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

Nei test di sicurezza delle applicazioni, i "falsi positivi" da soli non determinano la piena precisione, sebbene la loro bassa incidenza sia spesso considerata l'indicatore più importante che rivela la bontà del tool in esame. I falsi positivi sono solo uno dei quattro aspetti che determinano l'accuratezza di uno strumento: gli altri tre sono i "veri positivi", i "veri negativi" e i "falsi negativi". Falsi Positivi (FP): false vulnerabilità che sono non ci sono. Veri Positivi (TP): vulnerabilità reali segnalate correttamente. Falsi negativi (FN): vulnerabilità reali che non sono state correttamente segnalate. Veri negativi (TN): false vulnerabilità che correttamente non sono state segnalate. Pertanto, il tasso dei veri positivi (TPR) è il tasso con il quale sono state segnalate correttamente le vulnerabilità reali. Il tasso di falsi positivi (FPR) è il tasso con cui le vulnerabilità false sono state segnalate come reali, in modo errato. Le formule per determinare i veri e i falsi positivi: • Tasso dei veri positivi (TPR) = FP / (FP + TN) • Tasso dei falsi positivi (FPR) = TP / (TP + FN)
CONSIDERAZIONI FINALI DEL VALUTATORE
Nonostante la presenza accertata di falsi positivi e falsi negativi nei risultati delle scansioni, il prodotto si presta a una grande facilità d'uso e a una buona flessibilità, sia nella personalizzazione delle regole, sia nella reportistica. Il prodotto prevede la scansione di molti tipi di linguaggi sviluppati su diverse piattaforme e s'integra nelle pipeline di DevOps. L'interpretazione dei risultati è tuttavia d'obbligo, per valutare l'effettiva presenza delle vulnerabilità segnalate.

TEAM DI VALUTAZIONE	Software Security team

b. CodeDX

PRODOTTO	**CATEGORIA**	**FASE SSE**	**SITO WEB**
CodeDx	**SAST/DAST**	Implementation/Verification	https://codedx.com/
DESCRIZIONE			
CodeDx è un Tool commerciale che serve ad effettuare la verifica di eventuali vulnerabilità di programmi e software presi in considerazione relative al codice sorgente. CodeDx riunisce una serie di strumenti di analisi del codice (sia gratuiti, sia commerciali) che consentono a loro volta di individuare agevolmente eventuali difetti nel codice da analizzare.			
Source analysis, Pattern matching, "scan rules" (customizable).			
ANALISI DEL VALUTATORE			**SCORE**
Livello di integrazione con i seguenti prodotti			
a. IDEs	CodeDx si integra con i seguenti ide: Eclipse, IntelliJ e Visual Studio.		8

b. source repository,	CodeDx si integra i seguenti repository: Git (direttamente); Subversion, Mercurial, o Team Foundation Version Control (TFVC) (tramite zip del "source outside" di CodeDx e successivo upload verso CodeDx).	8
c. build server,	CodeDx si integra con i seguenti build server: Azure DevOps, Jenkins, Maven, TeamCity, Bamboo.	7
d. bug tracking tools	CodeDx supporta AlienVault, Git, Jira Software, Microsoft Threat Modeling, SD Elements.	
Le tipologie di applicazione supportate (Web, Mobile, Client-Server…)	Client Server, Web, Mobile (Android Studio).	7
I linguaggi di programmazione supportati	C/C++, Java, Javascript, JSP, .NET(C#, Visual Basic), PHP, Python, Ruby, Scala.	8
I framework applicativi supportati (es. Spring, Hibernate, …)	Il tool supporta i più popolari frameworks tra i quali Spring-MVC, JQuery e molti altri.	7
Gli Standard supportati (OWASP Top 10, SANS 25, …)	7PK (Seven Pernicious Kingdoms), CERT Coding Standards for C/C++ & Java, CLASP Vulnerability Lexicon, CWE/SANS Top 25 Most Dangerous Software Errors, DISA STIGs version 3.1 and 4.3, HIPAA Compliance Check, MISRA C, Mobile OWASP Top 10, NIST 800-53, OWASP Top 10 Project, PCI DSS, Software Fault Patterns (SFP), WASC Threat Classification v2	9
Le vulnerabilità riconosciute (Sql injection, Cross-site scripting, Code injection…)	Le vulnerabilità riportate dai seguenti tools, direttamente incorporati nel prodotto: Brakeman, Checkstyle, CppCheck, ESLint, SpotBugs, Find Security-Bugs, Gendarme, OWASP Dependency Check, JSHint, PHP_CodeSniffer, PHPMD, PMD, Pylint, Retire.js, ScalaStyle.	8
L'integrazione di "Custom rules"	È possibile all'interno di CodeDx creare delle regole personalizzate.	7
Possibilità di inibire la segnalazione di particolari vulnerabilità	È possibile all'interno del Tool gestire la segnalazione di una particolare vulnerabilità.	7
L'incidenza dei "Falsi positivi"	Dai riscontri, l'incidenza di falsi positivi è accettabile.	8
La capacità di analisi "raw source code" vs "need to compile"	CodeDx (a seconda dei tool embedded che vengono invocati) permette di analizzare il codice in entrambe le modalità (sia source-code che raw-code).	Entrambe
La capacità di analizzare le dipendenze da librerie esterne al fine di controllare se sono presenti vulnerabilità note	Black Duck (by Synopsys), OWASP Dependency Check, Retire.js, Synopsys Protecode, Sonatype Nexus	8

La capacità di correlare lo scan statico con l'esito di uno scan dinamico (correlazione White Box con Black Box)	Il prodotto è in grado di effettuare correlazioni tra entrambe le tipologie di scan del codice.	7
LE PERFORMANCE		
a. Full scan vs Incremental scan	Il prodotto è in grado di effettuare entrambe le tipologie di scan del codice.	8
b. Client-side scan vs Server-side scan	Il prodotto consente di effettuare scan sia lato server che client.	7
Supporto alla Remediation	Il tool guida nella localizzazione del problema ed offre supporto informativo utile per sanarlo.	6
Funzionalità di prioritizzazione delle Remediation	Il tool permete di evidenziare i bugs in base a delle priorità di intervento.	7
La facilità d'uso	Il prodotto è piuttosto facile da installare e assolutamente intuitivo da utilizzare.	8
I costi di licenza	CodeDx è un prodotto commerciale a pagamento dai costi non eccessivi rispetto a strumenti similari commerciali. L'argomento andrebbe comunque analizzato in una logica commerciale complessiva aziendale.	MEDIO
Il supporto alla reportistica	Il tool consente di produrre un'ottima reportistica in vari tipi di formato (Pdf, xml, Excel).	8
La classificazione degli errori riportati	Il Tool CodeDx permette di classificare gli errori secondo quattro tipologie di gravità: High, Medium, Low e Info.	7
CONSIDERAZIONI FINALI DEL VALUTATORE		
Dopo aver preso in considerazione tutti i punti descritti nella scheda si ritiene che il Tool CodeDx sia un ottimo strumento di facile uso e integrabile con molti altri tool sia gratuiti che a pagamento. Il tool permette agli sviluppatori di software, tester e analisti della sicurezza di individuare e gestire con modalità abbastanza semplici le vulnerabilità nel software. Il tool permette di integrare una quantità molto ampia di plugin e di altri tool che danno una copertura estesa di tutti i linguaggi più diffusi e degli IDE. L'integrazione fra i risultati di scansioni di tool differenti e la reportistica molto dettagliata e disponibile in vari formati, sono i veri punti di forza di CodeDx. Dalle evidenze riscontrate, è emerso che i tool ai quali CodeDx si appoggia forniscano risultati per lo più affidabili. Si ritiene pertanto che CodeDx sia utilizzabile proficuamente per gli scopi aziendali.		
TEAM DI VALUTAZIONE	Software Security team	

c. SonarQube

PRODOTTO	CATEGORIA	FASE SSE	SITO WEB
SonarQube	**SAST**	Implementation	http://www.sonarqube.org

DESCRIZIONE		
SonarQube è un prodotto avanzato per l'analisi statica del codice sorgente, finalizzato alla ricerca di errori di programmazione e di costrutti che costituiscono delle bad practise. I Bug rilevati sono tracciati ed evidenziati in un'interfaccia web intuitiva, in modo da poter seguire e gestire il processo di remediation. Dato che si tratta di un prodotto open source, il miglioramento dei pattern per il riconoscimento dei problemi è demandato all'ampia community in rete.		
SonarQube esegue le sue analisi attraverso appositi plugin che applicano al codice sorgente dei pattern match pre-definiti.		
ANALISI DEL VALUTATORE		**SCORE**
Livello di integrazione con i seguenti prodotti		
a. IDEs	S'integra tramite il plugin SonarLint con Eclipse, Visual Studio, IntelliJ. SonarLint è uno strumento che analizza il codice dal punto di vista della qualità, ma è possibile utilizzarlo in collegamento con SonarQube, per sfruttare le regole di sicurezza di quest'ultimo.	8
b. source repository,	S'integra, tramite plugin, a Git, Svn, CVS, TFVC, Jazz RTC, ClearCase.	8
c. build server,		
d. bug tracking tools	SonarQube comprende la gestione completa dei bug riscontrati (tracciamento incluso).	8
Le tipologie di applicazione supportate (Web, Mobile, Client-Server…)	Web, Mobile Android.	8
I linguaggi di programmazione supportati	ABAP, Apex, C#, C, C++, COBOL, CSS, Flex, Go, Java, JavaScript, Kotlin, Objective-C, PHP, PLI, PLSQL, Python, RPG, Ruby, Scala, Swift, TypeScript, TSQL, VB.NET, VB6, HTML, XML	10
I framework applicativi supportati (es. Spring, Hibernate, …)		
Gli Standard supportati (OWASP Top 10, SANS 25, …)	SonarQube comprende fra le sue rules CWE, SANS TOP 25 e OWASP TOP 10	10
L'integrazione di "Custom rules"	SonarQube offre la possibilità di creare delle regole personalizzate, attraverso dei custom templates	10
Possibilità di inibire la segnalazione di particolari vulnerabilità	Il tool consente di "sopprimere" la segnalazione di una particolare vulnerabilità in maniera agevole.	9
L'incidenza dei "Falsi positivi"	Coloro che scoprono un falso positivo possono segnalarlo alla Community. Per questo motivo l'incidenza dei falsi positivi è tenuta bassa.	7
La capacità di analisi "raw source code" vs "need to compile"	SonarQube fa le sue valutazioni su bytecode, per cui presuppone un rebuild del codice modificato.	Need to Compile

La capacità di analizzare le dipendenze da librerie esterne al fine di controllare se sono presenti vulnerabilità note	Attraverso plugin	7
La capacità di correlare lo scan statico con l'esito di uno scan dinamico (correlazione White Box con Black Box)		
LE PERFORMANCE		
a. Full scan vs Incremental scan	Il prodotto è in grado di eseguire entrambe le tipologie di scan del codice.	8
b. Client-side scan vs Server-side scan	Il prodotto consente di eseguire scan sia lato server, sia lato client.	8
Supporto alla Remediation	SonarQube offre la possibilità di organizzare e seguire la fase di correzione dei bugs.	9
Funzionalità di prioritizzazione delle Remediation	SonarQube classifica i bugs in base all'urgenza con la quale devono essere corretti.	8
La facilità d'uso	Il prodotto è piuttosto facile da installare e assolutamente intuitivo da utilizzare.	7
I costi di licenza	La Community edition di SonarQube è Open Source, con licenza GNU Lesser GPL License, Version 3, quindi non comporta alcun costo di licenza. Le edizioni Developer, Enterprise e Data Center sono commerciali.	Free
Il supporto alla reportistica	Si realizza tramite plugin open source o commerciali. La dashboard e l'interfaccia web costituiscono, di per sé, una valida reportistica.	7
CONSIDERAZIONI FINALI DEL VALUTATORE		
Sebbene l'aspetto della sicurezza non sia ancora il core delle funzionalità di SonarQube, sono stati fatti molti passi avanti per migliorare la scoperta delle vulnerabilità insite nella scrittura di codice sorgente. SonarQube ha diversi punti di forza che ne hanno fatto lo strumento preferito dai gruppi di sviluppo per il controllo statico del codice: • Un'estesa community che lavora costantemente al suo miglioramento. • Una grande disponibilità di plugin che ne ampliano le funzionalità, fino a coprire molteplici aspetti dello sviluppo sicuro. • La possibilità di utilizzarlo all'interno di una moderna pipeline di delivery DevOps-oriented, per automatizzare l'efficientamento del codice ad ogni rilascio. • Metriche sofisticate che servono a stabilire complessità e leggibilità del codice e l'adesione alle best practises di programmazione. • La gestione grafica delle vulnerabilità emerse. • L'adesione ai principali standard di sicurezza: CWE, SANS To 25 e OWASP Top 10.		
TEAM DI VALUTAZIONE	Software Security team	