Paper: Onderzoeksvoorstel

Netwerkautomatisering aan de Universiteit Gent: Een Scriptmatige Aanpak voor Efficiënt Beheer van IP-Adressen.

Bachelorproef, 2023-2024

Stijn Coppens

E-mail: stijn.coppens@student.hogent.be
Project repo: https://github.com/stcoppens/Bachelorproef

Samenvatting

Het beheren van netwerken en het reserveren van netwerkadressen gebeurt momenteel grotendeels handmatig, wat inefficiënt is als proces, en tevens gevoelig voor menselijke fouten.
Deze bachelorproef richt zich op het uitwerken van een innovatieve, geautomatiseerde aanpak voor het beheren van netwerken en het toewijzen van netwerkadressen met behulp van scripts. Het doel van deze bachelorproef is tweeledig, namelijk (1) het creëren van een tussenlaag van scripts boven een bestaand beheerprogramma voor netwerken, en (2) het nagaan van de impact hiervan op het tijdverbruik voor het toevoegen, wijzigen en verwijderen van nieuwe IP-reserveringen in vergelijking met het huidige handmatige beheerproces. Als resultaat van de bachelorproef kan men via een webpagina die beschikbaar is binnen het bedrijfsnetwerk, mits toestemming van de netwerkbeheerder, eenvoudig netwerkadresreservaties aanmaken, wijzigen of verwijderen. De webpagina zal de scripts, die binnen het project geschreven worden, aanroepen om de nodige gegevens op de juiste manier aan te leveren aan het beheerprogramma.
Het verminderen van handmatig beheer van netwerkconfiguraties als resultaat van deze bachelorproef, zal leiden tot efficiëntiewinsten, tijdsbesparingen, een vereenvoudigde aanpak en minder fouten.

Keuzerichting: System & Network Administrator
Sleutelwoorden: DNS, DHCP, IPAM, Python

Inhoudsopgave

1. Inleiding

In de snel evoluerende wereld van technologie is het doeltreffend beheren van netwerken cruciaal geworden voor het succes van organisaties. Echter, het handmatig beheren van netwerkconfiguraties, waaronder het toewijzen van specifieke netwerkadressen, blijft een tijdrovend en foutgevoelig proces.

Om deze uitdagingen aan te pakken, zijn er onder andere geautomatiseerde oplossingen voor Internet Protocol Address Management (IPAM) ontwikkeld. Een goed uitgewerkte IPAM-tool kan een grote meerwaarde bieden voor elk netwerk doordat deze o.a. een overzicht kan geven van alle IP adressen die in gebruik zijn en hoeveel IP adressen er nog beschikbaar zijn per subnet.

Dit onderzoek zal scripts voorzien die, via een webportaal en de goedkeuring van netwerkbeheerders, gebruikers zullen toelaten te communiceren met de IPAM-tool om IP reservaties te maken, wijzigen of verwijderen. Dankzij autorisatie zullen gebruikers op het webportaal enkel wijzigingen kunnen aanbrengen voor de netwerken waarvoor de gebruiker gemachtigd is dit te doen. Nadat de netwerkbeheerder die wijzigingen via hetzelfde webportaal goedkeurt, zullen de scripts in werking treden waarbij de nodige aanpassingen gemaakt worden binnen de gebruikte IPAM-tool.

Het automatisch beheren van IPAM via scripts beoogt niet alleen de efficiëntie van het netwerkbeheer te verhogen, maar ook tijdsbesparingen

te realiseren en een meer gestroomlijnde aanpak te bieden. Deze tijdsbesparingen zijn de focus van het onderzoek, waarbij we de volgende onderzoeksvraag trachten te beantwoorden: Hoe beïnvloedt de implementatie van geautomatiseerde scripts het tijdverbruik voor het toevoegen, wijzigen en verwijderen van nieuwe IP-reserveringen in vergelijking met het huidige handmatige beheerproces?

1.1. Probleemstelling

Deze bachelorproef zal uitgevoerd worden bij Universiteit Gent (UGent), directie ICT. Momenteel werkt UGent met scripts die op basis van zogenaamde *subnetbestanden* de nodige acties doen om het netwerk te beheren.

Deze ongecrypteerde subnetbestanden stellen elk een subnetwerk (een aantal opeenvolgende netwerkadressen) voor en beschrijven cruciale informatie zoals belangrijke naamservers, welk *Virtual Local Area Network (VLAN)* nummer, gateway, etc. Hiernaast bevatten deze zowel alle beschikbare als gereserveerde netwerkadressen met daarbij eventueel enkele regels voor domeinnamen en beveiliging.

Voor elke netwerkadresreservatie die moet gebeuren, vullen geautoriseerde gebruikers via een webportaal alle nodige info in. Nadat ze deze inzenden stuurt het webportaal een mail naar de groep mailbox van de netwerkbeheerders. In deze mail zitten dan alle nodige commando's en teksten om de gevraagde wijzigingen aan te brengen aan de relevante subnetbestanden. In sommige gevallen dient de netwerkbeheerder eerst zelf nog uit te zoeken welk subnetbestand nodig is op basis van de documentatie en opzoekwerk. Indien de aanvrager van de reservatie dit heeft meegegeven in een veld voorzien voor commentaar, moet de netwerkbeheerder de eventuele domeinnaam- of beveiligingsregels handmatig toevoegen aan de reservatie.

Het overzicht van de beschikbare subnetwerken is beschreven in een interne wikipediapagina met daarbij de beschrijving van elk subnetwerk. Deze huidige aanpak brengt meerdere uitdagingen met zich mee:

- **Tijd**: Het manueel onderhouden van de scripts, subnetbestanden, netwerkadresreservaties (maken en opkuisen) kan veel tijd vragen. Alle mails met IP registraties moeten elk afzonderlijk bekeken en verwerkt worden.
- **Schaalbaarheid**: Doordat elke wijziging het bestaande bestand overschrijft en er dus geen historische data is kan men moeilijk trends herkennen. Ook wikipediapagina's moet men manueel bijwerken bij grote wijzigingen in de structuur.
- **Consistentie**: De huidige aanpak vraagt meerdere manuele acties, waardoor die vatbaar is voor menselijke fouten of vergissingen. Ook het manueel opzoeken van de relevante subnetbestanden als de aanvrager niet weet in welk subnet de aanvraag terecht moet komen, kan leiden tot registraties in verkeerde subnetten.
- **Beveiliging**: Zoals beschreven door Liao e.a. (2020) is een van de eerste stappen in een cybersecurity aanval het verzamelen van netwerk informatie via netwerk scan tools zoals Nmap. Het bewaren van alle IP registraties van het volledige domein centraal in cleartext bestanden, zoals nu het geval is, is dan ook een schat aan informatie voor elke individu met al dan niet slechte bedoelingen.

UGent is momenteel stappen aan het ondernemen voor het implementeren van *Efficiënt IP (EIP)*, een IPAM-softwarepakket. Dankzij deze implementatie is de verwachting dat de hierboven beschreven indicatoren zullen verbeteren. Om de onderzoeksvraag te beantwoorden zullen er metingen gedaan worden betreffende tijdverbruik. Hierbij zal de huidige manier van werken in tijdsgebruik vergeleken worden met die na de implementatie van het webportaal.

1.2. Doelstelling

Deze bachelorproef zal een abstractielaag maken boven EIP waarbij scripts via de *Application Programming Interface (API)* van EIP commando's zullen uitvoeren op EIP. Door de omvang van het EIP-project is het binnen de voorziene tijd van de bachelorproef niet haalbaar om UGent volledig over te zetten op de werking van EIP. Aangezien dit kritische componenten zijn, zal alles eerst uitvoerig getest worden waarbij elke stap in de migratie naar EIP weloverwogen is.

Daarom stel ik als doel om een eerste versie op te leveren van een webportaal waarop men reeds één of meerdere netwerkadresreservaties kan aanmaken, wijzigen of verwijderen. Aangezien elk subnet specifiek is voor bepaalde verdiepingen, gebouwen en/of vakgroepen, en elke medewerker binnen UGent tot een specifieke vakgroep behoort, zullen we deze informatie gebruiken om te bepalen voor welke relevante subnetten de aanvrager wijzigingen kan aanvragen. Deze aanvragen komen in een duidelijk overzicht terecht waar de netwerkbeheerders al dan niet openstaande reservaties kunnen wijzigen, goedkeuren of weigeren. Na goedkeuring worden de scripts gebruikt om de reservaties toe te passen in EIP.

Dit project geeft een antwoord op de vraag: Hoe kunnen netwerkbeheerders hun werk vereenvoudigen door het gebruik van scripts? Het

beoogde resultaat is:

- Het vereenvoudigen van veelvoorkomende taken, zoals het reserveren van internetadressen.
- Tijd besparen door het vermijden van handmatige handelingen.
- Efficiënter werken door menselijke fouten te voorkomen.
- De gebruiksvriendelijkheid verbeteren.

Deze verbeteringen zullen helpen bij het optimaliseren van de netwerkinfrastructuur.

2. Literatuurstudie

Internet Protocol (IP) is het fundament van elk gestructureerd, goed functionerend en veilig netwerk. Het geeft de mogelijkheid efficiënt gegevens te routeren, netwerken te verdelen in meer beheersbare eenheden, toegang te beperken tot gevoelige data of systemen, services te identificeren en het oplossen van netwerkproblemen (Postel, 1981). Dit hoofdstuk legt uit wat *Domain Name System (DNS)* en *Dynamic Host Configuration Protocol (DHCP)* is, waarom IPAM helpt bij het beheren van IP netwerken en waarom HTTP nodig is om te communiceren met EIP.

2.1. DNS

Mockapetris (1987) schrijft dat DNS een systeem is dat *resource records* gebruikt om onder andere vertalingen te voorzien tussen domeinnamen en IP-adressen. Als voorbeeld kan je via de browser naar google surfen via het IP-adres *142.251.36.35* of via domeinnaam *www.google.be*.

Zoals beschreven door Mockapetris (1987) voorziet DNS meerdere types resource records die netwerkbeheerders kunnen meegeven:

- **A**: Dit resource record beschrijft een host adres. Vb. *"server1.voorbeeld.com. IN A 192.168.1.1"* maakt de vertaling zodat het toestel met de domeinnaam *server1.voorbeeld.com* bereikbaar is zowel via het IP-adres *192.168.1.1* als via de domeinnaam.
- **CNAME**: Dit resource record beschrijft de canonieke naam van een host, het wordt gebruikt om een alias of subdomein naar het hoofddomein door te verwijzen. Vb. *"www.voorbeeld.com. IN CNAME server1.voorbeeld.com"* zorgt dat server1 ook bereikbaar is via "*www.voorbeeld.com*".
- **MX**: Dit resource record is een *mail exchange* record en wordt gebruikt om aan te geven welke mailservers verantwoordelijk zijn voor het ontvangen van mails binnen een domein. vb. *"voorbeeld.com. IN MX 10 mailserver.voorbeeld.com"* geeft de DNS server mee welke server de mailserver is.
- **NS**: Dit resource record is een *name server* record, het beschrijft welke DNS-servers verantwoordelijk zijn voor het beheren van DNS-informatie voor een domein. Vb. *"voorbeeld.com. IN NS dns1.voorbeeld.com"* verwijst naar *dns1* als DNS-server voor het domein "*voorbeeld.com*".
- **PTR**: Dit resource record is een *Pointer* record, het wordt gebruikt om via IP een vertaling te vragen aan de DNS-server in plaats van via de naam.
- **SOA**: Dit resource record is een *Start of Authority* record die belangrijke informatie bevat over de zone, zoals welke de primaire DNS-server, contactpersonen, etc. zijn.

2.2. DHCP

Dit protocol voorziet een framework voor het doorgeven van configuratie-informatie naar hosts (lees: computers) op het netwerk . Zo kan een computer bijvoorbeeld een IP-adres ontvangen waarmee die kan communiceren binnen het netwerk waarop die is aangesloten (Droms, 1997).

IP-netwerken worden door netwerkbeheerders op een logische manier opgesplitst in subnetwerken. Hierbij worden de beschikbare IP-adressen verdeeld in subnetwerken (subnet). Toestellen binnen subnet A zullen elkaar kunnen bereiken terwijl een toestel in een subnet B zonder de nodige routering geen verbinding zal kunnen maken met de toestellen in subnet A.

Voor DHCP zullen netwerkbeheerders subnets (of pools van IP-adressen) aanbieden aan de DHCP-server. Die zal gebruik maken van deze pools door (onder andere) IP-adressen uit te delen aan toestellen die verbinden op het netwerk en daarbij de DHCP-server laten weten dat ze nog geen IP-adres hebben.

Droms (1997) schrijft dat DHCP drie mechanismes gebruikt voor het uitdelen van IP-adressen:

- **Automatisch**: Permanent toewijzen van een IP-adres.
- **Dynamisch**: IP-adres voor een bepaalde tijd toewijzen.
- **Manueel**: Een (door de netwerkbeheerder) vooraf bepaald IP-adres toewijzen, in vakjargon noemt met dit een IP-reservatie.

2.3. IPAM

Naast de vele uitdagingen die zowel DNS als DHCP met zich meebrengen, is het beheren van de vele DNS records, IP-adres ranges en de vaak

vele IP-reservaties zeker iets waar een netwerkbeheerder over moet waken. Een mogelijke oplossing hiervoor is het gebruiken van IP Address Management (IPAM) via softwarepakketten die IPAM aanbieden. IPAM laat toe IP-adressen efficiënt te beheren in een netwerk, het leidt tot een gestructureerde aanpak waardoor conflicten tussen subnetten worden vermeden. Het geeft een compleet overzicht van het netwerk met percentages van hoeveel adressen beschikbaar en in gebruik zijn. IPAM geeft eveneens de mogelijkheid om de historiek bij te houden waardoor het van pas komt voor schaalbaarheid en beveiliging van het netwerk (Rooney & Dooley, 2020).

2.4. HTTP/HTTPS

Om communicatie met de API van EIP mogelijk te maken wordt er gebruik gemaakt van *Hypertext Transfer Protocol (HTTP)*. Dit is een client-serverprotocol die communicatie mogelijk maakt op het Internet. Zoals beschreven door Fielding en Reschke (2014) maakt HTTP gebruik van *Uniform Resource Identifiers (URI's)* om unieke web-resources te identificeren en biedt het verschillende methoden *(GET, POST, PUT, DELETE)* waarmee clients acties kunnen uitvoeren op serverresources. HTTP is *stateless*, elke aanvraag is onafhankelijk, en statuscodes zoals "200 OK" en *headers* worden gebruikt om de resultaten en aanvullende informatie van serververzoeken aan te geven, waardoor een gestandaardiseerde communicatie tussen clients en servers mogelijk is. Om deze informatieoverdracht te beveiligen, wordt *Hypertext Transfer Protocol Secure (HTTPS)* gebruikt. HTTPS bouwt voort op HTTP, maar voegt een extra beveiligingslaag toe door middel van *Secure Sockets Layer (SSL)/Transport Layer Security (TLS)*-encryptie, waardoor de uitwisseling van gegevens tussen client en server wordt versleuteld. Hierdoor worden alle IP-registraties en andere gegevens beter beschermd tegen potentiële aanvallen.

3. Methodologie

In dit hoofdstuk wordt beschreven hoe lang en hoe de nodige metrics voor de onderzoeksvraag worden opgemeten, welke software het project gebruikt, hoe deze worden toegepast, en welke fases het project zal doorlopen.

3.1. Software

Alle scripts worden geschreven in **Visual studio code** in de programmeertaal **Python**. Om de HTTP methodes naar de API van EIP te testen, wordt er gebruik gemaakt van **Postman**.

- **Visual Studio Code**: Vanwege de ondersteuning voor meerdere programmeer- en scripttalen, geïntegreerde Git-ondersteuning, debug- en extensie mogelijkheden wordt gekozen om alle scripts in Visual Studio Code te schrijven.
- **Python**: Van Rossum en Drake (2011) beschrijven Python als een eenvoudige, doch krachtige programmeertaal. De beschikbaarheid van Python-pakketten zoals 'requests' maakt het mogelijk om efficiënt gegevens door te sturen naar API's zoals die van EIP. Als interpretatieve taal biedt Python snelle ontwikkeling zonder de noodzaak van compilatie. Python is ook uitbreidbaar, waardoor het eenvoudig is om nieuwe functies toe te voegen. Kortom, Python, met zijn netwerkbibliotheken, vormt een ideale keuze voor dit onderzoeksproject (Van Rossum & Drake, 2011).
- **Postman**: Dit programma geeft de mogelijkheid alle HTTP-verzoeken manueel te maken, versturen en ontvangen. Hierbij is het mogelijk om alle onderdelen van het verzoek te manipuleren en na te gaan wat als resultaat verwacht kan worden bij het uitsturen van bepaalde verzoeken naar de API.

3.2. Methodiek scripts

Voordat een script wordt geschreven, worden eerst gerichte testen gedaan waarbij commando's met parameters naar de API van EIP worden verzonden. Hierbij wordt er zowel naar de resultaten van de API gekeken als naar wat er op EIP zelf gebeurt via de webpagina. Eens deze testen voor een commando afgelopen zijn, wordt er pas overgegaan tot het schrijven van het Python-script. Hierbij worden telkens de parameters binnen elk script opgezet volgens de voorwaarden van de API.

3.3. Fase 1. Pre-analyse

In de eerste fase van de bachelorproef worden er dagelijks metingen gedaan om na te gaan hoeveel tijd een netwerkbeheerder van UGent gebruikt om IP reservaties aan te maken, te wijzigen en/of te verwijderen. Hierbij wordt voor elk type actie de tijd afzonderlijk gemeten en beschreven in een spreadsheet die als bijlage aan het onderzoek wordt toegevoegd.

3.4. Fase 2. Literatuurstudie

Voor deze tweede fase worden veertien dagen uitgetrokken. In deze fase wordt gezocht naar documentatie en literatuur van gelijkaardige projecten waarbij een webpagina met formulieren en logins gebruiken om scripts aan te roepen. Daarnaast wordt er ook uitgebreid aandacht gegeven aan de literatuur van Efficiënt IP zelf om na te gaan welke eisen deze stelt voor het aanmaken, wijzigen en verwijderen van IP-adressen. Op het einde van deze fase zal er een verslag geschreven

worden met alle belangrijke punten die zijn meegenomen uit de literatuur.

3.5. Fase 3. Scripts schrijven

Binnen de derde fase wordt het grootste stuk van de Python-scripts geschreven. Deze fase voorziet telkens één week om een script te schrijven en daarop aansluitend één week om het geschreven script te testen en te troubleshooten. De drie scripts zijn:

- Maken van een IP reservatie
- Wijzigen van een bestaande IP reservatie
- Verwijderen van een bestaande IP reservatie

3.6. Fase 4. Webpagina schrijven

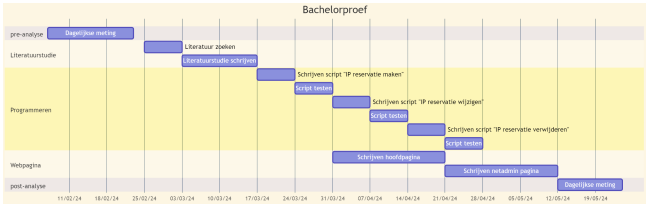
Het schrijven van de webpagina begint halverwege fase twee aangezien deze nauw op elkaar aansluiten. Voor het schrijven van de hoofdpagina van waar de drie scripts kunnen worden aangeroepen, worden drie weken voorzien. Hierna zijn er nog drie weken voorzien voor het schrijven van de netwerkbeheerderspagina. Op deze pagina kan de netwerkbeheerder de gevraagde wijzigingen nakijken, aanpassen en goed- of afkeuren.

3.7. Fase 5. Post-analyse

Tijdens deze laatste fase worden er opnieuw metingen gedaan om na te gaan hoeveel tijd de netwerkbeheerders van UGent nodig hebben voor elke afzonderlijke registratie. Ook in deze fase zullen de metingen in een spreadsheet komen om de nodige vergelijkingen te maken met de data uit de eerste fase van de bachelorproef.

3.8. Gantt stappenplan

Een overzicht van alle fases worden weergegeven in dit Gantt stappenplan



Figuur 1: Gantt Stappenplan

4. Verwachte resultaten

De verwachte resultaten omvatten een succesvolle eerste implementatie van een webportaal waarop men reeds één of meerdere netwerkadresreservaties kan aanmaken, wijzigen of verwijderen. Deze eerste versie van het intuïtieve webportaal zou een verbeterde gebruikerservaring moeten bieden door middel van geoptimaliseerde API-aanroepen. Verder zou de automatisering van netwerkconfiguraties,
met name IP-adresallocatie, moeten leiden tot verminderde complexiteit, verbeterde efficiëntie en algemene gebruiksvriendelijkheid in het netwerkbeheerproces. Dit zal een belangrijke stap zijn om na het project op voort te bouwen om uiteindelijk een product op te leveren.

5. Verwachte conclusie

Dankzij het implementeren van deze webpagina met de onderliggende scripts, kunnen medewerkers van Universiteit Gent eenvoudig, consistent, en snel IP-reservaties maken, wijzigen en verwijderen. De netwerkbeheerders kunnen al deze wijzigingen dan op hun eigen webpagina controleren, aanpassen en beoordelen. Na dit onderzoek zullen er nog voldoende mogelijkheden zijn om de webpagina aan te vullen met extra functies, in die mate dat netwerkbeheerders zelf geen wijzigingen meer dienen te maken binnen de IPAM tool zelf.

Referenties

Droms, R. (1997). *Dynamic Host Configuration Protocol* (tech. rap.). RFC Editor. https://doi.org/10.17487/RFC2131

Fielding, R. T., & Reschke, J. (2014, juni). Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. https://doi.org/10.17487/RFC7231

Liao, S., Zhou, C., Zhao, Y., Zhang, Z., Zhang, C., Gao, Y., & Zhong, G. (2020). A Comprehensive Detection Approach of Nmap: Principles, Rules and Experiments. *2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 64–71. https://doi.org/10.1109/CyberC49757.2020.00020

Mockapetris, P. (1987). *Domain names-concepts and facilities* (tech. rap.). https://doi.org/10.17487/RFC1034

Postel, J. (1981). *Internet protocol* (tech. rap.). https://doi.org/10.17487/RFC0791

Rooney, T., & Dooley, M. (2020). *IP Address Management*. John Wiley & Sons. https://doi.org/10.1002/9781119692263

Van Rossum, G., & Drake, F. L. (2011). *An Introduction to Python*. Network Theory Ltd. http://atk.fam.free.fr/fichiers/stage/Python/JF/site/pytut.pdf