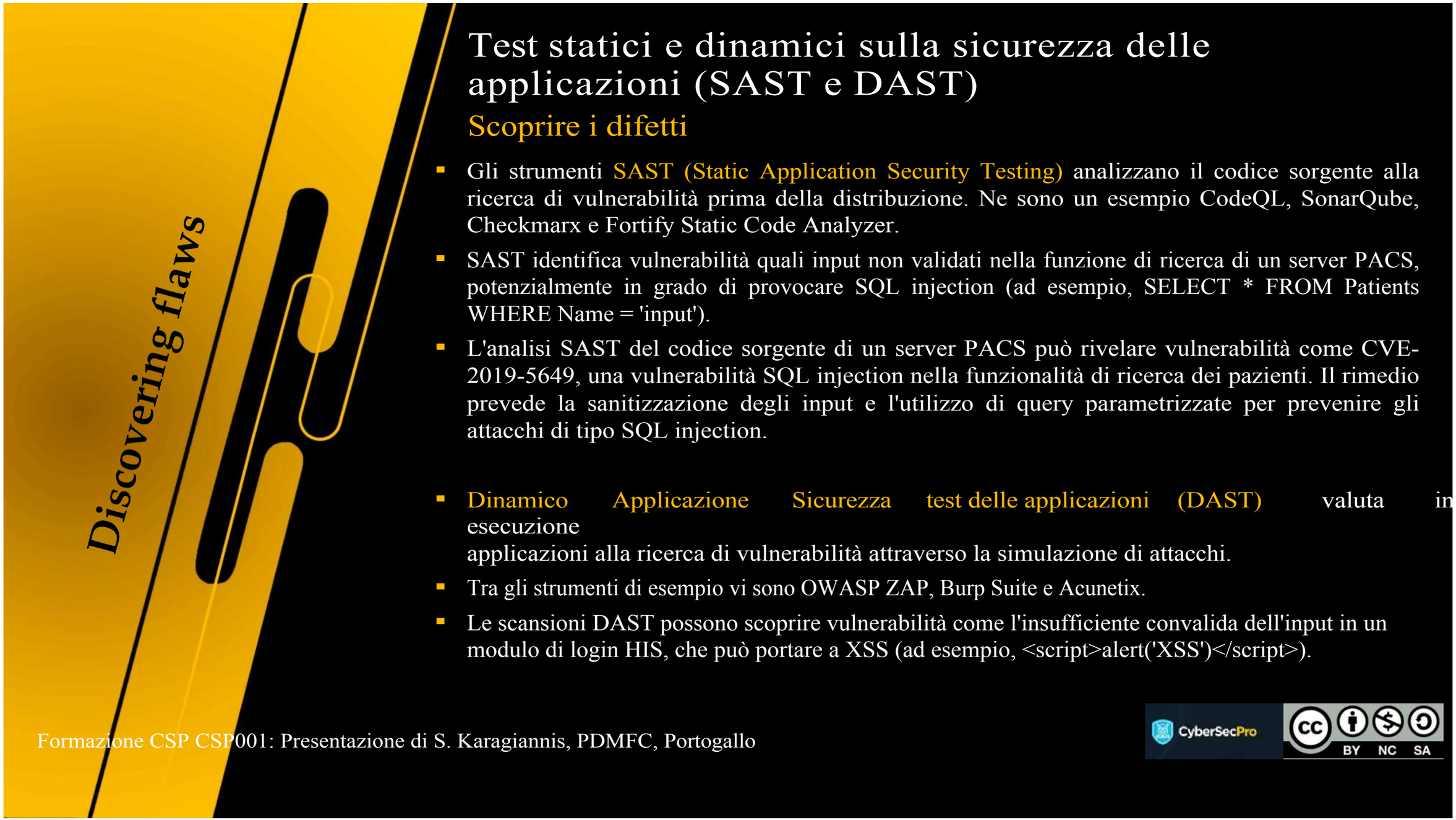
Sviluppo di software sanitario sicuro

CSP009

PRESENTAZIONE DA PARTE DI: STYLIANOS KARAGIANNIS (PDMFC, PORTOGALLO)



Test statici e dinamici sulla sicurezza delle applicazioni (SAST e DAST)

Scoprire i difetti

- Gli strumenti SAST (Static Application Security Testing) analizzano il codice sorgente alla ricerca di vulnerabilità prima della distribuzione. Ne sono un esempio CodeQL, SonarQube, Checkmarx e Fortify Static Code Analyzer.
- SAST identifica vulnerabilità quali input non validati nella funzione di ricerca di un server PACS, potenzialmente in grado di provocare SQL injection (ad esempio, SELECT * FROM Patients WHERE Name = 'input').
- L'analisi SAST del codice sorgente di un server PACS può rivelare vulnerabilità come CVE-2019-5649, una vulnerabilità SQL injection nella funzionalità di ricerca dei pazienti. Il rimedio prevede la sanitizzazione degli input e l'utilizzo di query parametrizzate per prevenire gli attacchi di tipo SQL injection.

- Dinamico Applicazione Sicurezza test delle applicazioni (DAST) valuta in esecuzione
applicazioni alla ricerca di vulnerabilità attraverso la simulazione di attacchi.
- Tra gli strumenti di esempio vi sono OWASP ZAP, Burp Suite e Acunetix.
- Le scansioni DAST possono scoprire vulnerabilità come l'insufficiente convalida dell'input in un modulo di login HIS, che può portare a XSS (ad esempio, <script>alert('XSS')</script>).

GitHub Code Scanning

SAST in pratica Pt.1

Scansione del codice GitHub

GitHub Code Scanning è uno strumento SAST integrato fornito da GitHub, che offre una perfetta integrazione con il flusso di lavoro di sviluppo.

Procedura

- Abilitare la scansione del codice GitHub nelle impostazioni del repository.
- Configurare il flusso di lavoro di scansione per attivare automaticamente le scansioni su push di codice o richieste di pull.
- Personalizzate le impostazioni di analisi e specificate le regole di sicurezza in base requisiti delle applicazioni sanitarie.
- Esaminare i risultati della scansione all'interno dell'interfaccia GitHub, comprese le vulnerabilità identificate, i livelli di gravità e le posizioni del codice interessato.
- Collaborare con i membri del team per affrontare i problemi direttamente all'interno del pull.
  richieste o problemi.

SAST in pratica Pt.2

Scansione del codice GitHub

Vantaggi

- L'integrazione con i repository GitHub snellisce il processo di sviluppo, consentendo agli sviluppatori di affrontare i problemi di sicurezza insieme alle modifiche al codice.
- Fornisce fattibile intuizioni in vulnerabilità, consentendosviluppatori di stabilire le priorità e di risolvere i problemi in modo efficiente.
- Supporta un'ampia gamma di linguaggi di programmazione e framework comunemente utilizzati nello sviluppo di software sanitario.

Procedura

- Dopo una richiesta di push o pull di codice, GitHub Code Scanning attiva automaticamente una scansione della base di codice. La scansione identifica vulnerabilità come SQL injection, XSS o metodi di autenticazione non sicuri all'interno del codice sorgente.
- Le vulnerabilità rilevate vengono segnalate all'interno dell'interfaccia GitHub, consentendo agli sviluppatori di esaminarle e risolverle tempestivamente.
- Gli sviluppatori collaborano per rimediare alle vulnerabilità apportando modifiche al codice. e l'invio di richieste di pull per la revisione.
- Una volta mitigate le vulnerabilità, le modifiche al codice vengono reinserite nel ramo principale, garantendo una distribuzione sicura del software.

DAST in pratica Pt.1

Scoprire i difetti

OWASP ZAP (Zed Attack Proxy)

- OWASP ZAP è uno scanner di sicurezza per applicazioni web open source utilizzato per identificare le vulnerabilità delle applicazioni web.

Procedura

- Configurare OWASP ZAP in modo che agisca come proxy tra il browser e l'applicazione applicazione target.
- Impostate scansioni automatiche o esplorate manualmente l'applicazione per identificare le vulnerabilità.
- Analizzare scansione risultatie generare rapporti che descrivono in dettaglio vulnerabilità identificate.

Vantaggi

- Fornisce un'interfaccia facile da usare sia per i test automatizzati che per quelli manuali.
- Offerte avanzate caratteristiche per l'autenticazione test e sessione gestione.

DAST in pratica Pt.2

DAST in pratica Pt.2

Suite di rutti

Burp Suite è uno strumento commerciale DAST ampiamente utilizzato per le applicazioni web.
test di sicurezza. La procedura

- Configurare la suite Burp per intercettare e analizzare le richieste HTTP e risposte.
- Eseguire scansioni automatiche o esplorare manualmente l'applicazione per identificare le vulnerabilità.
- Utilizzare gli strumenti integrati per la conferma e lo sfruttamento delle vulnerabilità.

Vantaggi

- Offre un'ampia copertura delle vulnerabilità web, compresa la OWASP Top 10.
- Offre funzioni avanzate per i test di penetrazione e lo sfruttamento delle vulnerabilità.

CWE and Code Scanning

Enumerazione delle debolezze comuni (CWE)

CWE e scansione del codice

Common Weakness Enumeration (CWE) è un elenco di punti deboli di software e hardware sviluppato dalla comunità. Il CWE fornisce un linguaggio comune per descrivere le vulnerabilità e i punti deboli dei sistemi software.

- Ogni voce CWE descrive un tipo specifico di debolezza, come l'overflow del buffer, l'iniezione SQL o il cross-site scripting (XSS). Le voci CWE includono descrizioni dettagliate, esempi, potenziali conseguenze e strategie di mitigazione per affrontare la debolezza. Gli strumenti di scansione del codice, come GitHub Code Scanning, utilizzano spesso CWE come riferimento per identificare e classificare le vulnerabilità nel codice sorgente.
- Grazie alla mappatura delle vulnerabilità identificate con le voci CWE, gli strumenti di scansione del codice aiutano gli sviluppatori a comprendere la natura e la gravità dei problemi di sicurezza presenti nelle loro basi di codice. L'integrazione di CWE consente agli sviluppatori di dare priorità agli interventi di correzione in base alla gravità e all'impatto delle debolezze identificate.
- I rapporti di scansione del codice includono in genere gli identificatori CWE insieme alle vulnerabilità rilevate, fornendo agli sviluppatori informazioni utili per migliorare la sicurezza del codice.

Code Scanning Pt.3

Opzioni GitHub Pt.1

Sicurezza

GitHub è in grado di identificare la percentuale di diversi linguaggi di programmazione utilizzati in un repository. GitHub analizza il codice all'interno di un repository e fornisce informazioni sulla percentuale di ciascun linguaggio di programmazione utilizzato.

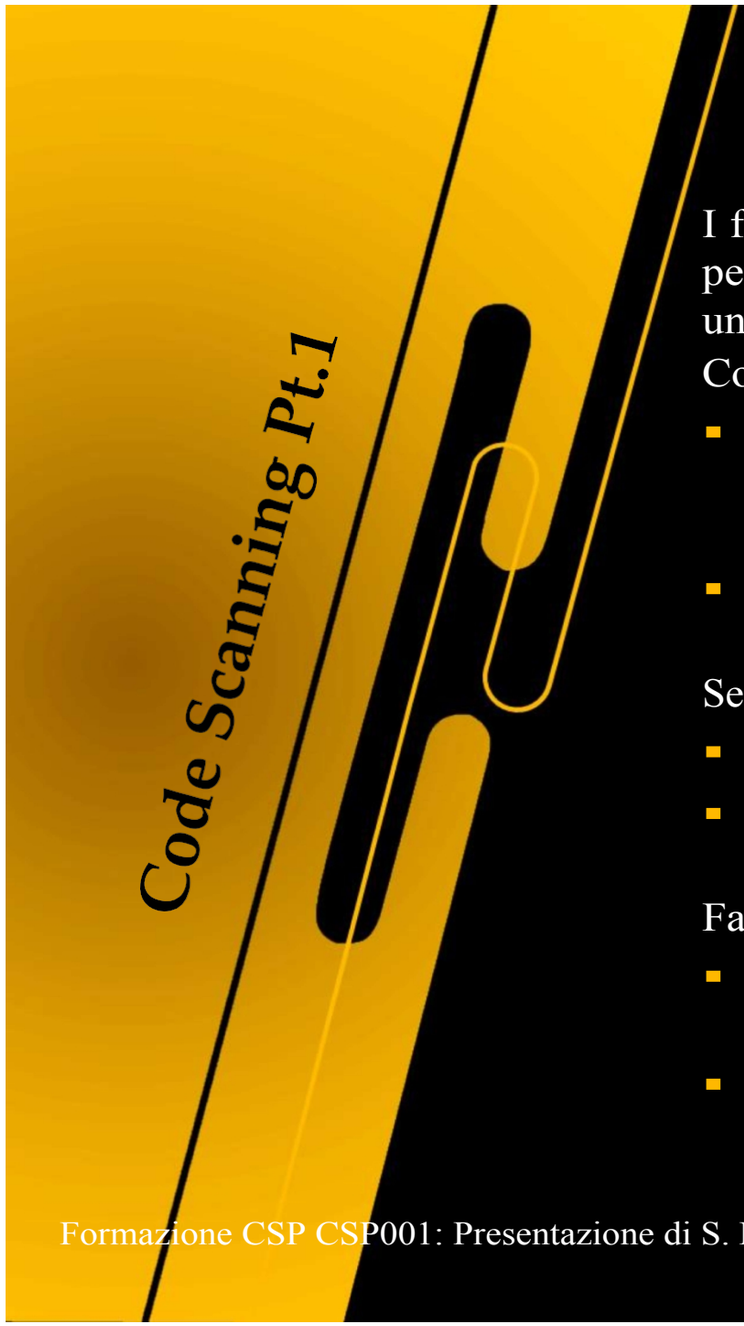
- Politica di sicurezza: Definire come gli utenti devono segnalare le vulnerabilità di sicurezza per questo repository. Abilitando un criterio di sicurezza, i proprietari dei repository possono definire le linee guida per la segnalazione delle vulnerabilità di sicurezza, migliorando la collaborazione e garantendo risposte tempestive alle potenziali minacce.
- Avvisi di sicurezza: Visualizza o divulga gli avvisi di sicurezza per questo repository. GitHub consente di divulgare gli avvisi di sicurezza relativi al repository, garantendo la trasparenza e consentendo agli utenti di rimanere informati sui potenziali rischi per la sicurezza.
- Segnalazione privata delle vulnerabilità: Consentire agli utenti di segnalare privatamente potenziali vulnerabilità di sicurezza. L'abilitazione della segnalazione privata delle vulnerabilità consente agli utenti di segnalare in modo riservato i problemi di sicurezza, favorendo un ambiente sicuro per la collaborazione e la rapida risoluzione delle vulnerabilità.

Code Scanning Pt.3

Opzioni GitHub Pt.2

Sicurezza

- Avvisi di Dependabot: Ricevere una notifica quando una delle vostre dipendenze presenta una vulnerabilità. Gli avvisi di Dependabot notificano ai proprietari dei repository le vulnerabilità presenti nelle dipendenze, aiutandoli a rimanere proattivi nella gestione e nell'aggiornamento delle dipendenze per mitigare i potenziali rischi per la sicurezza.
- Avvisi di scansione del codice: Rileva automaticamente le vulnerabilità comuni e gli errori di codifica. La funzione di scansione del codice di GitHub rileva automaticamente le vulnerabilità comuni e gli errori di codifica nella base di codice del repository, consentendo agli sviluppatori di identificare e affrontare le potenziali minacce alla sicurezza fin dalle prime fasi del processo di sviluppo.
- Avvisi di scansione dei segreti: Ricevere una notifica quando un segreto viene inviato a questo repository. Gli avvisi di scansione dei segreti notificano ai proprietari dei repository quando informazioni sensibili, come token di accesso o credenziali, vengono accidentalmente inviate al repository, aiutando a prevenire accessi non autorizzati e violazioni dei dati.



Scansione del codice Pt.1

Scansione del codice GitHub - Parte 1

I flussi di lavoro per la scansione del codice in GitHub forniscono processi automatizzati per integrare l'analisi statica del codice nel ciclo di vita dello sviluppo del software. Ecco una descrizione in pillole:

Configurazione:

- Definire i flussi di lavoro di scansione del codice utilizzando file di configurazione YAML (ad es, .github/workflows/code-scanning.yml).
- Specificare le condizioni di attivazione delle scansioni del codice, ad esempio su push, creazione di richieste di pull o su base programmata.

Selezione degli utensili:

- Scegliere gli strumenti di scansione dei codici da utilizzare nel flusso di lavoro.
- GitHub Code Scanning si integra nativamente con i repository GitHub, semplificando l'impostazione e la configurazione.

Fasi del flusso di lavoro:

- Definire le fasi del flusso di lavoro per eseguire le scansioni del codice utilizzando gli strumenti selezionati.
- Configurare parametri quali la frequenza di scansione, la profondità di scansione e i criteri di scansione.

Scansione del codice Pt.2

Scansione del codice GitHub - Parte 2

Esecuzione della scansione:

- Automatizzato codice scansioni sono attivate in base su definite condizioni del flusso di lavoro.
- Le scansioni analizzano il codice sorgente per vulnerabilità di sicurezza , la codifica
  errori e altri problemi.

Analisi dei risultati:

- Al completamento, i risultati della scansione sono generati e resi disponibili . e resi disponibili
  all'interno dell'interfaccia GitHub.
- Gli sviluppatori possono accedere a dettagliato rapporti che evidenziano vulnerabilità identificate, i loro livelli di gravità e le posizioni del codice interessato.

Notifica e revisione:

- Facoltativamente, configurare le notifiche per avvisare gli sviluppatori e le relative gli stakeholder sui risultati delle scansioni.
- Recensione scansione risultati all'interno di richieste di richieste, problemi, o dashboard dedicate, a seconda della configurazione del flusso di lavoro scelta.

Scansione del codice Pt.3

Risoluzione

Bonifica e mitigazione:

- Collaborare con membri del team membri per affrontare identificate vulnerabilità individuate.
- Gli sviluppatori possono assegnare priorità e risolvere i problemi direttamente all'interno di GitHub.
  interfaccia, sfruttando le richieste di pull o i sistemi di tracciamento dei problemi.

Miglioramento continuo:

- Iterare su codice scansionare flussi di lavoro basati su feedback e evolvere requisiti del progetto.
- Regolare le politiche di scansione, le configurazioni degli strumenti e i punti di integrazione per ottimizzare le pratiche di sicurezza del codice.

Integrazione con le pipeline CI/CD:

- Senza soluzione di continuità integrare codice scansione flussi di lavoro con esistenti CI/CD
  pipeline per la validazione completa del codice e l'automazione della distribuzione.
- Garantire che i risultati della scansione del codice informino il processo di rilascio del software, impedendo l'introduzione di vulnerabilità negli ambienti di produzione.

CodeQL and Workflows

Flussi di lavoro GitHub nella scansione di sicurezza

Definizione

I flussi di lavoro di GitHub nella scansione della sicurezza si riferiscono a processi o sequenze di azioni automatizzate innescate da eventi in un repository GitHub, con l'obiettivo di migliorare la sicurezza rilevando vulnerabilità, errori di codifica o fughe di informazioni sensibili nella base di codice.

- I flussi di lavoro di GitHub semplificano le pratiche di sicurezza automatizzando i controlli di sicurezza, garantendo la qualità del codice e prevenendo le violazioni della sicurezza prima della distribuzione.
- I flussi di lavoro di GitHub possono essere configurati per eseguire vari strumenti di scansione della sicurezza, come CodeQL, per analizzare il codice alla ricerca di potenziali vulnerabilità, difetti di sicurezza o problemi di conformità.
- CodeQL è un potente strumento di analisi statica fornito da GitHub per identificare le vulnerabilità di sicurezza e gli errori di codifica nelle basi di codice. Consente agli sviluppatori di scrivere query personalizzate per analizzare i modelli di codice e individuare potenziali rischi per la sicurezza.

Scansione del codice con GitHub

Informazioni

- Scansione continua: GitHub Code Scanning fornisce un'analisi continua del codice alla ricerca di vulnerabilità, garantendo il rilevamento proattivo e la mitigazione dei rischi per la sicurezza.
- La scansione del codice classifica le vulnerabilità in diversi livelli di gravità. Identificazione del file sorgente e della linea di codice in cui sono state trovate le potenziali vulnerabilità.
- Scansioni programmate: Gli utenti possono programmare la frequenza delle scansioni in base ai requisiti del progetto, consentendo controlli di sicurezza regolari e tempestivi sulla base di codice.
- Dettagli: Spiegazione estesa di il problema della sicurezza , aiutando in comprensione e risoluzione.
- Corrispondenza CWE: le vulnerabilità vengono abbinate agli standard CWE (Common Weakness Enumeration), fornendo una classificazione standardizzata delle debolezze del software.

Grazie

Presentatore: Stylianos Karagiannis (PDMFC, Portogallo)

Si prega di inviare tutte le domande a: stylianos.karagiannis@pdmfc.com