Realisatiedocument

Stage Toreon

IT-Factory

Bram Bleux 3CCS01
R0831237

Academiejaar 2022-2023

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

INHOUDSTAFEL

FIGUURLIJST

INLEIDING

Dit document is opgesteld naar aanleiding van de stage die deel uitmaakt van het traject voor de Professionele Bachelor Elektronica-ICT (specialisatie Cloud & Cybersecurity) van de Thomas-More hogeschool te Geel.

In dit document zal informatie terug te vinden zijn over de stageopdrachten.

Voor elk van de drie verkregen opdrachten zullen volgende punten besproken worden:

- De oorspronkelijke opdracht
- De gebruikte technologieën
- Het verloop
- De problemen
- Het resultaat

De stage is een periode van 13 weken, waar de stagestudent voltijds zal meedraaien in het bedrijf waar deze aan de stageopdracht(en) kan werken. Deze 13 weken gaan van 27/02/2023 tot en met 26/05/2023.

In dit document zal sporadisch verwezen worden naar “we” of “wij”. Dit verwijst naar mijn mede student / stagiair van Thomas more Matthias Minschart en mezelf. Samen hebben we deze stage tot een goed einde kunnen brengen. Per opdracht zal er duidelijk aangegeven worden wie welk deel van de opdracht gerealiseerd heeft.

De geschreven code zal niet volledig opgenomen worden in dit document. Om deze code te raadplegen, gelieve contact op te nemen met Steven Wierckx, één van mijn stagementoren.

1 OPDRACHT 1: AUTOMATISATIE SECURITYREPORTER & CWE

1.1 De oorspronkelijke opdracht

Binnen Toreon wordt er gebruik gemaakt van SecurityReporter (of Reporter in het kort). Deze tool wordt gebruikt voor het genereren van rapporten aan het einde van een verkocht project.

Als Toreon een penetratietest verkoopt, worden alle gevonden zwakheden opgenomen in Reporter. Voor zwakheden die vaker voorkomen worden templates aangemaakt. In dit template zit een stadaard tekst die van toepassing is voor de desbetreffende zwakheid, maar ook een verwijzing naar een CWE-nummer. Dit wordt gebruikt om te kijken of de software voldoet aan bepaalde standaarden binnen de cybersecurity.

Sommige van deze templates hebben nog geen CWE-nummer. Aan ons werd gevraagd om een programma te schrijven om dit te automatiseren / te vereenvoudigen.

1.2 De gebruikte technologieën

1.2.1 CWE[1]

Figuur 1: CWE logo

CWE staat voor Common Weakness Enumeration en is een gemeenschappelijke classificatie van softwarebeveiligingsproblemen die worden veroorzaakt door zwakke punten in softwareontwerp, -implementatie of -configuratie.

CWE is ontwikkeld door het MITRE Corporation, een non-profitorganisatie die werkt aan het oplossen van complexe problemen in de nationale veiligheid, cybersecurity en gezondheidszorg. Het doel van CWE is om een gemeenschappelijke taal te bieden voor het beschrijven en categoriseren van beveiligingsproblemen in software, zodat deze problemen gemakkelijker kunnen worden geïdentificeerd, gerapporteerd en opgelost.

CWE bevat momenteel meer dan 1.000 zwakke punten die zijn georganiseerd in een hiërarchische structuur. Elk beveiligingsprobleem is toegewezen aan een uniek identificatienummer en wordt beschreven met een korte tekstuele beschrijving, voorbeelden van hoe het kan worden misbruikt en suggesties voor het voorkomen ervan.

CWE wordt veel gebruikt door softwareontwikkelaars, beveiligingsprofessionals en andere belanghebbenden om beveiligingsproblemen in software te identificeren en op te lossen. Het wordt ook gebruikt door organisaties om te voldoen aan regelgeving op het gebied van informatiebeveiliging, zoals de Payment Card Industry Data Security Standard (PCI DSS).

De CWE classificatie wordt binnen Toreon gebruikt voor het categoriseren van zwakheden die gevonden worden bij projecten. Ook kan er op basis van deze classificatie gekeken worden of het bedrijf in kwestie al dan niet voldoet aan bepaalde security standaarden, waaronder de OWASP Top 10. Dit wordt later in het document duidelijk.

[1] https://cwe.mitre.org/

1.2.2 GitHub[2]

GitHub is een webgebaseerd platform dat is ontworpen om softwareontwikkeling te ondersteunen en te faciliteren. Het biedt een opslagplaats (repository) voor code waar ontwikkelaars kunnen samenwerken aan softwareprojecten en code kunnen delen met andere ontwikkelaars. GitHub maakt gebruik van Git, een populaire versiebeheersysteem, waarmee ontwikkelaars wijzigingen in de code kunnen bijhouden en beheren.

Op GitHub kunnen gebruikers openbare en privé-repositories aanmaken, waarin ze code kunnen opslaan en delen. Andere ontwikkelaars kunnen bijdragen aan deze repositories door veranderingen (pull requests) voor te stellen en te implementeren. Het platform biedt ook mogelijkheden voor code reviews en discussies, zodat ontwikkelaars kunnen samenwerken en de kwaliteit van de code kunnen verbeteren.

Figuur 2: GitHub Logo

Naast het hosten van repositories, biedt GitHub ook mogelijkheden voor projectmanagement en issue tracking, waarmee ontwikkelaars taken en bugs kunnen bijhouden. GitHub is ook populair voor open source projecten, waarbij code vrij toegankelijk is voor andere ontwikkelaars om bij te dragen en te verbeteren. GitHub wordt gebruikt door individuele ontwikkelaars, kleine teams en grote organisaties over de hele wereld.

GitHub wordt binnen Toreon gebruik voor het beheer van verschillende software projecten. GitHub maakt het mogelijk om aan versie-beheer te doen, als ook opslag en delen van deze projecten / code.

[2] https://github.com/

1.2.3 SecurityReporter[3]

Figuur 3: SecurityReporter logo

Reporter is een tool die wordt gebruikt voor het genereren van rapporten binnen bedrijven. Het is ontworpen om bedrijven te helpen belangrijke gegevens te verzamelen en deze gegevens vervolgens te gebruiken om bruikbare rapporten te genereren die kunnen worden gebruikt voor verschillende doeleinden, waaronder: financiële rapporten, verkooprapporten, marketingrapporten, klanttevredenheidsrapporten en meer.

Reporter heeft een intuïtieve interface waarmee gebruikers gemakkelijk rapporten kunnen samenstellen en de rapportopmaak te configureren. Gebruikers kunnen vaak ook aangepaste rapporten maken op basis van hun specifieke behoeften en vereisten.

Binnen Toreon wordt deze tool gebuikt voor het opstellen van rapporten over de uitgevoerde penetratietesten voor de klant. In SecurityReporter worden er templates aangemaakt voor de zwakheden die men tegen komt. Deze templates worden dan aangepast en in een assessment samengebracht. Eens de testen klaar zijn, kan een rapport gegenereerd worden waar men zeker is van een consistente lay-out.

[3] https://securityreporter.app/

1.2.4 Python[4]

Python is een algemene programmeertaal die wordt gebruikt voor een breed scala aan toepassingen, zoals webontwikkeling, gegevensanalyse, machine learning, wetenschappelijke berekeningen, automatisering van taken en nog veel meer.

Één van de belangrijkste kenmerken van Python is de leesbaarheid van de code. Python-code is zeer leesbaar en begrijpelijk, waardoor het gemakkelijk is voor nieuwe programmeurs om te leren en te begrijpen. Het wordt vaak beschouwd als een "beginnersvriendelijke" taal vanwege de eenvoud van de syntax.

Python heeft ook een grote standaardbibliotheek met veel nuttige modules en functies die het gemakkelijker maken om veelvoorkomende taken uit te voeren. Het heeft ook een actieve gemeenschap van ontwikkelaars die veel nuttige modules en pakketten hebben ontwikkeld voor specifieke toepassingen.

Figuur 4: Python logo

Een ander voordeel van Python is dat het platformonafhankelijk is, wat betekent dat het kan worden uitgevoerd op verschillende besturingssystemen zoals Windows, Linux en macOS.

Over het algemeen wordt Python beschouwd als één van de meest populaire en nuttige programmeertalen vanwege de eenvoud van de syntax, leesbaarheid van de code en het brede scala aan toepassingen.

Alle opdrachten van de stage werden gecodeerd in Python. Dit door de voorkennis waarover ik beschik, alsook omdat er veel bibliotheken voor beschikbaar zijn die het programmeren kunnen versnellen / vergemakkelijken.

[4] https://www.python.org/

1.2.5 MITRE SANS top 25[5] [6]

Figuur 5: MITRE SANS Top 25 logo

De MITRE SANS top 25 is een lijst met de 25 meest kritieke beveiligingszwaktes (CWE's) in software. Het is een samenwerking tussen MITRE Corporation en het SANS Institute, een organisatie die zich richt op cybersecurity-onderzoek, training en certificering.

De MITRE SANS top 25 richt zich op de meest kritieke beveiligingszwaktes die worden misbruikt door cybercriminelen en die een ernstig risico vormen voor de beveiliging van software. De lijst wordt regelmatig bijgewerkt en geeft beveiligingsprofessionals een duidelijk overzicht van de zwaktes die ze moeten aanpakken om hun software te beschermen tegen cyberaanvallen.

De MITRE SANS top 25 is bedoeld als een gids voor beveiligingsprofessionals om hun software te beschermen tegen de meest voorkomende en gevaarlijke beveiligingszwaktes. Het kan ook worden gebruikt door ontwikkelaars om hun software vanaf het begin te beveiligen en te testen op de meest voorkomende zwaktes.

1.2.6 OWASP ASVS[7]

Figuur 6: OWASP logo

De OWASP Application Security Verification Standard (ASVS) is een gestandaardiseerd testkader dat wordt gebruikt om de beveiliging van webapplicaties te beoordelen. Het is ontwikkeld door de Open Web Application Security Project (OWASP), een non-profitorganisatie die zich richt op het verbeteren van de beveiliging van software.

De ASVS bevat een gedetailleerde lijst van beveiligingsvereisten die moeten worden geëvalueerd om de beveiliging van een webapplicatie te beoordelen. De lijst is opgesplitst in drie niveaus, die variëren van basisbeveiligingsvereisten tot geavanceerde beveiligingsvereisten.

De ASVS kan worden gebruikt door auditors om de beveiliging van webapplicaties te beoordelen en te verbeteren. Het testkader biedt een gestandaardiseerde aanpak voor het beoordelen van beveiligingsrisico's en het identificeren van zwakke punten in een webapplicatie.

Door de ASVS te gebruiken, kunnen organisaties zorgen voor een consistente aanpak van beveiligingstests en ervoor zorgen dat alle aspecten van de beveiliging van een webapplicatie worden beoordeeld. Dit kan helpen om beveiligingsrisico's te verminderen en de kans op succesvolle cyberaanvallen op de applicatie te verminderen.

In het kort is de OWASP ASVS een gestandaardiseerd testkader dat kan worden gebruikt om de beveiliging van webapplicaties te beoordelen en te verbeteren, door middel van het evalueren van gedetailleerde beveiligingsvereisten die zijn opgesplitst in drie niveaus.

---

[5] https://cwe.mitre.org/top25/archive/2022/2022_cwe_top25.html
[6] https://www.darkreading.com/application-security/memory-corruption-issues-lead-2021-cwe-top-25
[7] https://owasp.org/www-project-application-security-verification-standard/

1.2.7 OWASP top 10[8]

De OWASP Top 10 is een lijst met de tien meest voorkomende beveiligingsrisico's voor webapplicaties. Het is samengesteld door de Open Web Application Security Project (OWASP), een non-profitorganisatie die zich richt op het verbeteren van de beveiliging van software.

Figuur 7: OWASP Top 10 logo

De OWASP Top 10 is bedoeld om beveiligingsprofessionals en ontwikkelaars te helpen de meest voorkomende beveiligingsrisico's te begrijpen en zich te richten op het beveiligen van webapplicaties tegen deze risico's. De lijst wordt regelmatig bijgewerkt en bevat de volgende tien categorieën:

1. **Injecties**

   Dit is het risico van het invoeren van kwaadaardige code via invoervelden zoals zoekvelden of gebruikersnamen.

2. **Broken Authentication and Session Management**

   Dit is het risico van onjuiste authenticatie en sessiebeheer, zoals zwakke wachtwoorden of sessiegegevens die worden opgeslagen in cookies zonder encryptie.

3. **Cross-Site Scripting (XSS)**

   Dit is het risico van het invoeren van kwaadaardige code via invoervelden of andere gebruikersinterfaces, die kan leiden tot het uitvoeren van kwaadaardige code op de computer van de gebruiker.

4. **Broken Access Control**

   Dit is het risico van onvoldoende controle over wie toegang heeft tot welke gegevens of functies in de applicatie.

5. **Security Misconfiguration**

   Dit is het risico van onjuist geconfigureerde beveiligingsinstellingen, zoals onveilige standaardwachtwoorden of toegangsrechten.

[8] https://owasp.org/Top10/

6. **Insecure Cryptographic Storage**

   Dit is het risico van het opslaan van gevoelige gegevens zonder encryptie, waardoor deze gegevens kwetsbaar zijn voor diefstal.

7. **Insufficient Transport Layer Protection**

   Dit is het risico van onvoldoende beveiliging van de communicatie tussen de client en de server, waardoor gegevens kunnen worden onderschept en gestolen.

8. **Unvalidated Redirects and Forwards**

   Dit is het risico van het doorsturen van gebruikers naar onveilige locaties via links of redirects, waardoor ze kunnen worden blootgesteld aan kwaadaardige websites.

9. **Insecure Communication Between Components**

   Dit is het risico van onveilige communicatie tussen verschillende componenten van de applicatie, waardoor deze kwetsbaar zijn voor aanvallen.

10. **Improper Error Handling**

    Dit is het risico van onjuiste of onvolledige foutafhandeling, waardoor kwaadaardige gebruikers informatie kunnen verkrijgen over de interne werking van de applicatie.

De OWASP Top 10 is een belangrijk hulpmiddel voor het begrijpen en beheren van de meest voorkomende beveiligingsrisico's voor webapplicaties. Door aandacht te besteden aan deze risico's kunnen beveiligingsprofessionals en ontwikkelaars de beveiliging van webapplicaties verbeteren en de kans op succesvolle cyberaanvallen verminderen.

1.3 Het verloop

1.3.1 Kick-off

Binnen Toreon is Georges deels verantwoordelijk voor quality assurance (QA). Tijdens de eerste meeting werd er verteld hoe er met Reporter gewerkt wordt binnen Toreon. Aan ons werd gevraagd om een programma te schrijven dat het controleren / toevoegen van correct CWE nummers aan een template te vereenvoudigen. Dit zou eventueel op basis van de beschrijving van de template gebeuren.

1.3.2 Onderzoek naar het CWE-framework

Na de kick-off meeting heb ik wat onderzoek gedaan naar het CWE framework. Zelf had ik er nog geen ervaring mee. Na wat onderzoek bleek dit complexer in elkaar te zitten dan eerder gedacht.

Het CWE-framework is opgedeeld in verschillende classificaties, namelijk: View, Category, Pilar, Class, Base, Variant. De Views en Categorieën worden gebruikt om zwakheden te groeperen gebaseerd op een bepaald onderwerp. De zwakheden worden dan op hun beurt ook opgedeeld van algemeen (Pilar), naar specifiek (Variant).

In het totaal (tijdens het opstellen van dit document) zijn er 933 zwakheden opgenomen in het CWE-framework.

Deze zal als basis gebruikt worden voor één van de stageopdrachten. Dit wordt later duidelijk.

1.3.3 Onderzoek naar een Python library voor CWE's

Om makkelijker te navigeren door de CWE's ben ik op zoek gegaan naar een Python library over CWE's. Tijdens dit onderzoek is gebleken dat de reeds bestaande libraries niet meer operationeel of up-to-date zijn met de huidige versie. Dit maakte, dat ik deze libraries niet kon gebruiken voor mijn project. Dus ik moest een oplossing zoeken.

Al snel besloot ik om zelf mijn eigen library te schrijven met de functionaliteiten dit ik nodig zou hebben.

1.3.4 Coderen van een python library voor CWE's

Op de website van het MITRE is er een XML bestand beschikbaar waar alle informatie over de huidige CWE's beschikbaar is[9]. Deze werd gebruikt als basis voor het schrijven van een Python library voor het CWE framework.

Figuur 8: Screenshot van de XML info over CWE-1266

Na het coderen van de library werd duidelijk dat het CWE framework meer complex in elkaar stak dan eerder gedacht. Hier werd ook duidelijk dat de opdracht niet mogelijk zou zijn aangezien de relaties tussen de zwakheden geen mooie boomstructuur had zoals eerder gedacht. Het was niet mogelijk om van een beschrijving van een zwakheid die gevonden werd bij een test de meest specifieke / correcte CWE te vinden met een programma.

Ik heb besloten om de basis functies te verfijnen, zodat deze toch nog nuttig zouden zijn voor andere projecten in de toekomst.

De library is zo geschreven dat deze altijd up-to-date blijft met de laatste versie, zolang de structuur van de XLM file van MITRE constant blijft.

1.3.5 Coderen van unit-tests

Na het schrijven van de Python library werden unit-tests geschreven. Deze werden geschreven met behulp van de unittest library die standaard in Python beschikbaar is. Hier werden zowel edge-cases als standaard functionaliteiten getest.

```python
def test_get_cve_by_name(self):
    """
    Test based on CWE-2020-5243 (last_change: 14-04-2023)
    Source: https://www.cve.org/CVERecord?id=CVE-2020-5243
    """
    # --Testing if attributes are correct-- #
    cve = helper.get_cve_by_name('CVE-2020-5243')
    self.assertEqual(cve.name, 'CVE-2020-5243')
    self.assertEqual(cve.link, 'https://www.cve.org/CVERecord?id=CVE-2020-5243')
    self.assertEqual(cve.description, 'server allows ReDOS with crafted User-Agent strings, due to overlapping capture groups that cause excessive backtracking.')

    # Check if related CWEs are correct
    for cwe_id in cve.related_cwes:
        self.assertIn(cwe_id, [1333, 405, 407])
```

Figuur 9: Screenshot van een unit-test

[9] Download link XML bestand (versie 4.10): https://cwe.mitre.org/data/xml/cwec_v4.10.xml.zip

1.3.6 Verandering van de opdracht

Tijdens de status-update meeting werd vermeld dat de oorspronkelijke opdracht geen meerwaarde zou zijn. Het zou even snel, of zelfs sneller zijn om de CWE's te controleren aan de hand van de website van het MITRE zelf. Dit werd met een kleine demo duidelijk gemaakt.

Later in de meeting stelde de klant een andere opdracht voor:

Er bestaan bepaalde standaarden in de cybersecurity wereld. Sommige van deze standaarden gebruiken het CWE framework om deze standaarden te bepalen. De klant stelde voor om tabellen te genereren op basis van de uitgevoerde testen om te kijken voor welke standaard de applicatie slaagt. De voorbeelden die aangehaald werden waren: OWASP Top 10, SANS MITRE Top 25 en OWASP ASVS.

Al snel kwam de opmerking van de kant dat het cruciaal is dat de CWE's die aan de templates gekoppeld zijn binnen Reporter heel accuraat moeten zijn. De klant was het hier mee eens en vertelde dat dit niet onze verantwoordelijkheid zou zijn.

Ook werd aangehaald dat ~68% van de templates in Reporter nog geen verbonden CWE hadden. Dit bleek uit een onderzoek van mede student / stagiair van Thomas More Matthias Minschart.

Er werd voorgesteld, om aan elk template caution tags toe te voegen. Deze zouden ervoor moeten zorgen dat op termijn alle templates een accurate CWE nummer hebben. De klant heeft ons een flow-chart bezorgd met de caution tags die toegevoegd moesten worden op basis van aantal CWE's die momenteel aan de template gekoppeld zijn. Dit zou variëren van 0 tot 2(+) CWE's.

Conclusie: De opdracht van dit project veranderde naar:

- Het genereren van tabellen (in markdown formaat) om na te kijken of een applicatie voldoet aan een bepaalde standaard op basis van de findings die tijdens een test gevonden zijn. De standaarden waarop gecontroleerd moet worden zijn:
    - OWASP Top 10
    - SANS MITRE Top 24
    - OWASP ASVS
- Het toevoegen van caution tags aan de templates binnen Reporter om zo op termijn de kwaliteit en het aantal CWE's gelinkt aan een template te verbeteren.

Het genereren van de tabellen nam ik voor mijn rekening, terwijl Matthias in stond voor de caution tags toe te voegen aan de templates binnen Reporter. Dit betekent echter niet dat we elkaar niet verder hebben geholpen met problemen die we ondervonden.

Na de meeting vroeg Robbe (onze stagementor) ook nog een programma te schrijven dat een overzicht zou bieden over de stand van zaken van de templates in Reporter. Dit overzicht zou kijken naar:

- Hoeveel templates x aantal CWE's toegewezen hebben.
    - Met uitbreiding een lijst met links naar deze templates. (Gesorteerd op aantal CWE's per template.)
- Welke type CWE's er aan de templates gekoppeld zijn.
    - Met uitbreiding een lijst met links naar deze templates. (Gesorteerd op type CWE's gebruikt per template.)

1.3.7 Onderzoek naar de verschillende standaarden

Om tabellen te kunnen genereren die een overzicht bieden of een applicatie wel of niet voldoet aan bepaalde standaarden is er eerst onderzoek gedaan naar welke CWE's er gekeken moet worden.

Voor de OWASP Top 10 en SANS MITRE Top 25 was dit niet moeilijk aangezien deze beschikbaar zijn in het CWE framework. Wel werd duidelijk dat deze CWE's niet altijd specifiek waren. Sommige CWE's waren een categorie waar meerdere verschillende CWE's onder vallen. Als het zou blijven bij één niveau zou dit geen probleem gevormd hebben, maar dit was niet het geval. Aangezien de structuur van het CWE framework zo verweven was, zou dit heel moeilijk te controleren zijn. Veelal zouden er circulaire redeneringen ontstaan.

Na bespreking met zowel mijn mentor als de klant is besloten om zoveel mogelijk CWE's aan een template in Reporter toe te voegen. Er wordt alleen gecontroleerd op het eerste niveau van CWE's die in de standaard worden vermeld.

1.3.8 Genereren van een tabel voor OWASP Top 10

Er werd een functie geschreven die op basis van een lijst met gevonden zwakheden (CWE-nummers) zou valideren welke categorieën van de OWASP Top 10 voldaan zouden zijn. Deze tabel werd in markdown geschreven. De iconen die gebruikt worden zijn specifiek voor SecurityReporter, maar kunnen aangepast worden door twee variabelen aan te passen in de code.

Het is zo gecodeerd dat alleen het jaartal van de laatste versie van de OWASP Top 10 aangepast moet worden om de testen te updaten. Dit jaartal wordt opgeslagen in de .env file.

Category	Pass
A01 - Broken Access Control	✔
A02 - Cryptographic Failures	✔
A03 - Injection	✔
A04 - Insecure Design	✖
A05 - Security Misconfiguration	✔
A06 - Vulnerable and Outdated Components	✔
A07 - Identification and Authentication Failures	✔
A08 - Software and Data Integrity Failures	✔
A09 - Security Logging and Monitoring Failures	✔
A10 - Server-Side Request Forgery (SSRF)	✔

Figuur 10: Voorbeeldtabel voor OWASP Top 10

1.3.9 Genereren van een tabel voor MITRE SANS Top 25

Er werd een functie geschreven die op basis van een lijst met gevonden zwakheden (CWE-nummers) zou valideren welke CWE's van de SANS Top 25 voldaan zouden zijn. Deze tabel werd in markdown geschreven. De iconen die gebruikt worden zijn specifiek voor Reporter, maar kunnen aangepast worden door twee variabelen aan te passen in de code.

Het is zo gecodeerd dat alleen het jaartal van de laatste versie van de Sans Top 25 aangepast moet worden om de testen te updaten. Dit jaartal wordt opgeslagen in de .env file.

CWE	Name	Pass
CWE-787	Out-of-bounds Write	✔
CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	✔
CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	✔
CWE-20	Improper Input Validation	✔
CWE-125	Out-of-bounds Read	✔
CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	✔
CWE-416	Use After Free	✔
CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	✔
CWE-352	Cross-Site Request Forgery (CSRF)	✔
CWE-434	Unrestricted Upload of File with Dangerous Type	✔
CWE-476	NULL Pointer Dereference	✔
CWE-502	Deserialization of Untrusted Data	✔
CWE-190	Integer Overflow or Wraparound	✔

Figuur 11: Voorbeeldtabel voor MITRE SANS Top 25

1.3.10 Genereren van een tabel voor OWASP ASVS

Req ID	Req Desc	CWE	Level 1 Pass	Level 2 Pass	Level 3 Pass
V1.1.1	Verify the use of a ...	N/A	N/A	N/A	N/A
V1.1.2	Verify the use of threat ...	CWE-1053	N/A	✔	✔
V1.1.3	Verify that all user stories ...	CWE-1110	N/A	✔	✔
V1.1.4	Verify documentation and justification of ...	CWE-1059	N/A	✔	✔
V1.1.5	Verify definition and security analysis ...	CWE-1059	N/A	✔	✔
V1.1.6	Verify implementation of centralized, simple ...	CWE-637	N/A	✔	✔
V1.1.7	Verify availability of a secure ...	CWE-637	N/A	✔	✔
V1.2.1	Verify the use of unique ...	CWE-250	N/A	✔	✔
V1.2.2	Verify that communications between application ...	CWE-306	N/A	✔	✔
V1.2.3	Verify that the application uses ...	CWE-306	N/A	✔	✔
V1.2.4	Verify that all authentication pathways ...	CWE-306	N/A	✔	✔
V1.4.1	Verify that trusted enforcement points, ...	CWE-602	N/A	✔	✔
V1.4.4	Verify the application uses a ...	CWE-284	N/A	✔	✔
V1.4.5	Verify that attribute or feature-based ...	CWE-275	N/A	✔	✔

Figuur 12: Voorbeeldtabel voor OWASP ASVS (per requirement, voor alle levels)

Er werd een functie geschreven die op basis van een lijst met gevonden zwakheden (CWE-nummers) zou valideren welke requirement, section of chapter van de OWASP ASVS voldaan zouden zijn. Deze tabel werd in markdown geschreven. De iconen die gebruikt worden zijn specifiek voor Reporter, maar kunnen aangepast worden door twee variabelen aan te passen in de code.

Het is zo gecodeerd dat alleen de link naar de RAW csv file van de laatste versie van de OSWAP ASVS aangepast moet worden om de testen te updaten. Deze link wordt opgeslagen in de .env file.

De tabel is zeer modulair gecodeerd. Het is mogelijk om een selectie te maken van naar welk level er gekeken moet worden (level 1 t.e.m. level 3) alsook hoe diep men wil ingaan op de requirements:

1. Per chapter (Vx)
2. Per section (Vx.x)
3. Per requirement (Vx.x.x)

Als er wordt gekeken per requirement, zal er ook rekening gehouden worden met welk level er geselecteerd is. Als een requirement niet van toepassing is voor de geselecteerde levels, zal deze ook niet opgenomen worden in de tabel. Dit voorkomt dat de tabel onnuttige informatie weergeeft. Ook de requirements die als [DELETED] in de lijst staan, zullen uiteraard niet in de tabel worden opgenomen.

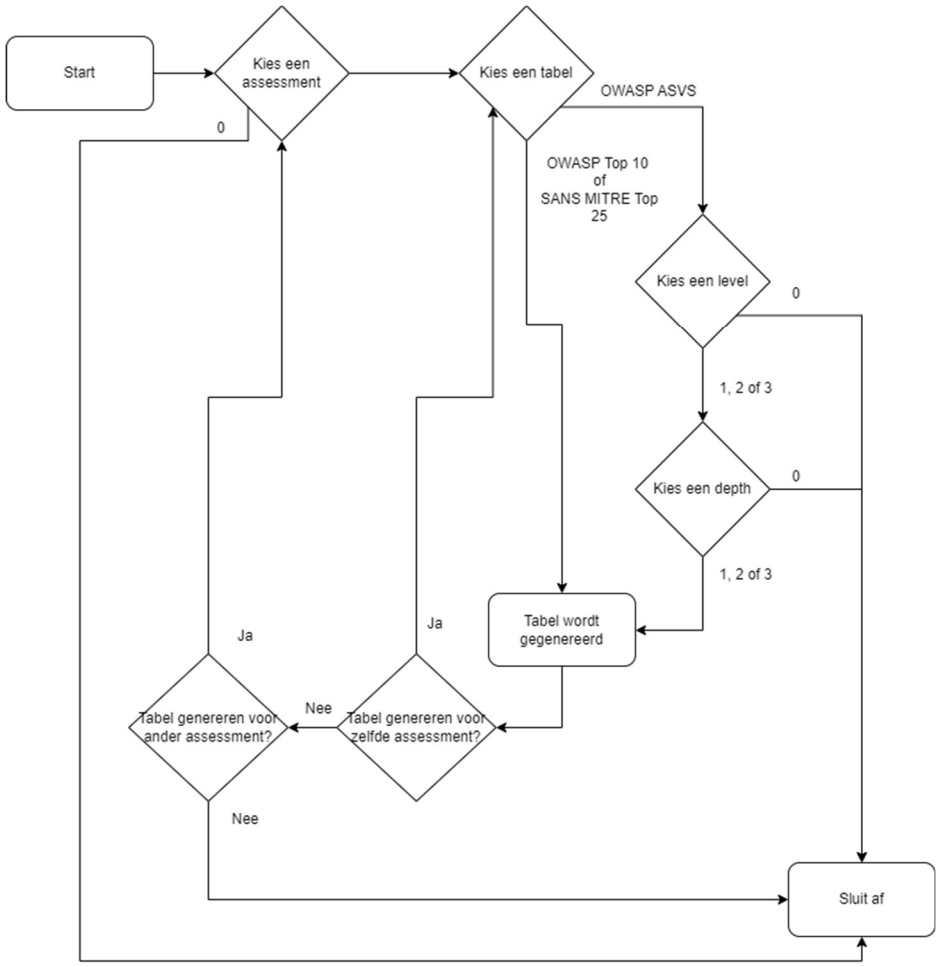
Indien de desbetreffende requirement niet toepasbaar is voor level 1, maar wel voor level 2, zal er in de kolom van level 1 “N/A” komen te staan. Dit zal overal het geval zijn als er geen CWE gekoppeld is aan de requirement. Dit kan gaan over documentatie, wat door Toreon niet zal nagekeken worden.

Door deze modulariteit kan Toreon ook aan de klant aantonen dat er vb. maar x aantal CWE’s opgelost moeten worden om het volgende niveau te halen, waardoor ze misschien nog een extra project kunnen verkopen. Ook kunnen ze extra geld vragen om deze tabellen te genereren.

1.3.11 Coderen van een programma met een user interface (UI)

Nu dat de functies voor het genereren van de tabellen klaar zijn, heb ik ook nog een kleine user interface geschreven voor de CLI (Command Line Interface). Dit maakt het makkelijker voor mensen die minder technisch zijn om de tabellen te genereren. Dit programma is in samenwerking met Matthias Minschart geschreven, aangezien er uit Reporter ook assessments opgehaald moeten worden.

De workflow van het programma is als volgt:



Figuur 13: Flowchart programma om tabel te genereren

1.3.12 Coderen van template overzicht

Samen met Matthias Minschart werd er een programma geschreven om een overview te genereren van alle templates binnen SecurityReporter. Het is zo gecodeerd, dat er bepaalde delen kunnen gevraagd worden om te genereren. Een UI werd niet geschreven, aangezien dit geen prioriteit was. Dit rapport bevestigde nogmaals dat ~68% van de templates in Reporter geen CWE bevatten.

De verschillende overzichten die kunnen opgevraagd worden zijn:

1. Een tabel met een overzicht van hoeveel templates een x aantal CWE's hebben toegewezen.

#CWEs per finding	amount
0 CWE	213
1 CWE	90
2 CWE	9
3 CWE	2

Figuur 14: Tabel met een overzicht van het aantal templates met x aantal CWE's

2. Een lijst met templates, gesorteerd op basis van het aantal toegewezen CWE's.

0 CWE

- test caution
- Cross-origin Resource Sharing: arbitrary origin trusted
- Known Vulnerable or Outdated Package
- Sensitive data stored in browser's local storage
- HSTS Missing From HTTPS Server
- Leaked Credentials
- Insecure Direct Object Reference
- SMBv1 enabled
- Unencrypted Telnet Server
- Email Bombing using the password forgotten functionality
- SSH Weak Algorithms Enabled
- Clickjacking
- Unrestricted and unauthenticated access to an NFS share
- SMB Signing Is Enabled But Not Required
- LLMNR/NBT-NS/mDNS Poisoning leads to domain credentials

Figuur 15: Deel van de lijst met templates zonde CWE

3. Een tabel met een overzicht van welk type van CWE's toegewezen zijn.

Type of CWE	amount
Category	6
Pillar	14
Class	31
Base	48
Variant	15

Figuur 16: Tabel met een overzicht van het aantal CWE's per type. (Totaal over alle templates)

4. Een lijst met templates, gesorteerd op basis van het type van de toegewezen CWE's.

Category

- TLS/SSL Settings are not secure
- Multiple Simultaneous User Sessions
- Outdated Version of JQuery
- HTTP TRACE Method is Enabled
- Multiple Simultaneous Front-end User Sessions
- _OBSOLETE_Outdated and Unsupported Web Server Software

Pillar

- Cookie Without HTTPOnly Flag Set
- Lack of Content Security Policy
- Cookie With Secure Flag Missing
- Strict Transport Security Policy Not Enforced
- Default Vendor Credentials In Use
- Misconfigured X-XSS-Protection Header
- Lack of Network Access Level Controls (802.1x)

Figuur 17: Lijst met templates die een CWE van het type Cathegory of Pillar bevatten

1.3.13 Feedback van Steven Wierckx

Nadat alles werkende was, werd er een meeting ingepland met Steven Wierckx, expert consultant, een heel belangrijk figuur binnen Toreon (alsook één van mijn stagementoren) met veel ervaring.

Na het voorstellen van alle oplossingen, was Steven tevreden met het resultaat.
De desbetreffende oplossingen zijn:

- De UI voor het genereren van tabellen
- Het programma om caution tags toe te voegen aan de templates
- Het programma om een overview te genereren van de templates

Hij stelde nog kleine verbeterpuntjes voor, maar over het algemeen was het goed. Graag zou hij nog een keer samen zitten met de mensen van sales om te kijken of er van daaruit nog extra input kan komen.

1.3.14 Code aanpassen na code review

Nadat Robbe alle code tot hier toe nagekeken had, waren er nog wat verbeterpunten. Hij was aangenaam verrast van de kwaliteit en vertelde dat hij niet verwacht had dat hij op zo kleine details zou moeten letten. Dit waren vooral algemene regels binnen Python, zoals: namen en locatie van files / variabelen, algemene functies verplaatsen naar een util file, etc.

Na de code review vroeg hij ook om custom exceptions te schrijven voor errors op te vangen aangezien ik tot hier toe enkel een print() of exit() had gebruikt met de error message. Ook vroeg hij om testen te schrijven om te kijken of deze exceptions aangeroepen konden worden en werken zoals verwacht.

1.3.15 Custom exceptions maken in Python

Zelf had ik nog geen ervaring met exceptions in Python. Exceptions zijn op maat gemaakte error messages die de gebruiker duidelijk moeten maken wat er mis is gelopen. Na wat opzoekwerk snapte ik nog niet 100% hoe ik dit het best kon aanpakken, dus heb ik Robbe om advies gevraagd. Hij heeft dit samen met mij bekeken, waarna ik alles snapte en in orde heb gebracht. Hier hebben we onder andere exceptions geschreven voor een ongeldige input, als een item niet gevonden werd, etc.

```python
class InvalidArgumentException(Exception):
    def __init__(self, arg_name, arg_type):
        super().__init__(f'Wrong input type: {arg_name} must be {arg_type}')


class NotFoundException(Exception):
    def __init__(self, type, value):
        super().__init__(f'No {type} found for {value}')


class CWENotFoundException(NotFoundException):
    def __init__(self, cwe_id):
        super().__init__('CWE', cwe_id)


class ViewNotFoundException(NotFoundException):
    def __init__(self, view_id):
        super().__init__('View', view_id)


class CategoryNotFoundException(NotFoundException):
    def __init__(self, cat_id):
        super().__init__('Categroy', cat_id)


class CVENotFoundException(NotFoundException):
    def __init__(self, cve_name):
        super().__init__('CVE', cve_name)
```

Figuur 18: Screenshot van de code die de zelf geschreven exceptrions test

1.3.16 Unit tests schrijven voor de exceptions

Na de exceptions gecodeerd te hebben, ben ik begonnen aan het aanpassen van de bestaande testen aangezien de errors nu op een andere manier behandeld worden. Verder heb ik nieuwe test-files aangemaakt om alle edge-cases en exceptions te testen om te kijken of deze op een correcte manier worden opgevangen. Dit ging redelijk vlot.

```
# --Testing Exceptions-- #
with self.assertRaises(InvalidInputException):
    helper.get_category_by_id(-1)

with self.assertRaises(InvalidInputException):
    helper.get_category_by_id('123')

with self.assertRaises(CategoryNotFoundException):
    helper.get_category_by_id(9999)
```

Figuur 19: Screenshot van de unit-tests die de zelf geschreven exceptions testen

1.3.17 Documenteren & schrijven van de README

Na het schijven van alle testen, heb ik alle code nog eens kritisch overlopen en commentaar toegevoegd / aangepast waar nodig. Ook werd er samen met Matthias de README geschreven. Samen hebben we beslist over de structuur. De inhoud werd geschreven door diegene die het desbetreffende deel ook effectief gecodeerd heeft.

De REAMDE ziet er als volgt uit:

Introduction

This GitHub Repository is part of the internship of Bram Bleux & Matthias Minschart: 27 feb 2023 - 26 may 2023. This README gives insight into the functionalities that were developed for both the CWE and SecurityReporter Library.

Setup

.env.sample

Do not forget to change the name of the file to ".env"

Info
A .env file is a configuration file used to store environment variables for an application. Environment variables contain information such as API keys, database details, and other settings. To set up a .env file, create a new file with the name .env in the root directory of your project and define environment variables using the format NAME=VALUE.

Variable Name	Description
`SANS_TOP_25_EDITION`	The year of the latest SANS MITRE Top 25 edition. (Default: 2022)
`OWASP_TOP_10_EDITION`	The year of the latest OWASP TOP 10 edition. (Default: 2021)
`OWASP_ASVS_CVE_LINK`	The link to the RAW csv file of the latest OWASP ASVS edition. (Default: link to version 4.0.3)
`REPORTER_URL`	The link to the SecurityReporter instance. (No Default, must be set by user)
`REPORTER_API_TOKEN`	The API token for the SecurityReporter instance. (No Default, must be set by user)

Commands to execute scripts

Naam	Commando
`import_templates.py`	`python -m scripts.import_templates <Path to template file>`
`add_caution.py`	`python -m scripts.add_caution`
`compliance_table.py`	`python -m scripts.compliance_table`
`template_report.py`	`python -m scripts.template_report`

More info Scripts

Name	Description
`import_templates.py`	This script uses a ,json file to import custom finding templates in SecurityReporter. The path of the ,json file must be given as an argument when executing the script.
`add_caution.py`	This script adds caution tags to custom finding templates in SecurityReporter. This is done base on the amount of CWE's assigned to the template (0, 1, 1+). These are defined in /scripts/config/caution_tags.py
`compliance_table.py`	This script is a UI to generate markdown tables of compliance for different security standards based on an assessment in SecurityReporter (The CWE numbers assigned to the findings).
`template_report.py`	This script will generate a markdown overview of the state of custom finding templates in SecurityReporter.

More info Reporter library

Name	Description
`exeptions.py`	File with custom exceptions.
`/tests/template_helper.py`	Unit tests for template_helper.py

More info CWE library

Name	Description
`category.py`	Category Object
`compliance_tables.py`	Contains functions to generate the markdown tables for the following security standards: - OWASP TOP 10 - SANS MITRE Top 25 - OWASP ASVS
`cve.py`	CVE Object
`Cwe_helper.py`	Helper with functions for CWE framework.
`Cwe.py`	CWE Object
`Exeptions.py`	File with custom exceptions.
`View.py`	View Object
`/tests/test_compliance_tables.py`	Unit tests for compliance_tables.py
`/tests/test_cwe_helper.py`	Unit tests for cwe_helper.py

More info misc files

Name	Description
`util.py`	File with functions used is both libraries.
`tests/util.py`	Unit tests for util.py

Figuur 20: Screenshot van de README voor opdracht 1

1.4 De problemen

Hieronder zullen alle problemen meer in detail beschreven worden die ik ben tegen gekomen tijdens deze opdracht.

1.4.1 Python library voor CWE

Na onderzoek naar een library voor het CWE framework is gebleken dat er geen werkende / up-to-date versie beschikbaar is. Daarom is er besloten om zelf een library te schrijven met de functionaliteiten die nodig zijn voor de rest van het project.

1.4.2 Structuur van CWE

Tijdens het coderen van de Python library was er te merken hoe verweven het CWE framework nu precies is. Er zijn CWE's, Views, Categorieën, etc. Maar een View is ook weer een CWE op zich, wat het onderscheid maken nogmaals vermoeilijkt.

Ook de relaties binnen het CWE framework zijn verwarrend. Ze werken met "ChildOf", "ParentOf", "PeerOf", "StartsWith", "CanPrecede", etc. Deze naamgeving is moeilijk te lezen (naar mijn mening). In de library heb ik dit opgelost door per CWE een lijst met "Parents", "Children" en "Peers" te maken. Dit maakt het, naar mijn mening, wat overzichtelijker.

1.4.3 Geen "ParentOf" relatie

In het XML document van MITRE is alleen de "ChildOf" relatie terug te vinden. De "ParentOf" relatie is terug te vinden op de website zelf, maar niet in het XLM bestand. Ik heb dit opgelost door een voorlopige mapping te maken en deze later te gebruiken om de lijst met "Parents" aan te vullen.

```
<Related_Weaknesses>
<Related_Weakness Nature="ChildOf" CWE_ID="770" View_ID="1000" Ordinal="Primary"/>
<Related_Weakness Nature="PeerOf" CWE_ID="789" View_ID="1000" Ordinal="Primary"/>
<Related_Weakness Nature="CanPrecede" CWE_ID="476" View_ID="1000"/>
</Related_Weaknesses>
```

Figuur 21: Screenshot van de gerelateerde CWE's en hun relatie

1.4.4 XML namespace

Bij het gebruik van het XML document om de python library te schrijven, ondervond ik dat elke waarde die werd opgevraagd begon met een bepaalde string. Na wat onderzoek vond ik dat dit veroorzaakt werd door het "xmlns" argument in het XML bestand. Zonder hier verder bij stil te staan verwijderde ik dit argument en codeerde ik verder.

```
<?xml version="1.0" encoding="UTF-8"?><Weakness_Catalog Name="CWE" Version="4.11" Date="2023-04-27" xmlns="http://cwe.mitre.org/cwe-6" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://cwe.mitre.org/cwe-6 http://cwe.mitre.org/data/xsd/cwe_schema_v6.10.xsd" xmlns:xhtml="http://www.w3.org/1999/xhtml">
```

Figuur 22: Screenhot van de XML namespace die verwijderd moest worden

Later in het coderen van de library zorgde ik er voor dat wanneer de library voor het eerst wordt ingeladen, het bestand automatisch gedownload wordt. Wanneer dit geïmplementeerd was, merkte ik dat niets meer wou werken. Na de error bekeken te hebben, merkte ik dat de string terug aanwezig was voor elke waarde die ik opvroeg.

Dit heb ik opgelost door de volgende stappen te automatiseren:

1. Laatste versie van het XML bestand wordt gedownload.
2. Het bestand wordt ingelezen
3. De xmlns wordt door een replace functie verwijderd uit het document
4. Het bestand zonder xmlns wordt geschreven naar "data.xml"

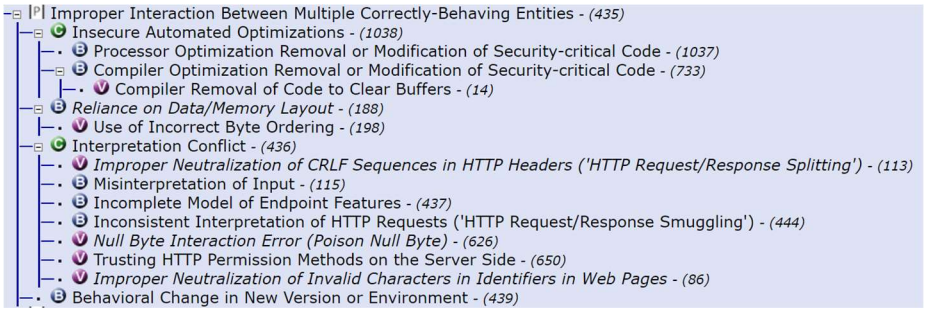
```
# --Get Latest XML if not present-- #
if not os.path.exists('data.xml'):
    # Download the ZIP file
    wget.download('https://cwe.mitre.org/data/xml/cwec_latest.xml.zip')

    # Wait until file is downloaded
    filename = 'cwec_latest.xml.zip'
    while not os.path.exists(filename):
        time.sleep(.5)

    # Extract XML file, remove the xmlns and write to new XML file
    with zipfile.ZipFile(filename) as zf:
        xml_filename = zf.filelist[0].filename
        zf.extract(xml_filename)
        with open(xml_filename, 'r+', encoding='utf-8') as original_file:
            data = original_file.readlines()
            data[0] = data[0].replace(' xmlns="http://cwe.mitre.org/cwe-6" ', ' ')
            with open('data.xml', 'w', encoding='utf-8') as data_file:
                data_file.writelines(data)

    # Clean up files
    os.remove(xml_filename)
    os.remove('cwec_latest.xml.zip')
```

Figuur 23: Screenshot van de code die ervoor zorgt dat de laatste versie van het CWE-framework gebruikt wordt

1.4.5 Zoeken naar de juiste CWE

Aangezien het CWE framework niet werkt met een mooie boomstructuur kwam er ter discussie hoe "diep" er gezocht moet worden naar een CWE. Het eerste idee is dat er naar alle children, children van children etc. wordt gekeken om te zoeken naar een compatibele CWE. Later bleek dit onmogelijk te zijn gezien de verweven structuur van het framework. Mijn voorstel was om alleen naar de directe children te kijken. Dit was mogelijk.

Later veranderde de opdracht, waardoor er naar standaarden werd gekeken. Ook hier kwam weer dezelfde vraag naar boven. Na wat overleg is er besloten om zo veel mogelijk CWE's aan een template toe te voegen en alleen te kijken naar de CWE's die exact in de standaarden voor komen. Dit kwam verder uit het feit dat de mensen / organisaties achter de standaarden "het wel beter zullen weten dan wij".



Figuur 24: Voorbeeld van de verbonden CWE's van CWE-435

1.4.6 Filteren van verwijderde records in OWASP ASVS

In het CSV bestand van de OWASP ASVS[10] staan [DELETED] records. Mijn eerste inzicht is dat deze records geen CWE toegekend hadden. Dit zag ik als mijn kans om deze makkelijk uit de lijst te filteren.

chapter_id	chapter_name	section_id	section_name	req_id	req_description
V1	Architecture, Design and Threat Modeling	V1.1	Secure Software Development Lifecycle	V1.1.1	Verify the use of a secure software development lifecycle that addresses security
V1	Architecture, Design and Threat Modeling	V1.1	Secure Software Development Lifecycle	V1.1.2	Verify the use of threat modeling for every design change or sprint planning to ic
V1	Architecture, Design and Threat Modeling	V1.1	Secure Software Development Lifecycle	V1.1.3	Verify that all user stories and features contain functional security constraints, suc
V1	Architecture, Design and Threat Modeling	V1.1	Secure Software Development Lifecycle	V1.1.4	Verify documentation and justification of all the application's trust boundaries, cc
V1	Architecture, Design and Threat Modeling	V1.1	Secure Software Development Lifecycle	V1.1.5	Verify definition and security analysis of the application's high-level architecture a
V1	Architecture, Design and Threat Modeling	V1.1	Secure Software Development Lifecycle	V1.1.6	Verify implementation of centralized, simple (economy of design), vetted, secure,
V1	Architecture, Design and Threat Modeling	V1.1	Secure Software Development Lifecycle	V1.1.7	Verify availability of a secure coding checklist, security requirements, guideline, o
V1	Architecture, Design and Threat Modeling	V1.2	Authentication Architecture	V1.2.1	Verify the use of unique or special low-privilege operating system accounts for al
V1	Architecture, Design and Threat Modeling	V1.2	Authentication Architecture	V1.2.2	Verify that communications between application components, including APIs, mic
V1	Architecture, Design and Threat Modeling	V1.2	Authentication Architecture	V1.2.3	Verify that the application uses a single vetted authentication mechanism that is l
V1	Architecture, Design and Threat Modeling	V1.2	Authentication Architecture	V1.2.4	Verify that all authentication pathways and identity management APIs implement
V1	Architecture, Design and Threat Modeling	V1.4	Access Control Architecture	V1.4.1	Verify that trusted enforcement points, such as access control gateways, servers, a
V1	Architecture, Design and Threat Modeling	V1.4	Access Control Architecture	V1.4.2	[DELETED, NOT ACTIONABLE]
V1	Architecture, Design and Threat Modeling	V1.4	Access Control Architecture	V1.4.3	[DELETED, DUPLICATE OF 4.1.3]

Figuur 25: Voorbeeld van een verwijderde entry in de OWASP ASVS

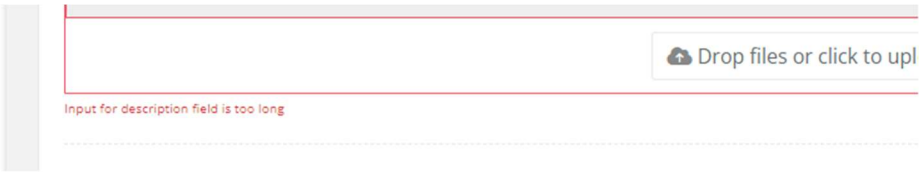
Later, na wat meer onderzoek bleek dat dit geen goede manier was om deze records uit de lijst te filteren. Er zijn namelijk records die te maken hebben met documentatie over de applicatie. Over documentatie bestaan echter geen CWE's. Met dit besef ben ik verder beginnen kijken naar de tabel die ik op dat moment kon genereren en besefte dat mijn oplossing hierboven, helemaal geen oplossing was.

Door te kijken naar de beschrijving van de requirement, was ik in staat dit op te lossen. Elk verwijderde of verouderde record begon namelijk met een "[". Door hier op te filteren is het me effectief gelukt om alleen deze records uit de lijst te halen. Voor de records zonder CWE zal er in de lijst van checks per level "N/A" komen te staan, aangezien Toreon hier niet op kan controleren.

[10] https://github.com/OWASP/ASVS/blob/v4.0.3/4.0/docs_en/OWASP%20Application%20Security%20Verification%20Standard%204.0.3-en.csv

1.4.7 Te lange tabel voor Reporter

Bij het genereren van de tabel voor de OWASP ASVS liepen we tegen het probleem dat de tabel niet helemaal wou inladen in Reporter. Bij het plakken van de markdown tabel in Reporter werd er vermeld dat er te veel tekst was om in te voegen. De oorzaak van dit was de beschrijving van de requirement. Dit werd snel opgelost door de eerste vijf woorden van de beschrijving, gevolgd door drie puntjes te gebruiken i.p.v. de gehele beschrijving.



Figuur 26: Screenshot van de foutmelding die weergegeven werd in SecurityReporter

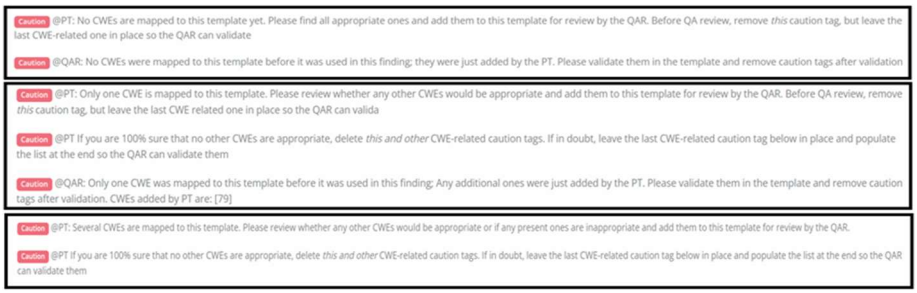
1.5 Het resultaat

Van de oorspronkelijke opdracht bleef er uiteindelijk weinig over. Door onderzoek bleek deze niet realistisch te zijn. Na vele veranderingen werden volgende zaken gerealiseerd:

1.5.1 Programma Caution Tags

Een programma werd geschreven voor het toevoegen van Caution tags aan templates van SecurityReporter op basis van het aantal reeds toegevoegde CWE's.

Dit programma werd in samenwerking met mijn mede student / stagiair Matthias Minschart geschreven. Matthias stond in voor het ophalen en aanpassen van de templates. Ikzelf verwerkte de opgehaalde data en gaf door wat er juist aangepast moest worden.



Figuur 27: Voorbeeld van toegevoegde caution tags

Voor Toreon is dit programma een hulpmiddel dat de kwaliteit van gebruikte templates op termijn zal doen stijgen. Er wordt duidelijk gemaakt aan de tester dat de template in kwestie nog niet volledig is en moet aangepast worden met een of meerdere CWE nummers.

Doordat de kwaliteit van de templates zal toenemen, zorgt dit op zijn beurt dat de kwaliteit van de rapporten van Toreon zullen stijgen.

1.5.2 Programma Compliance Tables

Een programma werd geschreven om, op basis van een assessment in SecurityReporter, tabellen te genereren. Deze tabellen hebben als nut om te kijken of de applicatie al dan niet compliant is met bepaalde security standaarden waaronder:

OWASP Top 10

Category	Pass
A01 - Broken Access Control	✔
A02 - Cryptographic Failures	✔
A03 - Injection	✔
A04 - Insecure Design	✖
A05 - Security Misconfiguration	✔
A06 - Vulnerable and Outdated Components	✔
A07 - Identification and Authentication Failures	✔
A08 - Software and Data Integrity Failures	✔
A09 - Security Logging and Monitoring Failures	✔
A10 - Server-Side Request Forgery (SSRF)	✔

Zie Figuur 10: Voorbeeldtabel

SANS MITRE Top 25

CWE	Name	Pass
CWE-787	Out-of-bounds Write	✔
CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	✔
CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	✔
CWE-20	Improper Input Validation	✔
CWE-125	Out-of-bounds Read	✔
CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	✔
CWE-416	Use After Free	✔
CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	✔
CWE-352	Cross-Site Request Forgery (CSRF)	✔
CWE-434	Unrestricted Upload of File with Dangerous Type	✔
CWE-476	NULL Pointer Dereference	✔
CWE-502	Deserialization of Untrusted Data	✔
CWE-190	Integer Overflow or Wraparound	✔

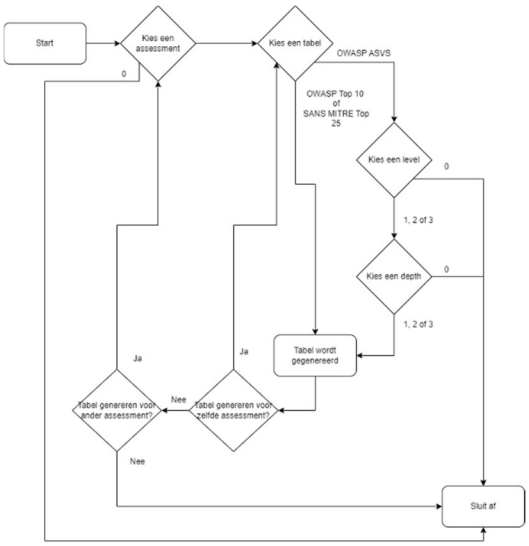
Zie Figuur 11: Voorbeeldtabel

OWASP ASVS

Req ID	Req Desc	CWE	Level 1 Pass	Level 2 Pass	Level 3 Pass
V1.1.1	Verify the use of a ...	N/A	N/A	N/A	N/A
V1.1.2	Verify the use of threat ...	CWE-1053	N/A	✔	✔
V1.1.3	Verify that all user stories ...	CWE-1110	N/A	✔	✔
V1.1.4	Verify documentation and justification of ...	CWE-1059	N/A	✔	✔
V1.1.5	Verify definition and security analysis ...	CWE-1059	N/A	✔	✔
V1.1.6	Verify implementation of centralized, simple ...	CWE-637	N/A	✔	✔
V1.1.7	Verify availability of a secure ...	CWE-637	N/A	✔	✔
V1.2.1	Verify the use of unique ...	CWE-250	N/A	✔	✔
V1.2.2	Verify that communications between application ...	CWE-306	N/A	✔	✔
V1.2.3	Verify that the application uses ...	CWE-306	N/A	✔	✔
V1.2.4	Verify that all authentication pathways ...	CWE-306	N/A	✔	✔
V1.4.1	Verify that trusted enforcement points, ...	CWE-602	N/A	✔	✔
V1.4.4	Verify the application uses a ...	CWE-284	N/A	✔	✔
V1.4.5	Verify that attribute or feature-based ...	CWE-275	N/A	✔	✔

Zie Figuur 12: Voorbeeldtabel

Voor dit programma heb ik een Python library geschreven voor het CWE framework. In combinatie met de verkregen gegevens uit SecurityReporter (geschreven door mede student / stagiair Matthias Minschart) heb ik een programma geschreven dat aan de hand van een command-line user interface, tabellen kan genereren voor drie security standaarden. Het programma werkt als volgt:



Zie figuur 13: Flowchart programma om tabel te genereren

Voor Toreon betekent dit concreet dat ze extra informatie aan een klant kunnen bieden, waardoor de waarde van hun rapporten stijgt. Naar de toekomst toe zullen deze standaarden als maar belangrijker worden. Moest een klant van Toreon hier specifiek naar vragen, kunnen ze een extra toeslag vragen voor dit rapport.

1.5.3 Programma Overzicht Templates

Een programma werd geschreven om een overzicht te bieden over de templates binnen SecurityReporter. Tijdens het onderzoek naar de templates in SecurityReporter bleek dat ongeveer 68% van de templates geen CWE-nummer had toegewezen gekregen.

Samen met mijn mede student / stagiair Matthias Minschart werd er een programma geschreven dat een overzicht zou bieden over de staat van de templates. De gegevens werden opgehaald uit SecurityReporter door Matthias, waarna ik deze data verwerkt heb in de volgende tabellen & lijsten:

1. Een tabel met een overzicht van hoeveel templates een x aantal CWE's hebben toegewezen.

#CWEs per finding	amount
0 CWE	213
1 CWE	90
2 CWE	9
3 CWE	2

Zie Figuur 14: Tabel met een overzicht van het aantal templates met x aantal CWE's

2. Een lijst met templates, gesorteerd op basis van het aantal toegewezen CWE's.

0 CWE

- test caution
- Cross-origin Resource Sharing: arbitrary origin trusted
- Known Vulnerable or Outdated Package
- Sensitive data stored in browser's local storage
- HSTS Missing From HTTPS Server
- Leaked Credentials
- Insecure Direct Object Reference
- SMBv1 enabled
- Unencrypted Telnet Server
- Email Bombing using the password forgotten functionality
- SSH Weak Algorithms Enabled
- Clickjacking
- Unrestricted and unauthenticated access to an NFS share
- SMB Signing Is Enabled But Not Required
- LLMNR/NBT-NS/mDNS Poisoning leads to domain credentials

Zie Figuur 15: Deel van de lijst met templates zonde CWE

3. Een tabel met een overzicht van welk type van CWE's toegewezen zijn.

Type of CWE	amount
Category	6
Pillar	14
Class	31
Base	48
Variant	15

Zie Figuur 16: Tabel met een overzicht van het aantal CWE's per type (Totaal over alle templates)

Category

- TLS/SSL Settings are not secure
- Multiple Simultaneous User Sessions
- Outdated Version of JQuery
- HTTP TRACE Method is Enabled
- Multiple Simultaneous Front-end User Sessions
- _OBSOLETE_Outdated and Unsupported Web Server Software

Pillar

- Cookie Without HTTPOnly Flag Set
- Lack of Content Security Policy
- Cookie With Secure Flag Missing
- Strict Transport Security Policy Not Enforced
- Default Vendor Credentials In Use
- Misconfigured X-XSS-Protection Header
- Lack of Network Access Level Controls (802.1x)

Zie Figuur 17: Lijst met templates die een CWE van het type Cathegory of Pillar bevatten

Voor Toreon betekent dit concreet dat er ten alle tijden een overzicht kan opgevraagd worden over de stand van zaken van de Templates binnen SecurityReporter. Dit zal een hulp zijn in het verbeteren van de templates, wat effect heeft op de kwaliteit van de rapporten van Toreon.

2 Opdracht 2: Burp Suite Extensie

2.1 De oorspronkelijke opdracht

Collega Yael is bezig met haar traineeship binnen Toreon. Nieuwe werknemers bij Toreon zijn ongeveer een zes-tal maanden "trainee". In deze periode kunnen ze inwerken in het bedrijf, alsook een onderwerp onderzoeken als een eindwerk.

Yael wou onder andere de business case bepalen van een Burp Suite extensie. Ze vroeg ons een Burp Suite extensie te schrijven voor het automatiseren van repetitieve taken die uitgevoerd werden door het team van ethische hackers binnen Toreon.

Op deze manier kan er een berekening gemaakt worden om de business case te bepalen van een Burp Suite extensie. Dit door te kijken naar hoe lang het duurt om een extensie te schrijven en onderhouden ten opzichte van de hoeveelheid tijd die er uitgespaard wordt door de extensie.

2.2 De gebruikte technologieën

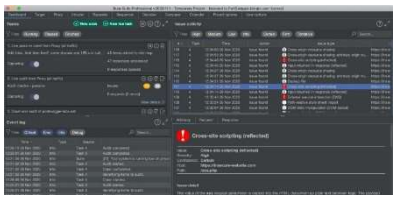
2.2.1 Burp Suite[11]

Figuur 28: Burp Suite logo

Burp Suite is een geavanceerde tool voor webapplicatiebeveiliging die is ontwikkeld door PortSwigger. Het is ontworpen om beveiligingsprofessionals en ontwikkelaars te helpen bij het testen en beveiligen van webapplicaties. Burp Suite biedt een reeks krachtige functies die gebruikers in staat stellen om verschillende soorten webaanvallen uit te voeren en kwetsbaarheden te ontdekken, zoals cross-site scripting, SQL-injectie en directory traversal.

Één van de belangrijkste kenmerken van Burp Suite is het gebruiksgemak. De tool heeft een intuïtieve en gebruiksvriendelijke interface, waardoor het gemakkelijk is om de verschillende functies en modules te begrijpen en te gebruiken. Burp Suite is ook zeer aanpasbaar en biedt tal van opties en instellingen waarmee gebruikers de tool kunnen afstemmen op hun specifieke behoeften en vereisten.

Burp Suite wordt vaak gebruikt door beveiligingsprofessionals en ontwikkelaars om webapplicaties te testen en te beveiligen en wordt beschouwd als één van de meest populaire en effectieve tools in zijn soort. Het biedt een uitgebreide reeks functies en modules waarmee gebruikers verschillende soorten webaanvallen kunnen uitvoeren, kwetsbaarheden kunnen ontdekken en de beveiliging van webapplicaties kunnen verbeteren.



Figuur 29: Screenshot van het Burp Suite dashboard

Binnen Toreon wordt Burp Suite gebruikt door de ethical hackers tijdens een penetratietest van een website of applicatie voor een klant.

[11] https://portswigger.net/

2.2.2 Jython[12]

Met Jython kunnen ontwikkelaars Python-code schrijven met alle voordelen van Java, zoals platformonafhankelijkheid, objectgeoriënteerd programmeren, enz. Jython biedt ook ondersteuning voor Java-bibliotheken en frameworks, zodat ontwikkelaars deze kunnen gebruiken in hun Python-toepassingen.

Figuur 30: Jython logo

Één van de belangrijkste voordelen van Jython is dat het de interoperabiliteit tussen Python en Java verbetert. Dit maakt het gemakkelijker om Python-toepassingen te integreren met bestaande Java-toepassingen en -systemen. Bovendien kunnen ontwikkelaars met Jython, Python en Java in dezelfde toepassing combineren, waardoor de flexibiliteit en veelzijdigheid van de applicatie worden vergroot.

Standaard worden extensies van Burp Suite in Java geschreven. Vandaag de dag wordt Jython meer en meer gebruikt voor het schrijven van Burp Suite extensies, aangezien veel mensen Python prefereren over Java.

Kortom, Jython is een implementatie van Python die op de Java Virtual Machine draait en Python-ontwikkelaars in staat stelt om de voordelen van Java te benutten en hun toepassingen te integreren met bestaande Java-toepassingen en -systemen.

[12] https://www.jython.org/

2.2.3 WSTG[13]

WSTG staat voor "Web Security Testing Guide" en is een open source-project dat is ontwikkeld door de OWASP-gemeenschap. Het biedt een uitgebreide gids voor het testen van webapplicaties op beveiligingskwetsbaarheden.

Figuur 31: OWASP WSTG logo

De WSTG-gids omvat verschillende fasen van het testen van webapplicaties, waaronder informatie verzamelen, identificatie van kwetsbaarheden, penetratietesten en rapportage. Het biedt een gestructureerde en systematische benadering van webapplicatietesten en bevat richtlijnen, technieken en methoden voor het identificeren van kwetsbaarheden en het uitvoeren van testcases.

De gids behandelt een breed scala aan beveiligingsproblemen die kunnen optreden in webapplicaties, zoals cross-site scripting, SQL-injectie, verificatie- en autorisatieproblemen, kwetsbaarheden in sessiebeheer en andere bekende kwetsbaarheden. Het biedt ook richtlijnen voor het identificeren van nieuwe en onbekende kwetsbaarheden.

Één van de belangrijkste voordelen van de WSTG is dat het een uitgebreide en gestructureerde gids is voor het testen van webapplicaties. Het helpt beveiligingsprofessionals om systematisch en grondig webapplicaties te testen en te identificeren waar mogelijks beveiligingsproblemen bestaan. Dit kan leiden tot een verbeterde beveiliging van webapplicaties en een verhoogde veiligheid voor eindgebruikers.

[13] https://owasp.org/www-project-web-security-testing-guide/

2.2.4 Shodan[14]

Figuur 32: Shodan logo

Shodan is een zoekmachine voor apparaten die zijn verbonden met het internet. In tegenstelling tot traditionele zoekmachines die zich richten op het indexeren van webpagina's, richt Shodan zich op het indexeren van informatie over apparaten zoals servers, routers, webcams, industriële controlesystemen en andere apparaten die met het internet zijn verbonden.

Shodan maakt gebruik van verschillende technieken om informatie over deze apparaten te verzamelen, zoals het scannen van openbare IP-adressen en het analyseren van banners en metadata die door de apparaten worden verzonden. De zoekmachine maakt deze informatie vervolgens doorzoekbaar en toegankelijk voor gebruikers.

Met Shodan kunnen gebruikers zoeken naar specifieke apparaten, diensten of protocollen en gedetailleerde informatie verkrijgen, zoals IP-adressen, geolocatie, gebruikte software en configuratiegegevens. Deze informatie kan waardevol zijn voor beveiligingsonderzoekers, systeembeheerders en hackers, omdat het inzicht biedt in mogelijke beveiligingskwetsbaarheden en onbedoeld blootgestelde systemen.

Binnen Toreon wordt Shodan gebruikt voor het vergaren van kennis van webservers, besturingssystemen, etc. Persoonlijk heb ik met Shodan gewerkt om standaard error pagina’s te zoeken voor veelgebruikte webservers.

[14] https://www.shodan.io/

2.3 Het verloop

2.3.1 Kick-off

Tijdens de kick-off meeting lichtte Yeal de opdracht wat meer toe. Yael is op het moment van het schrijven een trainee bij Toreon. Om haar traineeship af te ronden en consultant te worden dient ze onderzoek uit te voeren naar een bepaald onderwerp, waar ze een eindverslag van zal schrijven. Voor haar verslag heeft ze gekozen voor het bepalen van een business case voor het (laten) schrijven van een Burp Suite extensie.

Ze wil een test van het WSTG framework automatiseren door een Burp Suite extensie. Hier zal er bijgehouden worden hoe lang dit duurt en hoeveel tijd dit uiteindelijk bespaard voor de ethical hackers. Zo kan de business case van een Burp Suite extensie bepaald worden.

2.3.2 Onderzoek naar Burp Suite

Allereerst heb ik op de website van Portswigger alle basis labo's gemaakt i.v.m. Burp Suite. Dit om mijn kennis van zaken wat bij te schaven aangezien ik nog niet heel bekend was met de werking van de tool.

2.3.3 Onderzoek naar Burp Suite extensies

Tijdens mijn onderzoek naar Burp Suite extensies merkte ik, dat er al verscheidene extensies beschikbaar zijn met zeer uiteenlopende functies. Vaak worden de extensies geschreven in Java, maar ook meer en meer in Jython. Zelf heb ik er voor gekozen om mijn extensie te schrijven in Jython, aangezien ik meer ervaring heb met coderen in Python dan in Java. Ook moet deze code niet gecompileerd worden, zoals bij Java. Ook Ruby kan gebruikt worden om de extensie te schrijven, maar hier heb ik totaal geen ervaring mee.

2.3.4 Schrijven van een test extensie

Voor mijn eerste simpele extensie volgde ik een tutorial die ik gevonden had en veranderde de functionaliteit. De originele extensie veranderde het woord "cloud" naar een ander woord. Ik heb de extensie herschreven zodat elk input-veld een opvallende achtergrondkleur zou krijgen. Met het idee dat het voor de ethical hackers makkelijker is om de input velden op websites en applicaties terug te vinden.

Google zag er als volgt uit:

Figuur 33: Screenhot van Google.com na het activeren van de Burp Suite extensie

2.3.5 Zoeken naar een goede functionaliteit

Na het schrijven van mijn test extensie ben ik beginnen zoeken naar een functionaliteit voor mijn echte extensie. Zelf vond ik het moeilijk om een functionaliteit te vinden. De meeste onderwerpen van de WSTG leken me te moeilijk om te automatiseren. Na het eens samen bekeken te hebben met Robbe, één van mijn mentoren, kwam hij op het idee om een extensie te schrijven die standaard error pagina's kon linken met de gebruikte webserver.

2.3.6 Schrijven van de Burp Suite extensie

Het schrijven van de extensie heeft me wat werk gekost. Ik ben begonnen met het schrijven van de herkenning van de webpagina's. Wanneer er gesurft wordt naar een pagina op het internet (maakt niet uit welke) wordt de HTML code bekeken en vergeleken met een database aan standaard error pagina's. Voor elk van deze pagina's wordt er een score berekend hoe gelijkaardig deze zijn. Vanaf een score groter dan 80% wordt deze bijgehouden in een andere lijst. Na alle pagina's in de database overlopen te hebben, wordt de naam van de server met de hoogste score (boven 80%) weergegeven. Dit gebeurt allemaal in de achtergrond.

```
1 Error Scraper Loaded
2 This server might run: traefik,
```

Figuur 34: Screenshot van de output wanneer een pagina herkend wordt

Voorlopig worden deze pagina's opgeslagen in een .json file. Deze file wordt door de extensie ingelezen bij het starten van de extensie. Dit maakt het ook mogelijk om de gevonden pagina's te delen met anderen. Op dit punt moeten de standaard pagina's manueel in de .json file worden opgenomen.

2.3.7 Zoeken naar standaard error pagina's – database aanvullen

Nadat de standaard functionaliteit geschreven was, ben ik op zoek gegaan naar standaard error pagina's. Dit kon ik op twee manieren doen: elke webserver lokaal opzetten en zo de errorpagina's bekijken, of zoeken met behulp van Shodan.

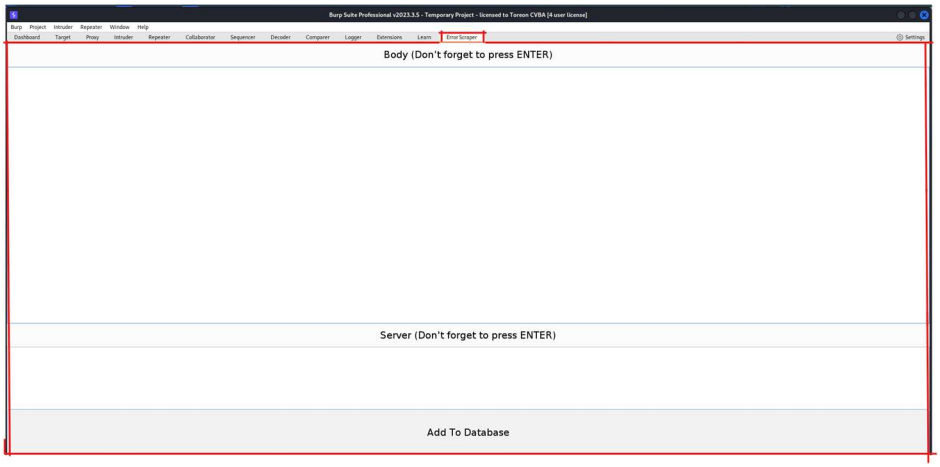
Ik zocht op basis van server en zocht naar websites waar ik merkte dat deze error pagina's niet waren aangepast. Deze nam ik dan op in mijn .json file.

2.3.8 Aanmaken van de UI

Aangezien de extensie op dit punt helemaal in de achtergrond werkt, was het geen meerwaarde om een User Interface te maken voor de extensie. Echter kwam de vraag van Robbe om een makkelijke manier te voorzien om pagina's toe te voegen aan de database. Tot hier toe moest dit manueel gebeuren door de .json file te openen en aan te passen, wat te veel tijd kostte.

Al snel had ik een idee over hoe de User Interface er uit moest zien. Samen met de hulp van Matthias hebben we deze dan gecodeerd. Matthias had al wat ervaring met het schrijven van een User Interface, aangezien zijn extensie al af was. Dit maakte dat ik geen dubbele research moest doen, wat tijd bespaarde.

De User Interface van de extensie zag er als volgt uit:



Figuur 35: Screenshot van de user interface in Burp Suite

Alle velden / knoppen zijn aanwezig, maar hebben nog geen functionaliteit.

2.3.9 Toevoegen van extra functionaliteit

Na het schrijven van de User Interface was het tijd om de functionaliteiten te schrijven. Het bovenste tekstveld werd voorzien om de HTML-inhoud van de standaard webpagina in te plakken.

Het onderste tekstveld is voorzien om aan te geven bij welke webserver de desbetreffende pagina behoord.

Het nadeel van deze tekstvelden is dat er op de ENTER toets gedrukt moet worden. Indien dit niet gebeurt, wordt de pagina niet opgenomen in de database omdat de extensie dit tekstveld nog als leeg wordt aanschouwt.

Als er op de knop “Add To Database” geklikt wordt, zijn er drie mogelijkheden, met hun overeenkomstige meldingen:

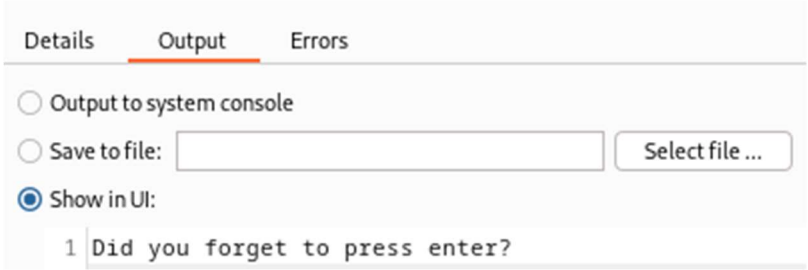
1. De webpagina wordt toegevoegd aan de database.

Figuur 36: Screenshot van de melding wanneer een webpagina wordt toegevoegd aan de database

2. De pagina wordt niet toegevoegd aan de database aangezien deze al in de database aanwezig is.

Figuur 37: Screenshot van de melding wanneer een webpagina al bestaat in de database

3. De pagina wordt niet toegevoegd aan de database aangezien één van de tekstvelden nog gezien wordt als leeg. De gebruiker is vergeten op de ENTER toets de drukken.



Figuur 38: Screenshot van de melding wanneer de gebruiker is vergeten de ENTER toets in te drukken

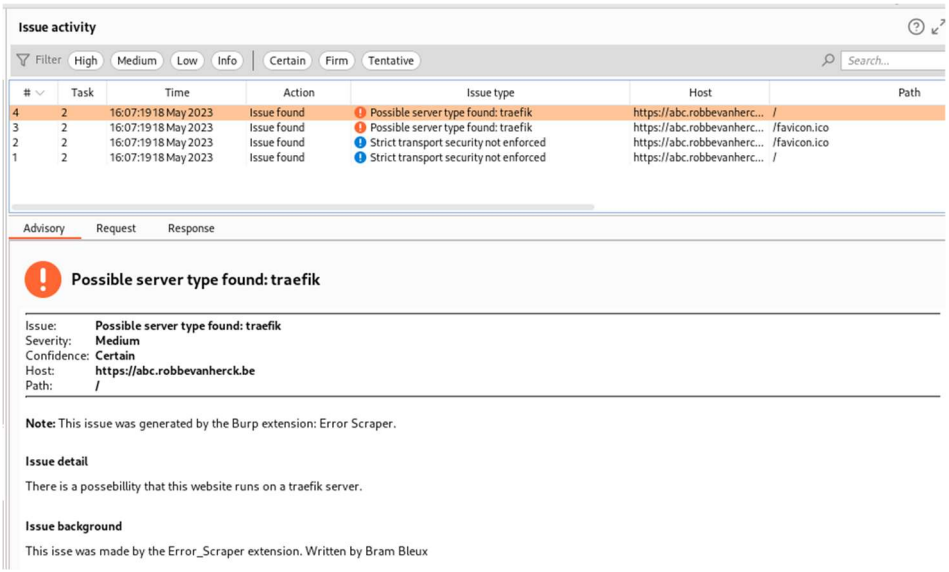
Wanneer de pagina aan de database toegevoegd wordt, zal de .json file automatisch aangepast worden. Ook is het niet nodig om de extensie opnieuw in te laden om de zonet toegevoegde pagina te herkennen.

2.3.10 Aanmaken van issues

Voorlopig werd er alleen in de output van de extensie duidelijk gemaakt als er een match gevonden werd voor een error pagina. Dit was niet gebruiksvriendelijk aangezien dit in een tabblad staat waar normaliter niet naar gekeken wordt.

Robbe vertelde me dat ik er "Issues" voor moest maken. Deze zouden op het dashboard van Burp Suite te zien zijn, wat veel duidelijker en gebruiksvriendelijker is. Een "Issue" is een soort van melding in Burp Suite om zwakheden aan te tonen.

Hij gaf me zijn voorbeeldcode van een test extensie die hij een tijd geleden geschreven had. Met dit ben ik aan de slag gegaan om een Issue aan te maken wanneer er een match wordt gevonden voor een webserver. Na het coderen van de functionaliteit verschijnt volgende melding wanneer er een match gevonden wordt:



Figuur 39: Screenshot van de Issue wordt aangemaakt wanneer er een match wordt gevonden met een pagina in de database

Hier is ook de volledige URL terug te vinden waar de pagina is gevonden. "Traefik" is in dit geval de naam van de webserver.

2.3.11 Herschikken van code

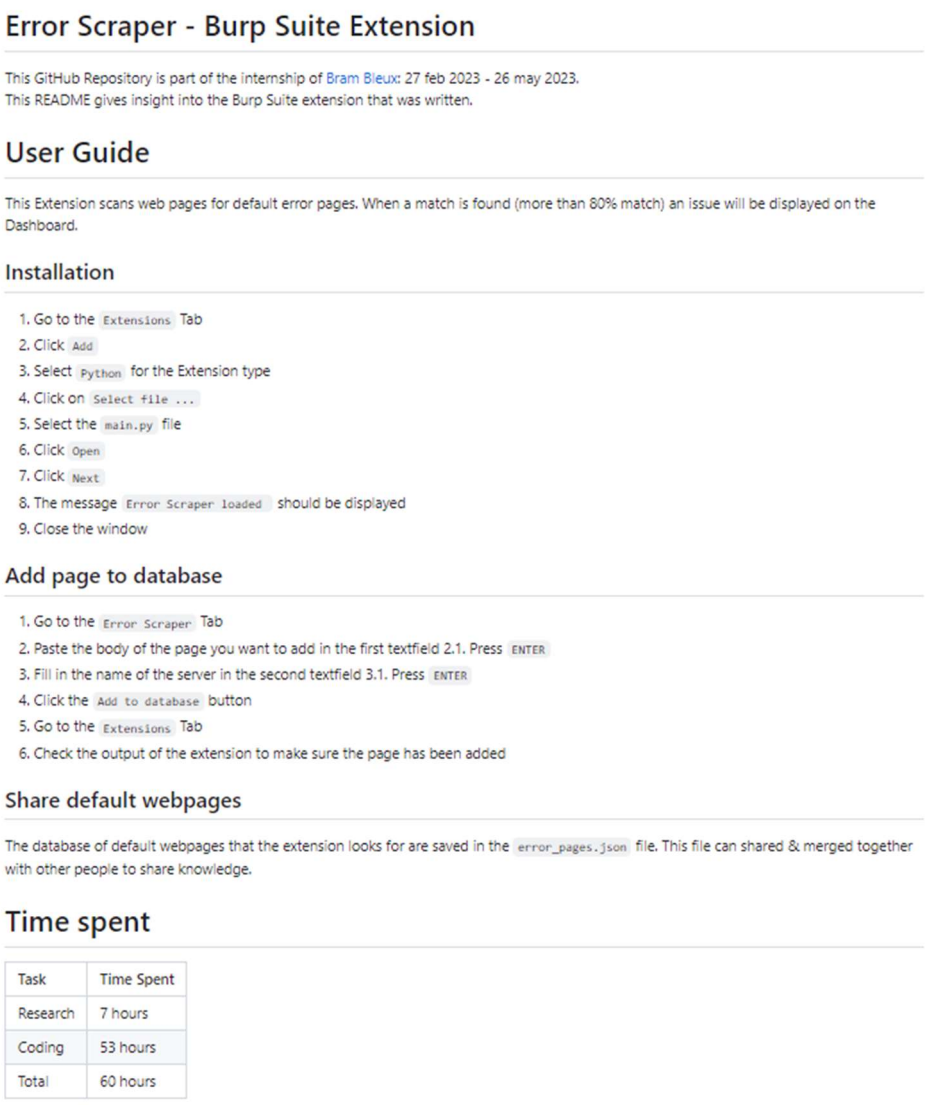
Nadat alle code operationeel was, moest deze opgeschoond worden. Op dit punt stond alles in één .py bestand (de .json file niet meegerekend). Dit moest opgesplitst worden in één file per classe. Hier heb ik best lang over gedaan. Ik had moeite met bepaalde variabelen die over files heen moesten bestaan. Met wat hulp van Robbe is dit uiteindelijk gelukt.

2.3.12 Schrijven van de README

Yael vroeg in het begin van de opdracht om de werkuren bij te houden voor het uitvoeren van deze opdracht. Zo is het mogelijk voor haar om de business case van een Burp Suite extensie te bepalen.

Bij het schrijven van de README voor de extensie heb ik dit, naast een omschrijving en een handleiding, mee opgenomen.

De README ziet er als volgt uit:



Error Scraper - Burp Suite Extension

This GitHub Repository is part of the internship of Bram Bleux: 27 feb 2023 - 26 may 2023.
This README gives insight into the Burp Suite extension that was written.

User Guide

This Extension scans web pages for default error pages. When a match is found (more than 80% match) an issue will be displayed on the Dashboard.

Installation

1. Go to the `Extensions` Tab
2. Click `Add`
3. Select `Python` for the Extension type
4. Click on `Select file ...`
5. Select the `main.py` file
6. Click `Open`
7. Click `Next`
8. The message `Error Scraper loaded` should be displayed
9. Close the window

Add page to database

1. Go to the `Error Scraper` Tab
2. Paste the body of the page you want to add in the first textfield 2.1. Press `ENTER`
3. Fill in the name of the server in the second textfield 3.1. Press `ENTER`
4. Click the `Add to database` button
5. Go to the `Extensions` Tab
6. Check the output of the extension to make sure the page has been added

Share default webpages

The database of default webpages that the extension looks for are saved in the `error_pages.json` file. This file can shared & merged together with other people to share knowledge.

Time spent

Task	Time Spent
Research	7 hours
Coding	53 hours
Total	60 hours

Figuur 40: Screenshot van de README file voor mijn Burp Suite extensie

2.4 De problemen

Hieronder zullen alle problemen meer in detail beschreven worden, die ik ben tegen gekomen tijdens deze opdracht.

2.4.1 Niet weten welke extensie te schrijven

Aangezien er al een groot aanbod bestaat aan Burp Suite extensies, vond ik het moeilijk om een functionaliteit te vinden voor een Burp Suite extensie. Samen met Robbe (mijn stagementor) zijn we door het WSTG framework gegaan en kwamen we uiteindelijk op het idee om een deel passief information-gathering te automatiseren. De extensie die ik zal proberen schrijven zal error pagina's analyseren en vergelijken met een lijst van standaard error pagina's van bepaalde webservers. Als er een match gevonden wordt, zal deze extensie dit aangeven.

2.4.2 Issues willen niet werken

Ondanks de voorbeeldcode die ik van Robbe had gekregen, lukte het me niet om de Issues aan te maken. Ik vroeg Robbe of hij dit even samen wou bekijken. Door hier en daar kleine foutjes op te lossen (waaronder typefouten) werd het probleem opgelost.

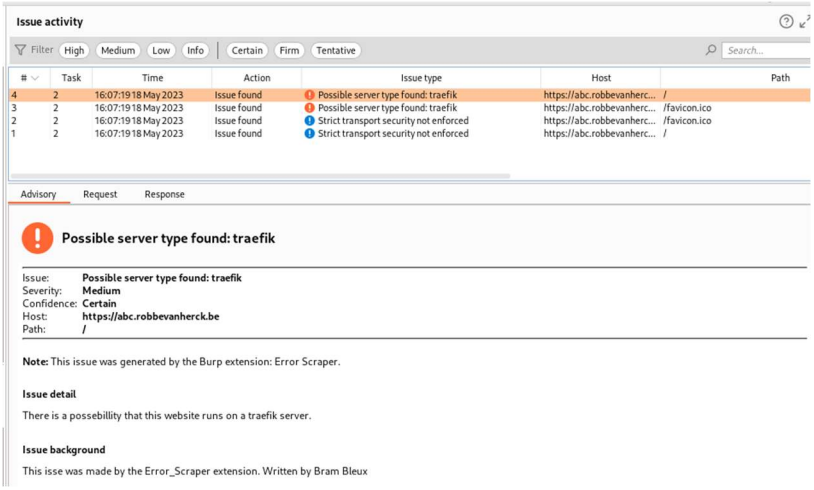
2.4.3 Herschikken van de code

Tijdens het herschikken van de code had ik moeite met de klassen op te delen in verschillende files. Dit omdat er bepaalde variabelen over de verschillende files moesten gedeeld worden. Welke dit waren en hoe ik dit best deed waren me niet duidelijk. Nadat Robbe me wat extra uitleg had gegeven is dit wel gelukt.

2.5 Het resultaat

Er werd een Burp Suite extensie geschreven voor collega / trainee Yael. Om haar traineeship binnen Toreon af te ronden en consultant te worden, dient ze een eindwerk te maken. Voor dit eindwerk wou ze een business case bepalen voor het schrijven van een eigen Burp Suite extensie. Tijdens het maken van mijn Burp Suite extensie heb ik de benodigde tijd voor haar in kaart gebracht (60 uur in totaal).

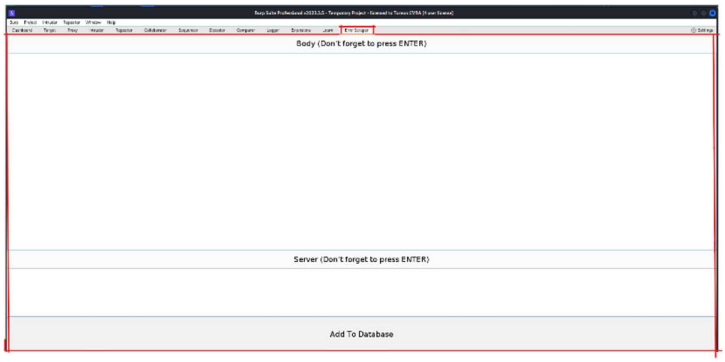
De geschreven Burp Suite extensie zoekt passief naar standaard error pagina's. Als er tijdens het surfen een match is gevonden van meer dan 80%, wordt er een Issue aangemaakt. Deze verschijnt dan op het dashboard van Burp Suite, waar het makkelijk te herkennen is voor de penetratietester. (Zie Figuur 43)



Figuur 41: Screenshot van de Issue wordt aangemaakt wanneer er een match wordt gevonden met een pagina in de database

Verder is het mogelijk om via de user interface van de extensie een nieuwe pagina toe te voegen aan de database. (Zie Figuur 39) Het is niet noodzakelijk om de extensie opnieuw op te starten wanneer er een nieuwe pagina wordt toegevoegd. Ook wordt in de output van de extensie duidelijk gemaakt of de pagina al dan niet is toegevoegd aan de database. (Zie Figuur 40)

Figuur 42: Screenshot van de melding wanneer een webpagina wordt toegevoegd aan de database



Figuur 43: Screenshot van de user interface in Burp Suite

Voor Toreon betekent dit concreet, dat de testers een nieuwe extensie hebben die ze kunnen gebruiken tijdens een test. Ook voor het eindwerk van Yeal werd het nodige onderzoek uitgevoerd.

3 OPDRACHT 3: AUTOMATISATIE ERP & CRM

3.1 De oorspronkelijke opdracht

Als er een nieuw project verkocht wordt bij Toreon, zijn er een aantal stappen die standaard moeten gebeuren. Suleyman, de project manager binnen Toreon, had ons gevraagd of het mogelijk zou zijn om deze taken te automatiseren.

De taken die nu nog manueel gebeuren zijn:

- Het aanmaken van een relatie binnen IonBIZ (het ERP systeem van Toreon) moest deze nog niet bestaan.
- Het aanmaken een Team / Teams kanaal(en) met een bepaalde mappenstructuur om de bestanden van het project op te slaan.

3.2 De gebruikte technologieën

3.2.1 HubSpot[15]

HubSpot is een alles-in-één softwareplatform voor inbound marketing, verkoop en klantenservice.

Figuur 44: HubSpot logo

Met HubSpot kunnen bedrijven hun marketing-, verkoop- en klantenserviceteams beter laten samenwerken en communiceren. Het platform biedt tools voor het creëren en beheren van content, het automatiseren van marketingcampagnes, het bijhouden van leads en het analyseren van marketingprestaties.

HubSpot biedt ook een CRM-systeem (Customer Relationship Management) dat het beheer van klantrelaties eenvoudiger maakt. Het CRM biedt tools voor het beheren van contacten, deals en klantcommunicatie. Bovendien kunnen bedrijven met de ingebouwde rapportage- en analysefuncties van HubSpot de prestaties van hun marketing-, verkoop- en klantenserviceteams volgen en optimaliseren.

HubSpot is ontworpen om bedrijven van elke omvang te helpen bij het aantrekken, converteren en behouden van klanten. Het biedt verschillende abonnementsniveaus en prijspakketten om tegemoet te komen aan de behoeften van kleine bedrijven tot grote ondernemingen. Bovendien biedt HubSpot een uitgebreide online community van gebruikers, evenals training en ondersteuning om klanten te helpen het maximale uit het platform te halen.

Binnen Toreon wordt HubSpot vooral als CRM systeem gebruikt. Alle deals, bedrijven en contacten zijn voor het sales team hier op één plaats beschikbaar.

[15] https://www.hubspot.com/

3.2.2 IonBIZ[16]

IonBIZ is een project management systeem en resource planning systeem dat is ontworpen om bedrijven te helpen bij het plannen en beheren van projecten en resources. Het platform biedt verschillende tools en functies om bedrijven te helpen bij het organiseren van hun projecten, waaronder:

Figuur 45: IonBIZ logo

Project management: IonBIZ biedt een krachtig project management systeem dat bedrijven helpt bij het plannen, organiseren en beheren van projecten van begin tot eind. Het systeem biedt functies zoals Gantt-diagrammen, tijdregistratie en taakbeheer.

Resource planning: Met IonBIZ kunnen bedrijven hun resources plannen en beheren, waaronder het toewijzen van medewerkers aan projecten, het bijhouden van hun beschikbaarheid en het beheren van hun workload.

Financieel beheer: IonBIZ biedt ook tools voor financieel beheer, zoals budgettering en facturering. Het platform kan helpen bij het bijhouden van de kosten van projecten en het genereren van facturen voor klanten.

Rapportage en analyse: IonBIZ biedt ook uitgebreide rapportage- en analysefuncties om bedrijven te helpen bij het bijhouden van hun projectprestaties en het identificeren van kansen voor verbetering.

IonBIZ is ontworpen om bedrijven van elke omvang te helpen bij het plannen en beheren van projecten en resources. Het platform biedt verschillende abonnementsniveaus en prijspakketten om tegemoet te komen aan de behoeften van kleine bedrijven tot grote ondernemingen.

Binnen Toreon wordt IonBIZ gebruikt om de uren van alle werknemers in kaart te brengen. Wekelijks moeten alle werknemers van Toreon de gewerkte uren doorgeven per dag & project. Hier is het nodig om data van de lopende projecten te verwerken om te weten hoe lang de werknemers gespendeerd hebben aan welk project.

[16] https://www.ionprojects.com/

3.2.3 Microsoft Teams[17]

Microsoft Teams is een samenwerkings- en communicatieplatform dat is ontworpen voor teams en organisaties om samen te werken en te communiceren, ongeacht waar ze zich bevinden. Het platform biedt verschillende functies waarmee gebruikers in realtime kunnen communiceren via chat, audio- en videogesprekken en kanalen kunnen creëren voor groepssamenwerking. Teams biedt ook integratie met andere Microsoft-apps zoals Word, Excel en PowerPoint, zodat gebruikers deze apps kunnen openen en bewerken vanuit Teams. Teams heeft ook functies voor videomeetings, scherm delen, etc. Teams is beschikbaar als een cloudgebaseerde dienst en als onderdeel van Microsoft Office 365. Het wordt vaak gebruikt door bedrijven en organisaties van elke omvang om de communicatie en samenwerking tussen teams te verbeteren en de productiviteit te verhogen.

Binnen Toreon is dit, naast e-mail, het algemene communicatiemiddel. Teams maakt het ook mogelijk om files op te slaan voor verschillende projecten, wat het makkelijk maakt om alle files gescheiden te houden en een overzicht te creëren.

Figuur 46: Microsoft Teams logo

3.2.4 Python

Python is een algemene programmeertaal die wordt gebruikt voor een breed scala aan toepassingen, zoals webontwikkeling, gegevensanalyse, machine learning, wetenschappelijke berekeningen, automatisering van taken en nog veel meer.

Voor meer informatie, zie 2.2.4.

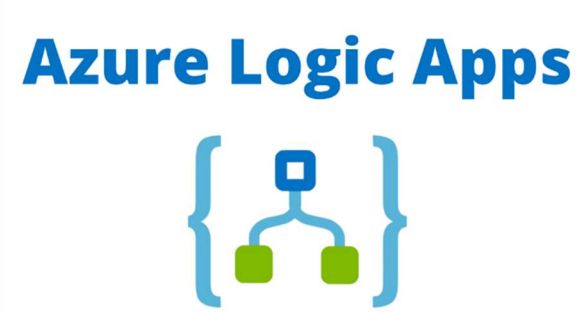
Zie Figuur 4: Python logo

[17] https://www.microsoft.com/en-us/microsoft-teams/group-chat-software/

3.2.5 Azure Logic App[18] [19]

Azure Logic App is een cloudgebaseerde service van Microsoft Azure die wordt gebruikt om bedrijfsworkflows en -processen te automatiseren en te integreren. Het stelt gebruikers in staat om visueel gestuurde workflows te maken door gebruik te maken van een reeks voorgedefinieerde connectoren en logische blokken. Met Logic App kunnen gebruikers eenvoudig verbinding maken met verschillende services, systemen en gegevensbronnen, zowel binnen Azure als extern.

Met Azure Logic App kunnen gebruikers workflows ontwerpen door logische blokken te combineren in een visuele ontwerpinterface. Deze logische blokken omvatten triggers, acties en voorwaardelijke logica. Triggers zijn gebeurtenissen die een workflow activeren, zoals een inkomende e-mail of een timer. Acties zijn taken die worden uitgevoerd zodra de trigger plaatsvindt, zoals het verzenden van een e-mail of het bijwerken van gegevens in een database. Voorwaardelijke logica stelt gebruikers in staat om de flow van de workflow te beheersen op basis van bepaalde voorwaarden.



Figuur 47: Azuze Logic Apps logo

Azure Logic App biedt een uitgebreide set van connectoren die integratie mogelijk maken met verschillende services en platforms, zoals Azure Services, Office 365, Dynamics 365, Salesforce, Twitter, Dropbox en nog veel meer. Deze connectoren vergemakkelijken het uitwisselen van gegevens en de interactie met externe systemen, waardoor gebruikers complexe workflows kunnen bouwen en bedrijfsprocessen kunnen automatiseren.

Het grote voordeel van Azure Logic App is de mogelijkheid om bedrijfsprocessen te automatiseren en te integreren zonder dat er veel code hoeft te worden geschreven. Door gebruik te maken van de visuele ontwerpinterface en de beschikbare connectoren, kunnen gebruikers snel en efficiënt workflows creëren en beheren die verschillende systemen en services met elkaar verbinden.

Azure Logic App maakt deel uit van het Azure-ecosysteem en kan naadloos worden geïntegreerd met andere Azure-services, zoals Azure Functions, Azure Service Bus en Azure Storage. Dit biedt gebruikers flexibiliteit en schaalbaarheid bij het bouwen en implementeren van geavanceerde integratieworkflows.

Binnen Toreon worden Logic Apps nog weinig / niet gebruikt. Zelf heb ik dit gebruikt om een workflow op te stellen voor het automatiseren van het verzenden van e-mails.

[18] https://learn.microsoft.com/en-us/azure/logic-apps/logic-apps-overview
[19] https://www.c-sharpcorner.com/article/what-is-azure-logic-apps/

3.2.6 Azure Automation Account[20] [21]

Een Azure Automation Account is een beheeromgeving in Microsoft Azure die wordt gebruikt om de automatisering van repetitieve taken en processen te vereenvoudigen en te stroomlijnen. Het biedt een reeks functies en services waarmee organisaties workflows kunnen automatiseren, beleidsregels kunnen toepassen en IT-beheerprocessen kunnen vereenvoudigen.

Met een Azure Automation Account kunnen gebruikers runbooks maken en beheren. Runbooks zijn geautomatiseerde workflows die bestaan uit een reeks stappen en acties die moeten worden uitgevoerd. Deze stappen kunnen variëren van eenvoudige taken zoals het verplaatsen van bestanden tot complexe procesautomatiseringen zoals het implementeren van virtuele machines.

Figuur 48: Azure Automation Account logo

Het Automation Account biedt ook integratie met andere Azure-services, waaronder Azure Logic Apps, waardoor gebruikers workflows kunnen creëren die zowel binnen als buiten het Azure-ecosysteem werken. Bovendien biedt het account een breed scala aan vooraf gebouwde runbooks en modules die gebruikers kunnen gebruiken als sjablonen om hun automatiseringsprocessen te versnellen.

Een ander belangrijk aspect van een Azure Automation Account is de mogelijkheid om beheer en monitoring van de geautomatiseerde processen te centraliseren. Gebruikers kunnen inzicht krijgen in de status en uitvoer van de uitgevoerde runbooks, logboeken raadplegen voor foutopsporing en het gebruik van resources volgen om efficiëntie te waarborgen.

Azure Automation Account biedt ook mogelijkheden voor planning en orchestratie van geautomatiseerde taken, waardoor gebruikers workflows kunnen plannen en beheren op basis van specifieke tijdschema's of gebeurtenissen. Hierdoor kunnen bedrijven repetitieve taken automatiseren en hun operationele efficiëntie verbeteren.

Kortom, een Azure Automation Account biedt een geïntegreerde omgeving voor het automatiseren en beheren van herhaalde taken en processen in Azure. Het stelt organisaties in staat om hun operationele workflows te vereenvoudigen, de productiviteit te verhogen en de foutgevoeligheid te verminderen door gebruik te maken van geautomatiseerde runbooks en geïntegreerde beheer- en monitoringtools.

Automation accounts worden gebruikt binnen het Cloud team van Toreon om bepaalde zaken te automatiseren. Zelf werd een Automation gebuikt voor het automatiseren van het verzenden van e-mails binnen deze opdracht.

---

[20] https://learn.microsoft.com/en-us/azure/automation/overview
[21] https://azure.microsoft.com/en-us/pricing/details/automation/

3.3 Het verloop

3.3.1 Kick-off

Tijdens de eerste meeting met Suleyman, de project manager binnen Toreon lichtte hij de opdracht toe. Er moest data opgehaald worden vanuit HubSpot om relaties aan te maken in IonBIZ. Naderhand zal er ook een Team / Teams kanaal aangemaakt worden voor het verkocht project.

3.3.2 Onderzoek naar HubSpot

Bij een eerste kijk naar de HubSpot API lijkt alles goed gedocumenteerd. Ook werd er een Python library gevonden om het coderen gemakkelijker te maken.

3.3.3 Onderzoek naar IonBIZ

De documentatie van IonBIZ is alles behalve uitgebreid. Er wordt gewerkt met scopes om permissies aan te vragen. De API maakt gebruik van Oauth2, waardoor men eerst een token dient aan te vragen, die dan gedurende vijf minuten operationeel is. Ook staat er een limiet op het aantal requests die kunnen gemaakt worden. (90 per minuut)

3.3.4 Ophalen van gegevens uit HubSpot

Al snel was ik begonnen met het schrijven van een programma om data op te halen vanuit HubSpot. Het is mogelijk alle nieuwe deals op te halen, alsook alle randinformatie. Wel werd duidelijk dat niet alle gegevens (waaronder BTW nummer) opgehaald kunnen worden uit HubSpot, wat een probleem kon worden.

3.3.5 Extra opdracht

Tijdens de status-update meeting met Suleyman brachten we hem om de hoogte van wat mogelijk en niet mogelijk was.

Na onderling kwamen we tot de conclusie dat het project realiseerbaar was, maar we ons alleen moesten beperken tot het aanmaken van relaties en geen projecten of quotes.

Ook stelde hij nog een extra opdracht voor, mochten we de tijd en interesse hiervoor hebben. Hij vertelde dat aan het einde van elke week, alle werknemers bij Toreon hun timesheet moeten invullen in IonBIZ. Dit wordt vaak vergeten, waardoor Suleyman deze mensen manueel een bericht / e-mail moet sturen met de vraag of ze dit in orde willen brengen. Hij toonde ons een PowerBI dashboard waar hij kon zien wie nog niet al de werkuren van de afgelopen twee weken had vervolledigd. Hij vroeg ons of het mogelijk was om dit proces te automatiseren.

3.3.6 Onderzoek extra opdracht

Betreft de extra opdracht voor Suleyman, ben ik eerst gaan kijken naar de mogelijkheid om de timesheets via de API van IonBIZ op te vragen. Dit is mogelijk, maar dit is per persoon, per dag. Dit maakt dat het veel data is om te verwerken. Niet onmogelijk, maar niet praktisch. Toen herinnerde ik me dat hij ons een PowerBI dashboard had laten zien, waarop hij kon zien wie hij moest contacteren.

Uit mijn ervaring als jobstudent, wist ik dat Power Automate, (geïntegreerd in Sharepoint / O365) kan samenwerken met PowerBI. Dit wou ik testen, maar aangezien ik geen toegang heb tot PowerBI, was testen onmogelijk.

Om automatisch mails te versturen via een Python programma, botste ik op een library genaamd "win32com.client". Deze library bleek heel krachtig te zijn, aangezien mails verstuurd konden worden zonder in te loggen. Persoonlijk leek met dit niet veilig, dus toonde ik dit aan Robbe, die het wel interessant vond. Jammer genoeg kan dit niet gebruikt worden voor dit project aangezien dit alleen werkt op Microsoft Windows systemen. Om andere libraries te testen, zou ik een service account nodig hebben, aangezien 2-factor authenticatie de automatisatie tegenhoud.

Beide ideeën, een handmatig programma, alsook de PowerBI route heb ik voorgesteld aan Suleyman, met de benodigdheden om dit uit te werken. Waarna hij besloot om het eerst met PowerBI te proberen.

3.3.7 Overschakelen naar Python

Na de mail van Suleyman, die besloot om gebruik te maken van PowerBI, ben ik aan de slag gegaan.

Al snel bleek dat het niet mogelijk was om met Power Automate gegevens op te halen uit een PowerBI dashboard. Daar ging mijn plan…

Na dit te bespreken met Suleyman, werd er besloten om toch over te schakelen op een Python programma. Wel werd er gevraagd om de mails te versturen met een Logic App. Dit is de equivalent van Power Automate in Azure.

Aangezien Toreon geen on-site server heeft, werd er aangeraden om met Azure Functions te werken. Dit omdat het huren van een server / virtuele machine te duur zou zijn.

3.3.8 Ophalen van bestaande relaties

Om dubbele relaties te vermijden wanneer er een nieuwe deal wordt aangemaakt, moest er nagekeken worden of deze reeds bestond. Deze verificatie diende te gebeuren in IonBIZ. Aan de hand van de API werd er een functie geschreven die op basis van bedrijfsnaam zal kijken of de relatie reeds bestaat.

```
def check_relation_exists(self, api_token, company_name):
    """
    Checks if a relations exists for a company in IonBIZ
        scope: r_relation


    Returns [match1, match2, etc.] if relation(s) were found
    Returns None if no relation is found
    """
    header = generate_header(api_token)

    url = self.api_url + '/api/Relations/GetBasicList?' \
                         'pageSize=1006' \
                         'q={}'

    response = requests.get(url=url.format((company_name.replace(' ', '+'))), headers=header).json()

    return response
```

Figuur 49: Functie voor het zoeken van bestaande relaties

3.3.9 Ophalen gewerkte uren

Figuur 50: Functie voor het ophalen van gewerkte uren

Voor de extra opdracht moet er wekelijks gekeken worden hoeveel uren elke werknemer heeft gepresteerd. Aan de hand van de API van IonBIZ worden alle timesheets opgehaald voor de voorbije week. Hierna wordt er per werknemer berekend hoeveel uren ze die week effectief gewerkt hebben. Hier zitten vakantie- en feestdagen niet inbegrepen.

3.3.10 Ophalen van worksheets

Tijdens het coderen van de berekening van het aantal uren kwamen we tot de vaststelling dat sommige werknemers van Toreon niet voltijds werken. Dit maakt dat er minder dan 40 uur per week gewerkt wordt.

Na wat onderzoek bleek dat er in IonBIZ "worksheets" aanwezig zijn. Deze worksheets geven aan hoeveel uur een werknemer per week moet presteren. Hier zitten de nationale feestdagen mee inbegrepen.

Figuur 51: Functie voor het ophalen van een worksheet

Er werd een functie geschreven om deze op te halen op basis van resource_id met een directe link per werknemer.

3.3.11 Ophalen van vakantiedagen / uren

Figuur 52: Functie voor het opalen van vakantiedagen

Het enige wat nu nog ontbrak zijn de vakantiedagen / uren van de werknemers.

In IonBIZ is het ook hier weer mogelijk om "leaves" op te halen op basis van een resource_id.

Er werd een functie geschreven om deze op te halen op basis van resource_id met een directe link per werknemer.

3.3.12 Workflow maken voor verzenden van een e-mail via Azure Logic App

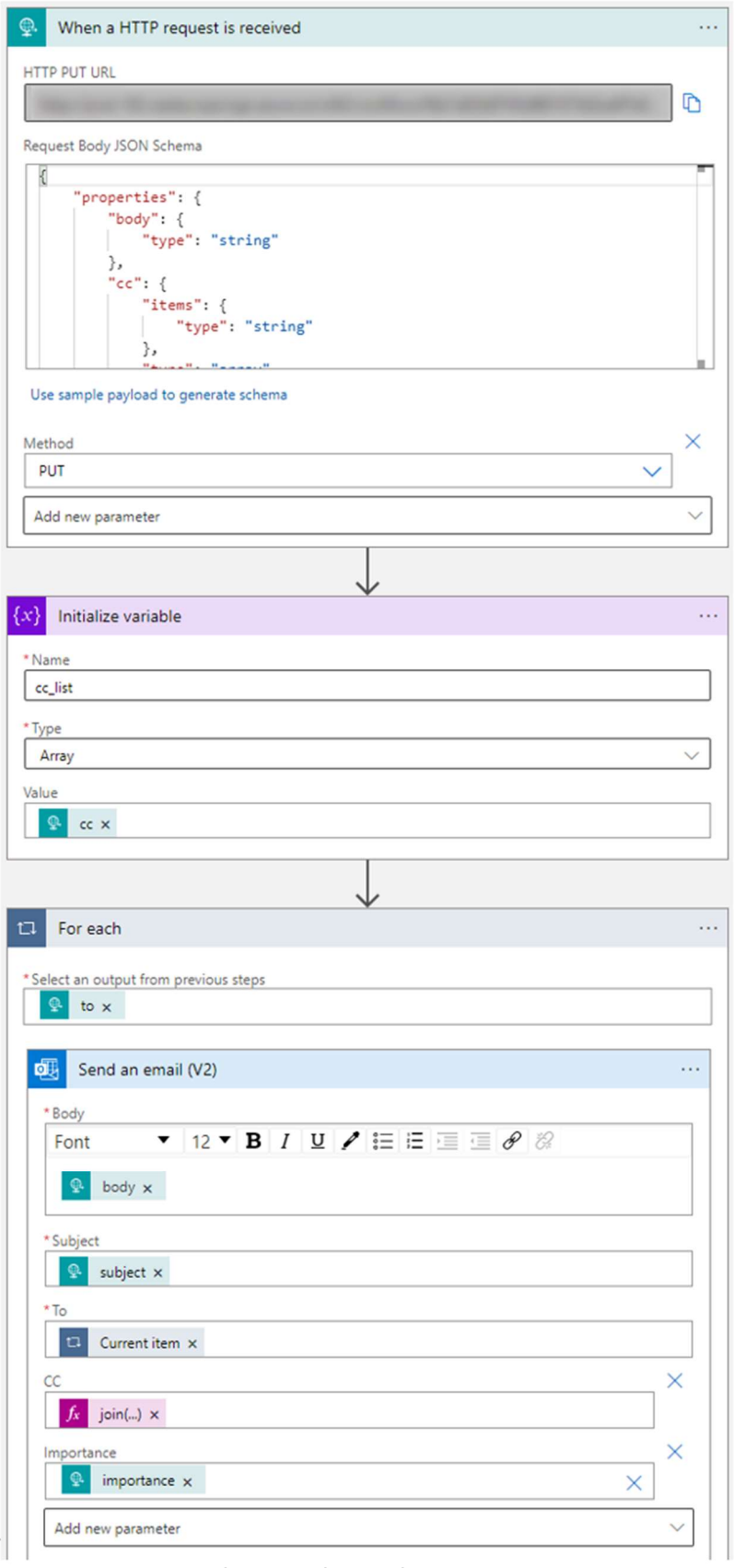
Een Azure Logic App werd aangemaakt om het verzenden van e-mails te automatiseren. Er werd een connectie gemaakt met de Outlook van een automatisatie account binnen Toreon om deze mail te versturen. De Logic app werd zo gemaakt dat er een JSON object via een PUT request moest gestuurd worden met de nodige informatie om de mail te verzenden. De opmaak van het JSON object moet als volgt opgebouwd zijn:

```
{
    "to": [
        "person.1@company.com",
        "person.2@company.com",
    ],
    "cc": [
        "person.3@company.com",
        "person.4@company.com",
    ],
    "subject": "subject of the e-mail",
    "body": "This is the body of the e-mail",
    "importance": "Low / Normal / High"
}
```

Figuur 53: Voorbeeld van een JSON object dat moet meegegeven worden om een mail te versturen

Deze zou twee e-mails verzenden. Een eerste mail naar persoon1 en een tweede mail naar persoon2. In beide mails staan persoon3 en persoon4 in het CC veld van de e-mail.

De Logic App ziet er als volgt uit:



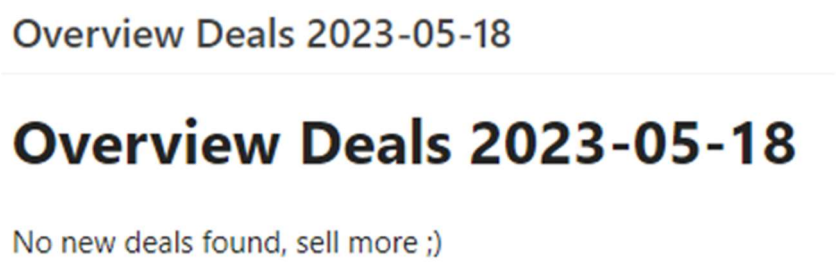
Figuur 54: Screenhot van de geschreven Logic App

3.3.13 Coderen van het programma voor nieuwe deals

Aan de hand van de geschreven functies werd er een programma geschreven dat alle gesloten deals ophaalt uit HubSpot van de laatste 24 uur. Indien de klant nog niet aanwezig is in IonBIZ wordt deze aangemaakt. Op het einde van het programma wordt er een status-update als e-mail verzonden naar Suleyman. In deze mail is terug te vinden wat er gebeurd is tijdens het uitvoeren van het programma. Indien er geen nieuwe deal gesloten is, wordt deze mail nog steeds verstuurd.

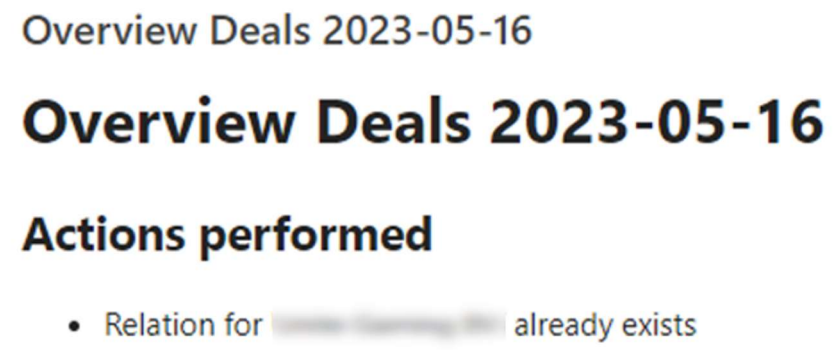
Deze status-update e-mail is als volgt opgebouwd:

1. Geen nieuwe deal



Figuur 55: Screenshot van de mail die verstuurd wordt wanneer er geen nieuwe deal gevonden werd

2. Nieuwe deal met bestaande klant



Figuur 56: Screenshot van de mail die verstuurd wordt wanneer er een deal is gevonden van een bestaande klant

3. Nieuwe deal met niet-bestaande klant

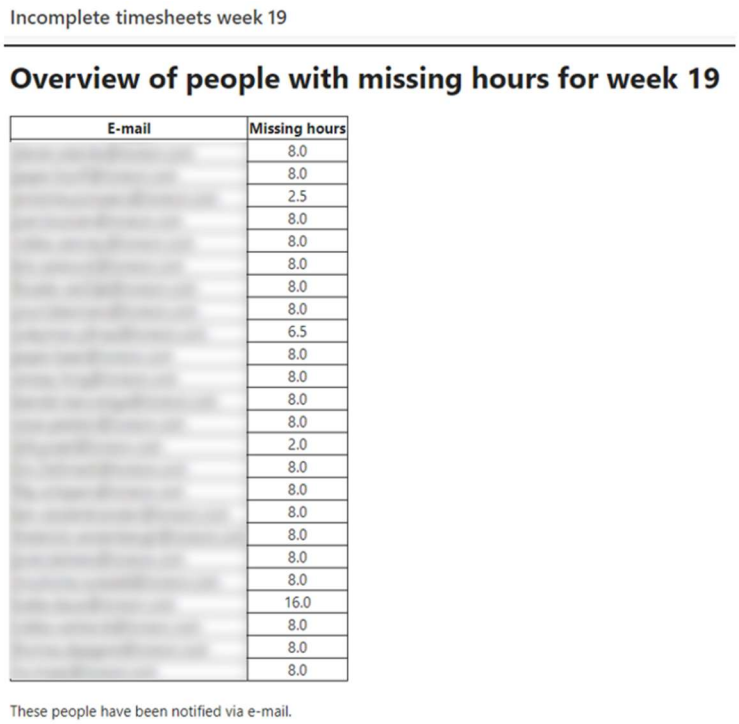
Figuur 57: Screenshot van de mail die verstuurd wordt wanneer er een deal is gevonden van een niet bestaande klant

3.3.14 Coderen van het programma voor timesheets

Aan de hand van de geschreven functies werd er een programma geschreven dat alle timesheets van de afgelopen zeven dagen ophaalt. Er wordt per medewerker nagekeken of er 40 uur gepresteerd is in de afgelopen zeven dagen. Als dit niet het geval is, worden de vakantiedagen en worksheets opgehaald van deze medewerker. Als uit deze berekening nog steeds blijkt dat niet alle uren ingevuld zijn, zal de medewerker in kwestie een e-mail ontvangen met de vraag om zijn of haar timesheets aan te vullen.

Aan het einde van het programma wordt er een e-mail gestuurd naar Suleyman met een overzicht van de mensen die verwittigd werden door het programma, als ook het aantal uren per persoon dat ontbreekt.

De e-mail met het overzicht dat verstuurd wordt naar Suleyman is als volgt opgebouwd:



Incomplete timesheets week 19

Overview of people with missing hours for week 19

E-mail	Missing hours
[redacted]	8.0
[redacted]	8.0
[redacted]	2.5
[redacted]	8.0
[redacted]	8.0
[redacted]	8.0
[redacted]	8.0
[redacted]	8.0
[redacted]	8.0
[redacted]	6.5
[redacted]	8.0
[redacted]	8.0
[redacted]	8.0
[redacted]	8.0
[redacted]	2.0
[redacted]	8.0
[redacted]	8.0
[redacted]	8.0
[redacted]	8.0
[redacted]	8.0
[redacted]	8.0
[redacted]	16.0
[redacted]	8.0
[redacted]	8.0
[redacted]	8.0

These people have been notified via e-mail.

Figuur 58: Screenshot van de mail die verstuurd word als overzicht van de werknemers die niet in orde zijn met hun timesheets

3.3.15 Overschakelen naar Azure Automation Account

Nadat alle code geschreven was, moest dit nog in Azure automatisch uitgevoerd worden. Er werd aangeraden om met Azure Functions te werken. Na veel gesukkel met Azure Functions, slaagde we er nog steeds niet is, het wou niet werken.

Na overleg met Robbe en iemand van het Cloud team binnen Toreon, werd er besloten om over te schakelen naar een Azure Automation account. Hier kunnen “Runbooks” aangemaakt worden om de code uit te voeren. Deze Runbooks kunnen op hun beurt gelinkt worden met een “Schedule”. Dit maakt dat we onze programma’s om de x aantal tijd automatisch kunnen uitvoeren.

3.3.16 Code implementeren in Azure Automation Account

Nadat het Azure Automation account werd aangemaakt, werden er twee Runbooks aangemaakt. Deze staan in voor het uitvoeren van de geschreven programma's. In deze Runbooks werd de code van de programma's geplaatst.

3.3.17 Importeren van Python packages

Ondanks dat de code geïmporteerd werd, konden de programma's nog niet worden uitgevoerd. In een Azure Automation account dienen de gebuikte Python libraries manueel geïmporteerd te worden. Deze moesten eerst gedownload worden van de Python Package Index22. Hiena moesten deze geüpload worden naar het Azure Automation account. Na de nodige libraries geïmporteerd te hebben, werkte de programma's zoals gewenst.

3.3.18 Instellen van Schedules

Voor elke Runbook in Azure werd er een overeenkomstig Schedule aangemaakt.

- Het programma van de deals zal dagelijks om 08:00 uitgevoerd worden
- Het programma van de timesheets zal elke maandag om 07:30 uitgevoerd worden.

3.3.19 Last-minute aanpassingen

Tijdens de laatste week van de stage, vroeg Suleyman nog om wat aanpassingen, waaronder:

- Update-mails verwijderen wanneer er niets gebeurd is
- Aanpassen van het timeframe van de timesheets
- Een onderscheid maken tussen consultants & backoffices met een aparte e-mail
- Externe medewerkers & freelancers niet te controleren

3.3.20 Aanpassen van timeframe

Momenteel kijkt het programma van de timesheets naar de afgelopen week om deze te controleren per persoon. De vraag kwam van Suleyman om dit aan te passen zodat het programma de berekening zou maken voor de afgelopen twee weken. Dit was makkelijker voor hem om in het begin te vergelijken met het Power BI dashboard dat momenteel al aanwezig is. De code werd zo aangepast dat er aan de hand van twee variabelen (dagen & weken) het timeframe van het programma aan te passen.

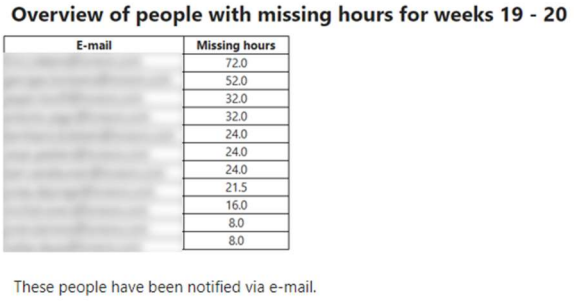
[22] https://pypi.org/

3.3.21 Veranderen van de update e-mails

Momenteel worden alle werknemers door het programma nagekeken. Dit geldt dus ook voor de externe medewerkers waaronder freelancers die voor Toreon werken. De vraag kwam om de externe medewerkers en freelancers die voor Toreon werken niet te controleren op hun timesheets, aangezien deze op basis van facturen werken.

Verder werd er gevraagd om een onderscheid te maken tussen de consultants & backoffices, aangezien hiervoor niet dezelfde persoon verantwoordelijk is.

Deze wijzigingen werden aangebracht en getest op de testomgeving.

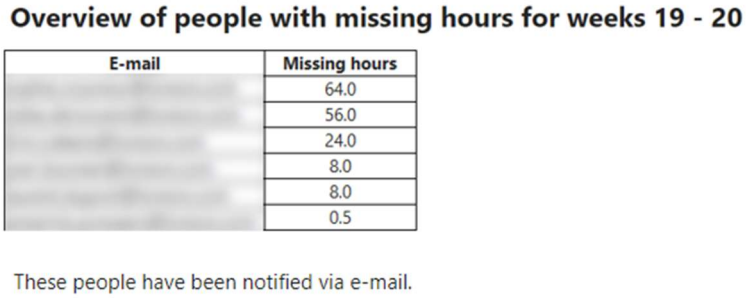


Overview of people with missing hours for weeks 19 - 20

E-mail	Missing hours
[redacted]	72.0
[redacted]	52.0
[redacted]	32.0
[redacted]	32.0
[redacted]	24.0
[redacted]	24.0
[redacted]	24.0
[redacted]	21.5
[redacted]	16.0
[redacted]	8.0
[redacted]	8.0

These people have been notified via e-mail.

Figuur 59: Voorbeeldmail voor consultants



Overview of people with missing hours for weeks 19 - 20

E-mail	Missing hours
[redacted]	64.0
[redacted]	56.0
[redacted]	24.0
[redacted]	8.0
[redacted]	8.0
[redacted]	0.5

These people have been notified via e-mail.

Figuur 60: Voorbeeldmail voor backoffices

3.3.22 Omzetten naar productieomgeving

Na een testrun in de testomgeving werd de code op Azure aangepast en een aanvraag ingediend voor de omzetting naar de productieomgeving. De sleutels voor de productieomgeving werden verkregen en aangepast.

3.4 De problemen

Hieronder zullen alle problemen meer in detail beschreven worden die ik ben tegen gekomen tijdens deze opdracht.

3.4.1 Permissies

Aangezien er gewerkt wordt met "least privilege" botste ik vaak op het tekortkomen van permissies bij het opvragen van data. Hierdoor zouden we tijd verliezen. Dit hebben we opgelost door ons werk te spreiden over de andere projecten tijdens het wachten op het verkrijgen van toegang.

3.4.2 Python library van HubSpot vraagt niet alle gegevens op

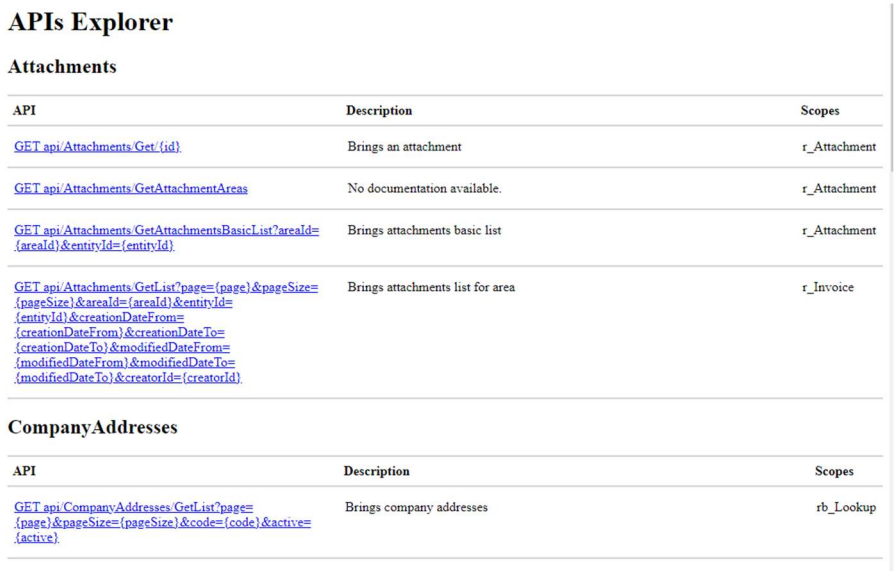
Tijdens het opvragen van gegevens uit HubSpot merkte ik al snel dat ik niet veel details te zien kreeg. Dit was omdat er anders te veel data verstuurd moest worden. Aangezien er geen manier was om meer data op te vragen via de Python library werd er overgeschakeld naar de API. Aan de hand van parameters die meegegeven kunnen worden, was het mogelijk om deze gegevens op te vragen.

3.4.3 Missende gegevens

Ondanks dat er via de API meer info kon verkregen worden, werden niet alle gewenste gegevens opgevraagd. Onder andere BTW nummer en vennootschapsvorm. Dit werd besproken met de klant. Hij stelde voor dat dit alsnog manueel toe te voegen, wat geen probleem was.

3.4.4 Slecht gedocumenteerde API

De API van IonBIZ is slecht gedocumenteerd. Aan de hand van de documentatie hebben we de algemene werking van de API kunnen ontcijferen. Ook bleek dat niet alle functionaliteiten gedocumenteerd zijn. Zo is het bijvoorbeeld mogelijk om een project aan te maken via de API, terwijl daar geen documentatie voor voorzien is.



APIs Explorer

Attachments

API	Description	Scopes
GET api/Attachments/Get/{id}	Brings an attachment	r_Attachment
GET api/Attachments/GetAttachmentAreas	No documentation available.	r_Attachment
GET api/Attachments/GetAttachmentsBasicList?areaId={areaId}&entityId={entityId}	Brings attachments basic list	r_Attachment
GET api/Attachments/GetList?page={page}&pageSize={pageSize}&areaId={areaId}&entityId={entityId}&creationDateFrom={creationDateFrom}&creationDateTo={creationDateTo}&modifiedDateFrom={modifiedDateFrom}&modifiedDateTo={modifiedDateTo}&creatorId={creatorId}	Brings attachments list for area	r_Invoice

CompanyAddresses

API	Description	Scopes
GET api/CompanyAddresses/GetList?page={page}&pageSize={pageSize}&code={code}&active={active}	Brings company addresses	rb_Lookup

Figuur 61: Screenshot van de IonBIZ API documentatie

3.4.5 Relatie aanmaken in IonBIZ (json= i.p.v. data=)

Bij het testen wat er nodig was om een relatie aan te maken in IonBIZ, merkte ik dat vanaf er een nesting in mijn object zat (een object in een object) de API vastliep. Ondanks het zoeken en het samen met Robbe bekeken te hebben konden we de oorzaak niet achterhalen. We gingen er van uit dat het aan de API lag. Later bleek dat ik de verkeerde parameter doorgaf in mijn programma. Ik verzond mijn data via de requests library van Python met de parameter "data=". Het moment dat ik dit veranderde naar "json=" werkte alles zonder problemen.

3.4.6 Dubbele contacten in IonBIZ

Tijdens het testen van het aanmaken van een relatie in IonBIZ werd er opgemerkt dat alle contacten twee maal werden aangemaakt. Na wat debuggen bleek de fout bij HubSpot te liggen. Via de API van HubSpot worden alle relaties tussen een bedrijf en een contact opgehaald (m.a.w. alle contacten verbonden met het desbetreffende bedrijf). Hier bleek dat er per contact twee types van relaties aanwezig zijn, namelijk: 'contact_to_company' en 'contact_to_company_unlabeled'. Dit maakte dat het contact twee keer werd opgenomen.

We merkte op dat de ID van het contact wel hetzelfde was. Dit was onze oplossing voor het probleem. Als het contact werd toegevoegd, werd deze ID voorlopig bijgehouden. Bij een volgend contact werd er nagekeken of deze reeds was toegevoegd of niet.

3.4.7 Overschakelen naar Python Programma

Tijdens het testen met PowerBI, merkte ik al snel dat de manier die ik in mijn hoofd had niet zou werken. Het was namelijk niet mogelijk om informatie van een PowerBI dashboard op te halen met een Logic App.

Na overleg met Suleyman met het nieuws dat mijn eerste plan niet zou werken werd er besloten om over te schakelen naar het manueel schrijven van een Python Programma.

3.4.8 Overschakelen naar Azure Automation Account

Tijdens het werken mar Azure Functions kwamen we meerdere problemen tegen. Kortom, dit was alles behalve gebruiksvriendelijk.

Na overleg te hebben met Robbe, één van mijn stagementoren en Jasper Beas (collega van het Cloud team binnen Toreon) werd er voorgesteld om met een Automation account te werken i.p.v. Azure Funtions. Jasper had hier reeds ervaring mee en wist dat dit gebruiksvriendelijk was.

Het werd al snel duidelijk dat dit een betere en snellere methode was. Relatief snel was alles operationeel zoals gewenst.

3.4.9 Confidentialiteit voor Teams kanalen

In de oorspronkelijke opdracht was het de bedoeling dat er voor elk verkocht project ook een Teams kanaal aangemaakt wordt. Aangezien er nagekeken moet worden of er al een Team bestaat voor de desbetreffende klant, was toegang tot alle Teams kanalen binnen Toren noodzakelijk. In deze kanalen is bevatten echter veel gevoelige informatie, dus is er besloten te werken met een update e-mail.

3.5 Het resultaat

Ook hier is van de oorspronkelijke opdracht afgeweken. Voor het Microsoft Teams gedeelte van de opdracht, zou toegang verleend moeten worden om alle Teams en kanalen uit te lezen. Aangezien dit over klantengegevens en gevoelige data gaat, is dit niet gebeurd.

Als compensatie werd er nog een extra onderdeel toegevoegd. Voor dit project werden volgende zaken gerealiseerd:

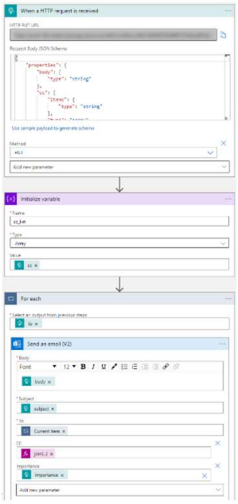
3.5.1 Logic App voor het versturen van e-mail

Om gemakkelijk update e-mails te versturen voor de andere twee delen van dit project, werd er in Microsoft Azure een Logic App aangemaakt. De Logic App wordt geactiveerd door een PUT request. Voor een correcte werking is het noodzakelijk dat deze PUT request een .json object bevat met de nodige informatie zoals: ontvanger, onderwerp, cc, bericht & prioriteit. (Zie Figuur 54)

```
{
    "to": [
        "person.1@company.com",
        "person.2@company.com",
    ],
    "cc": [
        "person.3@company.com",
        "person.4@company.com",
    ],
    "subject": "subject of the e-mail",
    "body": "This is the body of the e-mail",
    "importance": "Low / Normal / High"
}
```

Zie Figuur 53: Voorbeeld van een JSON object dat moet meegegeven worden om een mail te versturen

Deze informatie wordt dan gebruikt door de Logic App op een mail te versturen vanaf een service account van Toreon. (Zie Figuur)



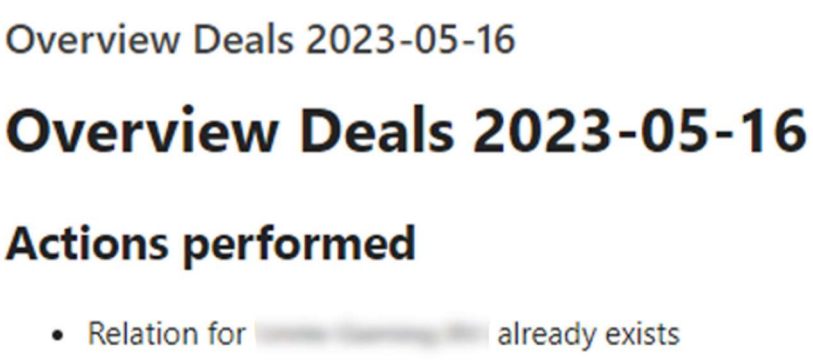
Zie Figuur 54: Screenshot van de geschreven Logic App

3.5.2 Automatische creatie van relaties voor nieuwe klanten

Wanneer er een nieuw project verkocht wordt binnen Toreon, wordt er manueel nagekeken in het IonBIZ (het ERP systeem van Toreon) of deze klant al dan niet bestaat. Indien deze niet bestaat, worden de gegevens van de klant opgezocht in HubSpot, waarna een relatie aangemaakt wordt in IonBIZ.

Dit proces werd geautomatiseerd. Er werd een programma geschreven dat kijkt of er in de afgelopen 24 uur een nieuw project werd verkocht. Voor elk gevonden project wordt er nagekeken of de klant al dan niet bestaat in IonBIZ. Indien de klant nog niet bestaat, worden de gegevens automatisch opgevraagd vanuit HubSpot en wordt er een relatie aangemaakt in IonBIZ.

Op het einde van het programma wordt er een e-mail verstuurd naar Suleyman, de project manager van Toreon waarin terug te vinden is wat het programma heeft uitgevoerd. (Zie Figuur 57 & 58) Indien er geen nieuw project verkocht is, wordt er geen e-mail verstuurd. Deze mail wordt verstuurd met behulp van de Logic App, besproken in 4.5.1.



Zie Figuur 56: Screenshot van de mail die verstuurd wordt wanneer er een deal is gevonden van een bestaande klant

Zie Figuur 57: Screenshot van de mail die verstuurd wordt wanneer er een deal is gevonden van een niet bestaande klant

Dit programma wordt met behulp van een Azure Automation Account dagelijks uitgevoerd.

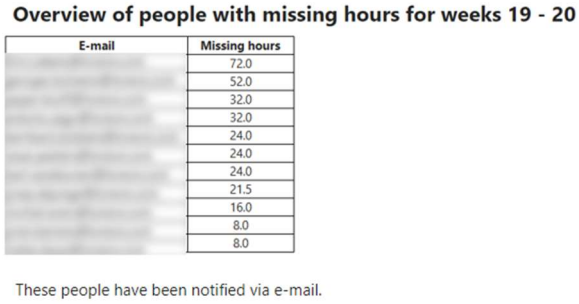
Voor Toreon betekent dit concreet dat er voor elke nieuwe deal minstens twee uur aan manueel werk wordt uitgespaard.

3.5.3 Wekelijkse controle van timesheets

Wekelijks moeten alle werknemers van Toreon hun Timesheets invullen in IonBIZ. Momenteel controleren Jasper en Suleyman (de project manager binnen Toreon) deze elke maandag manueel. Jasper (Operations Officer binnen Toreon) staat in voor de backoffices, terwijl Suleyman verantwoordelijk is voor alle consultants. Dit werd gedaan aan de hand van een PowerBI dashboard dat de uren voor alle backoffices & consultants nakijkt van de afgelopen twee weken. De mensen die niet genoeg uren ingevuld hebben, brengen Jasper en Suelyman manueel op de hoogte.

Er werd een programma geschreven dat dit proces automatiseert. Het programma vraagt alle timesheets op van alle backoffices & consultants binnen Toreon. Het timeframe, in dit geval twee weken, kan ingesteld worden naar behoeven. De externe medewerkers en freelancers worden hier niet mee in opgenomen. Hier wordt een berekening gemaakt op basis van gewerkte uren, verlofdagen en feestdagen t.o.v. het werkschema van de werknemer (voltijds, halftijds etc.). Wanneer blijkt dat er niet genoeg uren gepresteerd zijn in dit timeframe, wordt de werknemer in kwestie verwittigd a.d.h.v. een e-mail. Deze e-mail wordt verzonden met behulp van de Logic App, besproken in 4.5.1.

Aan het einde van het programma wordt er een mail naar zowel Jasper, als Suleyman. In deze mail is een tabel terug te vinden met de werknemers waarvan hun timesheets niet in orde zijn. Ook het aantal uren dat ontbreekt wordt opgenomen in deze tabel. Jasper ontvangt een mail betreffende alle backoffices (Zie Figuur 60), Suleyman voor alle consultants binnen Toreon. (Zie Figuur 61)

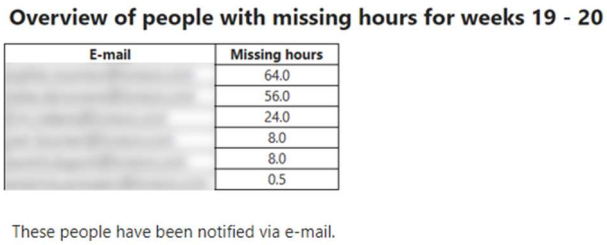


Overview of people with missing hours for weeks 19 - 20

E-mail	Missing hours
[redacted]	72.0
[redacted]	52.0
[redacted]	32.0
[redacted]	32.0
[redacted]	24.0
[redacted]	24.0
[redacted]	24.0
[redacted]	21.5
[redacted]	16.0
[redacted]	8.0
[redacted]	8.0

These people have been notified via e-mail.

Zie Figuur 59: Voorbeeldmail voor consultants



Overview of people with missing hours for weeks 19 - 20

E-mail	Missing hours
[redacted]	64.0
[redacted]	56.0
[redacted]	24.0
[redacted]	8.0
[redacted]	8.0
[redacted]	0.5

These people have been notified via e-mail.

Zie Figuur 60: Voorbeeldmail voor backoffices

Dit programma wordt met behulp van een Microsoft Azure Automation Account elke maandag uitgevoerd.

Voor Toreon betekent dit concreet voor Jasper alsook voor Suleyman een wekelijkse besparing van twee uur manueel werk.

4 CONCLUSIE

In totaal werden er drie projecten (zeven sub-opdrachten) gerealiseerd voor Toreon tijdens deze 13 weken durende stage. Niet alleen hebben deze projecten financieel hun waarde bewezen, maar ook op gebied van onderzoek, kwaliteit, etc. Onderstaand is terug te vinden welke voordelen elk project biedt voor Toreon.

- **Opdracht 1: Automatisatie SecurityReporter & CWE**
    - Verhoogde kwaliteit templates
        - Verhoogde kwaliteit rapporten
    - Extra informatie (i.v.m. compliance) in het rapport voor de klant
        - Verhoogde waarde van het rapport voor de klant
    - Overzicht situatie templates in SecurityReporter
        - Monitoring van kwaliteit templates
- **Opdracht 2: Burp Suite Extensie**
    - Extra extensie voor ethical hackers
        - Sneller herkennen van webservers
    - Research voor eindwerk collega Yael (trainee binnen Toreon)
        - Bepalen business case van Burp Suite extensie
- **Opdracht 3: Automatisatie ERP & CRM**
    - Automatisch aanmaken nieuwe relaties
        - 3 uur / nieuwe klant uitgespaard voor project manager
    - Automatisch controleren van timesheets
        - 2 uur / week uitgespaard voor project manager
        - 2 uur / week uitgespaard voor operations officer

5 REFERENTIES

All-in-One Pentest Reporting Workspace | Reporter. (n.d.). https://securityreporter.app/

CWE - 2022 CWE Top 25 Most Dangerous Software Weaknesses. (n.d.). https://cwe.mitre.org/top25/archive/2022/2022_cwe_top25.html

CWE - Common Weakness Enumeration. (n.d.). https://cwe.mitre.org/

Ecfan. (2023, May 24). *Overview - Azure Logic Apps*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/logic-apps/logic-apps-overview

GitHub: Let's build from here. (n.d.). GitHub. https://github.com/

Home. (n.d.). Jython. https://www.jython.org/

HubSpot | Software, Tools, Resources for Your Business. (n.d.). https://www.hubspot.com/

ionProjects. (n.d.). *Project Management & Resource Planning software - ionBIZ.* https://www.ionprojects.com/

Krishnat, D. (n.d.). *What Is Azure Logic Apps*. https://www.c-sharpcorner.com/article/what-is-azure-logic-apps/

Owasp. (2022, March 24). *ASVS/OWASP Application Security Verification Standard 4.0.3-en.csv at v4.0.3 · OWASP/ASVS*. GitHub. https://github.com/OWASP/ASVS/blob/v4.0.3/4.0/docs_en/OWASP%20Application%20Security%20Verification%20Standard%204.0.3-en.csv

OWASP Application Security Verification Standard | OWASP Foundation. (n.d.). https://owasp.org/www-project-application-security-verification-standard/

OWASP Top 10:2021. (n.d.). https://owasp.org/Top10/

OWASP Web Security Testing Guide | OWASP Foundation. (n.d.). https://owasp.org/www-project-web-security-testing-guide/

Pricing - Automation | Microsoft Azure. (n.d.). Microsoft Azure. https://azure.microsoft.com/en-us/pricing/details/automation/

PyPI · The Python Package Index. (n.d.). PyPI. https://pypi.org/

Shodan. (n.d.). Shodan. https://www.shodan.io/

SnehaSudhirG. (2022, October 18). *Azure Automation overview*. Microsoft Learn. https://learn.microsoft.com/en-us/azure/automation/overview

Video Conferencing, Meetings, Calling | Microsoft Teams. (n.d.). https://www.microsoft.com/en-us/microsoft-teams/group-chat-software/

Vijayan, J. (2021, July 22). Memory Corruption Issues Lead 2021 CWE Top 25. *Dark Reading*. https://www.darkreading.com/application-security/memory-corruption-issues-lead-2021-cwe-top-25

Web Application Security, Testing, & Scanning - PortSwigger. (n.d.). https://portswigger.net/

Welcome to Python.org. (2023, May 31). Python.org. https://www.python.org/