Naam: **Philippe Boets**
Datum: **6 december 2020**
Richting: **3 ITF CCS B (Elektronica ICT)**
Vak: **Emerging Technologies (Z15779)**
Docent: **Bart Portier, Bram Verbruggen**

Assignment 2: DevSecOps

1. Inleiding: wat is DevSecOps?

De meeste onder jullie hebben al gehoord van DevOps, het samenbrengen van software development en software operations. Maar wat is DevSecOps nu juist?

In deze paper wordt onderzocht

- wat men bedoeld met DevSecOps,
- hoe het in verschillende fases van de CI/CD (Continous Integration/Continuous Deployment) pipeline geïntegreerd wordt,
- waarom het een beter principe is dan de traditionele manier van werken
- …

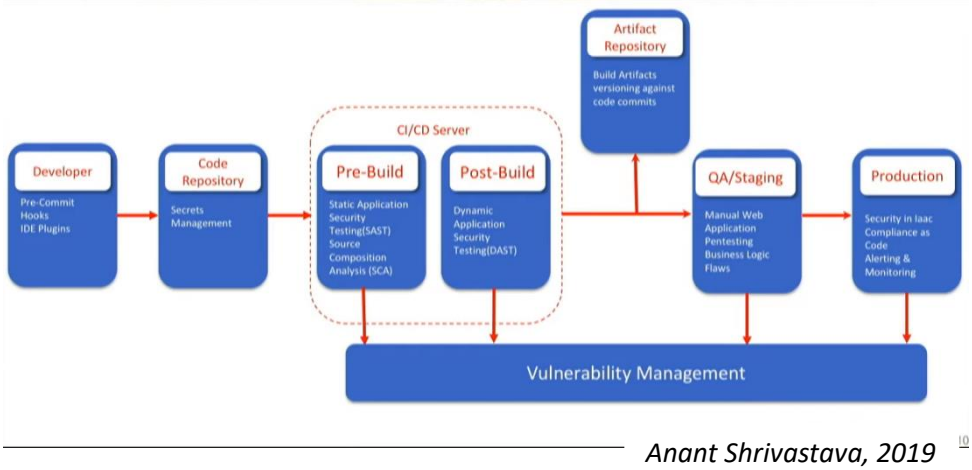
DevSecOps is een methode waarbij getracht wordt de DevOps pipeline 'Secure By Default' te maken. Waar in het verleden het security gedeelte van een DevOps pipeline door een apart team aan het einde van de DevOps cyclus verzorgd werd, wordt security vanaf nu in het begin van de development cycle in acht genomen. Daarvoor is het belangrijk om één multi-skilled team samen te stellen dat kennis heeft over alle onderdelen van DevSecOps. Dit in plaats van drie teams, één voor elk onderdeel development, security en operations, die elk hun eigen spcialiteit hebben.

> ***"If DevOps is about increasing the level of communication between development and operations, then DevSecOps is about inviting security into the conversation."***
>
> (New Context, 2019)

Het belang van DevSecOps wordt duidelijk als we kijken naar traditionele methodes om software secure te maken. Deze methodes, zoals pentesting, kunnen dagen in beslag nemen en dit is nadelig voor de snelheid van de development pipeline. Door gebruik te maken van verschillende tools kan security vanaf het begin geïntegreerd worden in de life cycle van software. Tools voor de automatisering van bepaalde security taken (bvb automated code review) en concepten zoals Code zijn hier ook uiterst belangrijk om security dichter bij de developer te krijgen. DevSecOps is dus het concept met als doel security een deel van het DevOps proces te maken.

2. Hoe wordt DevSecOps geïmplementeerd?

In verschillende fases van de pipeline wordt security op verschillende manieren geïmplementeerd. Volgende is een schema waar te zien is welke stappen er in elke fase van de pipeline genomen kunnen worden om security te verbeteren.



Anant Shrivastava, 2019 [10]

Om taken zoals email notificaties en security checks te automatiseren binnen de CI/CD pipeline wordt gebruik gemaakt van tools zoals Jenkins. Jenkins maakt gebruik van verchillende plugins om deze taken uit te voeren in de verschillende fases van de pipeline.

2.1 Developer

In eerste instantie heeft de developer zelf al verschillende tools ter beschikking om fouten te voorkomen. Deze tools kunnen op de computer van de developer geïnstalleerd worden of kunnen als plugins in de IDE (de programmeeromgeving) geïntegreerd worden.

IDE security plugins, zoals Pytyhon Security en Snyk Vulnerability Scanner, worden geïntegreerd in de programmeeromgeving van de developer (bvb PhpStorm) om makkelijk voorkombare fouten direct op te sporen. Deze kunnen enkel simpele fouten opsporen en helpen dus de developer om geen “domme” fouten te maken. Ze zorgen er dus niet voor dat complexe security issues opgespoord worden.

Pre-commit Hooks daarintegen zijn niet geïntegreerd in de IDE, maar worden rechtstreeks op de computer van de developer geïnstalleerd. Deze tools zorgen ervoor dat de developer niet per ongeluk sensitieve informatie mee commit in een git-repository (of eender welk ander version-control systeem). Dit is om te voorkomen dat informatie zoals access keys, SSH keys of wachtwoorden mee gepusht worden naar een repository.

2.2 Code repository

Vaak worden zogenaamde secrects, geheime informatie niet voor de buitenwereld bestemd, ook opgeslagen in een repository. Om dit veilig te laten verlopen worden Secret Management Tools, zoals Vault, AWS Secrets Manager en Keywhiz, gebruikt. Dit gaat over privé certificaten voor TLS of SSL, wachtwoorden en encryptie sleutels. Deze tools helpen dus bij het beheren en beveiligen van deze data.

2.3 CI/CD server

Op de CI/CD server kan gebruikt gemaakt worden van Static Analysis Security Testing (SAST) en Dynamic Analysis Security Testing (DAST), dit zijn automatische review tools, zoals Fortify Static Code Analyzer, Veracode en Netsparker. SAST, ook wel White Box Testing, zoekt voor kwetsbaarheden in de code zelf zonder de code uit te voeren. DAST, ook wel Black Box Testing, gaat de verbindingen tussen de web front-end en de applicatie zelf nakijken op kwetsbaarheden. Deze gaat dus kijken naar een lopende applicatie om fouten op te sporen. SAST en DAST zijn er beiden om in een vroege fase van development al meteen feedback te geven over de veiligheid.

Er kan ook gebruik gemaakt worden van Software Composition Analysis tools. Deze tools, zoals Black Duck en Sonatype Nexus Platform, zijn specifiek bedoeld om de code van third party libraries na te kijken op kwetsbaarheden. Een heel groot deel van de software is namelijk niet zelf geschreven, maar wordt door middel van third-party modules geïmplementeerd. Deze modules kunnen outdate of kwetsbaar zijn. Software Composition Analysis tools zijn dus bedoeld om deze fouten op te sporen.

Deze tools kunnen ook gebruikt worden om de beveiliging voor Infrastructure as Code (IaC) te verbeteren. Als een organisatie gebruik maakt van het IaC principe kunnen deze tools ingezet worden om de code voor de infrastructuur na te kijken. Dit zorgt voor een algemene verbetering van de veiligheid binnen de organistatie.

2.4 Productie omgeving

In de productie omgeving is beveiliging ook een belangrijke schakel. Hier kan een organisatie veel voordeel halen uit het gebruik van Infrastructure as Code (IaC). IaC is het concept van het beheer van het datacenter door code, in tegenstelling tot de traditionele opvatting van het beheer van fysieke hardware. IaC maakt gebruik van containers, Docker of Kubernetes, Virtual Machines (VM's) en hypervisors, zoals Vmware, die runnen op bare-metal servers. Dit zorgt ervoor dat de hardware gemanaged wordt door code, en deze code is zoals hierboven besrpoken onderwerpbaar aan verschillende audit tools om de beveiliging te verbeteren. IaC zorgt niet alleen voor minder risico's, maar is ook sneller en voordeliger dan traditionele datacenters.

3. DevSecOps in een cloud omgeving

Als een organisatie gebruik maakt van een cloud oplossing zijn er een aantal zaken waar de organisatie zelf rekening mee moet houden, die niet de verantwoordelijkheid van de cloud provider zijn. Een deel van de beveiliging is wel de verantwoordelijkheid van de cloud provider. Een organisatie kan verschillende tools gebruiken om ervoor te zorgen dat haar cloudomgeving veilig blijft. Dit zijn tools voor DAST, SAST en Software Composition Analysis die eerder al besproken zijn. Het is dus belangrijk dat als IaC in een cloudomgeving gebruikt wordt dat de software ook veilig is.

4. Vulnerability Management

Al de tools hierboven besproken genereren natuurlijk rapporten, heel veel rapporten. Om manueel al deze rapporten te filteren is een onbegonnen zaak, daarvoor worden Vulnerability Management tools, zoals Jackhammer of Archery, gebruikt. Deze tools hebben elk een eigen manier van presenteren, maar doen allemaal hetzelfde. Ze verzamelen alle rapporten, filteren, normaliseren en standaardiseren deze om al de rapporten leesbaar te maken. Daarna worden de rapporten verzameld in een centraal dashboard.

5. Conclusie: Culturele shift nodig

Om DevSecOps efficiënt te implementeren is een hele culturele shift nodig. In eerste instantie zijn het alleen de security experts die alle rapporten kunnen begrijpen en verwerken, maar op termijn is het dus wenselijk dat alle leden van het team dit kunnen. Er is dus een grootschalige (her)scholing nodig op het terrein om DevSecOps de norm te maken en dit efficiënt te implementeren. Het implementeren van al deze tools alleen is dus niet voldoene om de security bottlenecks op te lossen, maar er is ook grootschalige shift in de manier van werken nodig om dit te implementeren.

Als conclusie kunnen we stellen dat DevSecOps de toekomst van DevOps is, omdat het op een efficiëntere manier met security omgaat. Zeker in deze tijden waar remote werken de norm is, is security een echte must.

Bibliografie

BeyondTrust. (2020). *Secrets Management*. Retrieved from beyondtrust: https://www.beyondtrust.com/resources/glossary/secrets-management

Jenkins. (2020). Retrieved from jenkins.io: https://www.jenkins.io/

New Context. (2019, 9 10). *The "What" "How" and "Why" of DevSecOps*. Retrieved from New Context: https://newcontext.com/what-is-devsecops/

Red Hat, I. (2020). *What is DevSecOps?* Retrieved from Redhet.com: https://www.redhat.com/en/topics/devops/what-is-devsecops

Shrivastava, A. (2020, 1 15). *DevSecOps : What, Why and How*. Retrieved from youtube: https://www.youtube.com/watch?v=DzX9Vi_UQ8o&ab_channel=BlackHat

Wikipedia. (2020). *Infrastructure as code*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Infrastructure_as_code

XenonStack. (2020, 11 2). *A Quick Guide to DevSecOps Pipeline*. Retrieved from XenonStack: https://www.xenonstack.com/insights/devsecops-pipeline/