APPENDICE 2. VALUTAZIONE STRUMENTI

a. CHECKMARX

PRODOTTO	CATEGORIA	FASE SSE	SITO WEB	
CxSAST	**SAST**	Implementation	https://www.checkmarx.com/	
DESCRIZIONE				
È un tool commerciale, per l'analisi statica del codice, posizionato da Gartner nel quadrante Leaders nell'ambito dell'Application Security Testing (AST). Supporta numerosi linguaggi (vedi oltre). Può essere integrato a vari livelli nell'ambito della fase di implementation: IDE, build server, bug tracking tools.				
Tainted analysis, Pattern matching, "scan rules" (customizable)				
ANALISI DEL VALUTATORE				**SCORE**
Livello di integrazione con i seguenti prodotti				
a. IDEs	Esistono plugin per i seguenti IDE: Eclipse, Visual Studio e IntelliJ. I plugin consentono la scansione del codice, l'analisi e la navigazione dei risultati in modo integrato con l'IDE.			7
b. source repository,	TFS, SVN, GIT, Perforce.			7
c. build server,	Jenkins, Bamboo, TeamCity, TFS, Anthill Pro, Maven.			7
d. bug tracking tools	Jira.			5
I linguaggi di programmazione supportati	C#, JavaScript and commonly used frameworks, Node.JS and commonly used frameworks, VB.NET, ASP, VB6, PHP, C/C++, Apex and VisualForce, Ruby, VBScript, Perl, HTML5, Python, Groovy, Scala, PL/SQL, JSP, Typescript, Go, Windows Mobile .NET/.NET Core			8

I framework applicativi supportati (es. Spring, Hibernate, …)	**[*] Requires minor adjustments** **Platform/Enviroment: Java** Struts, Spring MVC, iBatis*, GWT, Hibernate, OWASP ESAPI, JSTL FMT Taglib, ATG DSP Taglib, Java Server Faces (JSF), JavaScript **Platform/Enviroment: .NET** Enterprise Libraries, Telerik, ComponentArt, Infragistics, FarPoint, iBatis*, Hibernate.Net [*], Entity framework up to 4.3.1 **Platform/Enviroment: PHP** Zend, Kohana, CakePHP, Symfony, Smarty, OWASP ESAPI **Platform/Enviroment: C/C++** MISRA **Platform/Enviroment: Ruby** Ruby on Rails **Platform/Enviroment: JavaScript** JQuery, Node.js, Ajax, Knockout, AngularJS, ExpressJS, Jade, Backbone, Handlebars, Hapi.JS **Platform/Enviroment: iOS** iOS mobile applications **Platform/Enviroment: Python** Django **Platform/Enviroment: Groovy** Grails	7
Le tipologie di applicazione supportate (Web, Mobile, Client-Server…)	Web application, Mobile, Client-Server.	7
Le vulnerabilità riconosciute (Sql injection, Cross-site scripting, Code injection…)	SQL Injection, Cross-site scripting, Code injection, Buffer Overflow, Parameter tampering, Cross-site request forgery, XXE injection, Unsecure deserializarion, HTTP splitting, Log forgery, DoS, Session Fixation, Session poisoning, path traversal, Unhandled exceptions, Unreleased resources, unvalidated input, URL redirection attack, Dangerous Files Upload, Hardcoded password.	7
Gli Standard supportati (OWASP Top 10, SANS 25, …)	OSWAP Top 10, OSWAP Mobile Top 10, SANS 25, HIPAA, FISMA, BSIMM, PCI DSS, Mitre CWE, MISRA.	7
L'integrazione di "Custom rules"	È possibile definire delle regole personalizzate.	4

L'incidenza dei "Falsi positivi"	In primo luogo, è possibile "spegnere" falsi positivi estendendo la lista dei "sanitizer" fornita out of the box da checkmarx (con pochi colpi di click). In secondo luogo, è possibile "spegnere" falsi positivi dichiarandoli come "Not Exploitable". In terzo luogo, è stato possibile apprezzare un approccio messo in atto da Checkmarx atto a limitare il numero di segnalazioni. La prova eseguita ha evidenziato che: in presenza di codice evidentemente prono a una SQL INJECTION, ma in assenza di un vettore di attacco, la segnalazione della vulnerabilità viene soppressa. Viceversa la segnalazione viene prodotta se viene individuato anche un vettore di attacco. Il side effect è che in una scansione parziale che considera il codice vulnerabile ma esclude in tutto o in parte il vettore d'attacco, non vengono prodotte segnalazioni.	4
La capacità di analisi "raw source code" vs "need to compile"	Lo strumento è in grado di funzionare in modalità "raw source code". È quindi possibile sottoporre anche porzioni di codice "out-of-context". Tuttavia, in questo caso potrebbero non essere segnalate certe vulnerablità che invece si manifestano in una scansione "in-context". È una scelta by design per limitare falsi positivi.	Raw Source
La capacità di analizzare le dipendenze da librerie esterne al fine di controllare se sono presenti vulnerabilità note	Questa funzionalità non è compresa fra quelle standard del prodotto. Esiste un add-on di CheckMarx (acquistabile a parte) che analizza le dipendenze da librerie esterne al fine di controllare se sono presenti vulnerabilità note, interrogando una base dati esterna.	1
La capacità di correlare lo scan statico con l'esito di uno scan dinamico (correlazione White Box con Black Box)	CxSAST non possiede questa funzionalità.	1
LE PERFORMANCE		
a. Full scan vs Incremental scan	Sono supportati sia Full sia Incremental scanning.	7
b. Client-side scan vs Server-side scan	Server-side scanning: i sorgenti vengono compressi e inviati al server dove vengono decompressi e riconosciuti, dopodichè avviene effettivamente lo scan. L'elaborazione è sempre centrale. Se più scansioni sono ordinate contemporaneamente, i lavori vengono accodati.	7
Eventuali funzionalità di prioritizzazione delle remediation	Le vulnerabilità individuate vengono ordinate secondo 4 livelli: High, Medium, Low, Information che indirizzano la priorità della remediation.	7
La facilità d'uso	Lo strumento è fortemente orientato alla facilità. Alla prova dei fatti, lo strumento è davvero molto user friendly e intuitivo.	7

I costi di licenza	Esistono varie forme di licenza. In generale i criteri per stabilire il costo della licenza sono: il numero di progetti, le linee di codice e il numero di sviluppatori. Il prezzo è stabilito attraverso una trattativa commerciale.	Medio /Alto
Il supporto alla reportistica	E' supportata una reportistica di tipo custom (non sono espressamente disponibili report pre-definiti, ad esempio specificamente orientati a CWE SANS Top 25, OWASP Top 10, PCI Data Security Standard, ecc). I formati supportati sono: PDF, CSV, RTF, XML.	4
La classificazione degli errori riportati	Sono riferiti agli standard supportati (es. "PCI DSS (3.1) - 6.5.1 - Injection flaws - particularly SQL injection", OWASP Top 10 2013 - A1-Injection).	7

CONSIDERAZIONI GENERALI

Considerazioni generali:

- L'installazione risulta agevole.
- La dashboard di gestione è semplice e intuitiva.
- Apprezzabile il riconoscimento automatico del linguaggio: è sufficiente eseguire lo zip dei sorgenti e farne l'upload verso il server.
- Agevole utilizzare il plug-in integrato con un IDE (tasto destro su un punto del progetto per eseguire la scansione)
- Supporto alla remediation in tutti gli ambienti: CxAudit, plug-in, browser
- Inserimento di regole custom agevole (esaminato il caso "sanitizer")
- Reportistica completa e flessibile in diversi formati.
- È possibile effettuare una scansione piena (iniziale) e una scansione incrementale (successiva alla prima).
- Il software caricato per la scansione non deve essere compilato
- Non è prevista la funzionalità di controllo delle vulnerabilità delle librerie utilizzate dal progetto, a meno di integrare un componente licenziato a parte.
- Integrazione con Jenkins, come step aggiuntivo della fase di build (Continuous Integration), agevole attraverso plug-in

Punti di forza:

- Vettore di attacco
- Funzionalità "Full Graph" che raccorda più vettori di attacco mostrando eventuali punti di intersezione (candidati ideali per il fix)

APPROCCIO PER LA VALUTAZIONE