POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Informatica

Decodificare le minacce

Un'esplorazione nel mondo del Threat Modeling

Supervisori:

Prof. Cataldo Basile

Fabio Vayr

Luca Luigetti

Candidato:

Anna Lisa Belardo

Anno Accademico 2023/2024
Torino

Abstract

La sempre crescente complessità ed interconnessione dei sistemi informatici nel contesto tecnologico contemporaneo rende la sicurezza informatica una sfida critica. In questo scenario, il Threat Modeling (TM) emerge come una pratica fondamentale per identificare e mitigare le minacce all'interno dei sistemi informativi, offrendo un approccio strutturato e metodologico per valutare e gestire i rischi.

Questa tesi, condotta nell'ambito aziendale di Spike Reply, si concentra sullo studio del TM partendo da una ricerca approfondita delle principali metodologie esistenti, con particolare attenzione al framework offerto dal MITRE, in quanto usato da uno dei clienti di Spike. Dopo aver analizzato quest'ultimo ed identificate così le sue principali limitazioni, sono state condotte due attività in parallelo. La prima ha coinvolto lo sviluppo di un framework personalizzato per offrire al cliente una soluzione efficace e gratuita. L'analisi di alcune problematiche reali ha permesso di creare un catalogo di minacce consistente, confermato mediante dei Proof of Concept. La seconda attività ha riguardato l'analisi del tool commerciale IriusRisk, allo scopo di evidenziare anche le caratteristiche di uno strumento a pagamento e i pro e i contro della sua adozione.

Le conclusioni di questa ricerca non solo offrono uno sguardo sulle potenzialità future del framework elaborato, ma indicano anche le direzioni di ricerca e sviluppo nel campo della sicurezza IT, fornendo così un contributo significativo alla continua evoluzione e miglioramento delle pratiche di gestione dei rischi informatici.

A Federica,
il mio angelo più prezioso.
Ti voglio bene

Indice

Elenco delle figure

Elenco delle tabelle

Acronimi

TM Threat Modeling

ICT Information and Communication Technology

IT Information Technology

IoT Internet of Things

SDLC Software Development Life Cycle

DFD Data Flow diagram

PFD Process Flow Diagram

STRIDE Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege

PASTA Process for Attack Simulation and Threat Analysis

OCTAVE Operationally Critical Threat, Asset, Vulnerability Evaluation

SEI Software Engineering Institute

CVSS Common Vulnerability Scoring System

NIST National Institute of Standards and Technology

ISO International Organization for Standardization

CTI Cyber Threat Intelligence

TTP Tattiche,Tecniche,Procedure

SOC Security Operations Center

ICS Industrial Control Systems

RBAC Role-Based Access Control

IDOR Insicure Direct Object Reference

SSRF Server-Side Request Forgery

SaaS Software as a Service

OWASP Open Web Application Security Project

PM Project Manager

PaaS Platform as a Service

IaaS Infrastructure as a Service

AWS Amazon Web Services

GCP Google Cloud Platform

NVD National Vulnerability Database

PoC Proof of Concept

OMB United States Office of Management and Budget

CIA Confidentiality,Integrity,Availability

DN Domain Name

Capitolo 1

Introduzione

Il continuo aumento della complessità delle infrastrutture ICT (Tecnologie dell'Informazione e della Comunicazione) si presenta come una sfida considerevole nella protezione dagli attacchi informatici. Alla luce delle crescenti minacce in questi ambienti e della limitata disponibilità di risorse esperte, le organizzazioni devono esplorare approcci più efficienti per valutare la propria resilienza e adottare misure proattive. Infatti possiamo ricordare numerosi attacchi informatici avvenuti non pochi anni fa, come Mirai Botnet [1]: avvenuto nel 2016, è stato uno dei più noti e devastanti attacchi DDoS[1] Il malware Mirai infettò dispositivi IoT vulnerabili, come telecamere di sicurezza e router, sfruttando password deboli o predefinite. Questi dispositivi infettati sono stati utilizzati per lanciare attacchi DDoS su larga scala, interrompendo servizi Internet vitali. Oppure ancora il WannaCry Ransomware [2] che pur non specificamente un attacco IoT ha dimostrato quanto le vulnerabilità nei sistemi IoT possano avere impatti devastanti. Questo ransomware si è diffuso rapidamente nel 2017 sfruttando una vulnerabilità di Windows nota come EternalBlue. Infatti molti dispositivi IoT che eseguivano versioni di Windows vulnerabili sono stati colpiti dall'attacco, causando interruzioni in molte organizzazioni ed infrastrutture critiche. Secondo il rapporto Clusit[2] 2022[3], nel corso del 2021 si è registrato un aumento del 10% nei casi di attacchi informatici gravi, arrivando ad un totale di 2.049 incidenti. Questi attacchi, che si verificano con una frequenza crescente ogni mese, presentano una maggiore pericolosità: il 79% di essi ha causato un impatto significativo, con il 32% classificato come "critico" e il restante 47% con una gravità considerata "alta". Questo porta a comprendere l'importanza di

[1] *Distributed Denial of Service* basati sull'*Internet of Things* (IoT). è un attacco informatico che rende un servizio online non disponibile sovraccaricandolo di traffico da più fonti. Emerse negli anni '90, con uno dei primi attacchi documentati nel 1999 contro il sito web della CNN.

[2] Club Italiano della Sicurezza Informatica, nato nel 2020

due concetti chiave nel mondo informatico: *security by default* e *security by design*.

La **Security by Default** è un principio di progettazione e configurazione dei sistemi informatici che implica la predisposizione di impostazioni di sicurezza predefinite e ottimali sin dall'attivazione o dalla distribuzione iniziale di un sistema o di un prodotto. In altre parole, si tratta di configurare un sistema in modo tale che, al momento dell'installazione o dell'attivazione, sia già protetto da rischi di sicurezza comuni senza richiedere all'utente finale di apportare modifiche o aggiustamenti per garantire la sicurezza. L'obiettivo è quello di fornire un livello minimo di security fin dall'inizio, riducendo così la possibilità di esposizione a minacce informatiche e violazioni della sicurezza.

La **Security by Design**, invece, è un principio ed un approccio alla progettazione e allo sviluppo dei sistemi informatici che pone l'accento sull'integrazione della sicurezza fin dall'inizio del processo di sviluppo. Questo principio mira a garantire che la sicurezza sia una considerazione fondamentale in ogni fase del ciclo di vita del software, dalla fase di progettazione e sviluppo fino alla distribuzione e alla manutenzione. In pratica, significa identificare e mitigare i rischi di sicurezza sin dalle prime fasi del processo, adottando misure di *security* adeguate e integrandole nel design del sistema. Lo scopo è quello di sviluppare sistemi robusti, resistenti agli attacchi e in grado di proteggere efficacemente i dati e le risorse dell'organizzazione.

Il report 2023 di Accenture [4] sullo *State of Cybersecurity* rileva che le aziende che si occupano di integrare la sicurezza informatica *by design* come elemento distintivo della propria trasformazione digitale (circa la metà di un campione di 3mila aziende in 14 paesi e 15 settori) sono il 18% più propense ad aumentare la capacità di generare crescita dei ricavi, aumentare la quota di mercato e migliorare la soddisfazione del cliente, la fiducia e la produttività dei dipendenti. Sono sei volte più propensi rispetto alla media ad applicare pratiche di gestione del rischio all'avanguardia (65% contro 11%); sono più capaci di proteggere il proprio ecosistema produttivo: il 45% coinvolge più frequentemente i partner dell'ecosistema o della catena di approvvigionamento nel loro piano di risposta agli incidenti; il 41% richiede loro di rispettare rigorosi standard di sicurezza informatica. Tuttavia esiste ancora un 18% di aziende, spiega Accenture, che implementa soluzioni di cybersecurity dopo essersi digitalizzate e solo se vengono rilevate vulnerabilità.

1.0.1 Software Development Life Cycle

Il *Software Development Life Cycle* (SDLC) è un processo strutturato utilizzato per guidare lo sviluppo e la gestione del software. Ha radici che risalgono agli anni '60 e '70, quando sono state introdotte le prime metodologie di sviluppo del software. È nato in risposta alla necessità di organizzare e razionalizzare il processo di sviluppo per garantire che i progetti fossero completati in modo efficace, efficiente e di alta

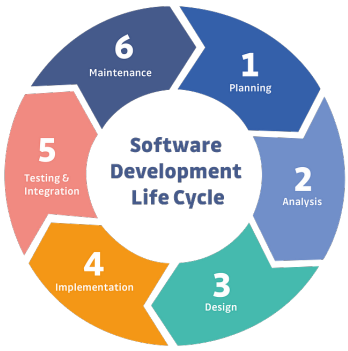


Figura 1.1: Software Development Life Cycle

qualità. Infatti, le sfide della sicurezza informatica che influenzano la definizione del SDLC includono:

- *Minacce e Vulnerabilità*: L'aumento della complessità delle minacce informatiche e la presenza di vulnerabilità nei sistemi richiedono un approccio strutturato alla sicurezza fin dalle prime fasi dello sviluppo del software.
- *Rischi per la Sicurezza*: Le organizzazioni devono affrontare una vasta gamma di rischi, tra cui perdita di dati, accessi non autorizzati e interruzioni del servizio. Un SDLC ben definito può aiutare a mitigare questi rischi attraverso l'implementazione di controlli di sicurezza appropriati.
- *Conformità Normativa*: Le normative e i regolamenti del settore richiedono alle organizzazioni di proteggere i dati e gestire i rischi per la sicurezza. Un SDLC strutturato può garantire che le applicazioni sviluppate siano conformi ai requisiti normativi.
- *Complessità Tecnologica*: L'evoluzione delle tecnologie ha reso più complesso proteggere le applicazioni da minacce sofisticate. Un SDLC ben definito fornisce linee guida chiare su come progettare, sviluppare e testare applicazioni sicure.
- *Metodologie Agili e DevOps*: L'adozione di metodologie di sviluppo agile e DevOps ha introdotto nuove sfide per garantire la sicurezza durante il ciclo di vita del software. Un SDLC adattabile può integrarsi efficacemente con queste metodologie per garantire che la sicurezza sia una priorità in ogni fase dello sviluppo.

Le fasi del *Software Development Life Cycle* variano leggermente a seconda della metodologia specifica utilizzata, ma in generale comprendono le seguenti:

1. *Pianificazione*: vengono definiti gli obiettivi del progetto, le risorse necessarie, il piano di lavoro e la pianificazione temporale. Si identificano anche i requisiti del progetto, si valutano i rischi e si stabilisce una strategia generale per il successo dell'iniziativa.

2. *Analisi*: vengono raccolte e analizzate le esigenze degli utenti e i requisiti del sistema. Questo coinvolge interviste con gli *stakeholder*, osservazioni sul campo e analisi dei documenti esistenti per comprendere appieno le necessità del progetto. L'obiettivo è definire in modo chiaro e completo i requisiti funzionali e non funzionali del software.

3. *Progettazione*: vengono definiti: l'architettura del software e i dettagli del design. Si traducono i requisiti raccolti durante la fase di analisi in un piano concreto per la creazione del software. Questo include la progettazione dell'interfaccia utente, la definizione delle componenti del sistema e la pianificazione delle interazioni tra di esse.

4. *Implementazione*: durante la fase di implementazione, il software viene sviluppato in base ai dettagli del design. Questo coinvolge la scrittura del codice, la creazione dei componenti del software e l'integrazione delle funzionalità. L'obiettivo è trasformare il design concettuale in un prodotto software funzionante.

5. *Testing*: il software viene sottoposto a una serie di test per valutarne la qualità e l'affidabilità. Questo include test funzionali per verificare che il software funzioni come previsto, test di prestazioni per valutarne le prestazioni e test di sicurezza per identificare e correggere eventuali vulnerabilità.

6. *Mantenimento*: Dopo l'implementazione, il software viene sottoposto a manutenzione continua per garantirne il corretto funzionamento nel tempo. Questo include la correzione di bug, l'implementazione di nuove funzionalità e l'aggiornamento del software per soddisfare i cambiamenti nei requisiti degli utenti o dell'ambiente operativo.

Quindi, si comprende quanto il SDLC fornisca una guida chiara ed una struttura ben definita per tutto il processo, garantendo che il prodotto finale sia di alta qualità, affidabile ed in grado di soddisfare le esigenze degli utenti.

Nel contesto attuale, dove la sicurezza delle applicazioni è diventata una priorità, è essenziale incorporare quindi le migliori pratiche di *security* in tutte le fasi dello sviluppo del software. Pertanto, nei prossimi capitoli, ci concentreremo su una di queste pratiche chiave: il Threat Modeling (TM).

Capitolo 2

Fondamenti del Threat Modeling

Il Threat Modeling è un componente essenziale del SDLC perché aiuta a garantire che il software sviluppato sia sicuro e resistente agli attacchi. Il TM ha radici storiche profonde e nel corso di questo capitolo esamineremo la sua evoluzione partendo dalle sue origini belliche fino alla sua applicazione nell'ambito dell'*Internet of Things*. Verrà analizzata nel dettaglio la sua definizione ed esploreremo il processo di costruzione di un modello delle minacce, valutando infine le diverse metodologie e framework esistenti.

2.1 Dalle origini alla pratica moderna

Comprendere l'evoluzione storica del Threat Modeling consente agli esperti della sicurezza di adottare una prospettiva strategica anziché reagire impulsivamente e in modo disorganizzato alle minacce. Anche se guardare al TM in un contesto al di fuori del mondo IT può sembrare irrilevante, è importante analizzare i suoi usi passati perchè forniscono una comprensione di come l'analisi tattica diventa parte fondamentale di un qualsiasi processo.

Nell'ambito militare, ad esempio, l'analisi delle minacce e la valutazione dei rischi sono da sempre pratiche cruciali per garantire la sicurezza e la sopravvivenza delle forze armate. Queste pratiche hanno radici storiche che risalgono ai tempi antichi, quando gli eserciti dovevano comprendere le tattiche nemiche e le vulnerabilità delle proprie difese per pianificare strategie di combattimento efficaci. Nel corso del tempo, con l'avvento della guerra moderna e delle tecnologie avanzate, l'analisi delle minacce si è evoluta per includere non solo le minacce fisiche, come le armi e

le tattiche nemiche, ma anche le minacce informatiche e cibernetiche. Inoltre, come afferma Sun Tzu: *"Solo chi conosce a fondo i mali della guerra può comprendere a fondo il modo proficuo di portarla avanti"* [5]. Quindi, voleva sottolinerare l'importanza della conoscenza dei rischi con i quali ci si può interfacciare durante la guerra da un punto di vista proficuo. Il concetto di profitto si riferisce proprio al vantaggio o alla ricompensa ottenuta dalla comprensione dei mali della guerra. Questi vantaggi sono rappresentati dai rischi evitati che potrebbero avere un impatto critico sulla missione. Il Threat Modeling consente di riconoscere meglio i pericoli mediante simulazioni ben ponderate. Anche se non tutti gli scenari possibili possono essere considerati e modellati, l'esercito cerca di sviluppare i più probabili scenari di attacco. In definitiva, il TM non è in grado di eliminare la possibilità di attacco, ma aumenta lo stato di prontezza con cui un'unità militare può rispondere efficacemente ad una minaccia. L'approccio militare al Threat Modeling ha influenzato significativamente le pratiche di sicurezza informatiche nel settore civile, contribuendo alla formalizzazione ed alla standardizzazione dei metodi di modellazione utilizzati oggi nelle organizzazioni aziendali e nei settori governativi.

La cybersecurity è essenziale per proteggere reti, computer, software e dati da minacce dannose. Senza adeguate misure di sicurezza, queste risorse rimarrebbero vulnerabili. Con l'aumento delle minacce e delle superfici di attacco, è cruciale adottare un approccio preventivo anziché reattivo alla sicurezza informatica.

2.2 Cosa si intende con Threat Modeling

*"Threat Modeling - un **processo strategico** mirato a considerare possibili **scenari di attacco** e vulnerabilità all'interno di un **ambiente applicativo** proposto o esistente allo scopo di identificare chiaramente i livelli di **rischio** e **impatto**."* [6]

Analizzando questa definizione, esploreremo approfonditamente gli elementi principali che emergono, al fine di ottenere una panoramica completa e agevole per progredire con successo nello studio del Threat Modeling.

Emerge immediatamente il concetto di *"Processo strategico"*, il quale rappresenta un elemento fondamentale del Threat Modeling che supera il semplice rilevamento delle minacce. Questo va a sottolineare la capacità del processo di TM di adottare un approccio globale e pianificato nell'analisi dei possibili rischi. Piuttosto che reagire passivamente agli attacchi una volta che si sono verificati, il Threat Modeling si concentra sull'anticipazione e sulla simulazione di scenari di attacco, permettendo alle organizzazioni di sviluppare strategie di difesa efficaci. Inoltre mette in risalto anche l'importanza di considerare non solo gli aspetti tecnici dei rischi, ma anche quelli strategici e aziendali. Questo implica valutare come gli attacchi potrebbero

influenzare gli obiettivi aziendali, la reputazione di un marchio, la conformità normativa ed altri fattori cruciali. In tal modo, il TM diventa non solo un'attività tecnica, ma anche un processo di gestione del rischio integrato nell'intera strategia aziendale.

È necessario esaminare quanti più *"scenari di attacco"* che potrebbero essere utilizzati da un attaccante per violare la sicurezza dell'applicazione. Questi scenari possono basarsi su conoscenze storiche di attacchi simili, trend nel panorama delle minacce o ipotesi su come un attaccante potrebbe cercare di sfruttare le vulnerabilità presenti. Utilizzando le metodologie di Threat Modeling, si disassembla la struttura di un attacco, mettendo in luce eventuali difetti nel design dell'applicazione e/o nel processo di sviluppo del software. Ciò consente anche di identificare i possibili motivi che hanno spinto l'attaccante ad iniziare il proprio attacco. Le vulnerabilità possono derivare da errori di programmazione, configurazioni errate o mancanza di controlli di sicurezza adeguati.

Inoltre, i rischi possono derivare anche dall'*"ambiente applicativo"* che si riferisce all'ambiente in cui opera l'applicazione oggetto di TM. Questo può essere un ambiente esistente già in uso o un ambiente proposto per un'applicazione in fase di sviluppo. È importante considerare l'ambiente in cui l'applicazione opererà, poiché questo può influenzare il tipo ed il livello di minacce che l'applicazione potrebbe affrontare.

Nel contesto del Threat Modeling, l'obiettivo è identificare e comprendere i *rischi* associati ad una applicazione o ad un ambiente informatico specifico, al fine di adottare misure di protezione adeguate per mitigarli. Il rischio è legato infine all'*"impatto"*. Calcolare l'impatto che una minaccia può avere può essere una sfida, poiché dipende da una serie di fattori complessi e variabili. Alcuni di questi fattori includono la natura della minaccia stessa, la vulnerabilità dell'applicazione o del sistema al suo attacco, il valore dei dati o delle risorse che potrebbero essere compromessi e le contromisure di sicurezza attualmente in uso. Inoltre, l'impatto di una minaccia può essere influenzato da fattori esterni come la reazione dei clienti, il rischio legale e le conseguenze sulla reputazione dell'organizzazione. Tuttavia, nonostante questa complessità, è importante cercare di valutare l'impatto in modo accurato e completo, anche se non è sempre possibile.

Il Threat Modeling aumenta la consapevolezza sulla sicurezza del prodotto coinvolgendo tutti gli attori partecipanti nello sviluppo. Garantisce un approccio integrato alla sicurezza dall'inizio alla fine del ciclo di vita del prodotto, partendo dalla progettazione fino al rilascio. Questo approccio combina sia misure proattive, implementate nelle fasi iniziali, che reattive, adottate dopo il rilascio del prodotto. Le strategie reattive includono *penetration test*, *code review* e *configuration review* per identificare e correggere vulnerabilità non previste durante le fasi iniziali. È essenziale valutare attentamente le azioni da intraprendere per garantire una

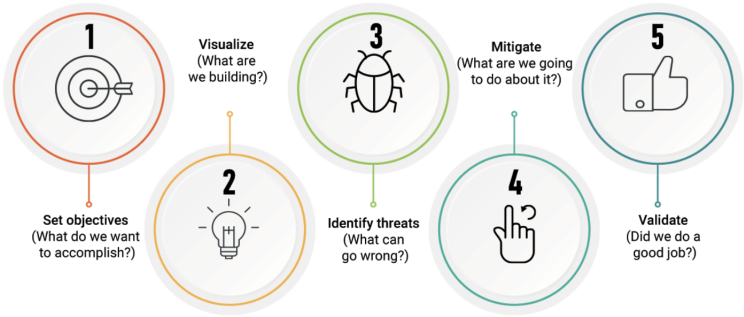


Figura 2.1: Fasi del processo di Threat Modeling

sicurezza completa.

2.3 Come costrutire un Threat Model

Un'esaustiva comprensione di come costruire un Threat Model è essenziale per garantire la sicurezza delle nostre applicazioni. Le fasi coinvolte durante il processo di TM sono le seguenti (cfr. Figura 2.1):

1. *Identificazione degli obiettivi di sicurezza*
2. *Analisi dell'applicazione*
3. *Identificazione e valutazione delle minacce*
4. *Mitigazione delle minacce*
5. *Validazione del lavoro svolto*

Fase 1 : Identificare gli obiettivi di sicurezza

Questa fase coinvolge l'identificazione e la comprensione delle risorse critiche all'interno dell'ambiente che si desidera proteggere il quale può includere, ad esempio, applicazioni software, server, database, reti, dati sensibili e qualsiasi altra risorsa che potrebbe essere obiettivo di attacchi. Prima di iniziare con gli strumenti e i metodi di Threat Modeling, è importante essere chiari sugli obiettivi che si vogliono raggiungere. Di solito, questi sono definiti tenendo presente che l'applicazione deve garantire:

- *Confidentiality*: Riservatezza per proteggere i dati da divulgazioni non autorizzate.
- *Integrity*: Integrità per prevenire modifiche non autorizzate alle informazioni.
- *Availability*: Capacità di erogare i servizi richiesti anche in caso di attacco al sistema.

Bisogna quindi rispondere ad una serie di quesiti, fornendo ad esempio le seguenti informazioni:

- I dati dei clienti che devono essere protetti, come account utente, password, numeri di carta di credito, informazioni bancarie.
- I requisiti di conformità da soddisfare, come politiche di sicurezza interne, leggi sulla privacy, regolamenti di settore.
- Specifici requisiti di qualità del servizio da considerare, compresi quelli relativi alla disponibilità e alle prestazioni dei sistemi.
- Asset immateriali che necessitano di protezione, come la reputazione aziendale, i segreti commerciali.

La comprensione di tali aspetti aiuta a stabilire non solo le basi per un solido modello di minaccia ma sopratutto le criticità dell'applicazione per l'organizzione.

Fase 2: Analizzare l'applicazione

La scomposizione dell'applicazione rappresenta il secondo passo nel processo di Threat Modeling. Consiste nel documentare in modo dettagliato i vari componenti che costituiscono il sistema. Questo passaggio è essenziale per ottenere una visione chiara e completa del prodotto, il che semplifica notevolmente il processo di valutazione delle minacce. Decomporre l'applicazione include la registrazione dei casi d'uso, dei flussi di dati, degli schemi dei dati e dei diagrammi di distribuzione, cosi da comprendere in modo approfondito tutto il funzionamento interno dell'app e sopratutto le sue interazioni con l'ambiente circostante.

Vi sono due tipi di visualizzazioni che è possibile creare:

- *Data Flow diagram* (DFD): illustra come i dati sono progettati per spostarsi attraverso il sistema. Esso mostra il livello operativo ed evidenzia chiaramente dove i dati entrano ed escono da ciascun componente.

- *Process Flow Diagram* (PFD): mostra come gli utenti interagiscono e si spostano attraverso vari scenari d'uso. Questo diagramma si concentra sull'interazione dell'utente e delle terze parti con il sistema a livello di applicazione.

Il vantaggio chiave dei PFD, rispetto ai DFD, è che i primi illuminano il modo in cui i componenti principali del sistema interagiscono con le risorse aziendali. Per questo motivo, i PFD possono essere utilizzati per dare priorità ai rischi.

Una volta individuati, quindi, gli attori e gli asset più importanti dell'applicazione, è il momento di procedere con l'identificazione e la valutazione delle minacce.

Fase 3: Identificare e Valutare le minacce

Le minacce possono provenire da fonti interne ed esterne, come hacker, malware, insider malevoli, errori umani, catastrofi naturali, e così via. La fase di identificazione è cruciale ed è necessario definire un mapping tra gli asset tecnologici o blueprint[1] e le minacce presenti nel Threat Landscape. Questo mapping consente di comprendere quali minacce sono potenzialmente applicabili ad un componente specifico all'interno di un determinato contesto di rete e flussi.

Ad esempio, consideriamo un'applicazione web che gestisce dati sensibili dell'utente. Durante questa fase, dovremmo identificare gli asset tecnologici coinvolti, come il server di database, il server web e il sistema di autenticazione. Successivamente, dovremmo analizzare il Threat Landscape per individuare le minacce che potrebbero influenzare questi componenti, come attacchi SQL injection, cross-site scripting e attacchi di brute force. Una volta completato il mapping, saremo in grado di identificare così le minacce specifiche che potrebbero compromettere la sicurezza dell'applicazione in base al contesto in cui opera. Questo ci fornisce una panoramica chiara delle vulnerabilità potenziali e ci aiuta a pianificare le misure di sicurezza appropriate per mitigare tali minacce.

Una volta identificati i *threats*, vengono valutati utilizzando delle classifiche basate su diversi fattori di rischio. Questi criteri possono essere adoperati per catalogare le minacce in diverse categorie di rischio, che includono elevato, medio o basso. Un esempio è offerto da Microsoft con l'approccio DREAD[2]. Per determinare la classifica di una minaccia, l'analista risponderà alle seguenti domande:

---

[1]Termine usato nel contesto della progettazione e della sicurezza delle informazioni per descrivere un modello o un'istanza di un ambiente tecnologico specifico

[2]Per approfondimenti si veda *https://shostack.org/files/papers/modsec08/Shostack-ModSec08-Experiences-Threat-Modeling-At-Microsoft.pdf*

1. *Damage* : Quanto sarebbe grande il danno se l'attacco avesse successo?
2. *Reproducibility*: Quanto è facile riprodurre un attacco?
3. *Exploitability*: Quanto tempo, sforzo e competenza sono necessari per sfruttare la minaccia?
4. *Affected users*: Se una minaccia fosse sfruttata, quale percentuale di utenti ne risentirebbe?
5. *Discoverability*: Quanto è facile per un utente malintenzionato scoprire questa minaccia?

Sarà quindi possibile associare ogni potenziale minaccia ad un valore specifico, creando una lista prioritaria che permette di gestirle al meglio. Questa lista è quindi ordinata secondo una prospettiva soggettiva e personalizzata in base all'applicazione oggetto di analisi.

Esistono anche metodi qualitativi per valutare il rischio e vengono considerati due aspetti chiave:

- la probabilità che venga effettuato un attacco,
- l'impatto che quell'attacco può avere sul nostro sistema

In questo caso non vi sono domande soggettive, ma piuttosto fattori come: "l'attaccante deve essere autenticato?", "l'attaccante può performare l'attacco anche da remoto?", "l'exploit può essere automatizato?".

Fase 4: Mitigare le minacce

Dopo aver identificato e valutato il rischio, si sviluppano ed implementano misure di mitigazione per ridurre la probabilità di attacco o ridurre il suo impatto. Queste misure possono includere l'applicazione di controlli di sicurezza, l'implementazione di patch, la formazione del personale, la segmentazione di rete ed altre pratiche di sicurezza. Le mitigazioni derivano da una combinazione di fonti, tra cui:

- *Best practice e linee guida di settore*: Le organizzazioni possono adottare le best practice raccomandate da organizzazioni come il *National Institute of Standards and Technology*(NIST), l'*International Organization for Standardization*(ISO) e altre entità di settore. Queste linee guida offrono raccomandazioni su come affrontare le minacce e proteggere i sistemi e i dati.

- *Analisi delle vulnerabilità*: Le mitigazioni spesso derivano dall'analisi delle vulnerabilità, che identifica le debolezze nei sistemi e nelle applicazioni. Una volta identificate le vulnerabilità, è possibile sviluppare contromisure specifiche per mitigare il rischio associato.
- *Esperienza passata*: Analizzando incidenti di sicurezza precedenti e casi di studio, è possibile identificare le cause sottostanti e sviluppare strategie di mitigazione per prevenire futuri incidenti simili.
- *Analisi del rischio*: Identifica e valuta i rischi per la sicurezza informatica associati a un particolare ambiente o sistema. Le mitigazioni sono spesso sviluppate in risposta ai risultati dell'analisi del rischio, mirando a ridurre la probabilità e l'impatto dei rischi identificati.
- *Requisiti normativi*: Le organizzazioni devono conformarsi a una serie di requisiti normativi che stabiliscono le misure di sicurezza che devono essere implementate. Le mitigazioni sono sviluppate per soddisfare questi requisiti e garantire la conformità alle normative applicabili.

Una volta individuate le minacce e le relative contromisure, è possibile classificarle in base al grado di mitigazione, secondo i seguenti criteri:

- Minacce non mitigate: rappresentano vulnerabilità prive di contromisure, che possono essere sfruttate completamente e provocare un impatto significativo.
- Minacce parzialmente mitigate: sono minacce che incontrano una o più contromisure, limitando la loro capacità di causare danni, ma che potrebbero ancora produrre un impatto limitato.
- Minacce completamente mitigate: queste minacce sono state adeguatamente affrontate con contromisure adeguate e non rappresentano più una vulnerabilità significativa.

Fase 5: Validare il lavoro svolto

Bisogna assicurarsi che tutte le minacce individuate siano state mitigate correttamente e che siano stati documentati chiaramente eventuali rischi rimanenti. Una volta completata questa fase, si passa alla pianificazione dei prossimi passi per gestire le minacce identificate e stabilire quando sarà opportuno condurre una nuova analisi di Threat Modeling. È cruciale comprendere che il TM non è un'attività isolata; piuttosto, dovrebbe essere ripetuto periodicamente o in concomitanza con specifici momenti chiave durante lo sviluppo dell'applicazione per garantire un livello di sicurezza costante ed adeguato. Di seguito alcuni esempi di quando aggiornare un modello di minaccia:

- adozione di un nuovo stack tecnologico (ad esempio web, mobile, IoT, cloud, IaC, FaaS);
- esplorazione di nuovi modelli di business (ad esempio mobile checking, self-service IT);
- ogni volta che un'applicazione cambia;
- ogni volta che i regolamenti cambiano (ad esempio, GDPR, PCI, PII)[3]

2.3.1 Errori comuni nella modellazione delle miancce

1. **Eccessiva dipendenza dai modelli predefiniti**: Molti approcci al Threat Modeling iniziano con un modello predefinito, basato su qualche modello di minaccia precedentemente efficace. Ha senso utilizzare un modello predefinito e evitare di dover iniziare ogni nuovo modello di minaccia da zero. Tuttavia, è un errore fare affidamento eccessivo sui modelli predefiniti. In primo luogo, il panorama delle minacce cambia troppo rapidamente per utilizzare qualsiasi modello predefinito così com'è. Inoltre, anche le aziende cambiano e questo deve essere riflesso anche nel modello di minaccia. Va bene iniziare con un modello predefinito, ma bisogna apportare modifiche di conseguenza.
2. **Affidarsi esclusivamente agli strumenti automatizzati**: Così come avviene con i modelli predefiniti, un'eccessiva dipendenza dagli strumenti automatizzati può portare a modelli di minaccia obsoleti o inefficaci. Così come con i modelli predefiniti, gli strumenti automatizzati possono indicare un punto di partenza che bisogna analizzare e odificare di conseguenza.
3. **Mancanza di coinvolgimento degli stakeholder**: Il Threat Modeling è un'attività a livello aziendale. Dovrebbe riflettere tutti gli aspetti della tecnologia e dell'azienda. Pertanto, non è possibile avere un quadro completo dell'azienda senza coinvolgere tutti gli stakeholder. Almeno, devono essere rappresentati esecutivi tecnologici, aziendali e fornire contributi significativi al modello di minaccia.
4. **Mancato aggiornamento del modello di minaccia**: Il panorama delle minacce cambia rapidamente. Nel cloud, cambia anche la superficie di attacco. Perché i modelli di minaccia siano efficaci, devono essere costantemente aggiornati. Ecco perché diciamo che il Threat Modeling è un processo, non un progetto. Un processo continuo per di più.

[3] *General Data Protection Regulation*, *Payment Card Industry Data Security Standard*, *Personally Identifiable Information*

5. **Ignorare le minacce non tecniche**: È facile presumere che il Threat Modeling debba modellare solo le minacce legate alla tecnologia. Minacce che sono il risultato di qualche difetto tecnologico. Ma non è così. Una delle minacce più difficili da modellare (e difendersi) sono le minacce basate sulle persone, sia con che senza intento. Con intento c'è il dipendente malintenzionato intenzionato a nuocere all'organizzazione. Senza intento è qualsiasi dipendente suscettibile di essere vittima di *social engineering* e/o attacchi di *phishing*. Questo include praticamente tutti. Quando si fa il TM, è un errore trascurare le minacce non tecniche.

6. **Concentrarsi su scenari di minaccia specifici invece dell'insieme completo**: La reazione istintiva potrebbe essere quella di esaminare un elenco delle principali vulnerabilità e concentrare l'energia di Threat Modeling su quelle. E va bene, ma non a discapito dell'insieme completo. Gli hacker adottano un approccio olistico nei confronti delle organizzazioni e del panorama delle minacce. Se tutti sono concentrati sulle principali vulnerabilità, gli hacker capiscono che saranno più efficaci nell'evitarle. Ecco perché è un errore non tenere presente l'insieme completo quando si modellano le minacce.

2.4 Principali metodologie e framework utilizzati

Nella nostra esplorazione del Threat Modeling e della valutazione della sicurezza informatica, ci addentriamo nei diversi approcci e strumenti che guidano questa pratica. Esistono, infatti, tre principali approcci utilizzati per analizzare le minacce, come anche riportato nell'articolo *"A Review of Asset-Centric Threat Modelling Approaches"* [7]:

- approccio centrato sugli asset, il che significa che si concentra sui beni critici del sistema che potrebbero essere bersaglio di attacchi;
- approccio centrato sugli attaccanti che si focalizza sulle possibili azioni che questi ultimi potrebbero compiere per compromettere il sistema;
- approccio centrato sul software o sul sistema stesso, che analizza le vulnerabilità presenti nel software o nell'architettura del sistema.

Questa sezione delinea un'esplorazione dettagliata, grazie all'articolo pubblicato dalla *Carnegie Mellon University* [8], dei principali metodi e framework utilizzati nel campo della sicurezza informatica. Dalle metodologie consolidate alle prospettive innovative, esaminiamo come queste risorse offrano una base fondamentale per affrontare le minacce informatiche in modo strategico ed efficace. È importante sottolineare che questa analisi delle metodologie ha rappresentato un punto di

partenza necessario, fornendo una panoramica completa dello stato dell'arte attuale. Questa comprensione approfondita ha poi guidato ulteriori avanzamenti e sviluppi nel percorso di ricerca ed implementazione.

2.4.1 STRIDE

Il framework STRIDE è stato sviluppato da Loren Kohnfelder e Praerit Garg durante il loro lavoro presso Microsoft. È stato introdotto per la prima volta nel libro *"Threat Modeling: Designing for Security"* di Adam Shostack, pubblicato nel 2014 [9]. Introdurre il concetto di STRIDE significa introdurre un metodo organizzato per esaminare e mitigare i rischi legati alla sicurezza informatica. Esplorare ciascuna categoria di minaccia all'interno di Stride consente alle organizzazioni di valutare in modo completo e dettagliato i potenziali punti deboli del proprio ambiente informatico e di adottare misure preventive e correttive adeguate.

Il processo inizia con la definizione di un diagramma di flusso dati, fondamentale per identificare confini del sistema, entità ed eventi. L'accuratezza dei DFD influenza l'efficacia di STRIDE. Tuttavia, affidarsi esclusivamente ai DFD come unico input per il Threat Modeling è limitante, poiché non tiene conto delle decisioni architetturali relative alla sicurezza. Per questo è stato definito un secondo step che riguarda l'identificazione delle minacce bansandosi su un insieme generale creato a partire dall'acrononimo di STRIDE: *Spoofing identity*, *Tampering with data*, *Repudiation*, *Information disclosure*, *Denial of service*, e *Elevation of privilege*. Questo acronimo può essere utilizzato come mnemonico per scoprire le minacce durante la navigazione del modello del sistema creato nella fase uno. Per aiutare in questo passaggio, alcune fonti offrono checklist e tabelle che assistono nella descrizione delle minacce, delle violazioni delle proprietà, dei tipici bersagli e delle azioni di un attaccante. Dopo aver raccolto le minacce scoperte e le strategie di mitigazione, queste informazioni dovrebbero essere documentate e priorizzate.

Anche se facile da implementare, può richiedere molto tempo, soprattutto in sistemi complessi dove il numero delle minacce può crescere rapidamente. Si è dimostrato che questo metodo ha una bassa incidenza di risultati falsi positivi ed una moderata incidenza di risultati falsi negativi.

La tabella 2.1 sintetizza le categorie di minacce che STRIDE considera.

2.4.2 PASTA

Il *Process for Attack Simulation and Threat Analysis* (P.A.S.T.A.) è un framework di Threat Modeling che si concentra sull'analisi dei processi aziendali per identificare e mitigare le minacce informatiche. È stato sviluppato da Tony UcedaVélez nel

Minaccia	**Proprietà violata**	**Definizione Minaccia**
Spoofing identify	Authentication	Pretendere di essere qualcuno o qualcosa diverso da te stesso
Tampering with data	Integrity	Modificare qualcosa sul disco, memoria, rete etc..
Repudiation	No-Repudiation	Affermare di non aver fatto qualcosa o di non essere responsabile
Information disclosure	Confidentiality	Fornire info a qualcuno non autorizzato
Denial of service	Availability	Esaurire le risorse necessarie per fornire un servizio
Elevation of privilege	Authorization	Consentire a qualcuno di fare qualcosa per cui non è autorizzato

Tabella 2.1: STRIDE Threat Categories

2012 come risposta alla crescente complessità delle minacce informatiche e alla necessità di approcci più pratici e mirati alla gestione dei rischi.

Il framework PASTA si compone di sette fasi, ciascuna delle quali affronta specifiche attività mirate ad identificare, valutare e mitigare le minacce:

1. *Planning*: si definiscono gli obiettivi del TM e si stabilisce il contesto operativo. Si identificano le risorse critiche e si coinvolgono le parti interessate per ottenere una comprensione completa delle esigenze aziendali.

2. *Asset Modeling*: Si procede a modellare gli asset critici dell'organizzazione, che possono includere dati sensibili, sistemi IT, applicazioni e infrastrutture di rete. Questo passaggio aiuta a comprendere meglio quali risorse devono essere protette e quali potrebbero essere soggette a minacce.

3. *Threat Analysis*: si esaminano le minacce potenziali che potrebbero mirare agli asset identificati. Si utilizzano modelli di minaccia come STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) per identificare le diverse categorie di minacce.

4. *Vulnerability Analysis*: Si analizzano le vulnerabilità presenti nei sistemi e nelle applicazioni aziendali che potrebbero essere sfruttate dagli attaccanti. Questo può coinvolgere la revisione del codice, la scansione dei sistemi ed altre tecniche di analisi della sicurezza.

5. *Risk Assessment*: Viene valutato il rischio associato a ciascuna minaccia identificata, considerando la probabilità di occorrenza e l'impatto potenziale

sull'organizzazione. Questa valutazione aiuta a stabilire le priorità per la mitigazione delle minacce.

6. *Mitigation Planning*: Si sviluppano piani per mitigare le minacce e le vulnerabilità identificate. Questo può includere l'implementazione di contromisure tecniche, la revisione delle politiche di sicurezza e la formazione del personale per migliorare la consapevolezza sulla sicurezza.

7. *Results Communication*: Infine, i risultati dell'analisi delle minacce e delle attività di mitigazione vengono comunicati alle parti interessate dell'organizzazione. Questo aiuta a garantire una comprensione condivisa dei rischi di sicurezza e delle azioni necessarie per affrontarli.

Il framework, quindi, si propone di portare il processo di Threat Modeling ad un livello più strategico, coinvolgendo i principali stakeholder decisionali all'interno dell'organizzazione. Questo approccio si focalizza principalmente sull'analisi dei rischi, adottando il punto di vista dell'attaccante. L'obiettivo è fornire un'analisi dettagliata e quantitativa delle minacce, mettendo in primo piano gli asset aziendali e fornendo una valutazione accurata delle minacce attraverso sistemi di punteggio e classificazione.

2.4.3 Attack trees

Gli *Attack Trees* sono un metodo di modellazione delle minacce utilizzato per visualizzare e analizzare le possibili vie attraverso le quali un attaccante potrebbe compromettere un sistema o raggiungere un obiettivo specifico. Questo approccio si basa sulla costruzione di alberi gerarchici che rappresentano graficamente le varie fasi di un attacco, partendo da un nodo radice che rappresenta l'obiettivo dell'attaccante e ramificandosi in nodi figli che rappresentano i possibili percorsi o le azioni che l'attaccante potrebbe compiere per raggiungere quell'obiettivo. Ogni nodo dell'albero può essere associato ad una specifica azione o tattica utilizzata dall'attaccante, e ogni ramo rappresenta una serie di passaggi necessari per compiere quell'azione. Un esempio è riportato in Figura 2.2, dove il caso d'uso è "Rubare denaro da un account bancario" e si estende in due sottocasi: il primo se il denaro viene prelevato presso un ATM ed il secondo se viene hackerato l'account online. A sua volta il primo sottocaso si estende in altri due sottocasi e così via.

Uno dei vantaggi nell'utilizzo di questa metodologia è la capacità di consentire un'analisi dettagliata delle possibili minacce, consentendo agli analisti di valutare la probabilità e l'impatto di ciascuno scenario di attacco. Questo può aiutare le organizzazioni a pianificare ed implementare misure di sicurezza mirate per proteggere i propri sistemi. Tuttavia, gli Attack Tree possono diventare complessi

Figura 2.2: Esempio Attack Tree

da gestire in presenza di sistemi molto grandi e la loro efficacia dipende dalla qualità delle informazioni utilizzate per costruirli. Inoltre, poiché gli Attack Trees sono statici e non tengono conto dei cambiamenti nel panorama delle minacce nel tempo, è importante mantenerli aggiornati e revisionarli periodicamente per assicurarsi che rimangano pertinenti e utili.

2.4.4 OCTAVE

OCTAVE, acronimo di *Operationally Critical Threat, Asset, and Vulnerability Evaluation*, è un framework sviluppato dal Software Engineering Institute (SEI) presso la Carnegie Mellon University per valutare e migliorare la sicurezza delle informazioni all'interno di un'organizzazione. A differenza di altre metodologie focalizzate principalmente su questioni tecniche, OCTAVE si concentra sulle operazioni aziendali e sulle pratiche di sicurezza, analizzando come queste influenzino i rischi informatici.

Il processo OCTAVE si articola in tre fasi fondamentali:

- *Costruzione di Profili di Minaccia Basati sugli Asset*: Questa fase prevede l'individuazione degli elementi cruciali all'interno dell'organizzazione, come dati sensibili, infrastrutture critiche e processi aziendali. Si analizzano le minacce che possono compromettere tali asset e si valutano le strategie di protezione.

- *Analisi delle Vulnerabilità dell'Infrastruttura*: In questa fase si esamina l'infrastruttura tecnologica utilizzata dall'organizzazione per individuare eventuali punti deboli che potrebbero essere sfruttati dagli attaccanti.

- *Definizione di Strategie di Sicurezza e Piani di Azione*: Una volta identificati i rischi principali, si sviluppano strategie e piani d'azione per mitigare tali rischi e rafforzare la sicurezza complessiva dell'organizzazione.

OCTAVE è progettato principalmente per organizzazioni di grandi dimensioni, ma esiste anche una versione adattata per realtà più piccole, denominata OCTAVE-S. Inoltre l'ultima versione rilasciata è chiamata OCTAVE Allegro che è stato applicato in un caso di studio presso Airlangga University [10] per valutare il security risk dell'IoT di case smart. Pur richiedendo un impegno considerevole in termini di tempo, questo metodo si distingue per la sua flessibilità e approfondimento nella valutazione dei rischi informatici.

2.4.5 TRIKE

TRIKE è un approccio di Threat Modeling open source orientato sulla valutazione della sicurezza da una prospettiva di gestione del rischio e difesa. Come molti altri approcci, Trike comincia delineando un sistema. In questa fase, l'analista sviluppa un modello dei requisiti comprendendo gli attori, gli asset, le azioni pianificate e le regole del sistema. Questo processo porta alla creazione di una matrice attore-asset-azione, dove gli asset sono rappresentati nelle colonne e gli attori nelle righe. Ogni cella di questa matrice viene suddivisa in quattro parti, corrispondenti alle azioni CRUD (Create, Read, Update, Delete), e l'analista assegna loro uno dei tre valori: consentito, non consentito o regolamentato. Successivamente, si procede con la costruzione di un DFD. Attraverso l'analisi del DFD, l'analista individua le minacce, classificandole in due categorie principali: elevazione dei privilegi e negazione del servizio. Ogni minaccia individuata diventa quindi un nodo radice in un attack tree.

La sua complessità potrebbe rappresentare una sfida per i team meno esperti, poiché richiede un livello di competenza più elevato rispetto ad altre metodologie di Threat Modeling. Inoltre, condurre un'analisi TRIKE approfondita può essere un processo che richiede molto tempo e risorse. Potrebbe essere necessaria una formazione adeguata per implementarlo in modo efficace all'interno di un'organizzazione. Infine, potrebbero sorgere sfide nell'allineare le misure di sicurezza identificate con gli obiettivi aziendali. Questo è particolarmente vero in contesti in cui ci sono priorità concorrenti o visioni divergenti sull'importanza delle minacce identificate.

2.4.6 Tool

- **Threat modeling software**:
  - Microsoft Threat Modeling Tool

  - IriusRisk
  - ThreatModeler

- **Code analysis techniques**:
  - SAST (*static application security testing*): analizza il codice sorgente;
  - DAST (*dynamic application security testing*): sonda il front-end per individuare vulnerabilità;
  - Automated SAST: Strettamente integrato con CI/CD, utilizza scansioni incrementali.

- **Automated security testing tools**:
  - OWASP ZAP (OWASP Zed Attack Proxy): strumento open-source progettato per testare la sicurezza delle applicazioni web. Utilizzato principalmente per eseguire test di sicurezza automatizzati e manuali, ZAP offre funzionalità come il fuzzing, l'iniezione di payload, l'intercettazione del traffico e l'analisi dei risultati dei test.
  - Burp Suite: è una suite di strumenti di sicurezza delle applicazioni web sviluppata da PortSwigger. È ampiamente utilizzata dagli specialisti della sicurezza informatica per testare la sicurezza delle applicazioni web durante lo sviluppo e dopo il rilascio.

- **Security testing frameworks**:
  - OWASP ASVS
  - NIST SP 800-53

2.5 Prioritizzazione delle minacce

Uno dei principali aspetti del Threat Modeling è la prioritizzazione delle minacce scoperte e in questa sezione vengono discusse alcune linee guida per fare ciò.

2.5.1 CVSS

Il CVSS, acronimo di *Common Vulnerability Scoring System*, rappresenta uno standard essenziale nell'ambito della sicurezza informatica, offrendo un metodo strutturato per valutare ed assegnare un punteggio numerico alla gravità delle vulnerabilità rilevata nei sistemi informatici. Il suo focus è quello di fornire agli operatori della sicurezza un mezzo chiaro ed obiettivo per comprendere e comunicare

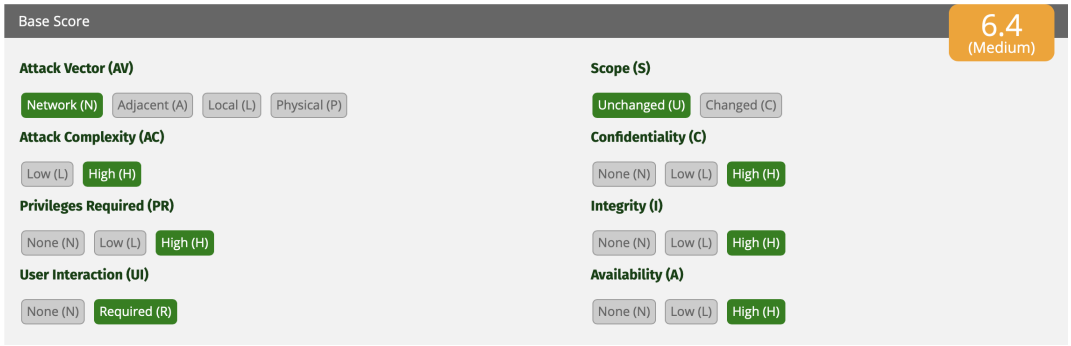


Figura 2.3: Esempio CVSS Calculator

il livello di rischio associato ad una determinata vulnerabilità, permettendo così di prendere decisioni in merito alla priorità di intervento ed alle risorse da impiegare per mitigare le minacce.

Il sistema di punteggio del CVSS è suddiviso in tre metriche principali:

- *Base Score*: Questa metrica valuta l'impatto potenziale della vulnerabilità in assenza di mitigazioni, includendo sottometriche che considerano l'effetto sulla riservatezza, sull'integrità e sulla disponibilità dei dati.
- *Temporal Score*: Questa metrica tiene conto di fattori che possono variare nel tempo, come la disponibilità di patch o l'evoluzione delle minacce informatiche.
- *Environmental Score*: Questa metrica si concentra sull'impatto specifico della vulnerabilità all'interno dell'ambiente operativo di un'organizzazione, considerando fattori come la configurazione di sicurezza, le contromisure implementate ed il contesto operativo.

Il risultato è un punteggio numerico compreso tra 0 e 10, dove punteggi più alti indicano una maggiore gravità della vulnerabilità. È disponibile una calcolatrice online che genera questo punteggio automaticamente (figura 2.3).

Uno dei principali vantaggi del CVSS è la sua standardizzazione. Questo sistema fornisce un *framework* comune ed uniforme per valutare le vulnerabilità, consentendo una valutazione coerente e comparabile delle minacce da parte di diverse organizzazioni e individui. Inoltre, è relativamente semplice da comprendere e utilizzare. Il suo sistema di punteggio fornisce una scala chiara per valutare la gravità delle vulnerabilità, rendendo più facile per gli utenti non esperti comprendere ed interpretare i risultati.

Il CVSS offre anche una vasta gamma di metriche per valutare diversi aspetti delle vulnerabilità, come il livello di accesso necessario per sfruttarle e l'impatto

potenziale. Questo fornisce una visione più dettagliata e completa delle minacce. Tuttavia, nonostante i suoi vantaggi, questa calcolatrice presenta anche alcune limitazioni. Ad esempio, il processo di assegnazione dei punteggi può essere soggetto ad interpretazione soggettiva, portando a variazioni nei punteggi assegnati per la stessa vulnerabilità. Inoltre, valuta le vulnerabilità in modo isolato, senza considerare pienamente il contesto specifico dell'ambiente in cui si trova il sistema vulnerabile. Questo può portare a valutazioni inesatte della reale minaccia che una vulnerabilità rappresenta per un'organizzazione. Infine, questo standard può diventare complesso quando si considerano tutte le metriche e i dettagli coinvolti nella sua valutazione, rendendolo difficile da interpretare per gli utenti meno esperti.

2.5.2 NIST Approach

L'approccio del *National Institute of Standards and Technology* (NIST) alla prioritizzazione delle minacce può essere trovato nella Pubblicazione Speciale 800-154 del NIST[4]. Per aiutare nella prioritizzazione delle minacce, il NIST si concentra sugli obiettivi di sicurezza dei dati (in contrasto con il sistema). In particolare, la confidenzialità, l'integrità e la disponibilità vengono utilizzate per stabilire la priorità.

Il primo passo è identificare quali set di dati sono più preziosi. Il secondo passo è identificare quale dei tre obiettivi è più importante per un particolare set di dati. Il terzo passo è identificare quali sono le minacce a quell'obiettivo di sicurezza.

Linee guida per quantificare la probabilità e l'impatto di ciascuna minaccia

Quantificare la probabilità e l'impatto di una minaccia in modo efficace è meglio realizzato adottando una prospettiva di gestione del rischio. Poiché i rischi individuali variano da un'organizzazione all'altra, le stesse minacce possono generare valutazioni molto diverse in termini di probabilità e impatto per organizzazioni diverse. Esistono numerose risorse disponibili che possono essere utilizzate per quantificare il rischio. Di seguito è riportato un elenco di alcune di esse:

- ISO/IEC 27001[5]

[4]M. Souppaya, K. Scarfone, "Guide to Data-Centric System Threat Modeling", March 2016

[5]standard che definisce i requisiti per un sistema di gestione della sicurezza delle informazioni (ISMS), anno pubblicazione 2005.

- ISO/IEC 27002[6]
- NIST Cybersecurity Framework[7]

[6]standard che fornisce linee guida e raccomandazioni per la gestione della sicurezza delle informazioni, anno pubblicazione 2005.

[7]insieme di linee guida, standard e best practice sviluppato dal National Institute of Standards and Technology (NIST) degli Stati Uniti per migliorare la sicurezza informatica delle organizzazioni, anno pubblicazione 2014.

Capitolo 3

Obiettivi pratici della tesi

Dopo gli studi preparatori effettuati sulla tematica del Threat Modeling, questo capitolo intende offrire una panoramica chiara delle motivazioni che hanno ispirato il presente lavoro di tesi, nonché degli obiettivi specifici che si è cercato di raggiungere.

3.1 Introduzione al problema

Questo lavoro di tesi è stato svolto in ambito aziendale presso Spike Reply, una società di consulenza specializzata in cybersecurity e protezione dei dati personali all'interno del Gruppo Reply. La sua missione primaria è garantire la sicurezza dei dati e della privacy delle persone, delle aziende e dei processi, contribuendo così alla crescita di un ambiente digitale globale attraverso l'innovazione. Durante questo periodo, ho concentrato il mio lavoro presso un importante cliente nel settore manifatturiero, dove mi sono dedicata all'analisi del loro approccio corrente al Threat Modeling. E' emerso che il cliente non eseguiva correttamente il processo di TM, in quanto non era svolto durante la fase iniziale di valutazione. In pratica, il Threat Modeling avveniva dopo l'*assessment*, dove venivano prima eseguite attività reattive come *penetration test* e *configuration review*. Gli output di queste attività, noti come *findings*[1], venivano poi mappati con le tattiche e le tecniche offerte dal framework del MITRE (argomento approfondito nel prossimo capitolo), framework di riferimento del cliente per guidare il processo di modellazione delle minacce. Inoltre, analizzando questi output, ho notato che la matrice fornita dal MITRE

[1]utilizzato per la prima volta nel contesto della ricerca e dell'indagine scientifica, si riferisce alle scoperte o alle constatazioni relative a vulnerabilità, rischi o problemi identificati durante un'analisi o un'esame.

non offriva neanche un adeguato supporto, evidenziando delle lacune nel processo complessivo.

3.2 Obiettivi

L'obiettivo della tesi è stato quello di valutare le ragioni per cui il framework del MITRE non si adattasse adeguatamente alle esigenze specifiche del cliente ed attraverso interviste approfondite ho cercato di comprendere al meglio le loro necessità e le sfide incontrate durante il processo di Threat Modeling. Questo approccio mi ha permesso di identificare le aree critiche in cui il MITRE non fornisce una soluzione ottimale e di esplorare, quindi, possibili alternative per migliorare l'efficacia del processo di valutazione delle minacce. Esaminando questi concetti, emerge chiaramente l'importanza del Threat Modeling lungo l'intero ciclo di vita di un qualsiasi prodotto e sebbene esistano numerose metodologie e framework disponibili, è importante comprendere e selezionare quello più adatto alle proprie necessità. Tali esigenze possono variare in base a diversi fattori, come il contesto aziendale, le risorse disponibili e le sfide specifiche che l'organizzazione deve affrontare. Pertanto, la scelta del metodo di Threat Modeling più idoneo deve essere guidata da una valutazione attenta ed approfondita delle circostanze e dei requisiti specifici dell'azienda.

3.3 Fasi del lavoro di tesi

Il lavoro è stato così suddiviso in diverse fasi :

1. **Studio e analisi del framework del MITRE**: Questa fase ha coinvolto un'analisi dettagliata della struttura dell'ATT&CK ovvero il framework definito dal MITRE, delle sue metodologie e delle sue varie applicazioni pratiche. In particolare ho esplorato le diverse categorie di tattiche e tecniche e ne ho approfondito l'utilizzo nel contesto della sicurezza informatica. Questo mi ha permesso di acquisire una conoscenza delle caratteristiche e delle potenzialità del framework, fornendomi una base solida per valutare la sua efficacia e la sua adattabilità.

2. **Analisi delle criticità e delle limitazioni del framework del MITRE**: La fase successiva si è concentrata sull'analisi approfondita delle criticità e delle limitazioni del MITRE ATT&CK. L'obiettivo principale era comprendere meglio perché l'attività di Threat Modeling non stesse funzionando in modo ottimale. Sono stati individuati i punti deboli dell'ATT&CK e si è compreso

chiaramente come tali limitazioni potessero influenzare le esigenze specifiche dell'organizzazione.

3. **Ricerca di soluzioni alternative**: Una volta comprese le limitazioni del framework del MITRE, ho avviato una ricerca approfondita di soluzioni alternative. Questa fase è stata suddivisa in due attività principali:

   (a) sviluppo di un framework ad-hoc, personalizzato e compatto, progettato appositamente per rispondere alle esigenze specifiche del cliente;

   (b) scouting di piattaforme di mercato per il supporto di attività di Threat Modeling allo scopo di valutare come efficientare il modello definito e avere un benchmark di riferimento secondo lo stato dell'arte disponibile da *best of bread*[2] dei vendor specializzati.

4. **Tool a confronto**: Infine, è stata condotta un'analisi di confronto tra la soluzione personalizzata implementata e quella offerta da IriusRisk (tool commerciale selezionato per l'analisi) comprendendo i pro e i contro di entrambe le alternative per poter offrire infine quella migliore al cliente.

[2]Locuzione inglese che significa "il migliore della categoria" o "il migliore della razza". Viene utilizzata per indicare un prodotto, un servizio o una soluzione che è considerata la migliore disponibile nel suo settore o categoria.

Capitolo 4

MITRE ATT&CK

In questo capitolo, verrà analizzato il framework del MITRE, partendo dalle motivazioni che hanno ispirato la sua creazione.

4.1 Introduzione al framework

Nell'ambito della cybersecurity, la *pyramid of pain*[1] costituisce un modello fondamentale per l'efficace utilizzo della Cyber Threat Intelligence (CTI) nelle operazioni di rilevamento delle minacce, con l'obiettivo principale di aumentare il costo operativo per gli aggressori. Infatti questa piramide è composta da diversi livelli, in ordine crescente di difficoltà nel contrastare le minacce informatiche. Questi livelli, rappresentati in Figura 4.1, includono:

- *Hash Values*: rappresenta l'indicatore più basico e immediato di una possibile minaccia informatica. Gli hash values sono delle stringhe alfanumeriche generate da algoritmi di hash e utilizzate per identificare univocamente file o dati.
- *IP Addresses*: gli indirizzi IP sospetti o noti per essere associati ad attività malevole costituiscono un altro livello di indicatori di compromissione.
- *Domain Names* (DN): possono essere utilizzati per identificare siti web o server sfruttati da aggressori per condurre attività dannose o per ospitare malware.

[1]Termine introdotto per la prima volta da David Bianco, un noto esperto di sicurezza informatica, nel 2013. Bianco ha sviluppato questo concetto per illustrare visivamente la relativa efficacia delle tecniche di rilevamento e di mitigazione delle minacce utilizzate dagli analisti di sicurezza.

- *Network/Host Artifacts*: si riferisce a qualsiasi traccia o residuo lasciato da un'attività malevola su una rete o su un host. Questi artefatti possono includere file sospetti, registri di sistema modificati, connessioni di rete anomale, modelli di traffico insoliti e altro ancora.
- *Tools*: si riferiscono agli strumenti, software o programmi utilizzati dagli attaccanti per condurre attività dannose o compromettere un sistema informatico. Questi strumenti possono includere malware, exploit, kit di hacking, script personalizzati.
- *Tactics, Techniques, and Procedures* (TTPs): questo rappresenta il livello più alto della piramide e indica le metodologie e le strategie utilizzate dagli aggressori per portare avanti i loro attacchi informatici. Le TTPs includono pratiche avanzate di hacking, tecniche di evasione, strategie di compromissione e altre attività sofisticate.

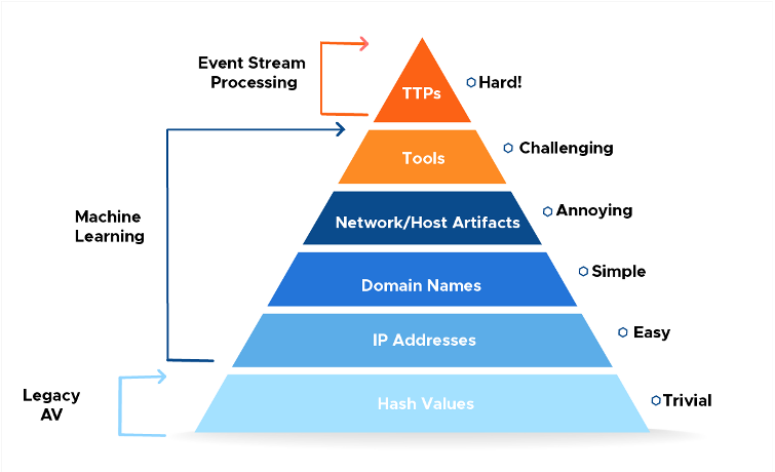


Figura 4.1: Pyramid of pain

In cima alla piramide si trovano quindi le metodologie che un attaccante può sfruttare, che rappresentano il livello più sofisticato e difficile da rilevare. Queste minacce richiedono un costante studio e monitoraggio per ottenere una conoscenza solida ed evitare che situazioni problematiche si ripetano. È proprio questa necessità di comprensione approfondita delle TTPs che ha portato alla nascita del framework del MITRE, il quale si propone di fornire un approccio strutturato per la categorizzazione e la gestione delle minacce informatiche, aiutando le organizzazioni a migliorare la loro capacità di difesa contro gli attacchi IT.

4.2 Analisi del framework

Il MITRE ATT&CK, *Adversarial Tactics, Techniques, and Common Knowledge*, emerge come un potente framework utilizzato per analizzare le minacce informatiche e pianificare le strategie di difesa. Creato dalla *MITRE Corporation* nel 2013, un'organizzazione senza scopo di lucro che si occupa di ricerca e sviluppo tecnologico, il framework ATT&CK fornisce una ricca panoramica delle tattiche, delle tecniche e delle conoscenze comuni utilizzate dagli attaccanti in varie fasi di un attacco. ATT&CK adotta, quindi, il punto di vista degli aggressori nelle sue descrizioni, facilitando la comprensione delle azioni e delle contromisure nel contesto da analizzare. Questo approccio è in contrasto con molti modelli di sicurezza che si concentrano sulla difesa senza considerare il comportamento degli attaccanti.

4.2.1 Casi d'uso

I casi d'uso offrono esempi pratici di come l'ATT&CK possa essere applicato per migliorare la sicurezza informatica e proteggere le organizzazioni. Prendendo come riferimento il documento rilasciato dalla MITRE Corporation [11] analizziamo diversi casi d'uso che dimostrano come le varie TTPs utilizzate dagli attaccanti possano essere identificate, analizzate e mitigate attraverso l'implementazione di difese mirate.

- *Detection Analytics*: è una tecnica che aiuta ad identificare e prevenire le attività fraudolente. Comporta l'uso dell'analisi dei dati per rilevare schemi ed anomalie. Ad esempio, MITRE ATT&CK è un framework che può aiutare i difensori informatici a sviluppare analisi che rilevino le tecniche utilizzate da un avversario;
- *Threat Intelligence*: settore della cybersecurity che prevede la raccolta, l'elaborazione e l'analisi dei dati per comprendere le motivazioni, gli obiettivi e i comportamenti degli attaccanti;
- *Adversary emulation*: è un tipo di impegno del Red Team che imita una minaccia nota per un'organizzazione, integrando le informazioni sulle minacce per definire le azioni e i comportamenti utilizzati dal "team rosso". Il red teaming è il processo di utilizzo di tattiche, tecniche e procedure per emulare le minacce del mondo reale al fine di addestrare e misurare l'efficacia delle persone, dei processi e della tecnologia utilizzati per difendere gli ambienti;
- *SOC Maturity Assessment*: il *Security Operations Center* (SOC) di un'organizzazione è un componente critico di molte reti aziendali di medie e grandi

dimensioni che monitora continuamente le minacce attive contro la rete. Comprendere il livello di maturità di un SOC è importante per determinarne l'efficacia. ATT&CK può essere utilizzato come uno dei parametri per valutare quanto un SOC sia efficace nel rilevare, analizzare e rispondere alle intrusioni. Similmente alla valutazione dei gap difensivi, una valutazione della maturità del SOC si concentra sui processi che un SOC utilizza per individuare, comprendere e rispondere alle minacce in evoluzione alla propria rete nel tempo.

4.2.2 Componenti

Tattiche, Tecniche, Procedure e Mitigazioni

Il framework MITRE ATT&CK è fondato sull'analisi dettagliata delle TTPs utilizzate dagli attaccanti durante le fasi di un'operazione offensiva. In quanto modello comportamentale, ATT&CK si basa quindi su una serie di componenti fondamentali:

- *Tattiche*: rappresentano gli obiettivi strategici che l'attaccante cerca di raggiungere. Alcuni esempi di tattiche includono l'ottenimento di accesso non autorizzato, il movimento laterale all'interno di una rete e il danneggiamento dei dati.
- *Tecniche*: Sono metodi specifici utilizzati dagli attaccanti per raggiungere le tattiche identificate. Ogni tattica può essere implementata attraverso diverse tecniche, ognuna delle quali rappresenta un modo specifico in cui gli attaccanti possono agire per perseguire un obiettivo. Ad esempio, le tecniche possono includere il phishing, l'utilizzo di malware, l'abuso di credenziali rubate o l'esecuzione di exploit per sfruttare vulnerabilità nei sistemi.
- *Sotto-Tecniche*: Raffigurano mezzi più specifici con cui gli avversari raggiungono obiettivi tattici ad un livello inferiore rispetto alle tecniche.
- *Procedure*: Forniscono una visione più dettagliata e specifica di come gli attaccanti possono attuare le tecniche durante un attacco. Queste possono includere passaggi puntuali, script o azioni specifiche che gli attaccanti seguono per portare a termine una tecnica.
- *Mitigazioni*: Definire le contromisure che potrebbero impedire agli avversari di raggiungere i loro obiettivi tattici attraverso l'uso di tecniche specifiche. Le mitigazioni affrontano la questione del "cosa fare" con le TTPs.

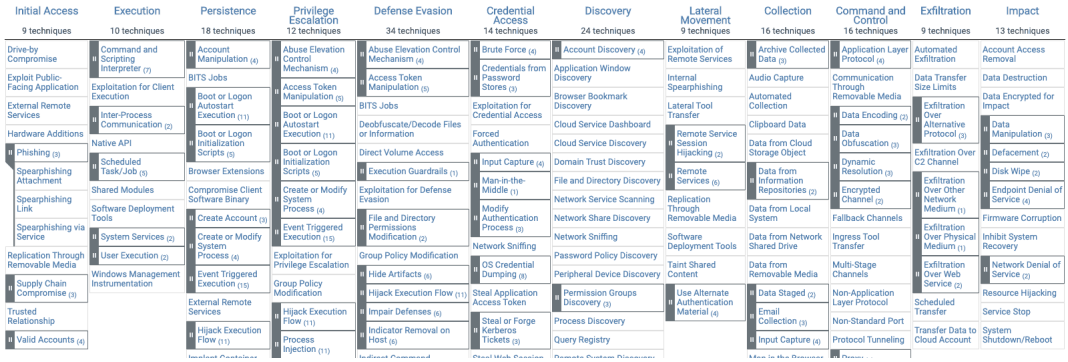


Figura 4.2: ATT&CK: Matrix for Enterprise

Matrici

A causa delle numerose TTPs, il MITRE ha suddiviso il suo database in diverse tabelle chiamate matrici: *Enterprise*, *Mobile*, *Industrial Control Systems* (ICS). Ogni matrice è organizzata in righe e colonne, con le righe che rappresentano le diverse tattiche utilizzate dagli attaccanti e le colonne che rappresentano le specifiche tecniche o procedure associate a ciascuna tattica. Il dominio *Enterprise* è il più esteso, composto da 196 tecniche e 411 sotto-tecniche. Questo dominio è ricco di attacchi a causa del suo ampio ambito tecnologico. Comprende piattaforme come Windows, macOS, Linux, Cloud, Network e Containers. Inoltre, è l'unico dominio che include tecniche specifiche (all'interno di 2 tattiche) per analizzare le attività preparatorie avversarie (PRE-ATT&CK) che sono "*Reconnaissance*" e "*Resource Development*". Per quanto riguarda il dominio *ICS* , questo comprende 81 tecniche. Si concentra sulle minacce e le vulnerabilità specifiche dei sistemi di controllo industriale, che sono cruciali per settori come il manifatturiero, l'energia e i servizi pubblici. Infine, il dominio *Mobile* include 66 tecniche e 41 sotto-tecniche ed è mirato alle minacce ed alle vulnerabilità dei dispositivi mobili come smartphone e tablet. La figura 4.2 mostra un esempio di *matrix for enterprise* che contiene al suo interno tutte le fasi di una sequenza di attacco e in figura 4.3 si evidenzia che per ciascun elemento della matrice è possibile esaminare in dettaglio tutte le informazioni connesse ad esso.

Gruppi

I gruppi nel MITRE ATT&CK sono categorie specifiche di attori minacciosi, come gruppi di hacker, criminali informatici o gruppi di stato, che operano con obiettivi o attributi simili e tecniche di attacco comuni. Questi gruppi vengono identificati e catalogati sulla base di prove e osservazioni raccolte attraverso attività di intelligence

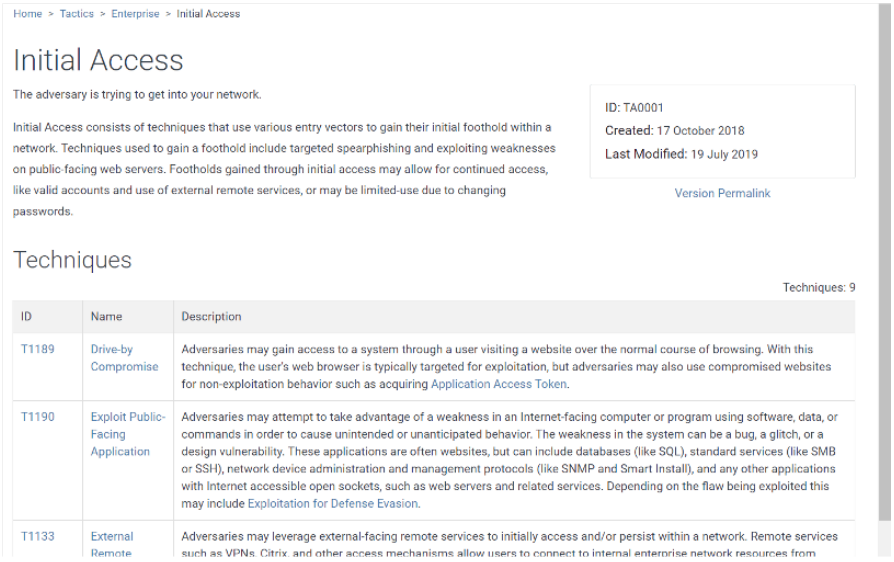


Figura 4.3: ATT&CK: Dettaglio di una Tattica

sulla minaccia, condotte dal MITRE e dalla comunità di sicurezza informatica.

Ad esempio, uno dei gruppi più noti all'interno del MITRE ATT&CK è *APT29*, conosciuto anche come Cozy Bear, un gruppo di attori minacciosi presumibilmente associato al governo russo. APT29 è stato associato ad una serie di attacchi sofisticati e mirati a scopo di spionaggio e sabotaggio in tutto il mondo. Le tattiche e le tecniche di attacco utilizzate da APT29 sono state documentate e catalogate all'interno del MITRE ATT&CK, consentendo agli analisti di sicurezza di comprendere meglio il comportamento di questo gruppo e adottare misure di difesa appropriate. Un altro esempio è il gruppo *FIN7*, noto per le sue attività di frode finanziaria e il furto di dati di pagamento. FIN7 è stato coinvolto in una serie di attacchi contro organizzazioni finanziarie e commerciali in tutto il mondo e le loro tattiche e tecniche di attacco, come il *phishing*, l'infiltrazione di malware e il furto di credenziali sono state tutte documentate e analizzate all'interno del framework.

Attraverso *Navigator*, strumento interattivo offerto dal MITRE, gli utenti possono esplorare il framework, compresi i gruppi e le loro tecniche associate (Figura 4.4). Quando l'utente accede al MITRE ATT&CK Navigator, trova la scheda "Gruppi", che consente di esplorare una vasta gamma di gruppi di attori minacciosi catalogati. Selezionando un gruppo specifico, l'utente ottiene una panoramica delle TTPs associate a quel collettivo. Questa visualizzazione aiuta l'utente a comprendere meglio il comportamento del gruppo e le sue strategie di attacco.

Il *Navigator* consente anche di esaminare le relazioni tra gruppi, tecniche ed

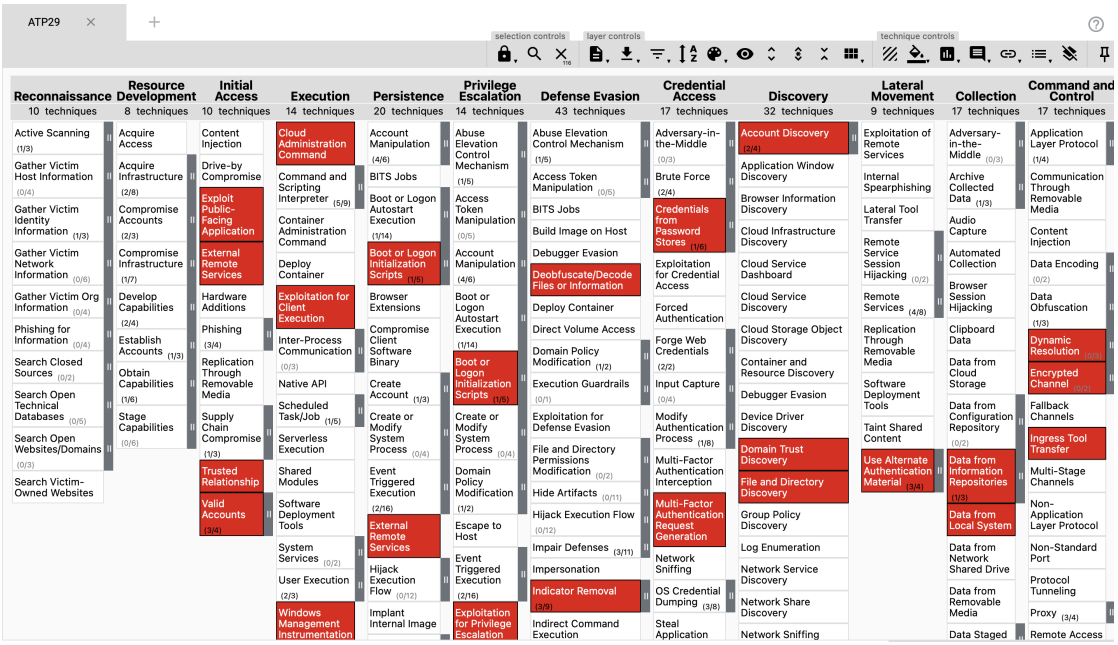


Figura 4.4: ATP29

altre entità all'interno del framework ATT&CK. È possibile visualizzare come le tecniche associate ad un gruppo possano essere collegate a quelle utilizzate da altri gruppi o come possano essere combinate per condurre specifici tipi di attacchi. Utilizzando le informazioni fornite dal *Navigator*, l'utente può pianificare le proprie difese in modo più efficace.

4.3 Applicazione della metodologia del MITRE

Il framework MITRE ATT&CK può essere utilizzato sia in modo proattivo che reattivo. In un approccio proattivo, le organizzazioni possono utilizzare la conoscenza fornita da ATT&CK per identificare potenziali vulnerabilità e punti deboli nel proprio ambiente di sicurezza prima che vengano sfruttati dagli attaccanti. Quindi viene svolta prima un'analisi preventiva per analizzare proattivamente il proprio ambiente, identificando potenziali vulnerabilità e punti deboli. Poi viene definto uno *scenario planning*, ovvero l'organizzazione può sviluppare scenari di minaccia ipotetici che riflettono le possibili modalità di attacco contro i propri sistemi. Utilizzando le informazioni fornite dal framework, l'organizzazione può così implementare mitigazioni mirate per ridurre il rischio di attacco e rafforzare la propria sicurezza informatica.

D'altro canto, nell'approccio reattivo, il framework può essere utilizzato per

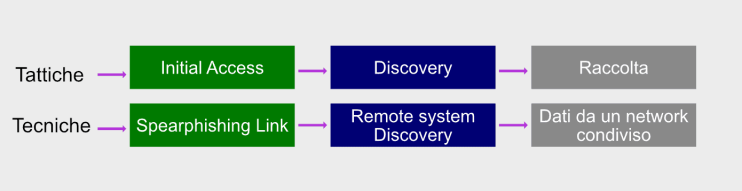


Figura 4.5: Esempio di attacco MITRE ATT&CK

analizzare incidenti passati o attività sospette, confrontando il comportamento degli attaccanti con le tecniche e le tattiche documentate in ATT&CK. Questo aiuta le organizzazioni a comprendere meglio la natura degli attacchi subiti, identificare eventuali lacune nelle loro difese e sviluppare piani di mitigazione per evitare che gli stessi tipi di attacchi si verifichino in futuro.

Non è essenziale per un aggressore impiegare tutte le tattiche elencate nella parte superiore delle matrici. Piuttosto, l'aggressore utilizzerà il minor numero possibile di tattiche per raggiungere il proprio obiettivo, poiché questo approccio risulta più efficiente e riduce le possibilità di essere scoperto. Ad esempio, nell'attacco descritto nella figura 4.5 , l'attaccante acquisisce l'accesso iniziale alle credenziali di un assistente amministrativo del CEO attraverso un link di *spear phishing* inviato via email. Successivamente, sfruttando le credenziali ottenute, l'attaccante esegue un'operazione di scoperta *Remote System Discovery* come parte della fase di *Discovery* [12].

Immaginiamo che l'obiettivo sia trovare dati sensibili all'interno di una cartella Dropbox accessibile anche all'amministratore, quindi non è necessario elevare i privilegi. La fase finale di raccolta avviene scaricando i file dalla cartella Dropbox alla macchina dell'attaccante. Fissate le linee guida per l'analisi comportamentale, un esperto di sicurezza potrebbe individuare l'attacco in atto riconoscendo il comportamento strano dell'utente. Questo corrisponde all'obiettivo di un SOC, che dovrebbe vigilare su tali attività. Supponiamo, infatti, che l'amministratore abbia interagito con un link mai visitato prima da nessuno nell'azienda e poi accede ad una specifica cartella su Dropbox in un momento inconsueto. Nella fase finale dell'attacco, il dispositivo dell'attaccante si connette per la prima volta alla cartella Dropbox. Attraverso l'analisi comportamentale, queste azioni potrebbero essere identificate come comportamenti insoliti dell'utente, sollevando un allarme di possibile compromissione.

4.4 Criticità e Limitazioni

Sebbene il framework MITRE ATT&CK possa apparire estremamente completo e dettagliato, con il potenziale di eclissare altre metodologie di Threat Modeling, è importante riconoscere che presenta alcune limitazioni nell'applicazione pratica che possono influire sulla sua efficacia nell'ambito operativo e richiedono una valutazione attenta per garantire una gestione adeguata dei rischi. Alcune delle limitazioni più evidenti includono:

- *Ampiezza delle minacce non completamente coperta*: potrebbe concentrarsi maggiormente su minacce tradizionali e ben note, trascurando quelle più recenti o specifiche di determinati settori o tecnologie.
- *Adattabilità ai contesti applicativi e cloud*: potrebbe essere meno adattabile ai contesti applicativi moderni ed alle infrastrutture cloud complesse. Questo potrebbe portare ad una mancanza di precisione nel rilevamento e nella mitigazione delle minacce che sono particolarmente rilevanti in questi ambienti.
- *Complessità e difficoltà di utilizzo*: la complessità del framework potrebbe rendere difficile la sua adozione ed implementazione da parte di organizzazioni con risorse limitate o senza una profonda esperienza nel campo della sicurezza informatica. Questo potrebbe portare ad una sottovalutazione o ad una cattiva applicazione delle pratiche di Threat Modeling.
- *Aggiornamenti e manutenzione*: potrebbe non essere aggiornato regolarmente per includere le nuove minacce e le migliori pratiche emergenti. Questo potrebbe far sì che diventi rapidamente obsoleto e meno efficace nel tempo, specialmente considerando la rapida evoluzione del panorama delle minacce informatiche.
- *Limitazioni nell'integrazione con altri processi di sviluppo*: potrebbe non essere completamente integrabile con altri processi di sviluppo del software, come DevOps o Agile, rendendo difficile la sua adozione all'interno di organizzazioni che utilizzano queste metodologie. Ciò potrebbe limitare la sua efficacia nel contesto di un approccio di sviluppo del software più moderno e dinamico.

4.4.1 Evidenze pratiche

Come evidenziato negli obiettivi della tesi, questo studio è stato condotto presso un cliente che utilizzava il MITRE per condurre l'attività di Threat Modeling. Questa circostanza ha consentito di esaminare da vicino le limitazioni dell'ATT&CK. A

ID	THREAT TACTICS	THREAT TECHNIQUES	Is applicable (pre-assessment)?	Is applicable (post-assessment)?	Is exploitable (post-assessment)?	IMPACT EVALUATION Pre-assessment			IMPACT EVALUATION Post-assessment			Probability	Severity
1	Reconnaissance	Active Scanning			X							0.7	Medium
2	Reconnaissance	Gather Victim Host Information											
3	Reconnaissance	Gather Victim Identity Information			X							0.5	Low
4	Reconnaissance	Gather Victim Network Information											
5	Reconnaissance	Gather Victim Org Information											
6	Reconnaissance	Phishing for Information											
7	Reconnaissance	Search Closed Sources											
8	Reconnaissance	Search Open Technical Databases											
9	Reconnaissance	Search Open Websites/Domains											
10	Reconnaissance	Search Victim-Owned Websites											
11	Resource Development	Acquire Infrastructure											
12	Resource Development	Compromise Accounts											
13	Resource Development	Compromise Infrastructure											
14	Resource Development	Develop Capabilities											
15	Resource Development	Establish Accounts											
16	Resource Development	Obtain Capabilities											
17	Resource Development	Stage Capabilities											
18	Initial Access	Drive-by Compromise	X										
19	Initial Access	Exploit Public-Facing Application											
20	Initial Access	External Remote Services											
21	Initial Access	Hardware Additions											
22	Initial Access	Replication Through Removable Media											
23	Initial Access	Spearphishing Attachment											
24	Initial Access	Spearphishing Link	X										
25	Initial Access	Spearphishing via Service											

Figura 4.6: Template Threat Model MITRE ATT&CK

tale scopo, erano stati sviluppati fogli di lavoro specifici come si vede in Figura 4.6, riportando le tattiche e le tecniche offerte dal MITRE.

Uno dei problemi più significativi riscontrati è scaturito dal fatto che l'attività di Threat Modeling veniva eseguita tardivamente, quando le iniziative erano già in fase di *testing* e non più nelle fasi iniziali di progettazione. Di conseguenza, si passava direttamente ad attività reattive come *penetration test* e *configuration review*, a seconda se si trattava di un'iniziativa applicativa o cloud, per identificare i rischi derivanti da implementazioni difettose o configurazioni errate.

Queste attività generavano output sotto forma di minacce che potevano causare rischi significativi al prodotto. Tali minacce venivano di conseguenza mappate con le tecniche fornite dalle matrici del MITRE, comportanto due principali criticità:

1. *Adattabilità delle Mitigation offerte*: Se il mapping poteva essere eseguito in modo consistente ed il rischio evidenziato durante le attività era presente nella matrice del framework, spesso si riscontravano problemi con le Mitigation offerte. Le soluzioni proposte non erano coerenti o appropriate rispetto alla natura del rischio, lasciando lacune nella protezione del prodotto.

   Viene riportato di seguito un esempio reale di un'attività svolta su un'iniziativa cloud in ambiente Azure. In questo caso specifico, è stata condotta un'attività di Configuration Review, che consiste nell'analizzare le configurazioni abilitate e disabilitate andando ad identificare eventuali rischi di sicurezza. Ogni configurazione offerta da Azure permette di effettuare determinate azioni e offre livelli di sicurezza differenti. Durante questa analisi, si valuta il servizio di Azure utilizzato e si controlla che le configurazioni adottate per quel servizio siano adeguate per prevenire rischi. Tra i vari servizi usati nell'iniziativa analizzata, uno di interesse era *Azure Virtual Machine*, e l'output su cui ci

Mitigations

ID	Mitigation	Description
M1036	Account Use Policies	Use conditional access policies to block logins from non-compliant devices or from outside defined organization IP ranges.[68]
M1015	Active Directory Configuration	Disable legacy authentication, which does not support MFA, and require the use of modern authentication protocols instead.
M1013	Application Developer Guidance	Ensure that applications do not store sensitive data or credentials insecurely. (e.g. plaintext credentials in code, published credentials in repositories, or credentials in public cloud storage).
M1027	Password Policies	Applications and appliances that utilize default username and password should be changed immediately after the installation, and before deployment to a production environment.[69] When possible, applications that use SSH keys should be updated periodically and properly secured. Policies should minimize (if not eliminate) reuse of passwords between different user accounts, especially employees using the same credentials for personal accounts that may not be defended by enterprise security resources.
M1026	Privileged Account Management	Audit domain and local accounts as well as their permission levels routinely to look for situations that could allow an adversary to gain wide access by obtaining credentials of a privileged account. [3] [70] These audits should also include if default accounts have been enabled, or if new local accounts are created that have not be authorized. Follow best practices for design and administration of an enterprise network to limit privileged account use across administrative tiers. [71]
M1018	User Account Management	Regularly audit user accounts for activity and deactivate or remove any that are no longer needed.
M1017	User Training	Applications may send push notifications to verify a login as a form of multi-factor authentication (MFA). Train users to only accept valid push notifications and to report suspicious push notifications.

Figura 4.7: Mitigazioni del MITRE per Valid Account

focalizzeremo è relativo al "valid account". In tale scenario un problema emerso è che un attaccante potrebbe ottenere e abusare delle credenziali di un utente per acquisire "Initial Access" o "Privilege escalation".

Si può notare,come si evince dalla Figura 4.7, che la minaccia è presente nella matrice del MITRE ma le mitigazioni offerte, non forniscono indicazioni precise, ma presentano diverse opzioni che potrebbero non abbassare il rischio in modo efficace.

Infatti, dalla documentazione di Azure emerge che è necessario applicare un'unica mitigazione ovvero utilizzare il sistema *Role-Based Access Control* (RBAC), un servizio gratuito offerto da Azure, per contenere questa vulnerabilità.

Riporto, anche, un esempio pratico legato ad un'iniziativa applicativa. Si tratta di un'applicazione web e mobile dove, dopo l'attività di penetration test, sono state individuate delle vulnerabilità significative che richiedono interventi di mitigazione. In particolare, è emerso che l'applicazione è vulnerabile all'IDOR (*Insicure Direct Object Reference*): gli utenti potrebbero accedere a dati o eseguire azioni che non dovrebbero essere autorizzati a fare. Nel dettaglio, durante il test è stato osservato che era possibile modificare il *personal number* dopo l'accesso dell'utente. Tuttavia, modificando questo numero, gli utenti potevano successivamente richiedere i dati di altri utenti, eludendo così le autorizzazioni. Anche in questo caso, il mapping con il MITRE potrebbe risultare coerente, ma le mitigazioni offerte sono vaghe e non affrontano direttamente il problema.

2. *Mapping forzato*: Le minacce identificate non corrispondevano facilmente alle tecniche di attacco specifiche descritte nelle matrici, rendendo difficile applicare le migliori pratiche di sicurezza suggerite dal framework. I motivi per cui spesso non è possibile mappare una minaccia con le tecniche offerte dal MITRE sono molteplici:

   (a) *Specificità della minaccia*: La minaccia individuata potrebbe essere molto specifica e non corrispondere esattamente a nessuna delle tecniche descritte nel database del MITRE. Questo potrebbe essere dovuto a nuove tecniche di attacco non ancora documentate o a scenari di minaccia unici che non rientrano nelle categorie esistenti.

   (b) *Limiti della documentazione del MITRE*: Nonostante il framework fornisca una vasta gamma di tecniche, potrebbe non coprire tutte le possibili minacce o scenari di attacco. La documentazione potrebbe essere limitata o potrebbero esserci delle lacune nelle descrizioni delle tecniche.

   (c) *Evoluzione delle minacce*: Le minacce informatiche evolvono rapidamente, e nuove metodologie e strumenti vengono costantemente sviluppati dagli attaccanti. Ciò significa che le minacce attuali potrebbero non essere ancora state identificate o documentate nel database del MITRE.

   (d) *Ambienti e tecnologie specifiche*: Le minacce possono variare a seconda dell'ambiente e delle tecnologie utilizzate. Le organizzazioni possono adottare configurazioni personalizzate o utilizzare piattaforme tecnologiche uniche che potrebbero non essere coperte dalle tecniche standard del MITRE.

   Un esempio pratico di questa limitazione è emerso dopo aver condotto un'attività di penetration test dove sono state individuate delle vulnerabilità significative che richiedono interventi di mitigazione. In particolare, è emerso che l'applicazione in esame era vulnerabile al *Server-Side Request Forgery* (SSRF): una vulnerabilità di sicurezza che consente ad un utente malintenzionato di indurre l'applicazione lato server ad effettuare richieste HTTP verso un dominio arbitrario da lui scelto. Nel caso specifico, la vulnerabilità riscontrata era un SSRF "cieco", in cui l'applicazione può essere indotta ad inviare richieste back-end ad un target specificato, ma non riceve mai una risposta nel front-end. In generale questa vulnerabilità potrebbe essere sfruttata per eseguire una scansione delle porte, identificando quindi quali sono aperte. E poiché questa minaccia potrebbe avere un impatto significativo, è fondamentale fornire mitigazioni adeguate. Il problema principale è stato il tentativo di mappare questa minaccia con le tecniche presenti nel database del MITRE, dato che non esiste nulla di così specifico. Si è stati quindi costretti a forzare la corrispondenza con "Active Scanning". La forzatura del mapping

non è stata efficace poichè le mitigazioni associate alla tecnica del MITRE non erano mirate alla mitigazione della vulnerabilità riscontrata.

Capitolo 5

Sviluppo ed Implementazione di un Framework Personalizzato

L'errata esecuzione del processo di Threat Modeling presso il cliente e le limitazioni del framework MITRE, hanno condotto all'esigenza di rivalutare il catalogo delle minacce sul quale lavorare, analizzando poi nello specifico, per ogni minaccia inserita nel catalogo, delle *remediation* pratiche e concise.

Nel seguente capitolo non verranno elencate tutte le nuove minacce e le remediation definite, ma per consultare l'intero progetto, si rimanda al *repository* su Github: https://github.com/ALB-19/Threat-Modeling-methodology.git.

5.1 Metolodologia

Al fine di mantenere il legame tra la vecchia soluzione adottata ed il nuovo framework e quindi non perdere durante la transizione informazioni rilevanti, è stata effettuata una valutazione preliminare della modalità di passaggio.

Considerando che nel MITRE il livello più alto è costituito dalle Tattiche, si è partito da queste per costruirne delle altre in base alle esigenze effettive del cliente.

Dopo aver identificato le nuove tattiche di interesse, come si può osservare in Figura 5.1, è stato effettuato un monitoraggio costante delle iniziative, andando ad identificare quali minacce potessero far parte di ciascuna tattica. Sono stati prima definiti gli ambienti di interesse: *mobile application*, *PaaS/IaaS*, etc..., per definire poi della *blueprint*, Figura 5.2, per avere un set di *default* di minacce per ciascuna blueprint.

Tactics		Description
MITRE ATT&CK	**NEW**	
Reconnaissance	**Reconnaissance**	The adversary is trying to gather information they can use to plan and support future operations.
Resource Development		
Discovery		
Collection		
Credential Access	**Exfiltration**	The adversary is trying to steal account names, passwords and data.
Exfiltration		
Defense Evasion	**Defense Evasion**	The adversary is trying to avoid being detected.
Initial Access	**Initial Access**	The adversary is trying to get into your network.
Impact	**Denial of Service**	The adversary is trying to interrupt, degradate or destroy the target systems and data.
Command and Control	**Command and Control**	The adversary is attempting to gain higher-level permissions, aiming to access additional functionalities or systems, including both vertical (elevating within a hierarchy) and horizontal (exploiting authorization issues) movements.
Persitence		
Execution		
Lateral Movement	**Privilege Escalation**	The adversary is trying to gain higher-level permissions with the aim to reach more functionalities or systems (for example doing lateral movements).
Privilege Escalation		
	Fraud	The adversary is trying to manipulate or use the target systems and data to perpetrate frauds.
	Data Manipulation	The adversary is trying to manipulate data.
	Physical Threats	Data loss caused by force majeure

Figura 5.1: Mapping Tattiche

ENVIRONEMENT	BLUEPRINT
Mobile Application COTS	Mobile
Mobile Application Custom	
PaaS/IaaS - COTS	Cloud
PaaS/IaaS - Custom	
SaaS	Web
On-Prem App. COTS	
On-Prem App. Custom	
Infrastructure	Infrastructure

Figura 5.2: Blueprint

Questo è risultato uno step fondamentale per fornire una base solida su cui nuove iniziative potessero lavorare e avere quindi una guida costante *pre-assessment*. Per ogni minaccia, è stato definito:

- *remediation*, cioè azioni volte a ridurre o eliminare il rischio che la minaccia venga sfruttata;
- come calcolare il rischio e la severity, quindi metodi di prioritizzazione;
- è stato previsto un parametro per indicare se l'attore malevolo potesse essere interno o esterno.

Attraverso attività reattive si comprende, infine, se quella minaccia, *post-assessment*, è *applicable* o *exploitable*. Dire che una vulnerabilità è "applicable" vuol dire che potrebbe manifestarsi, invece con "exploitable" si indica che esiste una modalità di sfruttamento di quella vulnerabilità, la quale può essere utilizzata da un attaccante per ottenere un vantaggio non autorizzato.

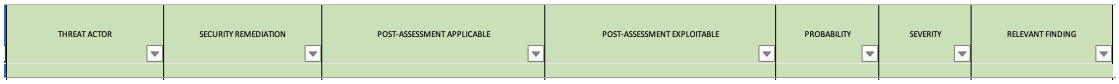


Figura 5.3: View filtri per iniziative

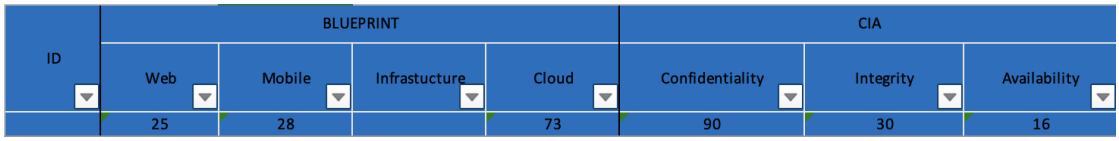


Figura 5.4: Filtri framework personalizzato - view 1

In Figura 5.3 si mostra quanto appena descritto, ovvero le colonne di interesse che devono essere compilate in fase di TM dallo specialista informatico.

Le Figure 5.4, 5.5, 5.6 mostrano lo schema di filtraggio organizzato nel foglio di lavoro excel dove, oltre a poter filtrare per blueprint, si può filtrare per valori come: confidenzialità, integrità e disponibilità e servizi. Il filtro *service* aiuta a velocizzare il processo, in quanto, considerando la vastità di componenti che esistono in ambito applicativo come in ambito cloud e considerando che la maggior parte delle volte non vengono usati tutti insieme, si può filtrare e considerare le minacce legate solo agli specifici componenti in scope durante la progettazione.

Lo strumento è quindi costituito dalla seguente lista di informazioni:

- *ID*: identificativo univoco della minaccia;
- *Blueprint*: dominio di interesse;
- *CIA*: confidentiality, integrity, availability;
- *Tactics*: tattiche precedentemente definite;
- *Threat Name*: nome della minaccia
- *Description*: descrizione della minaccia;
- *Security Remediation*: rimedi per evitare o ridurre la possibilità che quella minaccia avvenga;
- *References*: riferimento diretto con OWASP e documentazione Azure;
- *References with MITRE*: riferimenti, quando sono possibili, con il MITRE per facilitare la transizione dal modello usato in passato con quello nuovo definito;
- *Services*: mapping diretto tra minaccia e servizio sia in ambiente applicativo (es. Login Panel, Network Infrastructure, Payment Panel) e sia in ambiente Azure.

Figura 5.5: Filtri framework personalizzato - view 2

Figura 5.6: Filtri framework personalizzato - view 3

E' riportato in Figura 5.7 il diagramma di flusso per avere una visione chiara di tutti gli step seguiti per arrivare ad ottenere il framework personalizzato.

5.2 Minacce

Creare un catalogo ad-hoc ed accurato delle minacce è fondamentale per diversi motivi:

- *Conoscenza approfondita dei rischi*: Identificare e comprendere le minacce specifiche che potrebbero compromettere la sicurezza del nostro sistema o delle nostre risorse digitali è il primo passo per proteggerle in modo efficace. Un catalogo dettagliato permette di avere una visione chiara delle potenziali minacce e dei rischi connessi.
- *Pianificazione della difesa*: Con un catalogo delle minacce ben definito, si è in grado di pianificare e implementare misure di sicurezza mirate e adeguate per mitigare i rischi. Possiamo concentrarci sulle vulnerabilità più critiche e sviluppare strategie di difesa efficaci per affrontarle.
- *Risposta rapida agli incidenti*: Quando si verifica un incidente di sicurezza, è essenziale poter rispondere rapidamente per limitare i danni. Un catalogo delle minacce ci fornisce un punto di riferimento per valutare la gravità dell'incidente e adottare le contromisure appropriate senza perdere tempo prezioso nell'identificazione della minaccia.
- *Migliore gestione dei rischi*: Con un catalogo delle minacce consistente, possiamo valutare e gestire i rischi in modo più efficace. Possiamo assegnare priorità alle minacce in base alla loro gravità e alle potenziali conseguenze, consentendoci di concentrare le risorse e gli sforzi dove sono più necessari.
- *Adattabilità alle nuove minacce*: Il panorama delle minacce informatiche è in continua evoluzione, con nuove minacce che emergono regolarmente.

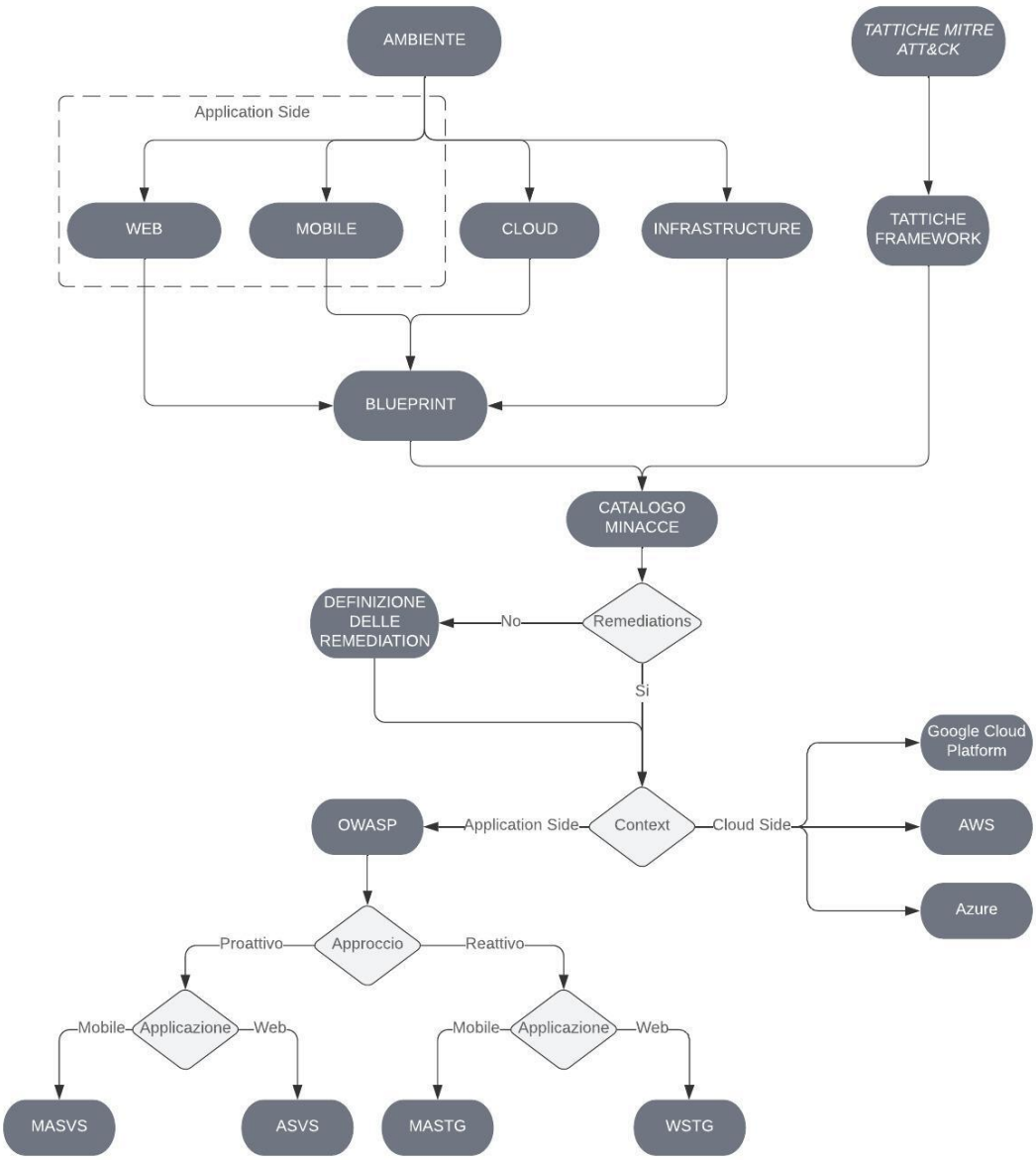


Figura 5.7: DFD Framework personalizzato

Mantenere un catalogo delle minacce aggiornato e consistente permette di adattarsi rapidamente ai cambiamenti e di proteggere proattivamente i nostri sistemi da nuove minacce.

Per ogni ambiente di interesse, vengono analizzate quali sono le reference utilizzate per identifivare le minacce considerate rimanendo ancorati al fatto che è impossibile creare un catalogo che vada bene per sempre ma appunto bisogna aggiornalo periodicamente per evitare criticità.

5.2.1 Application-Side

Si parla di "ambiente applicativo" cui ci si riferisce specificamente ai contesti in cui le applicazioni software operano. Questi ambienti possono includere diverse piattaforme e tipologie di applicazioni, quali:

1. *Applicazioni web*: applicazioni che vengono eseguite su un server web e sono accessibili tramite un browser web. Le applicazioni web possono essere pubbliche o private e offrono una vasta gamma di funzionalità, dalle semplici pagine web statiche alle complesse applicazioni web interattive.

2. *Applicazioni mobile*: applicazioni progettate per essere eseguite su dispositivi mobili come smartphone e tablet. Le applicazioni mobile possono essere native, sviluppate specificamente per una piattaforma mobile come iOS o Android, o basate sul web, eseguite all'interno di un browser mobile.

3. *Ambiente SaaS* (Software-as-a-Service): Nonostante i SaaS siano in realtà servizi cloud, sono stati trattati come applicazioni nel contesto della modellazione delle minacce, poiché il processo di identificazione e gestione delle minacce, dopo delle valutazioni con il team tecnico, è risultato analogo per entrambi. Questo è un modello di distribuzione del software in cui l'applicazione è ospitata su un server remoto e resa disponibile agli utenti attraverso Internet. Gli utenti accedono all'applicazione tramite un'interfaccia web o mobile e pagano un abbonamento o una tariffa basata sull'uso.

Il punto cardinale di tale ambiente è il sistema hardware, che è essenzialmente il telaio su cui si basa tutto. Questo può essere: server potenti, computer robusti, dispositivi di rete intelligenti e dispositivi di archiviazione affidabili, tutti indispensabili per garantire che l'applicazione funzioni senza intoppi. Poi ci sono i sistemi operativi, come Windows, Linux o macOS, che fungono da linguaggio comune tra l'applicazione e l'hardware sottostante. Sono come il terreno su cui l'applicazione poggia le sue radici, fornendo il supporto necessario per eseguire tutte le sue funzioni. Per sviluppare e gestire l'applicazione, ci sono strumenti e

piattaforme specializzate come librerie di codice, framework, database e ambienti di sviluppo integrati (IDE). Questi sono come gli strumenti del mestiere per gli artigiani digitali, consentendo loro di plasmare e perfezionare l'applicazione secondo le loro esigenze. Oltre a ciò, ci sono le risorse di rete che consentono all'applicazione di comunicare con altri sistemi e utenti, sia all'interno dell'organizzazione che su Internet. Queste sono le vie di comunicazione che collegano l'applicazione al mondo esterno, consentendo lo scambio di dati e informazioni vitali. Naturalmente, non possiamo dimenticare i dati e le risorse che l'applicazione utilizza o elabora durante le sue operazioni. Questi possono essere database ricchi di informazioni, file di configurazione che guidano il comportamento dell'applicazione e file di log che registrano ogni azione e evento. Tutto questo avviene sotto lo sguardo vigile delle politiche di sicurezza e dei regolamenti. Questi sono come le leggi e le regole che governano il comportamento dell'applicazione, garantendo che sia conforme agli standard di sicurezza, alla privacy dei dati e alla normativa vigente. E infine, ma non meno importante, ci sono gli utenti e i ruoli che interagiscono con l'applicazione. Ogni utente ha il proprio ruolo, autorizzazioni e privilegi, che determinano cosa possono fare all'interno dell'applicazione e come possono farlo. Insieme, tutti questi elementi e fattori creano l'ambiente applicativo in cui vive e prospera un'applicazione software, svolgendo il suo ruolo nel mondo digitale.

Progetto OWASP

Nel 2001 nasce il progetto *Open Web Application Security Project* (OWASP), con lo scopo di promuovere la coscienza e la cultura nel merito delle problematiche della sicurezza degli applicativi. OWASP ha sviluppato una guida *open-source* per lo sviluppo delle applicazioni web e mobile sicure, rivolgendosi:

- agli sviluppatori, per implementare correttamente i meccanismi di sicurezza ed evitare quindi le principali vulnerabilità;
- ai project manager (PM) e agli analisti, fornendo *best practices* e indicazioni di riferimento per le attività di progettazione, di analisi dei rischi, per il code review e il penetration test;
- ai team di sicurezza per apprendere le tematiche di application security e l'approccio per la messa in sicurezza di applicativi web.

Catalogo Minacce e Security Remediation

Il catalogo implementato ha avuto come punto di partenza proprio la conoscenza che OWASP ha accumulato negli anni, che è ampiamente riconosciuta come una fonte autorevole di informazioni e risorse sulla sicurezza delle applicazioni web/mobile,

ID	SECURITY REMEDIATION	ID TEST GUIDE*
SR_1	Verify that the application returns consistent generic error messages in response to invalid account name, password or other user credentials entered during the log in process.	WSTG-IDNT-04 WSTG-IDNT-05
SR_2	Verify that default system accounts and test accounts are deleted prior to releasing the system into production (or exposing it to an untrusted network).	WSTG-IDNT-04
SR_3	Verify that web server information in headers are not disclosed	WSTG-INFO-02
SR_4	Verify that a hardened reverse proxy server is being used to create an additional layer of security between the web server and the Internet.	WSTG-INFO-02
SR_5	Verify that web servers are kept up-to-date with the latest software and security patches	WSTG-INFO-02
SR_6	Verify that metadata files (e.g. robots.txt) do not include hidden or obfuscated paths and functionality.	WSTG-INFO-03
SR_7	Verify that cookie names, file/directory paths and known headers do not disclose the web application framework in use.	WSTG-INFO-08
SR_8	Verify to carefully consider the sensitivity of design and configuration information before it is posted online and to periodically review it.	WSTG-INFO-01
SR_9	Verify that files with specific extensions (e.g. .inc, .config, .zip, .bak, .old) are thoroughly examined to confirm their intended service, absence of sensitive information, and to ensure they are not unnecessary leftovers.	WSTG-CONF-03
SR_10	Verify that configuration policies prevent obsolete files, ensure applications don't rely on web-served directories for sensitive files.	WSTG-CONF-04

Figura 5.8: Security Remediation Application-Side

con una vasta esperienza nel rilevare e indirizzare le vulnerabilità più comuni e gravi. I documenti presi come riferimento sono stati: *OWASP Application Security Verification Standard 4.0.3* insieme al *Web Security Testing Guide* e *OWASP Mobile Application Security Verification Standard 2.0.0* insieme al *Mobile Application Security Testing Guide*.

Esempi di minacce considerate sono: *User Enumeration*, *Information Disclosure*, *Harcoded Sensitive Information*, *Exploit insecure logs*, *Access Token Manipulation*, *HTTP flood attacks*, *Abuse of logic flows and unsafe functionality by design*.

In figura 5.8, viene riportato un esempio di come sono organizzate le *remediation* legate al mondo applicativo. Attraverso l'ID c'è un mapping diretto con il foglio di lavoro principale. E' stato inoltre aggiunto un riferimento diretto con la *testing guide* offerta da OWASP per mappare la remediation anche con il test che è necessario effettuare in una fase reattiva per comprendere se quel rimedio è stato implementato in modo corretto o no.

5.2.2 Cloud-Side

Un ambiente cloud è un'infrastruttura informatica che consente agli utenti di accedere e utilizzare risorse informatiche, come server, archiviazione dati, database e software, tramite Internet. In un ambiente cloud, le risorse sono fornite e gestite da un provider di servizi cloud e possono essere scalate in base alle esigenze dell'utente, offrendo flessibilità e agilità nell'utilizzo delle risorse informatiche. Il cloud computing viene solitamente classificato in tre categorie: SaaS, PaaS e IaaS.

L'ambiente SaaS, come già descritto in precedenza, segue le stesse regole di un applicativo. Quindi il focus è sulle altre due categorie:

- PaaS (*Platform-as-a-Service*): è un modello di distribuzione cloud che fornisce un ambiente completo di sviluppo e distribuzione per gli sviluppatori di software. Con PaaS, gli sviluppatori possono accedere a risorse di calcolo, archiviazione, database, middleware e altri servizi di sviluppo attraverso Internet, senza dover installare, configurare o mantenere l'hardware e il software necessari per eseguire le loro applicazioni. La piattaforma cloud si occupa di tutti questi compiti, consentendo agli sviluppatori di concentrarsi esclusivamente sullo sviluppo delle loro applicazioni.
- IaaS (*Infrastructure-as-a-Service*): è un modello di distribuzione cloud che fornisce risorse informatiche virtualizzate tramite Internet. In sostanza, IaaS offre agli utenti accesso a risorse di calcolo, archiviazione, rete e altri componenti infrastrutturali, consentendo loro di creare e gestire infrastrutture IT senza dover possedere o gestire l'hardware sottostante. Con IaaS, gli utenti possono noleggiare risorse informatiche da un provider di servizi cloud e utilizzarle su richiesta, pagando solo per le risorse effettivamente utilizzate. Questo modello offre flessibilità e scalabilità, poiché le risorse possono essere facilmente aggiunte o rimosse in base alle esigenze del carico di lavoro.

Alcuni dei principali fornitori di servizi cloud al giorno d'oggi includono *Amazon Web Services* (AWS), *Microsoft Azure*, *Google Cloud Platform* (GCP). È importante condurre un Threat Model in un ambiente cloud perché le applicazioni e i dati ospitati in cloud sono soggetti a molteplici rischi di sicurezza, tra cui accesso non autorizzato, perdita di dati, interruzioni del servizio e vulnerabilità dei software. Un TM aiuta ad identificare e valutare queste potenziali minacce, consentendo agli utenti di implementare misure di sicurezza adeguate per proteggere i propri dati e le proprie risorse nell'ambiente cloud. Questo è particolarmente importante data la natura condivisa e multi-tenant[1] dei servizi cloud, dove più utenti condividono la stessa infrastruttura fisica e virtuale.

Microsoft Azure

Tra i vari fornitori si è analizzato Microsoft Azure, in quanto fornitore principale presso il cliente. Azure è stata lanciata da Microsoft nel 2010 come una piattaforma

[1]Un'architutturа multi-tenant consente a più clienti di utilizzare la stessa istanza di un'applicazione o di un servizio, mantenendo al contempo una separazione logica e fisica dei dati e delle risorse tra di loro.

cloud computing completa e flessibile. Tuttavia, la storia di Azure risale ai primi anni 2000, quando Microsoft ha iniziato ad investire in tecnologie e infrastrutture cloud per supportare le proprie offerte di servizi online e applicazioni aziendali.

Azure include molti servizi nella sua piattaforma di *cloud computing* [13]:

- **Compute services**: Azure Virtual Machines (Linux e Windows), Cloud Services, App Services (Web Apps, Mobile Apps, Logic Apps, API Apps e Funtion Apps), Batch, RemoteApp, Service Fabric e Azure Container Service.
- **Data services**: Microsoft Azure Storage (compreso Blob Queue, Table e Azure Files services), Azure SQL Database, DocumetDB, StorSimple e Redis Cache.
- **Application services**: include servizi che puoi usare per aiutarti a costruire le applicazioni come Azure Active Directory, Service Bus per connettere sistemi distribuiti, HDInsight per processare grandi dati, Azure Scheduler e Azure Media services.
- **Network services**: include features come Virtual Networks, Express Route, Azure DNS, Azure Traffic Manager e Azure Content Delivery Network.

Quando si esegue la migrazione di un'applicazione, è utile conoscere i diversi servizi disponibili in Azure, perché è possibile utilizzarli per semplificare la migrazione dell'applicazione e migliorarne la robustezza.

Catalogo e Security Remediation

Per la costruzione del catalogo, per prima cosa sono stati elencati tutti i servizi, maggiormente usati dal cliente, offerti da Azure e poi per ogni servizio attraverso la documentazione fornita da Microsoft (`https://azure.microsoft.com/it-it/`) ho analizzato i rischi e definito le *remediation*. Riporto una serie di esempi di minacce per alcuni servizi:

- Azure Storage:
  - Data from Cloud Storage Object
  - Use Alternate Authentication Material
  - Credentials from Password Stores
  - Network sniffing
- Azure SQL Database:

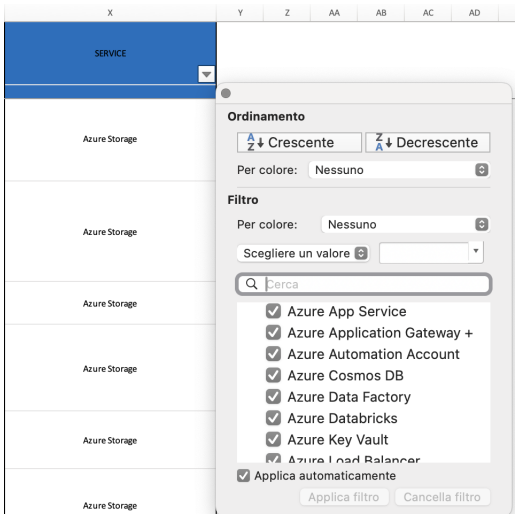


Figura 5.9: Filtro *Service* Azure

  - Network Sniffing
  - Automated Collection

- Azure Key Vault:
  - Valid Accounts
  - Credentials from Password Stores
  - Automated Collection

Si può notare che le minacce possono ripetersi, come ad esempio *Network Sniffing*. Per avere un'analisi mirata e non dispersiva, si è deciso di suddividere le minacce in base alla tipologia di servizio/risorsa cloud. Si filtra quindi per blueprint *Cloud* e per il filtro *service* (Figura 5.9). In Figura 5.10 sono riportati degli esempi di alcune *remediation* definite.

5.3 Analisi dell'impatto

Quando si parla di sicurezza informatica, non vengono considerate solo minacce astratte o teoriche ma si parla di situazioni concrete che potrebbero verificarsi nel mondo reale e avere un impatto tangibile sui dati e sulla continuità operativa. Immaginiamo di dover valutare il rischio di un potenziale attacco informatico. Non possiamo semplicemente affermare la presenza di una minaccia. È necessario capire quanto è probabile che quell'attacco si verifichi e quanto dannoso potrebbe essere se dovesse accadere.

ID	SECURITY REMEDIATION
SC_CL01	You shall always create only Resource Manager Storage accounts in order to strictly control access through the use of RBAC. If you have deployed a storage account with the classic model, you shall migrate to Azure Resource Manager whenever possible
SC_CL02	You should consider to lock all of your storage accounts with an Azure Resource Manager lock to prevent accidental or malicious deletion or configuration changes
SC_CL03	You shall configure storage accounts to deny access to traffic from all networks whenever you are able to restrict the access to selected networks only
SC_CL04	You shall only permit HTTPS protocol for REST APIs calls in order to ensure that Azure Storage data is encrypted between the client and Azure Storage
SC_CL05	You must leverage ACL assignments in order to granularly grant permission to specific files or directories to security principals, adhering to the least privilege principle
SC_CL06	You shall ensure encryption of data at-rest and consider leveraging infrastructure encryption whenever dealing with confidential or highly confidential data

Figura 5.10: Security Remediation Cloud-Side

5.3.1 Probability e Severity

La *probabilità* indica quanto è probabile che un determinato evento si verifichi. La *severity* indica l'entità del danno che la minaccia potrebbe causare se accadesse.

Il rischio è dato dalla *severity* moltiplicata per la *probability*. Questi dati ci aiutano a comprendere quali minacce sono le più urgenti e ci guidano nel decidere come affrontarle.

Esistono vari modi per calcolare questi due valori, a seconda se si esegue un'analisi empirica, una valutazione quantitativa o qualitativa.

E' stata utilizzata la calcolatrice offerta dal CVSS (trattata nel Capitolo 2). Il *National Vulnerability Database* (NVD) [14] genera un punteggio di base per ogni vulnerabilità e poi assegna una classifica in base al punteggio. In figura 5.11 sono riportati i valori che la severity può avere e la probability associata ad ogni valore sarà un numero che rispetta la seguente classifica:

1. *LOW*: 0.0 a 3.9
2. *MEDIUM*: 4.0 a 6.9
3. *HIGH*: 7.0 a 8.9
4. *CRITICAL*: 9.0 a 10.0

La *severity* aiuta a definire una priorità nelle remediation dei *findings* emersi dalle attività post-assessment.

Severity Value	Description
Critical (C)	*Loss of [Confidentiality | Integrity | Availability] is likely to have a catastrophic adverse effect on the organization or individuals associated with the organization (e.g., employees, customers).*
High (H)	*Loss of [Confidentiality | Integrity | Availability] is likely to have a major adverse effect on the organization or individuals associated with the organization (e.g., employees, customers).*
Medium (M)	*Loss of [Confidentiality | Integrity | Availability] is likely to have a serious adverse effect on the organization or individuals associated with the organization (e.g., employees, customers).*
Low (L)	*Loss of [Confidentiality | Integrity | Availability] is likely to have only a limited adverse effect on the organization or individuals associated with the organization (e.g., employees, customers).*

Figura 5.11: Severity CVSS

5.3.2 Exposure e Impact

Per qualsiasi nuova iniziativa, ancor prima di svolgere il TM e quindi calcolare *probability* e *severity*, viene eseguita una fase di *scoping*, con l'obiettivo di approfondire la tipologia di progetto, i dati trattati ed l'esposizione, così da ottenere una classificazione attraverso un *criticality profile*.

Durante questa fase iniziale vengono quindi poste delle domande specifiche che serviranno poi a definire due principali dati:

- *exposure*: si riferisce alla vulnerabilità o all'esposizione di un sistema, di dati sensibili o di risorse a potenziali minacce o attacchi. Indica la situazione in cui una particolare componente o aspetto del sistema informatico è accessibile o suscettibile di essere sfruttato da parte di terzi non autorizzati;
- *impact*: si riferisce agli effetti o alle conseguenze che derivano da un evento di sicurezza o da un incidente. Indica il grado di danno o di perdita che può verificarsi a seguito di un attacco informatico, di una violazione della sicurezza o di un'altra minaccia. L'impatto può riguardare diversi aspetti, come la perdita di dati sensibili, il danneggiamento dei sistemi, la violazione della privacy degli utenti, il fermo delle attività aziendali o finanziarie, e altro ancora.

Le successive tabelle mostrano in che modo si è deciso di calcolare questi due parametri.

Nel dettaglio, *exposure* segue le informazioni riportate in Tabella 5.1, dove si valuta che tipo di *device* verrà utilizzato, chi potrà utlizzare l'applicazione e come e dove la si potrà utilizzare. In base alle risposte viene assegnato un numero per ogni sezione e lo *score* finale deriva dalla somma dei singoli *score*. In base allo *score* finale si ottiene anche il *rating* che segue i criteria indicati nella tabella.

	Risk Rating	**Response**	**Score**
DEVICE (B1)	CRITICAL	Custom-made device	5
	HIGH	Client/Server and mobile application	3
	MODERATE	Client/Server or mobile application	2
	LOW	Embedded Systems	1
WHO (B2)	CRITICAL	External - Public	5
	HIGH	External - Customers	3
	MODERATE	External - Partners, Vendors and Suppliers	2
	LOW	Internal - Employees and Consultants, A2A	1
HOW (B3)	CRITICAL	External - via Web and Mobile Application	5
	HIGH	External - via Web or Mobile Application	3
	MODERATE	External - via VPN and/or restricted and/or whitelisted channels	2
	LOW	Internal - Network Only	1
WHERE (B4)	CRITICAL	External - Customer	5
	HIGH	External - Vendor/Supplier, Partner, Service Provider (Iaas, Paas)	3
	MODERATE	External - Vendor/Supplier, Partner, Service Provider (Saas)	2
	LOW	Internal - Network	1
SCORE (B1 + B2 + B3 + B4)			5
RATING (HIGH, MODERATE or LOW)			Low
CRITERIA			
CRITICAL		15-20	5
HIGH		10-14	
MODERATE		6-9	
LOW		4-5	

Tabella 5.1: Valutazione score Exposure

	Risk Rating	**Criteria**	**Score**
CON (A3)	CRITICAL	Secret	5
	HIGH	Confidential	3
	MODERATE	Internal Use Only	2
	LOW	Public Information	1
INT (A4)	HIGH	High	3
	LOW	Normal	1
AVA (A5)	HIGH	Very Critical	3
	MODERATE	Critical	2
	LOW	Important	1
SCORE (A3 + A4 + A5)			8
RATING (HIGH, MODERATE or LOW)			High
CRITERIA			
CRITICAL		≥ 10	8
HIGH		8-9	
MODERATE		5-7	
LOW		3-4	

Tabella 5.2: Valutazione score Impact

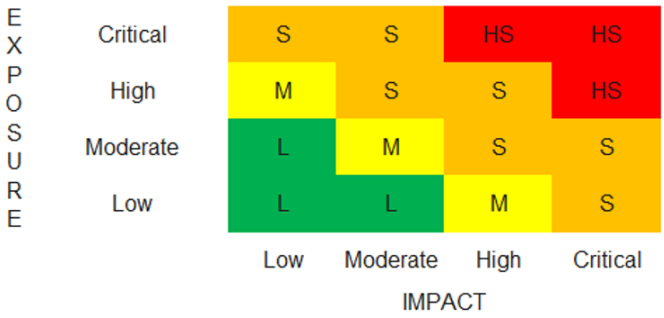


Figura 5.12: Criticality profile

Nel dettaglio, *impact* segue le informazioni riportate in Tabella 5.2, dove viene valutato l'impatto attraverso i valori della CIA. Anche qui lo *score* finale è la somma dei singoli *score* e attraverso i criteria indicati si avrà il *rating*.

L'importanza di questi due valori è data dal fatto che il loro insieme, come mostrato in figura 5.12, fornisce un punto di partenza nella definizione di *Criticality Profile* ovvero una valutazione o a una descrizione della criticità di un determinato elemento o sistema all'interno di un contesto specifico.

La creazione di un profilo di criticità aiuta ad identificare e gestire in modo

appropriato le risorse e le attività di sicurezza, consentendo un'allocazione efficace delle risorse e una priorizzazione delle azioni per ridurre i rischi e proteggere gli asset criticamente importanti. Quindi è stato un punto di partenza importante per poi eseguire tutte le fasi del TM.

5.4 Proof Of Concept

La sezione *Proof of Concept* (PoC) di questo documento mira a dimostrare la fattibilità e l'efficacia del framework implementato per condurre il processo di Threat Modeling. L'obiettivo è valutare la validità e l'utilità del framework nell'individuare e gestire le minacce alla sicurezza in vari contesti, tra cui applicazioni web e mobile e ambienti Azure. Si cerca quindi di rispondere alle seguenti domande:

- È possibile integrare con successo il framework personalizzato nel processo aziendale esistente?
- Il framework è in grado di identificare con precisione le minacce alla sicurezza che possono influenzare la sicurezza delle iniziative?
- Quali sfide tecniche o di implementazione potrebbero emergere durante l'integrazione del framework nell'ambiente di sviluppo attuale?

Durante i diversi PoC, il framework è stato testato utilizzando dati di esempio e scenari di minaccia realistici, conservando in ogni passo la privacy del cliente per il quale ho condotto questo lavoro di tesi. È stata monitorata l'efficacia del framework nell'individuare e gestire le minacce. I risultati di questi PoC sono un primo passo per guidare lo sviluppo e l'implementazione successiva del framework personalizzato, garantendo che risponda pienamente alle esigenze e alle sfide specifiche dell'organizzazione.

La successiva sotto sezione descrive, attraverso le fasi del TM espresse nel Capitolo 2, gli step ad alto livello del flusso di lavoro eseguito per condurre l'analisi delle minacce per una applicazione Azure.

5.4.1 PoC Applicazione Azure

- FASE 1: Sono stati effettuati degli approfondimenti con il PM di progetto per condividere le informazioni principali. E' stato quindi definito:
  - il servizio in questione che si occupa di instradare le email, senza salvare alcun dato in locale ad eccezione dei log di routing;
  - le informazioni processate che sono *Email transactional logs*;

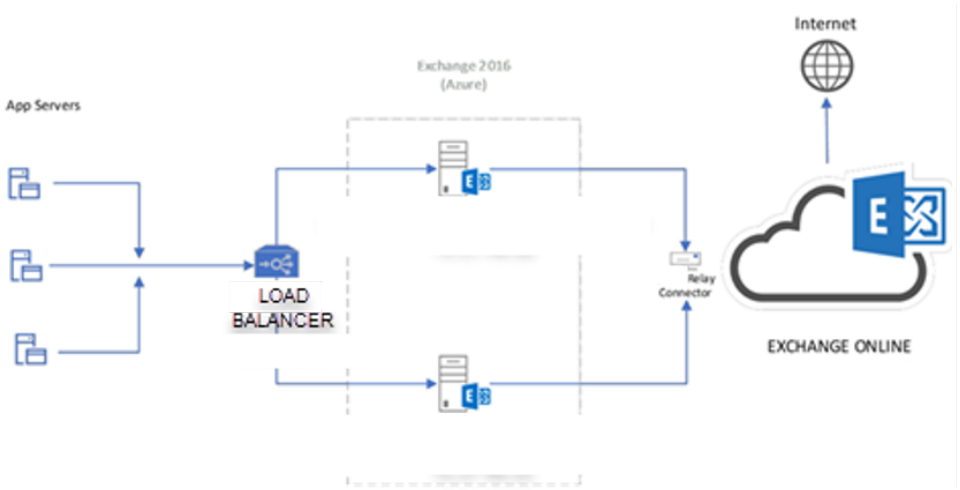


Figura 5.13: Diagramma Iniziativa Azure

- i dispositivi che possono essere usati per accedere alle informazioni: Windows, MAC, Linux e dispositivi interni dell'azienda;
- chi può accedere alle informazioni: i dipendenti;
- la CIA: la confidenzialità richiesta è normale così come l'integrità ma si richiede un'alta disponibilità (ogni 15 minuti);
- in che modo può avvenire l'accesso: sia dall'interno dell'organizzazione che tramite VPN;
- Il tipo di piattaforma richiesto per sviluppare il prodotto finale: Azure.

A questo punto, grazie a tutte queste informazioni si definiscono i valori di partenza per determinare il *Criticality Profile*:

- Exposure: score 6 -> rating Moderate;
- Impact: score 7 -> rating Moderate;
- Criticality Profile: Moderate.

- FASE 2: Proseguendo è stato definito il disegno del diagramma progettuale, per comprendere nello specifico i servizi di Azure in uso nel progetto.

  I servizi Azure presenti sono: 2 Azure Virtual Machine e 1 Load Balancer.

- FASE 3: Sono stati analizzati i servizi per identificare vulnerabilità e rischi. Si prosegue a questo punto sul framework personalizzato che permette di eseguire il TM attraverso lo schema di filtraggio. Infatti si filtra prima per blueprint

e poi per servizi in modo da ottenere una lista di minacce *pre-assessment* (Figura 5.14).

- FASE 4: In questa fase sono state analizzate le minacce e per ognuna di essa si ha un set di *remediation* ben definite che devono essere implementate. In applicazioni Azure si tratta di configurazioni che devono essere abilitate o disabilitate. Riporto l'esempio di alcune:
  - **Network Service Scanning** per Azure Virtual Machine:
    - SC_CL39: Ogni macchina virtuale deve essere creata all'interno di una sottorete privata e non deve avere indirizzi IP pubblici.
    - SC_CL40: Le macchine virtuali a cui si deve accedere da Internet devono essere create sotto Azure Load Balancers e Azure Application Gateway.
    - SC_CL43: Configurare le regole del gruppo di sicurezza per consentire solo agli host o alle reti fidate di accedere alle porte dell'istanza.
  - **Network Service Scanning** per Load Balancer:
    - SC_CL52: Ogni volta che le risorse non devono essere esposte direttamente ad un endpoint Internet, è necessario un load balancer interno.
- FASE 5: Nell'ultima fase si verifica se le *security remediation* individuate vanno a diminurire le minacce identificate pre-assessment. Quindi viene ripetuto il Threat Model, dopo la fase di configuration review, per controllare se si è a questo punto coperti da tutti i rischi inizialmente identificati e nel caso fare il check delle minacce ancora presenti post-assessment:

  Come si nota chiaramente in Figura 5.15, le minacce sono diminuite rispetto a quelle identificate pre-assessment. Ciò vuol dire che il TM è stato utile per identificare i rischi e diminuirli già prima della fase di design. A questo punto il team tecnico ha definito quali minacce sono *exploitable* e sono stati calcolati i valori di probability e severity per prioritizzare i rischi. Sono poi state comunicate le minacce al PM con le rispettive remediation ancora non applicate, tramite dei report.

THREAT NAME / TECHNIQUE	DESCRIPTION	SECURITY REMEDIATION	REFERENCES	SERVICE
Network Service Scanning	Adversaries may attempt to get a listing of services running on remote hosts and local network infrastructure devices, including those that may be vulnerable to remote software exploitation.	SC_CL39 SC_CL40 SC_CL43	https://learn.microsoft.com/en-us/azure/virtual-machines/	Azure Virtual Machine
Network Sniffing	Adversaries may sniff network traffic to capture information about an environment, including authentication material passed over the network.	SC_CL38 SC_CL41	https://learn.microsoft.com/en-us/azure/virtual-machines/	Azure Virtual Machine
Valid Accounts	Adversaries may obtain and abuse credentials of existing accounts as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion	SC_CL41	https://learn.microsoft.com/en-us/azure/virtual-machines/	Azure Virtual Machine
Account Discovery	Adversaries may attempt to get a listing of valid accounts, usernames, or email addresses on a system or within a compromised	SC_CL42	https://learn.microsoft.com/en-us/azure/virtual-machines/	Azure Virtual Machine
Non-Standard Port	Adversaries may communicate using a protocol and port pairing that are typically not associated	SC_CL44 SC_CL45	https://learn.microsoft.com/en-us/azure/virtual-machines/	Azure Virtual Machine
Ingress Tool Transfer	Adversaries may transfer tools or other files from an external system into a compromised environment	SC_CL46	https://learn.microsoft.com/en-us/azure/virtual-machines/	Azure Virtual Machine
Exploitation for Privilege Escalation	Adversaries may exploit software vulnerabilities in an attempt to elevate privileges	SC_CL47	https://learn.microsoft.com/en-us/azure/virtual-machines/	Azure Virtual Machine
Taint Shared Content	Adversaries may deliver payloads to remote systems by adding content to shared storage locations, such as network drives or internal code repositories. Content stored on network drives or in other shared locations may be tainted by adding malicious programs, scripts, or exploit code to otherwise valid files	SC_CL48	https://learn.microsoft.com/en-us/azure/virtual-machines/	Azure Virtual Machine
Automated Collection	Once established within a system or network, an adversary may use automated techniques for collecting internal data	SC_CL49 SC_CL51	https://learn.microsoft.com/en-us/azure/virtual-machines/	Azure Virtual Machine
Data Destruction	Adversaries may destroy data and files on specific systems or in large numbers on a network to interrupt availability to systems, services, and network resources	SC_CL49	https://learn.microsoft.com/en-us/azure/virtual-machines/	Azure Virtual Machine
Unsecured Credentials	Adversaries may search compromised systems to find and obtain insecurely stored credentials.	SC_CL50	https://learn.microsoft.com/en-us/azure/virtual-machines/	Azure Virtual Machine
Cloud Infrastructure Discovery	An adversary may attempt to discover infrastructure and resources that are available within an infrastructure-as-a-service (IaaS)	SC_CL50	https://learn.microsoft.com/en-us/azure/virtual-machines/	Azure Virtual Machine
Network Service Scanning	Adversaries may attempt to get a listing of services running on remote hosts and local network infrastructure devices, including those that may be vulnerable to remote software exploitation.	SC_CL52	https://learn.microsoft.com/en-us/azure/load-balancer/load-balancer-overview	Azure Load Balancer

Figura 5.14: Es. Minacce Azure Pre-Assessment

R	S	T	V	W	X	Y	Z	AA	AC
THREAT NAME / TECHNIQUE	DESCRIPTION	REFERENCES	SERVICE	SECURITY REMEDIATION	POST-ASSESSMENT APPLICABLE	POST-ASSESSMENT EXPLOITABLE	PROBABILITY	SEVERITY	
Network Service Scanning	Adversaries may attempt to get a listing of services running on remote hosts and local network infrastructure devices, including those that may be vulnerable to remote software exploitation.	https://learn.microsoft.com/en-us/azure/virtual-machines/	Azure Virtual Machine	Configure security group rules to only allow trusted hosts or networks to access ports on your instance	x	x	0,45	HIGH	
Network Sniffing	Adversaries may sniff network traffic to capture information about an environment, including authentication material passed over the network.	https://learn.microsoft.com/en-us/azure/virtual-machines/	Azure Virtual Machine	You shall leverage RBAC in order to granularly control access to your Virtual Machines resources	x	x	0,5	HIGH	
Valid Accounts	Adversaries may obtain and abuse credentials of existing accounts as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion	https://learn.microsoft.com/en-us/azure/virtual-machines/	Azure Virtual Machine	You shall leverage RBAC in order to granularly control access to your Virtual Machines resources	x	x	0,5	HIGH	
Taint Shared Content	Adversaries may deliver payloads to remote systems by adding content to shared storage locations, such as network drives or internal code repositories. Content stored on network drives or in other shared locations may be tainted by adding malicious programs, scripts, or exploit code to otherwise valid files	https://learn.microsoft.com/en-us/azure/virtual-machines/	Azure Virtual Machine		x				
Automated Collection	Once established within a system or network, an adversary may use automated techniques for collecting internal data	https://learn.microsoft.com/en-us/azure/virtual-machines/	Azure Virtual Machine		x				
Data Destruction	Adversaries may destroy data and files on specific systems or in large numbers on a network to interrupt availability to systems, services, and network resources	https://learn.microsoft.com/en-us/azure/virtual-machines/	Azure Virtual Machine		x				

Figura 5.15: Es. Minacce Azure Post-Assessment

Capitolo 6

Tool commerciale

Nell'ambito dello stato dell'arte, diversi strumenti commerciali facilitano l'attività di Threat Modeling. Tra questi, due in particolare hanno attirato l'attenzione: *ThreatModeler* e *IriusRisk*. Per questo elaborato, ci si è concentrati sull'analisi di quest'ultimo, sebbene sia emersa anche la validità del primo. Dopo un periodo di test della versione gratuita *Community*, sono state programmate demo ufficiali con i gestori del tool, permettendo così di esaminare anche la versione a pagamento *Enterprise*.

In questo capitolo, verranno analizzate da vicino le caratteristiche chiave di IriusRisk, inclusa la sua capacità di integrarsi con processi di sviluppo del software esistenti, la sua flessibilità nell'adattarsi alle esigenze specifiche dell'organizzazione e la sua capacità di fornire analisi dettagliate dei rischi in tempo reale.

6.1 Introduzione ad IriusRisk

Nel 2015 [15], Stephen de Vries e Cristina Bentué hanno fondato IriusRisk con l'obiettivo di sviluppare applicazioni sicure e di facile utilizzo per team con limitate risorse temporali. Durante la loro esperienza nel campo della sicurezza informatica, hanno riscontrato le sfide significative che affrontano i team all'inizio del processo di sviluppo software. Con pochi esperti di sicurezza disponibili rispetto al numero di sviluppatori, la produzione di codice diventava un ostacolo, rallentando i processi e aumentandone i costi. Stephen e Cristina hanno riconosciuto l'opportunità di automatizzare le fasi iniziali dello sviluppo e si sono dedicati a questa missione per un anno e mezzo. Il loro impegno nel fornire un prodotto di alta qualità ha portato IriusRisk da una startup a una società consolidata, con una crescita annua del 100%. Da un team iniziale di due fondatori, sono diventati un'azienda globale con oltre 200 dipendenti distribuiti in Europa, Nord America e Nuova Zelanda,

e continuano a espandersi. Grazie alla loro dedizione e passione per la sicurezza, IriusRisk ha trasformato la sua visione in realtà, fornendo soluzioni di modellazione delle minacce a oltre 100 clienti in tutto il mondo.

In alcune industrie, come quelle finanziarie o mediche, le normative spingono le organizzazioni ad implementare misure di sicurezza proattiva, come la modellazione delle minacce. Tuttavia, sia che si tratti di veicoli elettrici che di dispositivi medici, si ritiene che la prevenzione degli attacchi informatici sia cruciale per tutte le sfere della società. Questa convinzione ha, quindi, ispirato la creazione di IriusRisk, con l'obiettivo di promuovere una cultura di sicurezza proattiva per tutti i prodotti fin dalle prime fasi di progettazione, prima che il codice sia anche solo scritto. Infatti, come viene riportato sul loro sito web: *"La nostra visione è quella di trasformare il security by design in un movimento e di rendere accessibile a tutti la modellazione delle minacce. Ecco perché investiamo anche in una versione gratuita del nostro prodotto, la Community Edition. La mancanza di budget non dovrebbe impedire a nessuno di creare un'applicazione solida."*

6.2 Funzionalità e caratteristiche di IriusRisk

6.2.1 Security Content Libraries

Affrontare la vasta gamma di metodologie disponibili per la sicurezza informatica può essere una sfida, poiché ognuna offre prospettive e approcci diversi. La scelta della metodologia giusta dipende da una serie di fattori, tra cui le dimensioni e la complessità dell'ambiente, il tipo di dati gestiti e le risorse disponibili. Tuttavia, può essere difficile capire quale metodologia seguire in uno specifico contesto, data la varietà di proposte e le implicazioni temporali di ognuna. In tal senso, può essere vantaggioso considerare un approccio ibrido, combinando elementi di diverse metodologie per adattarsi meglio alle esigenze dell'organizzazione e massimizzare l'efficacia delle attività di sicurezza, riducendo al contempo il dispendio di tempo e risorse.

Uno dei vantaggi distintivi di IriusRisk è la sua vasta raccolta di librerie integrate, che comprendono dati sulle minacce, modelli di attacco, contromisure di sicurezza e best practice di settore. Questa ricca biblioteca fornisce agli utenti una risorsa completa e affidabile per guidare il processo di Threat Modeling e la gestione dei rischi.

Attualmente, esistono numerosi mandati e framework progettati per migliorare la sicurezza informatica a livello globale. Ad esempio, l'OMB (*United States Office of Management and Budget*) ha imposto alle agenzie federali di aderire al quadro NIST

SSDF [14] per lo sviluppo del software. Inoltre, la *Food and Drug Administration*[1] (FDA) [16] ha acquisito il potere di rifiutare i dispositivi medici che non soddisfano gli standard di sicurezza informatica.

Di fronte ad un panorama della sicurezza in continua evoluzione, IriusRisk si impegna a fornire le ultime e migliori risorse attraverso le sue librerie di contenuti sulla sicurezza. Queste risorse coprono una vasta gamma di standard, inclusi EU-GDPR, PCI DSS, FedRamp e Mitre ATT&CK. Inoltre, le organizzazioni hanno la flessibilità di aggiungere i propri standard personalizzati, se necessario, per garantire un livello ottimale di sicurezza informatica.

Regulatory & Compliance

- EU-GDPR
- FedRAMP
- HIPAA
- IEC/ANSI 62443
- ISO/ IEC 27002: 2013
- ISO/ SAE 21434
- NIST Cybersecurity Framework
- PCI-DSS v3.2.1
- PCI-DSS v4.0
- PCI Software Security Standard
- UNECE WP.29 Cybersecurity Regulation (CSMS)

Industry Standards

- CWE Top 25
- MITRE ATT&CK Enterprise & ICS
- NIST 800-190

[1]Si tratta di un'agenzia federale degli Stati Uniti d'America responsabile della regolamentazione e del controllo della sicurezza e dell'efficacia degli alimenti, dei farmaci, dei cosmetici, dei dispositivi medici, dei prodotti per l'igiene e altri prodotti correlati alla salute pubblica.

- NIST 800-204
- NIST 800-53
- NIST 800-63
- OWASP API Security Top 10
- OWASP ASVS v4
- OWASP CSVS
- OWASP MASVS
- OWASP Mobile Top Ten 2016
- OWASP Top 10 2021

Industrial Automation

- IEC/ ANSI 62443 3-3 and 4-2
- UNECE WP.29 Cybersecurity Regulation (CSMS)

Operational

- AWS Foundations Benchmark
- AWS Three-Tier Web Architecture Benchmark
- Azure Security Benchmark
- Docker Community Edition Benchmark
- Google Cloud Platform Foundations Benchmark
- Kubernetes Benchmark
- Microsoft Azure Foundations Benchmark
- OWASP Docker Top 10 2018

Internet of Things

- IoT Security Foundation
- Machine Learning and Artificial Intelligence

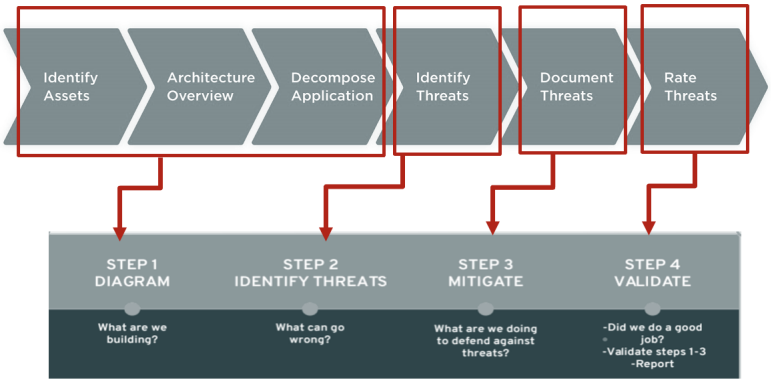


Figura 6.1: Fasi del Threat Model con IriusRisk

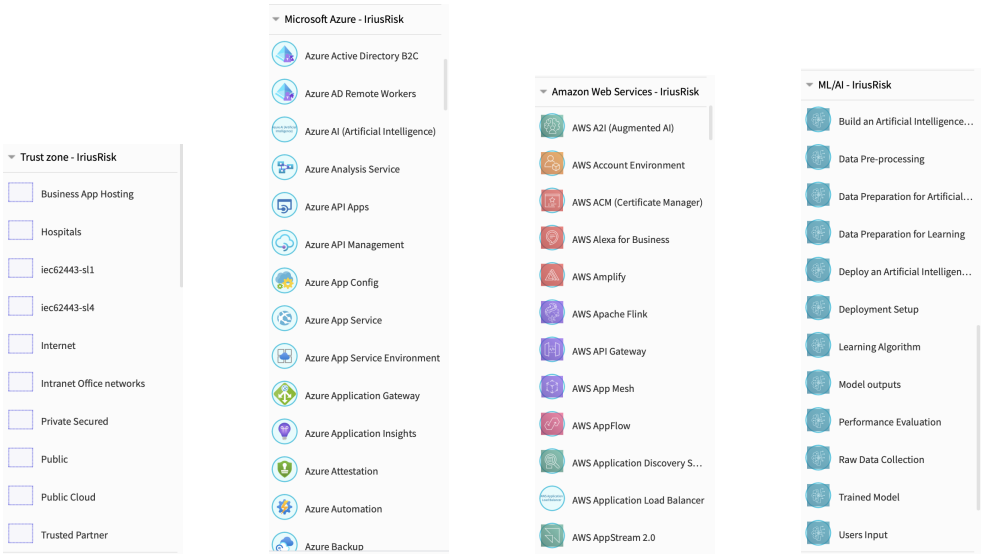
6.2.2 Threat Model con IriusRisk

La Figura 6.1 illustra in modo chiaro le fasi da seguire per la realizzazione di un Threat Model, confrontate con le fasi seguite da IriusRisk. Nella parte superiore dello schema sono elencate le fasi tradizionali di creazione di un Threat Model, che possono includere l'identificazione delle risorse, l'analisi delle minacce, la valutazione dei rischi e la definizione delle contromisure di sicurezza. Nella parte inferiore dello schema, sono invece mostrate le fasi seguite da IriusRisk, che offrono una rappresentazione visiva di come il tool mappa e integra le diverse fasi del processo di TM. Questo confronto evidenzia la corrispondenza e l'integrazione tra le fasi tradizionali di Threat Modeling e le funzionalità offerte dal tool, fornendo una panoramica chiara e dettagliata del processo e delle potenzialità di questo strumento nella gestione delle minacce informatiche.

Fase 1: Diagram

La fase iniziale di IriusRisk rappresenta un punto cruciale nel processo di gestione delle minacce informatiche, in quanto si concentra sul disegno dell'architettura del sistema. Questo passaggio è fondamentale per comprendere appieno il contesto in cui operano le minacce e i rischi, e IriusRisk offre una serie di strumenti e funzionalità per agevolare questa fase.

IriusRisk disponde di componenti predefiniti già mappati con librerie di minacce e contromisure, Figura 6.2. Questo significa che gli utenti possono selezionare e posizionare rapidamente i componenti dell'architettura, sapendo che sono già associati alle minacce e alle contromisure corrispondenti. Questa integrazione consente una maggiore coerenza e uniformità nelle valutazioni dei rischi, garantendo che i componenti siano allineati con le migliori pratiche e le conoscenze aggiornate



(a) Trust Zone **(b)** Azure **(c)** AWS **(d)** AI

Figura 6.2: Esempi di componenti in IriusRisk

nel campo della sicurezza informatica. Tra i componenti offerti dal tool, vi è una sezione dedicata alle *Trust zone* che sono aree di fiducia all'interno di un'applicazione o di un sistema, che suddividono l'architettura in base ai livelli di sicurezza. Ogni zona rappresenta un'area con requisiti specifici di sicurezza e gestione del rischio. Questo approccio consente una valutazione e una gestione mirata delle minacce in ciascuna zona, con l'implementazione di misure di sicurezza adeguate al livello di rischio.

Attraverso i componenti si disegna quindi l'architettura che si desidera progettare/analizzare, esempio in figura 6.3. IriusRisk offre un editor grafico *draw.io*, un'applicazione web open-source che consente agli utenti di creare facilmente diagrammi di architettura del sistema. Questo editor offre un'interfaccia intuitiva ed user-friendly, consentendo agli utenti di rappresentare in modo chiaro e dettagliato tutti gli elementi e i componenti del sistema, tra cui server, applicazioni, database e reti.

Una grande feature è che oltre alle informazioni di default predefinite da IriusRisk, ogni componente IT inserito nel diagramma di architettura può essere personalizzato in base al contesto specifico selezionando gli scenari di minacce applicabili (Figura 6.4). In aggiunta possono essere selezionati anche i dati trattati dallo specifico componente (Figura 6.5). In questo modo si creano casi d'uso specifici che avranno un impatto sulle minacce di quel componente.

Inoltre, per affinare ancora di più lo scenario architetturale, IriusRisk permette di

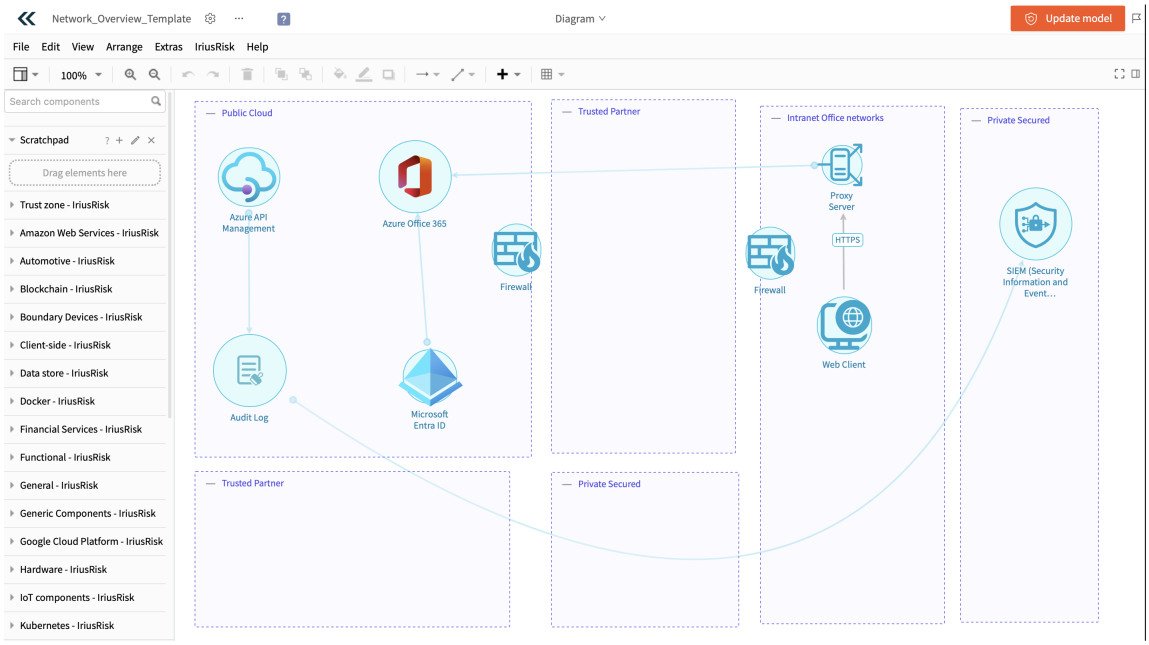
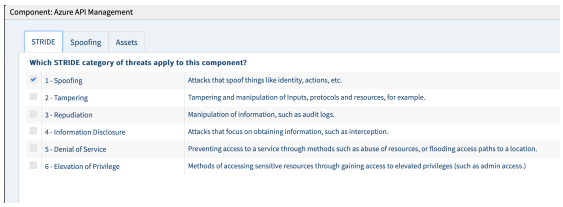


Figura 6.3: Diagram in IriusRisk



(a) STRIDE

(b) Spoofing

Figura 6.4: Questionario componente in IriusRisk

selezionare quali dati *in transit* sono trattati nei flussi definiti, figura 6.6.

Dopo aver analizzato tutto ciò, per quanto ci sia un grande margine di customizzazione, il primo dubbio emerso è stato: *"e se non dovessero esserci tutti i componenti di cui necessito nella mia architettura?"*. IriusRisk risponde in modo consistente a questo quesito, infatti dispone di una sezione *Security Context* nella quale è possibile creare componenti ad-hoc e mantenerli salvati, in modo tale da poterne usufrire anche in tutti i successivi TM. Questo è importante perchè se il tool è usato da un'azienda che ha delle linee guida di default sulle quali poi si basano tutte le successive iniziative, si possono definire una sola volta i confini specifici e riusarli senza dover risettare tutto ogni volta dall'inizio.

Riassumendo, quindi, questa prima fase in steps:

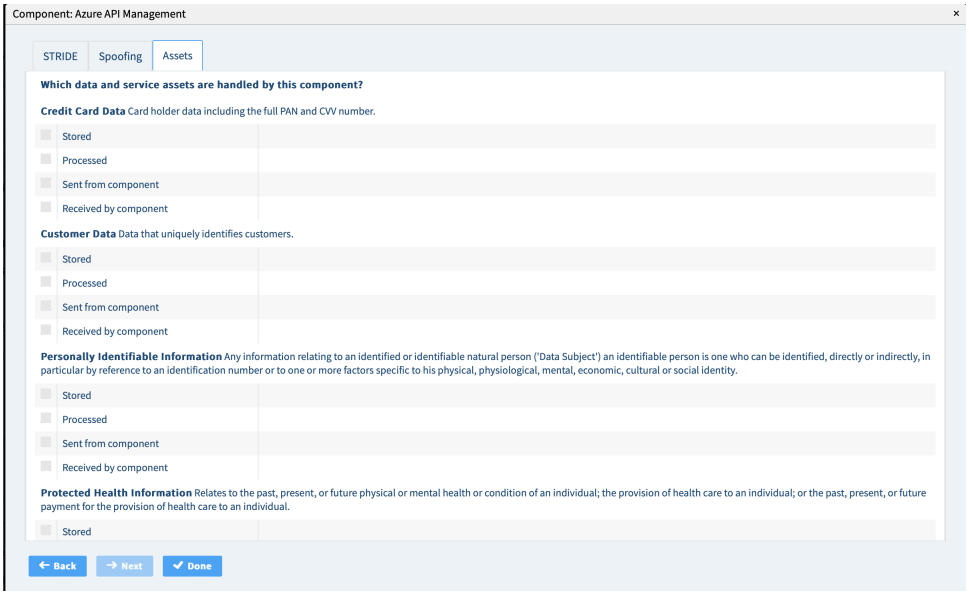


Figura 6.5: Customizzazione dati in un componente

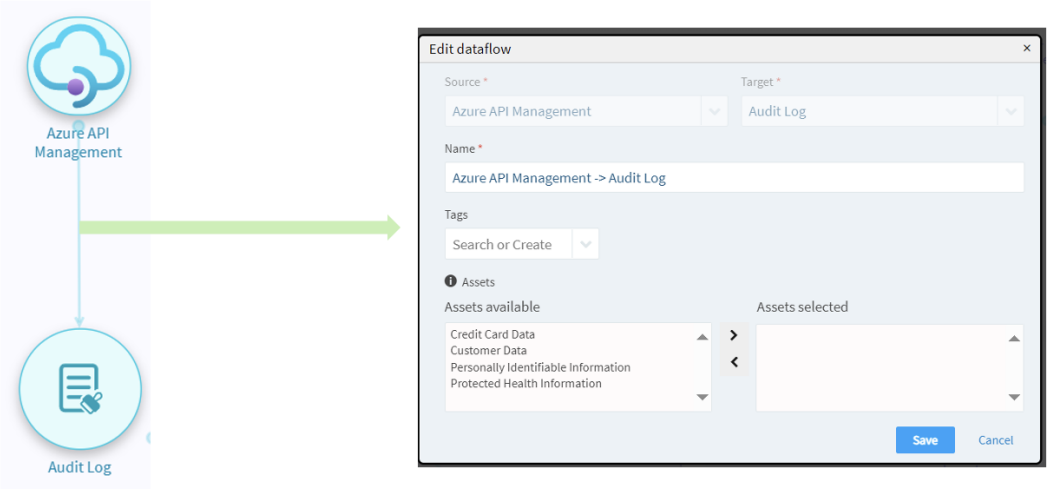


Figura 6.6: Customizzazione dati in transit

- selezione/creazione delle Trust Zone di interesse che sono le zone sulle quali si inseriranno i componenti,
- selezione/creazione dei componenti di interesse,
- personalizzazione dei componenti e del flusso tra due componenti in base alle necessità architetturali,

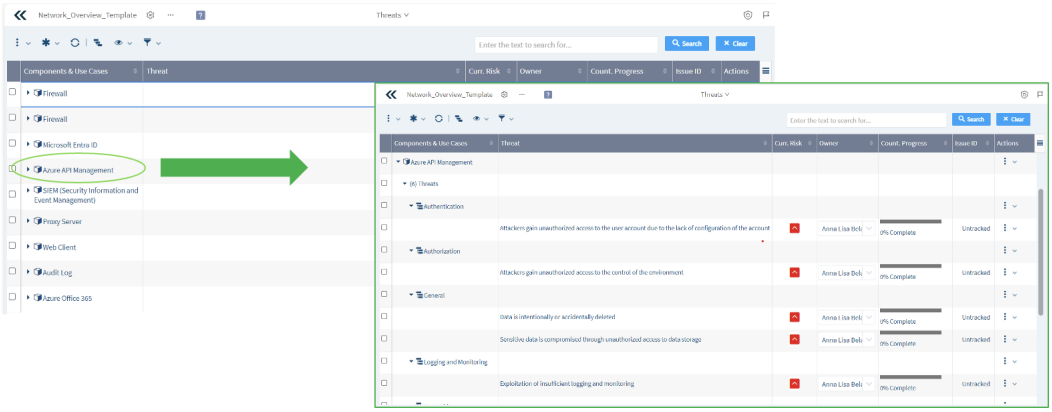


Figura 6.7: Dashboard minaccia in IriusRisk

- *upload* del diagramma, necessario ogni volta che si apporta una modifica all'architettura.

Fase 2: Identify threats

Una volta che l'architettura del sistema è stata disegnata e rappresentata in IriusRisk, il software utilizza le informazioni fornite per identificare e valutare potenziali minacce per ogni componente inserito nel diagramma. Questo processo è reso possibile grazie all'integrazione di IriusRisk con una vasta gamma di librerie di minacce e contromisure, come discusso in precedenza. Questo approccio consente agli utenti di ottenere una panoramica completa e accurata delle minacce che possono influenzare la sicurezza del sistema, consentendo loro di adottare misure preventive e contromisure appropriate per mitigare i rischi identificati.

IriusRisk offre una dashboard intuitiva e completa, Figura 6.7, dove vengono elencate tutte le minacce individuate. La visualizzazione è fornita per impostazione predefinita utilizzando una struttura ad albero tabellare che raggruppa le minacce per i componenti inclusi nel diagramma.

Quindi, il livello più esterno è rappresentato dai componenti, all'interno vi sono gli *use cases* come autorizzazione, autenticazione, logging e monitoring etc.., e all'interno di ogni use case vi sono le minacce. Ogni minaccia è associata a dei dati, come si vede in Figura 6.8:

Questi dati sono definiti di default ma possono essere tutti modificati. Ed in particolare per *impacts* è importante perchè modificando il livello di *confidentiality*, *integrity*, *availability* e *ease of exploitation* si avrà un rischio attuale differente e quindi una priorità di mitigazione maggiore o minore in base ai nuovi valori selezionati. Ci sono tre tipi di rischi che IriusRisk calcola:

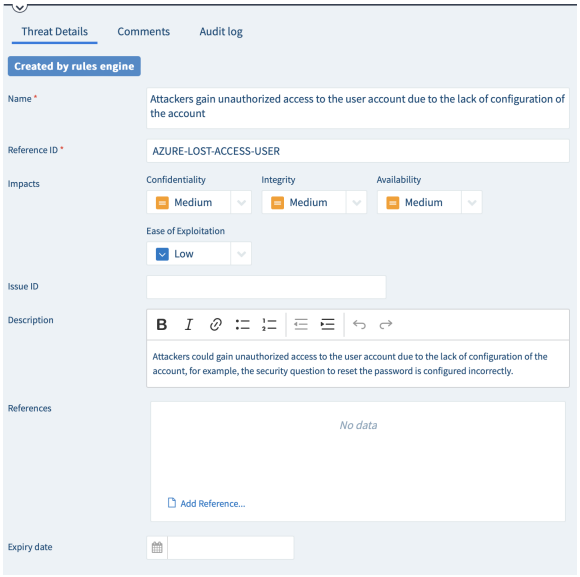


Figura 6.8: Dashboard Minaccia in IriusRisk

- *Rischio intrinseco*: la valutazione del rischio che una minaccia comporta per un componente in base ai seguenti attributi:
  - la classificazione di sicurezza degli Asset ad esso associati,
  - il rating di fiducia della Trust Zone in cui si trova il componente,
  - il rating di impatto della minaccia,
  - la valutazione dell'impatto delle debolezze,
  - il rating di facilità di sfruttamento della minaccia.
- *Rischio attuale* : questa valutazione si basa sul Rischio intrinseco e poi viene aggiustata in base alle contromisure implementate ed ai risultati dei test. Le contromisure implementate ridurranno questo punteggio, in base alla loro percentuale di mitigazione, così come i test superati. Il fallimento dei test annullerà qualsiasi riduzione del rating del rischio; ad esempio, se il rating è stato ridotto da Alto a Medio a causa dell'implementazione di una contromisura, ma il test per tale contromisura fallisce, il rating del rischio tornerà al valore originario di Alto.
- *Rischio previsto*: il rating di rischio futuro che otterremmo se tutte le contromisure pianificate venissero implementate con successo.

È importante sottolineare che così come possono essere aggiunti dei nuovi componenti e nuove trust zone, è possibile aggiungere nuove minacce anche partendo

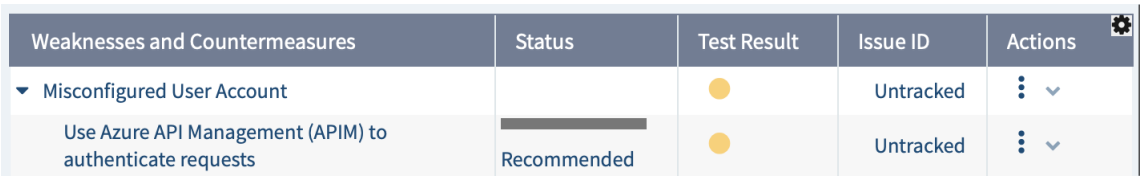


Figura 6.9: Weaknesses e Countermeasures in IriusRisk

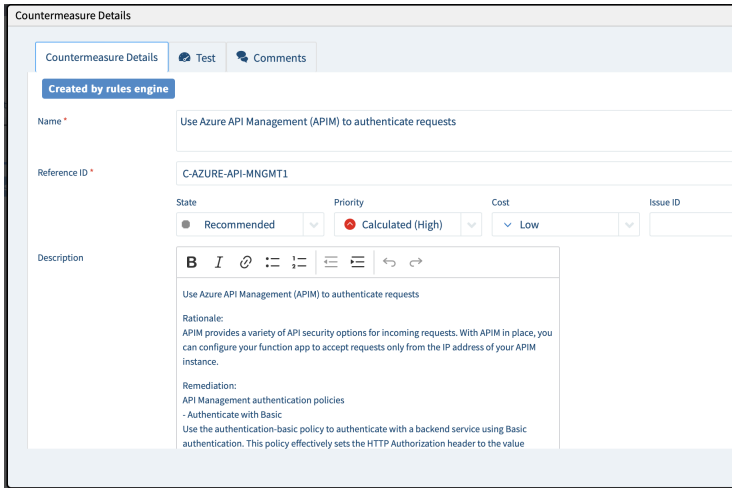


Figura 6.10: Dashboard contromisure in IriusRisk

da quelle esistenti. Aggiungere nuove minacce in questa fase del lavoro non andrebbe ad influenzare altri Threat Model che si stanno performando perché si aggiungerebbero minacce legate allo specifico scenario e varranno solo per esso.

Fase 3: Mitigate

Ogni minaccia è associata a delle *weaknesses* e a delle *countermeasures* e viene indicato lo stato della contromisura in quanto potrebbe essere ad esempio raccomandata oppure obbligatoria e ovviamente questo fa oscillare il rischio del progetto. Oltre alle contromisure che IriusRisk offre di default, è possibile aggiungerne ulteriori e vale lo stesso discorso fatto per le minacce, ovvero che non si va a compromettere altri TM in evoluzione ma si apportano aggiunte legate allo specifico scenario che si sta analizzando. In Figura 6.9 vi è un esempio di visualizzazione delle weaknesses e delle countermeasures.

Entrando all'interno della specifica contromisura è possibile valutare diversi dettagli, anch'essi modificabili in base all'avanzamento della mitigazione:

Vi è, inoltre, una sezione dedicata ai test, Figura 6.11, in quanto è possibile

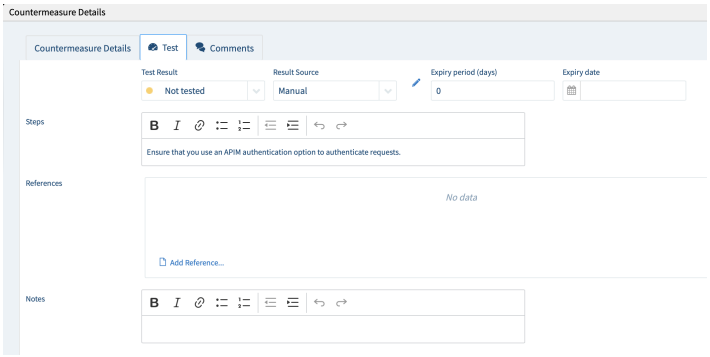


Figura 6.11: Test in IriusRisk

selezionare di aver implementato quella contromisura ma è necessario andare a testare se effettivamente la debolezza è stata mitigata.

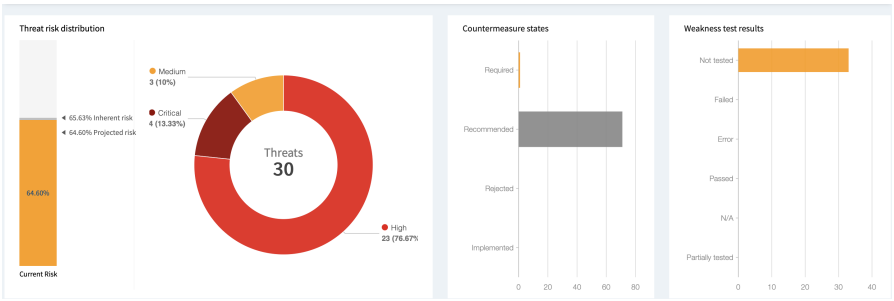
Nella sezione dedicata al *testing* si può tenere traccia di quando il test è fatto, del risultato e degli step effettuati. Un'ulteriore proprietà che IriusRisk offre è quella di poter importare in modo automatico tramite API i risultati di test ottenuti con tool esterni come Fortify SSC, Fortify SCA, ThreadFix, OWASP ZAP e Cucumber. Inoltre, la gestione delle contromisure può essere inviata tramite integrazione al sistema di ticketing e tiene automaticamente traccia dell'ID del ticket (integrazione con Jira, ServiceNow, Microsoft TFS, Azure DevOps, Rally e Redmine). Lo stato del ticket nel sistema di ticketing viene sincronizzato con lo stato della contromisura di IriusRisk. Proprio pensando ad una inclusione di ogni stakeholder durante l'attività di TM, vi è una sezione dedicata ai commenti.

Fase 4: Validate

Durante qualsiasi fase, è possibile tenere traccia dell'avanzamento del processo di Threat Modeling grazie alla *Home Page* che racchiude, attraverso dei grafici, lo stato attuale delle minacce, delle contromisure e dei test, calcolando così un *Risk Score*.

Infine, è offerto un servizio di *reporting*. Sono disponibili quattro report pre-configurati:

- *Current Risk Summary Report*: Identifica e analizza i rischi delle applicazioni, fornendo raccomandazioni per la mitigazione.
- *Technical Threat Report*: fornisce una panoramica dettagliata delle minacce tecniche che potrebbero compromettere la sicurezza delle applicazioni



(a) HomePage

(b) Risk Score

Figura 6.12: HomePage in IriusRisk

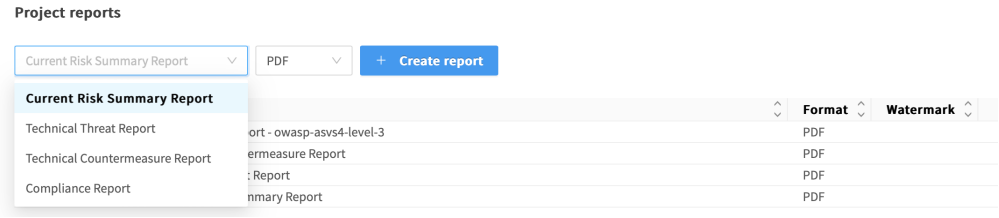


Figura 6.13: Report in IriusRisk

- *Technical Countermeasure Report*: fornisce una guida dettagliata su come mitigare i rischi di sicurezza tecnica all'interno delle applicazioni
- *Compliance Report*: Valuta la conformità delle applicazioni rispetto agli standard di sicurezza.

È possibile creare report in vari formati tra cui PDF, Word, XLS, CSV. Selezionando il formato Word si garantisce la personalizzazione.

Capitolo 7

Tool a confronto

In questo capitolo, esamineremo e confronteremo le caratteristiche, i vantaggi e gli svantaggi di IriusRisk rispetto al framework personalizzato di Threat Modeling, esplorando le considerazioni chiave che le organizzazioni devono valutare nel prendere una decisione informata tra l'adozione di una soluzione commerciale e lo sviluppo di un approccio personalizzato alla gestione delle minacce.

IriusRisk

- **Pro**:
  - *Struttura organizzata*: IriusRisk fornisce una struttura organizzata e metodica per condurre il threat modeling, che può aiutare a garantire un approccio completo e sistematico all'identificazione e alla gestione delle minacce.
  - *Automazione*: Lo strumento offre funzionalità di automazione che possono semplificare e accelerare il processo di TM, riducendo al contempo la possibilità di errori umani.
  - *Integrazione*: IriusRisk può essere integrato con altri strumenti e processi di sviluppo del software, consentendo una maggiore coerenza e collaborazione all'interno dell'organizzazione.
  - *Analisi dettagliata delle minacce*: Lo strumento offre funzionalità per condurre un'analisi dettagliata delle minacce, in quanto vengono racchiuse in un unico tool le diverse metodologie esistenti sul mercato, consentendo di identificare e valutare con precisione i rischi per la sicurezza.
- **Contro**:

  - *Complessità*: L'uso di uno strumento come IriusRisk potrebbe richiedere una curva di apprendimento per comprendere appieno le sue funzionalità e utilizzarlo in modo efficace, specialmente per gli utenti meno esperti.
  - *Costo*: IriusRisk ha dei costi di licenza e di implementazione che potrebbero non essere accessibili a tutte le organizzazioni, specialmente quelle più piccole o meno finanziariamente stabili.
  - *Personalizzazione limitata*: Sebbene offra una struttura ben definita, potrebbe non essere possibile personalizzare completamente IriusRisk per adattarsi alle esigenze specifiche di ogni organizzazione, il che potrebbe limitare la sua flessibilità in certi contesti.
  - *Dipendenza dalla tecnologia*: L'uso esclusivo di IriusRisk potrebbe portare ad una dipendenza eccessiva dalla tecnologia, a discapito di un approccio più ampio che coinvolge anche persone e processi all'interno dell'organizzazione.

Framework personalizzato

- **Pro**:
  - *Adattabilità*: Un framework personalizzato può essere progettato e adattato specificamente per soddisfare le esigenze uniche dell'organizzazione, tenendo conto dei requisiti, delle infrastrutture e delle minacce specifiche.
  - *Flessibilità*: Avendo sviluppato il framework personalmente, si ha il controllo completo sulle funzionalità, le metodologie e i processi inclusi, consentendo una maggiore flessibilità nell'adattarlo e modificarlo secondo necessità.
  - *Integrazione completa*: Puoi progettare il framework per integrarsi perfettamente con altri strumenti e processi esistenti all'interno dell'organizzazione, migliorando la coerenza e l'efficienza complessiva delle operazioni di sicurezza.
  - *Conoscenza interna*: Creare un framework personalizzato ti dà una profonda comprensione delle sue funzionalità e dei suoi processi, consentendo una migliore gestione e manutenzione nel lungo termine.
- **Contro**:
  - *Complessità di sviluppo*: La progettazione e lo sviluppo di un framework personalizzato richiedono tempo, risorse e competenze significative, specialmente se si desidera una soluzione completa e sofisticata.

- *Manutenzione*: Una volta sviluppato, il framework richiede manutenzione continua per rimanere aggiornato con le nuove minacce, le tecnologie emergenti e i cambiamenti nelle esigenze aziendali.
- *Curva di apprendimento*: Gli utenti potrebbero richiedere una curva di apprendimento significativa per padroneggiare il framework personalizzato e utilizzarlo in modo efficace, specialmente se non sono stati coinvolti nel processo di sviluppo.
- *Rischio di errori*: Poiché il framework è stato creato internamente, potrebbe essere più soggetto a errori e vulnerabilità rispetto a soluzioni sviluppate e validate da terze parti. È importante eseguire rigorosi test e revisioni per garantire la sua affidabilità e sicurezza.
- *Metologie considerate*: Adattandolo alla specifica organizzazione, non vengono considerate tutte le metologie esistenti sul mercato e quindi si hanno riferimenti più stringenti.

In definitiva, la scelta tra l'adozione di un tool commerciale come IriusRisk e lo sviluppo di un framework personalizzato per il Threat Modeling dipende dalle esigenze specifiche e dalle risorse disponibili per ciascuna organizzazione. Entrambi hanno i loro punti di forza e di debolezza, e l'importanza di valutare attentamente questi fattori non può essere sottovalutata.

È importante notare che il framework personalizzato attualmente utilizzato dal cliente ha dimostrato di affrontare con successo le sfide del mondo IT, rappresentando un solido punto di partenza per il processo di Threat Modeling. Sebbene possa essere limitato rispetto a soluzioni commerciali più avanzate, il suo adattamento alle esigenze specifiche dell'organizzazione ne garantisce l'efficacia e la rilevanza.

Capitolo 8

Conclusioni e sviluppi futuri

Con lo studio affrontato in questa tesi si è voluto dimostrare il potenziale significativo del processo di Threat Modeling. In particolare partendo dall'analisi del *Software Development Life Cycle*, è emersa chiaramente l'importanza di integrare il Threat Modeling in tutte le fasi del processo di sviluppo del software. Questo approccio garantisce non solo una maggiore sicurezza dei prodotti, ma contribuisce anche ad una cultura aziendale orientata alla sicurezza informatica fin dalla fase di progettazione.

Il percorso teorico intrapreso nel comprendere il Threat Modeling, dalle sue origini storiche alla pratica contemporanea, ha permesso di esplorare una vasta gamma di metodologie, framework e strumenti utilizzati per identificare e gestire le minacce alla sicurezza, sottolineando i pro e i contro di ogni metodologia. Ci si è soffermati sul MITRE ATT&CK, in quanto framework di riferimento del cliente presso cui ho svolto il lavoro di tesi. L'analisi di questo framework ha offerto una panoramica dettagliata delle tattiche e delle tecniche utilizzate dagli aggressori, permettendo di comprendere meglio le minacce e di sviluppare contromisure più mirate ed efficaci. L'ATT&CK per quanto si sia rivelato uno strumento prezioso per acquisire una visione più approfondita delle minacce informatiche, si è dimostrato avere delle criticità e limitazioni, sottolineando l'importanza di adattare il Threat Modeling alle specifiche esigenze e contesti delle organizzazioni. Da qui, quindi, la necessità di proporre al cliente nuove soluzioni per poter gestire in modo più efficace il processo di TM.

Lo stato dell'arte ci propone diversi tool commerciali che automatizzano la fase di modellazione delle minacce. Acquistare un tool non sempre però risulta l'opzione percorribile per alcune aziende. Il lavoro di tesi si è concentrato sullo:

1. sviluppo di un framework personalizzato per identificare e gestire le minacce in modo più mirato, fornendo un vantaggio competitivo nell'affrontare le sfide

sempre mutevoli del panorama della sicurezza informatica;

2. studio di un tool commerciale (IriusRisk) per valutare le potenzialità e le limitazioni di una soluzione pronta all'uso.

Se da un lato un tool commerciale offre una vasta gamma di funzionalità e caratteristiche, dall'altro presenta delle restrizioni legate alla flessibilità ed alla personalizzazione che possono essere cruciali per alcune organizzazioni. Il confronto tra il framework personalizzato e il tool commerciale ha evidenziato le differenze e le similitudini tra i due approcci, offrendo spunti interessanti per futuri sviluppi e miglioramenti.

Dato che il framework personalizzato è già in uso presso il cliente, questo costituisce un solido punto di partenza che sottolinea la sua efficacia. Tuttavia, il framework personalizzato richiede un aggiornamento periodico sulla base delle nuove minacce e i cambiamenti nelle regolamentazioni come il *General Data Protection Regulation* per garantire un miglioramento continuo e una maggiore completezza, ed è necessario intraprendere ulteriori passi. Attualmente, il framework implementato per il cliente si concentra esclusivamente sui servizi offerti da Azure, pertanto uno dei prossimi step potrebbe essere l'ampliamento della sua portata, includendo altri provider di servizi cloud. Inoltre, sarà importante continuare ad arricchire il catalogo delle minacce con nuove informazioni e a perfezionare le strategie di mitigazione già esistenti. Questo processo richiederà un costante monitoraggio del panorama delle minacce informatiche, nonché una valutazione delle nuove tecnologie e degli scenari emergenti.

È essenziale riconoscere che il cambiamento è la nuova costante e che la difesa cibernetica deve essere altrettanto dinamica e adattabile. Le regolamentazioni come il GDPR sono un riflesso di questa consapevolezza crescente dell'importanza di proteggere i dati personali e sensibili. Esse indicano chiaramente che la sicurezza informatica non può essere trascurata e che è necessario adottare misure adeguate per garantire la privacy e la sicurezza dei dati. Pertanto, qualsiasi tipo di organizzazione deve mantenere un impegno continuo nei confronti della *security*, investendo nelle migliori pratiche, tecnologie avanzate e aggiornamento professionale.

Sono convinta che il vero punto di forza risieda nella collaborazione tra i vari *stakeholders* di una compagnia. Ognuno ha la possibilità di contribuire alla progettazione e alla definizione dei prodotti in modo sicuro, il che porta ad un miglioramento complessivo della sicurezza dell'intera organizzazione. Solo attraverso un approccio proattivo e vigilante si può sperare di mantenere una consistenza nella difesa contro le minacce informatiche.

Bibliografia

[1] Zohaib Ahmed, Syed Muhammad Danish, Hassaan Khaliq Qureshi e Marios Lestas. «Protecting IoTs from Mirai Botnet Attacks Using Blockchains». In: *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. 2019, pp. 1–6. DOI: 10.1109/CAMAD.2019.8858484.

[2] Da-Yu Kao e Shou-Ching Hsiao. «The dynamic analysis of WannaCry ransomware». In: *2018 20th International Conference on Advanced Communication Technology (ICACT)*. 2018, pp. 159–166. DOI: 10.23919/ICACT.2018.8323682.

[3] Clusit. *Rapporto Clusit 2022 sulla sicurezza ICT in Italia*. URL: https://clusit.it/wp-content/uploads/download/Rapporto-Clusit-ottobre-2022_web.pdf.

[4] Accenture. *State of Cybersecurity Resilience 2023*. URL: https://www.accenture.com/us-en/insights/security/state-cybersecurity.

[5] Tzu Sun. *The art of war*. Hachette UK, 1994.

[6] Tony UcedaVelez e Marco M Morana. *Risk Centric Threat Modeling: process for attack simulation and threat analysis*. John Wiley & Sons, 2015.

[7] Livinus Obiora Nweke e Stephen Wolthusen. «A review of asset-centric threat modelling approaches». In: (2020).

[8] Nataliya Shevchenko, Timothy A Chick, Paige O'Riordan, Thomas P Scanlon e Carol Woody. *Threat modeling: a summary of available methods*. Rapp. tecn. Carnegie Mellon University Software Engineering Institute Pittsburgh United ..., 2018.

[9] Adam Shostack. *Threat modeling: Designing for security*. John Wiley & Sons, 2014.

[10] Indri Sulistyowati e RV Hari Ginardi. «Information security risk management with octave method and iso/eic 27001: 2013 (case study: Airlangga university)». In: *IPTEK Journal of Proceedings Series* 1 (2019), pp. 32–38.

[11] Blake E Strom, Andy Applebaum, Doug P Miller, Kathryn C Nickels, Adam G Pennington e Cody B Thomas. «Mitre att&ck: Design and philosophy». In: *Technical report*. The MITRE Corporation, 2018.

[12] Secure Online Desktop. *Mitre Att&ck ™: una panoramica*. URL: `https://www.secure-od.com/it/mitre-attck-una-panoramica/`.

[13] Michael Collier e Robin Shahan. *Microsoft azure essentials-fundamentals of azure*. Microsoft Press, 2015.

[14] NIST. *Software Supply Chain Security Guidance Under Executive Order (EO) 14028*. URL: `https://www.nist.gov/system/files/documents/2022/02/04/software-supply-chain-security-guidance-under-EO-14028-section-4e.pdf`.

[15] IriusRisk. *Not your average origin story.* URL: `https://www.iriusrisk.com/about-us`.

[16] James Reddick. «FDA can now reject new medical devices over cyber standards». In: *The Record* (2023). URL: `https://therecord.media/fda-medical-device-cyber-standards`.

[17] Anna Georgiadou, Spiros Mouzakitis e Dimitris Askounis. «Assessing MITRE ATT&CK Risk Using a Cyber-Security Culture Framework». In: *Sensors* 21.9 (2021). ISSN: 1424-8220. DOI: `10.3390/s21093267`. URL: `https://www.mdpi.com/1424-8220/21/9/3267`.

Ringraziamenti

Ho il cuore in gola mentre scrivo quest'ultima pagina di una lunghissima avventura. E' emozionante ripensare ad ogni istante vissuto che mi ha portato qui con voi ora. Mi sminuisco in ogni momento e invece oggi vorrei ringraziarmi perchè non è stato affatto semplice arrivare qui. La sensazione di non farcela è stata una costante in questi anni. Ma ho resistito, a testa alta perchè alla fine chi lo decide quanto valgo? Quindi grazie a me che anche in ogni momento di sconforto, mi sono asciugata le lacrime e ho pensato "che me ne frega, andiamo avanti".
Ma questa non è stata solo la mia storia, ma anche la vostra:

La tua mamma, che dal giorno 0 hai creduto che nonostante tutto io potessi farcela. Sei sempre stata la mia più grande fan e credo fortemente che può capitare qualsiasi cosa nella vita, ma la fortuna di avere una donna come te che mi guida è il dono più prezioso che ho. Mi hai insegnato ad avere fede e credere in me stessa, anche se con molte difficoltà, perchè prima o poi la vita ti sorprende. Spero un giorno di poterti regalare anche solo in minima parte l'amore che mi regali tu da sempre. Sei un essere speciale e non lo dico perch' sono di parte, ma perchè è così, hai una luce diversa e tutti se ne rendono conto.

La tua Pino, che in punta di piedi da oramai un'infinità di anni ti prendi cura di me. Ti sei innamorato della mamma e hai deciso di prenderti la responsabilità di avere una figlia non tua. Questo probabilmente è stato il gesto d'amore più grande che uomo può fare. Mi hai insegnato ad essere riflessiva, puntigliosa, scrupolosa ma anche ad essere disponibile e altruista e mai come in questi ultimi mesi mi sono resa conto di essere proprio la tua fotocopia. Com'è stato possibile? Grazie perchè anche se sei un lamentoso di prima categoria (anche questo mi hai trasmesso purtroppo) ho sempre saputo di poter contare su di te, senza riserva. Ti voglio bene ma in un modo che non si può nemmeno descrivere. E sappi, caro garante, che ora sei incastrato per almeno altri 7 anni...Tu e la mamma siete il mio orgoglio più grande.

La tua Nonna, che con gratificazione hai sempre parlato di me a tutte le persone

che conosci, facendomi sentire così contenta ad ogni traguardo, non tanto per il traguardo in sè ma perchè sapevo di renderti felice.

La tua Nonno, che sei il mio gioiello più raro. Mi avevi promesso di vivere così a lungo da poter assistere a questo momento e sappiamo tutti quanto tu sia un uomo testardo e di parola. Ho iniziato a fare i compiti con te, mi hai insegnato a leggere l'orologio e ad allacciarmi le scarpe e ora mi guardi con rispetto e ammirazione. Credo che alla fine mi basti questo.

La tua Marcy, che sei da sempre un fratello per me. Sai quanto ti voglio bene ma probabilmente ora ancora di più dopo aver messo al mondo insieme ad Anna, la mia gioia più grande: Chiaraluna. Quell'esserino minuscolo mi ha rubato il cuore e non l'abbandonerò mai. Conserverò con cura ogni insegnamento di questi anni per poterla accompagnare e supportala in ogni sua scelta.

La Tua Alessio... quando ci siamo incontrati un pò di anni fa dovevo laurearmi in triennale e quindi dopo pochi mesi ti sei trovato a festeggiare un traguardo di una storia che avevi vissuto molto poco. Poi con caparbietà, abbiamo fatto le valigie e siamo atterrati a Torino e da lì è iniziata una nuova storia, la nostra, nella nostra casa. Hai scoperto quanto fosse difficile e frustrante per me combattere contro l'ansia, la preoccupazione, la sensazione di inferiorità. E mentre ti ho sempre parlato di tutte queste cose brutte, tu riuscivi solo a dirmi che ero perfetta così, che non ero inferiore a nessuno e che qualsiasi cosa sarebbe successa, avrei trovato te a casa a braccia aperte. Voglio ripeterti di nuovo, che hai dimostrato di valere molto di più di quanto tu creda. Più di tutti, questa è veramente anche la tua storia. Oggi metto un punto io e tu lo fai con me. Un punto che spero ci porti a tanti altri traguardi insieme, che sogniamo da anni. Ti amo amorino (ma questo te l'ho dimostrato a Inter-Juve finale di supercoppa, 120esimo minuto... tu sai) e grazie perchè è facile comprendersi quando si fa la stessa vita ma la vera sfida è farlo quando si appartiene a due mondi diversi.

La tua Marika. Mentre sono arrivata a questo punto, mi hai appena inviato 14 minuti di audio. Questo sintetizza alla grande il nostro rapporto. Non ce n'è mai fregato niente di essere così distanti perchè il bene va oltre. Anzi la distanza ha rafforzato le mura di un palazzo già molto solido, ci ha permesso di spingerci oltre e dirci "Mi manchi" e di capire l'importanza di quanto è bello esprimere i nostri sentimenti. Tu mi hai ascoltata e mi hai sempre detto con onestà il tuo pensiero. Mi hai aiutata a vincere i momenti più bui sebbene tu avessi i tuoi a cui pensare. E' iniziato tutto con te dal primo anno di liceo, abbiamo poi deciso di andare a vivere a Napoli insieme una sera mentre mangiavamo al MC, ci siamo trovate la casa senza dire nulla ai nostri genitori e poi all'improvviso da vere sorelle

abbiamo condiviso ogni istante. Grazie per far parte della mia vita, grazie per non avermi mai giudicata, per avermi fatto fare tante, troppe risate... mi sento davvero fortunata. E ora, dopo aver messo un punto insieme al tuo percorso accademico, eccoci qui a mettere un punto insieme al mio. Se l'ultima volta ti ho promesso che la distanza non ci avrebbe mai separato perchè tu sei la mia casa ovunque io sia e abbiamo dimostrato entrambe di mantenere questa promessa, ora ti voglio promettere che nel nostro futuro le cose più belle le vivremo insieme e vorrò sempre te ad accompagnarmi in ogni nuovo passo.

La tua Andrea. Qui ci vorrebbe un grosso momento di pausa per respirare e con calma andare avanti. Non è stato un caso incontrarci, non è stato un caso niente. Ci siamo proprio scelti e non trovo le parole per noi perchè mi sembra tutto riduttivo, ma sono obbligata e quindi ci provo. Oltre al fatto che sei stato determinante per me in questi ultimi anni universitari, tu sei incredibile, ma non perchè appunto ci siamo scoperti affini nello studiare insieme o nel modo di affrontare le cose. Tu sei incredibile perchè hai avuto la determinazione di accettare e di volere ogni mio difetto e se penso al perchè, non trovo le risposte. L'hai fatto e basta senza chiedere niente in cambio, anzi. La cosa più sorprendente è che tu mi dici "Ma come farei senza di te?" e mi sembra assurdo perchè io penso "Ma come farei senza Andrea?". Sarò per sempre la tua più grande sostenitrice. Ma tu già sai tutto, non è stato mai necessario parlare troppo tra di noi... e ora basta perchè già saremo tutti in lacrime, inutile continuare questa tortura... Grazie Andre ma dal più profondo del cuore.

La tua Luca, che con calma abbiamo imparato a volerci bene e ad ascoltarci. Se qualcuno ci sentisse parlare, probabilmente la prima cosa che penserebbe è "Ma come fanno ad essere amici sti due che sono così diversi?". La diversità è stata proprio il nostro punto di incontro perchè ci ha permesso di analizzarci e di volerci bene proprio per come siamo ma nel tempo ho capito che forse gli interessi sono diversi, ma i valori sui quali basiamo le nostre vite sono gli stessi. Sono passati oramai due anni da quando ci siamo conosciuti e posso dire con fermezza che sei proprio una bella persona. Ti nascondi dietro le tue insicurezze avvolte ma non smetterò mai di ricordarti quanto invece tu sia fortissimo.

La tua Davide, che sei stato una delle prime persone che ho incontrato al Poli. Vabbè effettivamente non poteva essere diversamente considerando che tu conosci tutti. Ma rispetto ai "tutti", io mi sento privilegiata perchè mi hai fatto entrare nella tua vita e questo non lo riservi ad ogni persona. Ci siamo amati ma anche odiati, ma se non è questo l'amore allora cos'è?

La tua Ilaria, che seppur non avevi alcun motivo per prenderti cura di me, in questi

ultimi mesi mi hai supportata, aiutata e stimolata. Sei una grande donna, con un cuore immenso e questo ha reso tutto più semplice. Grazie veramente!

Un grazie speciale va anche a Carlo, Chicca e Valeria perchè seppur non sentendoci ogni giorno, ho sempre trovato in voi conforto. Credo che l'amicizia non sia sentirsi costantemente ma piuttosto la gioia di ritrovarsi dopo mesi e raccontarsi ogni cosa. Avete creduto sempre in me e non mi avete mai giudicata. Vi voglio bene!

Prima di concludere questi ringraziamenti, ho bisogno di fare una menzione d'onore a Federica. Questo traguardo è per te. Ti dedico la mia gioia. E ora mentre mi si bloccano le parole, voglio solo dirti che ti sento, lo so che sei qui con me e continuerò a combattere sempre come mi hai detto di fare. Sei sempre stata fiera di me, e mi dispiace solo di non averti detto in tempo che anche io lo sono sempre stata di te. Ciao Amica, non ti divertire troppo con Papà e con Siria.

Rimanendo in tema macchina che ovviamente è la gioia delle ultime settimane, siete stati tutti la benzina del mio percorso. È stata una tortura, una lotta continua fino all'ultimo secondo ma voi avete combattuto con me, sempre. Chi da vicino e chi da lontano. La cosa più bella è pensare che abbiamo solo definito un inizio per una nuova storia che spero continueremo a vivere insieme. Vi voglio davvero troppo bene e raga mado vorrei urlarlo fortissimo: "HO FINITOOOOOOO".