PENGEMBANGAN EKOSISTEM *STATIC APPLICATION SECURITY TESTING* UNTUK MENGIDENTIFIKASI KERENTANAN KEAMANAN SUMBER KODE APLIKASI MENGGUNAKAN PENDEKATAN DEVSECOPS

[1]Fajar Syahruddin, [2]Dedy Hariyadi

INTISARI

Latar Belakang: Pengembangan perangkat lunak konvensional sering mengabaikan keamanan dalam *Software Development Life Cycle* (SDLC), menyebabkan aplikasi rentan terhadap serangan seperti injeksi dan *cross-site scripting* (XSS). Penelitian ini mengembangkan ekosistem DevSecOps dengan mengintegrasikan *Secure Software Development Life Cycle* (SSDLC).

Tujuan: Mengintegrasikan *Continuous Integration/Continuous Deployment* (CI/CD) untuk mengidentifikasi kerentanan kode sumber aplikasi *website* menggunakan *Static Application Security Testing* (SAST) dengan SonarQube, serta mengevaluasi hasilnya dalam ekosistem Docker.

Metode Penelitian: Mengadopsi pendekatan SSDLC, Dalam tahap identifikasi dan analisis, *Vulnerability Assessment* (VA) menggunakan reNgine mengidentifikasi kelemahan eksternal, sementara SAST dengan SonarQube memindai kode sumber secara statis untuk kelemahan keamanan *internal* (*insecure coding practices*, *hardcoded credentials*). Alat analisis keamanan diintegrasikan ke dalam *pipeline* Jenkins.

Hasil: Pengujian awal VA reNgine mendeteksi temuan informasional dan potensi "*Credentials Disclosure Check*" berstatus *UNKNOWN*. Analisis SAST dengan SonarQube mengidentifikasi total 23 kerentanan kode (Tinggi, Sedang, Rendah).

Kesimpulan: Seluruh kerentanan berhasil direduksi dan dieliminasi setelah implementasi perbaikan kode. Pemindaian ulang menunjukkan VA reNgine berstatus "*Closed*" dan SAST SonarQube melaporkan semua 23 kerentanan kode telah "*Fixed*". Pendekatan DevSecOps dan tujuan SSDLC untuk deteksi dini serta mitigasi kelemahan keamanan pada aplikasi web telah tercapai.

Kata-kunci: DevSecOps, Static Application Security Testing (SAST), Secure Software Development Life Cycle (SSDLC), Kerentanan, SonarQube, Jenkins, Docker.

[1]Mahasiswa Teknologi Informasi (S-1) Fakultas Teknik dan Teknologi Informasi Universitas Jenderal Achmad Yani Yogyakarta

[2]Dosen Teknologi Informasi (S-1) Fakultas Teknik dan Teknologi Informasi Universitas Jenderal Achmad Yani Yogyakarta

ECOSYSTEM DEVELOPMENT STATIC APPLICATION SECURITY TESTING TO IDENTIFY APPLICATION CODE SOURCE SECURITY VULNERABILITIES USING THE DEVSECOPS APPROACH

[1]Fajar Syahruddin, [2]Dedy Hariyadi

ABSTRACT

Background: *Conventional software development often ignores security in the Software Development Life Cycle (SDLC), leaving applications vulnerable to attacks such as injection and cross-site scripting (XSS). This research develops the DevSecOps ecosystem by integrating the Secure Software Development Life Cycle (SSDLC).*

Objective: *Integrate Continuous Integration/Continuous Deployment (CI/CD) to identify vulnerabilities in website application source code using Static Application Security Testing (SAST) with SonarQube, as well as evaluate the results in the Docker ecosystem.*

Research Method: *Adopting the SSDLC approach, In the identification and analysis stage, Vulnerability Assessment (VA) using reNgine identifies external vulnerabilities, while SAST with SonarQube statically scans the source code for internal security weaknesses (insecure coding practices, hardcoded credentials). Security analysis tools are integrated into the Jenkins pipeline.*

Results: *Initial VA testing of reNgine detected informational findings and potential "Credentials Disclosure Check" status as UNKNOWN. SAST analysis with SonarQube identified a total of 23 code vulnerabilities (High, Medium, Low).*

Conclusion: *All vulnerabilities were successfully mitigated and eliminated after the implementation of code fixes. A re-scan showed the reNgine VA was in the status of "Closed" and SonarQube's SAST reported that all 23 code vulnerabilities had been "Fixed". The DevSecOps approach and SSDLC's goals for early detection and mitigation of security vulnerabilities in web applications have been achieved.*

Keywords*: DevSecOps, Static Application Security Testing (SAST), Secure Software Development Life Cycle (SSDLC), Vulnerability, SonarQube, Jenkins, Docker*

---

[1]*Information Technology Student (S-1) Faculty of Engineering and Information Technology, Jenderal Achmad Yani University Yogyakarta*

[2]*Lecturer of Information Technology (S-1) Faculty of Engineering and Information Technology, Jenderal Achmad Yani University, Yogyakarta*