

LINEE GUIDA PER L'ADOZIONE DI UN CICLO DI SVILUPPO DI SOFTWARE SICURO

SOMMARIO

1 INTRODUZIONE	6
1.1 SCOPO	6
1.2 STRUTTURA DEL DOCUMENTO.....	6
2 RIFERIMENTI	7
2.1 DOCUMENTI DI RIFERIMENTO	7
3 DEFINIZIONI E ACRONIMI.....	8
3.1 DEFINIZIONI	8
3.2 ACRONIMI	8
4 ESIGENZE E AMBITI DI APPLICAZIONE	10
4.1 IL PANORAMA DELLE VULNERABILITÀ APPLICATIVE	10
4.2 SVILUPPO APPLICAZIONI SICURE.....	11
4.3 SECURITY TOOLS	14
5 ANALISI DELLE INIZIATIVE E DEGLI STANDARD	18
5.1 INIZIATIVE INTERNAZIONALI	18
5.1.1 <i>Open Web Application Security Project (OWASP)</i>	18
5.1.2 <i>Common Criteria (CC)</i>	20
5.1.3 <i>IEEE Computer Society</i>	21
5.1.4 <i>International Organisation for Standardization (ISO)</i>	22
5.1.5 <i>International Society of Automation (ISA)</i>	24
5.1.6 <i>Software Assurance Forum for Excellence in Code (SAFECODE)</i>	26
5.1.7 <i>SANS Software Security Institute (SANS SSI)</i>	27
5.1.8 <i>Web Application Security Consortium (WASC)</i>	28
5.1.9 <i>Institute For Software Quality (ifSQ)</i>	29
5.2 INIZIATIVE EUROPEE	30
5.2.1 <i>Networked European Software and Services Initiative (NESSI)</i>	30
5.2.2 <i>Piattaforme Nazionali NESSI</i>	31
5.2.3 <i>OWASP Local Chapters</i>	33
5.2.4 <i>Motor Industry Software Reliability Association (MISRA)</i>	36
5.2.5 <i>European Space Agency (ESA)</i>	37
5.3 INIZIATIVE US	38
5.3.1 <i>CERT Secure Coding</i>	38
5.3.2 <i>Software Assurance Metrics and Tool Evaluation (SAMATE)</i>	39
5.3.3 <i>Common Weakness Enumeration (CWE)</i>	41
5.3.4 <i>Common Attack Pattern Enumeration and Classification (CAPEC)</i>	43
6 LA SICUREZZA IN TUTTE LE FASI DEL CICLO DI SVILUPPO DEL SOFTWARE	45
6.1 SECURE SDLC	45
6.2 RISK ASSESSMENT	46
6.2.1 <i>Tool per l'analisi del rischio</i>	49
6.3 REQUISITI	50
6.3.1 <i>Linguaggi per la specifica dei requisiti</i>	50
6.3.2 <i>Tool per la specifica dei requisiti</i>	53
6.4 PROGETTAZIONE	54
6.4.1 <i>Secure Design Languages</i>	54
6.4.2 <i>Software Design Tools</i>	54

6.5	IMPLEMENTAZIONE.....	55
6.5.1	<i>Software Implementation Tools.....</i>	55
6.6	VERIFICA	57
6.6.1	<i>Software Verification Tools.....</i>	57
6.7	VALIDAZIONE.....	61
6.7.1	<i>Software Release Tools.....</i>	62
6.8	SUPPORTO	63
6.8.1	<i>Software Response Tools</i>	63
6.9	CATALOGO SECURITY TOOLS.....	66
6.10	TRAINING E FORMAZIONE.....	66
6.10.1	<i>Secure Coding in C and C++</i>	67
6.10.2	<i>Writing Secure Code - C++</i>	67
6.10.3	<i>Writing Secure Code - Java (J2EE).....</i>	68
6.10.4	<i>Foundstone (Mcafee) Courses</i>	68
6.10.5	<i>Threat Modeling</i>	68
6.10.6	<i>Writing Secure Code - ASP.NET (C#)</i>	69
6.10.7	<i>Oracle Courses</i>	69
6.10.8	<i>Developing Secure Java Web Services, Java EE 6.....</i>	69
6.10.9	<i>MySQL and PHP - Developing Dynamic Web Applications</i>	70
6.10.10	<i>Google Gruyere.....</i>	71
6.10.11	<i>OWASP Training Courses</i>	71
7	CERTIFICAZIONI PROFESSIONALI	72
7.1	GIAC SECURE SOFTWARE PROGRAMMER (GSSP) CERTIFICATION	72
7.2	INTERNATIONAL COUNCIL OF E-COMMERCE CONSULTANTS (EC-COUNCIL) CERTIFICATIONS	72
7.3	CERTIFIED ETHICAL HACKER (CEH)	73
7.4	CERTIFIED SECURITY ANALYST (ECSA).....	73
7.5	CERTIFIED SECURE PROGRAMMER (ECSP)	73
7.6	CERTIFIED SOFTWARE SECURITY LIFECYCLE PROFESSIONAL (CSSLP) AND CERTIFIED INFORMATION SYSTEMS SECURITY PROFESSIONAL (CISSP).....	74
7.7	CERTIFICAZIONI ISACA (CISA, CISM, CRISC)	75
7.8	INTERNATIONAL SECURE SOFTWARE ENGINEERING COUNCIL (ISSECO)	76
8	SECURE SOFTWARE DEVELOPMENT LIFE CYCLE (SSDLC): ANALISI DELLE METODOLOGIE E DEI PROCESSI.....	78
8.1	LIFE CYCLE & Maturity Models	78
8.1.1	<i>Software Assurance Maturity Model (SAMML)</i>	78
8.1.2	<i>Systems Security Engineering Capability Maturity Model (SEE-CMM)</i>	79
8.1.3	<i>Building Security In Maturity Model (BSIMM)</i>	80
8.2	ANALISI DEI PROCESSI SSDLC.....	82
8.2.1	<i>McGraw's Secure Software Development Life Cycle Process</i>	82
8.2.2	<i>Microsoft Software Development Life Cycle (MS SDL)</i>	83
8.2.3	<i>Appropriate and Effective Guidance for Information Security (AEGIS)</i>	84
8.2.4	<i>Secure Software Development Model (SSDM)</i>	85
8.2.5	<i>Aprville and Pourzandi's Secure Software Development Life Cycle Process.....</i>	85
8.2.6	<i>Secure Software Development Model (SecSDM)</i>	86
8.2.7	<i>Software Security Assessment Instrument (SSAI).....</i>	86
8.2.8	<i>Hadawi's Set of Secure Development Activities</i>	86
8.2.9	<i>Comprehensive, Lightweight Application Security Process (CLASP)</i>	87
8.2.10	<i>Secure Software Development Process Model (S2D-ProM)</i>	87
8.2.11	<i>Team Software Process for Secure Software Development (TSP Secure)</i>	87
9	LINEE GUIDA PER L'ADOZIONE DI UN CICLO DI SVILUPPO SOFTWARE SICURO	89
9.1	DEFINIZIONE DEI REQUISITI DI SICUREZZA.....	90

9.1.1	<i>Identificazione degli strumenti a supporto</i>	92
9.2	PROGETTAZIONE DI APPLICAZIONI SICURE	93
9.2.1	<i>Identificazione degli strumenti a supporto</i>	93
9.3	IMPLEMENTAZIONE DI APPLICAZIONI SICURE	94
9.3.1	<i>Identificazione degli strumenti a supporto</i>	94
9.4	VERIFICA DELLA SICUREZZA DELLE APPLICAZIONI	97
9.4.1	<i>Identificazione degli strumenti a supporto</i>	98
9.5	SUPPORTO PER LA MANUTENZIONE DI APPLICAZIONI SICURE	98
9.5.1	<i>Identificazione degli strumenti a supporto</i>	99
10	LINEE GUIDA PER L'IMPLEMENTAZIONE DELLA PRIVACY BY DESIGN NEL SDLC	100
10.1	INTRODUZIONE E CONCETTI BASE	100
10.1.1	<i>Principi della Privacy</i>	100
10.1.2	<i>Obiettivi di protezione</i>	103
10.1.3	<i>Privacy by design</i>	104
10.1.4	<i>Data protection Impact Assessment</i>	105
10.1.5	<i>Flusso informativo del trattamento</i>	110
10.1.6	<i>Privacy Implementation Strategy</i>	111
10.2	CICLO DI VITA DELLO SVILUPPO SOFTWARE NELL'AMBITO DEL GDPR	111
10.3	IMPLEMENTAZIONE DELLA STRATEGIA NELLE FASI DI SVILUPPO DEL SOFTWARE	114
10.3.1	<i>Scopo</i>	114
10.3.2	<i>Le fasi di implementazione della Engineering Privacy by Design</i>	114
10.4	INTEGRAZIONE DELLA ENGINEERING PRIVACY BY DESIGN NEL SOFTWARE LIFE CYCLE PROCESS	115
APPENDICE 1.	CATALOGO SECURITY TOOLS	116
APPENDICE 2.	VALUTAZIONE STRUMENTI	128
A.	CHECKMARX	128
B.	CODEDX	132
C.	SONARQUBE	134
11	BIBLIOGRAFIA	137

LISTA DELLE TABELLE

Tabella 1 - Documenti di Riferimento	7
Tabella 2 - Definizioni	8
Tabella 3 - Acronimi	9
Tabella 4 - Struttura del Catalogo Security Tool	66
Tabella 5 - Principi generali della privacy	101
Tabella 6 - I sette principi della Privacy by Design	105
Tabella 7 - Tipologie di trattamento che rappresentano un rischio elevato	107
Tabella 8 - Esempi di attributi per indentificare una persona	109
Tabella 9 - Fasi dell'Engineering Privacy by Design.....	114

LISTA DELLE FIGURE

Figura 1 - Andamento delle vulnerabilità per anno [Fonte: https://www.cvedetails.com]	11
Figura 2 - Il costo degli attacchi	12
Figura 3 - Defence-in-Depth model for IT	13
Figura 4 - OWASP Top 10 - 2017	14
Figura 5 - Augment the life cycle with security tools.....	16
Figura 6 - Una porzione dell'albero di classificazione CWE	42
Figura 7- CWE Top 25 [Fonte: https://cwe.mitre.org/top25/archive/2019/2019_cwe_top25.html].....	43
Figura 8 - Secure development activities.....	45
Figura 9 - Modello fasi SSDLC	45
Figura 10 - Esempio di Schema di Risk Assessment.....	47
Figura 11 - Gestione del rischio nel ciclo di vita del Software	48
Figura 12 - Cyber Risk Management di AgID – Report dei rischi per categoria di minaccia	49
Figura 13 - Input e Output della fase Final Review - Secure Release.....	62
Figura 14 - SAMM Structure	79
Figura 15 - BSIMM SSF	81
Figura 16 - Training practice BSIMM.....	82
Figura 17 - Microsoft SDL.....	83
Figura 18 - Input e Output della fase Risk Assessment	92
Figura 19 - Input e Output della fase Threat Modeling Attack Surface Analysis	93
Figura 20 - Input e Output della fase Static Analysis	94
Figura 21 - Report di Checkmarx.....	96
Figura 22 - Interfaccia CodeDx.....	96
Figura 23 - Info Security Product Guide 2016 : Recensione CodeDX.....	97
Figura 24 - SonarQube	97
Figura 25 - Input e Output della fase Dynamic Analysis	98
Figura 26 - Continuous Security.....	99
Figura 27 – Esempio di flusso di valutazione necessità DPIA.....	106
Figura 28 - Esempio di flusso informativo del trattamento	111
Figura 29 - Integrazione della Engineering privacy by design nel Software Life Cycle Process	115

1 INTRODUZIONE

1.1 Scopo

Scopo del presente documento è fornire le linee guida per intraprendere un processo di sviluppo del software “sicuro”, nel corso di tutte le fasi SDLC, attraverso l’identificazione e l’implementazione di opportune azioni di sicurezza.

1.2 Struttura del documento

Il presente documento è articolato come di seguito:

- **Esigenze e Ambiti di Applicazione**, come nasce l’esigenza dello sviluppo di software sicuro.
- **Analisi delle iniziative e degli standard**, analizza lo scenario nazionale e globale fornendo una vista delle iniziative e dei risultati prodotti in termini di: metodi e modelli, standard best practices, strumenti. L’analisi dello scenario ha consentito la creazione del *Catalogo dei Security Tools*.
- **Secure Software Development Life Cycle (SSDLC): Analisi delle metodologie e dei processi**, analizza i diversi metodi e modelli SDLC esistenti, con l’obiettivo di identificare le caratteristiche che rendono un ciclo di sviluppo software sicuro ed efficace.
- **La sicurezza in tutte le fasi del ciclo di sviluppo software** è un approfondimento sulle fasi SDLC poiché, tradizionalmente, gli aspetti di sicurezza non sono mai considerati con sufficiente attenzione fin dall’inizio del SDLC stesso.
- **Training e formazione**, focalizza l’attenzione sul fatto che molti degli attuali problemi di sicurezza derivano da errori di progettazione e/o di implementazione, risolvibili solo disponendo di personale formato e consapevole.
- **Certificazioni professionali**, è un elenco delle principali certificazioni riconosciute in ambito InfoSec.

2 RIFERIMENTI

2.1 Documenti di Riferimento

Rif.	Codice	Titolo
DR-1.	--	Reg. (UE) 679/2016 “Regolamento generale sulla protezione dei dati” del 27/04/2017
DR-2.	--	ISO/IEC 12207:2008 “Software life cycle processes”, 01/02/2008
DR-3.	--	ENISA “Privacy and Data Protection by Design – from policy to engineering”, 12/2014
DR-4.	--	Ann Cavoukian “Privacy by Design – The 7 Foundational Principles”, Information & Privacy Commissioner, Ontario, Canada, 01/2011
DR-5.	--	ISO/IEC 29151:2017 “Code of practice for personally identifiable information protection”, 01/08/2017
DR-6.	--	MITRE Privacy Engineering Framework, MITRE Privacy Community of Practice (CoP), 18/07/2014
DR-7.	--	WP ART29 “Guidelines on Data Protection Impact Assessment (DPIA) and determining whether processing is “likely to result in a high risk” for the purposes of Regulation 2016/679”, 04/10/2017
DR-8.	--	32nd International Conference of Data Protection and Privacy Commissioners. “Privacy by design Resolution”, Jerusalem, Israele, 10/2010

Tabella 1 - Documenti di Riferimento

3 DEFINIZIONI E ACRONIMI

3.1 Definizioni

Vocabolo	Titolo
Applicazione Cloud	Applicazione sviluppata sfruttando la tecnologia Cloud Computing
Defence in Depth	Difesa a differenti livelli-Layered Defense.
Gray Box	Metodo di test del software costituito da una combinazione tra il metodo Black Box Testing e il metodo White Box Testing. Nel Black Box Testing, la struttura interna dell'elemento da testare è sconosciuta al tester e nel White Box Testing la struttura interna è nota. Nel Gray Box Testing, la struttura interna è parzialmente nota. Ciò comporta l'accesso alle strutture dati e agli algoritmi interni per definire i casi d'uso. Il test finale è assimilabile al tipo Black Box.
Hardening	processo che mira attraverso operazioni di configurazione specifica di un dato sistema e dei suoi componenti, a minimizzare l'impatto di possibili vulnerabilità, migliorandone quindi la sicurezza complessiva

Tabella 2 - Definizioni

3.2 Acronimi

Codice	Titolo
Amministrazione	AgID
AgID	Agenzia per l'Italia Digitale
APP	Atom Publishing Protocol
AsmL	Abstract State Machine Language
BRP	Business Risk Profile
CAPEC	Common Attack Pattern Enumeration and Classification
CC	Common Criteria
CCRA	Common Criteria Recognition Agreement
CE	Contratto Esecutivo
CERT	Computer Emergency Response Team
CQ	Contratto Quadro
CWE	Common Weakness Enumeration
DAST	Dynamic Application Security Testing
DiDI	Defense-in-Depth Index
ESA	European Space Agency
IACS	Automation and Control Systems
IASP	Instrumented application security testing

Codice	Titolo
IFSQ	Institute for Software Quality
IDS	Intrusion Detection System
IFSQ	Institute for Software Quality
ISA	International Society of Automation
I&O	IT Infrastructure & Operations
MISRA	Motor Industry Software Reliability Association
MSAT	Microsoft Security Assessment Tool
NESSI	Networked European Software and Services Initiative
OWASP	Open Web Application Security Project
PII	Personal Identification Information
RASP	Runtime application security testing
RTI	Raggruppamento Temporaneo di Impresa
SAFE CODE	Software Assurance Forum for Excellence in Code
SAMATE	Software Assurance Metrics and Tool Evaluation
SAMML	Software Assurance Maturity Model
SANSI	SANS Software Security Institute
SAST	Static Application Security Testing
SCA	Software composition analysis
SDLC	Software Development Life Cycle
SSA	Software Security Assessment
SSE	Secure Software Engineering
SSDLC	Secure Software Development Life Cycle
S&R	Security & Risk
SW	Software
WASC	Web Application Security Consortium

Tabella 3 - Acronimi

4 ESIGENZE E AMBITI DI APPLICAZIONE

4.1 Il panorama delle vulnerabilità applicative

Il panorama delle minacce per la sicurezza delle applicazioni è in costante evoluzione.

Secondo la fonte Gartner¹, già nel 2005, **OLTRE IL 75% DEGLI ATTACCHI ERANO INDIRIZZATI DIRETTAMENTE VERSO LE APPLICAZIONI.**

I fattori chiave di questa evoluzione sono i progressi fatti dagli attaccanti, il rilascio di nuove tecnologie, l'uso di sistemi sempre più complessi. Gli obiettivi degli attacchi sono le vulnerabilità, celate all'interno delle applicazioni software, che forniscono un facile percorso d'ingresso per compromettere i sistemi o lanciare nuovi attacchi e malware.

I dati riportati dal Clusit² nel “Rapporto 2019 sulla Sicurezza ICT”³, confermano un trend ancora in crescita degli attacchi informatici:

- Nel biennio 2017-2018 il tasso di crescita del numero di attacchi gravi è cresciuto del 37,7% aumentando fino a dieci volte rispetto al precedente biennio 2015-2016. Il settore pubblico rimane sempre in primo piano dei criminali (+44%).
- I punti deboli delle applicazioni e le vulnerabilità del software continuano a essere il mezzo più comune con cui i criminali informatici compiono attacchi esterni e, ancora più grave, lo sfruttamento di vulnerabilità note è ancora in crescita (+39,4%). Nonostante queste vulnerabilità possano essere risolte con misure adeguate, le vulnerabilità più comuni nelle applicazioni Web continuato a essere le stesse degli ultimi anni: il 60% presenta errori di convalida dell'input, il 70% difetti di encapsulamento di dati o funzionalità critiche all'interno dei componenti e oltre un terzo (35%) presenta problematiche provocate dall'abuso di API.
- Il numero di vulnerabilità segnalate al National Vulnerability Database (NVD)⁴ nel 2018 ha raggiunto quota 16.517, con un incremento del 12,8% rispetto all'anno precedente (14.647 nel 2017). Secondo CVE Details⁵, nel 2017 il totale delle fallo di sicurezza è cresciuto più del doppio rispetto al 2016 ed è continuato a crescere anche nel 2018:

¹ http://selagroup.sela.co.il/_Uploads/dbsAttachedFiles/GartnerNowIsTheTimeForSecurity.pdf

² Clusit (Associazione Italiana per la Sicurezza Informatica): www.clusit.it

³ <https://web.uniroma1.it/infosapienza/sites/default/files/RapportoClusit2019.pdf>

⁴ www.nvd.nist.gov/

⁵ <https://www.cvedetails.com/>

	1677	January	February	March	April	May	June	July	August	September	October	November	December
2001	1677												
2002	2156	January	February	March	April	May	June	July	August	September	October	November	December
2003	1527	January	February	March	April	May	June	July	August	September	October	November	December
2004	2451	January	February	March	April	May	June	July	August	September	October	November	December
2005	4935	January	February	March	April	May	June	July	August	September	October	November	December
2006	6610	January	February	March	April	May	June	July	August	September	October	November	December
2007	6520	January	February	March	April	May	June	July	August	September	October	November	December
2008	5632	January	February	March	April	May	June	July	August	September	October	November	December
2009	5736	January	February	March	April	May	June	July	August	September	October	November	December
2010	4652	January	February	March	April	May	June	July	August	September	October	November	December
2011	4155	January	February	March	April	May	June	July	August	September	October	November	December
2012	5297	January	February	March	April	May	June	July	August	September	October	November	December
2013	5191	January	February	March	April	May	June	July	August	September	October	November	December
2014	7946	January	February	March	April	May	June	July	August	September	October	November	December
2015	6484	January	February	March	April	May	June	July	August	September	October	November	December
2016	6447	January	February	March	April	May	June	July	August	September	October	November	December
2017	14714	January	February	March	April	May	June	July	August	September	October	November	December
2018	16556	January	February	March	April	May	June	July	August	September	October	November	December
2019	12046	January	February	March	April	May	June	July	August	September	October		

Vulnerabilities By Year

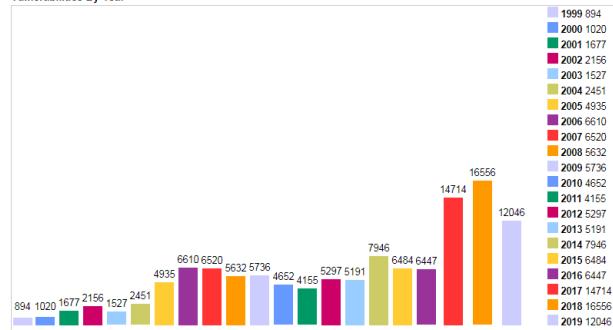


Figura 1 - Andamento delle vulnerabilità per anno [Fonte: <https://www.cvedetails.com>]

Nella Figura 1 è rappresentato il numero delle vulnerabilità nel periodo 1999-2019. Sull'asse delle ascisse sono riportati gli anni progressivamente dal 1999 al 2019, mentre nelle ordinate è indicata la numerosità delle vulnerabilità riscontrate per anno.

Tra le principali cause, si riscontra l'adozione di metodologie concentrate, soprattutto, sulla correzione di difetti funzionali e di attenzione alle performance delle logiche applicative, trascurando l'attuazione di pratiche di progettazione e programmazione che garantiscono la sicurezza del codice.

Da qui anche l'appello della comunità OWASP⁶ che sottolinea la necessità di accrescere la consapevolezza sulla sicurezza delle applicazioni, poiché il SW non sicuro mette a repentaglio le infrastrutture anche più critiche (finanziarie, sanitarie e difensive).

E' necessario rispondere in modo efficace alle sfide sulla sicurezza delle applicazioni, dotandosi di soluzioni adeguate per:

- Migliorare la gestione del programma di sicurezza delle applicazioni. Le componenti chiavi di un programma di sicurezza devono includere:
 - Risk Management Integration,
 - Architect & Developer Guidance,
 - Process Improvement (SDLC),
 - Secure Development Activities,
 - Vulnerability Management Integration;
- Valutare il codice software e le applicazioni al fine di identificare le vulnerabilità;
- Automatizzare la correlazione dei risultati della verifica della sicurezza per applicazioni interattive, statiche e dinamiche.

4.2 Sviluppo applicazioni sicure

La sicurezza informatica è l'insieme delle tecniche che mirano a proteggere l'ambiente informatico che include: gli utenti, le reti, le applicazioni, i processi e i dati. Questa sicurezza "integrata" implica una visione della security a 360° il cui obiettivo principale è di ridurre i rischi, compreso la prevenzione e la mitigazione degli attacchi informatici.

⁶ A free and open software security community (<https://www.owasp.org>)

Le applicazioni software dovrebbero avere caratteristiche di sicurezza base di default (**Secure By Default**) quali, ad esempio, l'abilitazione automatica di meccanismi di costruzione di password complesse piuttosto che procedure di rinnovo delle stesse secondo una scadenza di natura temporale. Un cambiamento di paradigma nello sviluppo di software (security by design/default) è invocato anche nel nuovo regolamento UE per la protezione dei dati (Art. 25⁷).

Le violazioni causano danni economici reali alle aziende che spesso richiedono mesi e addirittura anni per risolversi. Secondo l'ultimo Report Cisco (2018 Annual Cybersecurity Report), più della metà di tutti gli attacchi (circa il 53%) ha causato danni finanziari per oltre 500.000 dollari americani e ha riguardato perdite di fatturato, di clienti, di opportunità e il dover sostenere costi aggiuntivi non previsti.

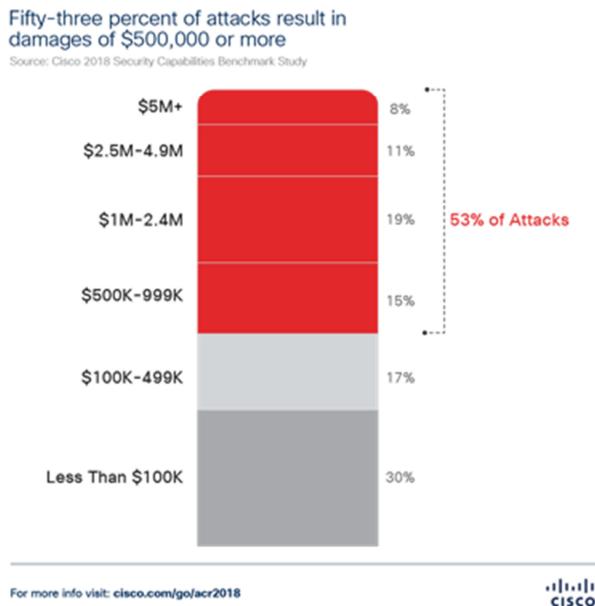


Figura 2 - Il costo degli attacchi

[Fonte: Cisco 2018 Security Capabilities Benchmark Study]

Lo studio ha coinvolto oltre 3600 intervistati in ventisei paesi. Riguardo l'Italia: il 38% delle aziende intervistate all'interno dello studio stima di aver subito danni inferiori ai 100.000 dollari, e il 37% ha subito danni che hanno superato i 500.000 dollari, mentre il 25% ha subito danni per cifre comprese tra i 100.000 e i 499.000.

⁷ <http://www.privacy-regulation.eu/it/25.htm>

L'approccio migliore per proteggere un sistema informativo, è garantire che ogni sua componente abbia un proprio meccanismo di protezione. La costruzione di strati multipli di controlli di sicurezza posti lungo un sistema è definita **Defence in Depth**.

La Defense-in-Depth è l'approccio alla sicurezza delle informazioni che prevede il raggiungimento di una adeguato livello di sicurezza attraverso l'utilizzo coordinato e combinato di molteplici contromisure.

Questa strategia difensiva si fonda sull'integrazione di differenti categorie di elementi: persone, tecnologie e modalità operative. La ridondanza e la distribuzione delle contromisure possono essere sintetizzate in una "difesa a differenti livelli" ("Layered Defenses"). Il concetto, di derivazione militare, si basa sull'assunto che nel caso in cui un attacco abbia successo, a causa del fallimento di un meccanismo di sicurezza, altri meccanismi di sicurezza possono intervenire per consentire un'adeguata protezione dell'intero Sistema.

Layered Approach to IT Security

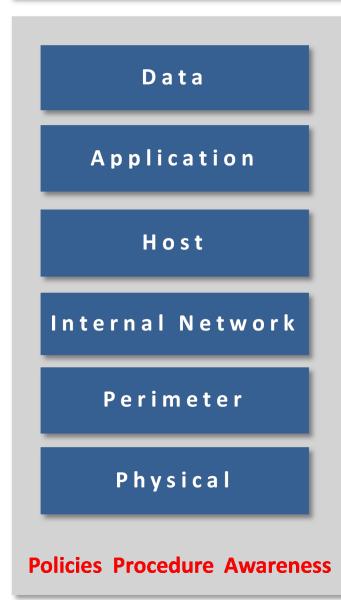


Figura 3 - Defence-in-Depth model for IT

Diverse sono le iniziative che si sono incentrate sulle problematiche Secure Development promuovendo azioni di sensibilizzazione (indirizzate ad aziende e community di sviluppatori) quali:

- la diffusione delle fondamentali best practices in materia di sicurezza applicativa (le prime tra tutte riconducibili a una buona ingegnerizzazione del software);
- una piena comprensione delle minacce più comuni (compresi i difetti propri dei linguaggi di programmazione);
- ancora più importante, una considerazione della problematica fin dalle prime fasi del ciclo di sviluppo.

L'OWASP traccia periodicamente la lista delle vulnerabilità più critiche delle applicazioni web. L'obiettivo è appunto, quello di educare e sensibilizzare sulle conseguenze che possono scaturire da implementazioni errate e facilmente vulnerabili. L'ultimo rapporto OWASP è stato rilasciato nel novembre del 2017 (OWASP Top 10 – 2017). La maggior parte delle problematiche identificate nella OWASP Top 10 – 2017 sono le stesse (o comunque molto simili) a quelle identificate nel rapporto precedente (OWASP Top 10 – 2013) con qualche novità, come si evince dalla figura che segue:

OWASP Top 10 - 2013	→	OWASP Top 10 - 2017
A1 – Injection	→	A1-Injection
A2 – Broken Authentication and Session Management	→	A2-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↗	A3-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	U	A4-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↗	A5-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	U	A7-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10-Insufficient Logging&Monitoring [NEW,Comm.]

Figura 4 - OWASP Top 10 - 2017

L'adozione di un Secure Software Development Life Cycle (SSDLC) atto a considerare e implementare opportune attività di sicurezza, nel corso di tutte le sue fasi del ciclo di vita del SW (dall'analisi fino alla manutenzione), è una necessità inderogabile per rispondere in modo efficace alle problematiche di sicurezza e per ridurre i costi che comportano trascurarla.

Ripensare alla sicurezza tra responsabilità e consapevolezza, oltre ad essere una buona pratica, è anche un obbligo di legge (Regolamento UE 679/2016).

4.3 Security Tools

Nell'ambito della cybersecurity, Forrester⁸, nel suo report “**Five steps to reinforce and harden application security**”⁹, rileva la necessità di cooperazione tra i team Security & Risk (S&R) e gli IT manager (I&O), ribadendo più volte come i primi non siano in grado, da soli, di coprire tutte le vulnerabilità scaturite dalle nuove esigenze in ambiti IT e digital business. Dal punto di vista dell'analista, infatti, l'IT team deve adottare, attraverso opportuni meccanismi di automazione e integrazione, le **security practices** all'interno di una ‘**continuous delivery pipeline**’. Questo garantisce una maggiore visibilità nelle interazioni tra hardware, software, servizi web e customer data. I professionisti I&O hanno, quindi, l'obiettivo di creare un ambiente di sicurezza ‘responsive’.

A tal fine, Forrester propone cinque steps per costruire un **responsive security environment**:

Step 1: rimuovere le ‘inconsistenze’ e creare un ‘conto’ dei materiali

Innanzitutto è necessario eliminare tutte le problematiche di sicurezza spesso derivanti da vulnerabilità riconducibili a servizi non più utilizzati e non più mantenuti o una cattiva gestione degli accessi e delle autorizzazioni. Tale attività deve essere svolta attraverso la collaborazione tra i team dedicati (I&O e S&R).

⁸ <https://www.forrester.com/>

⁹ <https://www.forrester.com/report/Five+Steps+To+Reinforce+And+Harden+Application+Security/-/E-RES127875>

In aggiunta, il censimento delle componenti applicative (attraverso un approccio ‘application modeling’) consente di ottenere ulteriori benefici in termini di: riduzione del mean-time-to-repair (attraverso l’impiego di strumenti di gestione della configurazione a sostegno del processo di monitoraggio delle modifiche applicative e dell’infrastruttura a supporto); utilizzo limitato di software per l’analisi delle vulnerabilità di terze parti (la visione completa dell’applicazione e di come interagisce con gli altri sistemi esistenti consente di limitare l’uso di ulteriori strumenti); rapida rimozione dei ‘difetti’ che possono generare vulnerabilità.

Step 2: limitare e rinforzare l’accesso ai sistemi e ai network device, monitorare i cambiamenti

Generalmente l’accesso intenzionale, non autorizzato, ai dati presenti all’interno della propria organizzazione, consegue, essenzialmente, da vulnerabilità derivanti da un hardening non adeguato, da problematiche di sicurezza nel software/hardware e/o da una cattiva progettazione del sistema stesso. E’ necessario lavorare a livello infrastrutturale per bloccare tutti gli accessi non autorizzati monitorando costantemente network e traffico sui sistemi. I team di gestione dell’infrastruttura e quelli della sicurezza dovrebbero cooperare nel processo di identificazione delle policy e dei tool per il monitoraggio, delle applicazioni in particolare, per verificare in tempo reale eventuali cambiamenti prima che questi si traducano in vulnerabilità.

Step 3: assistere i team di Security&Risk sul fronte intrusion detection & response

E’ richiesto l’impiego di sistemi infrastrutturali e tool tecnologici a supporto delle politiche di sicurezza. Questi svolgono un ruolo determinante nella prevenzione (e nella risposta) delle intrusioni in quanto, a fronte di anomalie (legate ad esempio all’utilizzo delle Cpu o al numero delle transazioni di sistema), avendo il controllo di tutto lo stack tecnologico, riescono a fornire in tempo utile alert e informazioni ai team di sicurezza. Un sistema di controllo di questo tipo accelera il mean-time-to-detection (il tempo di localizzazione di una vulnerabilità o di un attacco) e il tempo di risposta. Inoltre, cosa molto importante, riduce il range dei falsi allarmi di sicurezza (grazie ai controlli incrociati tra i team di infrastruttura e i team della sicurezza).

Step 4: ‘loggare’ quanto più possibile

E’ estremamente importante l’attività di tracciamento e di monitoraggio in tutte le fasi del ciclo di vita di sviluppo dell’applicazione. L’obiettivo è di analizzare tutte le fonti dati nonché il materiale di ciascuna applicazione, e monitorarne ogni minimo cambiamento. A tal fine, dal punto di vista tecnologico, Forrester suggerisce:

- i) l’integrazione degli Application Release Automation tool nei processi di auditing;
- ii) l’adozione di sistemi di Automate Change Tracking e dashboard a supporto dai team di I&O e S&R.

Step 5: creare uno stack di application security tool

Gli step precedenti concorrono alla creazione di un vero e proprio stack tecnologico incentrato sulla sicurezza applicativa. Al fine di indirizzare correttamente una protezione efficace delle applicazioni, è di

fondamentale importanza individuare le vulnerabilità (e porvi rimedio) sin dalle prime fasi del ciclo di vita dello sviluppo, quando è ancora poco costoso e poco rischioso intervenire.

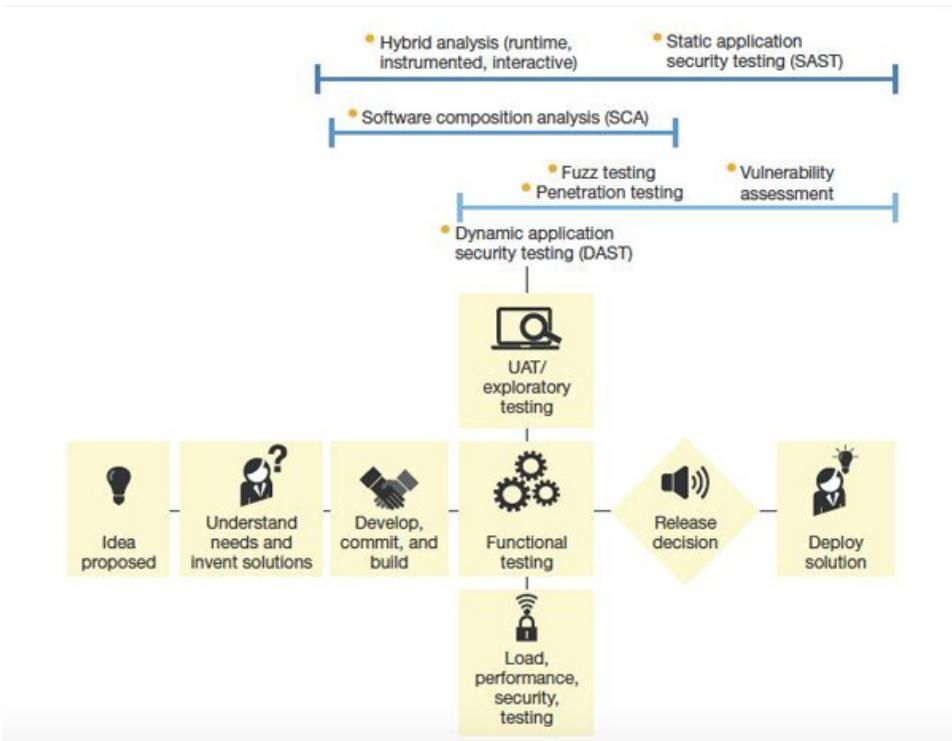


Figura 5 - Augment the life cycle with security tools

[Fonte: Forrester, Five Steps To Reinforce And Harden Application Security]

Per comporre lo stack, queste le tecnologie cui gli I&O professional dovrebbero porre attenzione:

- **Static Application Security Testing (SAST)**, tool che esaminano il codice binario e il codice di programmazione delle applicazioni senza ‘mandare in esecuzione’ l’applicazione (ossia senza la necessità di farla girare sui sistemi nei processi di testing);
- **Software composition analysis (SCA) tool**, tecnologie che consentono di analizzare le building block applicative per scovare vulnerabilità all’interno, per esempio, delle librerie, dei componenti open source o dei vari ‘blocchi’ di software che compongono l’applicazione.
- **Dynamic Application Security Testing (DAST)**, sistemi che permettono di osservare in dettaglio come si comporta l’applicazione quando è in funzione per scovarne imperfezioni o vulnerabilità prima che si prosegua con lo step di sviluppo successivo;
- **Fuzz testing tool**, sistemi che analizzano le vulnerabilità sul fronte di protocolli network, application data e input location (sempre durante i cicli di testing applicativo);
- **Hybrid analysis tool**, si tratta di tecnologie di testing per la sicurezza delle applicazioni che integrano funzionalità di Instrumented application security testing (LAST) e Runtime application security testing (RASP) utili per ridurre i falsi positivi e i falsi negativi generalmente evidenziati dai sistemi DAST;

- **Vulnerability assessment tool**, sistemi utili a rendere visibili eventuali criticità a livello di sistema operativo, configurazione dei sistemi, micro-configurazioni dei server e delle altre architetture con cui l'applicazione in sviluppo dovrà interagire una volta messa in produzione;
- **Penetration testing tool**, tecnologie utili a ‘validare’ l’assessment delle vulnerabilità perché mostrano come potrebbero avvenire gli attacchi simulando la penetrazione nei sistemi e nelle applicazioni.

5 ANALISI DELLE INIZIATIVE E DEGLI STANDARD

5.1 Iniziative Internazionali

5.1.1 Open Web Application Security Project (OWASP)

L'Open Web Application Security Project (chiamato semplicemente OWASP) è un progetto open-source per la sicurezza delle applicazioni Web. L'OWASP offre guide con consigli sulla creazione di applicazioni Internet sicure, e indicazioni per i test cui andrebbero sottoposte. È stato, inoltre, pubblicato WebGoat, utile ad apprendere, attraverso esempi concreti, le minacce più diffuse per la sicurezza delle applicazioni web. Nel 2004 è stata istituita una fondazione no-profit che supporta l'OWASP e che persegue l'obiettivo di aumentare la sicurezza delle applicazioni consentendo di prendere le decisioni in base ai rischi. In Europa è un'organizzazione no-profit registrata da giugno 2011 ed è presente anche in Italia.

La filosofia cui si ispira OWASP si può riassumere nei seguenti punti:

- **Apertura.** Tutto in OWASP è aperto e trasparente, dal codice sorgente ai bilanci societari.
- **Innovazione.** OWASP incoraggia e supporta l'innovazione e la sperimentazione per trovare nuove e sempre più efficaci soluzioni alle sfide della sicurezza del software.
- **Universalità.** Chiunque è incoraggiato a partecipare alla comunità OWASP.
- **Integrità.** OWASP è una comunità globale, che si basa sull'onestà e sull'indipendenza.

URL	https://www.owasp.org/
Country of HQ location	US
Geographic Scope	International
Type	Various Industry (not for profit)

L'iniziativa è organizzata come una comunità collaborativa che produce tool e documenti nelle seguenti tre aree principali:

- Protection,
- Detection,
- Life-cycle security.

Relativamente a queste tre aree, OWASP ha prodotto:

- un insieme di guide sulle buone pratiche quali: OWASP Testing Guide, OWASP Code Review e Software Assurance Maturity Model;
- il Report' OWASP Top 10' sui rischi per le applicazioni web.

Da considerare inoltre, come attività rilevanti svolte da OWASP, quanto segue:

Good Practice	[Protection Area] OWASP Secure Coding Practices - Quick Reference Guide v2.0 - Un insieme indipendente dalla tecnologia di pratiche di codifica della sicurezza generale del software, in formato checklist, che può essere integrata nel ciclo di vita dello sviluppo del software. [Protection Area] OWASP Developers Guide v2.0 (2005) - Un documento completo che copre tutti gli aspetti della sicurezza delle applicazioni e dei servizi web. [Detection Area] OWASP Code Review Guide v2.0 - Una guida che raccoglie le
----------------------	--

migliori pratiche per la revisione del codice.

[Detection Area] OWASP Testing Guide v4.0 - Una guida sulle procedure e checklist di test di sicurezza dell'applicazione.

[Detection Area] OWASP Mobile Security Testing Guide (MSTG). Un manuale completo per il test di sicurezza delle applicazioni "mobile" e il reverse engineering per il security testing delle piattaforme iOS e Android.

Standards

[Detection Area] Application Security Verification Standard (ASVS). L'ASVS definisce uno standard internazionale per la valutazione della sicurezza delle applicazioni e copre sia la verifica delle applicazioni automatizzata che quella manuale, utilizzando tecniche di test di sicurezza e di revisione del codice.

[Detection Area] OWASP Mobile Application Security Verification Standard (MASVS). Uno standard per la sicurezza delle applicazioni mobili.

Tools (Projects)

[Detection Area] Progetto OWASP Web Testing Environment (WTE). Una raccolta di strumenti di sicurezza delle applicazioni e di documentazione disponibile in diversi formati come VM, pacchetti di distribuzione Linux, installazioni basate su cloud e immagini ISO. Il progetto OWASP WTE è un miglioramento dell'originale OWASP Live CD Project.

[Detection Area] Progetto Zed Attack Proxy (ZAP) - Questo progetto di punta di OWASP è tecnicamente uno strumento proxy per intercettare, attraverso il traffico di rete, le vulnerabilità nelle applicazioni web. È stato progettato per essere utilizzato da persone con un'esperienza consolidata in materia di sicurezza e, come tale, è ideale per gli sviluppatori e tester funzionali chiamati a svolgere il penetration testing. Include le caratteristiche dei vecchi progetti WebScarab e DirBuster.

[Detection Area] Progetto SWFIntruder. È uno strumento per analizzare e testare la sicurezza delle applicazioni flash in fase di esecuzione.

[Life cycle security Area] Progetto OWASP WebGoat. Un'applicazione web insicura per insegnare la sicurezza delle applicazioni web attraverso lezioni pratiche interattive.

[Life cycle security Area] Piattaforma OWASP O2. Una raccolta di moduli Open Source a supporto dei professionisti della sicurezza delle applicazioni web per massimizzare i loro sforzi e ottenere rapidamente una significativa conoscenza del profilo di sicurezza di un'applicazione.

[Protection Area] OWASP OWASP OWTF. Un altro strumento di punta di OWASP per i pen-test.

[Detection Area] OWASP Dependency Check. Strumento per controllare e verificare la vulnerabilità delle librerie di terze parti utilizzate nei progetti di sviluppo software.

[Protection Area] OWASP Security Shepherd. Strumento destinato a migliorare la capacità di pen-test del personale di sicurezza.

[Protection Area] OWASP DefectDojo. Uno strumento open source di gestione delle vulnerabilità che semplifica il processo di testing, fornendo template, report, metriche e strumenti di base.

[Life cycle security Area] OWASP Juice Shop. Un'applicazione web volutamente insicura per i corsi di sicurezza scritta interamente in JavaScript che comprende l'intera Top Ten di OWASP e altri gravi difetti di sicurezza.

[Protection Area] OWASP Security Knowledge Framework. Uno strumento che viene utilizzato come guida per la creazione e la verifica di software sicuro; può essere utilizzato anche per formare gli sviluppatori sulla sicurezza delle applicazioni.

[Detection Area] OWASP Dependency Track. Una piattaforma di analisi della

composizione del software (SCA) che tiene traccia di tutti i componenti di terze parti per identificare proattivamente le vulnerabilità dei componenti che mettono a rischio le applicazioni.

[Life cycle security Area] OWASP Software Assurance Maturity Model (SAMM). Un framework aperto per aiutare le organizzazioni a formulare e implementare una strategia per la sicurezza del software su misura per i rischi specifici dell'organizzazione.

Code Projects	<p>[Protection Area] Progetto OWASP AntiSamy - Una libreria per la codifica HTML e CSS: API Java e .NET per la convalida degli input HTML/CSS forniti dagli utenti al fine di prevenire gli attacchi di cross-site scripting e phishing.</p> <p>[Life cycle security Area] Progetto OWASP Enterprise Security API (ESAPI) - Una raccolta di librerie di sicurezza gratuite e open source che possono essere utilizzate dagli sviluppatori per costruire applicazioni web sicure.</p> <p>[Protection Area] Progetto OWASP ModSecurity Core Rule Set (CRS). Un insieme di regole di sicurezza per configurare strumenti di firewall come ModSecurity.</p> <p>[Protection Area] Progetto OWASP CSRFGuard. Una libreria da includere nei progetti di sviluppo software per costruire una difesa contro gli attacchi CSRF (Cross-Site Request Forgery).</p> <p>[Detection Area] Progetto OWASP AppSensor. Un quadro concettuale e una metodologia che offre una guida prescrittiva per implementare il rilevamento delle intrusioni e la risposta automatica nelle applicazioni.</p> <p>[Protection Area] Progetto OWASP Top Ten. La pubblicazione OWASP più famosa: le prime 10 minacce per le applicazioni web, classificate per prevalenza, sfruttabilità, rilevabilità e impatto.</p>
----------------------	--

5.1.2 Common Criteria (CC)

I Common Criteria sono uno standard pubblicato dall'ISO (ISO/IEC 15408-1:2009¹⁰), lo standard è costituito da tre parti:

- Introduzione e modello generale
- Requisiti di sicurezza funzionali
- Requisiti di sicurezza di assurance

Con i CC è fornita anche una metodologia per la valutazione, la Common Criteria Evaluation Methodology (CEM), anch'essa standardizzata dall'ISO (ISO/IEC 18405:2008). Il processo di valutazione CC di un prodotto (software o hardware) riguarda diverse fasi del SDLC applicato:

- Requisiti (Protection Profile document - PP)
- Implementazione (Security Target document – ST)
- Test

Le verifiche previste sul sistema/prodotto, nel corso della valutazione da parte dello sviluppatore e del valutatore, mirano ad accertare che siano stati soddisfatti opportuni requisiti di assurance che diventano sempre più severi al crescere del livello di valutazione.

I CC definiscono una scala di sette livelli di valutazione:

¹⁰ <https://www.iso.org/standard/50341.html>

- EAL1. Functionally tested
- EAL2. Structurally tested
- EAL3. Methodically tested and checked
- EAL4. Methodically designed, tested and reviewed
- EAL5. Semi-formally designed and tested
- EAL6. Semi-formally verified design and tested
- EAL7. Formally verified design and tested.

I seguenti paesi hanno firmato l'accordo Common Criteria Recognition Agreement (CCRA) che si applica da EAL1 to EAL4:

- Paesi EU/EFTA: Austria, Repubblica Ceca, Danimarca, Finlandia, Francia, Germania, Grecia, Ungheria, Italia, Paesi Bassi, Norvegia, Spagna, Svezia e Regno Unito;
- Paesi Non-EU/EFTA: Australia, Canada, India, Israele, Giappone, Corea, Malesia, Nuova Zelanda, Pakistan, Singapore, Turchia e Stati Uniti.

L'European Mutual Recognition Agreement of IT Security Evaluation Certificates o 'SOGIS-agreement' è un accordo tra alcune nazioni europee con l'adesione dell'UE o dell'EFTA relativo al mutuo riconoscimento dei certificati di valutazione secondo gli standard CC per tutti i livelli di valutazione (EAL1 EAL7).

URL	https://www.commoncriteriaportal.org
Country of HQ location	International
Geographic Scope	
Type	Government

I criteri comuni per la valutazione della sicurezza informatica e la metodologia comune per la sicurezza delle tecnologie di valutazione sono stati pubblicati come standard ISO.

Risultati più rilevanti:

Standard	Common Methodology for Information Technology Security Evaluation and Common Criteria for Information Technology Security Evaluation Queste costituiscono la base tecnica di un accordo internazionale (CCRA). La versione 2.3 è stata pubblicata anche come ISO/IEC 15408:2009 e ISO/IEC 18045:2008.
Future	JTC 1/SC 27
Related Standard	ISO/IEC NP 20004 Tecnologie dell'informazione, tecniche di sicurezza, sviluppo di software sicuro e valutazione secondo le norme ISO/IEC 15408 e ISO/IEC 18405.

5.1.3 IEEE Computer Society

L'Iniziativa IEEE Computer Society è un'organizzazione senza fini di lucro, i principali progetti sono finalizzati alla pubblicazione di standard su tecnologie IT.

URL	https://www.computer.org
Country of HQ location	US
Geographic Scope	International
Type	Academic (not for profit)

Risultati di questa iniziativa sono libri, conferenze, pubblicazioni relative a conferenze, riviste, corsi on-line, certificazioni di sviluppo software, standard e riviste tecniche.

Risultati più rilevanti:

Good Practice	Guide to the Software Engineering Body of Knowledge (SWEBOK), la guida descrive le conoscenze generalmente accettate in materia di ingegneria del software. Le sue 15 aree di conoscenza (knowledge areas) riassumono i concetti di base e includono un elenco di riferimento per informazioni più dettagliate. Enterprise Information Technology Body of Knowledge (EITBOK) Guide. Un compendio di descrizioni di alto livello delle aree di conoscenza (knowledge areas) che sono generalmente necessarie per il buon funzionamento della tecnologia dell'informazione (IT).
Standard	Software & Systems Engineering Standards Committee (S2ESC) Formal Liaisons with ISO/IEC JTC1/SC7.

5.1.4 International Organisation for Standardization (ISO)

ISO è il più grande sviluppatore e editore al mondo di standard internazionali. Industrie ed esperti del settore generalmente contribuiscono come membri dei comitati tecnici ISO proponendo nuove normative che devono essere approvate almeno dal 70% dei membri ISO.

Il comitato tecnico che opera nell'ambito degli standard IT è il JTC 1 che, a sua volta, è organizzato in 22 sottocomitati che coprono aree specifiche. Si riporta di seguito un sottoinsieme significativo:

- ISO / IEC JTC 1 / SC 7: Ingegneria del software e dei sistemi;
- ISO / IEC JTC 1 / SC 22: Linguaggi di programmazione, compresi ambienti e interfacce software di sistema;
- ISO / IEC JTC 1 / SC 27: Sicurezza delle informazioni, sicurezza informatica e protezione della privacy;
- ISO / IEC JTC 1 / SC 38: Cloud Computing e piattaforme distribuite;
- ISO / IEC JTC 1 / SC 41: Internet of Things e tecnologie correlate;
- ISO / IEC JTC 1 / SC 42: Intelligenza artificiale.

Relativamente agli ambiti SSE troviamo:

- pubblicazione di rapporti tecnici e standard:
 - ISO / IEC TR 15026-1: 2013, ISO / IEC TR 24731-1: 2007, ISO / IEC TR 24772: 2013, ISO / IEC 15408 e ISO / IEC 18405
- 2 progetti in corso.

URL	https://www.iso.org
Geographic Scope	International
Type	Network of national standards institutes

Risultati più rilevanti:

ISO/IEC JTC 1/SC 7	<u>ISO/IEC 15026-1:2019 Systems and software engineering -- Systems and software assurance -- Part 1: Concepts and vocabulary</u>
ISO/IEC JTC 1/SC 22	<p><u>ISO/IEC TR 24731-1:2007</u> Information technology - Programming languages, their environments and system software interfaces - Extensions to the C library - Part 1: Bounds-checking interfaces. Specifica una serie di estensioni del linguaggio di programmazione C, specificato dalla norma internazionale ISO/IEC 9899: 1999. Queste estensioni possono essere utili nella mitigazione delle vulnerabilità di sicurezza nei programmi.</p> <p><u>ISO/IEC TR 24731-2:2010</u> Information technology — Programming languages, their environments and system software interfaces — Extensions to the C library — Part 2: Dynamic Allocation Functions. Fornisce funzioni alternative per la libreria C che favoriscono la programmazione sicura.</p> <p><u>ISO/IEC TR 24772:2013</u> Information technology - Programming languages - Guidance on avoiding vulnerabilities in programming languages through language selection and use. Specifica le vulnerabilità del linguaggio di programmazione software da evitare nello sviluppo di sistemi in cui è richiesto un comportamento sicuro ai fini security/safety, mission critical e software business-critical. In generale, questa guida è applicabile al software sviluppato, rivisto, o mantenuto per qualsiasi applicazione. Le vulnerabilità sono descritte in modo generico, applicabili a una vasta gamma di linguaggi di programmazione. Questa guida può essere anche utilizzata dagli sviluppatori per produrre o selezionare gli strumenti di valutazione del codice sorgente capaci di scoprire ed eliminare alcuni costrutti che potrebbero portare alla vulnerabilità del software o per selezionare un linguaggio di programmazione che consenta di evitare i problemi attesi.</p>

Progetti in corso:

ISO/IEC JTC 1/SC 7	<p><u>ISO/IEC 15026-2:2011</u> - Systems and software engineering - Systems and software assurance -- Part 2: Assurance case.</p> <p>Specifica i requisiti minimi per la struttura e il contenuto di un Assurance Case per migliorare la coerenza e la comparabilità degli Assurance Case e per facilitare le comunicazioni delle parti interessate, le decisioni d'ingegneria e altri Assurance Case.</p> <p>Secondo questo documento ISO “<i>An assurance case includes a top-level claim for a property of a system or product (or set of claims), systematic argumentation regarding this claim, and the evidence and explicit assumptions that underly this argumentation. Arguing through multiple levels of subordinate claims, this structured argumentation connects the top-level claim to the evidence and assumptions</i>”.</p> <p>ISO/IEC CD 15026-3 Systems and software engineering -- Systems and software assurance -- Part 3:2015 System Integrity levels.</p> <p>Si riferisce ai livelli d'integrità dell'Assurance Case e include i requisiti relativi al loro utilizzo con e senza un Assurance Case.</p> <p>Secondo questo documento ISO “<i>A software integrity level denotes a range of values of a software property necessary to maintain system risks within tolerable limits</i>”.</p>
ISO/IEC JTC 1/SC 27	<p>ISO/IEC 27021:2017 Information technology -- Security techniques -- Competence requirements for information security management systems professionals</p> <p>ISO/IEC/IEE 15026-1:2019: Systems and software engineering -- Systems and software assurance -- Part 1: Concepts and vocabulary</p> <p>ISO/IEC 15026-2:2011 Systems and software engineering — Systems and software assurance — Part 2: Assurance case</p> <p>ISO/IEC NP 20004: Information technology - Security techniques - Secure software development and evaluation under ISO/IEC 15408 and ISO/IEC 18405. Si riferisce a un problema differente e più urgente associato all'uso pratico dei Common Criteria, ossia la relazione tra i processi di sviluppo e di valutazione con l'analisi dei potenziali attacchi. E' legato all'iniziativa CAPEC.</p> <p>ISO/IEC TS 19608: 2018 Guidance for developing security and privacy functional requirements based on ISO/IEC 15408</p> <p>ISO/IEC TS 19249: 2017 Information technology — Security techniques — Catalogue of architectural and design principles for secure products, systems and applications</p> <p>ISO/IEC 27002:2013 Information technology — Security techniques — Code of practice for information security controls</p>

5.1.5 International Society of Automation (ISA)

ISA è un'organizzazione globale no-profit che sviluppa standard per l'industria, certifica i professionisti di settore, offre istruzione e formazione, pubblica libri e articoli tecnici, ospita convegni e fiere per i professionisti dell'automazione.

La cybersecurity per l'industria è diversa dalle altre aree. Nell'automazione industriale la priorità è mantenere l'impianto in funzione garantendo, laddove possibile, integrità e riservatezza (AIC - availability,

integrity and confidentiality) mentre nelle altre aree la priorità è la protezione dei dati (CIA - confidentiality, integrity, availability).

URL	https://www.isa.org/
Country of HQ location	US
Geographic Scope	International
Type	Industry (not for profit)

I membri ISA pagano una tassa regolare (annuale o biennale), in base al loro tipo di appartenenza, al fine di ottenere i benefici ISA come l'accesso alle informazioni tecniche e alle risorse per lo sviluppo professionale.

Risultati più rilevanti:

Standards	ANSI/ISA 62443 (formerly ISA-99) - Security for industrial automation and control systems - è una serie di standard, report tecnici e relative informazioni che definiscono le procedure per l'implementazione di sistemi sicuri di automazione e controllo industriale (IACS). La presente guida si applica a tutte le parti interessate che attuano o gestiscono l'IACS. Tutti gli standard ISA-62443 e i report tecnici sono organizzati in quattro categorie generali denominate <i>General, Policies and Procedures, System and Component</i> .
INDUSTRIAL CYBERSECURITY STANDARDS	ANSI/ISA-62443-4-2-2018, Security for industrial automation and control systems – Part 4-2: Technical security requirements for IACS components. ANSI/ISA-62443-4-1-2018, Security for industrial automation and control systems, Part 4-1: Product security development life-cycle requirements. Definisce un secure development life-cycle (SSDLC) allo scopo di realizzare e mantenere prodotti software sicuri. Questo ciclo di vita comprende la definizione dei requisiti di sicurezza, la progettazione sicura, l'implementazione sicura (incluse le linee guida di codifica), la verifica e la convalida, la gestione dei difetti di sicurezza, la gestione delle patch e la fine del ciclo di vita del prodotto. Tali requisiti possono essere applicati a processi nuovi o esistenti per sviluppare, mantenere e dismettere hardware, software o firmware per prodotti nuovi o esistenti. Tali requisiti si rivolgono allo sviluppatore e al manutentore del prodotto, ma non agli addetti all'integrazione né all'utente finale del prodotto. ANSI/ISA-62443-2-4-2018 / IEC 62443-2-4:2015+AMD1:2017 CSV, Security for industrial automation and control systems, Part 2-4: Security program requirements for IACS service providers (IEC 62443-2-4:2015+AMD1:2017 CSV, IDT). ANSI/ISA-62443-3-3 (99.03.03)-2013, Security for Industrial Automation and Control Systems Part 3-3: System Security Requirements and Security levels. Questo standard definisce i requisiti di sicurezza che sono raggruppati in sette categorie: 1) Controllo degli accessi, 2) Controllo dell'utilizzo, 3) Integrità dei dati, 4) Riservatezza dei dati, 5) Limitazione dei flussi di dati, 6) Risposta tempestiva a un evento e 7) Disponibilità delle risorse di rete. Ogni categoria comprende una mappatura dei requisiti per garantire un adeguato livello di sicurezza.

	ANSI/ISA-62443-2-1 (99.02.01)-2009, Security for Industrial Automation and Control Systems: Establishing an Industrial Automation and Control Systems Security Program.
	ANSI/ISA-62443-1-1 (99.01.01)-2007, Security for Industrial Automation and Control Systems Part 1: Terminology, Concepts, and Models.
INDUSTRIAL CYBERSECURITY CERTIFICATE PROGRAM	Certificate 1: ISA/IEC 62443 Cybersecurity Fundamentals Specialist. Certificate 2: ISA/IEC 62443 Cybersecurity Risk Assessment Specialist. Certificate 3: ISA/IEC 62443 Cybersecurity Design Specialist. Certificate 4: ISA/IEC 62443 Cybersecurity Maintenance Specialist. ISA/IEC 62443 Cybersecurity Expert.
CONFORMITY ASSESSMENT	Cybersecurity Certification to ISA/IEC 62443 Standards – It certifies devices and systems to the ISA/IEC 62443 Industrial Automation and Control Systems (IACS) cybersecurity standards.
TRAINING COURSES	Introduction to Industrial Automation Security and the ANSI/ISA99 Standards (IC32C). Using the ANSI/ISA99 Standard to Secure Your Control System (IC32). Industrial Networking and Security (TS12). Assessing the Cybersecurity of New or Existing IACS Systems (IC33). IACS Cybersecurity Design & Implementation (IC34). IACS Cybersecurity Operations & Maintenance (IC37).

5.1.6 Software Assurance Forum for Excellence in Code (SAFECode)

SAFECode è un'iniziativa privata creata da sviluppatori software e fornitori. Individuando e promuovendo le migliori pratiche in SSE, questa iniziativa sostiene che l'industria del software potrebbe rilasciare software, hardware e servizi più sicuri e affidabili. Tra le sue uscite principali, ci sono i documenti che raccolgono le migliori pratiche, tenendo conto del ciclo di vita di sviluppo del software.

URL	https://www.safecode.org
Country of HQ location	US
Geographic Scope	International
Type	Industry (not for profit)

SAFECode afferma che i suoi obiettivi sono:

1. Identificare e condividere collaudate pratiche di garanzia del software;
2. Promuovere una più ampia adozione di tali pratiche nell'ecosistema informatico;
3. Lavorare con istituzioni e fornitori di infrastrutture critiche per sfruttare le pratiche nella gestione dei rischi aziendali.

Risultati più rilevanti:

Training	Security Engineering Training Un quadro di riferimento per i programmi di formazione aziendale sui principi dello sviluppo sicuro del software. Security engineering training by SAFECode è una risorsa della comunità online che offre corsi gratuiti di formazione sulla sicurezza del software erogati
-----------------	--

tramite webcast on-demand.

Good Practice**Software Integrity Controls**

Un approccio impiegato per ridurre al minimo i rischi nella catena di fornitura del software. Sulla base delle pratiche dei membri SAFECode, il rapporto fornisce controlli di integrità per l'approvvigionamento, lo sviluppo, i test, la consegna e la resilienza del software.

The Software Supply Chain Integrity Framework

Documento che definisce i rischi e le responsabilità per rendere sicuro il software nella catena di fornitura globale. Sulla base dell'esperienza dei membri del SAFECode, descrive la catena di fornitura del software (modello a scala dei fornitori di software) e i principi per la progettazione dei controlli di integrità del software.

Fundamental Practices for Secure Software Development

Sulla base delle pratiche dei membri SAFECode, questo documento delinea un insieme di pratiche per lo sviluppo sicuro del software che possono essere applicate nelle diverse fasi del ciclo di vita dello sviluppo del software.

Software Assurance: An Overview of Current Industry Best Practices

Documento che descrive i metodi di sviluppo e i controlli di integrità utilizzati dai membri SAFECode per migliorare la sicurezza del software e la sicurezza nel rilascio.

Practices for Secure Development of Cloud Applications

SAFECode e la Cloud Security Alliance (CSA) rilasciano una guida per lo sviluppo sicuro di applicazioni cloud. Questo documento rappresenta il prodotto di tale collaborazione ed è destinato ad aiutare i lettori a comprendere meglio e implementare le migliori pratiche per lo sviluppo di software cloud sicuro.

Tactical Threat Modeling

Questo documento sfrutta le intuizioni dei membri del SAFECode per offrire modi efficaci per integrare meglio la modellazione delle minacce nei processi di sviluppo.

Managing Security Risks Inherent in the Use of Third-party Components

L'uso di componenti di terze parti (TPC), compresi i componenti software open source (OSS) o commerciali off-the-shelf (COTS), è diventato di fatto uno standard nello sviluppo del software. Questo documento analizza il processo e le procedure di cui gli sviluppatori necessitano per testare, migliorare e quantificare la sicurezza dei componenti di terze parti.

5.1.7 SANS Software Security Institute (SANS SSI)

SANS SSI offre una libreria di iniziative di ricerca e di community per aiutare sviluppatori, architetti, programmatore e responsabili della sicurezza delle applicazioni a proteggere le loro applicazioni software/web.

Questa iniziativa raccoglie e fornisce informazioni tecniche aggiornate, come l'accesso gratuito alle risorse sui più recenti vettori di attacco e sulle vulnerabilità di sicurezza delle applicazioni, tra cui un blog aggiornato, news-letters settimanali, webcast, articoli e documenti in materia di sicurezza del software.

URL	https://www.sans.org
Country of HQ location	US
Geographic Scope	International

Type	Academic
------	----------

SANS pubblica relazioni annuali (Top 25 Software Errors) con l'analisi sugli errori di programmazione più pericolosi: <http://www.sans.org/top25-software-errors/>.

L'ultima release (**2019 CWE Top 25 Most Dangerous Software Errors**) è fruibile al seguente link: https://cwe.mitre.org/top25/archive/2019/2019_cwe_top25.html.

Risultati più rilevanti:

Resources	<p>Application Security Resources: Whitepapers e webcasts sulla sicurezza delle applicazioni.</p> <p>Security Laboratory: Il "Security Laboratory" è un insieme informale di articoli e whitepaper sulla sicurezza, l'informatica e l'industria della sicurezza informatica.</p> <p>Fundamental Practices for Secure Software Internet Storm Center (ISC) Il ISC fornisce un servizio gratuito di analisi e di allarme agli utenti di Internet e alle organizzazioni. I volontari donano il loro tempo per analizzare difetti e anomalie e pubblicare un diario giornaliero delle loro analisi e riflessioni sul sito web di Storm Center.</p> <p>Application Security Procurement Language: Questo è un progetto di contratto software per gli acquirenti di software personalizzato. Il suo obiettivo è quello di rendere gli sviluppatori di codice responsabili del controllo del codice e della correzione dei difetti di sicurezza prima della consegna del software.</p> <p>Top 25 Software Errors. Sono elencate in tre categorie:</p> <ul style="list-style-type: none">• Interazione non sicura fra componenti• Risky Resource Management• Difesa insufficiente. <p>Ciascun errore include:</p> <ul style="list-style-type: none">• La classificazione all'interno della Top 25• Collegamenti a tutti i riferimenti alla CWE• Frequenza delle CWE e relative conseguenze nei campi dati• Costi di risanamento• Facilità di rilevamento• Esempi di codice• Metodi di rilevamento• Frequenza degli attacchi e consapevolezza degli aggressori• Le relative CWE e i modelli di attacco per questa vulnerabilità. <p>Comprende anche misure di prevenzione e bonifica sufficientemente estese che gli sviluppatori possono adottare per mitigare o eliminare la vulnerabilità.</p>
------------------	---

5.1.8 Web Application Security Consortium (WASC)

WASC produce best practice per le applicazioni web. WASC riassume la sua missione nella seguente frase "*to develop, adopt, and advocate standards for web application security*".

URL	http://www.webappsec.org/
Country of HQ location	US
Geographic Scope	International
Type	Industry (not for profit)

Risultati più rilevanti:

Resources	<p>Web Application Security Scanner Evaluation Criteria (WASSEC) Una serie di linee guida per valutare gli strumenti di scansione delle applicazioni web riguardo la loro efficacia nel testare e identificare le vulnerabilità.</p> <p>The Web Hacking Incidents Database (WHID) WHID è un progetto del Web Application Security Consortium dedicato al mantenimento di un elenco di applicazioni web relative agli incidenti di sicurezza.</p> <p>WASC Script Mapping Project Elenco delle modalità di esecuzione degli script all'interno di una pagina web senza usare i tag <script>.</p> <p>Distributed Web Honeypot (DWH) Project Identificare gli attacchi emergenti contro le applicazioni web e segnalarli alla comunità.</p> <p>Web Security Glossary Indice dei termini e della terminologia relativa alla sicurezza delle applicazioni web.</p> <p>WASC Threat Classification v2.0 È un effort per classificare le debolezze e gli attacchi che possono portare alla compromissione di un sito web, dei suoi dati o dei suoi utenti.</p> <p>Web Application Firewall Evaluation Criteria Sviluppo di criteri dettagliati per la valutazione di un firewall di un'applicazione web (WAF).</p> <p>WASC Web Application Security Statistics Raccolta di statistiche sulla vulnerabilità delle applicazioni per identificare e mappare i problemi di sicurezza delle applicazioni sui siti web aziendali.</p>
------------------	--

5.1.9 Institute For Software Quality (ifSQ)

L'Istituto per la Qualità del Software, con sede nei Paesi Bassi, è un gruppo di professionisti coinvolti nello sviluppo e nella distribuzione di software. ifSQ persegue un obiettivo comune: aumentare gli standard software (e dello sviluppo software) in tutto il mondo attraverso la promozione del Code Inspection, come prerequisito del Software Testing nel ciclo di produzione e rilascio del software.

URL	http://ifsq.org
Country of HQ location	The Netherlands
Geographic Scope	International
Type	Industry (non profit)

ifSQ ha analizzato, quantificato e migliorato lo stato dell'arte della ricerca sulla qualità del software, e ha definito un set di indicatori (Defect Indicators) che sono stati raccolti in un insieme coordinato di tre standard, pubblicati sul sito, in forma di opuscolo e sotto forma di corsi e workshop. La maggior parte dei criteri di valutazione, in particolare "major string", "parametri non controllati" e "unexpected state not trapped", sono rilevanti per migliorare la sicurezza del software.

Risultati più rilevanti:

Resources	Software Quality Standards - Levels 1 (An Entry-Level Standard for Computer Program Source Code), 2 (A Foundation-Level Standard for Computer Program Source Code) and 3 (Industry Best Practice for Computer Program Source Code - <i>is not yet complete</i>) are available.
------------------	--

5.2 Iniziative europee

Questo paragrafo ha l'obiettivo di fornire una vista delle iniziative in ambito Europeo. Le iniziative di seguito presentate sono state classificate sulla base dell'ambito geografico e della tipologia di appartenenza (accademiche, governative, industria).

Analizzando ambiti, obiettivi e risultati di ognuna, emerge che:

- Un insieme di iniziative rappresentano per obiettivi e risultati una categoria isolata. Tra queste iniziative, definiamole 'non raggruppabili', ci sono: NESSI, OWASP Local Chapters, MISRA e Serenity Forum.
- Altre iniziative posso essere 'raggruppate' sulla base di alcuni elementi che li caratterizzano e li accomunano: Events and Periodicals, Certifications, Academic Education. Queste iniziative potrebbero essere classificate con più tag sulla base dei loro risultati rilevanti o attesi in SSE: standardisation, industry platform, vulnerability detection, vulnerability protection, information sharing, specialised workshop, certification and training.

5.2.1 Networked European Software and Services Initiative (NESSI)

NESSI è la piattaforma tecnologica europea dedicata al Software e ai Servizi. L'obiettivo principale di NESSI si indirizza sul potenziamento dei servizi Internet attraverso attività di ricerca, standard e policy, e contributi costruiti attraverso una community industria/università.

I partecipanti NESSI sono divisi in tre gruppi:

- partner NESSI: prevalentemente industriale, ma ci sono anche alcuni profili accademici - coordinano la piattaforma e forniscono il sostegno finanziario per le attività NESSI;
- I membri NESSI: industria, mondo accademico e gli utenti - rappresentano i principali stakeholders del dominio della fornitura di servizi ICT. Non è obbligatorio un contributo finanziario
- abbonati NESSI: usano diversi canali di informazione per tenersi aggiornati sulle attività di NESSI.

URL	http://www.nessi-europe.com
Country of HQ location	Belgium
Geographic Scope	Europe
Type	Industry

Piattaforme tecnologiche nazionali e regionali sono parte della rete NESSI: gestiscono obiettivi NESSI da un punto di vista locale.

I focus NESSI hanno alcune correlazioni SSE:

- Identificare le direzioni della ricerca futura sui servizi;
- costruire contributi formali sui settori chiave;
- investire sulla rete NESSI per migliorare il coordinamento tra i programmi di ricerca europei, nazionali e regionali.

Risultati più rilevanti:

Research Agenda	<i>NESSI Strategic Research and Innovation Agenda (NESSI SRIA 2017)</i>
------------------------	---

Next Generation Software Technologies Empowering the Digital Transformation of Europe. Recommendations on Software Technology Research for Horizon Europe.

Working Group related to SSE	<i>Security and Privacy: From the Perspective of Software, Services, Cloud and Data.</i> NESSI è la Horizon 2020 European Technology Platform (ETP) per il software, i servizi e i dati. Il presente white paper si concentra sul ruolo crescente della sicurezza e della privacy e mette in evidenza le direzioni di ricerca di una prospettiva NESSI. Software and the Next Generation Internet (2019-05-09). Per sfruttare il potenziale delle NGI sono necessarie ricerca e innovazione per affrontare le sfide poste dalle crescenti minacce derivanti da attacchi informatici, compresa la gestione dei rischi e il contenimento delle intrusioni, nonché le minacce derivanti dalle nuove tecnologie.
-------------------------------------	---

5.2.2 Piattaforme Nazionali NESSI

L'obiettivo generale delle Piattaforme NESSI è di promuovere lo sviluppo e l'applicazione di tecnologie e servizi ICT per affrontare le sfide future all'interno dell'industria europea e del governo.

Nella tabella che segue vengono sintetizzate le attività di ciascuna piattaforma nazionale il cui scopo è di gestire gli obiettivi NESSI da un punto di vista locale e di pubblicare le proprie SRA nazionali.

URL	http://www.nessi-europe.com
NESSI - Norway	E' la filiale norvegese del NESSI. Il suo obiettivo principale è quello di creare un'arena norvegese per gli stakeholders del settore industria, ricerca/mondo accademico e pubblico e di influenzare la strategia di ricerca ICT del governo norvegese.
URL	http://www.nessi-europe.com
NESSI - Slovenia	Alla base di queste attività è che NESSI assumerà la responsabilità del contenuto e dell'attuazione del 7° programma quadro dell'UE per R&D. Essi invitano chiunque sia coinvolto in attività di R&D a partecipare a questo lavoro.

URL	http://www-it.fmi.uni-sofia.bg/nessibg/
NESSI-Bulgaria	<p>NESSI-Bulgaria è stata fondata nel 2005. Si tratta di un forum per lo scambio di conoscenze, lo sviluppo di strategie e la ricerca di nuove potenzialità a livello internazionale IT e servizi industriali. La visione centrale della piattaforma è di consentire nuovi modelli di business orientate ai servizi. I loro obiettivi sono:</p> <ul style="list-style-type: none"> • Definire una Roadmap bulgara e l'SRA per l'evoluzione del programma di innovazione R&D bulgaro. • Supporto alle attività R&D nei settori del software e dei servizi. • Fornire formazione: nuovi corsi, programmi MSc, programmi PhD e formazione
URL	http://nessi.ik.bme.hu/
NESSI- Hungary	<p>NESSI-Ungheria è stata fondata nel 2007 con lo scopo di evolvere la direzione della ricerca e dello sviluppo strategico nel settore del software e dei servizi, sulla base di un approccio unificato.</p> <p>Gruppi di lavoro di questa piattaforma sono divisi in due sottogruppi: domain-oriented e technological-oriented. La piattaforma è aperta a qualsiasi altra organizzazione ungherese.</p>
URL	http://www.bicc-net.de/
Germany Bicc-Net	<p>BICC-NET, Piattaforma di NESSI tedesca, è il Polo ICT bavarese della Germania. Fondata nel 2007, intende stimolare selettivamente l'innovazione. BICC-NET comprende quanto segue:</p> <ul style="list-style-type: none"> • sviluppo e distribuzione del software • lo sviluppo e la distribuzione di hardware • telecomunicazioni • sistemi software e hardware embedded nei prodotti • processi basati su software in fase di sviluppo, la produzione, i servizi e della pubblica amministrazione • Servizi nelle aree di cui sopra. <p>BICC-NET viene utilizzato per garantire la crescita ICT in Baviera. Essa è guidata dalla BICC sede ufficiale "cluster", che è stato direttamente commissionato dal Ministero bavarese per gli Affari economici, infrastrutture, trasporti e tecnologia.</p> <p>BICC-NET supporterà i profili di innovazione delle aziende ICT bavaresi e gli sviluppi in corso.</p>
URL	https://www.fi-stockholm.eu/
NESSI- Sweden	NESSI svedese è stata fondata nel 2010. L'obiettivo generale di NESSI Svezia è di promuovere lo sviluppo e l'applicazione di tecnologie e servizi ICT per affrontare le sfide future all'interno dell'industria svedese e del governo

URL	http://www.nessi-europe.com/
NESSI- Romania	<p>NESSI Romania è stata fondata nel 2010. Gli obiettivi a breve termine di NESSI-Romania sono:</p> <ul style="list-style-type: none"> • istituire gruppi di lavoro nazionali su diversi argomenti definiti in NESSI SRA • Definire un SRA nazionale per l'evoluzione futura del programma nazionale R&D e innovazione relativamente a software e servizi • Diffondere i risultati NESSI dei progetti strategici e compatibili

5.2.3 OWASP Local Chapters

Questa sezione fornisce una vista dei gruppi di lavoro OWASP distribuiti sul territorio Europeo.

URL	https://www.owasp.org/index.php/Belgium
OWASP Belgium Local Chapter	Le principali attività svolte riguardano l'organizzazione di incontri su come difendere le applicazioni web da attacchi.
URL	https://www.owasp.org/index.php/Aarhus
OWASP Denmark Local Chapter	Le principali attività svolte riguardano l'organizzazione di incontri su diversi argomenti di sicurezza delle informazioni legate alle applicazioni web. Le presentazioni sono disponibili sul sito web
URL	https://www.owasp.org/index.php/France
OWASP France Local Chapter	Le principali attività svolte riguardano l'organizzazione di incontri e la traduzione della documentazione OWASP in francese. Questo Chapter fornisce anche la formazione su progetti e risorse OWASP attraverso il programma "OWASP projects and resources you can use today", che ha lo scopo di promuovere progetti OWASP, fornendo una selezione di progetti maturi ed enterprise-ready, insieme con esempi pratici di come usarli.
URL	https://www.owasp.org/index.php/Germany
OWASP Germany Local Chapter	Le principali attività riguardano l'organizzazione di incontri, conosciuti come AppSec Germany Conference, che si svolge ogni anno.
URL	https://www.owasp.org/index.php/Geneva
OWASP Geneva Local Chapter	Le principali attività svolte da questo capitolo riguardano l'organizzazione di incontri legati alle identità digitali e autenticazione nelle applicazioni web.
URL	https://www.owasp.org/index.php/Greece
OWASP Greece Local Chapter	Il gruppo di lavoro OWASP greco è stata fondata nel 2005 con l'obiettivo di informare la comunità greca sui rischi per la sicurezza nelle applicazioni web. Il motivo principale che ha spinto alla sua creazione è il sempre crescente numero di incidenti di sicurezza su Internet, come ad esempio i tentativi di phishing a banche greche. Oggi, il gruppo greco promuove localmente

l'iniziativa OWASP attraverso il Software Libero/Open e la traduzione in greco della documentazione OWASP. Emettono una newsletter mensile, mantengono una mailing list per gli aggiornamenti e gestiscono dibattiti online su problemi di sicurezza di attualità.

La comunità greca OWASP vuole riunire tutti coloro che sono interessati e preoccupati per la sicurezza delle applicazioni web. Allo stesso tempo, accoglie i volontari che sono disposti a lavorare su progetti coordinati dall'OWASP, utilizzando software libero/open source. Invitano a chiunque di condividere le proprie idee, pensieri e riflessioni sugli attacchi, la difesa, i metodi di risposta, strumenti e buone pratiche in materia di sicurezza di Internet.

URL	https://www.owasp.org/index.php/Category:Ireland
-----	---

OWASP Ireland Local Chapter	Questo paese ha quattro gruppi locali: Belfast, Cork, Dublino e Limerick. Il gruppo più attivo è quello di Dublino le cui attività principali riguardano l'organizzazione di eventi e conferenze. Questo gruppo fornisce anche la formazione su progetti e risorse OWASP attraverso il programma " OWASP projects and resources you can use today". Questo ha lo scopo di promuovere i progetti OWASP, fornendo una selezione di progetti maturi ed enterprise-ready con esempi pratici di come usarli.
------------------------------------	---

URL	https://www.owasp.org/index.php/Italy
-----	---

OWASP Italy Local Chapter	Le attività riguardano l'organizzazione di eventi e lo sviluppo di tool. Il gruppo cerca di organizzare almeno 2 conferenze l'anno, uno in primavera e un altro in autunno. Recentemente, hanno lavorato sullo sviluppo di sqlmap, un <i>automatic SQL injection tool</i> sviluppato in Python. L'iniziativa è sostenuta da partner come IsecLab, CLUSIT e ISACA Roma.
----------------------------------	--

URL	https://www.owasp.org/index.php/Latvia%20
-----	---

OWASP Latvia Local Chapter	E' stata creata nell'ottobre 2007. Le attività principali riguardano l'organizzazione di eventi. Il gruppo non si è dimostrato molto attivo negli ultimi anni.
-----------------------------------	--

URL	https://www.owasp.org/index.php/London
-----	---

OWASP London Local Chapter	Le attività di OWASP Londra si concentrano sulla preparazione e l'organizzazione di eventi, conferenze e presentazioni. Il gruppo ha registrato elevata attività nel corso del 2010. Esso prevede anche la formazione su progetti e risorse OWASP attraverso il programma "OWASP projects and resources you can use today", che mira a promuovere progetti OWASP, fornendo una selezione di progetti maturi ed enterprise-ready con esempi pratici di come usarli.
-----------------------------------	---

URL	https://www.owasp.org/index.php/Luxembourg
-----	---

OWASP Luxembourg Local Chapter	Le attività del gruppo riguardano la preparazione e l'organizzazione di eventi e conferenze come il Java User Group (YAJUG) o Chaos Computer Club
---------------------------------------	---

Letzebuerg (C3L). Attualmente sembra che vi sia poca attività in questo gruppo.

URL	https://www.owasp.org/index.php/Norway
------------	---

OWASP Norway Local Chapter Le attività di OWASP Norvegia riguardano la preparazione e l'organizzazione di eventi e conferenze. Questo gruppo è stato molto attivo negli anni passati, quando ha organizzato 8 conferenze in Norvegia in un anno.

URL	https://www.owasp.org/index.php/Poland
------------	---

OWASP Poland Local Chapter L'attività principale che questo gruppo è di organizzare eventi. In questo gruppo sembra essere molto attivo, sono stati coinvolti in 11 conferenze nel corso del 2010. L'iniziativa è sostenuta da ISSA.

URL	https://www.owasp.org/index.php/Porto
------------	---

OWASP Portugal Local Chapter Le attività di questo gruppo riguardano l'organizzazione di conferenze e pubblicazioni. Ha organizzato uno dei più importanti eventi di OWASP: *Ibero-American Web Application Security Conference IBWAS'2010*.

URL	https://www.owasp.org/index.php/Scotland
------------	---

OWASP Scotland Local Chapter Le principali attività svolte da questo gruppo, secondo quanto riportato sul loro sito, sono finalizzate a fornire risposte insieme ad altri gruppi britannici locali ai diversi uffici governativi del Regno Unito. Questo gruppo sembra che organizzi anche incontri annuali.

URL	https://www.owasp.org/index.php/Spain
------------	---

OWASP Spain Local Chapter Questo gruppo svolge due attività principali. Da un lato collabora attivamente con OWASP su un progetto per fornire le specifiche e i requisiti legali per le applicazioni Web. D'altra parte, come la maggior parte degli altri gruppi locali di questa sezione, organizza eventi e conferenze annuali. Ha partecipato anche all'evento IBWAS'2010 [<https://www.owasp.org/index.php/IBWAS10>] in collaborazione con il gruppo portoghese.

URL	https://www.owasp.org/index.php/Sweden
------------	---

OWASP Sweden Local Chapter Questo gruppo si concentra sull'organizzazione di meeting ed eventi. Ha organizzato conferenze anche in collaborazione con altri gruppi del nord, come il norvegese e il finlandese.

URL	https://www.owasp.org/index.php/Switzerland
------------	---

OWASP Switzerland Local Chapter Questo gruppo organizza incontri su base periodica, soprattutto nella parte tedesca della Svizzera. I loro incontri e gli eventi sono principalmente su temi come test di sicurezza, lo sviluppo sicuro, hacking e architetture sicure. Sul loro sito Web sono fruibili diapositive di eventi e conferenze.

URL	https://www.owasp.org/index.php/Ukraine
OWASP Ukraine Local Chapter	E' un gruppo di recente formazione ancora in fase di organizzazione.

5.2.4 Motor Industry Software Reliability Association (MISRA)

MISRA è un *Motor Companies Consortium* all'interno del Regno Unito. I suoi risultati (ricerca, risultati della ricerca e standard de facto, linee guida) sono finalizzati principalmente allo sviluppo di software sicuro e affidabile per sistemi embedded nel settore automobilistico.

MISRA instaura quindi una collaborazione tra costruttori di veicoli, fornitori di componenti e di consulenza ingegneristica. Esso mira a promuovere le migliori pratiche nello sviluppo di sistemi elettronici legati alla sicurezza dei veicoli stradali e di altri sistemi embedded.

La sua documentazione non è accessibile al pubblico, ma può essere acquistata sul sito web del consorzio.

URL	https://www.misra.org.uk
Country of HQ location	UK
Geographic Scope	National
Type	Industry

I lavori in corso MISRA includono:

Model based development and autocode – Incoraggia alle buone pratiche.

- MISRA Autocode (Produzione di best practice di modellazione)
- MISRA C++ (Produzione di una serie di linee guida per l'uso di C ++ in sistemi critici)
- MISRA C3 (3rd review of MISRA C)
- Mira a promuovere le migliori pratiche nello sviluppo di sistemi elettronici legati alla sicurezza nei veicoli stradali e di altri sistemi embedded (è stato adottato e utilizzato in una vasta gamma di settori e applicazioni, tra cui il settore ferroviario, aerospaziale, militare e medico)

MISRA Safety Analysis – Linee Guida che descrivono come il ciclo di vita della sicurezza dei sistemi automotive si inserisce nel ciclo di vita dello sviluppo dei veicoli.

Risultati più rilevanti:

Good Practice	MISRA Compliance 2016: Achieving compliance with MISRA coding guidelines, ISBN 978-906400-13-2 (PDF), April 2016. Guidelines for the Use of the C Language in Vehicle Based Software, ISBN 978-0-9524156-6-5, April 1998, October 2002 Guidelines for the Use of the C Language in Critical Systems, ISBN 0 9524156 2 3 (paperback), ISBN 0 9524156 4 X (PDF), October 2004 Guidelines for safety analysis of vehicle based programmable systems, ISBN 978-0-9524156-5-7 (paperback), ISBN 978-0-9524156-7-1 (PDF), November 2007. Guidelines for the Use of the C++ Language in Critical Systems, ISBN 978-906400-03-3
----------------------	---

(paperback), ISBN 978-906400-04-0 (PDF), June 2008.

Standard	MISRA AC GMG: Generic modeling design and style guidelines, ISBN 978-906400-06-4 (PDF), May 2009. MISRA Compliance: MISRA C, MISRA C++ coding guidelines
-----------------	---

5.2.5 European Space Agency (ESA)

Dall'inizio degli anni '90 l'ESA si è occupata di definire la qualità dei prodotti software. La famiglia PSS¹¹ di standard (poi sostituito da standard ECSS) include un software engineering standard e una serie di guide.

URL	https://www.esa.int/
Country of HQ location	Paris
Geographic Scope	European
Type	Collaboration of Several European Countries

Uno degli standard di software ampiamente utilizzato, chiamato "Guide to applying the ESA Software Engineering Standards to small software projects" è disponibile all'indirizzo: http://emits.sso.esa.int/emits-doc/e_support/Bssc962.pdf

Questo standard definisce una serie di criteri di qualità per i requisiti software e di design, che hanno una influenza diretta e indiretta sulla sicurezza del software. Nell'ambito dei quality criteria requirements sono rilevanti i seguenti aspetti:

- *Sono menzionate le caratteristiche degli utenti e delle funzionalità del software maggiormente utilizzate? (Non risultano mancanti categorie di utenti)*
- *Sono esplicitamente menzionate tutte le interfacce esterne del software? (Non risultano mancanti interfacce)*
- *E' stata definita una priorità per ciascun requisito? (Il significato dei livelli di priorità è chiaro?)*
- *Ciascun requisito è verificabile (in un test di accettazione provvisoria)? (Misurabile: dove possibile, quantificare; capacità, prestazione e accuratezza).*
- *I requisiti sono consistenti? (Non sono in conflitto)*
- *I requisiti sono sufficientemente accurati e inequivocabili? (Quali interfacce sono coinvolte, chi ha l'iniziativa, chi fornisce quali dati, nessuna voce passiva).*
- *I requisiti sono completi? Tutto ciò che non è esplicitamente vincolato può essere considerato dal punto di vista dello sviluppo libero? Un prodotto che soddisfa tutti i requisiti è davvero accettabile? (Nessun requisito mancante)*
- *I requisiti sono comprensibili per coloro che li dovranno successivamente utilizzare?*
- *I requisiti sono realizzabili all'interno del budget?*
- *La maggior parte dei criteri di qualità di progettazione sono rilevanti per la sicurezza del software.*

¹¹ http://www.esa.int/TEC/Software_engineering_and_standardisation/TECBUCUXBQE_2.html

Risultati principali:

Good Practice	The PSS [https://www.esa.int/TEC/Software_engineering_and_standardisation/TECBUCUXBQE_2.html] famiglia di standard per la qualità del software. Una guida per l'applicazione degli standard ESA di ingegneria del software ai piccoli progetti è disponibile all'indirizzo: ftp://ftp.estec.esa.nl/pub/wm/wme/bssc/Bssc962.pdf L'Università di Tecnologia di Eindhoven fornisce ulteriori requisiti semplificati e checklists di progettazione. [https://www.win.tue.nl/is/doku.php]
----------------------	---

5.3 Iniziative US

In questa sezione viene fornita una panoramica delle iniziative SSE negli Stati Uniti. Tali iniziative sono state classificate in funzione della tipologia: accademiche o governative.

5.3.1 CERT Secure Coding

Il CERT Secure Coding è un'iniziativa di sicurezza del programma Computer Emergency Response Team (CERT). Questo programma fa parte del Software Engineering Institute (SEI) alla Carnegie Mellon University¹² (Pennsylvania, USA). Alcuni dei suoi programmi sono finanziati dal governo degli Stati Uniti.

Nel novembre 1988, la Defense Advanced Research Projects Agency (DARPA) incaricò il SEI di creare un centro per coordinare la comunicazione tra gli esperti di sicurezza durante le emergenze e per aiutare a prevenire futuri incidenti, a fronte di ciò, il CERT ha sviluppato il Software Initiative Assurance, che comprende: Secure Coding Standards, Source Code Analysis Lab (SCALE), Vulnerability analysis, Function extraction for malicious code.

Il SEI è un centro di ricerca e sviluppo finanziato dal governo federale, che conduce ricerche di ingegneria del software in acquisizione, architetture e linee di prodotto, miglioramento dei processi e misurazione delle performance, sicurezza e l'interoperabilità del sistema e l'affidabilità.

Il SEI lavora a stretto contatto con le organizzazioni di difesa e di governo, soprattutto l'Ufficio Secretary of Defense/Acquisition, Technology, and Logistics (OSD/AT&L)¹³, l'industria e il mondo accademico, con l'obiettivo di migliorare continuamente i sistemi software-intensivi.

URL	https://www.sei.cmu.edu
Country of HQ location	US
Geographic Scope	National
Type	Academic

Le aree di lavoro CERT Secure Coding sono:

- **Secure coding standards**

¹² <https://www.cmu.edu/>

¹³ <http://www.acq.osd.mil/>

[<https://wiki.sei.cmu.edu/confluence/display/seccode/SEI+CERT+Coding+Standards>] - Propone standard per migliorare la sicurezza nell'uso dei linguaggi di programmazione (Android, C, C++, Java, Perl).

- **International Standards Development** /- Standard di sviluppo Internazionale.
- **Source Code Analysis Laboratory (SCALE)** [cert.org/secure-coding/products-services/scale.cfm] SCALE consente di valutare il codice sorgente rispetto a una serie di standard di codifica sicura. SCALE rilascia e certifica i test di conformità quando le risultanze dei test sono state indirizzate dagli sviluppatori.
- **Secure Coding Tools** - Tali strumenti sono utilizzati nell'auditing SCALE, ma possono anche essere di supporto agli sviluppatori di software per ridurre il numero di vulnerabilità presenti nel loro codice.

CERT Secure Coding vuole influenzare i fornitori per migliorare la sicurezza base all'interno dei loro prodotti. Al fine di raggiungere questo obiettivo, CERT Secure Coding lavora con sviluppatori di software e organizzazioni di sviluppo software per ridurre le vulnerabilità derivanti da errori di codifica (C, C ++ o linguaggi di programmazione Java) prima di essere distribuiti. Inoltre, gli analisti CERT valutano le cause della vulnerabilità e identificano le pratiche di secure coding.

CERT collabora con ISO per la creazione di diversi standard su secure coding.

Risultati più rilevanti:

Training	Secure Coding in C and C++ [http://www.sei.cmu.edu/training/p63.cfm] Course of secure coding in C and C++ based on Addison-Wesley's material: "Secure Coding in C and C++" and "The CERT C Secure Coding Standard"
Standards for Software Developers	SEI CERT C Coding Standard [https://wiki.sei.cmu.edu/confluence/display/c/SEI+CERT+C+Coding+Standard] SEI CERT C++ Coding Standard [https://wiki.sei.cmu.edu/confluence/pages/viewpage.action?pageId=88046682] SEI CERT Oracle Coding Standard for Java [https://wiki.sei.cmu.edu/confluence/display/java/SEI+CERT+Oracle+Coding+Standard+for+Java] SEI CERT Perl Coding Standard [https://wiki.sei.cmu.edu/confluence/display/perl/SEI+CERT+Perl+Coding+Standard] Android TM Secure Coding Standard [https://wiki.sei.cmu.edu/confluence/display/android/Android+Secure+Coding+Standard]

5.3.2 Software Assurance Metrics and Tool Evaluation (SAMATE)

SAMATE è un'iniziativa US Government software assurance, un progetto inter-agenzie tra gli Stati Uniti e il DHS National Institute of Standards and Technology (NIST).

Obiettivo di SAMATE è migliorare la garanzia software:

- sviluppando metriche e metodologie per valutare i tool di sicurezza del software;
- identificando le vulnerabilità relative alla pratiche di codifica e dei metodi di ingegneria del software.

Il progetto di riferimento di SAMATE sviluppa casi di test al fine di esaminare il codice sorgente di strumenti e applicazioni. Rileva e segnala le debolezze in modo da fornire, agli utenti finali e sviluppatori, tool di garanzia del software con una serie di flaws noti attraverso i quali valutare i propri tool.

L'uscita principale di questa iniziativa è il SAMATE Reference Dataset (SRD), un database online alimentato regolarmente da SAMATE. Questa banca dati online, a disposizione del pubblico, fornisce casi di test per gli sviluppatori e utenti finali, attraverso i quali è possibile effettuare valutazioni di tool di sicurezza.

URL	https://samate.nist.gov/
Country of HQ location	US
Geographic Scope	National
Type	Governement

SAMATE è finalizzato al miglioramento del software assurance attraverso lo sviluppo di metodologie che consentano la valutazione software dei tool, misurare l'efficacia dei tool e delle tecniche, individuare le lacune negli strumenti e nei metodi. Il progetto sostiene Tools Software Assurance della US DHS e R&D Requirements Identification Program (in particolare, la Parte 3, tecnologia -strumenti e requisiti-), che affronta l'individuazione, la valorizzazione e lo sviluppo di software assurance tools.

Il progetto SAMATE compone di due parti:

- sviluppo di metriche per l'efficacia dei software security assessment (SSA) tools
- valutazione di metodi e strumenti SSA attuali al fine di individuare le carenze che possono portare a guasti dei prodotti software e vulnerabilità

Infine, SAMATE sta sviluppando anche alcune specifiche rivolte agli sviluppatori di strumenti di garanzia del software, che gli consentano di classificare e valutare questa tipologia di tool.

Risultati più significativi:

Specifications	Source Code Security Analysis [https://samate.nist.gov/index.php/Source_Code_Security_Analysis.html] “Source Code Security Analysis Tool Functional Specification Version 1.1” Specifiche e piani di test per gli strumenti di analisi della sicurezza del codice sorgente. Questo tipo di strumento esamina il codice sorgente al fine di rilevare e segnalare le difettosità che possono portare a vulnerabilità di sicurezza.
	Web Application Scanner [https://samate.nist.gov/index.php/Web_Application_Scanner.html] “Web Application Scanner Functional Specification Version 1.0”. Queste specifiche sono raccolte nella pubblicazione NIST Special Publication 500-269 [https://samate.nist.gov/docs/webapp_scanner_spec_sp500-269.pdf] .
Test Cases	SAMATE reference datasheet [https://samate.nist.gov/SRD/] Fornisce a utenti, ricercatori e sviluppatori di strumenti di garanzia della sicurezza del software una serie di difetti di sicurezza noti. Questi consentiranno agli utenti finali di valutare tali strumenti e agli sviluppatori degli

strumenti di testare le loro metodologie applicate.

SRD database

[<https://samate.nist.gov/SRD/view.php>]

Una raccolta di casi di test per individuare le debolezze del codice.

5.3.3 Common Weakness Enumeration (CWE)

CWE è un'iniziativa sostenuta e co-sponsorizzata dalla NCSD della US DHS e dal NIST. Attualmente è mantenuta e guidata da MITRE Corporation.

Il CWE è una lista formale o tassonomia, che classifica le tipologie più comuni di vulnerabilità del software. Gli obiettivi principali di CWE sono:

- Gestire la *common taxonomy* per la classificazione delle vulnerabilità comuni del software relativamente ad architettura, progettazione e codice;
- Fornire una classificazione standard per tool di protezione del software
- Fornire una linea di base da cui partire per aiutare la community SSE a identificare, attenuare e prevenire questo tipo di debolezza software.

URL	https://cwe.mitre.org https://nvd.nist.gov/cwe.cfm
Country of HQ location	US
Geographic Scope	National
Type	Government

Questo progetto utilizza i risultati del progetto SAMATE per creare l'elenco CWE delle vulnerabilità e la sua tassonomia associata e l'albero di classificazione (vedi figura sotto tratta dal NIST).

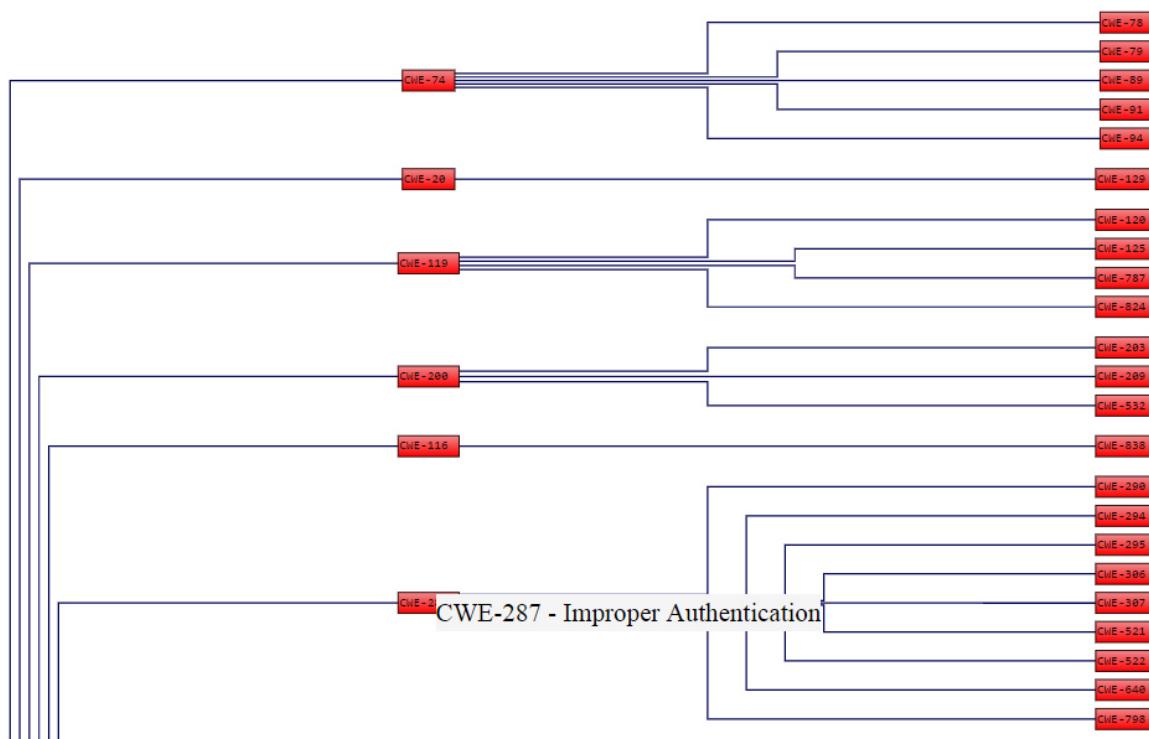


Figura 6 - Una porzione dell'albero di classificazione CWE

[Fonte: <https://nvd.nist.gov/vuln/categories/cwe-layout>]

La Figura 6 mostra la classificazione gerarchica delle CWE, come proposto nella pagina del National Vulnerability Database¹⁴ (NVD) della NIST. Il grafico presenta le varie CWE raggruppate in categorie. Ogni CWE può essere in relazione con una CWE di livello superiore (più generica).

Va inoltre sottolineato che CWE è una community-developed, l'elenco formale delle vulnerabilità comuni del software coinvolgono il mondo accademico, il settore commerciale e il governo degli Stati Uniti.

Risultati più rilevanti:

- **CWE List** (Version 3.4): <https://cwe.mitre.org/data/index.html>

Le definizioni e le descrizioni di CWE supportano la scoperta delle tipologie di flaw di sicurezza software nel codice, prima di rilasciarlo. Ciò significa che sia gli utilizzatori che gli sviluppatori dei tool e dei servizi di sicurezza software possono utilizzare CWE come un meccanismo per descrivere i flaw di sicurezza del software.

L'elenco CWE è disponibile in tre diversi formati:

- Research Concepts [<https://cwe.mitre.org/data/definitions/1000.html>];
- Development Concepts [<https://cwe.mitre.org/data/definitions/699.html>];
- Architectural Concepts [<https://cwe.mitre.org/data/definitions/1008.html>].

- **CWE Top 25 Most Dangerous Software Errors.** Di seguito è riportato un l'elenco pubblicato nel 2019:

¹⁴ <https://nvd.nist.gov/vuln/categories>

Rank	ID	Name	Score
[1]	CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer	75.56
[2]	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	45.69
[3]	CWE-20	Improper Input Validation	43.61
[4]	CWE-200	Information Exposure	32.12
[5]	CWE-125	Out-of-bounds Read	26.53
[6]	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	24.54
[7]	CWE-416	Use After Free	17.94
[8]	CWE-190	Integer Overflow or Wraparound	17.35
[9]	CWE-352	Cross-Site Request Forgery (CSRF)	15.54
[10]	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	14.10
[11]	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	11.47
[12]	CWE-787	Out-of-bounds Write	11.08
[13]	CWE-287	Improper Authentication	10.78
[14]	CWE-476	NULL Pointer Dereference	9.74
[15]	CWE-732	Incorrect Permission Assignment for Critical Resource	6.33
[16]	CWE-434	Unrestricted Upload of File with Dangerous Type	5.50
[17]	CWE-611	Improper Restriction of XML External Entity Reference	5.48
[18]	CWE-94	Improper Control of Generation of Code ('Code Injection')	5.36
[19]	CWE-798	Use of Hard-coded Credentials	5.12
[20]	CWE-400	Uncontrolled Resource Consumption	5.04
[21]	CWE-772	Missing Release of Resource after Effective Lifetime	5.04
[22]	CWE-426	Untrusted Search Path	4.40
[23]	CWE-502	Deserialization of Untrusted Data	4.30
[24]	CWE-269	Improper Privilege Management	4.23
[25]	CWE-295	Improper Certificate Validation	4.06

Figura 7- CWE Top 25 [Fonte: https://cwe.mitre.org/top25/archive/2019/2019_cwe_top25.html]

5.3.4 Common Attack Pattern Enumeration and Classification (CAPEC)

CAPEC è un'iniziativa co-sponsorizzata dal NCSD dell'US DHS e guidata dalla Digital¹⁵. Costruttori di software sicuro devono proteggersi da importanti vulnerabilità potenziali. Per identificare e mitigare le vulnerabilità relative al software, la community di sviluppo ha bisogno di capire la prospettiva dell'attaccante e gli approcci utilizzati per sfruttare il software.

Gli schemi di attacco sono le descrizioni di metodi comuni per lo sfruttamento del software, fornendo sia la prospettiva che la guida dell'attaccante sui modi per mitigare il loro effetto. Essi derivano dal concetto di pattern design applicato in un distruttivo, piuttosto che costruttivo, contesto e sono generati da un'analisi approfondita di specifici esempi di casi del mondo reale.

Questa iniziativa mira a fornire un catalogo a disposizione del pubblico di schemi di attacco, insieme ad uno schema di classificazione e tassonomia completo. La filosofia è di evolvere il catalogo con la partecipazione e i contributi pubblici e così consolidare un meccanismo standard per l'identificazione, la raccolta, la raffinazione, e la condivisione di modelli di attacco nella community software.

URL	https://capec.mitre.org
Country of HQ location	US
Geographic Scope	National
Type	Government

¹⁵ <https://www.synopsys.com/software-integrity.html>

Secondo questa iniziativa, le informazioni sugli schemi di attacco, se catturati in modo formale, possono portare un notevole valore per considerazioni di sicurezza del software attraverso tutte le fasi del SDLC e le altre attività relative alla sicurezza, tra cui:

- Raccolta dei requisiti: Identificazione dei requisiti di sicurezza pertinenti, dei misuse e abuse cases.
- Architettura e design: Fornisce il contesto per l'analisi dei rischi architettonici e le linee guida per la sicurezza nelle architetture del software.
- Implementazione e codifica: Prioritizzazione e guida delle attività di revisione sicura del codice.
- Test del software e controllo qualità: Fornisce il contesto per una appropriata analisi del rischio e test di penetrazione.
- Operatività dei sistemi: Sfruttare le esperienze apprese dagli incidenti di sicurezza per fornire una guida preventiva.
- Politiche e generazione di standard: Guida all'identificazione di adeguate politiche e standard organizzativi prescrittivi.

Risultati più rilevanti:

- **List of Attack Patterns** [<http://capec.mitre.org/>]. L'elenco è disponibile in due diversi formati:
 - View by Mechanisms of Attack [<http://capec.mitre.org/data/definitions/1000.html>].
 - View by Domains of Attack [<http://capec.mitre.org/data/definitions/3000.html>].

6 LA SICUREZZA IN TUTTE LE FASI DEL CICLO DI SVILUPPO DEL SOFTWARE

6.1 Secure SDLC

Generalmente gli aspetti di sicurezza sono sottovalutati fin dalle prime fasi del ciclo di vita dello sviluppo software e di conseguenza sono molte le vulnerabilità che vengono introdotte e trasmesse negli stadi successivi. È stato stimato, ad esempio, che un errore introdotto nella fase di specifica dei requisiti, può costare fino a 200 volte, se lo si corregge nelle successive fasi di sviluppo, rispetto a quanto sarebbe costata la sua immediata rimozione. L'attuazione corretta e completa delle **attività di sicurezza** nelle prime fasi consente di incrementare sensibilmente il livello di sicurezza di ogni singola fase successiva con un beneficio di ritorno importante:

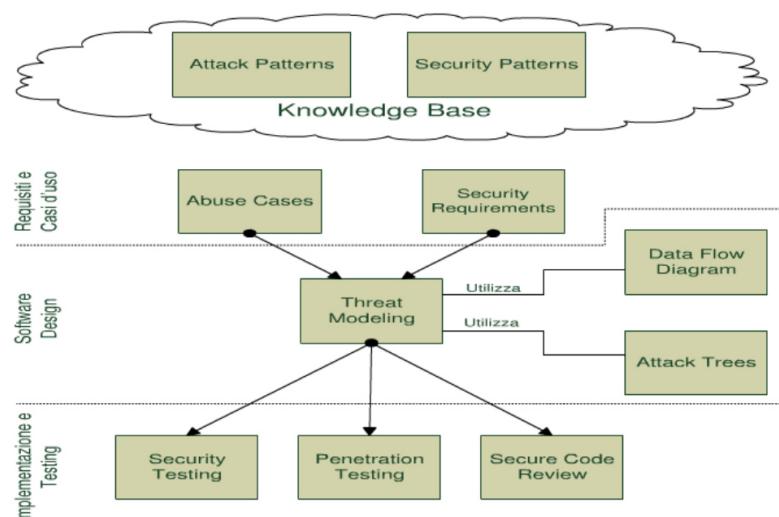
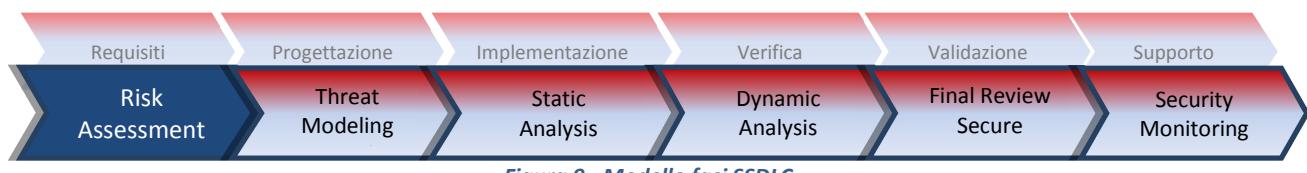


Figura 8 - Secure development activities

Un **Secure Software Development Life Cycle (SSDLC)** considera e implementa opportune attività di sicurezza nel corso di tutte le fasi del processo SDLC, come illustrato nella figura che segue:



Requisiti: in questa fase vengono effettuate, tramite rappresentazione UML, le analisi dei requisiti di sicurezza, dei rischi, delle probabilità di impatto delle minacce, dei casi di abuso. E' importante sottolineare che in questa fase si devono adottare le best practices di carattere generale nella definizione dei requisiti di sicurezza.

Progettazione: in questa fase si esamina il sistema in divenire con l'ausilio di tecniche di analisi e modellazione delle minacce. Requisiti di sicurezza di maggior dettaglio si aggiungono quindi a quelli prodotti nella precedente fase.

Implementazione: in questa fase si realizza il sistema attraverso la stesura di codice sicuro. Seguono l'esecuzione di test di sicurezza basati sull'analisi delle minacce e l'analisi statica del codice sorgente. Quest'ultima può produrre nuovi requisiti di sicurezza, che possono portare alla revisione del codice.

Verifica: in questa fase si analizzano gli aspetti di sicurezza del sistema in esecuzione in un ambiente controllato impiegando tecniche e strumenti di analisi dinamica;

Validazione: è la fase immediatamente prima del rilascio, nella quale viene effettuata una final security review per la verifica del rispetto dei requisiti.

Supporto: in questa fase si esamina il sistema in essere con l'ausilio di tecniche di: analisi e modellazione delle minacce e/o verifica statica/dinamica del codice applicativo, al fine di produrre nuovi requisiti di sicurezza di dettaglio per indirizzare un'eventuale fase di reingegnerizzazione e/o di patching del sistema in oggetto.

6.2 Risk Assessment

L'obiettivo dell'analisi del rischio è da una parte identificare, valutare e misurare la probabilità e la gravità dei rischi (ciò che viene generalmente indicato con il nome di *Risk Assessment*) nei diversi processi dell'organizzazione e, dall'altra decidere come comportarsi a fronte dei rischi identificati (ciò che viene generalmente indicato con il nome di *Risk Management*) al fine di minimizzarli o eliminarli.

Si fornisce di seguito, una classificazione dei principali rischi:

- Rischio strategico, derivante dall'incompatibilità tra due o più dei seguenti fattori:
 - obiettivi strategici,
 - strategie di business,
 - mezzi utilizzati per raggiungere gli obiettivi,
 - quadro macroeconomico nel quale opera l'organizzazione.
- Rischio reputazionale, che può manifestarsi in molteplici situazioni, per esempio in caso di mancato soddisfacimento della clientela.
- Rischio finanziario, derivante dall'incapacità di assolvere gli oneri finanziari assunti.
- Rischio operativo, che è connesso ai processi utilizzati per definire le strategie.
- Rischi di compliance, derivanti da inadempienze legislative (normative e regolamenti).
- Rischi di gestione delle informazioni, derivanti da un insufficiente livello di sicurezza dei sistemi informatici.
- Rischi emergenti e/o potenziali che potrebbero danneggiare il business dell'organizzazione e/o le persone che vi operano.

La gestione del rischio comprende tre attività principali:

- Risk Assessment che include l'identificazione e la valutazione dei rischi e degli impatti; le raccomandazioni e le misure per la riduzione del rischio;
- Mitigazione del rischio, che si riferisce alla prioritizzazione, implementazione e mantenimento delle misure appropriate per la riduzione del rischio raccomandate dal processo di Risk Assessment;
- Valutazione e analisi dei processi e delle misure per l'implementazione di un programma di gestione del rischio di successo (vedi paragrafo 6.2.1).

Una metodologia di gestione del rischio ben strutturata, se utilizzata in modo efficace, può aiutare l'organizzazione a identificare i controlli adeguati per garantire le capacità di sicurezza essenziali.

Ridurre al minimo l'impatto dei rischi sull'organizzazione e fornire solide basi nel processo decisionale sono i motivi fondamentali per cui le organizzazioni sono chiamate a implementare un processo di gestione dei rischi per i loro sistemi IT.

Il Risk Assessment è uno strumento di analisi, semplice e accurato, che studia i rischi dell'organizzazione (operativi, strategici, finanziari ed esterni) al fine d'individuare successivamente le soluzioni e le misure più adeguate. I passi fondamentali del Risk Assessment possono riassumersi come segue:

- Identificazione dei rischi. Devono essere individuati i fattori di pericolo per l'organizzazione, evidenziando chi o cosa può essere danneggiato e in quale modo. Per ogni fattore di pericolo identificato, bisogna definire ciò che è esposto maggiormente al pericolo.
- Valutazione dei rischi e definizione delle azioni di mitigazione. È necessario valutare le azioni e le tecniche per ridurre il pericolo e portarlo a livelli accettabili.
- Annotazione dei risultati e attuazione del piano di mitigazione del rischio. La valutazione precedentemente effettuata va trasformata in un piano operativo, per ottenere una gestione consapevole dei rischi dell'organizzazione.
- Revisione periodica della valutazione e aggiornamenti. È necessario rivedere periodicamente ciò che si sta facendo. Viene identificato il profilo di rischio e viene proposto un modello di gestione integrato dei pericoli, che evidenzia i singoli fattori di rischio. In seguito vengono valutate le varie misure preventive, agevolando la protezione del valore dell'ente.

Si riporta di seguito uno schema per il *Risk Assessment*:

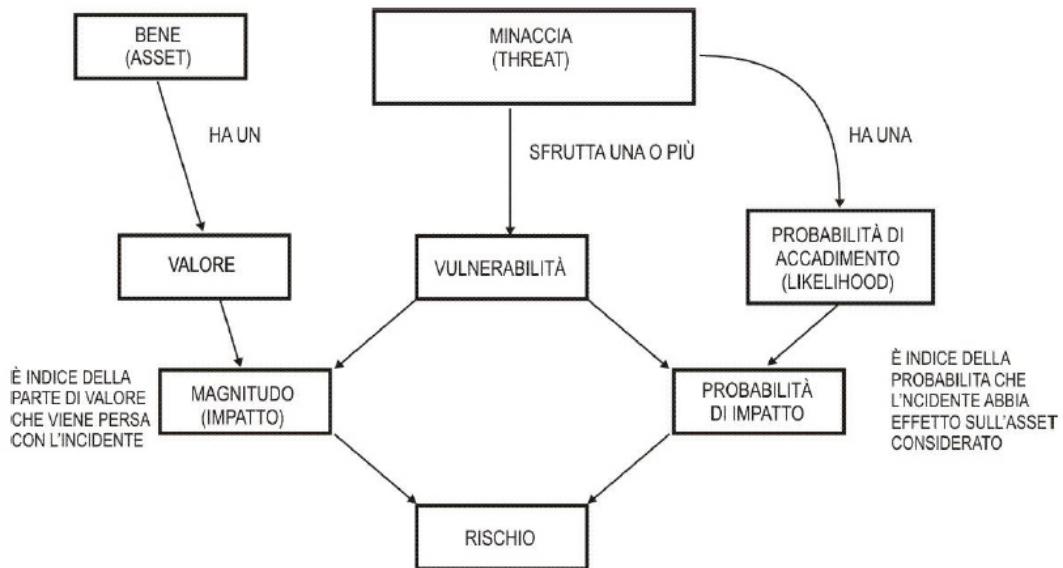


Figura 10 - Esempio di Schema di Risk Assessment

La gestione dei rischi per essere effettivamente efficace, deve essere totalmente integrata nell'SDLC:

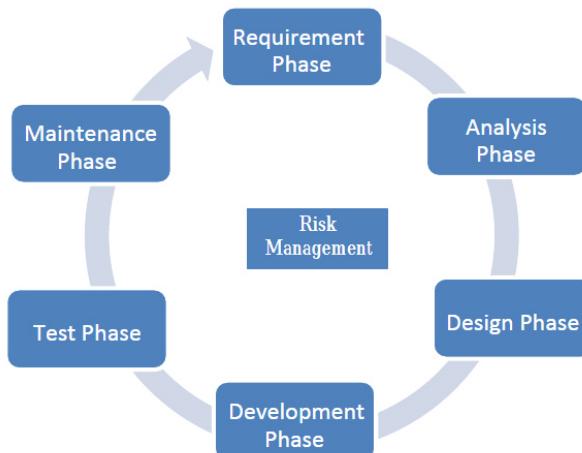


Figura 11 - Gestione del rischio nel ciclo di vita del Software

- Avvio del progetto/Requisiti. La valutazione preliminare del rischio è volta a definire l'ambiente di minaccia in cui opererà il prodotto o il sistema. Questa valutazione è seguita da una prima identificazione dei controlli di sicurezza richiesti che devono essere soddisfatti per proteggere il sistema nell'ambiente operativo previsto.
- Disegno. I requisiti di sicurezza del sistema vengono identificati attraverso un processo formale di Risk Assessment. L'analisi parte dalla valutazione del rischio effettuata nella fase precedente di avvio/inizializzazione e viene approfondita per il contesto specifico. Durante questa fase vengono rivisti attentamente i requisiti e le aspettative di sicurezza e privacy al fine di identificare problemi di sicurezza e rischi per la privacy. In questo passaggio vengono identificate le vulnerabilità presenti nell'ambiente software o derivanti dall'interazione con altri sistemi (Security Assessment). Una volta identificati i rischi, devono essere valutati in merito alla loro potenziale gravità dell'impatto e alla probabilità che si verifichino (Risk Assessment). Nel processo di valutazione è necessario definire le priorità per l'attuazione del piano di gestione dei rischi. La mitigazione del rischio (Risk Mitigation) è il piano delle azioni volte a ridurre o eliminare le priorità più alte. Lo scopo è di valutare la progettazione del sistema, i requisiti dichiarati e i requisiti minimi di sicurezza derivanti dal processo di categorizzazione della sicurezza al fine di determinarne l'efficacia delle azioni di mitigazione per i rischi previsti. I risultati dovrebbero mostrare come i controlli di sicurezza specifici forniscono la protezione appropriata o evidenziare le aree in cui è necessaria un'ulteriore pianificazione. La valutazione del rischio deve essere eseguita prima dell'approvazione delle specifiche progettuali (design specifications) poiché potrebbe fornire specifiche aggiuntive o ulteriori elementi da valutare per le specifiche identificate (ad esempio si dovrebbe considerare come il sistema potrebbe influenzare altri sistemi a cui sarà direttamente o indirettamente collegato; ciò implica che ci potrebbero essere controlli comuni che devono essere ereditati dall'applicazione in oggetto o ulteriori rischi che devono necessariamente essere mitigati).
- Implementazione. In questa fase è necessario determinare i rischi residui accettabili (le specifiche possono imporre oneri e costi eccessivi se i rischi residui accettabili non sono conosciuti). L'obiettivo del processo di valutazione della sicurezza è verificare che il sistema sia conforme ai requisiti funzionali e di sicurezza e operi all'interno di un livello accettabile di rischio residuo per la sicurezza.
- Monitoraggio continuo. L'obiettivo finale del monitoraggio continuo è determinare se i controlli di sicurezza continuano a essere efficaci nel tempo alla luce degli inevitabili cambiamenti che si potrebbero verificare nel sistema e nell'ambiente in cui opera. La valutazione del potenziale

impatto di tali modifiche sulla sicurezza del sistema è un'attività essenziale per assicurare un monitoraggio continuo e prevenire l'abbassamento del livello di sicurezza del sistema.

6.2.1 Tool per l'analisi del rischio

6.2.1.1 AGID Cyber Risk Management

Cyber Risk Management¹⁶ di AgID è lo strumento nazionale per la valutazione e il trattamento del rischio cyber. Per la protezione dei dati in formato digitale, a garanzia della loro riservatezza, integrità e disponibilità, il tool AgID di Cyber Risk Management identifica le situazioni e i vari ambiti nei quali le informazioni possono venirsi a trovare, consentendo di valutare i rischi per la loro sicurezza.

Lo strumento è stato realizzato per consentire alle pubbliche amministrazioni di analizzare l'esposizione al rischio dei servizi erogati dalle amministrazioni stesse ed in caso predisporre i "Piani di Trattamento del Rischio". AgID, dal canto suo, fornisce supporto alle amministrazioni che ne hanno necessità sia in fase di analisi che nella fase d'implementazione di tali piani, pianificati e realizzati in base ai risultati forniti attraverso la fase di Risk Treatment.

La Pubblica Amministrazione ha la peculiarità di erogare servizi verso i cittadini, verso i propri dipendenti, verso le imprese e verso altre amministrazioni.

L'analisi del rischio può essere svolta sulle singole entità (cioè sulla pubblica amministrazione come entità unica) ed anche su parti di esse, ad esempio sui dipartimenti ritenuti più critici. A essere esaminati sono i servizi erogati dalla pubblica amministrazione in correlazione con i servizi trasversali, cioè quelli utilizzati dalle pubbliche amministrazioni ma forniti da terzi, siano essi appartenenti a una PA oppure no.

Il quadro normativo sul quale AgID ha costruito il processo di Risk Management si basa sulle linee guida e sui principi dettati dallo standard ISO 31000 [DR-3] e sull'Information Risk Assessment Methodology 2 (IRAM2) dell'Information Security Forum (ISF).

La metodologia adottata è di tipo "Gray Box", poiché l'analisi parte da una situazione nota solo in parte. I servizi, ad esempio, non devono necessariamente essere esaminati in dettaglio: le informazioni fornite formano una matrice di correlazione che viene analizzata con un algoritmo sviluppato ad hoc e che fornisce come risultato l'elenco delle minacce con i relativi dettagli. Parte del report può essere visto nell'immagine seguente:

Report dei rischi per categoria di minaccia								
● Attacchi Logici e/o Fisici ● Attacchi al sistema di autenticazione								
Minaccia	Impatto R-I-D	Vulnerabilità	Esposizione alla minaccia	Probabilità di accadimento	Rischio attuale	Propensione al rischio	Opzioni di trattamento	Rischio derivato
Accesso non autorizzato a credenziali di autenticazione valide	ALTO	ALTO	ALTO	ALTO	ALTO	MEDIO	MITIGARE	ALTO
Session Hijacking	ALTO	ALTO	CRITICO	CRITICO	CRITICO	MEDIO	MITIGARE	CRITICO
Sfruttare vulnerabilità nei meccanismi di autenticazione	ALTO	ALTO	CRITICO	CRITICO	CRITICO	MEDIO	MITIGARE	CRITICO
● Attacchi al sistema di comunicazione ● Attacchi fisici								
Minaccia	Impatto R-I-D	Vulnerabilità	Esposizione alla minaccia	Probabilità di accadimento	Rischio attuale	Propensione al rischio	Opzioni di trattamento	Rischio derivato
Attacchi all'infrastruttura fisica dell'organizzazione	ALTO	ALTO	MEDIO	BASSO	BASSO	MEDIO	ACCETTARE	BASSO
Furto o perdita di sistemi informativi	ALTO	ALTO	MEDIO	BASSO	BASSO	MEDIO	ACCETTARE	BASSO
● Azioni non autorizzate ● Compromissione dei sistemi informatici di Terze Parti ● Denial of service ● Errori di configurazione ● Exploit del software ● Information Gathering ● Information leakage ● Malware ● Social engineering								

Figura 12 - Cyber Risk Management di AgID – Report dei rischi per categoria di minaccia

Ad esempio, per valutare l'impatto in termini di riservatezza, integrità e disponibilità delle informazioni, ci si sofferma sugli aspetti economico/finanziario, operativo, reputazionale e legale (compliance).

Le fasi che costituiscono la gestione del rischio effettuato con Cyber Risk Management di AgID sono le seguenti:

- 1) Analisi del contesto. Vengono identificati i servizi erogati e i servizi trasversali utilizzati in ambito pubblica amministrazione. Di ogni servizio viene descritto un profilo di criticità.
- 2) Valutazione di ciascun servizio erogato e da ciascun servizio trasversale in termini d'impatto su riservatezza, integrità e disponibilità delle informazioni trattate.
- 3) Calcolo del rischio attuale, sulla base dei valori di probabilità di accadimento e d'impatto, per ogni minaccia identificata. La fase di Risk Assessment prevede anche l'identificazione delle contromisure da implementare per un'efficace mitigazione del rischio.
- 4) Applicazione delle contromisure previste dal piano di trattamento del rischio, volte a mitigare, accettare o trasferire i rischi individuati.
- 5) Analisi del rischio residuo, cioè la valutazione del rischio che permane, nonostante l'applicazione del piano di trattamento del rischio.
- 6) Fase di monitoraggio dell'intero processo, con eventuale adeguamento in seguito a modifiche del contesto o in presenza di nuove minacce alla sicurezza delle informazioni.

Il tool AGID di Risk Management è gratuito ed a completa disposizione di tutte le Pubbliche Amministrazioni: www.sicurezzait.gov.it

6.3 Requisiti

La fase di analisi e specifica dei requisiti è fondamentale nel ciclo di vita dello sviluppo software.

Di seguito si riportano i linguaggi e gli strumenti utili alla fase di definizione dei requisiti di sicurezza del software.

6.3.1 Linguaggi per la specifica dei requisiti

Un linguaggio di specifica in ambito sicurezza può essere considerato:

- un linguaggio di specifica software utilizzato per indicare gli attacchi (AsmL e UML state charts),
- l'estensione di un linguaggio di specifica software utilizzato per rappresentare gli attacchi (Misuse Cases , Abuse Cases, AsmLSec e UMLintr) e i requisiti di sicurezza (UMLsec, SecureUML, Secure Tropos e Misuse Cases),
- un linguaggio per la specifica degli attacchi (*attack specification language*), per esempio STATL e Snort Rules.

UMLsec¹⁷ è un'estensione di UML per lo sviluppo di sistemi sicuri e usa stereotype, tag e constraint per specificare i requisiti di sicurezza. Gli stereotype servono come etichette per gli elementi del modello UML allo scopo di introdurre informazioni al modello e specificare i vincoli che devono essere soddisfatti da questo. I tag sono associati con gli stereotype e sono utilizzati per specificare in modo esplicito una

¹⁷ <https://en.wikipedia.org/wiki/UMLsec>

semplice proprietà di un elemento del modello. UMLsec definisce 21 stereotype da utilizzare per rappresentare i seguenti requisiti di sicurezza:

- fair exchange (la necessità di uno scambio leale),
- non-repudiation (un'azione non si può negare),
- role-based access control,
- secure communication link,
- confidentiality,
- integrity,
- authenticity,
- freshness of a message (ad esempio nonce),
- secure information flow among components,
- guarded access (uso di protezioni per imporre il controllo di accesso).

Sette di questi stereotype hanno dei tag associati e nove hanno vincoli. Gli stereotype possono essere utilizzati per i diagrammi dei casi d'uso, i diagrammi delle classi, diagrammi di stato, diagrammi di attività, diagrammi di sequenza, i diagrammi e le implementazioni per specificare i requisiti di sicurezza in un modello UML (per le specifiche relative sia ai requisiti, sia al design). Un insieme di **tools** sono stati rilasciati per la modellazione attraverso l'impiego di UMLsec e per la verifica dei modelli così realizzati (utilizzando il model checking).

SecureUML¹⁸ SecureUML è un'altra estensione di UML che si concentra sulle politiche di controllo degli accessi ad un modello basato sui ruoli. Queste politiche possono essere considerate come requisiti di sicurezza. SecureUML propone nove stereotype che possono essere utilizzati per annotare un diagramma delle classi, con informazioni di controllo di accesso basato sui ruoli. SecureUML utilizza l'oggetto Constraint Language (OCL) per specificare i vincoli, le azioni e le autorizzazioni per le risorse. Contrariamente a UMLsec, questi vincoli possono essere specificati in base alle esigenze del singolo componente software.

Snort Rules¹⁹ è un network intrusion detection system (IDS) ampiamente utilizzato. Esso utilizza scenari di attacchi specificati come regole per rilevare gli attacchi attraverso la rete. Una snort rule specifica quale azione deve essere intrapresa se la regola è associata a un pacchetto di rete, gli indirizzi IP di origine e destinazione e le porte, il protocollo della rete osservato, e la direzione del pacchetto di rete. Un certo numero di opzioni possono anche essere specificate. Queste opzioni vanno dalla registrazione di un messaggio alla ricerca di una particolare stringa nel pacchetto.

Secure Tropos²⁰ può essere utilizzato per lo sviluppo di software sicuro ed è un'estensione della metodologia di sviluppo Tropos. Secure Tropos utilizza le nozioni di *actor* (person(s), organization(s), software), *goal* (obiettivi che gli attori vogliono ottenere), *soft goal* (un obiettivo la cui realizzazione non può essere determinata in modo esplicito), *task* (un compito per raggiungere un obiettivo), *resource* (fisica o dati), *security constraint* (specificato come le dichiarazioni di alto livello), *secure goal* (utilizzato per soddisfare un vincolo di sicurezza), *secure task* (un compito per raggiungere un obiettivo di sicurezza), *secure resource* (una risorsa che è connessa a *security constraints*, *secure goal*, *secure task*, oppure a un'altra *secure resource*). Un *actor* può dipendere da un altro *actor* per raggiungere un *goal/soft goal*, per

¹⁸ <https://ieeexplore.ieee.org/document/6997358>

¹⁹ <https://www.snort.org/downloads>

²⁰ <http://www.troposproject.eu/node/301>

svolgere un *task*, o rilasciare una risorsa. La notazione SecureTropos può essere utilizzato per rappresentare vincoli di sicurezza (requisiti) sulle interazioni tra gli attori durante la fase di specifica dei requisiti.

Misuse Cases²¹ è una tipologia di Use Case UML utilizzata per descrivere comportamenti indesiderati del software. Un *misuse case* è avviato da un particolare tipo di attore chiamato *mis-actor* (ad esempio, l'attore con intenti malevoli). *Misuse cases* e *mis-actors* possono essere utilizzati per suscitare più casi d'uso per neutralizzare le minacce poste dai casi di uso improprio. *Misuse cases* e *mis-actors* sono rappresentati in colore nero pieno per distinguerli dai casi d'uso e dagli attori UML. Due relazioni speciali chiamati "prevents" e "detects" mettono in relazione *use cases* e *misuse cases*. Il processo può essere utilizzato in modo graduale per sviluppare un diagramma dei casi d'uso (compresi i *misuse cases*) oppure, se necessario, può essere utilizzato anche in modo iterativo. Secondo tale processo, dovrebbero essere specificati prima gli *use cases* e poi i *misuse cases*. Dopo di che, devono essere identificate le relazioni potenziali tra gli *use cases* e i *misuse cases* perché spesso la funzionalità del software viene utilizzata per attaccarlo. Infine, i nuovi *use case* devono essere specificati per individuare o prevenire i *misuse cases*. Questi nuovi use case costituiscono i requisiti di sicurezza di alto livello del software e sono chiamati come "security use cases".

Abuse Cases²² Un altro modo per specificare il comportamento indesiderato di un pezzo di software utilizzando i diagrammi UML è di sviluppare un *abuse case model*. Un *abuse case model* specifica le interazioni pericolose usando attori e *abuse case*. Non c'è differenza di notazione tra i componenti di un *UML use case diagram* e un *abuse case model*. Si raccomanda l'utilizzo di una struttura ad albero per gli approcci multipli. Questo aggiunge ulteriori dettagli al modello e permette di identificare tutte le possibili misure di sicurezza. Dettagli sugli attori come le loro risorse, le competenze, e l'obiettivo dovrebbero essere inclusi come testo. Gli *abuse case model* possono essere utilizzati nelle fasi di progettazione e collaudo.

UMLintr²³ è un'estensione di UML che utilizza stereotype e tag per specificare intrusioni (attacchi) utilizzando use case diagrams, class diagrams, state charts, package diagrams. Gli attacchi vengono divisi in quattro tipologie diverse. Ogni tipo è rappresentato come un pacchetto fornito di stereotype. Ci sono tre stereotype definiti per le classi e dodici per lo use case diagram. Gli stereotype per le classi hanno anche i tag.

Abstract State Machine Language (AsmL)²⁴ ASML è un linguaggio a stati finiti machine-based eseguibile utilizzato anche per specificare scenari di attacco. In generale, in ASML possono essere specificati attacchi con step multipli. Tali scenari di attacco possono essere tradotti automaticamente in *Snort rules* che possono poi essere utilizzati con un'estensione di IDS Snort; sono altresì in grado di catturare più attacchi con step multipli, utilizzando le informazioni di contesto. Le Snort rules, l'input standard di Snort, non possono rappresentare attacchi con step multipli.

AsmLSec²⁵ è un'estensione di ASML sviluppata per specificare scenari di attacco. AsmLSec utilizza stati, eventi e transizioni per rappresentare gli attacchi. Ogni transizione ha un'origine e uno stato di destinazione, una serie di condizioni da soddisfare e le azioni da compiere. Gli scenari di attacco rappresentati in AsmLSec possono essere tradotti automaticamente in ASML attraverso un compilatore appositamente sviluppato. E' stato sviluppato un IDS che prende in input gli scenari di attacco tradotti.

²¹ https://en.wikipedia.org/wiki/Misuse_case

²² https://en.wikipedia.org/wiki/Abuse_case

²³ <https://ieeexplore.ieee.org/document/1607377>

²⁴ <https://www.microsoft.com/en-us/research/project/asml-abstract-state-machine-language/>

²⁵ <https://ieeexplore.ieee.org/document/4159874>

UML State Charts for Security²⁶ i diagrammi di stato UML (senza alcuna estensione) sono stati utilizzati per specificare gli attacchi, che a loro volta possono essere collegati alle snort rules. Questi diagrammi di stato possono essere tradotti manualmente nelle snort rules, che poi potranno essere utilizzati con un'estensione di IDS Snort. Attraverso l'impiego dei diagrammi di stato, è possibile rappresentare attacchi complessi con step multipli che normalmente non possono essere rappresentati con snort rules ordinarie.

STATL²⁷ sta per “State Transition Analysis Technique Language” e utilizza due costrutti principali per specificare un attacco: stato e transizione. Ogni transizione deve avere un evento associato che, quando si verifica, avvia la transizione. All'avvio le transizioni possono eseguire azioni facoltative. Stato e transizione specifiche possono anche avere del codice eseguibile al loro interno. Un ambiente di sviluppo per STATL è inoltre disponibile e può essere utilizzato, tra le altre cose, per visualizzare scenario di attacco specificati come macchina a stati.

6.3.2 Tool per la specifica dei requisiti

Il CATALOGO SECURITY TOOLS 6.9 raccoglie i tool disponibili, divisi per fase del processo SSDLC, che offrono funzionalità applicabili in ambito secure application development.

Si riporta di seguito la tabella ‘Software Requirements Tools’:

Prodotto	Categoria	Fase SSE	Tipo Licenza	Sito Web
CaseComplete	Requirements management	Requirements	Versione trial disponibile su richiesta	https://casecomplete.com/
IBM Engineering Requirements Management DOORS Next	Requirements management	Requirements	Versione trial non disponibile	https://www.ibm.com
IBM Rational RequisitePro solution	Requirements management	Requirements	Versione trial non disponibile	https://www.ibm.com
Microfocus Atlas	Requirements management	Requirements	Versione trial disponibile	https://www.microfocus.com/
Objectives	Requirements management	Requirements	Versione trial disponibile	http://www.objectiver.com
Open Source Requirements Management Tool (OSRMT)	Requirements management	Requirements	Open Source	http://sourceforge.net/projects/osrmt/
Reqtify	Requirements management	Requirements	Demo non disponibile	https://www.3ds.com/it/prodotti-e-servizi/catia/prodotti/reqtify/
rmtoo	Requirements management	Requirements	Open Source	http://rmtoo.florath.net/

²⁶ <https://ieeexplore.ieee.org/document/7042284>

²⁷ <https://pdfs.semanticscholar.org/8e78/63430446f610f5015a484d084ccb7e3c376.pdf>

Simulink Requirements	Requirements management	Requirements	Versione disponibile	Trial	https://it.mathworks.com/products/simulink-requirements.html
Teamcenter Systems Engineering Requirements (TcSE)	Requirements management	Requirements	Versione trial non disponibile		https://www.plm.automation.siemens.com/global/it/products/teamcenter
Telelogic DOORS	Requirements Management	Requirements	Gratuito		http://telelogic-doors.software.informer.com/
Visual Trace Spec	Requirements management	Requirements	Versione trial disponibile		http://visualtracespec.com/#
Visure Requirements Management Tool	Requirements management	Requirements	Versione trial disponibile su richiesta		https://visuresolutions.com/requirements-management-tool/

6.4 Progettazione

La fase di progettazione identifica i requisiti generali e individua la struttura più adatta per la realizzazione del software. In questa fase viene definita l'architettura di sicurezza, adottando le linee guida di progettazione; vengono altresì documentati gli elementi che delimitano la superficie d'attacco e vengono modellate le minacce.

6.4.1 Secure Design Languages

Molti dei linguaggi per specificare i requisiti di sicurezza sono utilizzati anche per le specifiche di design. Ciò è dovuto al fatto che i requisiti di basso livello sono davvero vicini alla progettazione statica e dinamica. Questi linguaggi (ad esempio, UMLsec, SecureUML, e SecureTropos) sono già stati discussi nella sezione precedente. Due sono i principali punti che dovrebbero essere considerati nella scelta di un linguaggio di design sicuro:

- la varietà di schemi disponibili per rappresentare un disegno, comprensivo dei vari aspetti e livelli di astrazione;
- la disponibilità degli strumenti.

UMLsec fornisce una varietà di schemi e ha strumenti disponibili.

SecureUML può essere utilizzato anche per la progettazione di software sicuro; tuttavia, si limita a rappresentare solo nozioni di controllo degli accessi basati sui ruoli in un diagramma delle classi UML.

Secure Tropos propone di utilizzare gli Agent UML capability diagrams. Questi schemi sono simili ai diagrammi di attività UML (piano e capacità) e diagrammi di sequenza (interazione agente).

6.4.2 Software Design Tools

Il CATALOGO SECURITY TOOLS 6.9 raccoglie i tool disponibili, divisi per fase del processo SSDLC, che offrono funzionalità applicabili in ambito secure application development.

Si riporta di seguito la tabella ‘Software Design Tools’:

Prodotto	Categoria	Fase SSE	Tipo Licenza	Sito Web
Coras	Threat Modeling tool/practices	Design	Open Source	http://coras.sourceforge.net/downloads.html
IriusRisk	Threat Modeling tool	Design	C'è una versione (limitata) open source	https://iriusrisk.com/
Microsoft Threat Modeling Tool	Threat Modeling tool	Design	Free	https://www.microsoft.com
ThreatModeler	Threat Modeling tool	Design	Demo disponibile	https://threatmodeler.com/
SeaMonster Security Modeling Software	Threat Modeling tool	Design	Open Source	https://sourceforge.net/projects/seamonster/
TRIKE	Threat Modeling tool/practices	Design	Open Source	http://www.octotrike.org/

6.5 Implementazione

Durante questa fase il team di sviluppatori mette in atto le contromisure secondo le specifiche della fase precedente ed effettua dei test sul codice sorgente per verificare l'assenza di security flaws.

6.5.1 Software Implementation Tools

Il CATALOGO SECURITY TOOLS 6.9 raccoglie i tool disponibili, divisi per fase del processo SSDLC, che offrono funzionalità applicabili in ambito secure application development.

Si riporta di seguito la tabella ‘Software Implementation Tools’:

Prodotto	Categoria	Fase SSE	Tipo Licenza	Sito Web
Brakeman	SAST	Implementation	Open Source	https://brakemanscanner.org/
Burp Suite by PortSwigger	SAST, DAST, Penetration Testing	Implementation / Verification	Versione Community liberamente scaricabile	https://portswigger.net
CppCheck	SAST	Implementation	Open Source	http://cppcheck.sourceforge.net/
Checkmarx	SAST, DAST, RASP	Implementation / Verification	Versione trial disponibile richiesta	https://www.checkmarx.com/
CodeDx	SAST, DAST	Implementation / Verification	Versione trial disponibile	https://codedx.com/
CodeProfiler by Virtual Forge	SAST per applicazioni SAP	Implementation	Nessuna versione trial disponibile	https://www.virtualforge.com

Contrast Enterprise	IAST, RASP	Implementation / Verification	Demo disponibile su richiesta	https://www.contrastsecurity.com
Dependency Check	Library Inspection	Implementation	Open Source	https://www.owasp.org/index.php/OWASP_Dependency_Check
SpotBugs	SAST	Implementation	Open Source	https://spotbugs.github.io/
Gendarme	SAST	Implementation	Open Source	https://github.com/mono/website/blob/gh-pages/docs/tools+libraries/tools/gendarme/index.md
Microfocus Fortify Static Code Analyzer	SAST, DAST, IAST, RASP	Implementation / Verification	Demo disponibile su richiesta	https://www.microfocus.com/it-it/products/static-code-analysis-sast/overview
HCL Security AppScan	SAST, DAST, IAST	Implementation / Verification	Versione trial non disponibile	https://www.hcltech.com
JSHint	SAST	Implementation	Open Source	https://jshint.com/
Klocwork	SAST	Implementation	Versione disponibile su richiesta trial	https://www.perforce.com/products/klocwork
MetaFlows	Cloud Security Scanning	Implementation	Demo disponibile su richiesta	www.metaflows.com
Microsoft BinScope	SAST	Implementation	Free	https://www.microsoft.com
Microsoft Code Analysis Tool	SAST	Implementation	Free	https://www.microsoft.com
Microsoft FxCop	Library Inspection	Implementation	Free	https://www.microsoft.com
Microsoft SDL Regex Fuzzer	SAST	Implementation	Free	https://www.microsoft.com
Microsoft SDL MiniFuzz File Fuzzer	SAST	Implementation	Free	https://www.microsoft.com
ModSecurity	WAF	Implementation / Verification	Open Source	http://modsecurity.org/
N-Stalker Cloud Web Scan	SAST, DAST	Implementation / Verification	Free Tier Available	https://www.nstalker.com
PYLINT	SAST	Implementation	Open Source	https://www pylint.org
PMD	SAST	Implementation	Open Source	https://pmd.github.io
Risk Fabric by Bay Dynamics	Predictive Security Analytics	Implementation / Verification / Response	Demo disponibile su richiesta	https://baydynamics.com
RSA Advanced Threat	DAST	Implementation / Verification	Available by Request	https://www.dellemc.com

Management Solution					
Website Malware Scanner	SAST, DAST	Implementation / Verification	Demo non disponibile		https://www.sitelock.com
SonarLint	SAST	Implementation	Open Source		https://www.sonarlint.org
SonarQube	SAST	Implementation	Open Source		https://www.sonarqube.org
Symantec Advanced Threat Protection	IAST, RASP	Implementation / Verification	Versione disponibile richiesta	Trial su	https://www.symantec.com
Tanium Endpoint Platform	Endpoint Security, App Security Scanning	Implementation / Verification	Demo disponibile	non	https://www.tanium.com
Trend Micro Deep Security Platform	SAST, DAST	Implementation / Verification	Versione disponibile	Trial	https://www.trendmicro.com
Tripwire Enterprise	IAST, RASP	Implementation / Verification	Demo disponibile su richiesta		https://www.tripwire.com
Veracode Cloud Platform	SAST, DAST, Mobile AST, Penetration Testing	Implementation / Verification	Demo disponibile su richiesta		www.veracode.com
WhiteHat Sentinel	SAST, DAST, MAST	Implementation / Verification	Demo di 30 giorni disponibile su richiesta		https://www.whitehatsec.com/info/security-check/

6.6 Verifica

Prima della fase di rilascio definitiva del software i team che lavorano in sicurezza effettuano un ulteriore verifica del codice elaborato mediante test di sicurezza. I test di sicurezza mirano a controllare la vulnerabilità delle superficie di attacco, in modo da agire in via preventiva alla correzione di eventuali problemi che potrebbero verificarsi in fase di rilascio.

6.6.1 Software Verification Tools

Il CATALOGO SECURITY TOOLS 6.9 raccoglie i tool disponibili, divisi per fase del processo SSDLC, che offrono funzionalità applicabili in ambito secure application development.

Si riporta di seguito la tabella ‘Software Verification Tools’:

Prodotto	Categoria	Fase SSE	Tipo Licenza	Sito Web

Acunetix Web Vulnerability Scanner	DAST, IAST	Verification	Versione trial a 14 giorni disponibile	https://www.acunetix.com/
AppSpider Pro by Rapid7	DAST	Verification	Versione trial disponibile	https://www.rapid7.com
BeEF	Penetration Testing	Verification	Open Source	https://beefproject.com/
BrightCloud Threat Intelligence by Webroot	DAST	Verification	Nessuna versione trial disponibile	https://www.brightcloud.com
Burp Suite by PortSwigger	AST, DAST, Penetration Testing	Implementation / Verification	Versione Community liberamente scaricabile	https://portswigger.net
Checkmarx	SAST, DAST, RASP	Implementation / Verification	Versione trial disponibile a richiesta	https://www.checkmarx.com/
Citrix Web App Firewall	WAF	Verification	Demo disponibile su richiesta	https://www.citrix.com/it-it/products/citrix-web-app-firewall/
CloudSOC Cloud Access Security Broker (CASB)	Cloud Security Testing/Scanning	Verification	Nessuna versione trial disponibile	https://www.symantec.com/products/cloud-application-security-cloudsoc
CodeDx	SAST, DAST	Implementation / Verification	Versione trial disponibile	https://codedx.com/
Contrast Enterprise	IAST, RASP	Implementation / Verification	Demo disponibile su richiesta	https://www.contrastsecurity.com
Endpoint Privilege Management	Endpoint Security	Verification / Response	Demo disponibile su richiesta	https://www.beyondtrust.com/
Falcon	Endpoint Security	Verification / Response	Versione trial disponibile	https://www.crowdstrike.com
GrayMatter Platform	Penetration Testing, App Security Scanning	Verification	Demo disponibile su richiesta	https://www.reliaquest.com/
HCL Security AppScan	SAST, DAST, IAST	Implementation / Verification	Versione trial non disponibile	https://www.hcltech.com
Kali Linux	Penetration Testing	Verification	Open Source	https://www.kali.org/
LogRhythm Security Intelligence	Predictive Security Analytics	Verification / Response	Demo disponibile su richiesta	www.logrhythm.com

Platform				
Malwarebytes Endpoint Security	Endpoint Security	Verification	Versione disponibile trial	https://www.malwarebytes.com/business/endpointsecurity/
MetaDefender	Predictive Security Analytics	Verification / Response	Available by Request	https://metadefender.opswat.com/
Metasploit by Rapid7	Penetration Testing	Verification	Open Source	https://www.metasploit.com/
Microfocus Fortify Static Code Analyzer	SAST, DAST, IAST, RASP	Implementation / Verification	Demo disponibile su richiesta	https://www.microfocus.com/it-it/products/static-code-analysis-sast/overview
Microsoft Application Verifier	DAST	Verification	Free	https://www.microsoft.com
Microsoft Attack Surface Analyzer	Intrusion Prevention	Verification	Free	https://www.microsoft.com
Microsoft Cloud Security App (MCAS)	Cloud Access Security Broker	Verification	Versione disponibile trial	https://www.microsoft.com/en-us/microsoft-365/enterprise-mobility-security/cloud-app-security
ModSecurity	WAF	Implementation / Verification	Open Source	http://modsecurity.org/
Network Security Monitoring and Management	CDN, App Security Scanning	Verification	Demo non disponibile	https://enterprise.verizon.com/products/security/
NEVIS Security Suite	WAF, Authentication, Identity mngt	Verification	Available by Request	https://www.nevis-security.ch/en/
Next-Generation Firewalls (NGFW)		Verification		
Nikto2	Web Server Scanner	Verification	Open Source	https://www.cirt.net/Nikto2
Nmap	Penetration Testing and Network Mapping	Verification / Response	Open Source	https://nmap.org/
NSFOCUS Web Application Firewall	DAST, WAF	Verification	Demo non disponibile	https://nsfocusglobal.com/web-application-firewall-waf/

N-Stalker Cloud Web Scan	Web	SAST, DAST	Implementation / Verification	Free Tier Available	https://www.nstalker.com
OWASP Zed Attack Proxy (ZAP)	Proxy	Penetration Testing	Verification / Response	Open Source	www.owasp.org
Paloalto Next-Generation Firewall		WAF	Verification	Demo disponibile su richiesta	https://www.paloaltonetworks.com
Peach Fuzzer		Penetration Testing	Verification / Response	Demo disponibile su richiesta	https://www.peach.tech/
Pradeo Security		Mobile AST	Verification	Nessuna versione trial disponibile	https://www.pradeo.com/IT/ protezione-flotta-mobile
Qualys Security & Compliance Suite		DAST, WAF	Verification / Response	Versione trial disponibile	https://www.qualys.com
Risk Fabric by Bay Dynamics		Predictive Security Analytics	Implementation / Verification / Response	Demo disponibile su richiesta	https://baydynamics.com
RSA Advanced Threat Management Solution		DAST	Implementation / Verification	Available by Request	https://www.dellemc.com
Runtime Application Self-Protection		RASP	Verification / Response	Demo disponibile su richiesta	https://www.imperva.com/products/runtime-application-self-protection-rasp/
Samurai Web Testing Framework		DAST, Penetration testing	Verification	Open Source	http://www.samurai-wtf.org/
SRX Series Firewall by Juniper Networks		WAF	Verification	Versione Trial disponibile	https://www.juniper.net/us/en/products-services/security/srx-series/
Sucuri Website Application Firewall		WAF	Verification	Demo non disponibile	https://sucuri.net/website-firewall/
Symantec Advanced Threat Protection		IAST, RASP	Implementation / Verification	Versione Trial disponibile su richiesta	https://www.symantec.com
Synopsys Black Duck Hub		Library Inspection	Verification	Demo disponibile su richiesta	https://www.blackducksoftware.com/
Tanium Endpoint Platform		Endpoint Security, App Security	Implementation / Verification	Demo non disponibile	https://www.tanium.com

	Scanning			
Thunder TPS by A10 Networks	DDoS Protection	Verification / Response	Versione disponibile	Trial https://www.a10networks.com/products/thunder-tps/
Trend Micro Deep Security Platform	SAST, DAST	Implementation / Verification	Versione disponibile	Trial https://www.trendmicro.com
Tripwire Enterprise	IAST, RASP	Implementation / Verification	Demo disponibile su richiesta	https://www.tripwire.com
Trustwave Secure Web Gateway	CDN, DAST	Verification	Demo non disponibile	https://www.trustwave.com-en-us/services/technology/secure-web-gateway/
Trustwave Web Application Firewall	WAF, Penetration Testing	Verification	Demo non disponibile	https://www.trustwave.com
Veracode Cloud Platform	SAST, DAST, Mobile AST, Penetration Testing	Implementation / Verification	Demo disponibile su richiesta	www.veracode.com
VMWare Carbon Black	Endpoint Security	Verification / Response	Demo disponibile su richiesta	https://www.carbonblack.com/
vSentry by Bromium	Endpoint Security	Verification / Response	Demo disponibile su richiesta	www.bromium.com
Website Malware Scanner	SAST, DAST	Implementation / Verification	Demo non disponibile	https://www.sitelock.com
WhiteHat Sentinel	SAST, DAST, MAST	Implementation / Verification	Demo di 30 giorni disponibile su richiesta	https://www.whitehatsec.com/info/security-check/
Wireshark	Penetration Testing and Packet-level Monitoring	Verification	Open Source	https://www.wireshark.org/
Yottaa	CDN, DDoS Protection, WAF	Verification	Demo disponibile su richiesta	https://www.yottaa.com

6.7 Validazione

Durante questa fase il software è oggetto di una Final Security Review finalizzata a stabilire se il software soddisfa tutti i requisiti di sicurezza individuati nella fase iniziale del progetto.

In questa fase ci si accerta, inoltre, che i bug di sicurezza precedentemente identificati siano stati corretti e che il SW sia sufficientemente robusto di fronte a nuove vulnerabilità.

Le azioni di sicurezza di questa fase possono essere così sintetizzate:

- **Software Remediation dopo un'analisi statica (SAST)**
 - Analisi della reportistica e classificazione degli errori, rilevati nella fase di analisi statica del codice;
 - Rimozione degli errori di sicurezza legati all'uso di librerie esterne vulnerabili, sostituendo queste ultime con le versioni sicure;
 - Ristrutturazione delle classi e funzioni identificate come vulnerabili alle varie injection, al cross site scripting, etc.
 - Applicazione delle modifiche ai costrutti sintattici che rendono il software vulnerabile;
 - Correzione del software in base ai warning sulla qualità del codice;
- **Software Remediation dopo un'analisi dinamica (DAST)**
 - Analisi della reportistica e classificazione degli errori per rilevanza e quindi per priorità e urgenza della loro correzione.
 - Rimozione degli errori messi in evidenza dal fuzzy testing, ad esempio aumentando i controlli applicativi.
 - Correzioni degli errori, eventualmente tramite implementazione di nuove funzioni, per esempio aggiungendo meccanismi di autenticazione o rivedendo la struttura delle classi e funzioni.
 - Adozione di attributi del protocollo per innalzare la sicurezza di cookie e sessioni.
- Definizione di un **Incident Response Plan** cioè la documentazione contenente le istruzioni per rispondere e limitare gli effetti di un incidente di sicurezza.
- Produzione di un documento di Security Review un processo collaborativo che identifica i problemi relativi alla sicurezza, il livello di rischio associato a tali problemi e le decisioni da prendere per ridurre o accettare tale rischio.
- Aggiornamento delle procedure di sicurezza, certificazione del rilascio del software, testing e archiviazione.



Figura 13 - Input e Output della fase Final Review - Secure Release

6.7.1 Software Release Tools

Il CATALOGO SECURITY TOOLS (vedi paragrafo 6.9) raccoglie i tool disponibili, divisi per fase del processo SSDLC, che offrono funzionalità applicabili in ambito secure application development.

Si riporta di seguito la tabella ‘Software Release Tools’:

Prodotto	Categoria	Fase SSE	Tipo Licenza	Sito Web
Armor Complete	Cloud Security Platform	Release	Available by Request	https://www.armor.com

6.8 Supporto

La fase di supporto riguarda la manutenzione e l'assistenza post rilascio. Questa fase nasce per seguire tutte le novità in materia di sicurezza, imposte dal dinamico mercato informatico, per adeguarsi all'evoluzione delle vulnerabilità del software.

Le azioni di sicurezza di questa fase possono essere così sintetizzate:

- **Vulnerability assessment:**
 - esecuzione di test che consentano di individuare le vulnerabilità dell'applicazione;
 - valutazione della priorità/severità dei problemi riscontrati;
 - definizione del Remediation Plan;
 - produzione di reportistica di sintesi e di dettaglio;
- **Data Loss/Leak Prevention:**
 - rilevazione, analisi e classificazione dei dati che transitano nell'organizzazione, ovunque siano archiviati;
 - creazione di regole predefinite per la protezione dei dati, per assicurarsi che siano usati in conformità con le politiche di privacy e sicurezza;
 - generazione automatica di alert nel caso in cui vengano violate le policy di sicurezza definite;
- **Database Security:**
 - analisi dei database e valutazione dei rischi mediante l'accertamento di nuove vulnerabilità;
 - individuazione delle alterazioni dei dati, degli utenti e dei profili di accesso;
 - arresto in tempo reale delle sessioni che violano le policy, evitando che i dati vengano compromessi;
 - applicazione delle ultime patch di sicurezza disponibili;
- **Web Application Firewall Management e Secure Web Gateway:**
 - funzionalità di standard firewall (policy enforcement, stateful inspection, packet filtering, NAT, VPN client-to-site e site-to-site);
 - anti-malware e anti-spam;
 - Intrusion Prevention (IPS) per il blocco delle minacce;
- **Patching Update:** notifica, installazione e test di nuovi security improvement packages.

6.8.1 Software Response Tools

Il CATALOGO SECURITY TOOLS 6.9 raccoglie i tool disponibili, divisi per fase del processo SSDLC, che offrono funzionalità applicabili in ambito secure application development.

Si riporta di seguito la tabella ‘Software Response Tools’:

Prodotto	Categoria	Fase SSE	Tipo Licenza	Sito Web
Airlock Suite by Ergon Informatik	WAF, Authentication, Identity	Response	Versione disponibile trial	https://www.airlock.com
Akamai	CDN, DDoS Protection, WAF	Response	Prova gratuita disponibile	https://www.akamai.com/it/it/
Alert Logic SIEMless Threat Management	Intrusion Prevention System, Cloud Access Security Broker, WAF, Container Security	Response	Versione disponibile trial	https://www.alertlogic.com/
AWS WAF	WAF	Response	Nessuna disponibile trial	https://aws.amazon.com/it/waf/
Potection Center	Mobile AST	Response	Nessuna demo disponibile	https://appmobi.com
AppWall by Radware	WAF, DDoS Protection	Response	Nessuna versione trial disponibile	https://www.radware.com/
Arbor DDoS Protection	DDoS Protection	Response	Nessuna versione trial disponibile	https://www.netscout.com/arbor-ddos
Arxan Application Protection	Mobile AST	Response	Nessuna versione trial disponibile	https://www.arxan.com/application-protection
Barracuda Web Application Firewall	WAF	Response	Versione disponibile trial su richiesta	https://www.barracuda.com/products/webapplicationfirewall
Lookout Mobile Endpoint Security	Mobile Access Security Broker	Response	Demo disponibile su richiesta	https://www.lookout.com/products/mobile-endpoint-security
CD Protection by CD Networks	CDN, WAF, DDoS Protection	Response	Nessuna versione trial disponibile	https://www.cdnetworks.com
CipherCloud	Cloud Access Security Broker	Response	Versione disponibile trial	https://www.ciphercloud.com
CloudFlare	CDN, DDoS Protection, WAF	Response	Nessuna versione trial disponibile	www.cloudflare.com

CloudFront by Amazon	CDN, DDoS Protection	Response	Nessuna versione trial disponibile	https://aws.amazon.com/it/cloudfront/
Cloud Access Security Broker (CASB)	Cloud Access Security Broker	Response	Demo gratuita a richiesta	https://umbrella.cisco.com/products/casb
CloudPassage Halo	Cloud Access Security Broker	Response	Versione trial disponibile	https://www.cloudpassage.com
DDoS Strike by Security Compass	DDoS Protection	Response	Demo disponibile su richiesta	https://www.securitycompass.com
R&S®Web Application Firewall	WAF	Response	Demo disponibile su richiesta	www.denyall.com
F5 Big-IP	WAF, DDoS Protection	Response	Demo disponibile su richiesta	https://f5.com
FireEye NX	Web Server Scanner, WAF	Response	Versione trial non disponibile	https://www.fireeye.com
FortiWeb: Web Application Firewall and API Protection	WAF	Response	Demo disponibile su richiesta	https://www.fortinet.com/products/web-application-firewall/fortiweb.html
FortiGate NGFW	WAF	Response	Demo disponibile su richiesta	https://www.fortinet.com/it/products/next-generation-firewall/models-specs.html
Imperva FlexProtect	WAF, DDoS Protection	Response	Demo disponibile su richiesta	https://www.imperva.com/products/flexportprotect-plans/
BloxOne Threat Defense	WAF	Response	Versione trial disponibile su richiesta	https://www.infoblox.com/products/bloxone-threat-defense/
Hillstone E-Series	WAF	Response	Demo non disponibile	https://www.hillstonenet.com
Kona Site Defender by Akamai	WAF, DDoS Protection	Response	Demo disponibile su richiesta	https://www.akamai.com/it/it/products/security/kona-site-defender.jsp
CenturyLink DDoS and Web Application Security	CDN, Protection	DDoS Response	Demo disponibile non	https://www.centurylink.com/business/security/ddos-and-web-application.html

Netsparker Web Application Security Scanner	DAST	Response	Demo disponibile su richiesta	https://www.netsparker.com/
Neustar	DDoS Protection, WAF	Response	Demo disponibile su richiesta	https://www.home.neustar/
Palo Alto Threat Prevention Services	RASP WAF	Response	Demo disponibile su richiesta	https://www.paloaltonetworks.com
Network Threat Detection	Intrusion Prevention System	Response	Demo disponibile su richiesta	https://www.bricata.com
Sophos Next-Gen Firewall	WAF	Response	Versione Trial a 30 giorni disponibile	https://www.sophos.com/en-us/products/next-gen-firewall.aspx
Sucuri Website Security Solutions	WAF, DDoS Protection, App Security Scanning	Response	Demo non disponibile	https://sucuri.net/website-security-platform/signup/
Ziften	Endpoint Security	Response	Demo disponibile su richiesta	https://ziften.com/

6.9 Catalogo Security Tools

Il CATALOGO SECURITY TOOLS raccoglie i tool disponibili che offrono funzionalità applicabili in ambito secure application development.

In Appendice 1 viene riportato il Catalogo Security Tools con il seguente formato:

Prodotto	Categoria	Fase SSE	Tipo Licenza	Sito Web
nome commerciale del tool	indica la macro-funzione: per es. DAST, SAST, WAF ecc.	la fase del sw life-cycle coperta dal tool	tipo licenza	indirizzo web per approfondimenti

Tabella 4 - Struttura del Catalogo Security Tool

6.10 Training e formazione

Le organizzazioni inoltre dovrebbero investire di più anche nello sviluppo di competenze interne sulla base anche del fatto che molti degli attuali problemi di sicurezza derivano da errori di progettazione o di implementazione, risolvibili solo disponendo di personale qualificato. Alcuni analisti affermano che il 64% degli sviluppatori non sono confidenti di poter scrivere applicazioni sicure [fonte: Microsoft Developer Research].

Questa sezione fornisce un elenco di riferimento dei corsi disponibili in ambito secure software development.

6.10.1 Secure Coding in C and C++

Il corso è basato su materiali di Addison-Wesley: "Secure Coding in C and C++" e "The CERT C Secure Coding Standard". Il training SEI può essere offerto anche fuori dall'area statunitense.

URL	http://www.sei.cmu.edu/training/p63.cfm
Country of HQ location	US
Geographic Scope	International
Type	Academic (SEI)

Questo corso fornisce una spiegazione dettagliata di errori di programmazione comuni in C e C++ e descrive come questi errori possono portare a codice vulnerabile. Il corso si concentra sulle questioni di sicurezza intrinseche dei linguaggi di programmazione C e C++ e delle librerie associate.

I partecipanti acquisiscono conoscenza sugli errori comuni di programmazione che portano a vulnerabilità del software, come questi errori possono essere sfruttati, e le strategie di mitigazione efficaci per impedire l'introduzione di tali errori. In particolare, i partecipanti acquisiscono competenze in merito a:

- migliorare la sicurezza complessiva di ogni tipo applicazione C o C++
- contrastare attacchi buffer overflow e stack-smashing che sfruttano la manipolazione logica di stringhe insicure
- evitare vulnerabilità e security flaws derivanti dal non corretto utilizzo delle funzioni di gestione della memoria dinamica
- eliminare i problemi integer-related: integer overflows, sign errors, truncation errors
- usare correttamente le funzioni di output formattato senza introdurre vulnerabilità format-string
- evitare le vulnerabilità di I/O, tra cui condizioni *race conditions*
- evitare I/O vulnerabilities, including race conditions

6.10.2 Writing Secure Code - C++

Questo corso di formazione computer-based spiega quali sono le funzioni di sicurezza principali del linguaggio C++, come evitare che gli sviluppatori cadano nelle trappole di sicurezza comuni e come costruire applicazioni aziendali sicure e affidabili utilizzando C++. Gli studenti sono guidati attraverso esempi di codice hands-on che evidenziano i problemi e le soluzioni prescritte.

Il corso ha i seguenti moduli:

- Introduction to Software Security
- Data Protection – in Storage and in Transit
- Authentication
- Authorisation
- Data Validation

- Process Handling
- Error Handling and Exception Management
- Logging and Auditing
- Memory Management

6.10.3 Writing Secure Code - Java (J2EE)

Questo corso di formazione computer-based illustra le caratteristiche chiave di sicurezza della piattaforma J2EE, come evitare che gli sviluppatori cadano nelle trappole di sicurezza comuni e come creare applicazioni web sicure e affidabili utilizzando Java. Gli studenti sono guidati attraverso esempi di codice hands-on che evidenziano i problemi e le soluzioni prescritte.

Il corso ha i seguenti moduli:

- Introduction to Software Security
- Data Protection – in Storage and in Transit
- Authentication
- Authorisation
- Data Validation
- Process Handling
- Error Handling and Exception Management
- Logging and Auditing
- Memory Management

6.10.4 Foundstone (Mcafee) Courses

Foundstone offre un programma di formazione di sicurezza di rete per la creazione di professionisti della sicurezza qualificati.

URL	http://www.foundstone.com
Contact Method	http://www.mcafee.com/us/about/contact-us.aspx Email, web form, phone and address
Country of HQ location	US
Geographic Scope	International
Type	Industry (McAfee)

6.10.5 Threat Modeling

Questo corso di formazione computer-based spiega i processi e i concetti di creazione di software sicuro al fine di designare un quadro di sicurezza, identificando quindi minacce e contromisure. Gli studenti possono apprendere come utilizzare la modellazione delle minacce per migliorare il SDLC.

Il corso ha i seguenti moduli:

- Introduction to Threat Modeling and Hacme Books
- Identify Security Requirements
- Understand the System and the Application
- Identify Threats and Countermeasures
- Post-Threat Modeling Activities

6.10.6 Writing Secure Code - ASP.NET (C#)

Questo corso di formazione computer-based spiega le caratteristiche chiave di sicurezza della piattaforma .NET, come evitare che gli sviluppatori web cadano nelle trappole di sicurezza comuni e quindi come creare applicazioni web sicure e affidabili utilizzando ASP.NET. Gli studenti sono guidati attraverso esempi di codice hands-on che evidenziano i problemi e le soluzioni più idonee.

Il corso ha i seguenti moduli:

- Introduction to Software Security
- Data Protection – in Storage and in Transit
- Authentication
- Authorization
- Data Validation
- Process Handling
- Error Handling and Exception Management
- Logging and Auditing
- Memory Management

6.10.7 Oracle Courses

Oracle University è il principale fornitore di formazione per le tecnologie e i prodotti Oracle. Offre corsi class-based, on-site, virtuali e su CD-ROM, molti dei quali si concentrano sulla programmazione Java o sui prodotti Oracle.

URL	http://education.oracle.com
Contact Method	Education Contact Email and phone
Country of HQ location	US
Geographic Scope	International
Type	Industry (Oracle)

6.10.8 Developing Secure Java Web Services, Java EE 6

Il corso Developing Secure Java Web Services fornisce le informazioni necessarie per progettare, implementare, distribuire e gestire secure web services e web service client utilizzando componenti di tecnologia Java e Java Platform, Enterprise Edition 6 (Java EE 6 della piattaforma).

Gli studenti vengono guidati sulla necessità di garantire servizi web sicuri e sulle sfide associate alla sicurezza dei servizi Web. Gli studenti vengono formati anche sui principali standard di settore e sulle iniziative sviluppate per fornire soluzioni di sicurezza complete per i servizi web; nonchè come applicarli per garantire servizi web sicuri. In particolare, gli studenti imparano come proteggere i servizi Web utilizzando tecnologie application-layer security, transport-layer security e message-layer security, come ad esempio come quelle specificate dalle estensioni di sicurezza WS- *.

Questo corso introduce anche i concetti di gestione delle identità, i driver che stanno dietro le soluzioni di gestione delle identità e le funzioni di Sun Java System Access Manager.

Gli obiettivi del corso sono i seguenti:

- Identify the need to secure web services
- List and explain the primary elements and concepts of application security
- Outline the factors that must be considered when designing a web service security solution
- Describe the issues and concerns related to securing web service interactions
- Analyse the security requirements of web services
- Identify the security challenges and threats in a web service application
- Evaluate the tools and technologies available for securing a Java web service
- Secure web services by using application-layer security, transport-layer security and message-layer security
- Describe the concept of identity and the drivers behind identity management solutions
- Explain the role of Sun Java System Access Manager in securing web services
- Secure web services by using UserName token profile
- Secure web services by relying on Sun Java System Access Manager

Il corso tratta i seguenti argomenti:

- Encapsulating the Basics of Security
- Examining Web Services Security Threats and Countermeasures
- Securing Java Web Services Using JavaEE
- Introduction to Web Services Security
- Web Services Security with JAX-WS and Project Metro
- Authentication in JAX-WS
- Identity Management and OpenSSO

6.10.9 MySQL and PHP - Developing Dynamic Web Applications

Il corso MySQL and PHP - Developing Dynamic Web Applications spiega come sviluppare applicazioni in PHP e come usare MySQL in modo efficiente per le applicazioni. Con un approccio hands-on, questo corso con istruttore migliorerà le capacità di PHP e di come combinarle con collaudate tecniche di gestione di database per creare applicazioni web best-of-breed che siano efficienti, solide e sicure.

Gli obiettivi del corso sono:

- Design web-based applications
- Design schemas based on MySQL
- Use „include files“ to make code easier to maintain
- Use PHP 5 and take advantage of its advanced features
- Build applications, following a precise flow
- Authenticate users in a secure way against a database
- Handle errors in your PHP applications efficiently and elegantly
- Write composite queries using JOINs and subqueries
- Use indexing in order to manipulate large amounts of data efficiently
- Use JOINs to extract data from multiple tables
- Use GROUP BY clauses and aggregate functions
- Write applications whose components can be scaled to meet increased demand
- Build a complete application that includes authentication and session management
- Understand how PHP, MySQL and the Apache web server work together to deliver dynamic web content

Il corso tratta i seguenti argomenti:

- PHP Foundations
- MySQL Foundations
- Manage Databases
- Manage Tables
- SQL SELECT Commands
- SQL Expressions
- SQL DML Commands
- SQL JOINS
- MySQL Database-Driven Web-Based Forms
- Session Handling
- Object-Oriented Programming
- Authentication
- Securing PHP and MySQL

6.10.10 Google Gruyere

Google Code University fornisce un ambiente di laboratorio gratuito chiamato Gruyère²⁸, dove gli studenti possono provare ad hackerare applicazioni web. Gli studenti hanno l'opportunità di fare qualche prova reale di penetrazione, sfruttando esempi reali con complessità crescente. In particolare, gli studenti possono imparare:

- come un'applicazione web può essere attaccata utilizzando vulnerabilità di sicurezza comune, come le vulnerabilità cross-site scripting (XSS) e cross-site request forgery (XSRF)
- come trovare, correggere ed evitare queste vulnerabilità comuni, e altri bug che hanno impattano sulla sicurezza, come ad esempio denial-of-service, la divulgazione di informazioni o l'esecuzione di codice remoto.

6.10.11 OWASP Training Courses

OWASP offre materiali di formazione gratuiti, video e presentazioni, e fornisce opportunità di formazione presso le sue conferenze sulla sicurezza delle applicazioni.

²⁸ <http://google-gruyere.appspot.com/>

7 CERTIFICAZIONI PROFESSIONALI

7.1 GIAC Secure Software Programmer (GSSP) Certification

GSSP Certification Exam coinvolge l'Istituto SANS, CERT CC, diverse agenzie governative statunitensi e aziende leader negli Stati Uniti, Giappone, India e Germania. SANS è il certificatore.

URL	https://www.giac.org/
-----	---

Questa certificazione si concentra sulle questioni reali che stanno dietro le vulnerabilità più comuni e i problemi di sicurezza applicativi.

Gli esami riguardano le tecniche e i linguaggi specifici (Java o .NET) e molte delle domande usano esempi di codice reale. Gli esami aiutano le organizzazioni a soddisfare quattro obiettivi, che sono:

- identificare carenze nella conoscenza della sicurezza dei programmatori in-house e aiutare gli individui a colmare il divario;
- assicurarsi che i programmatori in outsourcing abbiano adeguate competenze Secure-coding;
- nominare nuovi dipendenti che non hanno bisogno di formazione correttiva in programmazione sicura;
- assicurarsi che ogni grande progetto di sviluppo abbia almeno una persona con avanzate capacità di programmazione sicura.

Dopo l'acquisizione di questa certificazione, i programmatori saranno a conoscenza dei difetti più comuni di sicurezza che si trovano in ambienti di programmazione specifici (Java o .NET), e sapranno come evitare questi problemi dovuti principalmente alla vulnerabilità delle applicazioni.

Web Application Defender. La certificazione GIAC Web Application Defender consente ai candidati di acquisire le conoscenze e le competenze di sicurezza necessarie per gestire gli errori comuni delle applicazioni Web che portano alla maggior parte dei problemi di sicurezza.

La certificazione GSSP rimane valida per quattro anni.

7.2 International Council of E-Commerce Consultants (EC-Council) Certifications

L'EC-Council è un'organizzazione member-based che certifica gli individui in varie competenze e-business e di sicurezza delle informazioni.

URL	http://www.eccouncil.org
Contact Method	http://www.eccouncil.org/contact_us.aspx Email, web form, phone and address
Country of HQ location	US
Geographic Scope	International
Type	Industry

I diversi tipi di certificazione offerti dal EC-Council nelle aree SSE-correlate sono descritti nelle sezioni che seguono.

7.3 Certified Ethical Hacker (CEH)

Si tratta di una certificazione riconosciuta e accreditata in conformità ANSI 17024. Il corso si pone l'obiettivo di formare una nuova figura professionale, l'hacker etico, che si dedichi alla difesa della sicurezza informatica. Il principio didattico è di apprendere da un lato le tecniche di intrusione e violazione informatica e dall'altro lato le metodologie di difesa da queste stesse tecniche. CEH dispone di 26 moduli, di cui i seguenti sono collegati a SSE:

- Module 17: Web Application Vulnerabilities
- Module 19: SQL Injection
- Module 24: Buffer Overflows
- Module 26: Penetration Testing Methodologies

7.4 Certified Security Analyst (ECSA)

La certificazione ECSA completa la certificazione CEH (vedi sopra) esplorando la fase analitica di hacking etico. ECSA fa un ulteriore passo in avanti, rispetto a CEH, approfondendo come analizzare l'esito di questi strumenti e tecnologie. Attraverso metodi e tecniche di *penetration testing* la certificazione ECSA aiuta i candidati a effettuare le valutazioni necessarie per identificare e mitigare efficacemente i rischi per la sicurezza delle informazioni dell'infrastruttura.

ECSA ha 47 moduli, di cui i seguenti sono collegati a SSE:

- Module 10: Advanced Exploits and Tools
- Module 11: Penetration Testing Methodologies
- Module 27: Stolen Laptop, PDAs and Cellphones Penetration Testing
- Module 28: Application Penetration Testing
- Module 40: Security Patches Penetration Testing
- Module 41: Data Leakage Penetration Testing
- Module 42: Penetration Testing Deliverables and Conclusion
- Module 43: Penetration Testing Report and Documentation Writing
- Module 44: Penetration Testing Report Analysis
- Module 45: Post-Testing Actions

7.5 Certified Secure Programmer (ECSP)

La certificazione ECSP è destinata ai programmatore e agli sviluppatori software e ha allo scopo di codificare e sviluppare applicazioni sicure durante tutto il ciclo di vita del software.

ECSP dispone di 33 moduli, di cui i seguenti sono collegati a SSE:

- Module 01: Introduction to Secure Coding
- Module 02: Designing Secure Architecture
- Module 03: Cryptography
- Module 04: Buffer Overflows
- Module 05: Secure C and C++ Programming
- Module 06: Secure Java and JSP Programming
- Module 07: Secure Java Script and VBScript Programming
- Module 08: Secure Microsoft.NET Programming

- Module 09: Secure PHP Programming
- Module 10: Securing Applications from Bots
- Module 11: Secure SQL Server Programming
- Module 12: SQL Rootkits
- Module 13: Secure Application Testing
- Module 14: VMware Remote Recording and Debugging
- Module 15: Writing Secure Documentation and Error Messages
- Module 16: Secure ASP Programming
- Module 17: Secure PERL Programming
- Module 18: Secure XML, Web Services and AJAX Programming
- Module 19: Secure RPC, ActiveX and DCOM Programming
- Module 20: Secure Linux Programming
- Module 21: Secure Linux Kernel Programming
- Module 22: Secure Xcode Programming
- Module 23: Secure Oracle PL/SQL Programming
- Module 24: Secure Network Programming
- Module 25: Windows Socket Programming
- Module 26: Writing Shellcodes
- Module 27: Writing Exploits
- Module 28: Programming Port Scanners and Hacking Tools
- Module 29: Secure Mobile Phone and PDA Programming
- Module 30: Secure Game Designing
- Module 31: Securing E-Commerce Applications
- Module 32: Software Activation, Piracy Blocking and Automatic Updates
- Module 33: PCI Compliance and Secure Programming

7.6 Certified Software Security Lifecycle Professional (CSSLP) and Certified Information Systems Security Professional (CISSP)

Il CSSLP ha lo scopo di convalidare le conoscenze di sviluppo software sicuro e di buone pratiche. Il CSSLP è un codice in lingua neutrale e applicabile a chiunque sia coinvolto nel SDLC.

La certificazione è rilasciata dal Consorzio di Certificazione Internazionale Information Systems Security, (ISC)², un'organizzazione globale no-profit specializzata nella formazione e certificazione di professionisti della sicurezza informatica. Esso fornisce prodotti di formazione vendor-neutral.

URL	https://www.isc2.org/csslp/default.aspx
Contact Method	CSSLP Contact [https://www.isc2.org/csslp/default.aspx] Web form CISSP Contact [https://www.isc2.org/cissp/default.aspx] Web form General Contact [https://www.isc2.org/contactus/default.aspx] Web form, phone and address
Country of HQ location	US
Geographic Scope	International
Type	Industry (no profit)

In accordo al (ISC)², il CSSLP è progettato per:

- Stabilire le migliori pratiche, al fine di limitare la proliferazione delle vulnerabilità di sicurezza che derivano da processi di sviluppo insufficienti
- attestare la capacità professionista di mitigare i problemi di sicurezza e dei rischi che circondano lo sviluppo di applicazioni in tutto il SDLC, dalla specifica e progettazione alla realizzazione e manutenzione

I seguenti domini compongono il CSSLP Common Body of Knowledge (CBK), che si concentra sulla necessità di integrare la sicurezza nel SDLC:

- Secure Software Concepts: implicazioni di sicurezza nello sviluppo di software.
- Secure Software Requirements: catturare i requisiti di sicurezza nei raccolti dei requisiti di fase
- Secure Software Design: tradurre i requisiti di sicurezza in elementi di design di applicazioni
- Secure Software Implementation/Coding: unit testing per la funzionalità sicurezza e la resilienza contro gli attacchi, e lo sviluppo di codice sicuro e sfruttare la mitigazione
- Secure Software Testing: test integrati di quality assurance per la funzionalità sicurezza e la resilienza contro gli attacchi
- Software Acceptance: implicazioni per la sicurezza in fase di accettazione del software
- Software Deployment, Operations, Maintenance and Disposal: problemi di sicurezza intorno alle operazioni di steady-state e la gestione del software.

La qualificazione CSSLP è valida per tre anni, dopo di che deve essere rinnovata. Può essere rinnovata rifacendo l'esame o, più comune, con l'acquisizione di crediti formativi professionali (CPE).

Il CISSP, un altro programma di certificazione da (ISC)² con regole simili, è destinato ai professionisti che sviluppano politiche e procedure in materia di sicurezza delle informazioni.

7.7 Certificazioni ISACA (CISA, CISM, CRISC)

Le certificazioni ISACA sono accettate e riconosciute a livello globale e sono destinate al management IT per rafforzare le loro competenze negli ambiti: audit IT, sicurezza, governance e gestione dei rischi. Nel dettaglio:

- Certified Information Systems Auditor (CISA). Certifica le competenze necessarie ad amministrare e controllare l'IT dell'azienda e a compiere un effettivo audit sulla sicurezza dell'organizzazione. La certificazione CISA ha per oggetto le seguenti aree: Processo di audit dei sistemi informatici; IT Governance e Management; Acquisizione, sviluppo e implementazione dei sistemi informatici; Operazioni, mantenimento e supporto dei servizi informatici; Protezione delle risorse informatiche.
- Certified in Risk and Information Systems Control (CRISC), prepara e abilita i professionisti IT alle sfide IT e alla gestione dei rischi aziendali. La certificazione CRISC ha per oggetto le seguenti aree della gestione degli IT Risk: Identificazione, e Valutazione dei Rischi; Risposta ai Rischi; Monitoraggio dei rischi; Impostazione e implementazione dei controlli IT; Monitoraggio e manutenzione dei controlli IT.
- Certified Information Security Manager (CISM). La certificazione CISM ha per oggetto le seguenti aree: Governance della sicurezza delle informazioni; Gestione dei rischi e Conformità; Sviluppo e Gestione dei programmi di Sicurezza delle Informazioni; Capacità di reagire agli incidenti di sicurezza.

La sicurezza IT è indirizzata nella gran parte di queste certificazioni, ma non viene data molta enfasi all'Ingegneria Secure Software.

URL	https://www.isaca.org/CERTIFICATION/Pages/default.aspx
Contact Method	General Contact: [http://www.isaca.org/About-ISACA/Contact-Us/Pages/default.aspx] Web form, phone and address
Country of HQ location	US
Geographic Scope	International
Type	Industry (no profit)

7.8 International Secure Software Engineering Council (ISSECO)

ISSECO promuove corsi di formazione sul SSE per ingegneri del software in modo che possano ottenere uno standard di certificazione (ISSECO Certified Professional for Secure Software Engineering). La certificazione è fornita dall'Istituto Internazionale Software Quality (iSQI)²⁹.

Secondo questa iniziativa, l'attenzione di ISSECO è sulla produzione di software sicuro e il suo obiettivo è quello di creare un ambiente informatico sicuro per tutti. Non è focalizzata su specifici linguaggi di programmazione.

URL	http://www.isseco.org/index.php?p=content
Contact Method	ISSECO Contact: http://www.isseco.org/index.php?p=contact ISQUI Contact: https://www.isqi.org/ Email, phone and address
Country of HQ location	Germany
Geographic Scope	National
Type	Industry (not for profit)

I temi principali della certificazione sono:

- Viewpoints of attackers and customers
- Trust and threat models
- Methodologies
- Requirements engineering with respect to security
- Secure design
- Secure coding
- Security testing
- Secure deployment
- Security response
- Security metrics
- Code and resource protection

²⁹ <https://www.isqi.org/>

Le attività di questa iniziativa sono supportati da partner diversi:

- Supporters (financial aid)
- Training providers (training material and classes)
- Certifiers (certification and certificate quality)

8 SECURE SOFTWARE DEVELOPMENT LIFE CYCLE (SSDLC): ANALISI DELLE METODOLOGIE E DEI PROCESSI

8.1 Life Cycle & Maturity Models

8.1.1 Software Assurance Maturity Model (SAMM)

SAMM è un framework aperto per aiutare le organizzazioni a formulare e attuare una strategia di sicurezza software, che più si adatti ai rischi specifici della particolare organizzazione. Il progetto OpenSAMM, un'attività di OWASP, mantiene e aggiorna la documentazione SAMM.

References	www.owasp.org/index.php/Category:Software_Assurance_Maturity_Model OWASP SAMM Project www.opensamm.org OpenSAMM
-------------------	--

Le risorse fornite da SAMM attraverso il sito web aiutano a:

- Valutare le pratiche di sicurezza software esistenti di un'organizzazione
- Costruire un programma software security assurance in iterazioni ben definite
- Dimostrare miglioramenti concreti al programma di security assurance
- Definire e misurare le attività relative alla sicurezza in tutta l'organizzazione

Essendo un progetto Open, i contenuti SAMM sono liberamente fruibili. Il modello si basa su 4 funzioni aziendali (Governance, Construction, Verification e Deployment) di sviluppo software e di 12 procedure di sicurezza. Ogni funzione all'interno dello sviluppo del software prevede tre pratiche di sicurezza:

- Governance
 - Strategy & Metrics
 - Education & Guidance
 - Policy & Compliance
- Construction
 - Security Requirements
 - Threat Assessment
 - Secure Architecture
- Verification
 - Design Review
 - Security Testing
 - Code Review
- Deployment
 - Environment Hardening
 - Vulnerability Management
 - Operational Enablement

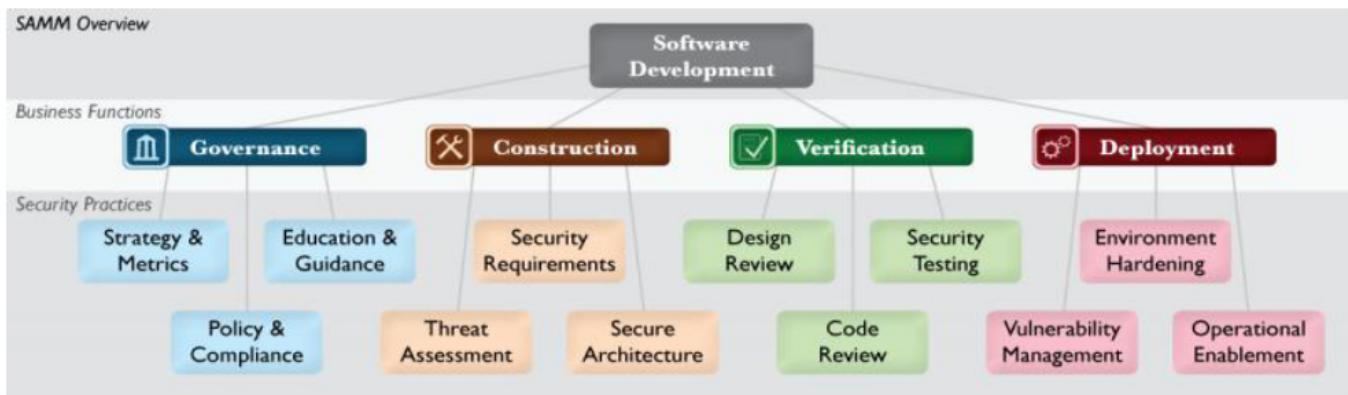


Figura 14 - SAMM Structure

Per ogni security practice, tre Maturity Levels sono definiti in termini di specifiche attività e metriche che un'organizzazione potrebbe adottare al fine di ridurre i rischi per la sicurezza e aumentare l'affidabilità del software.

Risultati più rilevanti:

Maturity Model: SAMM Il modello è disponibile in formato XML ed è stato tradotto in diverse lingue.
version 1.0 Nella stessa pagina sono evidenziati i tool a supporto:
<http://www.opensamm.org/download/>

8.1.2 Systems Security Engineering Capability Maturity Model (SEE-CMM)

Il modello SSE-CMM si indirizza sui requisiti per l'implementazione della sicurezza in un sistema. Le attività di ingegneria di sicurezza coprono l'intero ciclo di vita del sistema (definizione dei concetti, analisi dei requisiti, progettazione, sviluppo, integrazione, installazione, manutenzione e disattivazione). L'SSE-CMM si applica a tutti i tipi di organizzazioni, a prescindere dalle loro dimensioni, da quelle commerciali a quelle di carattere governativo o accademico.

URL	http://www.sse-cmm.org
Country of HQ location	US
Geographic Scope	International
Type	Industry (not for profit)

Questo modello ha undici aree di processo di sicurezza ciascuna delle quali comprende un insieme di pratiche di base. Queste aree si concentrano sui controlli, sulle minacce, sulla scoperta e sull'eliminazione delle vulnerabilità:

- Administer Security Controls
- Assess Impact
- Assess Security Risk
- Assess Threat

- Assess Vulnerability
- Build Assurance Argument
- Coordinate Security
- Monitor Security Posture
- Provide Security Input
- Specify Security Needs
- Verify and Validate Security

Risultati più significativi:

Maturity Model	Capability Maturity Model - http://all.net/books/standards/ssecmmv3final.pdf	Description Document -
Standard	ISO/IEC 21827 - http://www.iso.org/iso/catalogue_detail.htm?csnumber=44716	

8.1.3 Building Security In Maturity Model (BSIMM)

BSIMM non è una guida completa ‘how to’ di sicurezza software, ma piuttosto una raccolta di idee e attività che sono oggi in uso all’interno delle aziende di sviluppo software. Il modello Building Security In Maturity (BSIMM) è uno studio delle iniziative di sicurezza del software in uso all’interno delle aziende che si occupano di sviluppo software. Mettendo insieme le pratiche di molte organizzazioni diverse, è possibile descrivere le misure comuni, quelle condivise da molti, e le peculiarità che rendono unico ogni singolo sistema. Il BSIMM è stato creato attraverso un processo di comprensione e analisi dei dati del mondo reale provenienti dalle esperienze di numerose aziende. Quelle che partecipano allo studio BSIMM provengono da differenti settori verticali, inclusi i servizi finanziari, il software indipendente, la tecnologia, la sanità, l’elettronica di consumo, ecc. Ogni mese al campione si aggiungono nuove aziende. Nove imprese nell’ambito sicurezza software, che sono stati a seguire validati e regolamentati con i dati provenienti da 21 aziende aggiuntive. Il BSIMM mette quindi insieme le esperienze di trenta imprese di sviluppo software - la maggior parte di essi si trovano negli Stati Uniti - che hanno implementato iniziative di sicurezza del software.

URL	https://www.bsimm.com/
Country of HQ location	US
Geographic Scope	International (mainly the US)
Type	Industry

BSIMM ha sviluppato il <https://www.bsimm.com/> (SSF), che fornisce un vocabolario comune per descrivere gli elementi più importanti di un quadro di sicurezza software all’interno di una società.

Sono stati identificati quattro domini e pratiche comuni alla maggior parte delle esperienze. Il BSIMM descrive 109 attività che ogni organizzazione può mettere in pratica. Le attività sono descritte in termini di SSF, che identifica dodici pratiche raggruppati in 4 domini, 3 pratiche di dominio, come mostrato nella figura presa dal documento BSIMM2: di funzionalità, all’interno dei quali sono previste delle attività da svolgere.

The Software Security Framework (SSF)			
Governance	Intelligence	SSDL Touchpoints	Deployment
Strategy and Metrics	Attack Models	Architecture Analysis	Penetration Testing
Compliance and Policy	Security Features and Design	Code Review	Software Environment
Training	Standards and Requirements	Security Testing	Configuration Management and Vulnerability Management

Figura 15 - BSIMM SSF

Per ogni livello di pratica e di maturità vi è un'associazione “one activity - one objective”. I domini sono:

1. Governance - Pratiche che aiutano a organizzare, gestire e calibrare un framework di sicurezza del software. Anche l'addestramento del personale è una pratica da far rientrare nella governance centrale.
2. Intelligence - Un insieme di conoscenze aziendali utili per svolgere attività di sicurezza del software all'interno di un'organizzazione; comprende sia indicazioni proattive sulla sicurezza che modelli di organizzazione delle minacce.
3. SSDL Touchpoints - Pratiche associate all'analisi e alla sicurezza di particolari sviluppi software, artefatti e processi. Tutte le metodologie di sicurezza del software includono queste pratiche.
4. Deployment - Pratiche che si rifanno alla sicurezza della rete tradizionale e alla manutenzione del software. La configurazione del software, la manutenzione e altri problemi ambientali hanno un impatto diretto sulla sicurezza.

Il modello di maturità si presenta come una serie di attività connesse con le pratiche. Gli obiettivi per ogni livello di pratica sono identificati. Gli obiettivi possono essere ulteriormente suddivisi in obiettivi per la pratica/livello e sono associati alle attività. A titolo di esempio, la figura seguente, tratta dal documento BSIMM2, mostra il modello di maturità per la pratica di addestramento del dominio Governance.

GOVERNANCE: TRAINING		
Objective	Activity	Level
▪ promote culture of security throughout the organization	provide awareness training	1
▪ ensure new hires enhance culture	include security resources in onboarding	
▪ act as informal resource to leverage teachable moments	establish SSG office hours	
▪ create social network tied into dev	identify satellite during training	
▪ build capabilities beyond awareness	offer role-specific advanced curriculum (tools, technology stacks, bug parade)	2
▪ see yourself in the problem	create/use material specific to company history	
▪ reduce impact on training targets and delivery staff	offer on-demand individual training	
▪ educate/strengthen social network	hold satellite training/events	
▪ align security culture with career path	reward progression through curriculum (certification or HR)	3
▪ spread security culture to providers	provide training for vendors or outsource workers	
▪ market security culture as differentiator	host external software security events	
▪ keep staff up-to-date and address turnover	require annual refresher	

Figura 16 - Training practice BSIMM

Risultati più rilevanti:

Maturity Model	BSIMM2 - https://www.bsimm.com/download/
----------------	--

8.2 Analisi dei Processi SSDLC

8.2.1 McGraw's Secure Software Development Life Cycle Process

McGraw³⁰ [1] si propone di accrescere il processo SDLC (cascata o iterativo) attraverso l'integrazione di alcune attività SSD. In sostanza, il processo di McGraw si focalizza su:

- incorporazione dei requisiti di sicurezza,
- esecuzione dell'analisi dei rischi durante le diverse fasi di sviluppo,
- applicazione di metodi di security assurance quali test di sicurezza risk-based,
- analisi statica e test di penetrazione.

Il processo suggerisce anche di utilizzare l'analisi dei rischi durante la fase di progettazione. Per la fase di security assurance, McGraw suggerisce di utilizzare gli abuse cases e i requisiti di sicurezza per guidare i test di penetrazione.

³⁰ G. McGraw, Software Security: Building Security In, Addison Wesley, 2006

8.2.2 Microsoft Software Development Life Cycle (MS SDL)

MS SDL è un modello che pone molta attenzione alla fase di specifica dei requisiti durante la quale prevede di interagire con il cliente (end-user), al fine di identificare gli obiettivi e le caratteristiche di sicurezza necessarie.

L'incorporazione di queste caratteristiche/funzionalità di sicurezza sono guidate da standard di settore e criteri di certificazione. Durante la fase di progettazione MS SDL suggerisce di svolgere le seguenti attività: l'identificazione dei componenti critici per la sicurezza, l'identificazione di tecniche di progettazione e linee guida, l'identificazione dei punti di accesso degli attacchi, la modellazione delle minacce e analisi del rischio componente per componente, l'identificazione dei requisiti di sicurezza per mitigare le minacce, l'identificazione dei componenti che necessitano di particolare attenzione durante le fasi di test e review del codice, e i criteri per il completamento del software.



Figura 17 - Microsoft SDL

MS SDL consiglia di seguire gli standard di secure coding nella fase di implementazione. L'accento è posto su:

- test specifici di sicurezza,
- analisi statica del codice utilizzando i tool SDL utili a tale scopo,
- revisione del codice (code review) nell'ultimo step della fase di implementazione. Terminata la fase di implementazione, il software completo viene nuovamente verificato attraverso un ulteriore test di sicurezza che si concentra principalmente sui componenti critici (si esaminano ad esempio, i punti di ingresso alle possibili aree di attacco).

URL	https://www.microsoft.com/en-us/securityengineering/sdl
Contact	https://support.microsoft.com/it-it/contactus/?ws=mscom#tab0 Email, chat, phone and address
Country of HQ location	US
Geographic Scope	International
Type	Industry (Microsoft)

Risultati più rilevanti:

Guidance	Microsoft SDL Process Guidance Questa guida illustra il modo in cui Microsoft applica il SDL ai suoi prodotti e tecnologie. Include requisiti di sicurezza e privacy e raccomandazioni per lo sviluppo di software sicuro. Si rivolge ai modelli di sviluppo Cascade, Spiral, Agile. I responsabili delle politiche IT e le organizzazioni di sviluppo software possono sfruttare questi
----------	--

	<p>contenuti per migliorare l'aspetto di sicurezza e di privacy.</p> <p>Microsoft SDL for Agile Development</p> <p>Microsoft SDL for Line-of-Business Applications</p>
	<p>The Security Development Lifecycle Fornisce una guida attraverso ogni fase della SDL, dall'istruzione e progettazione alla sperimentazione e post-rilascio. Gli autori sono esperti di sicurezza del team Microsoft Security Engineering.</p> <p>Simplified Implementation of the Microsoft SDL Questo documento illustra i concetti chiave e le singole attività di sicurezza che devono essere eseguite per la conformità con il processo Microsoft SDL. Gli aspetti da tenere in considerazione includono ruoli e responsabilità, attività di sicurezza obbligatorie, attività di sicurezza opzionali e processo di verifica della sicurezza dell'applicazione.</p> <p>SDL Quick Security Reference (QSR) Con SDL QSR, il team SDL introduce una serie di documenti di orientamento di base, progettati per affrontare le vulnerabilità comuni, dal punto di vista di molteplici ruoli aziendali: decisori aziendali, architetti, sviluppatori e tester / QA.</p>
	<p>Securing Applications Questa documentazione è rivolta agli sviluppatori di .NET Framework per la scrittura di codice sicuro. Comprende: concetti chiave sulla sicurezza, sicurezza dell'accesso al codice, sicurezza basata sui ruoli, servizi crittografici, gestione delle politiche di sicurezza, best practice sulle politiche di sicurezza, linee guida per la codifica sicura e strumenti di sicurezza.</p>
Tools & Templates	<p>Microsoft SDL Threat Modeling Tool La modellazione delle minacce consente agli architetti di software di identificare e mitigare tempestivamente potenziali problemi di sicurezza, quando risolverli è relativamente facile ed economico. È uno strumento gratuito che richiede Visio. Lo strumento è focalizzato sulle tecniche di analisi del progetto.</p>
	<p>Microsoft SDL Process Template Un modello scaricabile che incorpora automaticamente la policy, il processo e gli strumenti associati a SDL nell'ambiente di sviluppo software di Visual Studio.</p>
	<p>MSF-Agile+SDL Process Template Un modello scaricabile che incorpora automaticamente la policy, il processo e gli strumenti associati alla guida allo sviluppo di SDL per Agile, nel Microsoft Solutions Framework per lo sviluppo di software Agile (MSF-Agile) e nell'ambiente Visual Studio.</p>
	<p>The Microsoft SDL Tools Una mappa degli strumenti e dei modelli gratuiti disponibili per ogni fase SDL.</p>

8.2.3 Appropriate and Effective Guidance for Information Security (AEGIS)

AEGIS^{31 32} [2] [3] è un processo SSDLC basato sul modello a spirale e si concentra sulla specifica dei requisiti di sicurezza, identificando gli elementi principali ed eseguendo l'analisi dei rischi. Le fasi di analisi dei

³¹ I. Flechais, M.A. Sasse, and S.M.V. Hales, "Bringing Security Home: A Process for Developing Secure and Usable Systems," In Proc. of the New Security Paradigms Workshop (NSPW'07), Ascona, Switzerland, ACM Press, 2003, pp. 49-57.

requisiti e di disegno sono strettamente collegati. Il modello propone quattro sessioni di progettazione tra gli sviluppatori e gli stakeholders del software. La prima e la seconda sessione modellano le caratteristiche principali del software e le loro relazioni, identificando i requisiti di alto livello di riservatezza, integrità e disponibilità. Nella terza sessione vengono identificati rischi, vulnerabilità e minacce per il software. La quarta sessione, orientata alla progettazione, indica i requisiti di sicurezza per rimuovere le vulnerabilità identificate.

AEGIS suggerisce anche una metodologia di analisi dei rischi da utilizzare durante le sessioni 3 e 4 finalizzate alla progettazione. Questo metodo di analisi dei rischi ha le seguenti fasi principali:

- Determinazione delle vulnerabilità.
- Determinazione del costo e della probabilità di un attacco in ambiente distribuito (inclusi i ruoli delle persone coinvolte e i task che verranno eseguiti sul software).
- Selezione dei requisiti di sicurezza basate sulle indicazioni dell'esperto di sicurezza.
- Valutazione costi-benefici dei requisiti di sicurezza selezionati.
- Il confronto tra il costo di ogni attacco, commisurato con la probabilità che possa verificarsi, e il costo dei requisiti di sicurezza.
- Selezione dei requisiti di sicurezza sulla base dell'efficacia e dei costi.

8.2.4 Secure Software Development Model (SSDM)

SSDM³³ è un processo che incorpora diverse attività di sicurezza in un modello SDLC a cascata (cascade). Secondo SSDM, la modellazione delle minacce dovrebbe essere eseguita in fase di specifica dei requisiti. Il risultato di questa modellazione dovrebbe essere una check-list contenente tutte le potenziali vulnerabilità e attacchi. Tali elenchi di fatto dovrebbero essere dati in input alla fase di sviluppo.

Dopo la modellazione delle minacce, è necessario definire una policy che indichi chiaramente come saranno raggiunti gli obiettivi di sicurezza prefissati.

Tale policy, come sottolineato dal SSDM, è un insieme di decisioni di gestione di alto livello come ad esempio minimizza l'impatto degli errori in tutto il processo di sviluppo, correggendoli non appena vengono rilevati. I test di penetrazione rappresentano, nel modello SSDM, l'unica attività SSD per la fase security assurance.

8.2.5 Aprville and Pourzandi's Secure Software Development Life Cycle Process

[4]Aprville e Pourzandi³⁴ propongono un processo SSDLC sulla base della loro esperienza, maturata durante lo sviluppo di un software di instant messaging. Secondo il loro processo [5], il primo passo nella fase di specifica dei requisiti è quello di individuare gli obiettivi di alto livello, per quanto riguarda la sicurezza (riservatezza, integrità e disponibilità) del software in fase di sviluppo, considerando il suo ambiente di distribuzione. Per gli obiettivi di sicurezza a basso livello, la modellazione delle minacce dovrebbe essere di supporto nella costruzione di un insieme di requisiti di sicurezza. La priorità di tali requisiti può essere

³² I. Flechais, C. Mascolo, and M.A. Sasse, "Integrating Security and Usability into the Requirements and Design Process," International Journal of Electronic Security and Digital Forensics, Inderscience Publishers, Geneva, Switzerland, 2007, vol. 1, no. 1, pp. 12-26.

³³ A.S. Sodiya, S.A. Onashoga, and O.B. Ajayi, "Towards Building Secure Software Systems," Issues in Informing Science and Information Technology, Informing Science Institute, California, USA, 2006, vol. 3, pp. 635-646.

³⁴ A. Aprville and M. Pourzandi, "Secure Software Development by Example," IEEE Security and Privacy, IEEE CS Press, 2005, vol. 3, no. 4, pp. 10-17.

modificata in base ai risultati dell'analisi del rischio. In fase di progettazione, si raccomanda l'uso di [6]UMLsec³⁵. Per la fase di implementazione, si suggerisce di scegliere un linguaggio di programmazione che meglio soddisfa gli obiettivi di sicurezza. Inoltre, particolare attenzione deve essere posta su come evitare: (i) buffer overflow, (ii) format string vulnerabilities. Essi sottolineano di utilizzare per la crittografia algoritmi già verificati. Per la fase di security assurance vengono indicate le seguenti attività: static vulnerability code scanning, code reviews, ad-hoc unit e system security testing, fuzz testing.

8.2.6 Secure Software Development Model (SecSDM)

SecSDM³⁶ utilizza l'analisi dei rischi nella fase di specifica dei requisiti al fine di dare priorità alla modellazione delle minacce. Gli obiettivi di sicurezza di alto livello quali la riservatezza, l'integrità e la disponibilità sono poi identificati sulla base delle minacce rilevate [7].

In fase di progettazione, vengono identificate e selezionate le funzionalità di sicurezza per mitigare le minacce e raggiungere gli obiettivi di sicurezza. SecSDM propone di seguire standard di secure coding durante la fase di implementazione.

8.2.7 Software Security Assessment Instrument (SSAI)

SSAI³⁷³⁸ raggruppa un insieme di attività che utilizzano determinate risorse e strumenti per lo sviluppo di software sicuro. [8] [9] La prima risorsa che SSAI fornisce è un database online³⁹ che contiene informazioni sulle varie vulnerabilità e le indicazioni per la loro mitigazione. [10] La seconda risorsa SSAI è una security checklist che può essere sviluppata e utilizzata come guida per lo sviluppo sicuro. Sono forniti i dettagli di come redigere una checklist e quali sono gli elementi potenziali che possono essere inclusi⁴⁰. La terza risorsa è un elenco di strumenti accessibili pubblicamente, per la scansione statica del codice. SSAI fornisce anche Flexible Modeling Framework (FMF), uno strumento di modellazione e il property-based testing tool (PBT), che utilizza le proprietà di sicurezza specificate nella security checklist o nel FMF come base dei test per il software.

8.2.8 Hadawi's Set of Secure Development Activities

Hadawi⁴¹ identifica 25 vulnerabilità (common vulnerabilities) da evitare durante lo sviluppo [11]. Egli propone anche una serie di requisiti di sicurezza per le fasi di progettazione e implementazione che, se adottati, aiuterebbero ad evitare queste vulnerabilità [12].

³⁵ J. Juerjens, Secure Systems Development with UML, Springer, 2005.

³⁶ L. Futcher and R.v. Solms, "SecSDM: A Model for Integrating Security into the Software Development Life Cycle," In IFIP International Federation for Information Processing, Volume 237, Proc. of the 5th World Conference on Information Security Education, Springer, 2007, pp. 41-48.

³⁷ D.P. Gilliam, T.L. Wolfe, J.S. Sherif, and M. Bishop, "Software Security Checklist for the Software Life Cycle," In Proc. of the 12th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'03), Linz, Austria, IEEE CS Press, 2003, pp. 243-248.

³⁸ D. Gilliam, J. Powell, E. Haugh, and M. Bishop, "Addressing Software Security Risk and Mitigations in the Life Cycle," In Proc. of the 28th Annual NASA Goddard Software Engineering Workshop (SEW'03), Greenbelt, Maryland, USA, 2003, pp. 201-206.

³⁹ DOVES: Database of Vulnerabilities, Exploits, and Signatures, <http://seclab.cs.ucdavis.edu/projects/DOVES/>. Last Accessed March 2009.

⁴⁰ D.P. Gilliam, T.L. Wolfe, J.S. Sherif, and M. Bishop, "Software Security Checklist for the Software Life Cycle," In Proc. of the 12th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'03), Linz, Austria, IEEE CS Press, 2003, pp. 243-248.

⁴¹ M.A. Hadawi, "Vulnerability Prevention in Software Development Process," In Proc. of the 10th International Conference on Computer & Information Technology (ICCIT'07), Dhaka, Bangladesh, 2007,

Durante la fase di implementazione, l'unica attività SSD è la scelta di un appropriato linguaggio di programmazione (sicuro). Per la fase di security assurance, Hadawi consiglia di utilizzare: (i) security code reviews, (ii) static code analysis tools.

8.2.9 Comprehensive, Lightweight Application Security Process (CLASP)

Comprehensive, Lightweight Application Security Process (CLASP)⁴² identifica un insieme di attività SSD classificate in base ai ruoli svolti durante lo sviluppo. CLASP suggerisce l'impiego di un esperto di sicurezza fin dall'inizio dello sviluppo. Per la fase di specifica dei requisiti, sottolinea la necessità di un'analisi dei rischi e della modellazione delle minacce. L'analisi dei rischi e la modellazione delle minacce devono essere eseguite anche nella fase di progettazione.

CLASP propone di annotare i diagrammi di classe con le informazioni di sicurezza. Nella fase di security assurance, consiglia di effettuare le seguenti operazioni: security code reviews, security code scanning, security testing.

CLASP fornisce anche un elenco di vulnerabilità (common vulnerabilities) con informazioni complete su come e quando possono essere introdotti durante lo sviluppo e come evitarli.

URL	https://www.owasp.org/index.php/CLASP_Concepts
-----	---

Risultati più rilevanti:

Security Process	CLASP version 1.2
------------------	-------------------

8.2.10 Secure Software Development Process Model (S2D-ProM)

S2D-PROM⁴³ specifica molteplici strategie possibili per avanzare da ogni fase di sviluppo all'altra [13]. Alla base di questo processo, c'è l'idea di fornire agli sviluppatori opzioni flessibili. Il processo si propone di condurre l'analisi dei rischi durante le fasi di specifica dei requisiti, progettazione, e implementazione. L'analisi del rischio, secondo S2D-PROM, può essere eseguita in modi diversi per ogni fase di sviluppo. I rischi identificati possono essere mitigati utilizzando varie strategie (ad esempio, definendo le norme di sicurezza o utilizzando meccanismi di difesa).

8.2.11 Team Software Process for Secure Software Development (TSP Secure)

[14]TSP-Secure⁴⁴ garantisce la sicurezza attraverso:

- la pianificazione per la sicurezza,
- la qualità e la gestione della sicurezza in tutto il ciclo di vita dello sviluppo,
- la formazione degli sviluppatori circa gli aspetti relativi alla sicurezza.

⁴² https://www.owasp.org/index.php/CLASP_Concepts

⁴³ M. Essafi, L. Labed, and H.B. Ghezala, "S2D-ProM: A Strategy Oriented Process Model for Secure Software Development," In Proc. of the 2nd International Conference on Software Engineering Advances (ICSEA'07), Cap Esterel, French Riviera, France, 2007, p. 24.

⁴⁴ N. Davis, "Secure Software Development Life Cycle Processes: A Technology Scouting Report", technical note CMU/SEI-2005-TN-024, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, 2005.

Durante la fase di progettazione, il team identifica obiettivi di sicurezza e produce un piano dettagliato come guida per lo sviluppo. Le attività di sviluppo possono includere l'identificazione dei rischi, l'identificazione dei requisiti di sicurezza, la progettazione sicura, le revisioni del codice, gli unit test, i fuzz test e l'analisi statica del codice. Il team può scegliere qualsiasi attività SSD che ritiene necessaria.

Secondo TSP-Secure, un membro del team svolge il ruolo di responsabile della sicurezza, facendosi carico di tutte le problematiche relative alla sicurezza.

9 LINEE GUIDA PER L'ADOZIONE DI UN CICLO DI SVILUPPO SOFTWARE SICURO

Negli ultimi anni, un numero considerevole di applicazioni e sistemi hanno dovuto affrontare gravi minacce alla sicurezza a causa di un significativo incremento nell'impiego di nuove tecnologie disponibili e nel contempo della mancanza di conoscenza e di tecniche di indagine che riguardano la sicurezza informatica. In passato, le problematiche in materia di sicurezza riguardavano essenzialmente i livelli dell'infrastruttura di rete. Attualmente, a causa del crescente utilizzo delle reti e della dominanza del concetto di Internet, come il cloud computing, Software as a Service (SaaS), gli aggressori stanno scoprendo sempre più la presenza di gravi vulnerabilità nel livello applicativo del software.

Il concetto di sicurezza a livello di applicazione è quindi emerso come una attività essenziale da integrare nel processo di sviluppo del software.

La sicurezza dell'informazione richiede una particolare attenzione a causa di un gran numero di vulnerabilità individuate nelle applicazioni/sistemi dichiarate come sicure. Sono ben note la complessità e la difficoltà nel realizzare un'applicazione priva di difettosità e/o vulnerabilità, tuttavia, le organizzazioni che producono hardware e software non possono astenersi nel migliorare i propri processi di sviluppo e adattarli agli attuali scenari. Oltre alle numerose pubblicazioni di ricercatori accademici e industrie del software che evidenziano l'importanza di integrare pratiche di sicurezza nel System Development Life Cycle (SDLC), esiste un paradosso nell'effettiva implementazione. La maggior parte dei centri di sviluppo non attua le raccomandazioni di cui sopra a causa della resistenza ai nuovi processi e al mancato adeguamento di mentalità da parte degli attori che operano nell'ambito del SDLC. E' anche solita una certa resistenza da parte d'ingegneri e sviluppatori nell'accettare che il software/hardware da loro realizzato possa essere soggetto a difetti di sicurezza. Anche i team di sviluppo oramai comprendono l'importanza di un nuovo paradigma di sicurezza per il SDLC, anche se, purtroppo, ciò non è sufficiente.

Per raggiungere i livelli di sicurezza adeguati, è necessaria una conoscenza approfondita e dettagliata delle procedure e delle tecniche di sicurezza da adottare: una Security Policy completa è il giusto riferimento per guidare lo sviluppo della sicurezza e tutti gli attori coinvolti come gli ingegneri hardware, sviluppatori, architetti applicativi, ingegneri software, collaudatori e project leader devono considerarla come una regola imprescindibile.

Questa deve stabilire le opportune indicazioni per ciascuna fase di sviluppo: requisiti, progettazione/architettura, implementazione, collaudo e manutenzione, e deve definire le responsabilità per tutti i ruoli coinvolti nel processo di sviluppo. Deve inoltre, stabilire le regole per la definizione dei requisiti di fase abilitando i principi di sicurezza, come la sicurezza delle informazioni, integrità, privacy, riservatezza, disponibilità delle informazioni, continuità, in base all'ambiente e alle minacce pubbliche che possono in qualche modo coinvolgere il sistema.

Al fine di dare copertura agli aspetti di sicurezza è necessario riunire i team di business, di sviluppo e di sicurezza per comprendere le principali vulnerabilità e le conseguenze sul business causate dal rischio dovuto alla presenza di difetti di sicurezza nella versione finale del prodotto. Poiché il SDLC è un processo di "feed forward" come tale, eventuali errori introdotti in questa fase, saranno poi diffusi nelle fasi successive. Per questo motivo è importante analizzare i rischi per la sicurezza sin dalle primissime fasi del ciclo di sviluppo del software.

L'analisi dei requisiti, rappresenta il primo passo nell'SDLC. Attraverso questo, vengono identificati e definiti gli obiettivi delle specifiche di sicurezza, i metodi necessari per implementarle e l'importanza che queste ricoprono. I requisiti di sicurezza definiscono i requisiti funzionali e non funzionali che devono essere soddisfatti per ottenere le caratteristiche di sicurezza di un sistema IT. Tali requisiti possono essere formulati a diversi livelli di astrazione; al più alto livello, riflettono fondamentalmente solo gli obiettivi di

sicurezza. Un esempio di obiettivo di sicurezza potrebbe essere "Il sistema deve mantenere la riservatezza di tutti i dati classificati come riservati".

I requisiti di sicurezza possono essere distinti in quattro diverse tipologie:

1. **Requisiti funzionali sicuri:** che descrivono i criteri di sicurezza integrati in ciascun requisito funzionale. Tipicamente indicano anche ciò che non deve accadere. Questi possono ad esempio essere derivati da casi di uso improprio.
2. **Requisiti di sicurezza funzionale:** definiscono i servizi di sicurezza che devono essere implementati nel sistema sottoposto ad analisi. Alcuni esempi sono l'autenticazione, l'autorizzazione, il backup, il server-clustering, ecc. Questi possono essere derivati dalle best-practices di sicurezza, dalle politiche adottate e dalle eventuali norme che il sistema stesso deve rispettare.
3. **Requisiti di sicurezza non funzionali:** trattasi di requisiti architetturali legati alla sicurezza, come "la robustezza" o "le prestazioni minime e la scalabilità". Questa specifica tipologia di requisiti è tipicamente derivata dai principi architetturali di secure-design e dagli standard in tale ambito.
4. **Requisiti di sviluppo sicuro:** descrivono le attività richieste durante lo sviluppo del sistema al fine di garantire che il sistema stesso nella sua versione finale sia esente da vulnerabilità. Alcuni esempi possono essere la "classificazione dei dati", le "linee guida di sviluppo sicuro" o la "metodologia di test". Tali requisiti sono derivati da framework metodologici basati su best-practices come "CLASP".

Tutti i requisiti di sicurezza devono essere identificati dall'analista e analizzati dal team di sicurezza come parte dei requisiti funzionali e quindi aggiunti nel documento "Specifiche dei requisiti di sicurezza", in una sezione dei requisiti di sistema o dei requisiti software. Di seguito si riportano alcune delle voci che dovrebbero essere presenti nel documento in questione:

- **Descrizione del prodotto** o sistema e relativo scopo. Definisce il perimetro del prodotto, in termini generali, sia in modo fisico sia logico.
- **Ambiente operativo:** definizione dei vincoli di sicurezza previsti per l'ambiente operativo al fine di facilitare l'identificazione e la formulazione delle premesse sull'uso previsto del prodotto. L'analista deve valutare l'uso dell'ambiente in cui opera il prodotto per verificare se il comportamento dell'utente può in qualche modo compromettere la sicurezza del prodotto stesso. A volte sarebbe necessario definire i criteri di protezione del prodotto e del suo ambiente operativo da adottare.
- **Funzioni di sicurezza di base:** descrizione delle features essenziali per implementare le necessarie politiche di sicurezza organizzativa.
- **Livello di garanzia della sicurezza:** tutti i prodotti devono avere un "Software Security Assurance" e questo, deve necessariamente essere incluso nel documento di specifica dei requisiti di sicurezza.
- **Requisiti normativi:** definizione dei requisiti normativi che il prodotto, lì dove applicabile, deve rispettare.

9.1 Definizione dei requisiti di sicurezza

I principali obiettivi di sicurezza da definire sono:

- **Riservatezza e Integrità.** I due più importanti aspetti della sicurezza sono Riservatezza e Integrità. La Riservatezza significa che le risorse possono essere utilizzate solo dalla parte legittima. L'integrità dei dati significa che devono essere modificabili solo dalle persone autorizzate.
- **Autenticità.** Il terzo requisito di sicurezza principale è l'Autenticità: Message authenticity (o *data origin authenticity*) ed entity authenticity.

- **Non-ripudio.** Garantisce che qualsiasi azione sul sistema non possa essere in seguito rinnegata.
- **Flusso Informativo.** Il livello di sicurezza può avere regole diverse. Generalmente si considerano due livelli: alto (altamente sensibile o altamente attendibile) e basso (meno sensibile o meno attendibile). Laddove componenti di sistema considerati di alto livello interagiscono con parti meno attendibili, si deve garantire che non vi sia alcuno scambio di dati dall'alto verso il basso (vale invece il contrario ossia ci può essere lo scambio di dati dal basso verso l'alto *non up-flow*).
- **Controllo Accessi.** Uno dei requisiti di sicurezza principali è il controllo degli accessi, il che significa che solo un utente fidato può avere accesso a un sistema sicuro. Il **Role-Based Access Control (RBAC)** assicura un meccanismo di controllo degli accessi per tutelare i beni. I privilegi di accesso alle risorse dipendono dal ruolo che assumono nel tempo gli individui all'interno dell'Organizzazione. Ai ruoli sono associati profili che definiscono comandi, transazioni e accessi ai dati. L'assegnazione dei ruoli è centralizzata. **ABAC** (Attribute Based Access Control) fornisce i diritti di accesso in base agli attributi dell'utente, delle risorse a cui si accede e dell'ambiente (contesto operativo, tecnico e persino situazionale in cui si verifica l'accesso alle informazioni). Gli attributi sono insiemi di etichette o proprietà che possono essere utilizzati per descrivere tutte le entità che devono essere considerate ai fini dell'autorizzazione. Le regole di sicurezza possono essere definite per una qualsiasi combinazione di attributi, offrendo la possibilità di creare regole specifiche per particolari risorse. Questa caratteristica rende ABAC particolarmente indicato per essere adottato nei sistemi che richiedono un controllo di accesso granulare come l'Internet of Things.

Le principali azioni di sicurezza da attuare sono:

- **Definizione degli elementi di sicurezza applicativa**, finalizzata alla valutazione dei requisiti relativamente a:
 - Integrità,
 - Autenticità,
 - Riservatezza,
 - Disponibilità,
 - Non-ripudio,
 - Autorizzazione.
- **Definizione dei requisiti di privacy**, attraverso la raccolta strutturata delle seguenti categorie di informazioni:
 - Dati personali,
 - Servizi di terze parti,
 - Policy.
- **Risk assessment**, finalizzato alla valutazione del rischio (vedi Paragrafo 6.2). In questa fase viene definito un profilo di rischio per l'applicazione che include: aree sensibili del software e aree che presentano superfici di attacco suscettibili a determinate minacce; aree del codice ad alto rischio che possono essere vulnerabili a diverse minacce. Viene condotta quindi, una fase critica di comprensione, analisi e classificazione dei vari rischi per l'applicazione. Durante questo processo è utile classificare i vari rischi utilizzando diversi framework di sicurezza quali: OWASP Top 10, SANS CWE Top 25 o OWASP ASVS.
 - **Consolidamento dei Requisiti**, review dei requisiti di sicurezza e privacy a seguito del Risk Assessment;

- A completamento di questa fase è necessario produrre la Reportistica/documentazione completa che sintetizza i risultati per ogni punto precedente.



Figura 18 - Input e Output della fase Risk Assessment

Si evidenzia che in questa fase devono essere tenuti in considerazione anche gli aspetti di integrazione e di interfaccia con eventuali altri moduli dell'ecosistema software. Inoltre vanno considerati i requisiti di sicurezza applicativa di carattere generale: Performance, Password nel codice sorgente, Privilegi esecutivi minimi, Fattore di integrità, Input data validation, Gestione dell'output, etc. (per ulteriori dettagli si rinvia al paragrafo 4.1 “Progettazione e sviluppo dell’Applicazione: direttive standard” del documento **Allegato 2 - Linee Guida per lo sviluppo sicuro di codice**). Tali requisiti di sicurezza applicativa devono essere mutuati in questa fase sulla base dei requisiti, funzionali e non funzionali, individuati.

9.1.1 Identificazione degli strumenti a supporto

Nel paragrafo 6.3.2 è stato presentato lo stato dell’arte dei tool a supporto di questa fase. Di seguito viene fornito un esempio di approccio metodologico per la valutazione dei tool.

La baseline comparativa è costituita da 8 parametri (Software Security Requirements). I tool vengono analizzati sulla base di questi parametri. Il risultato è illustrato nella tabella che segue:

Tools	Fair Exchange	Non-repudiation	Rbac	Secrecy & Integrity	Authenticity	Secure Informat. Flow	Guarded Access	Freshness
RequisitePro	√	X	X	X	X	X	X	√
CaseComplete	√	X	X	X	X	X	X	√
Analyst Pro	√	X	√	X	√	X	X	X
DOORS	√	X	√	X	√	X	√	√
Objectiver	X	X	X	X	X	X	X	√
RDT	X	X	X	X	X	X	X	√
RDD-100	√	X	X	X	X	X	X	√
RTM	X	X	√	√	√	X	√	√
Reqtify	√	X	X	X	X	X	X	√
TcSE	X	√	√	√	√	√	√	√
Atlas	X	√	√	√	√	√	√	√
Visure RMT	X	X	√	X	X	X	X	X

9.2 Progettazione di applicazioni sicure

Le azioni di sicurezza di questa fase possono essere così sintetizzate:

- **Analisi e modellazione delle minacce**, attraverso l'identificazione dei componenti applicativi coinvolti, delle interfacce e degli agenti che potrebbero minacciare il sistema;
- **Analisi della superficie d'attacco e della finestra di opportunità**, allo scopo di individuare le parti del sistema che possono essere esposte ad attacchi e pertanto lo rendono vulnerabile;
- **Piano di mitigation**, attraverso l'identificazione delle contromisure da adottare in questa fase al fine di mitigare le potenziali minacce individuate (utilizzando anche tool automatici e semiautomatici);
- **Secure Design Refactoring**, revisione progettuale che attua le contromisure individuate; produzione di un High Level Design conforme ai principi del Secure by Design;
- Questa fase produce come output finale la Reportistica/documentazione completa che sintetizza i risultati per ogni punto precedente (Specifiche Software comprensive delle contromisure).

Questa fase è inoltre responsabile della revisione dei requisiti di sicurezza individuati nella fase precedente di definizione dei requisiti di sicurezza (paragrafo 9.1).

La figura che segue sintetizza gli elementi in input e l'output prodotto dal processo di Progettazione di software sicuro:

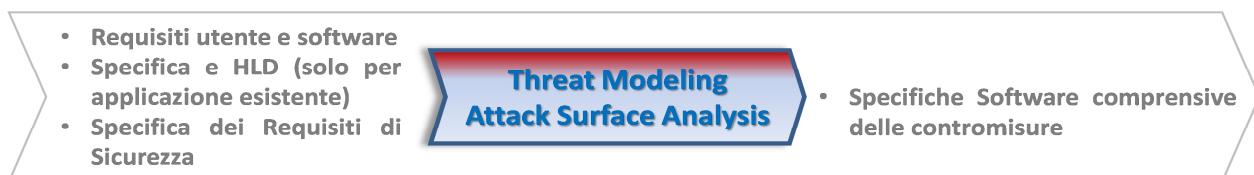


Figura 19 - Input e Output della fase Threat Modeling Attack Surface Analysis

Le linee guida di progettazione sicura sono oggetto del documento **Allegato 4 - Linee Guida per la modellazione delle minacce ed individuazione delle azioni di mitigazione conformi ai principi del Secure/Privacy by Design**. Si rinvia a quest'ultimo per ulteriori dettagli della metodologia da adottare.

9.2.1 Identificazione degli strumenti a supporto

Dopo aver identificato e documentato le esigenze di sicurezza, viene eseguita la modellazione delle minacce col fine di riconoscere e assegnare delle priorità a queste ed individuare le opportune contromisure per la loro mitigazione. A differenza delle tecniche di verifica, come ad esempio il penetration testing, il modello di minacce ottenuto attraverso la relativa modellazione, deve essere eseguito prima che un prodotto o un servizio venga implementato. Questo contribuisce a realizzare un prodotto finale più sicuro indirizzando problematiche di sicurezza ad un early-stage del ciclo di sviluppo. Il processo per la costruzione di un modello di minacce consiste dei seguenti step:

- Disegno dell'architettura del sistema;
- Individuazione dei confini di fiducia;
- Identificazione delle minacce;
- Individuazione delle contromisure da attuare per mitigare le minacce;
- Eventuale riprogettazione dei componenti per mitigare le minacce;

- Convalida del modello architetturale;
- Verifica dell'esistenza di una contromisura per ogni potenziale minaccia identificata.

I tool a supporto di questa fase sono stati identificati nel paragrafo 6.4.2.

9.3 Implementazione di applicazioni sicure

Le azioni di sicurezza che devono essere intraprese in questa fase possono essere così sintetizzate:

- **Data Validation:** verificare la presenza di vulnerabilità che possono riguardare eventuali dati corrotti in ingresso e che possono portare a un comportamento anomalo dell'applicazione;
- **Control Flow:** verificare i rischi collegati all'assenza di specifiche sequenze di operazioni che, se non eseguite nel corretto ordine, possono portare a violazioni sulla memoria o sull'uso scorretto di determinati componenti;
- **Analisi Semantică:** rilevare eventuali problematiche legate all'uso pericoloso di determinate funzioni o API (es. funzioni deprecate);
- **Controllo Configurazioni:** verificare i parametri intrinseci di configurazione dell'applicazione;
- **Buffer Validation:** verificare la presenza di buffer overflow sfruttabile attraverso la scrittura o la lettura di un numero di dati superiore alla reale capacità del buffer stesso.

L'esame del codice sorgente applicativo deve portare alla produzione, mediante la Static Analysis, delle seguenti tipologie di documenti:

- **Report delle Vulnerabilità riscontrate:** report di dettaglio delle vulnerabilità riscontrate nella fase di analisi statica del codice tramite gli strumenti a supporto;
- **Remediation Plan:** report che analizza i falsi positivi ed indirizza la risoluzione delle problematiche riscontrate nell'analisi stessa.

La figura che segue sintetizza gli elementi in input e l'output prodotto dal processo SAST:



Figura 20 - Input e Output della fase Static Analysis

9.3.1 Identificazione degli strumenti a supporto

Nel paragrafo 6.5.1 sono stati presentati i tool a supporto di questa fase. Si riporta di seguito un estratto:

- closed source
 - IBM App Scan (versione SAST),
 - Checkmarx,
 - CodeDx,
 - HP fortify.

- open source
 - SonarQube,
 - FxCop (.NET),
 - BRAKEMAN (Ruby on Rails),
 - PMD (Java, XML e XSL),
 - PYLINT (Python),
 - CppCheck (C/C++),
 - FindBugs (Java),
 - JSHint (Javascript),
 - OWASP Dependency-Check (Java,.NET, Ruby, Node.js, Python, supporto limitato per C/C++).

L'utilizzo combinato dei tool sopra indicati consente una copertura ad ampio spettro semplificando significativamente la revisione manuale che richiederebbe molto tempo.

In Appendice 2 è fornito un approccio metodologico per la valutazione dei tool. L'approccio è basato su una 'Scheda valutazione' che identifica la baseline per l'analisi. I parametri di valutazione includono, ad esempio:

- Linguaggi di programmazione supportati (C/C++, java, JPS, ..);
- Standard supportati (OWASP, Top 10, SANS 25, ..);
- Le vulnerabilità riconosciute (Sql injection, Cross-site scripting, ..);
- L'incidenza dei falsi positivi;
- La capacità di analizzare le dipendenze da librerie esterne;
- Il supporto alla reportistica,
- Altro.

Ogni elemento viene valutato assegnando uno score (da 0 a 10) adeguatamente motivato. La metodologia è stata applicata, a titolo di esempio, su tre tool (i risultati sono indicativi con finalità di linee guida):

Tools	Categoria	Fase	Report
Checkmarx	SAST	Implementation / Verification	Vedi Appendice 2.a
CodeDx	SAST/DAST	Implementation/Verification	Vedi Appendice 2.b
SonarQube/SonarLint	SAST	Implementation	Vedi Appendice 2.c

- 1) **Checkmarx**, è un tool per l'analisi statica del codice, posizionato da Gartner nel quadrante Challengers nell'ambito dell'Application Security Testing (AST). Supporta numerosi linguaggi (vedi scheda nella tabella di cui sopra). Può essere integrato a vari livelli nell'ambito della fase di Implementation: IDE, build server, bug tracking tools.

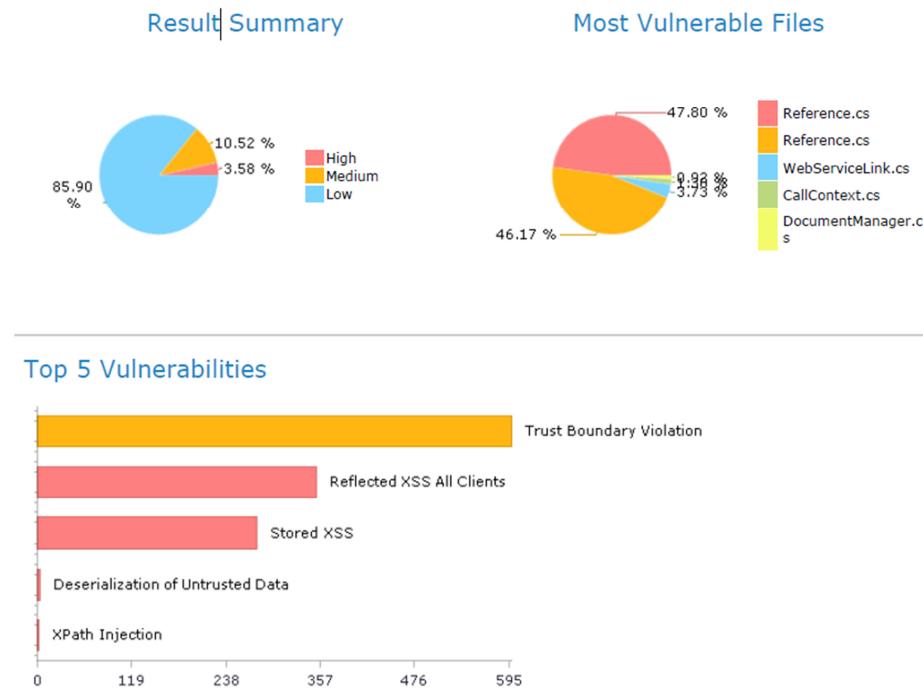
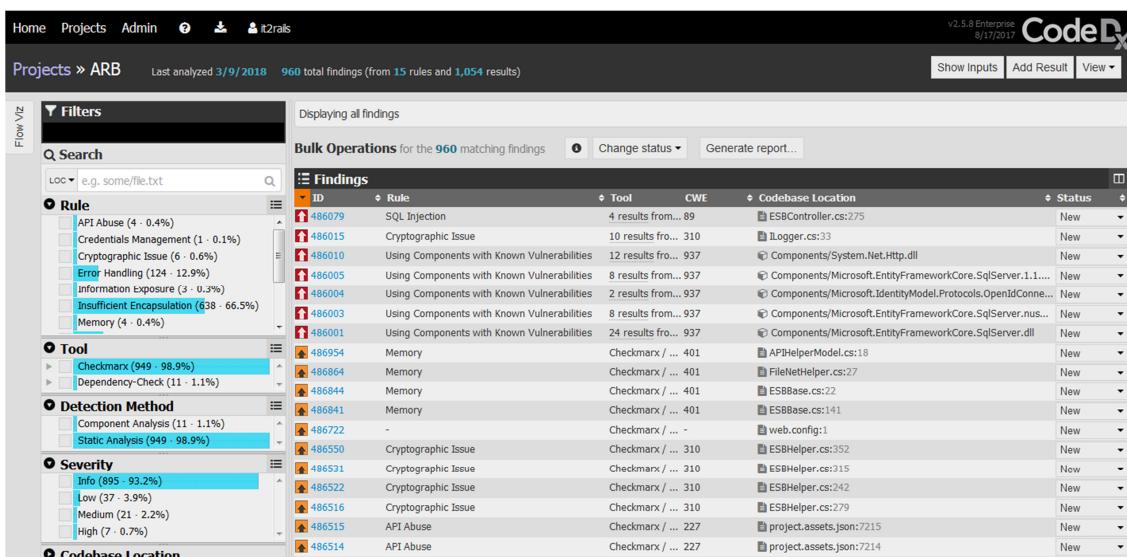


Figura 21 - Report di Checkmarx

- 2) **CodeDx** consente di effettuare la verifica di eventuali vulnerabilità di programmi e software presi in considerazione. CodeDx riunisce una serie di strumenti di analisi del codice (Checkmarx, **IBM App Scan**, **Veracode**), sia gratuiti sia commerciali, che consentono a loro volta di individuare e correggere agevolmente eventuali bug nel codice da analizzare. Uno screenshot dell'interfaccia CodeDx è riportata nella figura che segue:



The screenshot shows the CodeDx interface with the following details:

- Header: Home, Projects, Admin, 8/17/2017, v2.5.8 Enterprise.
- Project: ARB, Last analyzed 3/9/2018, 960 total findings (from 15 rules and 1,054 results).
- Filters: Q. Search, Loc: e.g. some/file.txt, Rule: API Abuse (4 - 0.4%), Checkmarx (949 - 98.9%), Tool: Checkmarx (949 - 98.9%), Dependency-Check (11 - 1.1%), Detection Method: Component Analysis (11 - 1.1%), Static Analysis (949 - 98.9%), Severity: Info (995 - 93.2%), Low (37 - 3.9%), Medium (21 - 2.2%), High (7 - 0.7%), Codebase Location: ESBController.cs:275, Logger.cs:33, Components/System.Net.Http.dll, Components/Microsoft.EntityFrameworkCore.SqlServer.1.1..., Components/Microsoft.IdentityModel.Protocols.OpenIdConne..., Components/Microsoft.EntityFrameworkCore.SqlServer.nus..., Components/Microsoft.EntityFrameworkCore.SqlServer.dll, APIHelperModel.cs:18, FileNetHelper.cs:27, ESBBase.cs:22, ESBBase.cs:141, web.config:1, ESBHelper.cs:352, ESBHelper.cs:315, ESBHelper.cs:242, ESBHelper.cs:279, project.assets.json:7215, project.assets.json:7214.
- Findings Table:

ID	Rule	Tool	CWE	Codebase Location	Status
#486079	SQL Injection	Checkmarx / ...	89	ESBController.cs:275	New
#486015	Cryptographic Issue	Checkmarx / ...	310	Logger.cs:33	New
#486010	Using Components with Known Vulnerabilities	Checkmarx / ...	937	Components/System.Net.Http.dll	New
#486005	Using Components with Known Vulnerabilities	Checkmarx / ...	937	Components/Microsoft.EntityFrameworkCore.SqlServer.1.1...	New
#486004	Using Components with Known Vulnerabilities	Checkmarx / ...	937	Components/Microsoft.IdentityModel.Protocols.OpenIdConne...	New
#486003	Using Components with Known Vulnerabilities	Checkmarx / ...	937	Components/Microsoft.EntityFrameworkCore.SqlServer.nus...	New
#486001	Using Components with Known Vulnerabilities	Checkmarx / ...	937	Components/Microsoft.EntityFrameworkCore.SqlServer.dll	New
#486954	Memory	Checkmarx / ...	401	APIHelperModel.cs:18	New
#486864	Memory	Checkmarx / ...	401	FileNetHelper.cs:27	New
#486844	Memory	Checkmarx / ...	401	ESBBase.cs:22	New
#486841	Memory	Checkmarx / ...	401	ESBBase.cs:141	New
#486722	-	Checkmarx / ...	-	web.config:1	New
#486550	Cryptographic Issue	Checkmarx / ...	310	ESBHelper.cs:352	New
#486521	Cryptographic Issue	Checkmarx / ...	310	ESBHelper.cs:315	New
#486522	Cryptographic Issue	Checkmarx / ...	310	ESBHelper.cs:242	New
#486516	Cryptographic Issue	Checkmarx / ...	310	ESBHelper.cs:279	New
#486515	API Abuse	Checkmarx / ...	227	project.assets.json:7215	New
#486514	API Abuse	Checkmarx / ...	227	project.assets.json:7214	New

Figura 22 - Interfaccia CodeDx

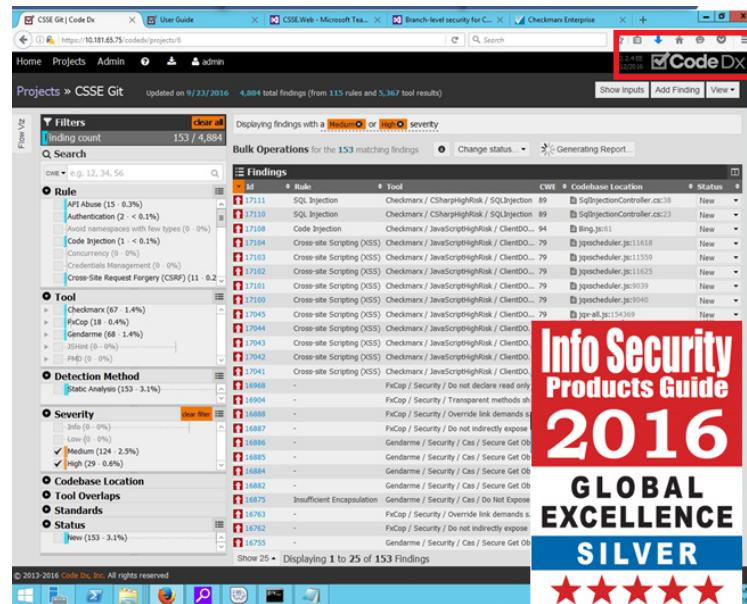


Figura 23 - Info Security Product Guide 2016 : Recensione CodeDX

- 3) **SonarQube**, consente di introdurre il controllo formale fin dall'inizio del ciclo di vita del software, attraverso l'introduzione di Quality Gate nelle fasi tipiche di passaggio tra lo sviluppo e la verifica e tra la verifica e la produzione.

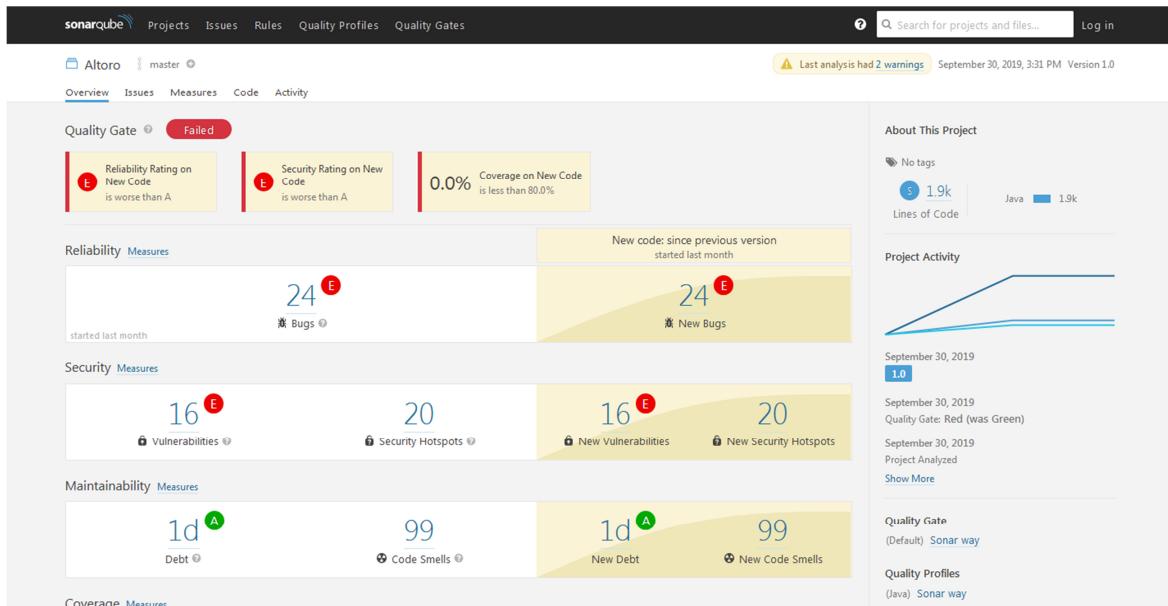


Figura 24 - SonarQube

9.4 Verifica della sicurezza delle applicazioni

Le azioni di sicurezza da intraprendere in questa fase sono così sintetizzate:

- **Analisi dinamica:** attraverso l'attuazione di test dinamici di sicurezza sull'applicazione in esecuzione in ambiente controllato;
- **Penetration Test:** attraverso l'esecuzione di scansioni ed analisi della superficie di attacco;
- **Test di autenticazione multilivello:** attraverso la verifica delle modalità di gestione dell'accesso degli utenti;
- **Business Logic test:** attraverso l'esecuzione di test manuali sulle applicazioni in fase di esecuzione;
- **Analisi dei risultati:** attraverso l'individuazione e la rimozione dei falsi positivi;
- **Remediation Plan:** attraverso la definizione del piano di rientro e la produzione di reportistica di sintesi e di dettaglio; Proof of Concept delle vulnerabilità riscontrate comprensiva di azioni per la riduzione della superficie d'attacco.

L'esame delle Applicazioni in esecuzione in ambiente di test, deve portare alla produzione, mediante la Dynamic Analysis delle seguenti tipologie di documenti:

- **Vulnerability Assessment:** report di dettaglio delle vulnerabilità riscontrate nella fase di analisi dinamica dell'applicazione tramite gli strumenti a supporto;
- **Remediation Plan:** report che analizza i falsi positivi ed indirizza la risoluzione delle problematiche riscontrate nell'analisi stessa.

La figura che segue sintetizza gli elementi in input e l'output prodotto dal processo DAST:



Figura 25 - Input e Output della fase Dynamic Analysis

9.4.1 Identificazione degli strumenti a supporto

Nel paragrafo 6.6.1 sono stati presentati i tool a supporto di questa fase. Si riporta di seguito un estratto:

- closed source
 - IBM App Scan (versione DAST),
 - Veracode,
 - CodeDx.
- open source
 - OWASP Zed Attack Proxy.

9.5 Supporto per la manutenzione di applicazioni sicure

L'obiettivo di questa fase è mantenere un prodotto sicuro, a partire dai nuovi trend sugli attacchi/minacce. Il team deve quindi analizzare le nuove minacce e individuare le contromisure necessarie rilasciando nuovi aggiornamenti/patch laddove necessario attraverso un processo di 'Continuous Security':

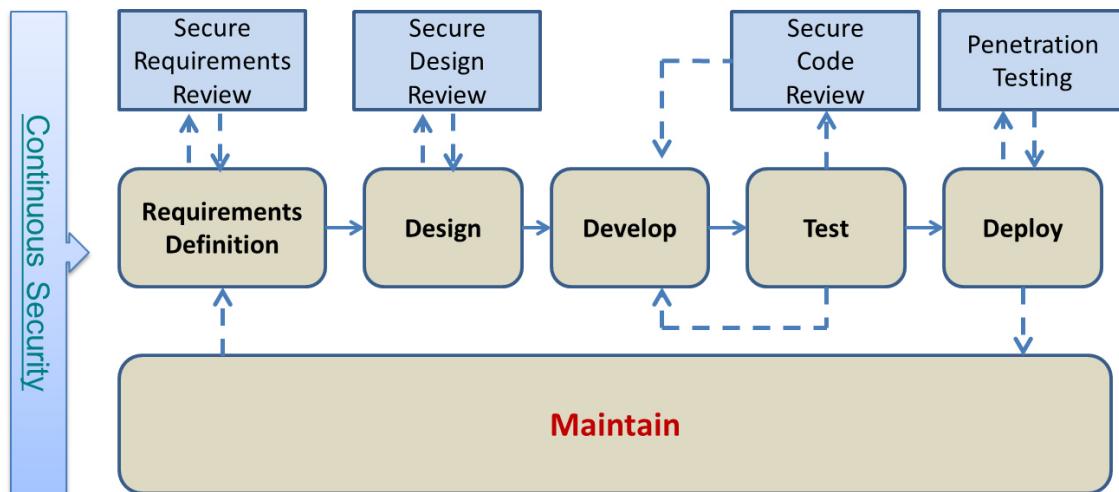


Figura 26 - Continuous Security

Qualsiasi modifica a un sistema ha il potenziale per ridurre l'efficacia dei controlli esistenti o per avere in qualche modo un impatto sulla riservatezza, sulla disponibilità o sull'integrità dello stesso. La soluzione è garantire che nella valutazione delle modifiche del sistema sia inclusa una fase di valutazione del rischio (paragrafo 6.2). Sfortunatamente, non solo i sistemi, ma anche le minacce possono cambiare. Quando vengono identificate nuove minacce, potrebbero essere necessari nuovi controlli per portare il rischio a un livello accettabile. Questo è il motivo per cui le valutazioni periodiche del rischio sono importanti, anche quando un sistema cambia raramente. La valutazione del rischio può fornire un ulteriore vantaggio in per migliorare l'efficacia di politiche, procedure e formazione.

9.5.1 Identificazione degli strumenti a supporto

In ottica di un processo di 'Continuous Security', in questa fase vengono attuate di nuovo le azioni afferenti alle diverse fasi di: Revisione dei requisiti di sicurezza, revisione dei risultati di progettazione, revisione degli aspetti di sicurezza del codice sorgente implementato, penetration test del codice rilasciato.

Gli strumenti per le fasi sopra menzionati sono stati già identificati e indicati nei precedenti paragrafi:

- Definizione dei requisiti di sicurezza [Par. 9.1.1];
- Progettazione di applicazioni sicure [Par. 9.2.1];
- Implementazione di applicazioni sicure [Par. 9.3.1];
- Verifica della sicurezza delle applicazioni [Par. 9.4.1].

10 LINEE GUIDA PER L'IMPLEMENTAZIONE DELLA PRIVACY BY DESIGN NEL SDLC

10.1 Introduzione e concetti base

10.1.1 Principi della Privacy

All'interno dalla ISO/IEC 29100:2011 sono descritti undici principi che indirizzano la progettazione, lo sviluppo e l'implementazione dei requisiti di protezione della privacy (10.1.4). Questi principi, sono anche un riferimento per quel che concerne il monitoraggio e la misurazione delle prestazioni del software e per gli aspetti del controllo dei programmi di gestione della privacy in un'organizzazione (vedere anche paragrafo 5.8.1 dell'*Allegato 4 - Linee Guida per la modellazione delle minacce e individuazione delle azioni di mitigazione conformi ai principi del Secure/Privacy by Design*).

Principi	Descrizione
Consenso e scelta	Secondo questo principio, l'interessato deve poter scegliere se acconsentire o meno al trattamento dei propri dati personali (Consenso Informato). Aderire a questo principio significa fornire all'interessato - in maniera chiara, facilmente comprensibile, accessibile e conveniente - i meccanismi per esercitare la scelta e fornire il consenso in relazione al trattamento dei suoi dati personali al momento della raccolta, al primo utilizzo o non appena possibile.
Scopo legittimo e specifico	Il principio di legittimità e specificità dello scopo assicura che quest'ultimo sia conforme alla legge applicabile e si basi su una base giuridica ammissibile.
Limitazione della raccolta	Limita la raccolta dei dati personali a ciò che è strettamente necessario per gli scopi specificati.
Minimizzazione dei dati	Prevede la progettazione, l'implementazione e l'elaborazione dei dati, attraverso procedure o sistemi ICT, in modo da ridurre al minimo i dati personali che vengono elaborati e il numero di parti interessate dalla privacy.
Limitazione dell'utilizzo, conservazione e divulgazione	Limita l'utilizzo, la conservazione e la divulgazione (incluso il trasferimento) dei dati personali a scopi specifici, esplicativi e legittimi del trattamento.
Precisione e qualità	Assicura che i dati personali elaborati siano accurati, completi, aggiornati (a meno che non vi sia una base legittima per mantenere dati obsoleti), e adeguati e pertinenti ai fini del trattamento.
Apertura, trasparenza e preavviso	Tale principio prevede di fornire informazioni chiare e facilmente accessibili sulle politiche stabilite dal titolare del trattamento e sulle procedure relative al trattamento dei dati personali.
Partecipazione individuale e accesso	Stabilisce che agli interessati sia data la possibilità di accedere e di rivedere i propri dati personali, a condizione che la loro identità sia

	autenticata con un livello adeguato di garanzia e che tale accesso non sia vietato dalla legge applicabile.
Responsabilizzazione	Stabilisce che siano documentate e comunicate in modo appropriato tutte le politiche, le procedure e le pratiche relative alla privacy. Prevede altresì l'assegnazione ad un individuo specifico all'interno dell'organizzazione del compito di attuare le politiche, le procedure e le best practice relative alla privacy.
Sicurezza delle informazioni	Stabilisce la protezione dei dati personali con controlli appropriati a livello operativo, funzionale e strategico. Al fine di garantire l'integrità, la riservatezza e la disponibilità dei dati personali e proteggerli dai rischi (quali l'accesso non autorizzato, la distruzione, l'utilizzo non consentito, la modifica, la divulgazione o la perdita in tutto il ciclo di vita dell'informazione).
Conformità alla privacy	Stabilisce di verificare e dimostrare che il trattamento rispetti la protezione dei dati e la tutela della privacy, attraverso requisiti specifici e mediante verifiche periodiche – anche attraverso il ricorso a revisori interni o esterni.

Tabella 5 - Principi generali della privacy

A seguire vengono illustrate quelle che sono considerate come migliori pratiche in cui è il titolare del trattamento che definisce lo scopo dei dati e l'incaricato del trattamento che lo attua, coerentemente con gli obblighi definiti nel GDPR.

LIMITAZIONE NELLA RACCOLTA

- Limitazione nella raccolta dei dati personali
 - Prima della raccolta di dati personali - ad esempio, quando si stipula un contratto con l'interessato - il titolare del trattamento deve ottenere il consenso preventivo e inequivocabile da parte dell'interessato o informare l'interessato della raccolta di suoi dati personali e delle finalità di utilizzo indicate in base alla normativa nazionale vigente.
 - Dal punto di vista del titolare del trattamento, il consenso è necessario soprattutto quando i dati personali vengono utilizzati in servizi commerciali. Tuttavia, nei casi di sicurezza e di servizi pubblici, può non essere necessario un consenso esplicito preliminare, anche se è probabile che il consenso implicito sia stato fornito nell'ambito dell'accordo contrattuale tra l'utente e il fornitore di servizi.
- Metodi di raccolta dei dati personali
 - Il titolare del trattamento non deve acquisire dati personali con mezzi fraudolenti o altri mezzi illeciti.
- Raccolta dati senza consenso
 - Le limitazioni alla raccolta dei dati non si applicano nei casi in cui il trattamento dei dati personali è disciplinato dalla normativa nazionale vigente. I titolari del trattamento dei dati dovrebbero raccogliere i dati senza il consenso, ad esempio se autorizzati da un'ordinanza giudiziaria nazionale o da uno strumento giuridico equipollente.
- Esclusione di quelle informazioni in grado di identificare un individuo dai dati raccolti

- Il responsabile del trattamento dei dati dovrebbe adottare le opportune misure per evitare di raccogliere dati dai quali una persona potrebbe essere identificata facendo riferimento ad una banca dati.
- Conferma del consenso da parte dell'interessato per la raccolta dei propri dati personali
 - Il titolare del trattamento dei dati deve adottare misure adeguate per ottenere la conferma sul consenso da parte dell'interessato alla raccolta dei propri dati.

QUALITA' DEI DATI

- Qualità dei dati raccolti
 - Il titolare del trattamento dei dati personali deve adoperarsi nel mantenere i dati personali esatti e aggiornati entro i limiti necessari per il raggiungimento degli scopi dell'utilizzo.

SPECIFICA DELLO SCOPO

- Specifica dello scopo d'uso
 - Nel trattare i dati personali, il titolare del trattamento deve specificare le finalità dell'utilizzo dei dati personali.
- Limitazioni al cambiamento dello scopo d'uso
 - Il titolare del trattamento dei dati non deve modificare le finalità d'uso al di fuori dell'ambito in cui le nuove finalità possono ragionevolmente essere considerate compatibili con quelle d'origine.
- La modifica delle finalità d'uso richiede il consenso preventivo
 - Prima che il titolare del trattamento dei dati modifichi le finalità d'uso che vanno oltre il campo di applicazione in cui le nuove finalità possono ragionevolmente essere considerate compatibili con le finalità di origine, deve informare l'interessato di tale modifica o ottenere un consenso preventivo e inequivocabile.

LIMITAZIONE NELL'USO DEI DATI

- Limitazione d'uso
 - Un responsabile del trattamento dei dati personali non deve trattare i dati personali, senza ottenere il consenso preventivo da parte dell'interessato, oltre quanto necessario per il raggiungimento delle finalità d'uso specificate.
- Restrizione della divulgazione a terze parti
 - Il titolare del trattamento non deve fornire dati personali a terzi senza ottenere il consenso preventivo da parte dell'interessato, tranne in casi molto limitati e ben definiti (ad esempio a seguito di richieste legali).
- Utilizzo senza consenso
 - Le disposizioni delle due specifiche precedenti non si applicano nei casi in cui il trattamento dei dati personali si basa su leggi nazionali vigenti. I titolari del trattamento dei dati dovrebbero concedere l'accesso ai dati solo alle autorità incaricate all'applicazione della legge, come autorizzato da un'ordinanza di un tribunale nazionale o da uno strumento giuridico equivalente.

MISURE DI SICUREZZA

- I dati personali devono essere protetti da adeguate misure di sicurezza contro rischi quali la perdita o l'accesso non autorizzato, la distruzione, l'uso, la modifica o la divulgazione dei dati.

APERTURA

- Dovrebbe esistere una politica generale di apertura nei riguardi di sviluppi, pratiche e politiche in materia di dati personali. Dovrebbero essere prontamente disponibili mezzi per stabilire l'esistenza e la natura dei dati personali e le principali finalità del loro utilizzo, nonché l'identità e la residenza abituale della persona che raccoglie i dati.

PARTECIPAZIONE INDIVIDUALE

- Un individuo può avere il diritto, tra gli altri, di:
 - a) ottenere dal titolare del trattamento la conferma dell'esistenza o meno di dati che lo riguardano;
 - b) di avergli comunicato i dati che lo riguardano:
 - i) entro un termine ragionevole;
 - ii) ad un onere, se del caso, non eccessivo;
 - iii) in modo ragionevole;
 - iv) in una forma per lui facilmente comprensibile;
 - c) essere motivati nel caso in cui una richiesta presentata ai sensi dei punti a) e b) viene respinta e di essere in grado di contestare tale rifiuto;
 - d) contestare i dati che lo riguardano e, se la contestazione è accolta, far cancellare, rettificare, completare o modificare i dati che lo riguardano.

RESPONSABILIZZAZIONE

- Il titolare del trattamento dei dati deve essere responsabile del rispetto delle misure che attuano i principi di cui sopra e di garantire che i responsabili del trattamento dei dati allo stesso modo si conformino.

EQUIVALENZA DI REGIME

- Il titolare del trattamento dei dati non dovrebbe trasferire dati personali al di fuori delle proprie frontiere, a meno che la destinazione non abbia un regime di privacy equivalente a quello di origine.

10.1.2 Obiettivi di protezione

Gli obiettivi di protezione mirano a fornire delle proprietà astratte, ossia indipendenti dal contesto per i sistemi IT. Nella sicurezza ICT la triade della riservatezza, dell'integrità e della disponibilità è stata ampiamente accettata. Sebbene siano state proposte diverse estensioni e perfezionamenti, questi obiettivi di protezione *core* sono rimasti stabili per decenni e sono serviti da base per molte metodologie di sicurezza ICT, (cfr. DR-3). A completamento di questi obiettivi di protezione della sicurezza, sono stati proposti tre obiettivi di protezione specifici per la privacy che approfonditi nella tabella che segue:

Obiettivo	Descrizione
Incollegabilità	Garantisce che i dati rilevanti per la privacy non possano essere collegati tra domini con scopo e contesto comuni. Ciò significa che i processi devono essere gestiti in modo tale che i dati rilevanti per la privacy non siano collegabili a qualsiasi altro insieme di dati rilevanti sulla privacy al di fuori del dominio.
La trasparenza	Garantisce che tutte le elaborazioni dei dati rilevanti per la privacy, comprese le impostazioni legali, tecniche e organizzative, possano essere comprese e ricostruite in qualsiasi momento. Le informazioni devono essere disponibili prima, durante e

	dopo l'elaborazione. Pertanto, la trasparenza deve riguardare non solo l'elaborazione effettiva, ma anche l'elaborazione pianificata (trasparenza ex ante) e il tempo trascorso dall'elaborazione per sapere cosa è successo esattamente (trasparenza ex post)
L'intervenibilità	Garantisce l'intervento in relazione a tutti i trattamenti di dati relativi alla privacy in corso o pianificati, in particolare da parte di coloro i cui dati vengono elaborati. L'obiettivo dell'intervenibilità è l'applicazione di misure correttive e controbilanci ove necessario. L'intervenibilità è legata ai principi relativi ai diritti degli individui, ad es. i diritti di rettifica e cancellazione dei dati, il diritto di revocare il consenso o il diritto di presentare un reclamo o di sollevare una controversia per ottenere il rimedio.

10.1.3 Privacy by design

10.1.3.1 Definizione della Privacy by design

La Privacy by Design (PbD) è definita come “un approccio olistico concettuale che può essere applicato - end-to-end - all'interno di un'organizzazione, includendo le sue tecnologie informatiche, le sue pratiche commerciali, i suoi processi, la progettazione fisica e le infrastrutture di rete” (cfr. DR-8). Secondo questa impostazione, l'utente dovrebbe essere considerato il centro di un sistema di protezione dei dati personali (per definizione, quindi il sistema è "user centric"). Qualsiasi progetto - sia strutturale, sia concettuale - andrebbe realizzato considerando, sin dalla fase di progettazione, la riservatezza e la protezione dei dati personali. La PbD comprende la seguente trilogia di applicazioni:

- Sistemi IT;
- Pratiche di business;
- Progettazione delle reti.

E' in questo contesto che si inserisce la necessità di prevedere l'ingegnerizzazione della privacy by design in ogni fase del ciclo di vita del software.

10.1.3.2 I sette principi della privacy by design

Principio	Descrizione
Proattivo non reattivo; Preventivo non correttivo	L'approccio di <i>Privacy by Design</i> (PbD) è caratterizzato da misure proattive piuttosto che reattive. Essa è diretta ad anticipare e prevenire gli eventi invasivi della privacy prima che accadano. PbD non attende che i rischi per la privacy si materializzino, né offre rimedi per la risoluzione delle infrazioni della privacy una volta che si sono verificati, in quanto è diretta ad impedire che si verifichino.
Privacy come impostazione predefinita	La <i>Privacy by Design</i> è diretta a garantire il massimo grado di privacy prevedendo che i dati personali siano automaticamente protetti in qualsiasi sistema IT o di business. Nessuna azione è richiesta da parte dei singoli per proteggere la loro privacy, in quanto è integrata nei sistemi per impostazione predefinita.
Privacy incorporata nel design	La <i>Privacy by Design</i> è incorporato nel design e nell'architettura dei sistemi IT e di business. Non è attuata successivamente ad un evento. Il risultato è che la privacy diventa una componente essenziale delle

	funzionalità principali. La privacy è parte integrante del sistema, senza diminuirne la funzionalità.
Funzionalità completa; somma positiva, non somma zero	La <i>Privacy by Design</i> cerca di tutelare tutti i legittimi interessi e gli obiettivi in un'ottica <i>win-win</i> , senza prevedere delle soluzioni a somma zero che includano degli inutili trade-off. <i>Privacy by Design</i> evita la pretesa di false dicotomie, come la sicurezza a discapito della privacy, in quanto dimostra che è possibile averle entrambe.
Sicurezza end-to-end - Protezione completa del ciclo di vita	La <i>Privacy by Design</i> che è stata incorporata in un sistema sin dal primo momento, si estende in modo sicuro durante l'intero ciclo di vita dei dati coinvolti: prevedendo robuste misure di sicurezza - essenziali per la privacy - dall'inizio alla fine di un ciclo di vita. Ciò garantisce che tutti i dati vengano conservati e distrutti – in modo sicuro e tempestivamente - alla fine del processo. Pertanto, la <i>Privacy by Design</i> garantisce una gestione delle informazioni sicura end-to-end.
Visibilità e trasparenza - Keep it Open	La <i>Privacy by Design</i> cerca di assicurare a tutti gli stakeholder che qualunque sia la pratica aziendale o la tecnologia coinvolta, essa opererà secondo le promesse e gli obiettivi dichiarati, anche assoggettandosi a verifiche indipendenti. Le sue componenti e le sue operazioni rimangono visibili e trasparenti, sia per gli utenti che per i fornitori.
Rispetto per la privacy degli utenti - Mantenerlo incentrato sull'utente	La <i>Privacy by Design</i> richiede ai progettisti e agli operatori di garantire gli interessi dei singoli, offrendo robuste misure di privacy per impostazione predefinita. Prevedendo degli avvisi appropriati e potenziando le opzioni user-friendly, pertanto garantendo l'impostazione user-centric.

Tabella 6 - I sette principi della Privacy by Design

Vedere anche il paragrafo 5.8.1.2 dell'*Allegato 4 - Linee Guida per la modellazione delle minacce e individuazione delle azioni di mitigazione conformi ai principi del Secure/Privacy by Design*.

10.1.4 Data protection Impact Assessment

La progettazione di qualsiasi software che coinvolga il trattamento dei dati personali deve essere preceduta da un'identificazione dei requisiti di protezione per la privacy, in quanto dal trattamento o dall'elaborazione dei dati personali potrebbero derivare dei rischi. I rischi per la privacy negli applicativi software che comportano il trattamento dei dati personali, dovrebbero essere trattati prima della loro implementazione, ossia sin dalla fase di progettazione (*Engineering Privacy by Design*). Dovranno, quindi, essere analizzati i rischi collegati alle applicazioni software.

In linea con i requisiti di attuazione previsti dal Regolamento (UE) 679 del 2016 (cfr. DR-1), di seguito indicato come **GDPR**, qualora un trattamento dei dati personali possa presentare un rischio elevato per i diritti e le libertà delle persone fisiche, i titolari di quest'ultimo dovranno effettuare una valutazione dell'impatto del trattamento sulla protezione dei dati personali o *Data Protection Impact Assessment*, di seguito indicata come "DPIA" (cfr. Art. 35 DR-1), quest'obbligo è applicabile anche al ciclo di vita del software.

Sulla base di quanto stabilito dal WP Art. 29 (cfr. DR-7), sarà necessario effettuare una valutazione della necessità di svolgere una DPIA, basandosi sulla mappa concettuale definita nella Figura 27.

In particolare, al fine di valutare se il trattamento - posto in essere all'interno di un'applicazione software - possa comportare un rischio elevato per i diritti e le libertà delle persone fisiche (Cfr. ART. 35 DR-1) sarà necessario determinare se rientra tra quelli indicati nella Tabella 7- in cui sono descritte alcune tipologie di trattamento che obbligano il titolare a svolgere una Data Protection Impact Assessment DPIA.

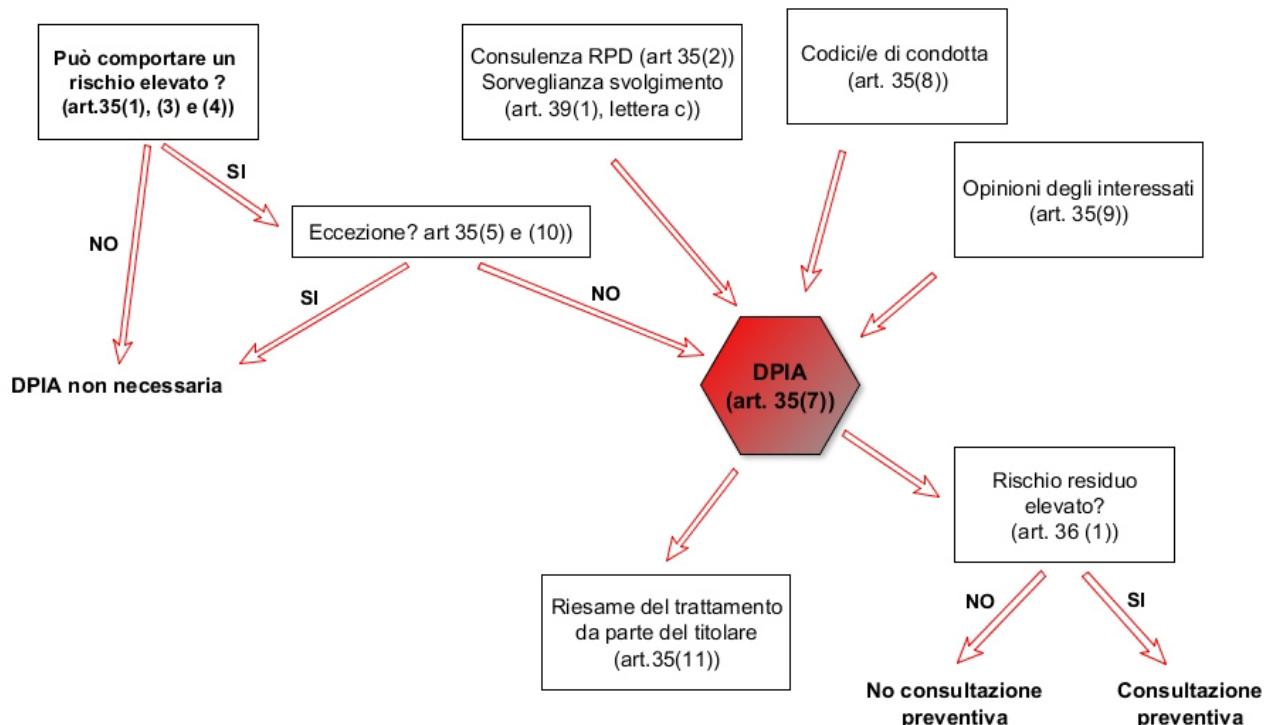


Figura 27 – Esempio di flusso di valutazione necessità DPIA

Alla luce di quanto sopra, se il trattamento, le sue modalità di attuazione o i dati trattati rientrano in quelli descritti nella Tabella 7, e non si configurano eccezioni – individuate all'interno di elenchi che dovranno essere redatti dagli Stati Membri (ad oggi non risultano essere stati ancora individuati) - sarà necessario svolgere una DPIA.

Tipologia di trattamento	Descrizione
1 - Valutazione di profilazione o scoring	Tutti quei trattamenti che analizzano i dati presenti all'interno dei propri archivi allo scopo di trarne informazioni riguardo il rendimento professionale, la situazione economica, la salute, le preferenze o gli interessi personali, l'affidabilità o il comportamento, l'ubicazione o gli spostamenti dell'interessato
2 - Decisioni automatizzate	Tutti quei trattamenti che producono effetti giuridici sulla persona fisica ovvero che incidono in modo analogo significativamente su dette persone fisiche
3 - Monitoraggio sistematico	Tutti quei trattamenti che sono utilizzati per osservare, monitorare o controllare gli interessati, compresa la raccolta di dati attraverso reti o la sorveglianza di

	un'area accessibile al pubblico
4 - Dati sensibili o estremamente personali	Tutti quei trattamenti che si riferiscono a particolari categorie di dati sensibili o estremamente personali
5 - Dati trattati su larga scala	Tutti i trattamenti che gestiscono dati personali su larga scala, in relazione al numero di soggetti interessati, al volume dei dati, alla durata o all'ambito geografico
6 - Combinazioni o raffronto di insieme di dati	Tutti quei trattamenti nei quali è prevista una presenza congiunta di due o più titolari distinti, secondo modalità che esulano dalle ragionevoli aspettative dell'interessato
7 - Dati relativi a interessati vulnerabili	Tutti quei trattamenti in cui la tipologia delle informazioni trattate determina uno squilibrio fra interessato e titolare, nel senso della mancanza del potere, in capo al primo, di acconsentire o di opporsi al trattamento. Si inseriscono in questa categoria i dati dei minori, dei dipendenti o delle persone richiedenti specifiche tutele
8 - Utilizzi innovativi	Tutti quei trattamenti che utilizzano tecnologie o tecniche innovative per la raccolta o l'utilizzo dei dati personali, dato che il livello di conoscenza tecnologica, in un dato momento storico, non è in grado valutare il livello di rischio connesso all'innovazione
9 - Trattamenti che impediscono di esercitare un diritto o avvalersi di un servizio o contratto	Tutti quei trattamenti che impediscono agli interessati di esercitare un diritto di avvalersi di un servizio o di un contratto, ossia tutti i trattamenti dai quali l'interessato non può esimersi qualora volesse accedere a detto servizio o concludere detto contratto

Tabella 7 - Tipologie di trattamento che rappresentano un rischio elevato

Nel caso in cui la DPIA sia stata valutata come necessaria (cfr. DR-7), si potrà procedere con l'analisi degli impatti potenziali sui diritti e le libertà dell'interessato (persone fisiche), a fronte del trattamento dei relativi dati personali, allo scopo di porre in essere le opportune attività di trattamento dei rischi per la protezione dei dati personali.

In linea con quanto previsto da regolamenti e standard applicabili in materia (cfr. **Errore. L'origine riferimento non è stata trovata.**), tale attività costituisce un processo composto da un insieme di attività ben definite, da compiersi in sequenza ordinata, nell'ambito delle seguenti fasi:

- 1) **Definizione del contesto**, tramite la comprensione dell'organizzazione, dell'architettura tecnologica e dei fattori che potrebbero influenzare la gestione del rischio privacy;
- 2) **Privacy risk assessment**, attraverso cui si identificano, si analizzano e si valutano i rischi per gli interessati;
- 3) **Privacy risk treatment**, in cui si identificano le strategie e le modalità operative per l'implementazione delle misure di sicurezza adeguate alla copertura dei rischi rilevati in sede di risk assessment. (I requisiti di protezione per la privacy, da implementare all'interno del piano di trattamento dei rischi individuati per il software possono essere ricavati dai controlli descritti nella ISO/IEC 29151 (cfr. DR-5)

10.1.4.1 Riconoscere le informazioni personali

Per poter definire adeguatamente il trattamento del rischio privacy per i software, sarà necessario individuare le tipologie di informazioni personali, ossia quelle da cui possono essere ricavati dei dati personali, che potrebbero essere trattate da un applicativo software. Per determinare se una persona fisica debba o meno essere considerata identificabile, sarà necessario prendere in considerazione diversi fattori. In particolare, si dovrebbe tenere conto dei mezzi che possono ragionevolmente essere utilizzati dai software per il trattamento dei dati personali. I software dovrebbero supportare meccanismi adeguati ad informare l'interessato, raccogliere il consenso e proteggere i suoi dati personali. Le seguenti specificazioni forniscono degli ulteriori chiarimenti su come determinare se un'informazione possa essere considerata personale.

Identificativi

In alcuni casi, l'identificabilità dell'interessato potrebbe essere molto semplice (e.g. quando l'informazione contiene o è associata ad un identificatore che è usato per riferirsi o per comunicare con l'interessato). Le informazioni possono essere considerate personali almeno nei seguenti casi:

- se contiene o è associato a un identificatore che fa riferimento a una persona fisica (ad esempio, il codice fiscale);
- se contiene o è associato a un identificatore che può essere correlato a una persona fisica (ad esempio, numero del passaporto, numero di conto);
- se contiene o è associato a un identificatore che può essere utilizzato per stabilire una comunicazione con una persona fisica identificata (ad esempio, una posizione geografica precisa, un numero di telefono);
- se contiene un riferimento che collega i dati a uno degli identificatori di cui sopra.

Altre caratteristiche identificative

Le informazioni non devono necessariamente essere associate a un identificatore per poter essere considerate personali. Le informazioni saranno considerate personali anche se contengono o sono associate a una caratteristica che distingue una persona fisica da altre persone fisiche, ad esempio i dati biometrici. Qualsiasi attributo che assume un valore che identifica univocamente un l'interessato deve essere considerato come una caratteristica identificativa. Si noti che indipendentemente dal fatto che una determinata caratteristica distingue una persona fisica da altre potrebbe cambiare a seconda del contesto di utilizzo. Ad esempio, mentre il cognome di una persona fisica potrebbe essere insufficiente per identificare quella persona fisica su scala globale, potrebbe invece esserlo su una scala aziendale. Inoltre, potrebbero anche esservi situazioni in cui una persona fisica è identificabile anche se non esiste un singolo attributo che la identifica in modo univoco. Questo è il caso in cui una combinazione di diversi attributi messi insieme consente di distinguere tale persona dalle altre, ad esempio la combinazione degli attributi "femmina", "45" e "avvocato" può essere sufficiente per identificare una persona fisica all'interno di una determinata organizzazione, ma con buona probabilità sarà insufficiente per identificare quella persona fisica al di fuori di tale contesto.

La tabella che segue fornisce alcuni esempi di attributi che potrebbero essere personali, a seconda del dominio.

Età o bisogni speciali delle persone fisiche vulnerabili Accuse di condotta criminale Qualsiasi informazione raccolta durante i servizi sanitari Conto bancario o numero di carta di credito	Posizione derivata dai sistemi di telecomunicazione Storia medica Nome Identificativi nazionali (ad es. Numero di
---	--

Identificatore biometrico	passaporto)
Estratto conto della carta di credito	Indirizzo e-mail personale
Condanne penali o reati commessi	Numeri di identificazione personale (PIN) o password
Rapporti di indagini penali	Interessi personali derivati dall'utilizzo di tracciamento di siti Web
Numero cliente	Profilo personale o comportamentale
Data di nascita	Numero di telefono personale
Informazioni sanitarie diagnostiche	Fotografia o video identificabili con una persona fisica
Disabilità	Preferenze di prodotto e servizio
Fatture del medico	Origine razziale o etnica
Stipendi dei dipendenti e file di risorse umane	Credenze religiose o filosofiche
Profilo finanziario	Orientamento sessuale
Genere	Appartenenza sindacale
Posizione GPS	Bollette
Traiettorie GPS	
Indirizzo di casa	
Indirizzo IP	

Tabella 8 - Esempi di attributi per identificare una persona

Dati pseudonimizzati

Al fine di limitare la capacità del titolare o del responsabile di identificare l'interessato, l'identità di quest'ultimo e le informazioni che lo riguardano possono essere sostituite da pseudonimi. Tale sostituzione viene solitamente eseguita da un soggetto terzo prima di trasmettere le informazioni a un destinatario. La sostituzione viene considerata pseudonimizzazione quando:

- (a) gli attributi collegati allo pseudonimo non sono sufficienti per identificare l'interessato;
- (b) l'assegnazione degli pseudonimi è tale da non poter essere invertita da parte delle persone che l'hanno eseguita.

La pseudonimizzazione evita il collegamento. Ma essendo diversi i dati collegabili allo stesso pseudonimo, esiste il rischio che la pseudonimizzazione sia violata, in quanto più grande è il set di dati associato a un dato pseudonimo, maggiore è il rischio che la proprietà (a) venga violata. Inoltre, più piccolo è il gruppo di persone fisiche a cui un insieme di dati pseudonimi si riferisce, maggiore sarà la probabilità che un interessato sia identificabile. Gli attributi contenuti direttamente nelle informazioni in questione e quelli che possono essere facilmente collegati a queste informazioni (ad es. utilizzando un motore di ricerca o dei riferimenti incrociati con altri database) devono essere presi in considerazione nel determinare se l'informazione si riferisce a un elemento identificabile dell'interessato.

La pseudonimizzazione è differente dall'anonymizzazione: i processi di anonymizzazione soddisfano entrambe le proprietà (a) e (b) di cui sopra, ma eliminano il collegamento. Durante l'anonymizzazione, le informazioni sull'identità vengono cancellate o sostituite da pseudonimi per i quali la funzione di associazione viene distrutta. Quindi, i dati resi anonimi non sono più personali.

Metadati

I dati personali possono essere memorizzati in un sistema ICT in modo tale da non essere facilmente visibili all'utente del sistema. Ad esempio, la memorizzazione del nome dell'interessato come metadato nelle proprietà di un documento, nei commenti o nelle modifiche. L'interessato deve essere a conoscenza dell'esistenza di tali dati sotto forma di metadati o del loro trattamento per tale scopo, in quanto potrebbe preferire che le informazioni personali non vengano elaborate in questo modo o condivise pubblicamente.

Dati non richiesti

Anche le informazioni personali non richieste da un titolare, cioè non intenzionalmente ottenute, potrebbero essere memorizzate da un software. Ad esempio, l'interessato potrebbe fornire delle informazioni personali anche quando non è stato richiesto dal trattamento (ad es. ulteriori informazioni personali fornite nel contesto di un modulo di feedback anonimo su un sito Web). Il rischio di raccogliere informazioni personali indesiderate può essere ridotto considerando le misure di tutela della privacy al momento della progettazione del software.

I dati personali stabiliti dal GDPR (cfr. DR-1) sono suddivisi nelle seguenti categorie di dati personali:

Categorie di dati personali	Descrizione
Dati identificativi	I dati identificativi rappresentano tutti quei dati che possono identificare, direttamente o indirettamente una persona, con particolare riferimento a un identificativo come il nome, un numero di identificazione, dati relativi all'ubicazione, un identificativo online.
Dati Particolari/Sensibili	I dati particolari sono tutti quei dati personali che rivelino l'origine razziale o etnica, le opinioni politiche, le convinzioni religiose o filosofiche, o l'appartenenza sindacale, nonché dati genetici dati biometrici intesi a identificare in modo univoco una persona fisica, dati relativi alla salute o alla vita sessuale o all'orientamento sessuale della persona.
Dati giudiziari	I dati giudiziari rappresentano tutti quei dati relativi alle condanne penali e ai reati o a connesse misure di sicurezza.

10.1.5 Flusso informativo del trattamento

Per definire l'architettura e il design di un software i progettisti dovranno prendere in considerazione la struttura del flusso informativo, descrivendo le interazioni tra interessato, titolare, responsabile e terze parti all'interno dell'applicativo software. Gli attori identificati possono interagire tra loro in vari modi, secondo i seguenti scenari, maturati dalla ISO/IEC 29134:2017⁴⁵:

⁴⁵ <https://www.iso.org/standard/62289.html>

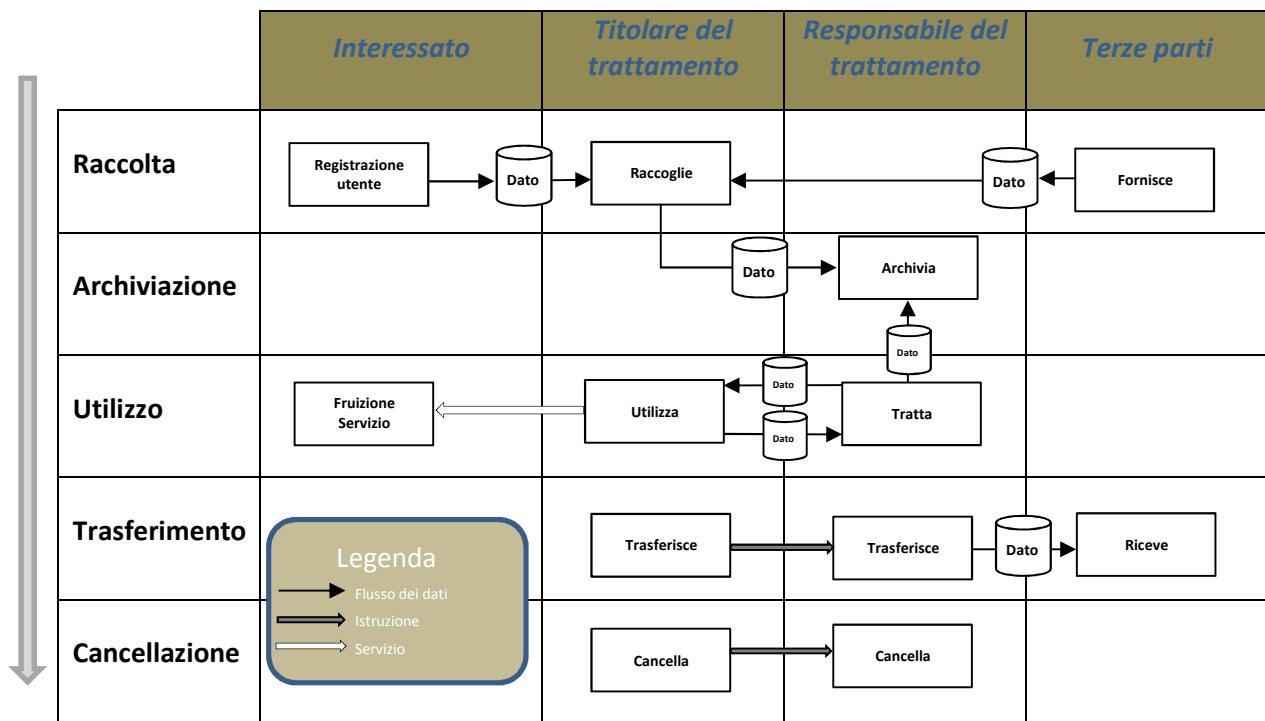


Figura 28 - Esempio di flusso informativo del trattamento

10.1.6 Privacy Implementation Strategy

La Privacy Implementation Strategy prevede che i progettisti del software definiscano e selezionino un modello di ciclo di vita adeguato all'ambiente di produzione e di sviluppo, all'ambito, all'ampiezza e alla complessità del progetto, parametrato sulle necessità emerse dai risultati della data protection impact assessment per la privacy (10.1.4).

Dovranno essere documentate:

- I principi generali della privacy applicabili alla progettazione del software (10.1.1)
- Gli obiettivi di protezione che il software dovrebbe garantire (10.1.2)
- I principi della privacy by design applicabili alla progettazione del software (10.1.3.2)
- I risultati della data protection impact assessment per il software e l'individuazione dei requisiti di protezione per la privacy (10.1.4)
- Le tipologie di Informazioni Personalari Identificabili (PII) trattate nell'ambiente software (10.1.4.1)
- La descrizione del flusso informativo derivante dal trattamento all'interno del software (10.1.5)

10.2 Ciclo di vita dello sviluppo software nell'ambito del GDPR

Molti articoli che trattano la tecnologia dell'informazione sulla base del regolamento generale sulla protezione dei dati dell'UE si focalizzano su specifici obblighi commerciali e legali in materia di dati personali. Tali articoli si concentrano spesso sul trattamento fisico dei dati e sugli obblighi del responsabile di quest'ultimo nella gestione dello stesso. Questa è una considerazione importante per le organizzazioni che operano nell'UE.

Tuttavia, oltre alla localizzazione dei dati, il GDPR ha un impatto profondo e significativo sul ciclo di vita dello sviluppo del software e sui relativi processi di sviluppo informatico per quelle organizzazioni che prevedono la realizzazione di progetti relativi a sistemi informativi all'interno dell'UE.

Il reparto IT di un'organizzazione può utilizzare uno dei molteplici e distinti tipi di SDLC (System Development LifeCycle) presenti sul mercato, come Agile, DevOPS, Waterfall, Iterative e così via. Nonostante i nomi e gli approcci differenti, queste tipologie di SDLC hanno diverse aree in comune: tutti gli SDLC hanno una qualche forma di pianificazione, progettazione, realizzazione, test, rollout e mantenimento che coprono l'intero ciclo di vita di un sistema informativo.

Gli SDLC vengono utilizzati per costruire sistemi informatici gestendo e controllando con successo il progetto IT, sfruttando il fatto che la maggior parte dei sistemi informatici hanno livelli o moduli comuni.

In generale, nella maggior parte delle tecnologie impiegate, troviamo in comune i seguenti moduli:

- Livelli di trasporto dati e sicurezza;
- I livelli di database e architettura dei dati;
- I livelli applicativi e logici;
- I livelli di presentazione e portale.

L'SDLC, qualunque sia il tipo utilizzato, gestisce e controlla il progetto informatico, dalla pianificazione all'implementazione, attraverso i suddetti livelli o moduli.

Nell'ambito del GDPR vi è un numero significativo di requisiti e cambiamenti a livello di attività, processo, politica e procedure.

Il GDPR ha un impatto incredibile sul processo SDLC per quelle imprese che installano sistemi nell'UE e aumenta notevolmente la complessità dei progetti funzionali e tecnici associati ai vari livelli tecnici sopra descritti (ad esempio il livello di database).

I requisiti funzionali e tecnici introdotti dal GDPR per i sistemi informatici, sono sostanziali e non irrilevanti. In effetti, influenzano quasi tutti gli aspetti della progettazione e della realizzazione dei sistemi attraverso ciascuno dei suddetti livelli tecnologici. Tali influenze da parte del GDPR devono essere affrontate nella fase di pianificazione dell'SDLC, ovvero all'inizio, per evitare sovraccosti significativi e rielaborazioni successive nel processo informatico.

Segue un inventario di sedici aree di pertinenza ad articoli del GDPR che influenzano la pianificazione funzionale e tecnica dell'SDLC e i requisiti per i reparti IT. Tale elenco può essere considerato come un insieme di consigli generali per i CIO e i responsabili IT che redigono i requisiti dei loro sistemi operanti nell'ambito dell'UE:

1. L' implementazione della protezione dei dati nel sistema e nell'organizzazione, per progettazione e per impostazione predefinita, è un requisito legale:
 - a. considerando 78 e Articolo 25
2. I dati devono essere protetti, e l'integrità e la riservatezza devono essere mantenute, utilizzando mezzi tecnici e organizzativi sotto la direzione del controllore:
 - a. considerando 49 e Articoli 5-1(f), 32-1(b-d)
3. Ove possibile, deve essere utilizzata la cifratura dei dati:
 - a. considerando 83 e Articoli 6-4(e), 32-1(a)
4. Ove possibile, deve essere utilizzata una pseudonimizzazione dei dati:
 - a. considerando 26, 28, 29, 78 e Articoli 6-4(e), 25-1, 32-1(a)
5. Ove possibile, i dati devono essere resi anonimi:
 - a. considerando 26
6. Al momento della raccolta dei dati, gli attributi del trattamento e le fasi elaborative devono essere forniti all'interessato, per via elettronica o per iscritto, in forma chiara e facilmente comprensibile:
 - a. considerando 39, 58 e Articoli 12-1, 13-2(a-f)

7. Le persone interessate hanno il diritto di accedere ai loro dati e di controllarne il trattamento in qualsiasi momento:
 - a. considerando 58, 61, 63 e Articoli 12, 15-1(a, d)
8. Separare le informazioni che potrebbero essere considerate dati personali o profili personali se trattati o combinati separatamente o insieme, al risultato di attività illecite:
 - a. considerando 30
9. I dati relativi a un soggetto interessato dovranno essere portabili verso un altro provider (anche se concorrente):
 - a. considerando 68 e Articoli 13-2(b), 14-2(c), 20
10. L'interessato ha diritto a una copia dei suoi dati in un formato comunemente utilizzato
 - a. Articolo 15-3
11. L'interessato ha il diritto di ottenere gratuitamente l'aggiornamento dei propri dati in caso di errore.
 - a. considerando 59, 65 e articolo 16 e, l'interessato ha il diritto di chiedere tale aggiornamento per via elettronica, riferimento 59
12. L'interessato ha il diritto di ottenere la cancellazione immediata dei dati che lo riguardano:
 - a. considerando 59, 65 e articoli 13-2(b), 14-2(b), 17 e, l'interessato ha il diritto di chiedere tale cancellazione per via elettronica, riferimento 59 (Nota: Esistono nel GDPR particolari eccezioni a tale diritto.)
13. Il titolare del trattamento deve comunicare ad altre organizzazioni IT che detengono i dati dell'interessato che questi ha richiesto la cancellazione dei propri dati:
 - a. considerando 66 e articolo 19 (quindi, il dipartimento IT deve sapere dove vengono conservati da terze parti tutti i dati degli interessati in modo che le parti coinvolte possano essere informate della richiesta di cancellazione. Sono essenziali inventari aggiornati dei dati interni ed esterni).
14. L'interessato ha il diritto di opporsi, revocare il consenso e rinunciare al trattamento. Questo può opporsi o revocare il proprio consenso in caso di trattamento elettronico dei propri dati:
 - a. considerando 59, 63 e articoli 7-3, 18, 21 (e con raccomandazione tecnica del Consiglio UE: riferimento 67)
15. I dati vengono conservati solo per il tempo necessario a conseguire gli obiettivi dell'interessato. I dati personali scaduti non devono essere memorizzati. (Parte di una strategia di gestione dei registri elettronici). La persona interessata deve essere informata di tale periodo o delle modalità di elaborazione al momento della raccolta dei suoi dati:
 - a. considerando 39, 45 e Articoli 13-2(a), 14-2(a), 25-2
16. Si deve stabilire, quasi immediatamente, se una violazione dei dati possa essere stata un "rischio elevato per i diritti e la libertà della persona fisica" in quanto deve essere predisposto l'opportuno ambiente tecnico per individuare, tracciare e valutare tali violazioni.
 - a. considerando 85, 86 (relativi agli obblighi di notifica), 87 (Nota: Molti articoli, ad esempio 33, 34) del GDPR riguardanti gli obblighi di comunicazione alla persona interessata e alle autorità competenti in materia.

Inoltre, molti dei punti di cui sopra, ad esempio l'undicesimo, richiedono aggiornamenti del contact center e interazioni e conferme con e da parte dell'interessato.

Una cosa è certa: ciascuno dei sedici punti di cui sopra dovrà avere una posizione nella documentazione di progettazione funzionale e tecnica dei sistemi realizzati con il supporto dell'SDLC, e ciascuno di essi apporterà una certa complessità alle fasi di progettazione del sistema nel suo complesso. In più, molti di questi influenzano anche i processi globali di assistenza verso i clienti dell'azienda, poiché il GDPR non

solo richiede determinati requisiti tecnici "puri", ma anche requisiti funzionali all'attività organizzativa supportati sia dalla tecnologia che dai processi aziendali.

In sintesi, il testo del GDPR contiene requisiti funzionali e tecnici del sistema, sia esplicativi che impliciti, che influiscono e influenzano l'SDLC adottato dalle organizzazioni che progettano l'introduzione dei nuovi sistemi nell'UE.

L'impatto del GDPR sullo sviluppo del software inizia a partire dall'architettura dei dati e dai livelli di trasporto di questi, per arrivare fino ai livelli di portale e di presentazione. La chiave di base per il successo dello sviluppo IT è la pianificazione di tali requisiti durante le fasi iniziali dell'SDLC; sebbene possano aggiungere una certa complessità alle fasi iniziali di pianificazione e progettazione dell'SDLC, i costi di sviluppo complessivi saranno notevolmente ridotti al minimo se considerati il più precocemente possibile nel processo di costruzione dei sistemi IT.

10.3 Implementazione della strategia nelle fasi di sviluppo del software

10.3.1 Scopo

Gli elementi definiti all'interno della Privacy Implementation Strategy (10.1.6), i requisiti di protezione della privacy e le strategie di design per la privacy (ricavabili sulla base di quelli individuati da ENISA in DR-3), dovranno essere inquadrati all'interno di ciascuna fase della Engineering privacy by design (10.3.2) e rimappati per ciascuna fase del ciclo di vita dei software), così come definiti nelle fasi *Software life Cycle Processes* (cfr. DR-2).

10.3.2 Le fasi di implementazione della Engineering Privacy by Design

La seguente impostazione è stata maturata dal *Privacy Engineering Framework* del MITRE (cfr. DR-6), prevedendo le seguenti attività:

Attività	Descrizione
Definizione dei requisiti privacy:	Definizione delle proprietà della privacy di un software in modo che possa essere integrato con il design e lo sviluppo
Design e sviluppo privacy:	Definizione del design e sviluppo dei requisiti previsti
Verifica e validazione privacy:	Riscontro della conferma che i requisiti di privacy sono stati correttamente implementati e validati attraverso delle verifiche

Tabella 9 - Fasi dell'Engineering Privacy by Design

10.3.2.1 Definizione dei requisiti privacy

Input: Requisiti di privacy di base e test; Normative, best practice e procedure applicabili sulla privacy; requisiti funzionali; Profili di rischio per la privacy.

Attività: Svolgere una Data Protection Impact Assessment modernizzata sugli obiettivi di protezione individuati; Selezionare e perfezionare i requisiti di protezione per la privacy di base e effettuare dei test; Sviluppare dei requisiti di protezione della privacy personalizzati e testarli sulla base dei risultati della DPIA.

Output: Requisiti di protezione per la privacy specifici per il software.

10.3.2.2 Design e sviluppo privacy

Input: Requisiti Architetturali e funzionali specifici per la privacy

Attività: Identificare delle strategie e dei modelli di design della privacy; Identificare dei controlli di privacy, dei criteri tecnici e delle policy; Sviluppare dei dati e dei modelli di processo che riflettano i controlli di privacy identificati; Allineare, integrare e implementare i controlli di privacy con gli elementi funzionali; Analizzare il rischio del design di privacy complessivo (vedi anche Allegato 4 – Linee Guida per la modellazione delle minacce e individuazione delle azioni di mitigazione conformi ai principi del Secure/Privacy by Design – Paragrafo 5.8).

Output: Componenti del software implementati; Mitigazione dei rischi accettabili per la privacy residua

10.3.2.3 Verifica e validazione privacy

Input: Componenti del software implementati; Requisiti di privacy specifici del sistema e test]; Politiche e procedure di privacy applicabili.

Attività: Sviluppare / perfezionare dei test sulla privacy; Eseguire delle verifiche sulla privacy; Verificare il comportamento operativo rispetto alle politiche e alle procedure sulla privacy applicabili.

Output: Risultati dei test di privacy; Documentazione delle Incoerenze sulla privacy documentate; Descrizione del piano di trattamento della privacy.

10.4 Integrazione della Engineering Privacy by Design nel Software Life Cycle Process

Il diagramma illustrato nella Figura 29, definisce la mappatura delle fasi della Engineering Privacy by Design sulle fasi del Software Life Cycle Process:

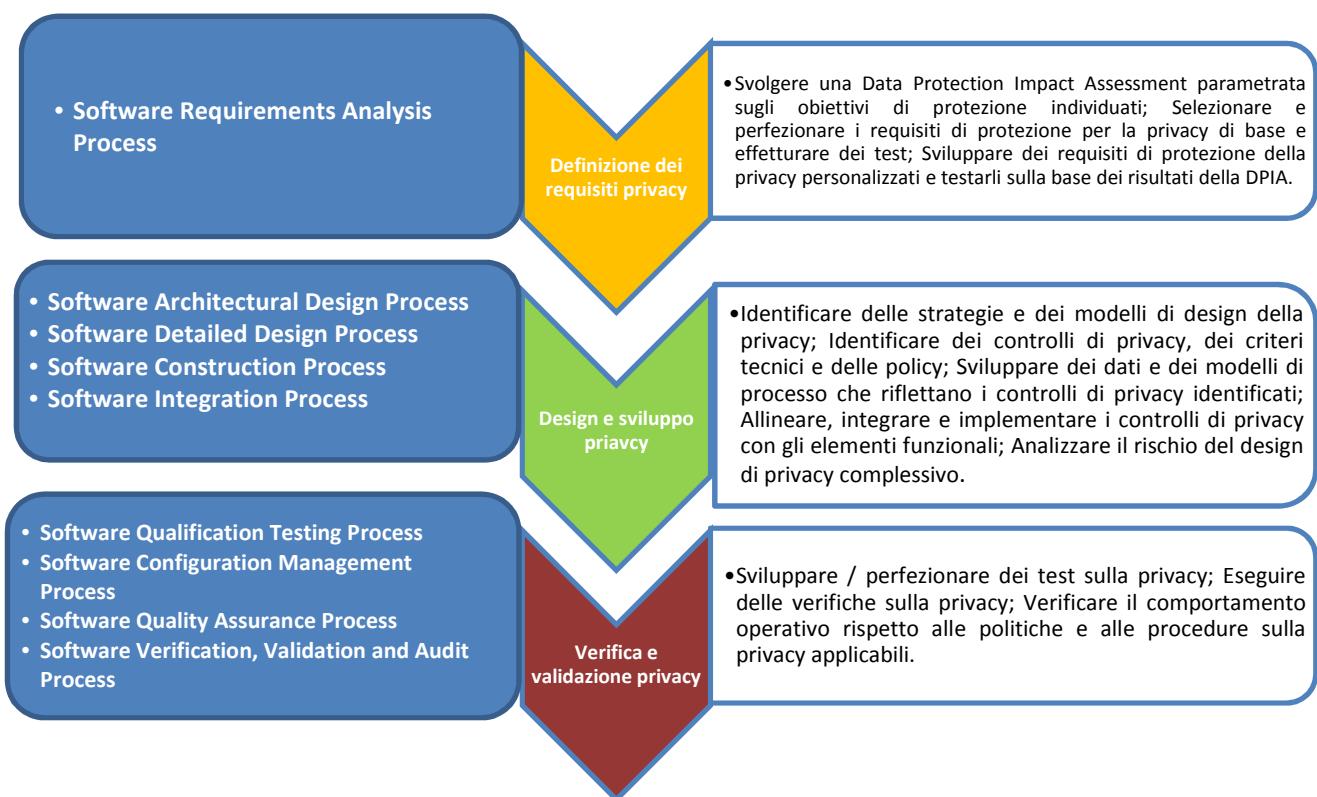


Figura 29 - Integrazione della Engineering privacy by design nel Software Life Cycle Process



APPENDICE 1. CATALOGO SECURITY TOOLS

Prodotto	Categoria	Fase SSE	Tipo Licenza	Sito Web
Acunetix Web Vulnerability Scanner	DAST, IAST	Verification	Versione trial a 14 giorni disponibile	https://www.acunetix.com/
Microsoft Cloud App Security (MCAS)	Cloud Access Security Broker	Verification	Versione trial disponibile	https://www.microsoft.com/en-us/microsoft-365/enterprise-mobility-security/cloud-app-security
Airlock Suite by Ergon Informatik	WAF, Authentication, Identity Management	Response	Versione trial disponibile	https://www.airlock.com
Akamai	CDN, DDoS Protection, WAF	Response	Prova gratuita disponibile	https://www.akamai.com/it/it/
Alert Logic SIEMless Threat Management	Intrusion Prevention System, Cloud Access Security Broker, WAF, Container Security	Response	Versione trial disponibile	https://www.alertlogic.com/
AWS WAF	WAF	Response	Nessuna trial disponibile	https://aws.amazon.com/it/waf/
Visual Trace Spec	Requirements management	Requirements	Versione trial disponibile	http://visualtracespec.com/#
Open Source Requirements Management ToolaNimble	Requirements management	Requirements	Open Source	https://github.com/osrmt/osrmt/



Potection Center	Mobile AST	Response	Non disponibile	https://appmobi.com
AppSpider Pro by Rapid7	DAST	Verification	Versione trial disponibile	https://www.rapid7.com
AppWall by Radware	WAF, DDoS Protection	Response	Nessuna versione trial disponibile	https://www.radware.com/
Arbor DDoS Protection	DDoS Protection	Response	Nessuna versione trial disponibile	https://www.netscout.com/arbor-ddos
Armor Complete	Cloud Security Platform	Release	Nessuna versione trial disponibile	https://www.armor.com
Arxan Application Protection	Mobile AST DAST	Response	Nessuna versione trial disponibile	https://www.arxan.com/application-protection
Pradeo Security	Mobile AST	Verification	Nessuna versione trial disponibile	https://www.pradeo.com/it-IT/protezione-flotta-mobile
Barracuda Web Application Firewall	WAF	Response	Versione trial disponibile su richiesta	https://www.barracuda.com/products/webapplicationfirewall
BeEF	Penetration Testing	Verification	Open Source	https://beefproject.com/
VMWare Carbon Black	Endpoint Security	Verification / Response	Demo disponibile su richiesta	https://www.carbonblack.com/
Synopsys Black Duck Hub	Library Inspection	Verification	Demo disponibile su richiesta	https://www.blackducksoftware.com/
Lookout Mobile	Mobile Access	Response	Demo	https://www.lookout.com/products/mobile-endpoint-security



Endpoint Security	Security Broker		disponibile su richiesta	
Brakeman	SAST	Implementation	Open Source	https://brakemanscanner.org/
BrightCloud Threat Intelligence by Webroot	DAST	Verification	Nessuna versione trial disponibile	https://www.brightcloud.com
Burp Suite by PortSwigger	SAST, DAST, Penetration Testing	Implementation / Verification	Versione Community liberamente scaricabile	https://portswigger.net
CaseComplete	Requirements management	Requirements	Versione trial disponibile su richiesta	https://casecomplete.com/
CppCheck	SAST	Implementation	Open Source	http://cppcheck.sourceforge.net/
CD Protection by CD Networks	CDN, WAF, DDoS Protection	Response	Nessuna versione trial disponibile	https://www.cdnetworks.com
Checkmarx CxSAST	SAST, DAST, RASP	Implementation / Verification	Versione trial disponibile a richiesta	https://www.checkmarx.com/
CipherCloud	Cloud Access Security Broker	Response	Versione trial disponibile	https://www.ciphercloud.com
CloudFlare	CDN, DDoS Protection, WAF	Response	Nessuna versione trial disponibile	www.cloudflare.com
CloudFront by Amazon	CDN, DDoS Protection	Response	Nessuna versione trial disponibile	https://aws.amazon.com/it/cloudfront/
Cloud Access Security Broker (CASB)	Cloud Access Security Broker	Response	Demo gratuita a richiesta	https://umbrella.cisco.com/products/casb



CloudPassage Halo	Cloud Access Security Broker	Response	Versione trial disponibile	https://www.cloudpassage.com
CloudSOC Cloud Access Security Broker (CASB)CloudSOC by Elastica	Cloud Security Testing/Scanning	Verification	Nessuna versione trial disponibile	https://www.symantec.com/products/cloud-application-security-cloudsoc
CodeDx	SAST, DAST	Implementation / Verification	Versione trial disponibile	https://codedx.com/
CodeProfiler by Virtual Forge	SAST per applicazioni SAP	Implementation	Nessuna versione trial disponibile	https://www.virtualforge.com
Yottaa	CDN, DDoS Protection, WAF	Verification	Demo disponibile su richiesta	https://www.yottaa.com
Contrast Enterprise	IAST, RASP	Implementation / Verification	Demo disponibile su richiesta	https://www.contrastsecurity.com
Coras	Threat Modeling tool/practices	Design	Open Source	http://coras.sourceforge.net/downloads.html
DDoS Strike by Security Compass	DDoS Protection	Response	Demo disponibile su richiesta	https://www.securitycompass.com
Endpoint Privilege Management	Endpoint Security	Verification / Response	Demo disponibile su richiesta	https://www.beyondtrust.com/
R&S®Web Application Firewall	WAF	Response	Demo disponibile su richiesta	www.denall.com
Dependency Check	Library Inspection	Implementation	Open Source	https://www.owasp.org/index.php/OWASP_Dependency_Check



F5 Big-IP	WAF, DDoS Protection	Response	Demo disponibile su richiesta	https://f5.com
Falcon	Endpoint Security	Verification / Response	Versione trial disponibile	https://www.crowdstrike.com
SpotBugs	SAST	Implementation	Open Source	https://spotbugs.github.io/
FireEye NX	Web Server Scanner, WAF	Response	Versione trial non disponibile	https://www.fireeye.com
FortiWeb: Web Application Firewall and API Protection Fortigate Firewall Platform by Fortinet	WAF	Response	Demo disponibile su richiesta	https://www.fortinet.com/products/web-application-firewall/fortiweb.html
FortiGate NGFW	WAF	Response	Demo disponibile su richiesta	https://www.fortinet.com/it/products/next-generation-firewall/models-specs.html
Gendarme	SAST	Implementation	Open Source	https://github.com/mono/website/blob/gh-pages/docs/tools+libraries/tools/gendarme/index.md
Microfocus Fortify Static Code Analyzer	SAST, DAST, IAST, RASP	Implementation / Verification	Demo disponibile su richiesta	https://www.microfocus.com/it-it/products/static-code-analysis-sast/overview
HCL Security AppScan	SAST, DAST, IAST	Implementation / Verification	Versione trial non disponibile	https://www.hcltech.com
IBM Engineering Requirements Management DOORS Next	Requirements management	Requirements	Versione trial non disponibile	https://www.ibm.com
IBM Rational RequisitePro solution	Requirements management	Requirements	Versione trial non disponibile	https://www.ibm.com
Imperva FlexProtect	WAF, DDoS	Response	Demo	https://www.imperva.com/products/flexportprotect-plans/



	Protection		disponibile su richiesta	
BloxOne Threat Defense	WAF	Response	Versione trial disponibile su richiesta	https://www.infoblox.com/products/bloxone-threat-defense/
Hillstone E-Series Next-Generation Firewalls (NGFW)	WAF	Response	Demo non disponibile	https://www.hillstonenet.com
		Verification	Demo non disponibile	https://www.hillstonenet.com
Visure Requirements Management Tool	Requirements management	Requirements	Versione trial disponibile su richiesta	https://visuresolutions.com/requirements-management-tool/
JSHint	SAST	Implementation	Open Source	https://jshint.com/
Kali Linux	Penetration Testing	Verification	Open Source	https://www.kali.org/
Klocwork	SAST	Implementation	Versione trial disponibile su richiesta	https://www.perforce.com/products/klocwork
Kona Site Defender by Akamai	WAF, DDoS Protection	Response	Demo disponibile su richiesta	https://www.akamai.com/it/it/products/security/kona-site-defender.jsp
CenturyLink DDoS and Web Application Security	CDN, DDoS Protection	Response	Demo non disponibile	https://www.centurylink.com/business/security/ddos-and-web-application.html
LogRhythm Security Intelligence Platform	Predictive Security Analytics	Verification / Response	Demo disponibile su richiesta	www.logrhythm.com
Malwarebytes Endpoint Security	Endpoint Security	Verification	Versione trial disponibile	https://www.malwarebytes.com/business/endpointsecurity/
MetaFlows	Cloud Security	Implementation	Demo	www.metaflows.com



	Scanning		disponibile su richiesta	
MetaDefender	Predictive Security Analytics	Verification / Response	Available by Request	https://metadefender.opswat.com/
Metasploit by Rapid7	Penetration Testing	Verification	Open Source	https://www.metasploit.com/
Microsoft Application Verifier	DAST	Verification	Free	https://www.microsoft.com
Microsoft Attack Surface Analyzer	Intrusion Prevention	Verification	Free	https://www.microsoft.com
Microsoft BinScope	SAST	Implementation	Free	https://www.microsoft.com
Microsoft Code Analysis Tool	SAST	Implementation	Free	https://www.microsoft.com
Microsoft FxCop	Library Inspection	Implementation	Free	https://www.microsoft.com
Microsoft SDL Regex Fuzzer	SAST	Implementation	Free	https://www.microsoft.com
Microsoft SDL MiniFuzz File Fuzzer	SAST	Implementation	Free	https://www.microsoft.com
Microsoft Security Assessment Tool (MSAT)	Risk Management	Risk Assessment	Free	https://technet.microsoft.com/it-it/security/cc185712.aspx
Microsoft Threat Modeling Tool	Threat Modeling tool	Design	Free	https://www.microsoft.com
ModSecurity	WAF	Implementation / Verification	Open Source	http://modsecurity.org/
MyAppSecurity ThreatModeler	Threat Modeling tool	Design	Demo disponibile	https://threatmodeler.com/
N-Stalker Cloud Web	SAST, DAST	Implementation /	Free Tier	https://www.nstalker.com



Scan		Verification	Available	
Citrix Web App Firewall	WAF	Verification	Demo disponibile su richiesta	https://www.citrix.com/it-it/products/citrix-web-app-firewall/
Netsparker Web Application Security Scanner	DAST	Response	Demo disponibile su richiesta	https://www.netsparker.com/
Neustar	DDoS Protection, WAF	Response	Demo disponibile su richiesta	https://www.home.neustar/
NEVIS Security Suite NEVIS Security Suite	WAF, Authentication, Identity mgmt	Verification	Available by Request	https://www.nevis-security.ch/en/
Nikto2	Web Server Scanner	Verification	Open Source	https://www.cirt.net/Nikto2
Nmap	Penetration Testing and Network Mapping	Verification / Response	Open Source	https://nmap.org/
NSFOCUS Web Application Firewall	DAST, WAF	Verification	Demo non disponibile	https://nsfocusglobal.com/web-application-firewall-waf/
Objectives	Requirements management	Requirements	Versione trial disponibile	http://www.objectiver.com
OWASP Zed Attack Proxy (ZAP)	Penetration Testing	Verification / Response	Open Source	www.owasp.org
Open Source Requirements Management Tool (OSRMT)	Requirements management	Requirements	Open Source	http://sourceforge.net/projects/osrmt/



Paloalto Next-Generation Firewall	WAF	Verification	Demo disponibile su richiesta	https://www.paloaltonetworks.com
Paloalto Next-Generation Firewall				
Palo Alto Threat Prevention Services	RASP WAF	Response	Demo disponibile su richiesta	https://www.paloaltonetworks.com
Peach Fuzzer	Penetration Testing	Verification / Response	Demo disponibile su richiesta	https://www.peach.tech/
PYLINT	SAST	Implementation	Demo disponibile su richiesta	https://polarion.plm.automation.siemens.com/products/polarion-requirements
PMD	SAST	Implementation	Demo disponibile su richiesta	https://www.imperva.com/products/runtime-application-self-protection-rasp/
Polarion REQUIREMENTS	Application Lifecycle Management (ALM)	Requirements		http://www.emerasoft.com/agile-application-lifecycle-management/polarion-alm/
Runtime Application Self-Protection	RASP	Verification / Response	Demo disponibile su richiesta	https://www.imperva.com/products/runtime-application-self-protection-rasp/
Network Threat Detection	Intrusion Prevention System	Response	Demo disponibile su richiesta	https://www.bricata.com
Network Security Monitoring and Management	CDN, App Security Scanning	Verification	Demo non disponibile	https://enterprise.verizon.com/products/security/



Qualys Security & Compliance Suite	DAST, WAF	Verification / Response	Versione trial disponibile	https://www.qualys.com
Reqtify	Requirements management	Requirements	Demo non disponibile	https://www.3ds.com/it/prodotti-e-servizi/catia/prodotti/reqtify/
Risk Fabric by Bay Dynamics	Predictive Security Analytics	Implementation / Verification / Response	Demo disponibile su richiesta	https://baydynamics.com
rmtoo	Requirements management	Requirements	Open Source	http://rmtoo.florath.net/
RSA Advanced Threat Management Solution	DAST	Implementation / Verification	Available by Request	https://www.dellemc.com
Samurai Web Testing Framework	DAST, Penetration testing	Verification	Open Source	http://www.samurai-wtf.org/
SeaMonster- Security Modeling Software	Threat Modeling tool	Design	Open Source	https://sourceforge.net/projects/seamonster/
Website Malware Scanner	SAST, DAST	Implementation / Verification	Demo non disponibile	https://www.sitelock.com
SonarLint	SAST	Implementation	Open Source	https://www.sonarlint.org
SonarQube	SAST	Implementation	Open Source	https://www.sonarqube.org
Sophos Next-Gen Firewall	WAF	Response	Versione Trial a 30 giorni disponibile	https://www.sophos.com/en-us/products/next-gen-firewall.aspx
SRX Series Firewall by Juniper Networks	WAF	Verification	Versione Trial disponibile	https://www.juniper.net/us/en/products-services/security/srx-series/
Sucuri Website	WAF	Verification	Demo non	https://sucuri.net/website-firewall/



Application Firewall			disponibile	
Sucuri Website Security Solutions	WAF, DDoS Protection, App Security Scanning	Response	Demo non disponibile	https://sucuri.net/website-security-platform/signup/
Symantec Advanced Threat Protection	IAST, RASP	Implementation / Verification	Versione Trial disponibile su richiesta	https://www.symantec.com
Tanium Endpoint Platform	Endpoint Security, App Security Scanning	Implementation / Verification	Demo non disponibile	https://www.tanium.com
Simulink Requirements	Requirements management	Requirements	Versione Trial disponibile	https://it.mathworks.com/products/simulink-requirements.html
Telelogic DOORS	Requirements Management	Requirements	Gratis	http://telelogic-doors.software.informer.com/
Thunder TPS by A10 Networks	DDoS Protection	Verification / Response	Versione Trial disponibile	https://www.a10networks.com/products/thunder-tps/
Trend Micro Deep Security Platform	SAST, DAST	Implementation / Verification	Versione Trial disponibile	https://www.trendmicro.com
TRIKE	Threat Modeling tool/practices	Design	Open Source	http://www.octotrike.org/
Tripwire Enterprise	IAST, RASP	Implementation / Verification	Demo disponibile su richiesta	https://www.tripwire.com
Trustwave Secure Web Gateway	CDN, DAST	Verification	Demo non disponibile	https://www.trustwave.com/en-us/services/technology/secure-web-gateway/



Trustwave Web Application Firewall	WAF, Penetration Testing	Verification	Demo non disponibile	https://www.trustwave.com
Veracode Cloud Platform	SAST, DAST, Mobile AST, Penetration Testing	Implementation / Verification	Demo disponibile su richiesta	www.veracode.com
vSentry by Bromium	Endpoint Security	Verification / Response	Demo disponibile su richiesta	www.bromium.com
GrayMatter Platform	Penetration Testing, App Security Scanning	Verification	Demo disponibile su richiesta	https://www.reliaquest.com/
WhiteHat Sentinel	SAST, DAST	Implementation / Verification	Demo di 30 giorni disponibile su richiesta	https://www.whitehatsec.com/info/security-check/
Wireshark	Penetration Testing and Packet-level Monitoring	Verification	Open Source	https://www.wireshark.org/
Ziften	Endpoint Security	Response	Demo disponibile su richiesta	https://ziften.com/



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, ...)	<p>[*] Requires minor adjustments</p> <p>Platform/Enviroment: Java Struts, Spring MVC, iBatis*, GWT, Hibernate, OWASP ESAPI, JSTL FMT Taglib, ATG DSP Taglib, Java Server Faces (JSF), JavaScript</p> <p>Platform/Enviroment: .NET Enterprise Libraries, Telerik, ComponentArt, Infragistics, FarPoint, iBatis*, Hibernate.Net [*], Entity framework up to 4.3.1</p> <p>Platform/Enviroment: PHP Zend, Kohana, CakePHP, Symfony, Smarty, OWASP ESAPI</p> <p>Platform/Enviroment: C/C++ MISRA</p> <p>Platform/Enviroment: Ruby Ruby on Rails</p> <p>Platform/Enviroment: JavaScript JQuery, Node.js, Ajax, Knockout, AngularJS, ExpressJS, Jade, Backbone, Handlebars, Hapi.JS</p> <p>Platform/Enviroment: iOS iOS mobile applications</p> <p>Platform/Enviroment: Python Django</p> <p>Platform/Enviroment: Groovy Grails</p>	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 deserialization, 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 prone 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 vulnerabilità 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 permette 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	
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.
b. source repository,	S'integra, tramite plugin, a Git, Svn, CVS, TFVC, Jazz RTC, ClearCase.
c. build server,	
d. bug tracking tools	SonarQube comprende la gestione completa dei bug riscontrati (tracciamento incluso).
Le tipologie di applicazione supportate (Web, Mobile, Client-Server...)	Web, Mobile Android.
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
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
L'integrazione di "Custom rules"	SonarQube offre la possibilità di creare delle regole personalizzate, attraverso dei custom templates
Possibilità di inibire la segnalazione di particolari vulnerabilità	Il tool consente di "sopprimere" la segnalazione di una particolare vulnerabilità in maniera agevole.
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.
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:		
<ul style="list-style-type: none">• 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	



11 BIBLIOGRAFIA

- [1] G. McGraw, «Software Security: Building Security In, Addison Wesley,» 2006.
- [2] S. H. Flechais, « Bringing Security Home: A Process for Developing Secure and Usable Systems,” In Proc. of the New Security Paradigms Workshop (NSPW’07),» Switzerland, 2003, pp. 49-57.
- [3] C. M. a. M. S. I. Flechais, in “*Integrating Security and Usability into the Requirements and Design Process,*” *International Journal of Electronic Security and Digital Forensics*, Inderscience Publishers, vol. 1, no. 1, , Geneva, Switzerland, 2007, pp. 12-26.
- [4] A. A. a. M. Pourzandi, in “*Secure Software Development by Example,*” *IEEE Security and Privacy* vol. 3, no. 4, IEEE CS Press, 2005, pp. 10-17.
- [5] S. O. a. O. A. A.S. Sodiya, in “*Towards Building Secure Software Systems,*” *Issues in Informing Science and Information Technology* vol. 3., California, USA, Informing Science Institute, 2006, pp. 635-646.
- [6] J. Juerjens, « Secure Systems Development with UML, Springer,,» 2005.
- [7] L. F. a. R. Solms, in “*SecSDM: A Model for Integrating Security into the Software Development Life Cycle,*” In *IFIP International Federation for Information Processing, Volume 237, Proc. of the 5th World Conference on Information Security Education,* .
- [8] T. W. J. S. a. M. B. D.P. Gilliam, « “Software Security Checklist for the Software Life Cycle,” In Proc. of the 12th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE’03), Linz, Au,» Linz, Austria, 2003, pp. 243-248.
- [9] J. P. E. H. a. M. B. D. Gilliam, « “Addressing Software Security Risk and Mitigations in the Life Cycle,” In Proc. of the 28th Annual NASA Goddard Software Engineering Workshop (SEW’03), Greenbelt,» Maryland, USA, 2003, pp. 2001-206.
- [10] «Database of Vulnerabilities, Exploits, and Signatures, <http://seclab.cs.ucdavis.edu/projects/DOVES/>,» 2009.
- [11] T. W. J. S. a. M. B. D.P. Gilliam, in “*Software Security Checklist for the Software Life Cycle,*” In *Proc. of the 12th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE’03),*, Linz, Austria.
- [12] M. Hadawi, in “*Vulnerability Prevention in Software Development Process,*” In *Proc. of the 10th International Conference on Computer & Information Technology (ICCIT’07)*, Dhaka, Banglades, 2007.
- [13] L. L. a. H. G. M. Essafi, in “*S2D-ProM: A Strategy Oriented Process Model for Secure Software Development,*” In *Proc. of the 2nd International Conference on Software Engineering Advances (ICSEA’07), Cap Esterel, French Riviera, France*, 2007, p. 24.
- [14] N. Davis, in “*Secure Software Development Life Cycle Processes: A Technology Scouting Report*”, technical note CMU/SEI-2005-TN-024, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, 2005.
- [15] T. W. J. S. a. M. B. D.P. Gilliam, « “Software Security Checklist for the Software Life Cycle,” In Proc. of the 12th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE’03), Linz, Au,» Austria, 2003, pp. 243-248.