ZESZYTY NAUKOWE
WYŻSZEJ SZKOŁY BANKOWEJ W POZNANIU
2025, T. 108, NR 1

TOMASZ JANCZEWSKI
Wyższa Szkoła Bankowa w Warszawie
ORCID 0009-0006-4583-4377
e-mail: tomasz@janczewski.it

Implementacja praktyk DevSecOps w środowiskach chmurowych dla infrastruktury krytycznej. Analiza bezpieczeństwa procesów CI/CD

Streszczenie. Celem artykułu jest analiza implementacji praktyk DevSecOps w środowiskach chmurowych, szczególnie tych związanych z infrastrukturą krytyczną, w kontekście rosnących zagrożeń cybernetycznych. Analiza i ocena efektywności narzędzi zabezpieczających, takich jak SAST, DAST, SCA i IAST, w procesach integracji, testowania i wdrażania oprogramowania (CI/CD) opiera się na danych ankietowych zebranych za pomocą ankiety przeprowadzonej wśród ekspertów zajmujących się implementacją DevSecOps w różnych sektorach infrastruktury krytycznej. Wyniki ankiety potwierdzają, że wdrożenie praktyk DevSecOps w środowiskach chmurowych dla infrastruktury krytycznej, w tym automatyzacja testów bezpieczeństwa w procesach CI/CD, przyczynia się do zmniejszenia liczby podatności wykrywanych na etapie produkcyjnym i skrócenia czasu reakcji na incydenty związane z bezpieczeństwem, a jednocześnie pozwala utrzymać ciągłość dostaw oprogramowania. Implementacja DevSecOps jest kluczowym sposobem zapewnienia bezpieczeństwa infrastruktury krytycznej w środowiskach chmurowych. Skuteczna strategia cyberbezpieczeństwa wymaga systematycznego podejścia do bezpieczeństwa w całym cyklu życia oprogramowania, wdrażania zautomatyzowanych narzędzi testowych oraz monitorowania wszystkich składników oprogramowania. Wyzwaniem pozostaje jednak znalezienie kompromisu między wymaganiami bezpieczeństwa a potrzebą szybkiego dostarczania oprogramowania.

Słowa kluczowe: DevSecOps, bezpieczeństwo chmury, infrastruktura krytyczna, SAST, DAST, SCA, IAST, SBOM, CI/CD

https://doi.org/10.58683/dnswsb.2064

1. Wprowadzenie

Transformacja cyfrowa i zwiększające się tempo zmian technologicznych stawiają przed organizacjami nowe wyzwania w obszarze cyberbezpieczeństwa. Szczególnie istotnym aspektem staje się ochrona infrastruktury krytycznej, obejmującej

sektory energetyczne, transportowe, bankowe, opieki zdrowotnej i administracji publicznej. Jak wskazują badania, organizacje zarządzające infrastrukturą krytyczną muszą dostosować swoje strategie bezpieczeństwa do dynamicznie zmieniającego się krajobrazu zagrożeń, uwzględniając zarówno aspekty techniczne, jak i organizacyjne (Tøndel, Line & Jaatun, 2015). Z uwagi na rosnącą migrację do środowisk chmurowych tradycyjne podejście do bezpieczeństwa aplikacji okazuje się niewystarczające wobec dynamiki współczesnych zagrożeń. Hashizume, Rosado, Fernández-Medina i Fernandez (2013) podkreślają, że architektury chmurowe wprowadzają nowe wektory ataku i wymagają zmiany w podejściu do bezpieczeństwa, aby skutecznie zarządzać ryzykiem związanym z rozproszonymi zasobami i współdzieloną odpowiedzialnością.

W tradycyjnym modelu tworzenia oprogramowania bezpieczeństwo było traktowane jako ostatni etap procesu deweloperskiego, co często prowadziło do późnego wykrywania podatności. W środowisku praktyków testowania bezpieczeństwa podejście to określane jest jako „shift-right" — oznacza ono wykrywanie zagrożeń na końcowych etapach cyklu życia oprogramowania. Myrbakken i Colomo-Palacios (2017) wskazują, że DevSecOps promuje podejście „shift-left", czyli integrację zabezpieczeń na wcześniejszych etapach cyklu życia oprogramowania. Dzięki temu organizacje mogą zmniejszyć ryzyko oraz koszty naprawy luk bezpieczeństwa, zanim produkt trafi do środowiska produkcyjnego.

Historyczne incydenty bezpieczeństwa, takie jak wyciek danych w Equifax w 2017 roku czy atak na SolarWinds z 2020 roku, dobitnie pokazały konsekwencje pomijania aspektów bezpieczeństwa na wczesnych etapach tworzenia oprogramowania. W przypadku Equifax niezałatana podatność w bibliotece Apache Struts umożliwiła hakerom dostęp do danych ponad 140 milionów użytkowników. U.S. Senate Committee on Homeland Security and Governmental Affairs (2018) wskazuje, że podatność CVE-2017-5638, która została wykryta i załatana w marcu 2017 roku, nadal pozostawała niezałatana w systemach Equifax do maja tego samego roku, co pozwoliło na skuteczny atak i masowe wycieki danych. Ataki APT często wykorzystują luki w łańcuchach dostaw oprogramowania do dystrybucji złośliwego kodu, jak opisano w badaniach Alshamrani et al. (2019). Przykładem takiej strategii był incydent SolarWinds w 2020 roku, w którym brak odpowiednich mechanizmów bezpieczeństwa w łańcuchu dostaw oprogramowania pozwolił na wstrzyknięcie złośliwego kodu do aktualizacji systemu, które następnie zostały dystrybuowane do tysięcy klientów, w tym agencji rządowych i dużych korporacji.

W odpowiedzi na te wyzwania rozwinęła się koncepcja DevSecOps, która integruje bezpieczeństwo na każdym etapie cyklu życia oprogramowania, od projektowania i kodowania po wdrożenie i monitorowanie (Myrbakken & Colomo-

-Palacios, 2017). DevSecOps stanowi rozszerzenie filozofii DevOps, dodając do niej perspektywę bezpieczeństwa jako integralną część procesu tworzenia aplikacji.

Niniejszy artykuł ma na celu przedstawienie kompleksowej analizy implementacji praktyk DevSecOps w środowiskach chmurowych dla infrastruktury krytycznej. Skupimy się na integracji bezpieczeństwa z procesami CI/CD (Continuous Integration/Continuous Deployment), wykorzystaniu narzędzi takich jak SAST (Static Application Security Testing), DAST (Dynamic Application Security Testing), SCA (Software Composition Analysis) czy IAST (Interactive Application Security Testing), automatyzacji testów bezpieczeństwa oraz generowaniu SBOM (Software Bill of Materials) dla pełnej przejrzystości komponentów oprogramowania.

W poszczególnych sekcjach artykułu omówimy historyczny kontekst problemów bezpieczeństwa w tradycyjnym podejściu do tworzenia oprogramowania, przybliżymy koncepcję DevSecOps i jej fundamentalne zasady, przedstawimy kluczowe narzędzia i praktyki stosowane w tym podejściu, a także przedstawimy praktyczne przykłady implementacji DevSecOps w środowiskach chmurowych dla infrastruktury krytycznej. Artykuł zakończymy analizą wyzwań i przyszłych kierunków rozwoju praktyk DevSecOps.

2. Przegląd literatury

2.1. Ewolucja podejścia do bezpieczeństwa aplikacji

Literatura przedmiotu wskazuje na systematyczną ewolucję podejścia do bezpieczeństwa aplikacji, od modeli reaktywnych do proaktywnych. Jak wskazują Leppänen, Honkaranta i Costin (2022), tradycyjne podejścia do bezpieczeństwa, które koncentrują się na dodawaniu zabezpieczeń jako ostatniego kroku rozwoju oprogramowania, stały się przestarzałe w obliczu rosnącej złożoności konfiguracji oprogramowania i zwiększającej się liczby naruszeń bezpieczeństwa. Wskazano, że wprowadzenie strategii DevSecOps, w tym wcześniejszej integracji testów bezpieczeństwa oraz lepszej współpracy między zespołami, pozwala na skuteczniejsze zarządzanie ryzykiem i szybsze wykrywanie podatności. Myrbakken i Colomo-Palacios (2017) wskazują, że przesunięcie testowania bezpieczeństwa na wcześniejsze etapy cyklu życia oprogramowania (tzw. podejście „shift-left") pozwala na wcześniejsze wykrywanie podatności, co redukuje koszty ich usuwania oraz minimalizuje ryzyko operacyjne. Wczesna integracja zabezpieczeń w procesie DevSecOps jest kluczowym czynnikiem poprawiającym efektywność kosztową organizacji.

Transformacja w kierunku DevOps, a następnie DevSecOps wynika z potrzeby zwiększenia efektywności procesu wytwarzania oprogramowania przy jednoczes-

nym zachowaniu wysokiego poziomu bezpieczeństwa. Bass et al. (2015) opisują DevOps jako zestaw praktyk i narzędzi mających na celu poprawę współpracy między zespołami deweloperskimi i operacyjnymi poprzez automatyzację procesów oraz wspólne zarządzanie cyklem życia oprogramowania. DevSecOps rozszerza koncepcję DevOps, integrując praktyki bezpieczeństwa na wszystkich etapach cyklu życia oprogramowania. Mohan i Othmane (2016) podkreślają, że kluczowym aspektem tego podejścia jest współpraca między zespołami programistów, operacji i bezpieczeństwa, co umożliwia efektywne zarządzanie ryzykiem i wdrażanie mechanizmów ochrony już na wczesnych etapach rozwoju aplikacji.

Makani i Jangampeta (2021) wskazują, że organizacje stosujące praktyki DevSecOps osiągają większą skuteczność w wykrywaniu i eliminacji podatności dzięki integracji narzędzi bezpieczeństwa w całym cyklu życia oprogramowania. Automatyzacja testów oraz stosowanie zaawansowanych narzędzi analizy statycznej i dynamicznej pozwalają na wcześniejsze wykrywanie zagrożeń, co skutkuje poprawą bezpieczeństwa oraz redukcją kosztów związanych z usuwaniem podatności. Ponadto Rajapakse, Zahedi, Babar i Shen (2022) wskazują, że implementacja DevSecOps znacząco przyspiesza wykrywanie i eliminację podatności poprzez automatyzację procesów bezpieczeństwa oraz integrację testów w cyklu życia oprogramowania. Praktyki DevSecOps redukują opóźnienia w naprawie podatności, umożliwiając szybszą reakcję na zagrożenia. Znacząca poprawa w tym kontekście oznacza zarówno skrócenie czasu reakcji na wykryte podatności, jak i zmniejszenie liczby incydentów wymagających interwencji na późnych etapach cyklu życia aplikacji. Dodatkowo, DevSecOps wprowadza mechanizmy wczesnego wykrywania błędów, co pozwala na ich eliminację jeszcze przed wdrożeniem do środowiska produkcyjnego, zmniejszając tym samym koszty naprawy i potencjalne ryzyko biznesowe.

2.2. Regulacje i standardy w kontekście bezpieczeństwa infrastruktury krytycznej

Infrastruktura krytyczna, z uwagi na swoje strategiczne znaczenie, podlega szczególnym regulacjom prawnym. W Unii Europejskiej kluczową rolę odgrywa dyrektywa NIS2, która zastępuje wcześniejszą dyrektywę NIS z 2016 roku. NIS2 znacząco rozszerza zakres podmiotów zobowiązanych do przestrzegania norm cyberbezpieczeństwa, obejmując takie sektory jak energetyka, transport, bankowość, opieka zdrowotna oraz administracja publiczna.

Dyrektywa NIS2 nakłada na organizacje obowiązek wdrożenia zaawansowanych środków technicznych i organizacyjnych, regularnych ocen ryzyka oraz zgłaszania poważnych incydentów (European Parliament and Council, 2022). Nieprze-

strzeganie tych wymogów wiąże się z sankcjami finansowymi sięgającymi nawet 10 milionów euro lub 2% globalnego rocznego obrotu przedsiębiorstwa.

W Stanach Zjednoczonych podobną rolę odgrywa framework NIST (National Institute of Standards and Technology), który definiuje standardy i wytyczne w zakresie bezpieczeństwa infrastruktury krytycznej. NIST Special Publication 800-53 i NIST Cybersecurity Framework dostarczają organizacjom narzędzi do oceny i poprawy zdolności w zakresie zarządzania ryzykiem cyberbezpieczeństwa.

Międzynarodowa norma ISO/IEC 27001 stanowi globalny standard dla systemów zarządzania bezpieczeństwem informacji, a jej uzupełnienie ISO/IEC 27017 odnosi się specyficznie do bezpieczeństwa w chmurze obliczeniowej. Z kolei standard OWASP (Open Web Application Security Project) Top Ten identyfikuje najczęstsze zagrożenia dla aplikacji webowych, dostarczając deweloperom praktycznych wskazówek dotyczących zapobiegania im (OWASP, 2021).

Tøndel et al. (2015) wskazują, że organizacje utrzymujące infrastrukturę krytyczną przyjmują różne podejścia do zarządzania incydentami bezpieczeństwa informacji, w zależności od ich rozmiaru i poziomu dojrzałości operacyjnej. Badanie podkreśla, że większe organizacje mają bardziej sformalizowane procedury reagowania na incydenty, natomiast mniejsze podmioty często stosują bardziej elastyczne, ale mniej usystematyzowane metody zarządzania ryzykiem. Ponadto, skuteczność strategii bezpieczeństwa jest silnie uzależniona od poziomu organizacyjnej gotowości oraz zasobów dedykowanych do ochrony infrastruktury krytycznej.

2.3. Wyzwania bezpieczeństwa w środowiskach chmurowych

Środowiska chmurowe, choć oferują liczne korzyści w zakresie skalowalności, elastyczności i efektywności kosztowej, wprowadzają również nowe wyzwania związane z bezpieczeństwem. Rittinghouse & Ransome (2010) identyfikują specyficzne wyzwania chmury, takie jak współdzielona odpowiedzialność za bezpieczeństwo, kwestie związane z wielodostępnością (multi-tenancy), ryzyko utraty kontroli nad danymi oraz złożoność zarządzania tożsamością i dostępem.

Alshamrani et al. (2019) podkreślili, że zagrożenia APT coraz częściej wykorzystują luki w środowiskach chmurowych, w tym błędy konfiguracyjne, nieprawidłowe zarządzanie dostępem oraz podatności w kodzie aplikacji. Badanie to wskazuje na konieczność kompleksowego podejścia do bezpieczeństwa w chmurze, łączącego zarówno aspekty infrastrukturalne, jak i aplikacyjne, co ma kluczowe znaczenie dla ochrony przed zaawansowanymi cyberatakami.

Gartner (2021) zwraca uwagę, że większość naruszeń bezpieczeństwa w chmurze wynika z błędów konfiguracyjnych i niewłaściwego zarządzania tożsamoś-

cią, co podkreśla znaczenie wdrożenia odpowiednich polityk kontroli dostępu, wzmacniając jeszcze bardziej potrzebę wdrożenia automatycznych mechanizmów kontroli bezpieczeństwa, które są kluczowym elementem podejścia DevSecOps.

2.4. Narzędzia i praktyki DevSecOps

Literatura przedmiotu identyfikuje szereg narzędzi i praktyk DevSecOps, które pozwalają na skuteczną integrację bezpieczeństwa z procesami CI/CD. Narzędzia bezpieczeństwa aplikacji można podzielić na kilka kategorii, w tym narzędzia do analizy statycznej kodu (SAST), analizy dynamicznej (DAST), analizy składu oprogramowania (SCA), interaktywnej analizy bezpieczeństwa aplikacji (IAST) oraz zarządzania podatnościami. Forward Security (2020) wskazuje, że każde z tych narzędzi pełni odmienną funkcję w wykrywaniu i eliminacji zagrożeń, a ich integracja pozwala na skuteczniejsze zabezpieczenie aplikacji na różnych etapach cyklu życia oprogramowania.

Zgodnie z AmberTeam Testing (2024) SAST (Static Application Security Testing) analizuje kod źródłowy lub pliki wykonywalne aplikacji bez jej uruchamiania, wykrywając potencjalne podatności, takie jak SQL Injection, Cross-Site Scripting (XSS) oraz niebezpieczne wywołania funkcji. Dzięki analizie kodu na wczesnym etapie cyklu życia oprogramowania, możliwe jest wychwycenie błędów jeszcze przed wdrożeniem aplikacji do środowiska produkcyjnego. Z kolei Mahmood i Mahmoud (2018) wskazują, że narzędzia SAST skutecznie identyfikują podatności w kodzie źródłowym jeszcze przed jego wdrożeniem, co może znacząco zmniejszyć ryzyko wystąpienia luk bezpieczeństwa na późniejszych etapach cyklu życia aplikacji, w tym aplikacji chmurowych. Autorzy podkreślają, że ich zastosowanie w procesach automatyzacji, w tym w pipeline'ach CI/CD używanych najczęściej w kontekście DevOps i DevSecOps, pozwala na wcześniejsze eliminowanie luk bezpieczeństwa, zanim kod trafi do środowiska produkcyjnego, co poprawia ogólną jakość i bezpieczeństwo oprogramowania.

DAST (Dynamic Application Security Testing) testuje aplikację w trakcie jej działania, symulując ataki na działający system. Narzędzia DAST, takie jak OWASP ZAP (Zed Attack Proxy, obecnie ZAP by Checkmarx), pozwalają na identyfikację podatności, które mogą nie być widoczne w analizie kodu źródłowego, np. problemy związane z konfiguracją serwerów czy nieodpowiednim zarządzaniem sesjami.

SCA (Software Composition Analysis) koncentruje się na analizie bibliotek i komponentów zewnętrznych używanych w aplikacji. Badania pokazują, że współczesne aplikacje składają się w 70–80% z kodu zewnętrznego, co czyni SCA kluczowym elementem strategii bezpieczeństwa. Leppänen et al. (2022) w swoim przeglądzie literatury na temat DevSecOps wskazują na znaczenie narzędzi SCA

w wykrywaniu podatności w komponentach zewnętrznych, wskazując na ich kluczową rolę w zarządzaniu bezpieczeństwem w cyklach CI/CD.

IAST (Interactive Application Security Testing) łączy elementy SAST i DAST, analizując aplikację od wewnątrz podczas jej działania. OWASP (2021) podkreśla, że metoda ta umożliwia dynamiczne testowanie bezpieczeństwa, jednocześnie zachowując precyzję analizy kodu źródłowego, co pozwala na wykrywanie podatności w rzeczywistym kontekście wykonania aplikacji. Narzędzia IAST monitorują przepływ danych w aplikacji, a to daje możliwość precyzyjnego lokalizowania podatności i redukcji fałszywych alarmów. Według raportu Gartner (2021) narzędzia IAST mają potencjał do zmniejszenia liczby fałszywych alarmów w porównaniu do narzędzi SAST i DAST, co pozwala na bardziej precyzyjną identyfikację rzeczywistych zagrożeń.

SBOM (Software Bill of Materials) to ustrukturyzowany wykaz wszystkich komponentów, bibliotek i modułów używanych w aplikacji. SBOM jest szczególnie istotny w kontekście infrastruktury krytycznej, ponieważ umożliwia szybką identyfikację i reakcję na nowo odkryte podatności w komponentach zewnętrznych. CISA podkreśla, że organizacje wykorzystujące SBOM mogą szybciej identyfikować i reagować na podatności w łańcuchu dostaw oprogramowania (CISA, 2023).

3. Metodologia

Badanie zostało przeprowadzone z zastosowaniem podejścia mieszanego, obejmującego systematyczny przegląd literatury oraz analizę wyników ankiet eksperckich. Celem było kompleksowe zrozumienie praktyk implementacji DevSecOps w środowiskach chmurowych, szczególnie w kontekście infrastruktury krytycznej.

W pierwszym etapie przeprowadzono systematyczny przegląd literatury naukowej i branżowej. Analizie poddano publikacje związane z praktykami DevSecOps, bezpieczeństwem aplikacji w środowiskach chmurowych oraz specyficznymi wyzwaniami infrastruktury krytycznej. Kryteria doboru literatury obejmowały prace recenzowane, raporty branżowe oraz dokumentacje techniczne publikowane w języku angielskim lub polskim, które ukazały się w ciągu ostatnich kilku lat. Publikacje zostały wybrane na podstawie ich relewantności oraz istotności dla tematu badania, szczególnie w zakresie integracji zabezpieczeń z procesami CI/CD (Continuous Integration/Continuous Deployment), zastosowania narzędzi automatyzujących bezpieczeństwo oraz zarządzania dostępem.

W drugim etapie przeprowadzono badania ankietowe wśród ekspertów praktycznie zajmujących się implementacją DevSecOps w różnych sektorach infrastruktury krytycznej, takich jak energetyka, finanse, opieka zdrowotna, admini-

stracja publiczna, telekomunikacja, transport, usługi komunalne oraz obronność. Ankieta została przygotowana w formie kwestionariusza internetowego, zawierającego pytania otwarte oraz zamknięte dotyczące najskuteczniejszych praktyk DevSecOps, efektywnych narzędzi zabezpieczających, największych wyzwań związanych z implementacją tych praktyk, stosowanych metryk oceny skuteczności oraz przyszłych innowacyjnych kierunków rozwoju w obszarze bezpieczeństwa.

Analiza wyników ankietowych umożliwiła identyfikację kluczowych aspektów praktycznych oraz porównanie ich z wnioskami z przeglądu literatury. Uzyskane informacje pozwoliły na wyciągnięcie wniosków na temat najczęściej stosowanych i rekomendowanych narzędzi bezpieczeństwa, takich jak Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), Software Composition Analysis (SCA) oraz Continuous Security Posture Management (CSPM). W analizie uwzględniono także znaczenie automatyzacji procesów bezpieczeństwa, zarządzania tożsamością i dostępem (IAM) oraz regularnych testów penetracyjnych.

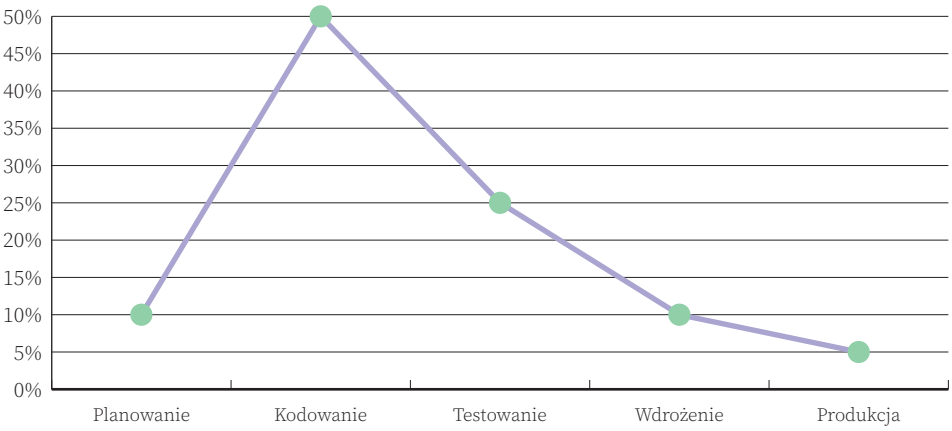
Podsumowując, wykorzystana metodologia dała możliwość szczegółowej oceny stanu wdrożeń DevSecOps oraz identyfikacji najlepszych praktyk stosowanych w ochronie infrastruktury krytycznej w środowiskach chmurowych.

4. Wyniki

4.1. Wdrożenie praktyk DevSecOps w środowiskach infrastruktury krytycznej

Analiza wyników badań ankietowych wśród specjalistów z sektora infrastruktury krytycznej wyraźnie wskazuje na rosnącą świadomość i szerokie zastosowanie praktyk DevSecOps. Respondenci potwierdzili, że podejście to skutecznie ogranicza liczbę krytycznych podatności wykrywanych na etapie produkcyjnym. Dominującym trendem jest stopniowe przenoszenie testów bezpieczeństwa na wcześniejsze etapy procesu rozwoju oprogramowania (tzw. „shift-left"), co pozwala na eliminację zagrożeń już na poziomie kodu źródłowego. Ilustracją powyższego trendu jest rysunek 1.

Specjaliści szczególnie podkreślają znaczenie automatyzacji procesów testowych, m.in. poprzez zastosowanie narzędzi typu Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST) oraz Software Composition Analysis (SCA). Ankietowani zgodnie twierdzą, że wdrożenie ciągłego skanowania podatności w pipeline'ach CI/CD znacząco skraca czas potrzebny na identyfikację i usunięcie luk.



Rys. 1. Wykres ilustrujący dominujący trend przenoszenia testów bezpieczeństwa na wcześniejsze etapy procesu rozwoju oprogramowania („shift-left”).
Źródło: Opracowanie własne

4.2. Najczęściej wykorzystywane narzędzia i technologie

Wśród narzędzi, które respondenci wskazali jako najbardziej efektywne w zabezpieczaniu aplikacji w chmurze, dominują te pozwalające na integrację z pipeline'ami CI/CD oraz zapewniające możliwość ciągłego monitorowania bezpieczeństwa. Szczególnym uznaniem cieszą się systemy typu Cloud Security Posture Management (CSPM), które automatycznie identyfikują błędne konfiguracje chmurowe oraz skanery kontenerów, takie jak Aqua Security czy Snyk.

Przykładowy fragment konfiguracji pipeline'a CI/CD ze zintegrowanym narzędziem SAST (SonarQube):

```
stages:
- build
- security_scan
- deploy

security_scan:
stage: security_scan
script:
- sonar-scanner -Dsonar.projectKey=CriticalInfra -Dsonar.sources=.
   -Dsonar.host.url=$SONAR_HOST
only:
- master
- merge_requests
```

Kolejnym elementem są narzędzia do analizy bezpieczeństwa konfiguracji infrastruktury jako kodu (IaC). Specjaliści szczególnie często wskazują na konieczność wdrażania polityk bezpieczeństwa jako integralnej części procesu rozwoju infrastruktury chmurowej.

Przykładowy fragment kodu dla automatycznej walidacji konfiguracji IaC za pomocą narzędzia Checkov:

```
checkov -d terraform/ --framework terraform --output junitxml >
   checkov-report.xml
```

4.3. Wyzwania integracji DevSecOps z istniejącymi systemami

Wyniki ankiet jasno wskazują, że jednym z głównych wyzwań podczas implementacji DevSecOps w infrastrukturze krytycznej jest konieczność integracji nowych praktyk z systemami legacy. Przestarzałe systemy sterowania przemysłowego (SCADA) i aplikacje działające w izolowanych sieciach często nie są przystosowane do współczesnych procesów ciągłej integracji i automatyzacji. Trendem zauważanym przez respondentów jest konieczność tworzenia dedykowanych warstw adaptacyjnych umożliwiających łączenie starszych technologii z nowoczesnymi praktykami DevSecOps.

4.4. Kultura współpracy i jej rola w skuteczności DevSecOps

Respondenci zgodnie podkreślają rolę kultury organizacyjnej jako kluczowego czynnika sukcesu implementacji DevSecOps. Coraz powszechniejsze stają się interdyscyplinarne zespoły złożone z przedstawicieli rozwoju, bezpieczeństwa oraz operacji (Dev, Sec, Ops), które systematycznie współpracują, definiując wspólne cele bezpieczeństwa. Regularne warsztaty i szkolenia mają za zadanie podnosić świadomość zagrożeń i wzmacniać współpracę, co bezpośrednio przekłada się na skuteczność identyfikacji i eliminacji podatności.

4.5. Efektywność monitorowania bezpieczeństwa

Kluczowym trendem zaobserwowanym w ankietach jest wzrost znaczenia zaawansowanego monitoringu bezpieczeństwa aplikacji działających w środowiskach produkcyjnych. Narzędzia SIEM (Security Information and Event Management), takie jak Splunk czy ELK Stack, są wykorzystywane do gromadzenia i analizy danych o zagrożeniach w czasie rzeczywistym. Wyniki ankiet wskazują, że integracja systemów monitoringu z automatycznymi mechanizmami reagowania na

incydenty (SOAR — Security Orchestration, Automation and Response) pozwala znacząco skrócić średni czas reakcji na zagrożenia.

4.6. Znaczenie regulacji i standardów

Specjaliści biorący udział w badaniu wskazali, że znaczący wpływ na praktyki DevSecOps mają regulacje prawne oraz międzynarodowe standardy branżowe. Jako determinanty strategii cyberbezpieczeństwa są często wymieniane przede wszystkim dyrektywa NIS2, RODO oraz standardy ISO/IEC 27001 oraz NIST. Odpowiednie stosowanie tych regulacji pozwala na ustrukturyzowanie procesów bezpieczeństwa i efektywne zarządzanie ryzykiem.

Tabela 1. Podsumowanie dominujących trendów w implementacji praktyk DevSecOps w środowiskach infrastruktury krytycznej na podstawie wyników badań ankietowych oraz analizy literatury przedmiotu

Obszar	Obszar
Automatyzacja bezpieczeństwa	Rosnące zastosowanie narzędzi SAST/DAST/SCA
Integracja z legacy systemami	Potrzeba tworzenia warstw adaptacyjnych
Kultura DevSecOps	Wzrost znaczenia interdyscyplinarnych zespołów Dev-Sec-Ops
Monitorowanie	Intensyfikacja wykorzystania narzędzi SIEM/SOAR
Regulacje	Kluczowe znaczenie NIS2, ISO/IEC 27001, RODO oraz frameworku NIST

Źródło: Opracowanie własne

4.7. Znaczenie automatyzacji i „shift-left”

Podsumowując wyniki ankietowe oraz dane z literatury przedmiotu, należy podkreślić, że wyraźnym trendem jest przeniesienie odpowiedzialności za bezpieczeństwo na wcześniejsze etapy cyklu życia oprogramowania. Organizacje, które skutecznie wdrażają ten model, czerpią korzyści w postaci niższej liczby incydentów bezpieczeństwa oraz wyraźnego skrócenia czasu naprawy luk bezpieczeństwa. Automatyzacja, zarówno na poziomie kodu źródłowego, konfiguracji infrastruktury, jak i monitorowania środowiska produkcyjnego, pozostaje fundamentem współczesnego podejścia do cyberbezpieczeństwa w środowiskach chmurowych infrastruktury krytycznej.

5. Wnioski

Analiza wyników badań oraz przegląd aktualnej literatury wskazują, że praktyki DevSecOps odgrywają kluczową rolę w skutecznym zabezpieczaniu infrastruktury krytycznej działającej w środowiskach chmurowych. Integracja bezpieczeństwa z procesami CI/CD umożliwia organizacjom wcześniejsze wykrywanie i eliminowanie podatności, znacząco redukując ryzyko ataków cybernetycznych oraz minimalizując potencjalne szkody.

Jednym z najważniejszych wniosków płynących z badań jest dominujący trend „shift-left", polegający na przenoszeniu testów bezpieczeństwa na początkowe fazy cyklu życia oprogramowania. Takie podejście zapewnia znacząco szybsze reagowanie na zagrożenia, obniżając koszty oraz skracając czas potrzebny na ich usunięcie. Automatyzacja procesów bezpieczeństwa, szczególnie testów statycznych (SAST), dynamicznych (DAST) oraz analizy zależności (SCA), stała się powszechnym standardem i podstawą współczesnej strategii bezpieczeństwa w infrastrukturze krytycznej.

Wyniki badań ankietowych wskazują również na istotne znaczenie kultury współpracy i świadomości cyberbezpieczeństwa. Interdyscyplinarne zespoły, łączące ekspertów z dziedzin rozwoju oprogramowania, bezpieczeństwa oraz operacji, przyczyniają się do bardziej skutecznego zarządzania ryzykiem cyberbezpieczeństwa. Tworzenie dedykowanych grup roboczych i systematyczne prowadzenie szkoleń pozwala budować organizacyjną kulturę bezpieczeństwa, zwiększając skuteczność procesów DevSecOps.

Ważnym aspektem, jaki został podkreślony przez respondentów, jest konieczność efektywnego monitorowania infrastruktury produkcyjnej. Zaawansowane narzędzia monitorujące oraz systemy klasy SIEM i SOAR, które integrują się bezpośrednio z procesami automatycznymi, umożliwiają organizacjom sprawniejsze reagowanie na incydenty. Skracają tym samym średni czas wykrycia i odpowiedzi na zagrożenia, co bezpośrednio przekłada się na podniesienie ogólnego poziomu bezpieczeństwa.

Ponadto szczególną uwagę należy zwrócić na trudności związane z wdrażaniem praktyk DevSecOps w środowiskach bazujących na starszych systemach legacy oraz infrastrukturze SCADA. Ze względu na specyfikę systemów przemysłowych, niezbędne staje się projektowanie odpowiednich warstw adaptacyjnych umożliwiających integrację z nowoczesnymi narzędziami i metodami pracy. Organizacje powinny zwracać szczególną uwagę na zapewnienie ciągłości działania, szczególnie w kontekście rygorystycznych wymogów regulacyjnych, takich jak NIS2 czy ISO/IEC 27001.

Warto również podkreślić rolę regulacji i standardów bezpieczeństwa jako czynników kształtujących strategię implementacji DevSecOps. W szczególności

przestrzeganie wymagań określonych w dyrektywach unijnych (np. RODO, NIS2) oraz standardach branżowych nie tylko zapewnia zgodność prawną, ale także wprowadza usystematyzowane podejście do zarządzania cyberbezpieczeństwem.

Podsumowując, implementacja DevSecOps w środowiskach chmurowych dla infrastruktury krytycznej staje się koniecznością strategiczną, która wymaga holistycznego podejścia, obejmującego technologię, procesy oraz kulturę organizacyjną. Integracja narzędzi bezpieczeństwa z procesami CI/CD, budowanie kultury bezpieczeństwa oraz efektywne zarządzanie regulacjami są kluczowymi determinantami sukcesu w zabezpieczaniu infrastruktury krytycznej w erze cyfrowej.

Bibliografia

Alshamrani, A., Myneni, S., Chowdhary, A. i Huang, D. (2019). A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. *IEEE Communications Surveys & Tutorials, 21*(2), 1851–1877.

AmberTeam Testing. (2024). *Testy bezpieczeństwa: Static Application Security Testing (SAST)*. https://amberteam.pl/testy-bezpieczenstwa/sast/

Bass, L., Weber, I. i Zhu, L. (2015). *DevOps: A software architect's perspective*. Addison-Wesley Professional.

CISA. (2023). *How to Proactively use an SBOM for Vulnerability Monitoring*. Cybersecurity and Infrastructure Security Agency. https://www.cisa.gov/sites/default/files/2023-02/How%20to%20Proactively%20use%20an%20SBOM%20for%20Vulnerability%20Monitoring%20508c.pdf

European Parliament and Council. (2022, 27 grudnia). Directive (EU) 2022/2555 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS2 Directive). *Official Journal of the European Union*, L 333, 80–152.

Forward Security. (2020). *SAST, SCA, DAST, IAST, RASP: What They Are and How You Can Automate Application Security*. Forward Security. https://forwardsecurity.com/sast-sca-dast-iast-rasp-what-they-are-and-how-you-can-automate-application-security/

Gartner. (2021). *Magic Quadrant for Application Security Testing, May 2021*. Gartner Research.

Hashizume, K., Rosado, D.G., Fernández-Medina, E., & Fernandez, E.B. (2013). An analysis of security issues for cloud computing. *Future Generation Computer Systems, 29*(3), 1221–1231. https://doi.org/10.1016/j.future.2012.06.006

Leppanen, T., Honkaranta, A. i Costin, A. (2022). Trends for the DevOps Security: A Systematic Literature Review. W: B. Shishkov (red.), Business Modeling and Software Design (s. 200–217). Springer International Publishing. https://doi.org/10.1007/978-3-031-11510-3_12

Mahmood, R. i Mahmoud, Q.H. (2018). *Evaluation of Static Analysis Tools for Finding Vulnerabilities in Java and C/C++ Source Code*. arXiv preprint arXiv:1805.09040. https://arxiv.org/abs/1805.09040

Makani, S.T. i Jangampeta, S. (2021). DevOps Security Tools Evaluating Effectiveness in Detecting and Fixing Security Holes. *International Journal of DevOps (IJDO), 1*(2), 1–12. https://iaeme.com/Home/issue/IJDO?Volume=1&Issue=2

Mohan, V. i Othmane, L.B. (2016). SecDevOps: Is it a marketing buzzword? Mapping research on security in DevOps. In *Proceedings of the International Conference on Availability, Reliability and Security (ARES)* (s. 542–547). IEEE. https://doi.org/10.1109/ARES.2016.10

Myrbakken, H. i Colomo-Palacios, R. (2017). DevSecOps: A multivocal literature review. W: R.V. O'Connor, R. Conboy, R.V.B. Machado, M.M.A. Barry, & B.K.O. Conchúir (red.), *Software Process Improvement and Capability Determination* (s. 17–29). Communications in Computer and Information Science, 770. Springer. https://doi.org/10.1007/978-3-319-67383-7_2

OWASP. (2021). *OWASP Top Ten 2021*. Open Web Application Security Project. https://owasp.org/Top10/

OWASP. (2023). *Interactive Application Security Testing (IAST) — OWASP DevSecOps Guideline*. Open Web Application Security Project. https://owasp.org/www-project-devsecops--guideline/latest/02c-Interactive-Application-Security-Testing

Rajapakse, R.N., Zahedi, M., Babar, M.A. i Shen, H. (2022). Challenges and solutions when adopting DevSecOps: A systematic review. *Information and Software Technology*, *141*, 106700. https://doi.org/10.1016/j.infsof.2021.106700

Rittinghouse, J.W. i Ransome, J.F. (2010). *Cloud computing: Implementation, management, and security*. CRC Press.

Tøndel, I.A., Line, M.B. i Jaatun, M.G. (2015). Current practices and challenges in industrial control organizations regarding information security incident management — Does size matter? *International Journal of Critical Infrastructure Protection*, 12, 1–11. https://doi.org/10.1016/j.ijcip.2015.12.003

U.S. Senate Committee on Homeland Security and Governmental Affairs. (2018). *How Equifax neglected cybersecurity and suffered a devastating data breach*. United States Senate Report. https://www.hsgac.senate.gov/wp-content/uploads/imo/media/doc/FINAL%20Equifax%20Report.pdf

Implementation of DevSecOps Practices in Critical Cloud-based Infrastructure: An Analysis of the Safety of CI/CD Processes

Abstract. This article seeks to analyze the implementation of DevSecOps practices in cloud-based environments, particularly those connected with critical infrastructure, in the context of escalating cybersecurity threats. The analysis and assessment of the effectiveness of security testing tools, such as SAST, DAST, SCA, IAST, in the processes of developing, testing and deploying software (CI/CD) is based on questionnaire data collected from experts responsible for implementing DevSecOps practices in various sectors of critical infrastructure. Results of the survey confirm that the introduction of DevSecOps practices in cloud environments for critical infrastructure, including the automation of security testing in CI/CD pipelines, contributes to reducing the number of vulnerabilities detected at the production stage and help to shorten times of response to security incidents, while enabling developers to maintain continuous software delivery. The implementation of DevSecOps practices is a key way of making critical infrastructure in cloud environments more secure. An effective cybersecurity strategy requires a systematic approach to integrating security mechanisms throughout the software development lifecycle, the use of automated security testing tools, and continuous monitoring of all software components (SBOM). The challenge, however, consists in striking the right balance between security requirements and the need for rapid software delivery.

Keywords: DevSecOps, cloud security, critical infrastructure, SAST, DAST, SCA, IAST, SBOM, CI/CD