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 Gartner[1], 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 Clusit[2] 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 incapsulamento 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 Details[5], nel 2017 il totale delle falle 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/

2001	1677	January	February	March	April	May	June	July	August	September	October	November	December
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		

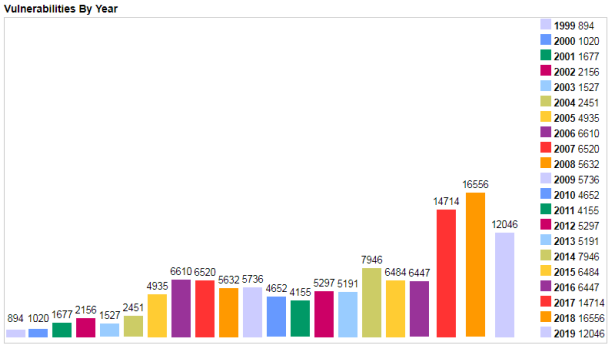

Figura 1 - Andamento delle vulnerabilità per anno [Fonte: https://www.cvedetails.com]

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 OWASP[6] 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. 25[7]).

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.

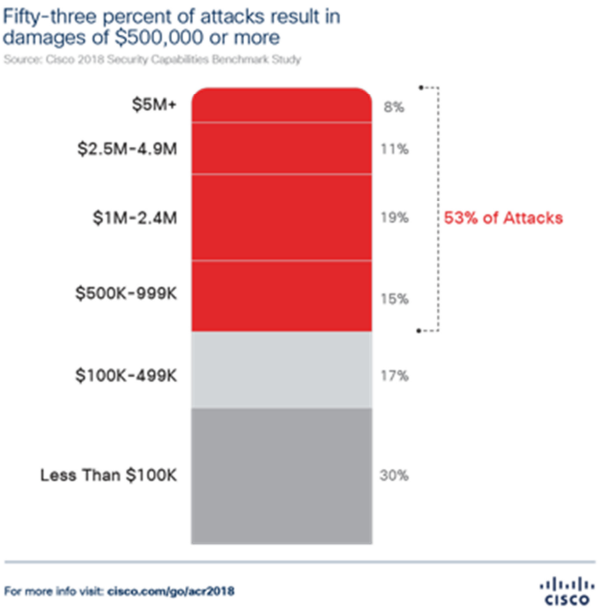

Figura 2 - Il costo degli attacchi

[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.

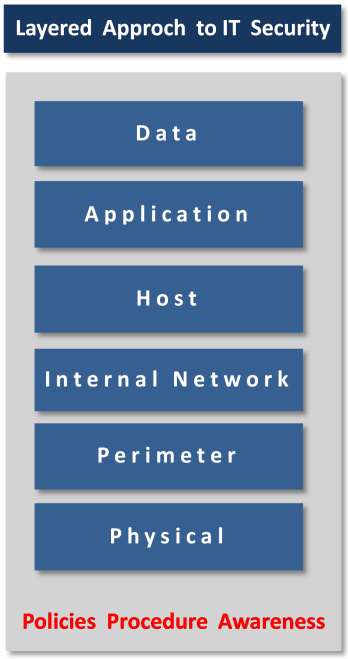

Figura 3 - Defence-in-Depth model for IT

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, Forrester[8], 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 al 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.

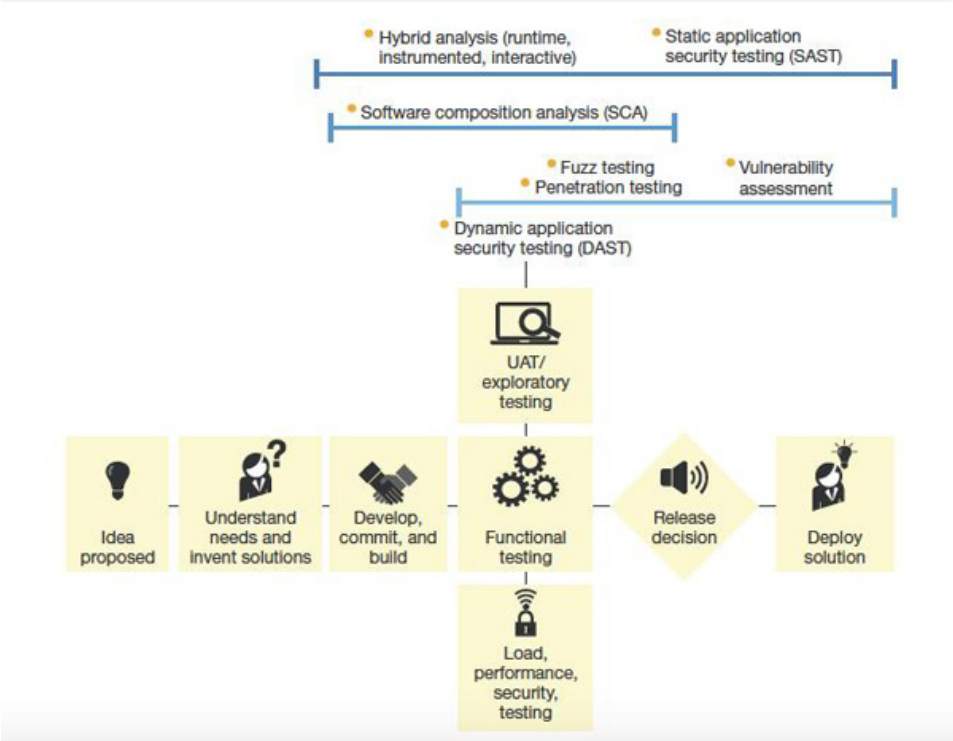

Figura 5 - Augment the life cycle with security tools

[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.