Резултат (Result) 2.3
Отчет със сравнителен анализ на автоматизирано внедряване на сигурен софтуер

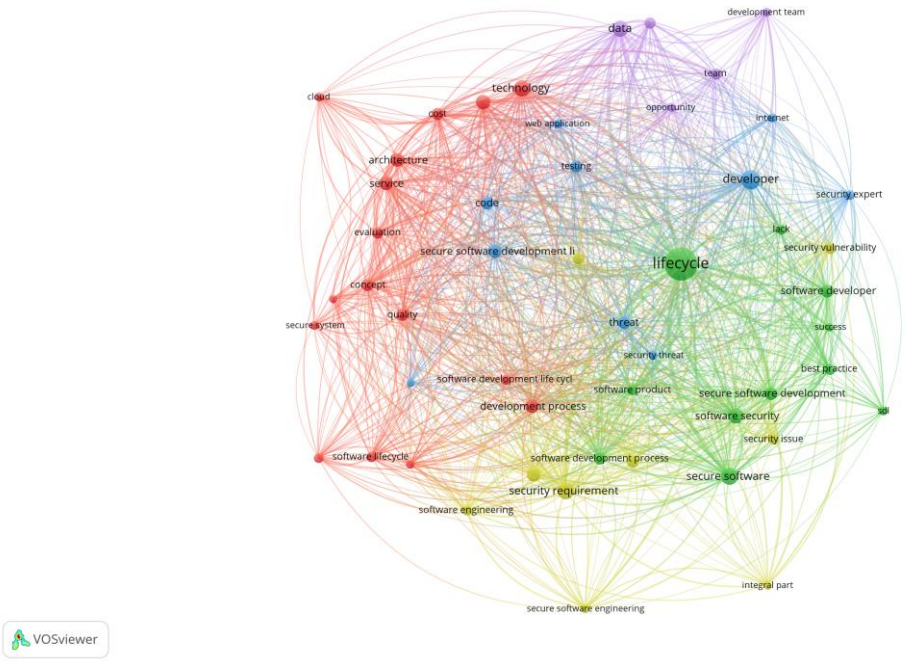
1. Методология на изследването.

Методологията на изследването за откриване и анализ на аномалии в софтуерни продукти вклщчва няколко стъпки:

1.) Търсене на научни публикации в различни източника с търсене по тема, анотация и ключови думи чрез използване на дигитални библиотеки. Заявката за търсене беше:

```
TITLE-ABS-KEY((''Secure Software'') AND (''Secure Software Development Lifecycle'') AND (''automated deployment''))
```

2.) След това създаваме подобна карта въз основа на индексираните ключови думи (фиг. 1). Те се различават от ключовите думи, предоставени от авторите, и предоставят повече възможности за наблюдение и заключения по време на анализа.



Фиг. 1 Индексна карта на ключовите думи.

На базата на тази карта са анализирани основните понятия и връзки, които да се разгледат в научните публикации.

2. Автоматизирано внедряване на сигурен софтуер

През последните години, процеса за разработката на софтуер премина през много етапи и адаптации в това как киберсигурността се внедрява при програмирането на код. Традиционните подходи към осигуряването на защитни механизми в софтуера често включват отделни екипи и процеси в една организация, което води до охлабени мерки за сигурност и забавено идентифициране на уязвимости в кода. Появата на DevSecOps в този смисъл направи революция в пейзажа за разработка на софтуер, като наблегна върху

внедряването на най-добрите практиките за киберсигурност от самото началото от жизнения цикъл за разработка на софтуер.[2].

DevSecOps (разработка (Dev), сигурност (Sec) и операции (Ops)) е разширение на DevOps, което се счита за средство за непрекъснат процес за разработване на код, също така неговите операции и сигурност. Поради своите съображения за сигурност и безопасност, голяма част от компаниите започват да обмислят за внедряването на DevSecOps като процес за разработка на сигурни приложения, когато става въпрос за прилагане на DevOps и информационна сигурност. Този документ има за цел да проучи състоянието и практиката на DevSecOps като се има за цел да наблегне върху сътрудничеството и автоматизацията при разработването на софтуер. Това проучване също така разглежда две от основните практики на DevSecOps за откриване на уязвимости в най-ранния етап от жизнения цикъл на разработката на софтуер.

2.1. Статично и динамично тестване за сигурността в приложенията

Този раздел от изследването се фокусира върху анализирането на съществуваща литература, свързана с темата на изследването, специално в областта на тестването на сигурността на приложенията, а именно статично тестване на сигурността на приложенията (SAST) и динамично тестване за сигурност на приложенията (DAST), които ще бъдат разгледани като различни подходи, които се вписват в конвейера на DevSecOps.

Повечето научни статии по тази тема разглеждат последователно характеристиките на двата вида методологии - идентифициране на разликите между тях и установяването на общо основание, че комбинираното им използване осигурява по-високо ниво на защита срещу кибератаки. Някои научни статии предоставят изключително подробна и систематична информация за характеристиките на статичното и динамично сканиране, такива статии като [3] и [4]. Има и научни статии, които разглеждат и двете методологии като отделна тема. Има примери за подобни научни статии, които се фокусират конкретно върху статичното тестване на сигурността на приложенията [5]. Те адресират само статичния метод за тестване на сигурността на приложения, но са представени в по-всеобхватен аспект. Други примерни научни статии са фокусират само върху динамично тестване на сигурността [6], тези статии представят динамичното сканиране на софтуер много по-подробно.

Основното разграничение между динамичното и статичното сканиране за сигурност е в начина и техните подходи за тестване на сигурността. Статичното сканиране изследва кода на приложение по време на фазата на разработка на софтуер, метод който има за цел да открие потенциални уязвимости в програмния код, които биха могли да бъдат злоупотребени от злонамерени лица. В обратния смисъл, динамичното тестване за сигурност на приложения е метод за оценка на сигурността, при който се използват активни техники за изпитване и анализиране на работещо приложение. Вместо да се фокусира върху изходния код или статичните характеристики на приложението, динамично сканиране се използва активни проверки по време на изпълнение на готовия софтуерен продукт.

DAST работи като метод за тестване black box, това представлява симулиране на атака от външна гледна точка, ако приемем, че тестерът не притежава познания за вътрешна работа на приложението. Този подход позволява на динамичното сканиране да идентифицира уязвимости в сигурността, които статично сканиране може да не успее да открие, особено тези уязвимости, които се проявяват по време на изпълнението на програмата.

Допълнителна информация за характеристиките, свойства и разликите между двата вида методи за сканиране на софтуер могат да бъдат намерени в [7]. Ясно е, че статичното тестване се използва за сканиране на изходния код в най-ранния етап от жизнения цикъл на разработка на софтуер, докато динамичното тестване извършва сканиране по време на изпълнение на приложението в края на жизнения цикъл [8]. По време на статичното сканиране разработчиците и тестерите имат пълни познания за структурата на приложението, реално имат достъп до изходния код, докато динамичното тестване се счита като подход за тестване от гледната точка на хакер, където тестерите нямат достъп до изходния код на приложението и нямат никаква идея за неговата структура.

2.2. Модели DevOps и DevSecOps

Въз основа на концепции на модела DevSecOps, тестване на сигурността (по отношение на кода и приложението в общ характер) се извършва главно на етапи, когато разработчиците се опитват да интегрират своите решения в съществуващото приложение. Тъй като уязвимостите могат да възникнат във всеки етап от традиционния DevOps модел, тестове за сигурност са изпълнявани основно в модела DevSecOps, включително моделиране на заплахите, както и статично и динамично сканиране за уязвимости в приложенията. DevOps акцентира върху автоматизацията на рутинните задачи, което не само подобрява ефективността, но и намалява вероятността от човешки грешки. Но често възникват проблеми със сигурността, които произтичат от конфликти между различните цели на разработчиците и екипите за киберсигурност.

Основната характеристика на DevOps представлява автоматизацията на много софтуерни тестове и интеграционни процеси, което позволява на бизнеса да създава и доставя нови версии на софтуера бързо и безпроблемно. Въпреки, че подходите Agile и DevOps доминират индустрията за разработка на софтуер, сигурността понякога е пренебрегната в двете страни.

Въз основа на предишното изявление за DevOps, DevSecOps има за цел да интегрира решения, свързани със сигурността във всяка една фаза от „тръбопроводния“ pipeline модел на DevOps с цел подобряване на сигурността на доставения продукт. DevSecOps предлага възможност за увеличаване на скоростта на процеса за интегриране на промени в жизнения цикъл на разработката на софтуер като същевременно поддържа високи стандарти за качество и сигурност. Идеалната цел е „откриване на проблеми в сигурността възможно най-рано в процеса от разработване на код“, както се твърди в статията на DevSecOps на Owasp.org [9].

2.3 DevSecOps практики и принципи

В днешно време организациите са изправени пред множество предизвикателства, когато се свежда до внедряването на DevSecOps в техните среди. Въз основа на изследването на ZScaler [10], известни са няколко проблема и препятствия, които провалят организациите по пътя към успешно внедряване на DevSecOps:

- Адресиране и смекчаване на уязвимостите - Според констатации от Security Boulevard [11], организации, които не са се адаптирали към DevSecOps, оставят 50% от приложенията си постоянно уязвими към хакерски атаки, за разлика от 22% степен на уязвимост в организации с добре установен DevSecOps подход. Обикновено тестовете за сигурност се провеждат към края на жизнения цикъл за разработка на софтуер, принуждавайки разработчиците да променят кода в по-късните етапи, което води до скъпи преработки и закъснения в проекта на разработване.
- Балансиране на скорост и сигурност - Постигането на баланс между скорост и сигурност е от решаващо значение в областта на DevOps. DevOps набляга на бързината и гъвкавостта, изисквайки всички екипи, включително екипа по киберсигурност да поддържат добро темпо за гладкото протичане на тези иновационни процеси. Придържайки се към Принципите на DevOps те включват установяване на сигурна основа, която е едновременно гъвкава, адаптивна и бърза. Остарели инструменти и процеси за сигурност влияят неблагоприятно на скоростта за развитие и внедряване.
- Сигурността се счита за пречка – Възприема се като пречка за прогреса. Както съобщава Gartner [11], 71% от CISO (главните служители по информационна сигурност) заявяват, че голяма част от техните служители продължават гледат на сигурността като на пречка за постигане на по-бързо и ефективно решение в DevOps. Често срещано погрешно схващане в рамките на екипите от разработчици и DevOps екипите са, че сигурността пречи на скоростта, често се разглежда като пречка в цялостния процес.
- Липса на ресурси и пропуски в знанията. Недостатъчни ресурси и разделение в знанията - Последните статистики разкриват че 70% от организациите не притежават достатъчно практическо разбиране на практиките на DevSecOps [12]. Освен това, съществено предизвикателство произтича от ограниченията в персонала, инструментите и бюджетните разпределения. Преодоляване на разликата в липсата на знания е също толкова предизвикателна. На разработчиците често им липса опит в сигурността - широко разпространено препятствие в областта на DevSecOps. По същия начин екипите по киберсигурност и оперативните екипи често не са запознати и с обратното - инфраструктура и среди за разработка на софтуер. Тази празнина в знанията, съчетана с липсата на универсална платформа за разпространение на знания, представлява значителна пречка пред ефективното приемане на DevSecOps.

- Разпределение на ролите и отговорностите – Съгласуването на ролите и отговорностите е трудна задача, поради динамичния характер на DevOps средите, където екипите претърпяват постоянни промени. Разработчиците обикновено приемат, че екипът по сигурността е единствено отговорен за сигурността и за намаляване на риска от хакерски атаки. В действителност обаче ролята на екипа по сигурност включва създаването на политики за сигурност, насочващи разработчиците и операторите да разберат изискванията и практики за сигурност с цел създаване на защитен код. Най-предизвикателният аспект на адаптирането и приемането на DevSecOps често се върти около хората и организационните структури.

DevSecOps е подход, който има за цел да интегрира практики и принципи за сигурност във всеки един етап от жизнения цикъл за разработка на софтуер. DevSecOps насърчава сътрудничеството, автоматизацията и проактивния подход към софтуерна сигурност чрез обединяване на разработчици, екипи по киберсигурност и оперативни екипи. Основните принципи и практики, свързани с DevSecOps са:

- Сигурност "Най-вляво"- DevSecOps подчертава концепцията за промяна на практиките за сигурност и представя методът "най-вляво" от процеса за разработване на софтуер чрез ранното интегриране на дейности по сигурността, като например по време на събиране на изисквания, проектиране и фази по моделиране на код, вместо да третира сигурността на по-късен етап.
- Автоматизация – Автоматизацията играе решаваща роля в DevSecOps. Тя включва използването на инструменти, скриптове и рамки за автоматизиране на проверки за сигурност, тестване, процеси на внедряване и наблюдение. Автоматизирано сканиране на сигурността, оценките на уязвимостта и непрекъснатото наблюдение помага за бързото идентифициране и решаване на проблеми със сигурността.
- Непрекъсната интеграция и непрекъснато внедряване(CI/CD) - DevSecOps насърчава използването на CI/CD pipeline модел за улесняване на непрекъсната интеграция на софтуерни решения, непрекъснато тестване и непрекъснато внедряване на софтуер. Организациите могат бързо и сигурно да допускат софтуерни актуализации като същевременно поддържат необходимите контроли за сигурност чрез автоматизиране на изграждането и тестването на процеси за внедряване.
- Сътрудничество и комуникация – DevSecOps набляга на сътрудничеството и ефективната комуникация между разработчиците, екипите по киберсигурност и оперативните екипи. Тясното сътрудничество гарантира за това, че изискванията за сигурност, най-добрите практики и обратната връзка са интегрирани безпроблемно в процеса за разработване на софтуер. Освен това насърчава споделената отговорност за сигурност сред членовете на екипа.

- Инфраструктура като код (IaC) - Инфраструктурата като код е практика, при която инфраструктурни компоненти, например като сървъри, мрежи и конфигурации са определени и се управляват чрез код. Приемането на IAC позволява на екипите повече гъвкавост, контрол на версиите и автоматизирането при промени в инфраструктура, което прави конфигурациите за сигурност повече последователни, повторяеми и подлежащи на одит.

- Тестване на сигурността – DevSecOps се застъпва за включващи тестове за сигурност навсякъде и във всеки един етап от жизнения цикъл от разработка на софтуер, който включва провеждане на редовни оценки на сигурността, тестове за проникване, прегледи на кода и сканиране за уязвимости за ранно идентифициране и смекчаване на уязвимостите в сигурността

- Постоянно наблюдение - Непрекъснатото наблюдение включва активно наблюдение на софтуер и инфраструктура за събития и аномалии в областта на сигурността. Това включва регистриране, откриване на заплахи в реално време и реакция при инцидент. Позволява непрекъснат мониторинг на организациите да реагират за своевременното откриване на пробиви в сигурността, уязвимости и подозрителни дейности.

- Култура на сигурност – DevSecOps цели да насърчи културата на сигурност в екипите за разработка на софтуер и организациите. Това включва създаване на осъзнатост, обучение на членовете на екипа за най-добри практики за сигурност. Чрез внушаване на сигурност – приемането на сигурността като начин на мислене гарантира, че тя ще бъде приета насериозно и по подразбиране във всеки един аспект от разработката на софтуер

Прилагането на тези ключови принципи и практики на DevSecOps помага на организациите да установят стабилна и проактивна позиция за сигурност в целия жизнен цикъл за разработване на софтуер. Чрез интегриране на сигурността още от самото в процеса за разработване на софтуер, приемането и внедряването на автоматизация и сътрудничество, организациите могат да се подобрят своята софтуерна сигурност, което също ще подобри и ефективността на разработката чрез доставянето на по-сигурни и устойчиви приложения.

3. Списък с използвана литература.

[1] “The 21 Latest Emerging Cyber Threats to Avoid”, https://www.aura.com/ learn/emerging-cyber-threats, (accessed May 24, 2023).

[2] Mao, Runfeng, et al. "Preliminary findings about devsecops from grey literature." 2020 IEEE 20th international conference on software quality, reliability and security (QRS). IEEE, 2020.

[3] J. Yang, L. Tan, J. Peyton and K. A Duer, "Towards Better Utilizing Static Application Security Testing," 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), Montreal, QC, Canada, 2019, pp. 51-60, doi: 10.1109/ICSE-SEIP.2019.00014.

[4] T. Rangnau, R. v. Buijtenen, F. Fransen and F. Turkmen, "Continuous Security Testing: A Case Study on Integrating Dynamic Security Testing Tools in CI/CD Pipelines," 2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC), Eindhoven, Netherlands, 2020, pp. 145-154, doi: 10.1109/EDOC49727.2020.00026.

[5] Oyetoyan, Tosin Daniel, et al. "Myths and facts about static application security testing tools: an action research at Telenor digital." Agile Processes in Software Engineering and Extreme Programming: 19th International Conference, XP 2018, Porto, Portugal, May 21–25, 2018, Proceedings 19. Springer International Publishing, 2018.

[6] M. R. Stytz and S. B. Banks, "Dynamic software security testing," in IEEE Security & Privacy, vol. 4, no. 3, pp. 77-79, May-June 2006, doi: 10.1109/MSP.2006.64.

[7] Sharma, Manish. "Review of the Benefits of DAST (Dynamic Application Security Testing) Versus SAST." INTERNATIONAL JOURNAL OF MANAGEMENT AND ENGINEERING RESEARCH 1.1 (2021): 05-08.

[8] Dencheva, Lyubka. Comparative analysis of Static application security testing (SAST) and Dynamic application security testing (DAST) by using open-source web application penetration testing tools. Diss. Dublin, National College of Ireland, 2022.

[9] “OWASP DevSecOps Guideline - v-0.2”, https://owasp.org/ www-project-devsecops-guideline/latest/,(accessed May 26, 2023).

[10] “The Top Challenges Faced by Organizations Implementing DevSecOps”,https://www.zscaler.com/blogs/product-insights/top-challenges-faced-organizations-implementing-devsecops, (accessed May 26, 2023).

[11] “20 Statistics That Today’s DevSecOps Teams Should Know”,https://securityboulevard.com/2021/05/20-statistics-that-todays-devsecops-teams-should-know/, (accessed May 26, 2023)

[12] “Developers lack skills needed for secure DevOps, survey shows”,https://www.computerweekly.com/news/450424614/Developers-lack-skills-needed-for-secure-DevOps-survey-shows, (Accessed May 26, 2023)