uiteindelijk na afronding een advies over wordt uitgebracht.

4.3.4 Afronding:

In deze fase worden alle producten waaronder adviesrapport en onderzoeksdocument bij elkaar gepakt en wordt hiervan een scriptie gevormd. En zullen de resultaten beschikbaar zijn voor het bedrijf incl. oplossingen.

Deze indeling is tevens te vinden in de bijlage planning van het PID document.

[1] http://competentafstuderenenstagelopen.noordhoff.nl/

5 Onderzoek

In dit hoofdstuk wordt er algemeen gesproken over web security, denk hierbij aan o.a. de technieken, statistieken en de hulpmiddelen die geholpen hebben bij het onderzoeken van de twee betreffende omgevingen. De gevonden resultaten zullen daarnaast globaal aanbod komen met scantijden, gevonden aantallen en in welke categorie ze vallen bij OWASP Top 10.

5.1 Organisatie/Standaard(en) Web Security

Hieronder worden 2 bedrijven/organisaties besproken die zich bezig houden met het informeren over het beveiliging van webapplicaties en risico`s. Deze organisaties stellen standaarden op waarmee security professionals hun webapplicaties kunnen controleren. Ook houden ze bij welke nieuwe technieken er ontstaan wat uiteindelijk teruggekoppeld wordt zodat hierover informatie beschikbaar is. Om de 3 jaar komt er vanuit OWASP een TOP10 van meest voorkomende beveiligingsrisico`s.

5.1.1 OWASP

OWASP ofwel **O**pen **W**eb **A**pplication **S**ecurity **P**roject is een non-profit/open community/bedrijf dat advies geeft aan bedrijven om hun applicatie betrouwbaar en veilig te maken.

Top 10:
Regelmatig stelt OWASP een top tien samen met de meest voorkomende risico`s waarvoor ze dus algemene termen hebben gebruikt.
Hieronder is een lijst te vinden die geüpdate is in 2010.

1. Injection
2. Cross-Site Scripting (XSS)
3. Broken authentication and Session Management
4. Insecure Direct Object References
5. Cross Site Request Forgery (CSRF)
6. Security Misconfiguration
7. Insecure Cryptographic Storage
8. Failure to Restrict URL Access
9. Insufficient Transport Layer Protection
10. Unvalidated Redirects and Forwards

Dit is een top 10 er zijn veel meer technieken die gevaarlijk zijn.

Een uitgeschreven versie van de OWASP Top 10 is te vinden in het onderzoeksdocument – Hoofdstuk 5.

5.1.2 WASC

Web **A**pplication **S**ecurity **C**onsortium is een non-profit organisatie bestaande uit een groep met daarin internationale deskundige, mensen uit bedrijfsleven en vertegenwoordigers met kennis over beveiliging. Deze hebben samen een open-source project opgezet die een beveiligingsstandaard hebben ontwikkeld vanuit goede praktijk ervaringen.

WASC zorgt voor het uitwisselen van ideeën en organiseert verschillende projecten op het gebied van beveiliging.

WASC Threat Classification:
De onderstaande lijst is om zwakheden en aanvallen te classificeren die kunnen leiden tot een gevaar voor de website, data of gebruikers.

Aanvallen	Zwakheden
Abuse of Functionality	Application Misconfiguration
Brute Force	Directory Indexing
Buffer Overflow	Improper Filesystem Permissions
Content Spoofing	Improper Input Handling
Credential/Session Prediction	Improper Output Handling
Cross-Site Scripting	Information Leakage
Cross-Site Request Forgery	Insecure Indexing
Denial of Service	Insufficient Anti-automation
Fingerprinting	Insufficient Authentication
Format String	Insufficient Password Recovery
HTTP Response Smuggling	Insufficient Process Validation
HTTP Response Splitting	Insufficient Session Expiration
HTTP Request Smuggling	Insufficient Transport Layer Protection
HTTP Request Splitting	Server Misconfiguration
Integer Overflows	
LDAP Injection	
Mail Command Injection	
Null Byte Injection	
OS COmmanding	
Path Traversal	
Predictable Resource Location	
Remote File Inclusion (RFI)	
Routing Detour	
Session Fixation	
SOAP Array Abuse	
SSI Injection	
SQL Injection	
URL Redirector Abuse	
XPath Injection	
XML Attribute Blowup	
XML External Entities	
XML Entity Expansion	
XML Injection	
XQuery Injection	

5.2 Aanvalstechnieken

Om te weten waar we over praten bij het uitvoeren en testen van een webapplicatie op zijn veiligheid ofwel de web security is het van belang om de technieken te begrijpen. Deze zijn tijdens het onderzoek bekeken.
Een aantal technieken die aan bod gekomen zijn:

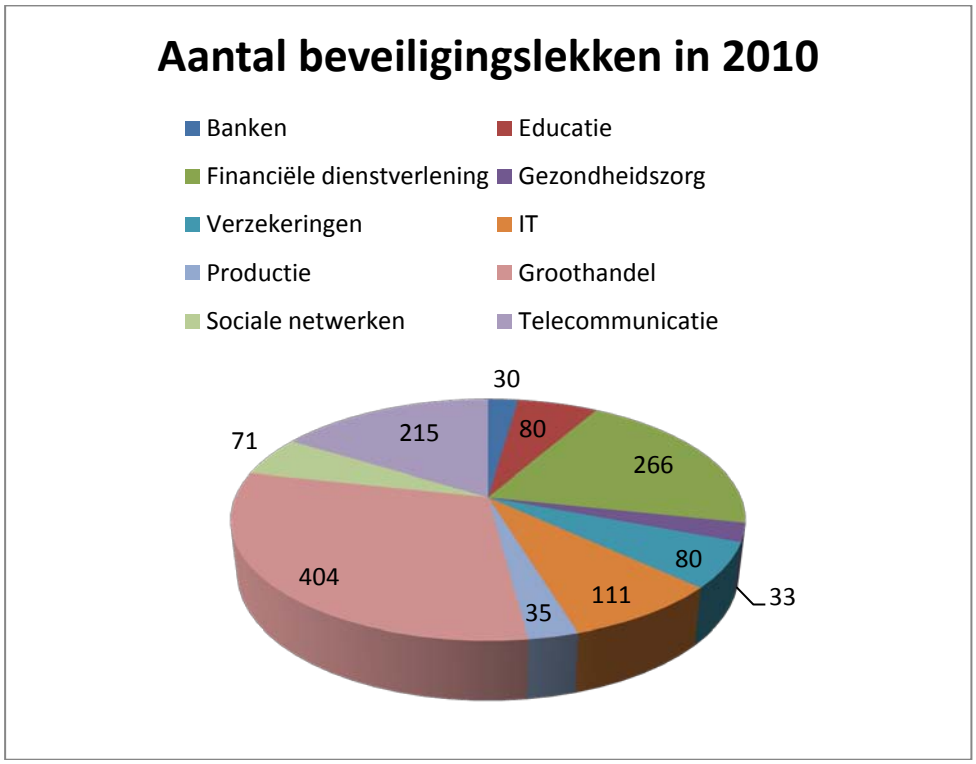
- SQL Injectie
- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery
- Fingerprinting
- HTTP Response Splitting
- Remote File Inclusion (RFI)
- Path Traversal
- Command injectie
- Buffer Overflows
- Brute-Force
- Configuratiefouten

Deze zijn onderzocht en verder gedocumenteerd in het onderzoeksdocument zie daarvoor:
Bijlage 2 – hoofdstuk 6: Technieken.

Tevens is er een volledige lijst van aanvalstechnieken te vinden in:
Bijlage 2 – hoofdstuk 4 / Paragraaf 2.

5.3 Web Security Statistieken

Web security is een belangrijk onderwerp in het nieuws de laatste tijd en blijft een belangrijk principe doordat steeds meer applicaties webbased worden uitgebracht. Zo is uit een recent onderzoek vanuit een bedrijf genaamd Whitesec[2] namelijk gebleken dat een website in een heel jaar namelijk 2010 minimaal 230 serieuze beveiligingslekken bevatte. Dit is een onderzoek geweest die gehouden is onder +/- 3000 websites van verschillende branches.



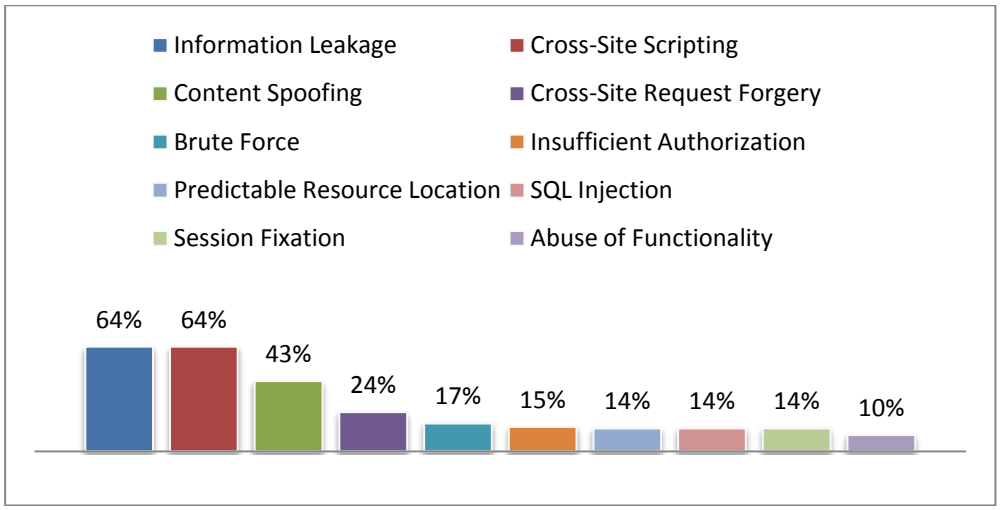
Cijfers met een doorlooptijd van heel 2010 totaal, gesorteerd op Branches en beveiligingslekken per website.

Zo zijn de gemiddelde aantal beveiligingslekken per website ongeveer 230 stuks.
Dit zijn best schokkende resultaten gezien het feit dit het over het gehele jaar loopt. Vooral de groothandel, financiële dienstverlening en productie springen eruit met vrij veel beveiligingslekken (404 - 266 - 215) wat uitkomt op een gemiddelde van 25 lekken per website per maand.

Waar aan de andere kant de banken, gezondheidszorg en verzekeringen er over het algemeen vrij weinig hebben (30 - 33 - 35) wat uitkomt op een gemiddelde van 2,5 lekken per website per maand maar uiteindelijk nog steeds lekken bevatten.

[2] http://mktg.whitehatsec.com/forms/GatedFormShort?doc=WPstats_winter11_11th

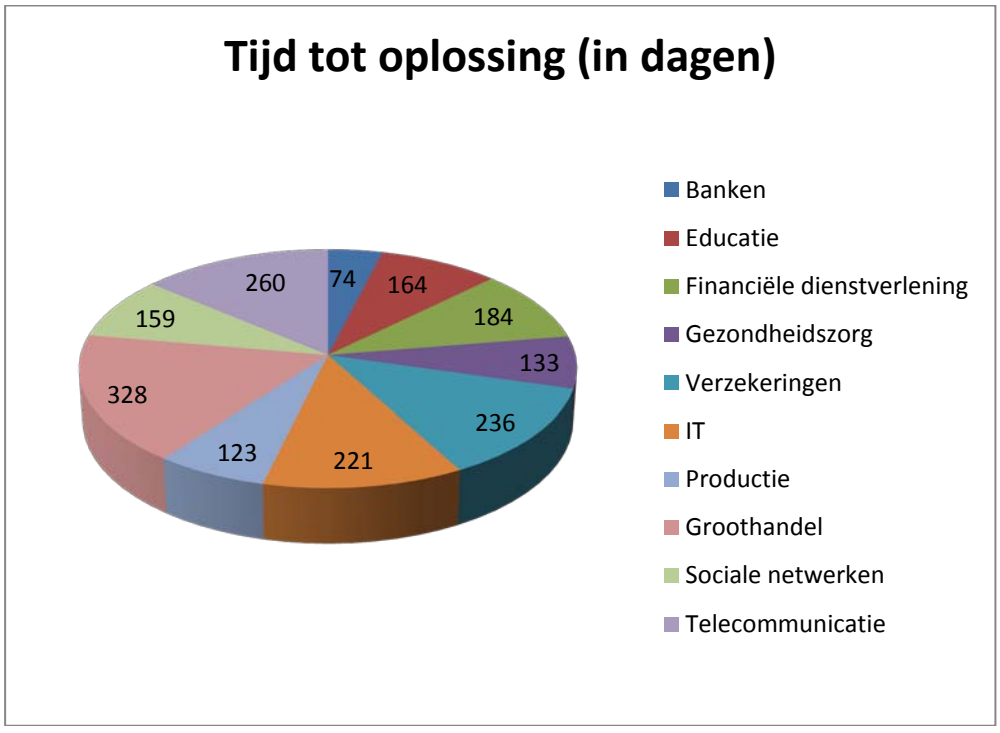
Om een overzicht te geven van de beveiligingslekken, zie onderstaande grafiek:



Top 10 Beveiligingslekken in 2010.

Zo is te zien in de bovenstaande grafiek dat in 2010 minstens 64% van de websites zeker één "Information Leakage." Lek had. Wat betekent dat er toegang was tot gevoelige informatie zoals bijvoorbeeld details over de applicatie, omgeving en gebruikers specifieke gegevens.

Er is daarnaast ook gekeken hoe lang de bedrijven erover deden om dit op te lossen.



Tijd tot het oplossen van de beveiligingslekken per branche

Zo is de gemiddelde doorlooptijd voor een oplossing ongeveer 233 dagen. Dit is redelijk lang gezien het feit dat de bedrijven in de tussentijd nog steeds de beveiligingslekken beschikbaar stellen.

5.4 Overzicht gevonden hulpmiddelen

Tijdens het zoeken naar al deze verschillende technieken is er ook gekeken naar hulpmiddelen ofwel tools/programma's die dit geautomatiseerd hebben of eventueel ondersteuning bieden bij het uitvoeren van verschillende aanvallen. Een overzicht hiervan is hieronder te vinden:

5.4.1 Commercieel:

- Acunetix WVS
- NeXpose
- NetSparker
- Retina Web Security Scanner
- Saint Exploit
- Cenzic Hailstorm
- N-Stalker
- Syhunt SandCat

5.4.2 SaaS:

- QualysGuard Web Application Scanning
- Sentinel
- VUPEN Web Application Security Scanner
- WebScanService

5.4.3 Open Source:

- Arachni
- Burp-Suite
- W3AF
- Websecurify
- Mantra

5.4.4 Live-CDs:

- Backtrack
- Samurai Web Testing Framework

Dit is nog maar een beknopte lijst. Er zijn namelijk zoveel andere spelers op dit vakgebied en het blijft groeien. Ze brengen allemaal een eigen "Web Application Security Scanner" uit.

5.5 Overzicht gekozen hulpmiddelen

Om te weten welke programma's geschikt waren zijn er een aantal gekozen uit de vorige lijst (*4.3 overzicht gevonden hulpmiddelen*) op het gebied van gratis/evaluatie versies. Door contact op te nemen met de desbetreffende makers van het programma waaronder o.a. SAINT, Qualys, Syhunt zijn er een evaluatieversie verkregen en daarnaast ook de benodigde ondersteuning bij het uitvoeren.

Deze programma`s zijn afzonderlijke van elkaar bekeken. Dit is gedocumenteerd in het *onderzoeksdocument bijlage 2 - hoofdstuk 9*. Hier zijn per programma de voor- en nadelen, een screenshot, de systeemvereisten en de kosten opgenomen. Zo is in één oogopslag te zien wat het kost en wat er nodig is om het te draaien bij het aanschaffen van deze programma`s. Uiteindelijk moeten de verschillende programma`s bij het vergelijken aan verschillende eisen voldoen om zo een goed overzicht te krijgen. De criteria eisen staan hieronder. Deze criteria eisen kunnen helpen bij het aanschaffen van één of enkele programma`s voor het controleren van de websecurity.

De criteria zijn bewust gekozen omdat bij het kiezen van een hulpmiddel voor het testen van de beveiliging van een webapplicaties:

1. De prijs graag wil weten wanneer deze eventueel aangeschaft moet/wordt door het bedrijf.
2. Makkelijk te gebruiken is en niet te veel hoeft in te vullen, configureren etc. Liefst overzichtelijk en met één druk op de knop zodat het start meteen.
3. Snel de website controleert zodat je niet uren of dagen hoeft te wachten
4. Niet te zware eisen stelt aan het systeem waarop het geïnstalleerd/gedraaid moet worden. Anders kan er in de tussentijd niet gewerkt worden.

5.5.1 Criteria:

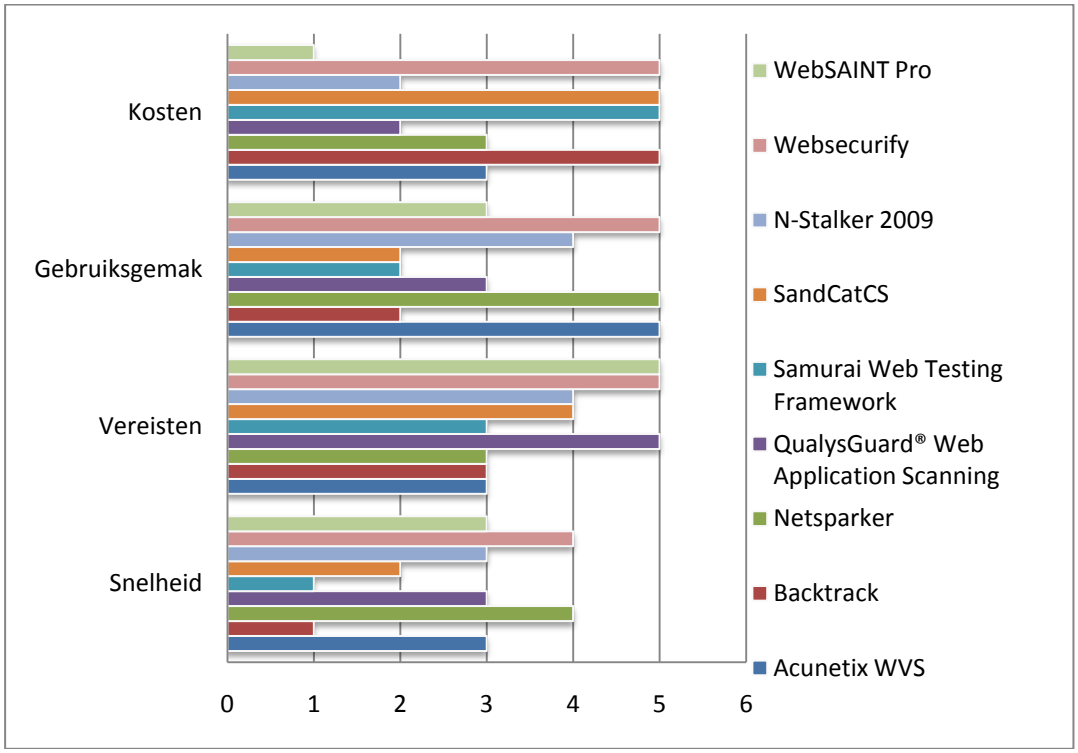
- Gebruiksgemak
- Kosten
- Snelheid
- Vereisten

Naam	Snelheid	Vereisten	Gebruiksgemak	Kosten
Acunetix WVS - Enterprise Edition	***	***	*****	***
Backtrack 4.0	*	***	**	*****
Netsparker - Community Edition	****	***	*****	***
QualysGuard® Web Application Scanning	***	*****	***	**
Samurai Web Testing Framework 0.9.5	*	***	**	*****
SAINT Exploit	***	*****	***	**
Syhunt SandCatCS	**	****	**	*****
N-Stalker 2009 - Free Edition	**	****	****	**
Websecurify	****	*****	*****	*****

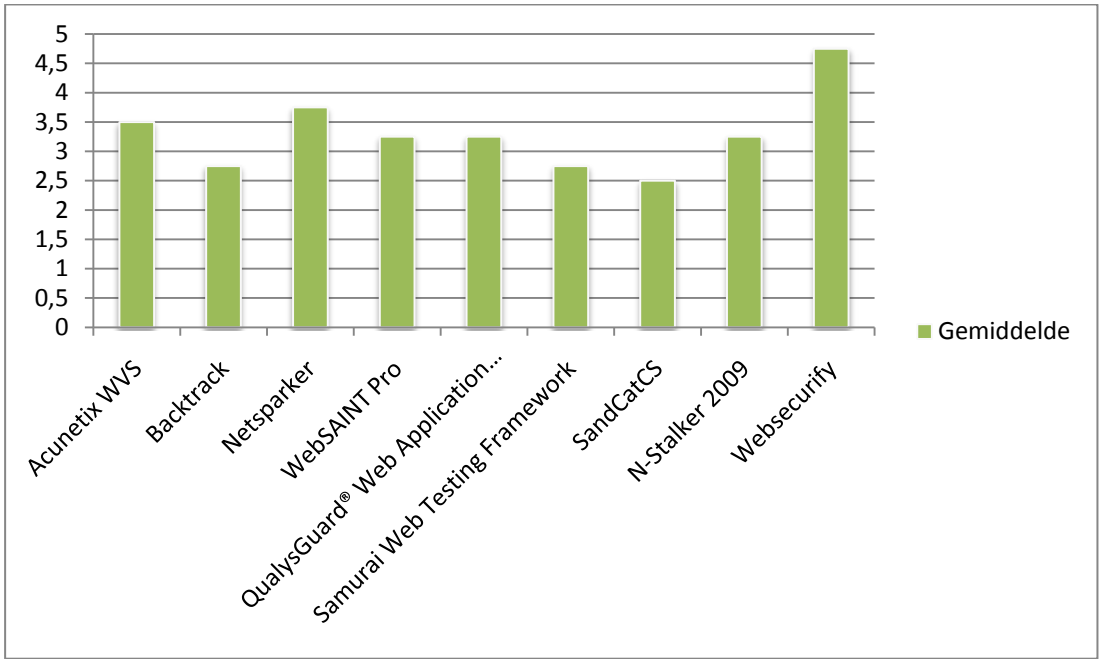
Legenda

Slecht	*
Matig	**
Redelijk	***
Goed	****
Zeer Goed	*****

5.5.2 Score per criteria:



5.5.3 Gemiddelde score:



5.6 Resultaten

De resultaten die hieronder zijn verwerkt van de twee omgevingen bevatten 10 categorieën gebaseerd op OWASP Top 10 van 2010[3]. Dit is een top 10 van meest voorkomende beveiligingsrisico`s. Dit wordt uitgebreid besproken in hoofdstuk 5 en tevens ook in het *onderzoeksdocument bijlage 2 – hoofdstuk 4/5*. De resultaten worden in het groen aangegeven in de tabel met het aantal gevonden meldingen dikgedrukt **(5)** een **(–)** betekent hier dat er meldingen zijn gevonden maar er zijn geen directe aantallen genoteerd omdat hier sprake is van meerdere gebruikte websecurity scanners en/of aanvalprogramma`s. Onder de tabel vindt je tevens de legenda met de top 10 en de scantijd (de tijd die de scanner erover heeft gedaan om het af te ronden).

5.6.1 ViaNota

			Beveiligingslekken/risico`s									
Naam	**Trail/Free**	**Versie**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Acunetix WVS Enterprise	-	7.0. 20110406						**5**				
Backtrack	Free	4.0 / R2						**-**				
Netsparker Community*	Free	1.7.2.13						**3**				
N-Stalker 2009	Free	7.0.0.223						**2**				
QualysGuard® Web Application Scanning	14d	6.18.55-1						**17**				
Samurai Web Testing Framework	Free	0.9.5						**-**				
SAINT Exploit	30d	7.7.8						**2**				
Syhunt SandCat CS	Free	4.0.3.0						**1**				
Websecurify	Free	0.8						**4**				

Notitie: *Backtrack en Samurai zijn een collectie van tools en hebben dus dubbele resultaten en daarom zijn de aantallen niet opgenomen. Ze hebben deze wel gevonden.*

Legenda:	
1	Injection (SQL and Command)
2	Cross Site Scripting (XSS)
3	Broken Authentication and Session Management
4	Insecure Direct Object References
5	Cross Site Request Forgery (CSRF)
6	Security Misconfiguration
7	Insecure Cryptographic Storage
8	Failure to Restrict URL Access
9	Insufficient Transport Layer Protection
10	Unvalidated Redirects and Forwards
*	Beperkte scan mogelijkheden

Scantijd:

Naam	**Snelheid**
Acunetix WVS EE	00u : 12m
Backtrack	-
Netsparker CE	00u :01 m
QualysGuard® Web Application Scanning	00u : 08m
Samurai Web Testing Framework	-
SAINT Exploit	00u : 15m
Syhunt SandCat CS	00u : 18m
N-Stalker 2009	00u : 04m
Websecurify	00u : 01m

[3] https://www.owasp.org/index.php/Top_10_2010-Main

5.6.2 Mijn Omgeving

			Beveiligingslekken/risico`s									
Naam	**Trail/Free**	**Versie**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Acunetix WVS Enterprise	-	7.0. 20110406			6			4			6	
Backtrack	Free	4.0 / R2		-				-				
Netsparker Community*	Free	1.7.2.13						7			1	
N-Stalker 2009	Free	7.0.0.223						2				
Samurai Web Testing Framework	Free	0.9.5		-				-				
Syhunt SandCat CS	Free	4.0.3.0	5	3				6				
Websecurify	Free	0.8						5				

Notitie: *Backtrack en Samurai zijn een collectie van tools en hebben dus dubbele resultaten en daarom zijn de aantallen niet opgenomen. Ze hebben deze wel gevonden.*

Legenda:	
1	Injection (SQL and Command)
2	Cross Site Scripting (XSS)
3	Broken Authentication and Session Management
4	Insecure Direct Object References
5	Cross Site Request Forgery (CSRF)
6	Security Misconfiguration
7	Insecure Cryptographic Storage
8	Failure to Restrict URL Access
9	Insufficient Transport Layer Protection
10	Unvalidated Redirects and Forwards
*	Beperkte scan mogelijkheden

Scantijd:

Naam	**Snelheid**
Acunetix WVS EE	00u : 12m
Backtrack	-
Netsparker CE	00u : 01m
Samurai Web Testing Framework	-
Syhunt SandCat CS	00u : 26m
N-Stalker 2009	00u : 05m
Websecurify	00u : 01m

5.6.3 Conclusie

Bij beide omgevingen is er sprake van “Security Misconfiguration”. Dit betekent dat beide omgevingen waarop gedraaid wordt niet geheel goed zijn geconfigureerd. Daarnaast zijn bij de “Mijn Omgeving” een aantal andere beveiligingsrisico`s gevonden in andere categorieën genaamd “Injection”, “Cross-site Scripting”, “Broken Authentication and Session Management” en “Insufficient Transport Layer Protection”.
Deze hebben met name betrekking op het feit hoe de data ingevoerd/verzonden wordt en/of deze voldoende wel gecontroleerd wordt.

De scantijden daarin tegen lopen geheel uiteen van 1min. tot 30min. Wat enigszins door sommige makers verbeterd kan worden. Want wanneer je een site hebt met zeer veel lekken zullen deze scanners toch wel even bezig zijn. En is dan te hopen dat ze ook daadwerkelijk alle lekken vinden gezien de tijd die ze erover doen.

Over het algemeen zijn er niet veel risicovolle beveiligingslekken gevonden. Dit is natuurlijk goed.

6 Analyse

In dit hoofdstuk komen de meldingen voor die zijn gevonden tijdens het onderzoek door de verschillende web security scanners. Deze meldingen hebben allemaal een andere prioriteit aan zich vast waaraan een risico zit verbonden. De risicoanalyse zal een overzicht brengen per melding met een classificatie van informatief tot hoge prioriteit.

6.1 Overzicht resultaten

De resultaten die gevonden zijn brachten natuurlijk ook meldingen met zich mee. Want alleen gecategoriseerde meldingen zonder enige betekenis is onvoldoende.
De twee omgevingen zijn naast elkaar gezet om ze te vergelijken en in één oogopslag te zien welke meldingen overeenkomen en hoeveel er precies bij deze omgeving gevonden zijn.

<table>
<tr><th>ViaNota</th><th>Mijn Omgeving</th></tr>
<tr><td>Silverlight:
Insecure clientaccesspolicy.xml file</td><td rowspan="2">Cross-site Scripting:
[Possible] XSRF
[Possible] XSSI
[Possible] XSS</td></tr>
<tr><td rowspan="2">ASP:
ASP.NET debugging enabled
ASP.NET Version Disclosure
ASP.NET custom error message
.NET ViewState encryption is disabled
.NET Event Validation disabled</td></tr>
<tr><td>Injection:
[Possible] SQL Injectie
[Possible] MX Injectie</td></tr>
<tr><td>IIS:
IIS Version Disclosure
Old version of Microsoft-IIS</td><td rowspan="2">ASP:
Auto Complete enabled
ASP.NET debugging enabled
ASP.NET Version Disclosure
ASP.NET custom error message
.NET ViewState encryption is disabled
.NET Event Validation disabled</td></tr>
<tr><td rowspan="2">Server:
Insecure HTTP Methods
FTP Server Banner
TCP reset using approximate sequence number
TCP timestamp requests enabled
Windows Terminal Services server running
Microsoft Windows Remote Desktop Protocol
Server Private Key Disclosure</td></tr>
<tr><td>IIS:
IIS Version Disclosure
Old version of Microsoft-IIS</td></tr>
<tr><td></td><td>Server:
Directory content listing detected
Internal Server Error
[Possible] Brute-Force at password entry forms.
Windows Terminal Services server running
User credentials are sent in clear text
Insecure HTTP methods
Internal IP Leakage
Internal Path Leakage (Windows, *nix)</td></tr>
<tr><td>Totaal: 15</td><td>Totaal: 21</td></tr>
</table>

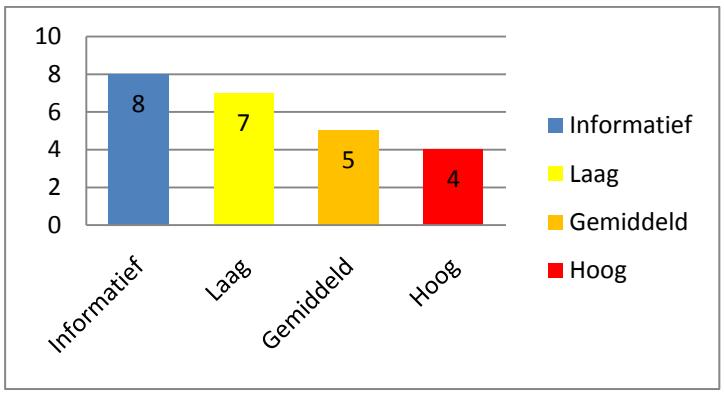
6.2 Risico-overzicht

Doordat de meldingen over het algemeen niks zeggen over een bepaalde risico worden deze hieronder geclassificeerd. Deze zijn aan de hand van verschillende scanners bepaald. Deze hebben allemaal een soort van classificatie ingebouwd zitten. De classificatie van de verschillende scanner zijn met elkaar vergeleken om zo een soort van standaard classificatie te krijgen die hieronder te vinden is. Hierdoor is in een snelle overzicht de risico van een melding te zien.

Tabel – Meldingen met classificatie:

Naam	Informatief	Laag	Gemiddeld	Hoog
Insecure clientaccesspolicy.xml file				
ASP.NET debugging enabled				
ASP.NET Version Disclosure				
ASP.NET custom error message				
.NET ViewState encryption is disabled				
.NET Event Validation disabled				
IIS Version Disclosure / Old Version of MS IIS				
Insecure HTTP Methods				
FTP Server Banner				
TCP reset using approximate sequence number				
TCP timestamp requests enabled				
Windows Terminal Services server running & Private Key Disclosure:				
[Possible] XSRF				
[Possible] XSSI				
[Possible] XSS				
[Possible] SQL Injectie				
[Possible] MX Injectie				
Auto Complete enabled				
Directory content listing detected				
Internal Server Error				
[Possible] Brute-Force at password entry forms.				
User credentials are sent in clear text				
Internal IP Leakage				
Internal Path Leakage (Windows, *nix)				

Grafiek – Overzicht classificatie/risico:



Zoals in de grafiek te zien is zijn er niet zo heel veel meldingen. Het is een dalende lijn die bij informatief naar hoog telkens zakt. De ingevulde waardes zijn daarnaast van twee omgevingen samen. Gezien de grafiek van deze twee omgevingen samen is er dus sprake van een redelijk veilig systeem.

7 Advies

In dit hoofdstuk worden een aantal oplossingen genoemd die geclassificeerd zijn met een hoge prioriteit. Omdat deze een hoge prioriteit hebben is het noodzakelijk om hier iets mee te doen. Deze oplossingen zijn hieronder uitgeschreven.

7.1 Oplossingen

Van alle gevonden meldingen is gekeken of hier een goede en veilige oplossing voor beschikbaar is. Om niet alle oplossing te documenteren is er gekozen om alleen de meldingen met de classificatie “Hoog” op te nemen.

7.1.1 Insecure clientaccesspolicy.xml file :

Clientpolicy.xml hoort over het algemeen in productieomgeving geen volledige toegang te geven. Wat nu dus het geval is door gebruik te maken van de wildcard “*”.

Om dit op te lossen is het noodzakelijk om dit te beperken. En het liefst gebruik te maken van een domein/URL i.p.v. een IP adres. Om het nog veiliger te maken is het ook nog eens slim om gebruik te maken van HTTPS verbindingen.

Code:

```
<?xml version="1.0" encoding="utf-8"?>
<access-policy>
 <cross-domain-access>
  <policy>
   <allow-from http-request-headers="*">
        <domain uri="http://mijndomain.nl" />
        <domain uri="https://mijndomain.nl" />
   </allow-from>
   <grant-to>
    <resource path="/" include-subpaths="true"/>
   </grant-to>
  </policy>
 </cross-domain-access>
</access-policy>
```

Wanneer er dus geen gebruik van wordt gemaakt kun je eventueel het bestand nog verwijderen.

7.1.2 Windows Terminal Services server running & Private Key Disclosure:

Er is geen echte oplossing bekend maar wel een soort workaround, zodat er toch veilig gewerkt kan worden met Terminal Services. Dit betekent natuurlijk dat er een goede beveiliging rondom het inloggen van terminal services gerealiseerd moet worden. De volgende authenticatie methodes zijn een optie:

1. Network Level Authenticatie
    a. TLS/SSL
    b. Kerberos
2. SSL/TLS
3. Network Level Protection
4. VPN

De beste beveiliging is NLA met Kerberos een eis van deze beveiliging is wel dat de cliënt en de server in hetzelfde of een vertrouwde domein zitten.

De onderstaande optie geldt voor VGO:
Wanneer NLA met Kerberos niet mogelijk is zou er gewoon gebruik gemaakt kunnen worden van SSL/TLS, wel moet je dan een officieel certificaat hebben. Dit wordt dan NLA met TLS/SSL.
Stappen die je moet doorlopen:

1. In de command prompt of uitvoeren, zeg tsconfig.msc
2. Dubbel klik op het RDP-TCP connectie object.
3. In de General tab, klik op Selecteer.
4. Selecteer het certificaat en klik op OK.
5. Extra:
   Hierna kun je nog policy`s instellen om zo rechten toe te kennen. Zodat er bijv. geen onveilige RDP worden toegelaten of bepaalde componenten niet toegankelijk mogen zijn.
   Te vinden in:
   Computer Configuration\Policies\Administrative Templates\Windows Components\Terminal Services

7.1.3 [Possible] SQL Injectie:

Er zijn verschillende manieren om dit te verkomen.
Een aantal manier zijn:

- Maak gebruik van SQL parameters om de ‘ karakter te voorkomen.
    - SqlCommand.CommandText = "SELECT * FROM tblTable WHERE TableID = @Parameter"; SqlCommand.Parameters.Add(new SqlParameter("@Parameter", parameterValue));
- Maak gebruik van HTMLEncode
    - Response.Write(HttpUtility.HtmlEncode(Request.Form["name"]));
- Maak gebruik van eigen error pagina`s om geen SQL data vrij te geven.
    - In web.config:

Code:

```
<System.Web>
   <customErrors mode="On" defaultRedirect="~/error/GeneralError.aspx">
      <error statusCode="403" redirect="~/error/Forbidden.aspx" />
      <error statusCode="404" redirect="~/error/PageNotFound.aspx" />
      <error statusCode="500" redirect="~/error/InternalError.aspx" />
   </customErrors>
</System.Web>
```

Omdat hier gebruik gemaakt wordt van een open source oplossing genaamd Umbraco (waar de sql lek in zit) zou er misschien een update/fix zijn die dit ook nog eens voorkomt.

7.1.4 [Possible] MX Injectie:

Een aantal maatregelen tegen MX injectie zijn bijna hetzelfde als wat er bij XSS en SQL injectie gebeurd.

1. Input validatie: Zorgen dat de input data die ingevuld mag worden geen onnodige karakters bevat.
2. Server Configuratie: Alle onnodige SMTP/IMAP onderdelen uitschakelen.
3. Applicatie Firewall: Een firewall die foute code afvangt een voorbeeld hiervan is Modsecurity.

8 Conclusie(s) & aanbevelingen

In hoofdstuk 2 zijn de doelstelling en deelvragen beschreven. Aan de hand hiervan wordt gekeken of deze zijn gehaald.

Conclusie:
In een tijdbestek van ongeveer 20 weken wordt er gekeken in hoeverre de beveiliging in orde is en wordt daarnaast in kaart gebracht. Daarnaast kan het zijn dat er verbeteringen worden doorgevoerd.
Tijdens het onderzoek is er dus gekeken naar de beveiliging van deze twee omgevingen. Hiervoor moest natuurlijk wel eerst even wat basiskennis opgedaan worden waarna uiteindelijk de beveiliging bekeken kon worden. Deze resultaten zijn genoteerd en uitgewerkt. Hiervan zijn de oplossing verwerkt zodat deze meldingen niet meer voorkomen. Dit alles is binnen 20 weken gehaald.

Hoe goed is de beveiliging geregeld, wat zijn de resultaten van de uitgevoerde testen en wat kan er verbeterd wordt aan de webomgevingen?
De beveiliging is redelijk goed geregeld er zijn wel een aantal aandachtpunten die redelijk snel te implementeren zijn maar er moet toch even naar gekeken worden. De resultaten vielen voor mijzelf een beetje tegen ik had namelijk meer resultaten verwacht maar voor het bedrijf was het juist goed omdat dit betekende dat de beveiliging toch redelijk op orde is.

Welke technieken/methodes zijn er beschikbaar?
Er zijn in de scriptie een aantal technieken en hulpmiddelen/programma`s genoemd die ondersteuning bieden bij het vinden van beveiligingsrisico`s. Daarnaast wordt er in het onderzoeksdocument veel meer informatie hierover aangeboden.

Wat zijn de meest voorkomende web security lekken?
Volgens de statistieken van 2010 gedaan door Whitesec onder +/- 3000 websites zijn de meest voorkomende risico`s te vinden bij o.a. Information Leakage, Cross-site scripting en Content Spoofing.

Bevat de klantomgeving beveiligingslekken?
De twee omgevingen samen bevatten een aantal beveiligingsrisico`s waar de meeste betrekking hebben op server configuratie.

Wat zijn de risico`s van de beveiligingslekken?
De risico`s lopen uiteen van informatief tot hoog maar er zijn niet schrikbarend veel zware tellende meldingen.

Welke oplossing zijn er om dit te voorkomen?
Deze lopen uiteen tot verwijderen van het bestand tot invoeren van enkele regels in de config bestanden of het wijzigen van instellingen in de server configuratie.

Aanbevelingen:
De aanbevelingen zijn gebaseerd op de oplossingen en bevindingen van de gevonden meldingen. Een aantal aanbevelingen die over het algemeen zeer veel afvangen en zelfs beveiligingslekken voorkomen zijn:

Error Pagina`s:
Het is slim om gebruik te maken van zelf gemaakte error pagina`s. Hierdoor vervang je de standaard foutpagina`s van IIS, deze bevatten meestal meer informatie dan nodig is. Wanneer deze beschikbaar zijn kunnen hackers informatie uit deze pagina`s verzamelen om deze te misbruiken.

SSL Certificaat:
Met het opzetten van een HTTPS verbinding met een SSL Certificaat beveilig je de verbinding onderling tussen server en cliënt. Hierdoor encrypt je de data die onderling verstuurd wordt. Tevens geef je aan de website ook een officiële identificatie. Dit wordt meestal ook gebruikt om internetfraude en phising tegen te gaan.

Web Application Firewall – WAF:
Een WAF kan heel wat bekende aanvallen tegenhouden, denk hierbij aan aanvallen zoals SQL-injecties, cross-site scripting, brute-force aanvallen en nog een aantal andere. Een web application firewall (WAF) kan dus helpen om aanvallen tegen te houden.

Software updaten:
Het is belangrijk om regelmatig de software te updaten om zo de beveiligingslekken te dichten.

Verwijderen bestanden:
Er zijn soms bestanden die niet gebruikt worden. Deze bevatten ooit informatie of geven je toegang tot.
Het is verstandig om deze bestanden te verwijderen of de toegang tot het bestand te weigeren.

Deze maatregelen kunnen ook voor andere websites gebruikt worden. Voorkomen is beter dan genezen.

Evaluatie

Terugblik:
Wanneer ik nu terugkijk op de dit project heb ik het goed gedaan en de documentatie begrijpelijk geschreven. Ik kan dus tevreden terugkijken naar het behaalde resultaat. Ik heb mijn kennis kunnen verbreden op het gebied van websecurity wat we een beetje hadden behandeld op school bij ethical hacking. En doordat mijn passie op het gebied van security ligt ook een zeer leuke uitdaging was. Waar ik dus met het VGO project dieper op ingegaan ben.

Daarnaast heb ik nog veel meer geleerd denk hierbij aan onderzoek doen, analyseren van de bronnen (bruikbaar of niet?), goed plannen, notities en brainstormen op papier (structuur verzinnen) om zo richtlijnen te volgen waar ik ongeveer naar toe wou. Dit was uiteindelijk een slim idee.
Daarnaast was de detailplanning die ik gemaakt had naar aanleiding van een tip van Alex, zeer fijn. Dit hielp me om alles goed op rails te houden en om meteen een structuur aan te brengen in mijn project.

In het begin verliep het een beetje moeizaam omdat ik niet precies wist welke kant ik op wilde. Maar naarmate de weken verstreken kwam er steeds meer vorm en structuur in en wist ik ook wat ik ongeveer wilde. Ook zijn er een aantal dingen die tegen zaten maar dan met name op softwaregebied. Soms kreeg ik zeer laat een reactie van de softwaremakers op een vraag/antwoord of reageerde ze helemaal niet op een aanvraag voor bijvoorbeeld een demoversies en werkte mijn testomgeving niet altijd mee door Linux/VMware probleem..

Zo vond ik het schrijven van de documentatie ook vrij lastig (opstellen van inhoud, keuzes wat je erin wilt hebben) maar dit is toch vrij goed gegaan, wel hier en daar spelfouten, zins opbouw, etc. Gelukkig heb ik optijd feedback gevraagd vooral bij het afronden van de documentatie. Ik kreeg zelfs een compliment vanuit Alex voor het schrijven van objectieve verslagen wie had dat verwacht.

Theorie vs praktijk:
Op het gebied van websecurity is er op school niet heel uitgebreid les gegeven wat eigenlijk jammer is, omdat er vrij veel leuke technieken zijn om dit tegen te gaan en te testen. Maar gelukkig hebben sommige theorievakken me hierin wel geholpen. Toch merk je dat er nog wel een verschil zit tussen theorie op school en praktijk in het bedrijfsleven.

Literatuurlijst

Personen:

Alex Slatman	Bedrijfsbegeleider
Robert Marselje	Systeembeheer/Programmeur
Ullrich Höxtermann	Programmeur
Stefan Roijers	Procesbegeleider

Boeken/whitepapers:

Raamwerk Beveiliging Webapplicaties	http://www.govcert.nl/dienstverlening/Kennis+en+publicaties/whitepapers/raamwerk-beveiliging-webapplicaties.html
Statistics Report 2010	http://mktg.whitehatsec.com/forms/GatedFormShort?doc=WPstats_winter11_11th
Pentesten doe je zo	http://www.govcert.nl/dienstverlening/Kennis+en+publicaties/whitepapers/pentesten-doe-je-zo.html
OWASP Top 10	http://code.google.com/p/owasptop10/downloads/list
Competent afstuderen en stagelopen	9 78001 46248

Internet:

VGO	http://www.vgo.nl/
OWASP	http://www.owasp.org/
WASC	http://www.webappsec.org/
GOVCERT	http://www.govcert.nl/
Wikipedia	http://www.wikipedia.org/
SecurityTube	http://www.securitytube.net/
PenTestIT	http://www.pentestit.com
Sony hack	http://tweakers.net/nieuws/74222/sony-hackers-maken-database-met-nederlandse-incassogegevens-buit.html

Hulpmiddelen:

Acunetix	http://www.acunetix.com/vulnerability-scanner/
Backtrack	http://www.backtrack-linux.org
Netsparker	http://www.mavitunasecurity.com
N-Stalker	http://www.nstalker.com/
QualysGuard WAS	http://www.qualys.com/products/qg_suite/was/
Samurai WTF	http://samurai.inguardians.com/
Syhunt SandCat	http://www.syhunt.com
WebSaint Pro	http://www.saintcorporation.com/index.html
Websecurify	http://www.websecurify.com/

Bijlagen

Bijlage A: Projectinitiatiedocument (PID)
Bijlage B: Onderzoeksdocument
Bijlage C: Adviesrapport