

Indice

| | |
|---|-----------|
| 1. Introduzione | 4 |
| 2. Progettazione | 5 |
| 2.1 Early Requirement Analysis | 6 |
| Fornitore | 7 |
| Trasformatore | 8 |
| Cliente | 8 |
| 2.2 Preliminary Risk Assessment..... | 9 |
| 2.2.1 Risk Identification..... | 9 |
| 2.2.2 Risk Analysis | 10 |
| Asset value and exposure assessment..... | 10 |
| Threat Identification | 10 |
| 2.2.3 Risk decomposition..... | 12 |
| Attack Assessment | 12 |
| Use Case | 13 |
| Misuse Case..... | 18 |
| Abuse Case | 20 |
| Attack Tree | 29 |
| 2.2.4 Risk Reduction..... | 32 |
| Control Identification | 32 |
| Feasibility Assessment..... | 32 |
| 3. Design | 34 |
| 3.1 Secure design | 34 |
| 4. Implementazione..... | 36 |
| 4.1 Blockchain | 36 |
| 4.1.1 Quorum..... | 36 |
| 4.1.2 Smart Contracts..... | 37 |
| CarbonFootprint | 37 |
| NFT_FootPrint | 38 |
| 4.3 Interfaccia utente | 40 |
| 4.3.1 JavaScript, Node.js e web3.js | 40 |
| 4.4 Struttura progetto | 41 |
| 5. Guida all'utilizzo..... | 43 |

Indice figure e tabelle

| | |
|--|----|
| Figura 1 - Schema di analisi dei requisiti guidata dal rischio | 5 |
| Figura 2 - Contesto dell'applicazione..... | 6 |
| Figura 3 - Modello I* | 7 |
| Figura 4 - Scala di Likert..... | 10 |
| Figura 5 - Attack Tree Violation of Integrity..... | 29 |
| Figura 6 -Attack Tree Unreliability..... | 29 |
| Figura 7 - Attack Tree Violation of Authentication | 30 |
| Figura 8 - Attack Tree Violation of Confidentiality, Absence of Resilience | 30 |
| Figura 9 - Attack Tree Violation of Authorization | 31 |
| Figura 10 - Attack Tree Danger | 31 |
| Figura 11 - Attack Tree Denial of Service | 31 |
| Figura 12 - Struttura Lot CarbonFootprint..... | 37 |
| Figura 13 - Mapping CarbonFootprint..... | 38 |
| Figura 14 - Strutture dati NFT_Footprint | 38 |
| Figura 15 - Funzione "mint" NFT_Footprint..... | 39 |
| Figura 16 - Metodo "tokenURI" NFT_Footprint..... | 39 |
| Figura 17 - Struttura progetto | 41 |
| Tabella 1 - Identificazione degli asset | 9 |
| Tabella 2 - Modello STRIDE | 11 |
| Tabella 3 - Matrice del rischio..... | 12 |
| Tabella 4 - Use Case 1..... | 13 |
| Tabella 5 - Use Case 2 | 14 |
| Tabella 6 - Use Case 3 | 15 |
| Tabella 7 - Use Case 4..... | 16 |
| Tabella 8 - Use Case 5 | 17 |
| Tabella 9 - Misuse Case 1 | 18 |
| Tabella 10 - Misuse Case 2 | 19 |
| Tabella 11 - Abuse Case 1..... | 20 |
| Tabella 12 - Abuse case 2..... | 21 |
| Tabella 13 - Abuse Case 3..... | 22 |
| Tabella 14 - Abuse Case 4..... | 24 |
| Tabella 15 - Abuse Case 5 | 25 |
| Tabella 16 - Abuse Case 6..... | 26 |
| Tabella 17 - Abuse Case 7 | 28 |
| Tabella 18 - Risk Reduction | 33 |

1. Introduzione

La seguente documentazione è atta a illustrare il processo di progettazione ed implementazione di un software distribuito per il tracciamento delle emissioni generate da prodotti alimentari. Il lavoro è stato diviso nelle seguenti attività:

- **Progettazione:** il punto di partenza della attività sarà la creazione di un modello I*, avente lo scopo di fornire una prima visione delle funzionalità richieste dal software e dei relativi attori in gioco. In seguito, verranno individuati tutti gli asset coinvolti, ai quali verranno associate delle politiche di sicurezza ed un modello STRIDE. Come ultimo passo nella progettazione, verranno catalogate, all'interno dello schema di Jacobson, tutte le azioni eseguibili dagli attori espressi nel diagramma I*: Use case, Abuse case e Misuse case.
- **Design e implementazione:** nei capitoli che racchiudono tali argomenti verranno analizzate le varie alternative software che possono garantire un design sicuro e, una volta scelta la soluzione più appropriata, verrà descritta l'intera fase di implementazione del sistema.
- **Guida all'utilizzo:** in questa sezione verrà illustrata tutta la procedura necessaria per usufruire del servizio in esame in tale elaborato.

2. Progettazione

Per la progettazione è stato preso come riferimento lo schema riportato in Figura 1: i paragrafi sottostanti sono strutturati seguendo tale flusso.

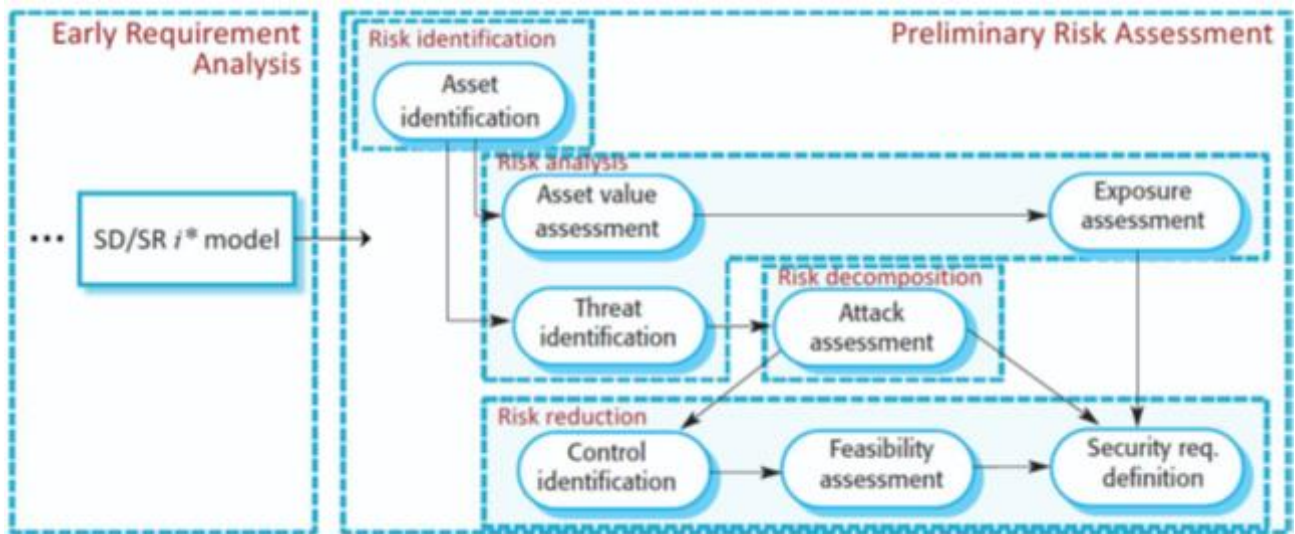


Figura 1 - Schema di analisi dei requisiti guidata dal rischio

2.1 Early Requirement Analysis

Il settore in cui si è operato è la filiera agroalimentare nonostante i contesti applicativi possano essere molteplici. La richiesta è quella di creare un software che si appoggia su una blockchain, in grado di effettuare il tracciamento delle emissioni di CO₂, generate nei processi di produzione dei prodotti alimentari. Lo scopo è quello di tener traccia di tali emissioni in una struttura che ne eviti la modifica. Entrando nello specifico il software sarà adibito alle seguenti figure:

Fornitori: coloro che riforniscono il mercato di materie prime

Trasformatori: coloro che apportano lavorazioni a tali materie generando prodotti finiti

Clienti: coloro che acquisteranno i prodotti per la consumazione

Lo scenario a cui ci si è attenuti è quello espresso in Figura 2. Il fornitore è il primo soggetto ad interfacciarsi con il software nella catena lavorativa. Egli avrà il compito di registrare nella blockchain le sue materie prime con le relative emissioni. In seguito, il trasformatore dovrà essere in grado di acquistare tali materie prime per poi lavorarle. Per ognuna delle lavorazioni si dovrà registrare il relativo apporto di emissioni fino a che non si arriverà al prodotto finito. Nel momento in cui verrà acquistato un prodotto finale o una materia prima dal cliente il software dovrà generare un NFT, di cui approfondiremo in seguito, il quale certificherà il totale delle emissioni generate nella lavorazione di quei prodotti.

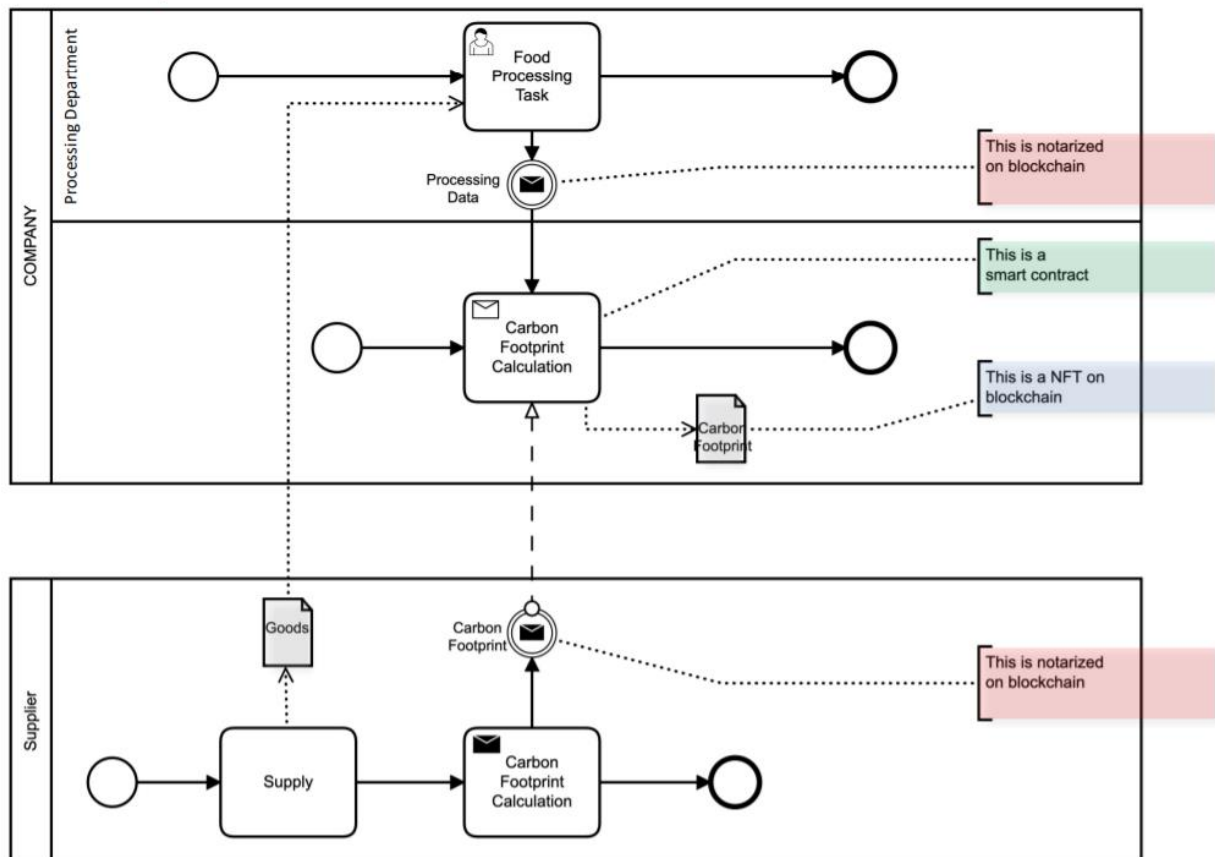


Figura 2 - Contesto dell'applicazione

Una volta recepiti i requisiti richiesti all'applicativo, è stato formulato un modello I* per astrarre il comportamento del software. Nel modello in Figura 3 sono stati indicati quattro attori, di cui tre corrispondenti alle figure sopra citate: Fornitore, Trasformatore e Cliente; come ultimo attore è stato messo il Software stesso.

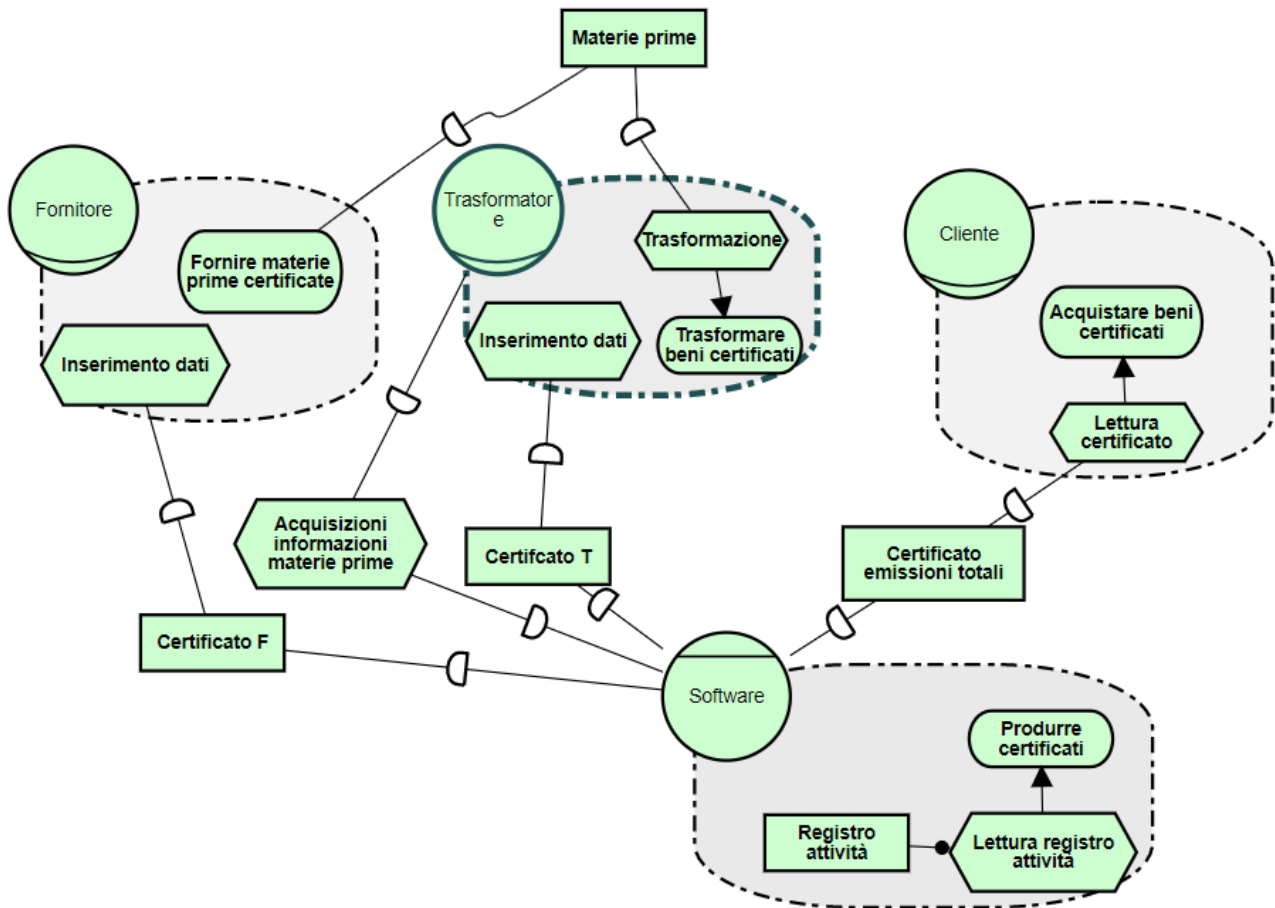


Figura 3 - Modello I*

Fornitore

Il Fornitore avrà a disposizione un task per l'inserimento dei dati relativi alle materie prime che vorrà introdurre nel sistema. L'output di tale task sarà una risorsa denominata Certificato F che sarà introdotta nel software. Come si evince dal modello, l'obiettivo del Fornitore è quello di fornire materie prime con emissioni certificate.

Trasformatore

Anche il Trasformatore avrà a disposizione il medesimo task di inserimento dei dati del Fornitore, ma questo sarà relativo ai prodotti finali, dunque, genererà un'altra tipologia di certificato. Avendo il Trasformatore il compito di lavorare le materie prime, bisognerà permettergli di acquisire le informazioni relative; per fare ciò sarà da mettere a disposizione una funzione per l'acquisizione di tali dati attraverso il software. Un ulteriore task sarà adibito al processo di trasformazione che, tramite l'acquisizione di materie prime dal fornitore, permetterà di conseguire l'obiettivo di trasformare beni certificati.

Software

Il Software avrà lo scopo di produrre i certificati relativi ai prodotti/materie prime al momento dell'acquisto da parte dei clienti. Per conseguire questo obiettivo, si dovrà interfacciare con un registro delle attività (quindi delle emissioni) dal quale verrà estrapolato il "footprint" finale associato.

Cliente

Il Cliente, avente lo scopo di acquistare beni certificati, avrà la possibilità di consultare il footprint finale associato ad ogni prodotto/materia prima.

2.2 Preliminary Risk Assessment

2.2.1 Risk Identification

La prima attività nella valutazione preliminare del rischio ovviamente è l'identificazione degli asset: elementi che hanno valore per il sistema (risorse e requisiti funzionali) che è necessario proteggere da eventuali attacchi.

Per ogni asset sono stati identificati gli obiettivi e le policy di sicurezza, ovvero l'insieme delle regole atte a garantire la sicurezza all'interno del software. La tabella sottostante è riepilogativa di questa fase.

| Asset | Value | Objective | Exposure |
|---------------------------------------|-------|--|----------|
| Certificato | 7 | I vari certificati e ricevute attestano l'impronta di CO2 durante il processo di produzione e trasformazione successiva di ogni prodotto. Deve essere integro e riportante dati coerenti e aggiornati. | 7 |
| Inserimento dati | 5 | La funzione di inserimento dei dati (come le quantità di prodotto e l'impronta di CO2 emessa) deve garantire che non vengano immesse informazioni diverse da quelle richieste | 6 |
| Lettura registro attività/certificato | 6 | Renderlo disponibile solo a utenti autorizzati. | 5 |
| Autorizzazione per ruoli | 7 | Il processo di autenticazione è obbligatorio al fine di utilizzare il software. In seguito, saranno disponibili solo le funzionalità predisposte alla tipologia di utente | 7 |
| Generazione certificato | 6 | Coerenza nel processo di generazione a partire dai dati sulle emissioni di CO2 di ogni processo. | 6 |
| Registro attività | 6 | Il registro deve riportare ogni azione effettuata sul prodotto, dalla fornitura, al trasporto, alla trasformazione. Tale registro deve essere consultato esclusivamente dal software. | 6 |

Tabella 1 - Identificazione degli asset

2.2.2 Risk Analysis

Asset value and exposure assessment

Conseguentemente è entrata in gioco l'attività di analisi tramite la quale per ciascun asset sono stati indicati, oltre agli obiettivi, due parametri: **value** e **exposure**. Il primo rappresenta una valutazione del valore che viene attribuito ad ogni asset, mentre il secondo una valutazione delle potenziali perdite conseguenti al verificarsi di un attacco a quell'asset. Per attribuire a **value** ed **exposure** un peso, è stata adottata la scala di Likert riportata in Figura 5.

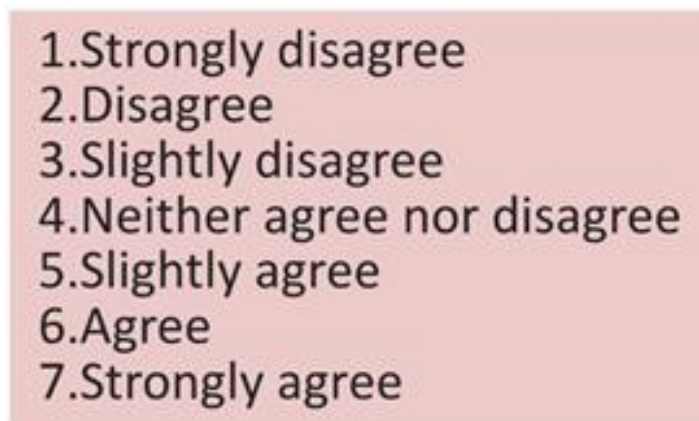


Figura 4 - Scala di Likert

Threat Identification

Una volta identificati e analizzati gli asset, sono state individuate le possibili minacce a cui quest'ultimi potrebbero essere esposti. Tali minacce, attraverso il modello Stride, sono state classificate nella tabella sottostante facendo riferimento a una o più minacce notevoli espresse dal modello:

- **Spoofing**: usare l'identità di qualcuno o qualcosa (violazione dell'autenticazione);
- **Tampering**: modifica di dati (violazione dell'integrità);
- **Repudiation**: dichiarare di non aver compiuto un'azione (violazione del non ripudio);
- **Information disclosure**: esporre delle informazioni a un soggetto non autorizzato (violazione della confidenzialità);
- **Denial of service**: non rendere disponibile un servizio (violazione della disponibilità);
- **Elevation of privilege**: guadagnare permessi senza un'autorizzazione (violazione dell'autorizzazione);
- **Danger**: danneggiare qualcuno o qualcosa (violazione della safety);
- **Unreliability**: impossibilità di fare affidamento (violazione dell'affidabilità);
- **Absence of resilience**: incapacità di riprendersi rapidamente da incidenti informatici (violazione della resilienza).

| Asset | Value | Spoofing | Tampering | Repudiation | Information disclosure | DOS | Elevation of privilege | Danger | Unreliability | Absence of Resilience | Exposure | Attack | Inherent Probability | Inherent Risk |
|---------------------------------------|-------|----------|-----------|-------------|------------------------|-----|------------------------|--------|---------------|-----------------------|-----------|---|----------------------|---------------|
| Certificato | 7 | | | | x | | | | | | Medium | CAPEC-117: Interception | Low | Low |
| | | | x | | x | | x | | x | | Very High | CAPEC-94: Adversary in the Middle | High | Very High |
| | | | | | x | x | | | | | Very High | CAPEC-123: Buffer Manipulation | High | Very High |
| Inserimento dati | 5 | | | | | | | x | | | Medium | CAPEC-153: Input Data Manipulation | High | Medium |
| | | | x | | x | | x | | x | | Very High | CAPEC-94: Adversary in the Middle | High | Very High |
| | | x | x | | x | | x | | | | High | CAPEC-560: Use of Known Domain Credentials | High | High |
| Lettura registro attività/certificato | 6 | x | x | | x | | | | | | High | CAPEC-21: Exploitation of Trusted Identifiers | High | High |
| | | | x | | x | | x | | x | | Very High | CAPEC-94: Adversary in the Middle | High | Very High |
| | | x | x | | x | | x | | | | High | CAPEC-560: Use of Known Domain Credentials | High | High |
| Autorizzazione per ruoli | 7 | x | | | | | x | | x | | Medium | CAPEC-115: Authentication Bypass | High | Medium |
| | | | | | | | x | | x | | Medium | CAPEC-114: Authentication Abuse | High | Medium |
| | | x | | | x | | x | x | | | Medium | CAPEC-122: Privilege Abuse | High | Medium |
| Generazione certificato | 6 | | x | | x | | x | x | x | x | Medium | CAPEC-233: Privilege Escalation | High | Medium |
| | | | | | | x | x | x | | | Medium | CAPEC-165: File Manipulation | High | Medium |
| | | | | | x | | | | | | High | CAPEC-548: Contaminate Resource | High | High |
| Registro attività | 6 | | | | x | | | | | | Medium | CAPEC-117: Interception | Low | Low |
| | | x | x | | x | | | x | x | | Medium | CAPEC-151: Identity spoofing | Low | Medium |
| | | | x | | | | x | | x | | Medium | CAPEC-115: Authentication Bypass | Low | Medium |
| | | | | | | | | | x | | Medium | CAPEC-148: Content Spoofing | High | Medium |

Tabella 2 - Modello STRIDE

Per maggiore completezza le minacce riportate sono state estrapolate da CAPEC, uno tra i più famosi raccoglitori di “pattern d’attacco”. Ciò ha permesso sia di raccogliere molte informazioni per ogni minaccia, ma anche di ricavarne le possibili soluzioni. Grazie a CAPEC sono state riportate anche l’impatto e le probabilità inerenti alle minacce.

2.2.3 Risk decomposition

Attack Assessment

Nel modello STRIDE riportato nella Tabella 2 si evince una valutazione numerica associata ad ogni rischio. Questi valori sono stati calcolati attraverso la seguente formula:

Rischio = Probabilità x Impatto

Un altro vantaggio di aver usato CAPEC, infatti, è stato quello di aver già identificati probabilità e impatto per ogni attacco. Ultimo passo è stato decidere una metrica per classificare i risultati: la cosiddetta matrice del rischio.

| | | Impact | | | |
|-------|-----------|--------|--------|------|-----------|
| | | Low | Medium | High | Very High |
| Prob. | Low | | | | |
| | Medium | | | | |
| | High | | | | |
| | Very High | | | | |

Tabella 3 - Matrice del rischio

La valutazione degli attacchi, inoltre, passa anche attraverso una classificazione di tutte quelle azioni che posso interfacciarsi con il sistema:

Use case: azioni che rappresentano le interazioni tra un attore e un sistema per raggiungere un obiettivo. L'attore può essere un essere umano o altro sistema esterno.

Abuse case: azioni intenzionali da parte di un attaccante che possono essere dannose per il sistema.

Misuse case: azioni dannose per il sistema (non dovrebbero essere permesse) dovute ad atti involontari da parte di qualche attore.

È facile evincere dalla classificazione, che le minacce fanno parte esclusivamente delle ultime due categorie. Attraverso lo schema di Jacobson sono state riportate tutta una serie di informazioni relative ad ognuna di queste azioni.

Use Case

| | | | |
|------------------------------------|--|----------------|----------|
| Case Type | Use Case | Case ID | 1 |
| Case Name | Inserimento dati | | |
| Actors | Fornitore, Trasformatore e il Software | | |
| Description | È la funzione che permette al fornitore e al trasformatore di inserire i dati relativi alle emissioni all'interno del software. | | |
| Data | Emissioni e ulteriori dati | | |
| Stimulus and preconditions | Gli attori in questione devono autenticarsi e successivamente possono accedere al servizio. Successivamente devono indicare il proprio ruolo e in base a questo selezionare l'operazione desiderata. | | |
| Basic Flow | 1) Autenticazione 2) Dichiarazione ruolo 3) Eseguire l'inserimento con i dati necessari 4) Il software immette questi dati nel registro attività | | |
| Alternative Flow | | | |
| Exception Flow | Potrebbero generarsi delle eccezioni durante l'inserimento dei dati quali: dati non corretti o non accettati dal sistema. | | |
| Response and Postconditions | Una volta effettuato l'inserimento verrà visualizzato un messaggio di riepilogo con l'avvenuta conferma o meno dell'invio. | | |
| Non Functional Requirements | Integrità, Presupposizione dell'affidabilità dei dati | | |
| Comments | | | |

Tabella 4 - Use Case 1

| | | | |
|-----------------------------|---|---------|---|
| Case Type | Use Case | Case ID | 2 |
| Case Name | Lettura registro attività | | |
| Actors | Software | | |
| Description | Funzione che permette la consultazione delle operazioni effettuate dagli utenti. | | |
| Data | Tutti i dati relativi ad ogni operazione che effettua un cambiamento del footprint (es: quantità/emissioni prodotti, quantità/emissioni materie prime, emissioni nel trasporto/trasformazione ecc.) | | |
| Stimulus and preconditions | Accessibilità del software al registro attività e integrità dei dati | | |
| Basic Flow | Il software inizialmente accede al registro ed effettua una ricerca tra le varie attività per estrapolarne il contenuto | | |
| Alternative Flow | | | |
| Exception Flow | Nel caso in cui non viene trovata l'attività coerente con la ricerca | | |
| Response and Postconditions | Il software acquisisce l'emissione relativa all'attività | | |
| Non Functional Requirements | Integrità, Confidenzialità | | |
| Comments | | | |

Tabella 5 - Use Case 2

| | | | |
|-----------------------------|--|---------|---|
| Case Type | Use Case | Case ID | 3 |
| Case Name | Generazione certificato | | |
| Actors | Software | | |
| Description | Attività che svolge il software in seguito alla lettura del registro delle attività dalla quale estrapola i dati delle emissioni | | |
| Data | Dati provenienti dall'inserimento che sono contenuti nel registro attività | | |
| Stimulus and preconditions | Accessibilità al registro attività e integrità dei dati in ingresso | | |
| Basic Flow | 1) Consultazione dati 2) Rilascio certificato | | |
| Alternative Flow | | | |
| Exception Flow | Problemi di consultazione nei dati (ostruzione, flooding ecc.) | | |
| Response and Postconditions | Certificato | | |
| Non Functional Requirements | Integrità | | |
| Comments | | | |

Tabella 6 - Use Case 3

| | | | |
|------------------------------------|--|----------------|---|
| Case Type | Use Case | Case ID | 4 |
| Case Name | Lettura del certificato | | |
| Actors | Cliente, Software | | |
| Description | Il cliente legge il certificato per verificare il corretto rispetto delle emissioni di un determinato prodotto | | |
| Data | Dati relativi al footprint totale necessario alla produzione/trasformazione del prodotto | | |
| Stimulus and preconditions | Esistenza del certificato | | |
| Basic Flow | Il cliente tramite il software richiede la lettura del certificato di un prodotto. Sarà compito del software inviare in uscita il certificato per permetterne la fruizione | | |
| Alternative Flow | | | |
| Exception Flow | Certificato non trovato | | |
| Response and Postconditions | Certificato relativo al prodotto rispetto al quale si è effettuata la ricerca | | |
| Non Functional Requirements | Integrità, Autorizzazione e Autenticazione | | |
| Comments | | | |

Tabella 7 - Use Case 4

| | | | |
|-----------------------------|--|---------|---|
| Case Type | Use Case | Case ID | 5 |
| Case Name | Autorizzazione per ruoli | | |
| Actors | Software | | |
| Description | Funzione demandata al software per permettere lo svolgimento delle proprie attività sulla base di una precedente autenticazione. | | |
| Data | Credenziali di accesso degli utenti | | |
| Stimulus and preconditions | Correttezza delle credenziali | | |
| Basic Flow | 1) Verifica livello di utenza tramite credenziali 2) Autorizzazione dell'operazione richiesta | | |
| Alternative Flow | | | |
| Exception Flow | Autorizzazione dell'operazione richiesta negata per esito negativo della verifica del livello di utenza | | |
| Response and Postconditions | Permesso di svolgere operazione | | |
| Non Functional Requirements | Autorizzazione, Autenticazione | | |
| Comments | | | |

Tabella 8 - Use Case 5

Misuse Case

| | | | |
|------------------------------------|--|----------------|---|
| Case Type | Misuse Case | Case ID | 1 |
| Case Name | Buffer Manipulation | | |
| Actors | Attaccanti, Utenti e Software | | |
| Description | Un avversario o un utente manipola (anche in modo non intenzionale) l'interazione del software con un buffer leggendo o modificando dati a cui non dovrebbe avere accesso. In quasi tutti i malfunzionamenti legati al buffer il contenuto che viene inserito nel buffer è irrilevante. Al contrario, il software sarà tanto più esposto a tale errore quanto più piccola sarà la dimensione scelta per il buffer. | | |
| Data | Dati sulle emissioni, informazioni di clienti,trasformatori e fornitori presenti nel buffer nel momento del malfunzionamento. | | |
| Stimulus and preconditions | L'avversario deve identificare un mezzo programmatico per interagire con un buffer previa autorizzazione. | | |
| Attack Flow 1 | Inserimento nel buffer una quantità di dati maggiore rispetto alla sua capacità. | | |
| Response and Postconditions | Esecuzione comandi non autorizzati Modifica dati immessi nel sistema dai produttori Lettura dati immessi nel sistema dai produttori | | |
| Non Functional Requirements | Dimensione limitata del buffer | | |
| Mitigations | Prima di iniziare lo sviluppo dell'applicazione, considerare l'utilizzo di un linguaggio di codice (ad es. Java) o di un compilatore che limiti la capacità degli sviluppatori di agire oltre i limiti di un buffer. Utilizzo di un buffer di dimensioni maggiori. | | |
| Comments | | | |

Tabella 9 - Misuse Case 1

| | | | |
|------------------------------------|---|----------------|---|
| Case Name | Misuse Case | Case ID | 2 |
| Case Name | Input Data Manipulation | | |
| Actors | Attaccanti, Utenti e Software | | |
| Description | Un utente, intenzionalmente o accidentalmente, sfrutta un punto debole nella convalida dell'input controllando il formato, la struttura e la composizione dei dati in un'interfaccia di elaborazione dell'input. Fornendo input di formato non standard o inaspettato, un utente può avere un impatto negativo sulla sicurezza del bersaglio. | | |
| Data | Tutti i dati contenuti nella memoria del software. | | |
| Stimulus and preconditions | L'interfaccia di input deve accettare i dati dell'utente per l'elaborazione e il modo in cui questi dati vengono elaborati deve dipendere da alcuni aspetti del formato o dei flag che l'attaccante può controllare. | | |
| Attack Flow 1 | Codificare diversamente i dati inseriti in input potrebbe far sì che essi vengano valutati come dati sicuri anziché pericolosi. | | |
| Attack Flow 2 | L'utente può utilizzare determinati flag, come le estensioni di file, per far credere al software che i dati forniti debbano essere gestiti utilizzando un determinato interprete quando i dati non sono effettivamente del tipo indicato dall'estensione fittizia. | | |
| Response and Postconditions | Bypassare i meccanismi di protezione; Costringere il software ad utilizzare componenti specifici per l'elaborazione dell'input; Dati inseriti dall'utente gestiti in modo diverso da quanto ci si potrebbe altrimenti aspettare. | | |
| Non Functional Requirements | - | | |
| Mitigations | Sistema efficiente di convalida dei dati in input. | | |
| Comments | | | |

Tabella 10 - Misuse Case 2

Abuse Case

| | | | |
|------------------------------------|---|----------------|----------|
| Case Type | Abuse Case | Case ID | 1 |
| Case Name | Interception / Adversary in the middle | | |
| Actors | Attaccanti, Utenti e Software | | |
| Description | Nella trasmissione di informazioni tra utenti (es. inserimento dei dati da parte dei produttori, lettura del certificato da parte dei clienti, ecc.) qualche malintenzionato potrebbe intercettare/manipolare dati sensibili. In tutte le varianti dell'attack flow l'avversario non è il destinatario previsto del flusso di dati. | | |
| Data | Dati sulle emissioni, informazioni personali di clienti,trasformatori e fornitori, certificati | | |
| Stimulus and preconditions | Esistenza di un trasferimento di dati tra produttori/clienti e software. Il bersaglio deve trasmettere i dati su un mezzo accessibile all'avversario. L'avversario deve possedere la tecnologia necessaria per intercettare le informazioni che passano tra i nodi di una rete. | | |
| Attack Flow 1 | L'attaccante può tentare di avviare la creazione di un flusso di dati esterno a quello legittimo per raccogliere informazioni sensibili | | |
| Attack Flow 2 | L'attaccante osserva passivamente le comunicazioni mentre si svolgono con l'eventuale scopo di supportare un ulteriore attacco futuro | | |
| Attack Flow 3 | Nel caso dell'avversario nel mezzo i dati passano prima attraverso l'avversario, che ha l'opportunità di osservarli o modificarli, prima di essere trasmessi al destinatario previsto come se non fossero mai stati osservati | | |
| Response and Postconditions | Fuga di dati sensibili e informazioni sullo stato degli attori attaccati | | |
| Non Functional Requirements | Confidentiality, Integrity, Authorization, Reliability | | |
| Mitigations | Utilizzo di un canale sicuro per lo scambio di informazioni; autenticazione a più fattori | | |
| Comments | Sono stati unificati in un unico caso i seguenti abuse case: Interception e Adversary in the middle. Questo perchè entrambi hanno in comune la struttura su cui poggia l'attacco. | | |

Tabella 11 - Abuse Case 1

| | | | |
|-----------------------------|--|---------|---|
| Case Type | Abuse Case | Case ID | 2 |
| Case Name | Exploitation of trusted identifiers | | |
| Actors | Attaccante, software. | | |
| Description | Un avversario indovina e/o ottiene un identificatore affidabile (es. ID sessione, ID risorsa, cookie, ecc.) per eseguire azioni autorizzate fingendosi un utente o servizio autenticato. | | |
| Data | Credenziali, ID sessione, ID risorsa, cookie, dati legati alle azioni eseguite dall'attaccante (emissioni o certificati). | | |
| Stimulus and preconditions | <p>Possibilità di distribuire software in rete; Capacità di comunicare in modo sincrono o asincrono con il server; Il software del server deve fare affidamento su schemi di prova e/o verifica dell'identificatore Gli identificatori devono avere una lunga durata e un potenziale di riutilizzabilità; Il software del server deve consentire l'esistenza di sessioni simultanee.</p> | | |
| Attack Flow 1 | L'attaccante esamina l'applicazione in cerca di indicatori di suscettibilità. Utilizza diversi metodi finché non ne trova uno che si applica al bersaglio: l'avversario cerca cookie, token di sessione o punti di ingresso che ignorano del tutto gli identificatori. | | |
| Attack Flow 2 | L'attaccante recupera molti campioni di identificatori. Ciò può avvenire tramite un accesso legittimo (login, connessioni legittime, ecc.) o tramite sondaggi sistematici. | | |
| Attack Flow 3 | Un avversario può utilizzare esperimenti o autenticazioni riusciti per impersonare un utente o un sistema autorizzato o per spostarsi lateralmente all'interno di un sistema o di un'applicazione. | | |
| Attack Flow 4 | Spoofing: i dati dannosi possono essere iniettati nel sistema di destinazione o nel sistema di un utente scelto come vittima da un avversario. L'avversario può anche spacciarsi per un utente legittimo per eseguire attacchi di ingegneria sociale. | | |
| Attack Flow 5 | L'attaccante può ottenere dati sensibili contenuti nel sistema o nell'applicazione attraverso l'esfiltrazione dei dati. | | |
| Response and Postconditions | <p>Ottenere dati sensibili; Scaricare/installare malware sul sistema; Presentarsi come un utente legittimo per scopi di ingegneria sociale.</p> | | |
| Non Functional Requirements | Authentication, Integrity, Confidentiality | | |
| Mitigations | <p>Crittografare e firmare i token di identità in transito; Utilizzare meccanismi di generazione della chiave di sessione standard del settore che utilizzano un'elevata quantità di entropia per generare la chiave di sessione; Se l'identificatore viene utilizzato per l'autenticazione, assicurarsi che sia protetto allo stesso livello di garanzia dei token di autenticazione; Utilizzare identificatori di sessione efficaci protetti in transito e inattivi; Utilizzare un timeout di sessione per tutte le sessioni; Verificare l'autenticità di tutti gli identificatori in fase di esecuzione.</p> | | |
| Comments | | | |

Tabella 12 - Abuse case 2

| | | | |
|-----------------------------|---|---------|---|
| Case Type | Abuse Case | Case ID | 3 |
| Case Name | Authentication bypass / Authentication abuse | | |
| Actors | Attaccante, software | | |
| Description | Un utente malintenzionato ottiene l'accesso al software con i privilegi di un utente autorizzato eludendo il meccanismo di autenticazione o aggirandolo senza mai autenticarsi, attraverso la conoscenza delle debolezze intrinseche del meccanismo stesso. | | |
| Data | Dati sulle emissioni, informazioni personali di clienti,trasformatori e fornitori | | |
| Stimulus and preconditions | Un meccanismo o sottosistema di autenticazione che implementa una qualche forma di autenticazione come password, autenticazione digest, certificati di sicurezza, ecc. | | |
| Attack Flow 1 | <p>Il modello di attacco di bypass dell'autenticazione differisce da altri attacchi di autenticazione in quanto gli attacchi di questo modello evitano completamente l'autenticazione, piuttosto che falsificare l'autenticazione sfruttando difetti o rubando credenziali da utenti legittimi.</p> <p>Ad esempio, un utente malintenzionato potrebbe essere in grado di raggiungere contenuti Web protetti inserendo esplicitamente il percorso del contenuto anziché facendo clic sul collegamento di autenticazione, evitando così del tutto il controllo.</p> | | |
| Attack Flow 2 | In un attacco di abuso di autenticazione vi è una sequenza di eventi, la quale fa sì che il meccanismo di autenticazione conceda l'accesso all'attaccante. Questo attacco può sfruttare le ipotesi fatte dalle procedure di autenticazione del bersaglio, come le ipotesi relative alle relazioni di fiducia o le ipotesi riguardanti la generazione di valori segreti. | | |
| Response and Postconditions | <p>In ogni versione dell'attacco, l'intruso ottiene privilegi che non gli appartengono e può ottenere dati sensibili, interagire col sistema al fine di danneggiarlo.</p> <p>Nel caso dell'abuso di autenticazione, l'attaccante può fare le veci di un utente terzo.</p> | | |
| Non Functional Requirements | Authorization, Reliability, Authentication | | |
| Mitigations | - | | |
| Comments | Sono stati unificati in un unico caso i seguenti abuse case: Authentication bypass e Authentication abuse. Questo perchè entrambi sfruttano una debolezza del sistema di autenticazione. | | |

Tabella 13 - Abuse Case 3

| | | | |
|----------------------------|--|---------|---|
| Case Type | Abuse Case | Case ID | 4 |
| Case Name | Privilege abuse / Privilege escalation | | |
| Actors | Attaccanti e Software | | |
| Description | <p>Il privilege abuse è un attacco nel quale l'avversario è in grado di sfruttare le funzionalità del target, le quali dovrebbero essere riservate a utenti o amministratori privilegiati. L'accesso a informazioni e funzionalità sensibili deve essere controllato per garantire che solo gli utenti autorizzati possano accedere a queste risorse. Se i meccanismi di controllo dell'accesso fossero assenti o configurati in modo errato, un utente potrebbe essere in grado di accedere a risorse destinate solo agli utenti di livello superiore. Un avversario potrebbe essere in grado di sfruttare questo per utilizzare un account meno attendibile per ottenere informazioni ed eseguire attività riservate ad account più attendibili. Questo attacco differisce dall'escalation dei privilegi in quanto l'avversario non aumenta mai effettivamente i propri privilegi, ma è invece in grado di utilizzare un grado inferiore di privilegi per accedere alle risorse che dovrebbero essere (ma non sono) riservate agli account con privilegi più elevati (tutte le funzionalità di controllo funzionano come configurate ma la configurazione non protegge adeguatamente le risorse sensibili a un livello appropriato).</p> | | |
| Data | <p>Informazioni sensibili degli utenti, dati sulle emissioni e funzionalità privilegiate (possibilità di effettuare operazione riservate ad un certo tipo di ruolo di utente: fornitore, trasformatore e/o cliente)</p> | | |
| Stimulus and preconditions | <p>Privilege abuse:</p> <p>1) L'obiettivo deve aver configurato in modo errato i propri meccanismi di controllo degli accessi in modo tale che le informazioni sensibili, che dovrebbero essere accessibili solo agli utenti più affidabili, rimangano accessibili agli utenti meno affidabili.</p> <p>2) L'avversario deve avere accesso al bersaglio, anche se con un account meno privilegiato di quanto sarebbe appropriato per le risorse mirate.</p> <p>Privilege escalation:</p> <p>1) L'esistenza di una debolezza che l'avversario sfrutta per elevare i suoi privilegi</p> | | |
| Attack Flow 1 | <p>Nel caso del "Privilege Abuse" l'attaccante dopo aver effettuato l'accesso al target sfrutta una debolezza nella configurazione. Infatti, le funzionalità di controllo sebbene funzionanti non sono state configurate in maniera adeguata da proteggere le risorse sensibili.</p> | | |
| Attack Flow 2 | <p>Anche nel "Privilege Escalation" l'avversario sfrutta una debolezza, ma in questo caso per aumentare i propri privilegi, consentendogli così di accedere a determinate funzionalità/informazioni che con i privilegi precedenti erano inaccessibili.</p> | | |

| | |
|------------------------------------|---|
| Response and Postconditions | In entrambi i casi una volta ottenuti determinati privilegi possono verificarsi le seguenti problematiche: <ul style="list-style-type: none"> - Modifica dati - Lettura dati - Esecuzione comandi non autorizzati |
| Non Functional Requirements | Authorization, Reliability, Authentication, Confidentiality |
| Mitigations | Privilege Abuse: Configurare i privilegi dell'account in modo tale che le funzionalità privilegiata/di amministratore non siano esposte agli account non privilegiati/inferiori. |
| Comments | Sono stati unificati in un unico caso i seguenti abuse case: Privilege abuse e Privilege escalation. Questo perchè entrambi hanno l'obiettivo di ottenere dei privilegi che non gli sarebbero concessi. |

Tabella 14 - Abuse Case 4

| | | | |
|-----------------------------|--|---------|---|
| Case Type | Abuse Case | Case ID | 5 |
| Case Name | File manipulation / Content spoofing / Contaminate resource | | |
| Actors | Attaccante e Software. | | |
| Description | <p>Nei primi due casi un utente malintenzionato modifica il contenuto dei file o gli attributi (come estensioni o nomi) dei file in modo da causare un'elaborazione errata da parte di un'applicazione. Tutto ciò mantenendo invariata la fonte apparente del contenuto.</p> <p>Nel caso della 'Contaminate resource' un avversario contamina i sistemi informativi organizzativi (inclusi dispositivi e reti) inducendoli a gestire informazioni di una classificazione/sensibilità per cui non sono stati autorizzati.</p> | | |
| Data | Qualsiasi dato contenuto o interagente con il sistema (dati sulle emissioni ma anche su tutto ciò che ruota intorno ai prodotti stessi oppure informazioni, personali e non, degli utenti del software) | | |
| Stimulus and preconditions | <p>File manipulation / Content spoofing: 1) Il target deve fornire contenuti ma non riesce a proteggerli adeguatamente contro la modifica.</p> <p>2) Se il contenuto deve essere modificato in transito, l'avversario deve essere in grado di intercettare i messaggi mirati.</p> <p>3) L'avversario deve avere i mezzi per alterare i dati ai quali non è autorizzato</p> | | |
| Attack Flow 1 | Gli aggressori utilizzano questa classe di attacchi per far entrare le applicazioni in stati instabili, sovrascrivere o esporre informazioni riservate e persino eseguire codice arbitrario con i privilegi dell'applicazione. | | |
| Attack Flow 2 | Il contenuto può essere modificato alla fonte (es. modificando il file sorgente per una pagina web) o in transito (es. intercettando e modificando un messaggio tra mittente e destinatario). | | |
| Response and Postconditions | Manipolazione dei dati, file di configurazione ecc... | | |
| Non Functional Requirements | Integrity, Confidentiality, Availability, Authorization, Safety, Reliability, Resilience | | |
| Mitigations | - | | |
| Comments | Sono stati unificati in un unico caso i seguenti abuse case: File manipulation, Content spoofing e Contaminate resource. Questo perchè entrambi hanno in comune il fatto di manipolare qualcosa (file di configurazione, dati o risorse informatiche) | | |

Tabella 15 - Abuse Case 5

| | | | |
|-----------------------------|--|---------|---|
| Case Type | Abuse Case | Case ID | 6 |
| Case Name | Identity spoofing | | |
| Actors | Attaccante, Utenti e Software. | | |
| Description | Si riferisce all'azione di assumere l'identità di un'altra entità (umana o non umana) e utilizzarla per raggiungere un obiettivo. Gli attacchi di Identity Spoofing non devono essere limitati ai messaggi trasmessi: qualsiasi risorsa associata a un'identità (ad esempio, un file con una firma) può essere l'obiettivo di un attacco in cui l'avversario tenta di modificare l'identità apparente. | | |
| Data | Qualsiasi risorsa associata ad un'identità (certificati in possesso di altri utenti o informazioni personali) | | |
| Stimulus and preconditions | L'identità associata al messaggio o alla risorsa deve essere rimovibile o modificabile in modo non rilevabile. | | |
| Attack Flow 1 | Un avversario può intercettare un messaggio da un mittente legittimo e tentare di far sembrare che il messaggio provenga da lui senza modificarne il contenuto | | |
| Attack Flow 2 | Un avversario può creare messaggi che sembrano provenire da un principio diverso o utilizzare credenziali di autenticazione rubate/falsificate. | | |
| Attack Flow 3 | L'ultima forma di questo attacco può essere utilizzata per dirottare le credenziali di utenti legittimi | | |
| Response and Postconditions | L'attaccante ottiene i privilegi di un determinato utente. L'attaccante, dunque, può utilizzare funzionalità che sono privilegiate/riservate solo a determinati attori del sistema (fornitore, trasformatore e/o cliente) | | |
| Non Functional Requirements | Authentication, Integrity, Confidentiality, Safety, Reliability | | |
| Mitigations | Impiegare processi di autenticazione robusti (ad es. autenticazione a più fattori). | | |
| Comments | | | |

Tabella 16 - Abuse Case 6

| | | | |
|----------------------------|---|---------|---|
| Case Type | Abuse Case | Case ID | 7 |
| Case Name | Use of Known Domain Credentials | | |
| Actors | Attaccanti, Utenti e Software | | |
| Description | Un avversario indovina o ottiene (vale a dire ruba o acquista) credenziali legittime (ad es. ID utente/password) per ottenere l'autenticazione e per eseguire azioni autorizzate sotto forma di utente o servizio autenticato. | | |
| Data | Credenziali e informazioni sensibili dell'utente | | |
| Stimulus and preconditions | <p>Il sistema/applicazione utilizza l'autenticazione basata su password a un fattore, SSO e/o l'autenticazione basata su cloud. Il sistema/applicazione non dispone di una politica password valida che viene applicata. Il sistema/applicazione non implementa un meccanismo di limitazione della password efficace. L'avversario possiede un elenco di account utente noti e password corrispondenti che possono esistere sul bersaglio. Un elenco di credenziali note. Uno script personalizzato che sfrutta l'elenco delle credenziali per lanciare un attacco.</p> | | |
| Attack Flow 1 | <p>Acquisire credenziali note: l'avversario deve ottenere credenziali note per accedere al sistema, all'applicazione o al servizio di destinazione. Tecniche:</p> <ul style="list-style-type: none"> - Un avversario acquista combinazioni di nome utente/password violate o password con hash trapelate dal dark web. - Un avversario sfrutta un keylogger o un attacco di phishing per rubare le credenziali dell'utente non appena vengono fornite. - Un avversario conduce un attacco di sniffing per rubare le credenziali mentre vengono trasmesse. - Un avversario ottiene l'accesso a un database ed esfiltra gli hash delle password. - Un avversario esamina i file di configurazione e proprietà rivolti verso l'esterno per scoprire le credenziali hardcoded. | | |
| Attack Flow 2 | <p>Determinare i criteri delle password di destinazione: Determinare i criteri delle password del sistema/applicazione di destinazione per determinare se le credenziali note rientrano nei criteri specificati. Tecniche:</p> <ul style="list-style-type: none"> - Determinare la lunghezza minima e massima consentita della password. - Determinare il formato delle password consentite (se devono o se possono contenere numeri, caratteri speciali, ecc., o se possono contenere parole del dizionario). - Determinare la politica di blocco dell'account (una rigorosa politica di blocco dell'account impedirà attacchi di forza bruta se sono note più password per un singolo account utente). | | |

| | |
|-----------------------------|--|
| Attack Flow 3 | Tentativo di autenticazione: prova ogni credenziale finché la destinazione non concede l'accesso. Tecniche: - Immettere manualmente o automaticamente ogni credenziale tramite l'interfaccia del target. |
| Attack Flow 4 | Impersonare: un avversario può utilizzare esperimenti o autenticazioni riusciti per impersonare un utente o un sistema autorizzato o per spostarsi lateralmente all'interno di un sistema o di un'applicazione |
| Attack Flow 5 | Spoofing: i dati dannosi possono essere iniettati nel sistema di destinazione o nel sistema di un utente vittima da un avversario. L'avversario può anche spacciarsi per un utente legittimo per eseguire attacchi di ingegneria sociale. |
| Attack Flow 6 | Esfiltrazione dei dati: l'avversario può ottenere dati sensibili contenuti nel sistema o nell'applicazione. |
| Response and Postconditions | Ottenimento privilegi Lettura dati (certificati, informazioni personali o sui prodotti ecc.) Modifica dati (certificati, informazioni personali o sui prodotti ecc.) |
| Non Functional Requirements | Authentication, Integrity, Confidentiality, Authorization |
| Mitigations | Sfrutta l'autenticazione a più fattori per tutti i servizi di autenticazione e prima di concedere a un'entità l'accesso alla rete del dominio. Crea una politica per le password complesse e assicurati che il tuo sistema applichi questa politica. Assicurati che gli utenti non riutilizzino combinazioni nome utente/password per più sistemi, applicazioni o servizi. Non riutilizzare le credenziali dell'account amministratore locale tra i sistemi. Nega l'uso remoto delle credenziali dell'amministratore locale per accedere ai sistemi di dominio. Non consentire agli account di essere un amministratore locale su più di un sistema. Implementa un meccanismo di limitazione delle password intelligente. Monitora i log di sistema e di dominio per l'accesso anomalo alle credenziali. |
| Comments | |

Tabella 17 - Abuse Case 7

Attack Tree

Con il fine di identificare tutti i vettori di attacco, per ogni abuse/misuse case è necessario valutare il rispettivo attack tree, il cui scopo è quello di rappresentare visivamente le minacce alla sicurezza in un modello di diramazione per determinare quali minacce sono più probabili e come bloccarle efficacemente.

Di seguito vengono riportati gli attack tree dell'intero sistema; per aumentare la leggibilità e facilitare la fruizione dello schema, esso è stato diviso in diverse figure, ognuna relativa ad una proprietà violata. Per gli stessi motivi di cui sopra, i task relativi ad un attacco vengono riportati una sola volta.

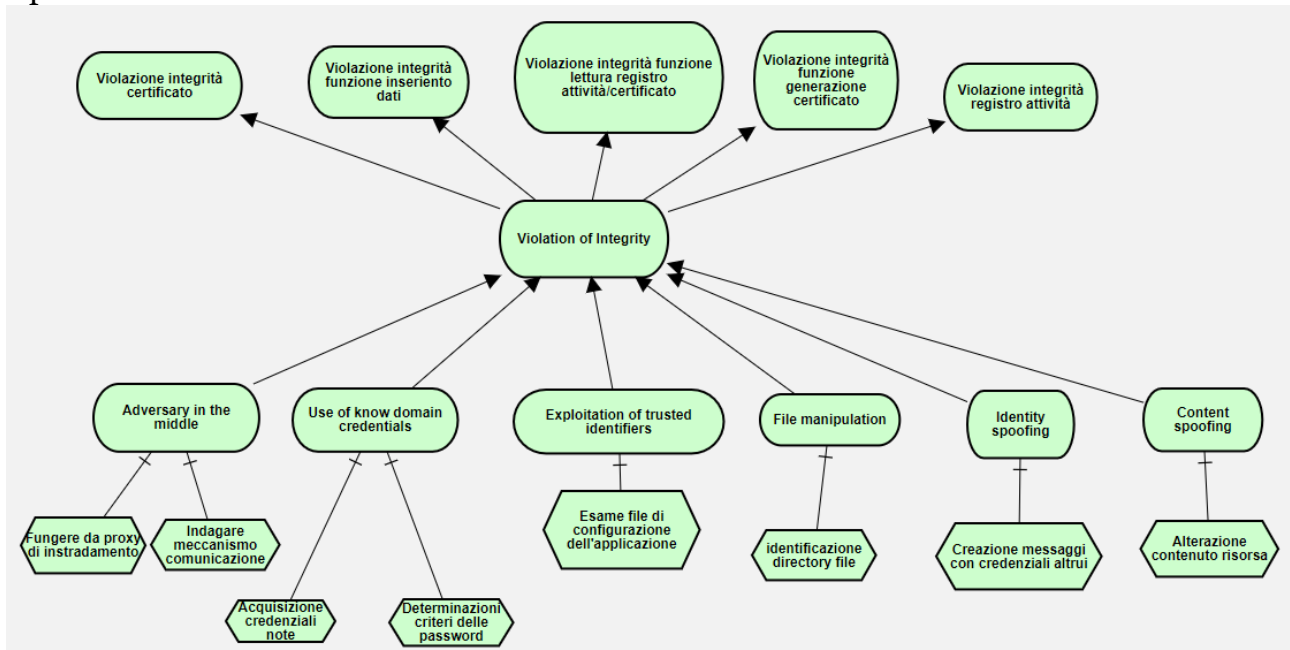


Figura 5 - Attack Tree Violation of Integrity

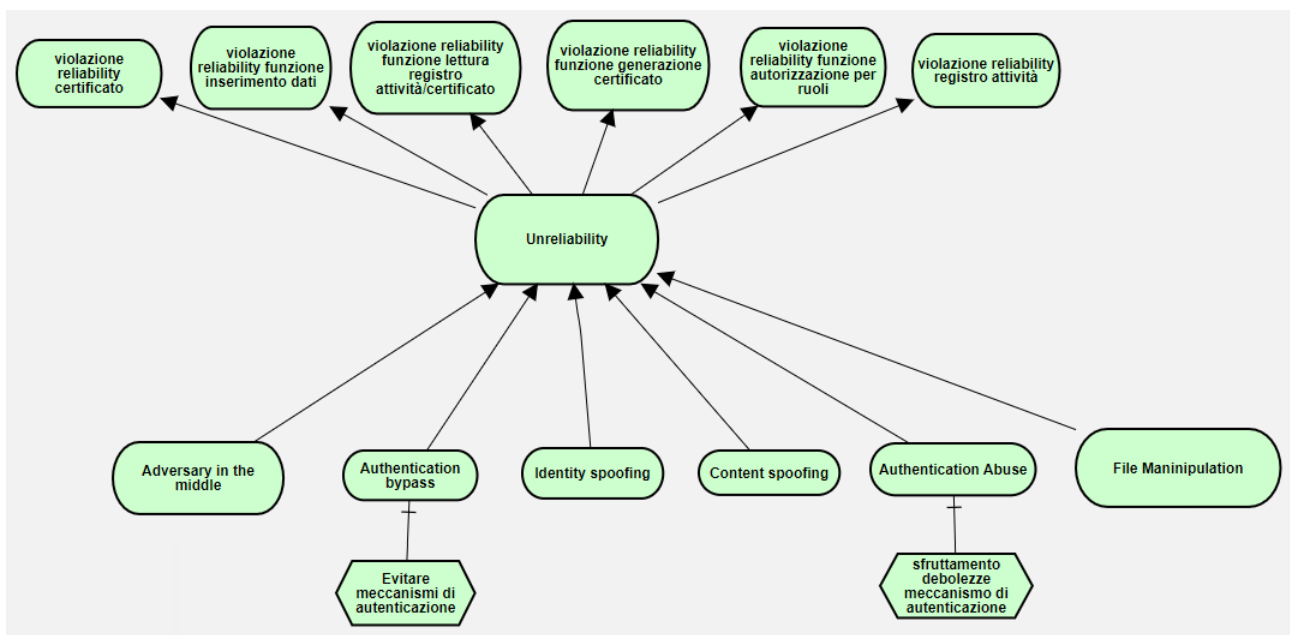


Figura 6 -Attack Tree Unreliability

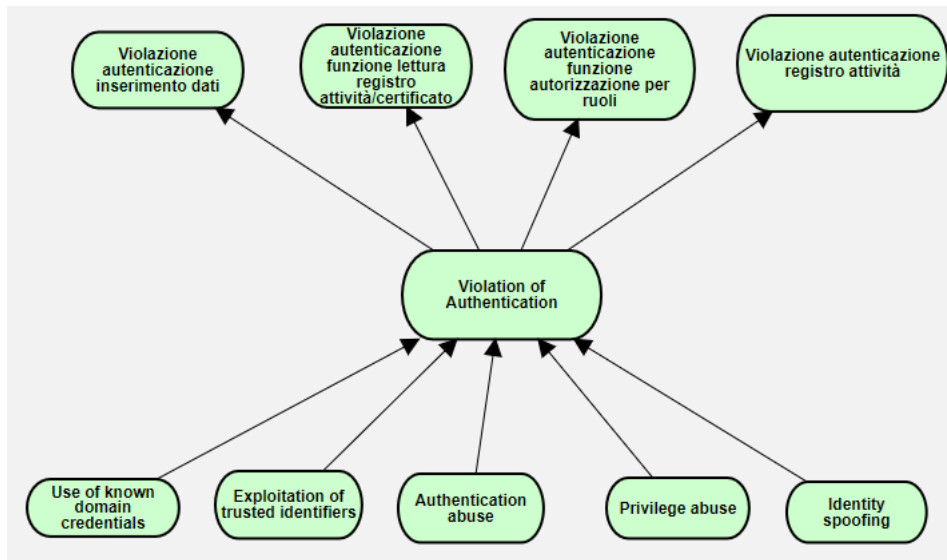


Figura 7 - Attack Tree Violation of Authentication

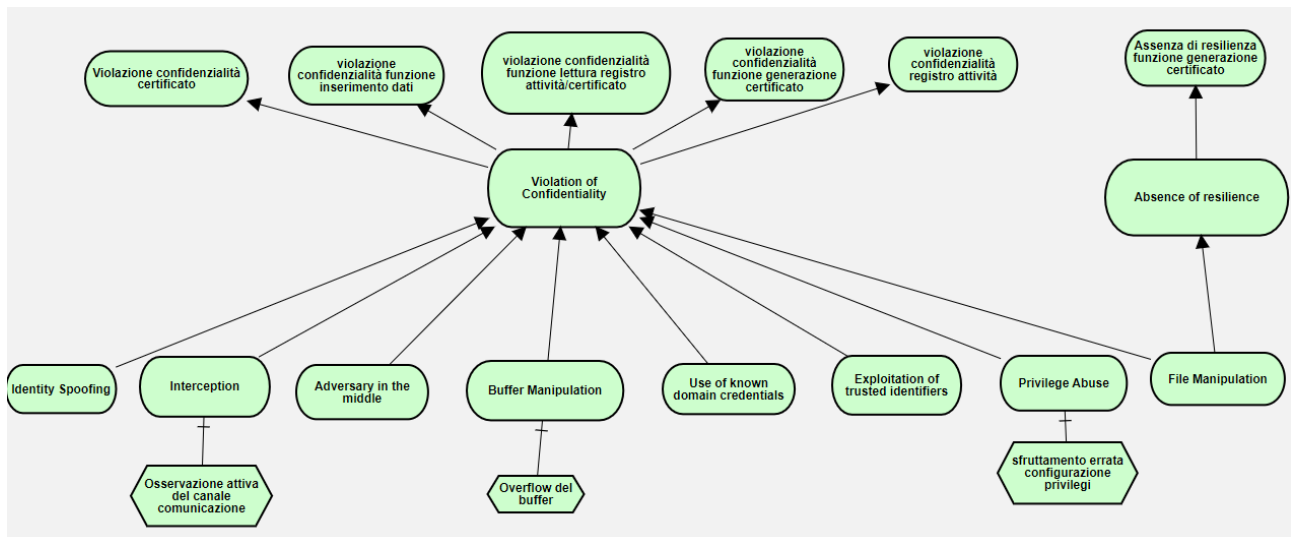


Figura 8 - Attack Tree Violation of Confidentiality, Absence of Resilience

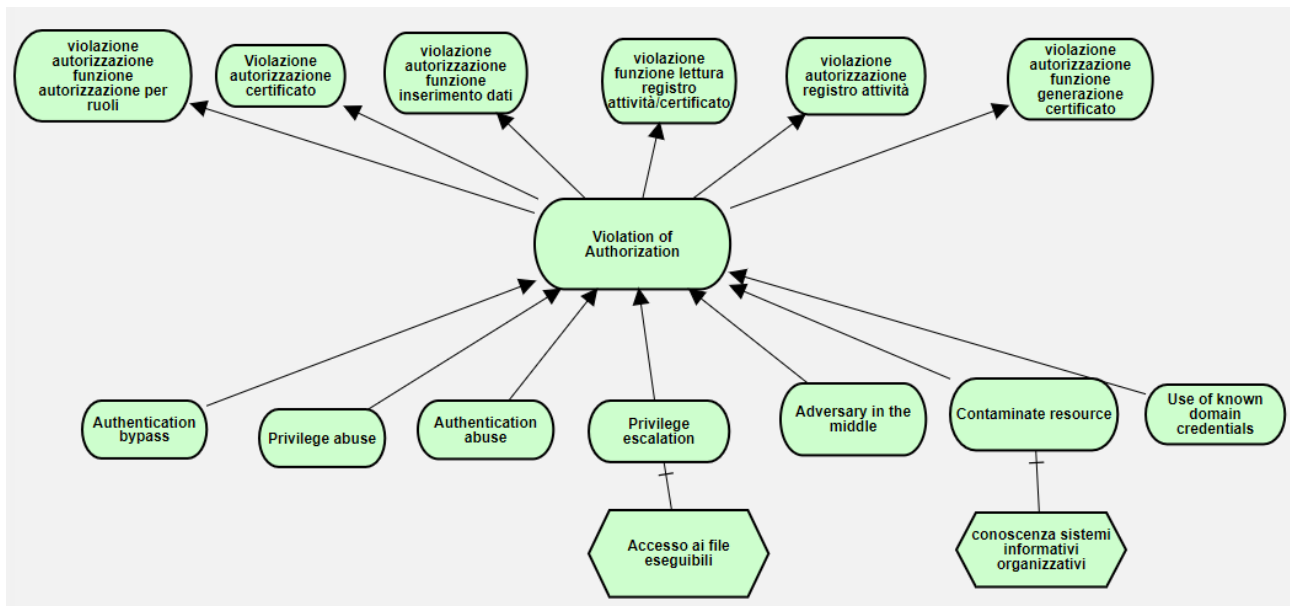


Figura 9 - Attack Tree Violation of Authorization

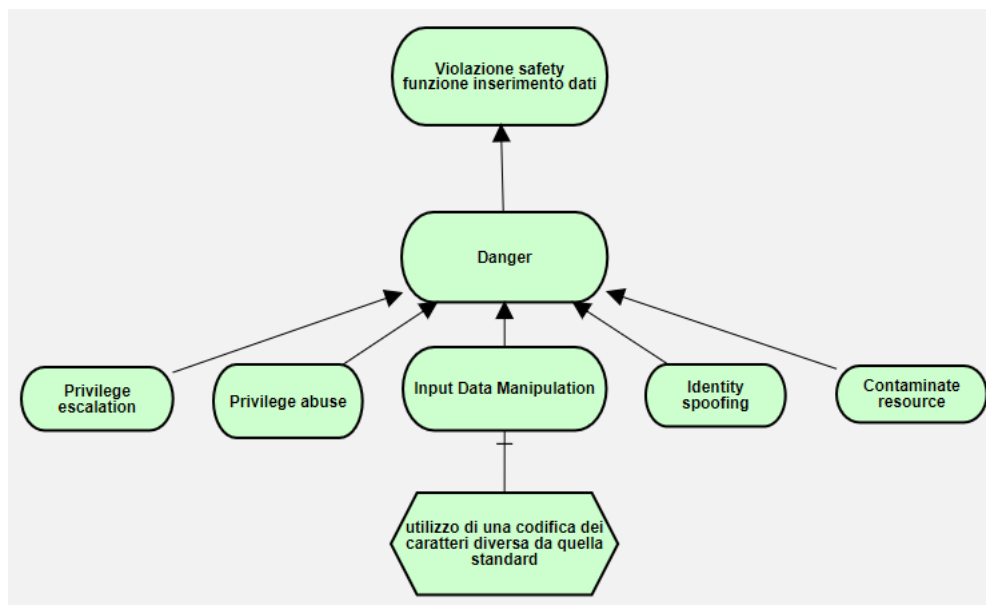


Figura 10 - Attack Tree Danger

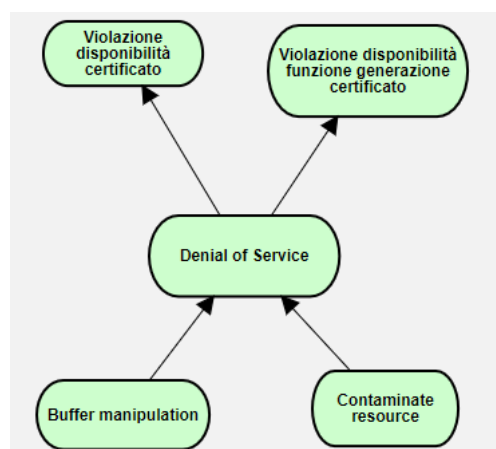


Figura 11 - Attack Tree Denial of Service

2.2.4 Risk Reduction

Control Identification

Riprendendo le metodologie STRIDE, in corrispondenza di ogni obiettivo della policy di sicurezza si trova la sua minaccia. Occorre quindi valutare come mitigare queste possibili minacce, utilizzando, come nella sezione precedente, il catalogo CAPEC. Dato un asset e il suo relativo obiettivo di sicurezza, possono essere valutate le varie tecniche utili a mitigare una determinata minaccia. In definitiva, per ogni vettore d'attacco è necessario identificare le opportune tecniche di mitigazione da adottare.

Feasibility Assessment

A questo punto della progettazione, per ogni tecnica di mitigazione occorre valutare diversi fattori:

- *overall cost*: utilizzando la scala di Likert;
- *feasibility*: ossia quanto è fattibile il tipo di tecnica di mitigazione dei vari vettori d'attacco;
- *residual probability*;
- *residual impact*;
- *residual risk*: rischio residuo dell'azione di mitigazione.

Nella tabella sottostante vengono riportati i risultati ottenuti attuando quanto descritto nei due paragrafi precedenti.

| Asset | Value | Spoofing | Impersonation | Reputation | Information disclosure | OS | Elevation of privilege | Danger | Unreliability | Absence of Resilience | Exposure | Attack | Inherent Probability | Inherent Risk | Control | Cost (1-7) | Feasibility | Residual Probability | Residual Impact | Residual Risk |
|---------------------------------------|-------|----------|---------------|------------|------------------------|----|------------------------|--------|---------------|-----------------------|-----------|---|----------------------|---------------|--|--------------------------------------|--|----------------------|-----------------|---------------|
| Certificato | 7 | | | | x | | | | | | Medium | CAPEC-117: Interception | Low | Low | Crittografia per codificare la trasmissione Adozione block-chain; | 6 | Fattibile ma prestare comunque attenzione nella trasmissione | Low | Medium | Low |
| | | | x | | x | | x | | x | | Very High | CAPEC-94: Adversary in the Middle | High | Very High | Chiavi pubbliche referenziate; Criptare comunicazione; Autenticazione reciproca forte; Canale sicuro per scambio chiavi; Adozione block-chain; | 6 6 6 6 | Tecnicamente tutte le soluzioni sono fattibili ma attenzione alla distribuzione del certificato | Medium | Very High | Very High |
| | | | | | x | x | | | | | Very High | CAPEC-123: Buffer Manipulation | High | Very High | Divieto di agire oltre limiti del buffer | 5 | Di facile implementazione | Medium | Very High | Very High |
| Inserimento dati | 5 | | | | | | | x | | | Medium | CAPEC-153: Input Data Manipulation | High | Medium | Validare formato dati input | 5 | Di facile implementazione | Medium | Medium | Medium |
| | | | x | | x | | x | | x | | Very High | CAPEC-94: Adversary in the Middle | High | Very High | Chiavi pubbliche referenziate; Criptare comunicazione; Autenticazione reciproca forte; Canale sicuro per scambio chiavi; Adozione block-chain; | 6 6 6 6 | Tecnicamente tutte le soluzioni sono fattibili ma attenzione alla distribuzione del certificato | Medium | Very High | Very High |
| | | x | x | | x | | x | | | | High | CAPEC-560: Use of Known Domain Credentials | High | High | Autenticazione a più fattori; Policy per password; Evitare riuso nome utente/password utenti per più servizi; Non riutilizzare le credenziali dell'account amministratore locale tra i sistemi; Nega l'uso remoto delle credenziali dell'amministratore locale per accedere ai sistemi di dominio; Non consentire agli account di essere un amministratore locale su più di un sistema; Implementazione meccanismo di limitazione delle password intelligente; Monitoraggio log di sistema e di dominio per l'accesso anomalo alle credenziali; | 7 4 4 4 4 5 5 6 | Tutte le soluzioni di mitigazione sono facilmente realizzabili, l'autenticazione a più fattori e il monitoraggio dei log richiedono l'impiego di tecnologie più complesse | Medium | High | High |
| Lettura registro attività/certificato | 6 | x | x | | x | | | | | | High | CAPEC-21: Exploitation of Trusted Identifiers | High | High | Implementazione meccanismi di generazione della chiave di sessione standard ad elevata entropia; La crittografia e/o la firma dell'identificatore (come il cookie) può proteggere l'ID se intercettato; Utilizzo identificatori di sessione efficaci protetti in transito e inattivi; Utilizzo timeout di sessione; Verifica l'autenticità di tutti gli identificatori in fase di esecuzione; | 6 6 5 5 5 | Tecnicamente tutte le soluzioni sono fattibili ma attenzione alla distribuzione del certificato | Medium | High | High |
| | | | x | | x | | x | | x | | Very High | CAPEC-94: Adversary in the Middle | High | Very High | Chiavi pubbliche referenziate; Criptare comunicazione; Autenticazione reciproca forte; Canale sicuro per scambio chiavi; Adozione block-chain; | 6 6 6 6 | Tecnicamente tutte le soluzioni sono fattibili ma attenzione alla distribuzione del certificato | Medium | Very High | Very High |
| | | x | x | | x | | x | | | | High | CAPEC-560: Use of Known Domain Credentials | High | High | Autenticazione a più fattori; Policy per password; Evitare riuso nome utente/password utenti per più servizi; Non riutilizzare le credenziali dell'account amministratore locale tra i sistemi; Nega l'uso remoto delle credenziali dell'amministratore locale per accedere ai sistemi di dominio; Non consentire agli account di essere un amministratore locale su più di un sistema; Implementazione meccanismo di limitazione delle password intelligente; Monitoraggio log di sistema e di dominio per l'accesso anomalo alle credenziali; | 7 4 4 4 4 5 5 6 | Tutte le soluzioni di mitigazione sono facilmente realizzabili, l'autenticazione a più fattori e il monitoraggio dei log richiedono l'impiego di tecnologie più complesse | Medium | High | High |
| Autorizzazione per ruoli | 7 | x | | | | | x | | x | | Medium | CAPEC-115: Authentication Bypass | High | Medium | - | - | - | | | |
| | | | | | | | x | | x | | Medium | CAPEC-114: Authentication Abuse | High | Medium | - | - | - | | | |
| | | x | | | x | | x | x | | | Medium | CAPEC-122: Privilege Abuse | High | Medium | Configurare i privilegi dell'account in modo tale che i privilegi dell'amministratore non siano esposti agli account non privilegiati | 7 | Tra le soluzioni più difficili da implementare visti i molteplici attacchi e le conseguenze che può subire | Medium | Medium | Medium |
| Generazione certificato | 6 | | x | | x | | x | x | | x | Medium | CAPEC-233: Privilege Escalation | High | Medium | - | - | - | | | |
| | | | | | | x | x | x | | | Medium | CAPEC-165: File Manipulation | High | Medium | Adozione block-chain | - | - | | | |
| | | | | | | | | | | | High | CAPEC-548: Contaminate Resource | High | High | Adozione block-chain | - | - | | | |
| Registro attività | 6 | | | | x | | | | | | Medium | CAPEC-117: Interception | Low | Low | Crittografia per codificare la trasmissione Adozione block-chain; | 6 | Fattibile ma prestare comunque attenzione nella trasmissione Tra le soluzioni più difficili da implementare visti i molteplici attacchi e le conseguenze che può subire | Low | Medium | Low |
| | | x | x | | x | | | x | x | | Medium | CAPEC-151: Identity spoofing | Low | Medium | Impiego processi di autenticazione robusti | 7 | | Low | Medium | Low |
| | | | | | | | x | | x | | Medium | CAPEC-115: Authentication Bypass | Low | Medium | - | - | - | Low | Medium | Low |
| | | x | | | | | | | x | | Medium | CAPEC-148: Content Spoofing | High | Medium | Adozione block-chain | - | - | Medium | Medium | Medium |

Tabella 18 - Risk Reduction

3. Design

3.1 Secure design

In questa sezione verranno analizzati i requisiti del sistema e, in seguito, si considereranno le componenti software in grado di garantire un sistema sicuro che rispetti le specifiche di cui sopra.

Tale scelta sarà guidata da due linee guida: la riduzione delle vulnerabilità, attraverso l'adozione di architetture di protezione, e la riduzione dell'impatto attraverso strategie di ridondanza, diversità e distribuzione.

Naturalmente nella selezione delle componenti occorre considerare un compromesso tra prestazioni e costi.

In letteratura esiste molta documentazione relativa alla progettazione del software sicuro. Per il progetto in esame in tale documento, sono state prese in considerazione le linee guida emesse da OWASP e da Sommerville.

Analizzando la sezione “Control” riportata in figura nel paragrafo “Feasibility Assessment”, si può notare come la principale soluzione riportata da adottare per proteggere i vari asset è senza dubbio l'adozione di una blockchain per la gestione del core del sistema.

Di seguito sono riportate le motivazioni di tale scelta attraverso l'argomentazione delle principali linee guida emesse da OWASP e Sommerville:

- *Establish secure defaults*: imponendo delle impostazioni sicure di default, risulterebbe difficile per un utente finale modificare le impostazioni di sicurezza del sistema; sarà prevista inoltre un'interfaccia che impedirebbe guidi l'accesso al sistema da parte dell'utente finale.
- *Log user actions*: la tecnologia scelta tiene traccia di tutte le transazioni eseguite dagli utenti.
- *Use redundancy and diversity to reduce risk*: la blockchain si basa sulla distribuzione delle informazioni su diversi nodi, garantendo la ridondanza e la diversità.
- *Specify the format of all system inputs*: verrà previsto, nel front-end un meccanismo di sanificazione degli ingressi da anteporre alla blockchain.
- *Design for recoverability*: in caso di attacco, grazie all'uso dei log, sarà possibile ripristinare lo stato del sistema ad una condizione immediatamente precedente al momento dell'attacco.
- *Least privilege*: la tecnologia scelta permette di creare account associando ad ognuno di essi un ruolo, sarà compito delle condizioni di sicurezza poste nella parte di back-end di monitorare il corretto accesso alle funzionalità del sistema.
- *Defense in depth*: la blockchain implementa meccanismi di difesa aggiuntivi come funzioni hash, ovvero primitive crittografiche in grado di aumentare la sicurezza generale del sistema.
- *Fail securely*: i meccanismi che la blockchain utilizza per gestire le transazioni fanno sì che in caso di fallimento la transazione non viene eseguita senza quindi mandare in blocco il sistema; verranno inoltre previsti dei meccanismi di gestione delle eccezioni per poter rendere ancora più sicuro una eventuale situazione di errore da parte del sistema.

- *Avoid single point of failure*: l'utilizzo della blockchain riduce il rischio di single point of failure in quanto è costituita da più nodi decentralizzati.
- *Avoid security by obscurity*: essendo la tecnologia scelta di tipo open design, l'intera comunità che ne farà utilizzo potrà constatare la presenza di eventuali errori e comunicarli agli sviluppatori del sistema con il fine di correggerli prima che possano essere sfruttati da utenti malintenzionati.

4. Implementazione

L'intero sistema è divisibile in due sezioni distinte che comunicano opportunamente tra loro:

- Back-end: costituita da una blockchain;
- Front-end: interfaccia da terminale.

Entrambe le sezioni verranno analizzate nel dettaglio nei paragrafi seguenti.

4.1 Blockchain

In questa sezione viene approfondita la parte back-end del sistema in analisi in tale elaborato.

4.1.1 Quorum

La blockchain scelta per il progetto è Quorum. Essa si basa sulla nota blockchain pubblica Ethereum ed è rilasciata sotto licenza open-source. Quorum fornisce meccanismi di sicurezza elevati, scrittura di transazioni private, vari algoritmi di consenso e una community di developers.

Nel caso specifico di tale sistema l'algoritmo utilizzato è di tipo RAFT, esso appartiene alla tipologia *Fault tollerant* e prevede una percentuale di consensi per l'approvazione di una transazione pari al 50% +1 dei miners votanti.

Il sistema è costituito da 3 nodi e ad ognuno di essi è associato un wallet relativo ad un account utente. Per comodità, come scelta progettuale, ognuno dei tre wallet si riferisce ad una tipologia diversa di utente, nello specifico:

- Produttore;
- Trasformatore;
- Cliente.

Tale scelta non è limitante per il sistema, in quanto ogni funzione è stata implementata in modo tale che sia scalabile ad un numero maggiore di utenti.

Il modello appena descritto è stato inizializzato a partire dal progetto di J.P. Morgan, reperibile al seguente link: <http://docs.goquorum.com/en/latest/Wizard/GettingStarted/>. Esso è utilizzabile tramite Quorum Wizard, strumento di configurazione di blockchain, realizzato dalla stessa azienda.

4.1.2 Smart Contracts

Gli smart contract sono programmi che regolano il comportamento degli account all'interno della blockchain. In questa sezione verranno approfonditi tali contratti.

Per la realizzazione di quest'ultimi è stato utilizzato Solidity: un linguaggio di alto livello orientato agli oggetti per l'implementazione di smart contract.

Solidity, tra le altre caratteristiche, è tipizzato in modo statico, supporta ereditarietà, librerie e tipi complessi definiti dall'utente.

Nel caso in esame sono stati definiti due smart contract:

- CarbonFootprint;
- NFT_Footprint.

CarbonFootprint

Questo smart contract è alla base del funzionamento dell'intera applicazione. Infatti, esso regola le transazioni che prevedono l'inserimento e la trasformazione delle materie prime e l'acquisto dei prodotti finiti.

La logica che vi è dietro tali meccanismi si basa sull'organizzazione delle materie prime e dei prodotti in lotti. Un lotto è un insieme di elementi dello stesso tipo, prodotti dalla stessa azienda, con lo stesso processo, che condividono lo stesso footprint.

Nello specifico:

- Sia le materie prime che i prodotti finiti e non, devono essere immessi nel sistema, dai produttori, organizzati in lotti.
- I trasformatori possono acquistare lotti di materie prime o di altri prodotti già lavorati necessari al loro processo di trasformazione. Tale processo, però, potrà interessare anche delle singole unità appartenenti ai lotti acquistati.
- I clienti finali possono decidere di acquistare singole unità di materie prime o di prodotti trasformati.

Scendendo nel dettaglio dell'implementazione, CarbonFootprint.sol contiene al suo interno diverse strutture atte a memorizzare lo stato dei lotti presenti nel sistema.

```
// LOTTI DI MATERIE PRIME E DI PRODOTTI
struct Lot {
    uint id;
    string name;
    uint carbonfootprint;
    uint amount;
    uint residual_amount;
    bool sold;
    address owner;
}
```

Figura 12 - Struttura Lot CarbonFootprint

```
// MEMORIZZAZIONE LOTTI PER ID E PER MATERIA PRIMA
mapping(uint => Lot) private getLotByID;
mapping(string => uint[]) private getLotByRawMaterialName;

// MEMORIZZAZIONE LOTTI PER PROPRIETARIO (TRASFORMATORE)
mapping(address => uint[]) private getLotByAddress;
```

Figura 13 - Mapping CarbonFootprint

Nello specifico:

- “*Lot*”: è una struttura dati che descrive lo stato di un singolo lotto tenendo traccia del relativo codice univoco, del nome, del footprint totale (footprint di ogni unità moltiplicato per la quantità del lotto), del numero di unità totali, del numero di unità disponibili, se è stato acquistato, dell’indirizzo del wallet del proprietario.
- Mapping: vi sono diverse tipologie di mapping che implementano un registro dei vari lotti accessibile a partire da determinate caratteristiche ad essi associate come codice univoco (“*getLotByID*”), nome della materia prima contenuta nel lotto (“*getLotByRawMaterialName*”), indirizzo wallet del proprietario (“*getLotByAddress*”).

NFT_FootPrint

NFT_FootPrint è lo smart contract dedito al conio degli NFT che fungeranno da certificazione del valore del footprint di ogni singola unità di prodotto acquistata dal cliente. Tali token sono implementati mediante la libreria OpenZeppelin che si fa riferimento allo standard ERC-721 proprio di Ethereum.

La logica per cui vengono emessi tali certificati si basa sull’acquisto di una singola unità di prodotto: il cliente può verificare la quantità di footprint associata al bene che desidera acquistare attraverso l’interfaccia front-end e, una volta acquistato l’articolo, riceverà un NFT che attesterà l’effettiva correttezza del dato controllato in precedenza.

Il proprietario del token generato in seguito all’acquisto sarà il cliente che ha richiesto tale transazione.

Le strutture dati che permettono di gestire i token sono due:

- *Struttura “Product”*: descrive un singolo certificato tenendo traccia di nome del prodotto, footprint associato all’articolo, id del lotto di provenienza;
- *Mapping “Attributes”*: fornendo in ingresso l’id del token restituisce l’oggetto di tipo Product corrispondente.

Entrambe sono riportate nella figura sottostante.

```
mapping(uint => Product) public attributes;

struct Product {
    string name;
    uint footprint;
    uint product_lot;
}
```

Figura 14 - Strutture dati NFT_Footprint

All’atto dell’acquisto di un prodotto da parte di un cliente, lo smart contract CarbonFootprint (analizzato nel paragrafo precedente) viene invocata la funzione “mint” (riportata nella figura sottostante). Tale metodo conia un NFT utilizzando le funzioni proprie di ERC721, in particolare “_safeMint”.

```
function mint(address _to, string memory _name, uint _footprint, uint _product_lot) public returns(uint) {
    uint256 tokenId = _tokenIdCounter.current();
    _tokenIdCounter.increment();
    _safeMint(_to, tokenId);
    attributes[tokenId] = Product(_name, _footprint, _product_lot);
    return tokenId;
}
```

Figura 15 - Funzione "mint" NFT_Footprint

In aggiunta a quanto già detto, è stato effettuato l’override del metodo “tokenURI” che associa ad ogni certificato il proprio URI.

Nello specifico viene costruito un JSON contenente gli attributi del token.

```
function tokenURI(uint256 tokenId) override(ERC721) public view returns (string memory) {
    string memory json = string(
        abi.encodePacked(
            '{',
            ' "id" : "', uint2str(tokenId), '",',
            ' "attributes" : [',
            '  { "name" : "', attributes[tokenId].name, ' ",',
            '  { "footprint" : "', uint2str(attributes[tokenId].footprint), ' ",',
            '  { "product_lot" : "', uint2str(attributes[tokenId].product_lot), ' " }',
            ']',
            '}'
        )
    );
    return json;
}
```

Figura 16 - Metodo "tokenURI" NFT_Footprint

4.3 Interfaccia utente

In tale sezione viene analizzata nel dettaglio l'implementazione dell'interfaccia che permetterà agli utenti di interagire con il sistema in maniera corretta.

4.3.1 JavaScript, Node.js e web3.js

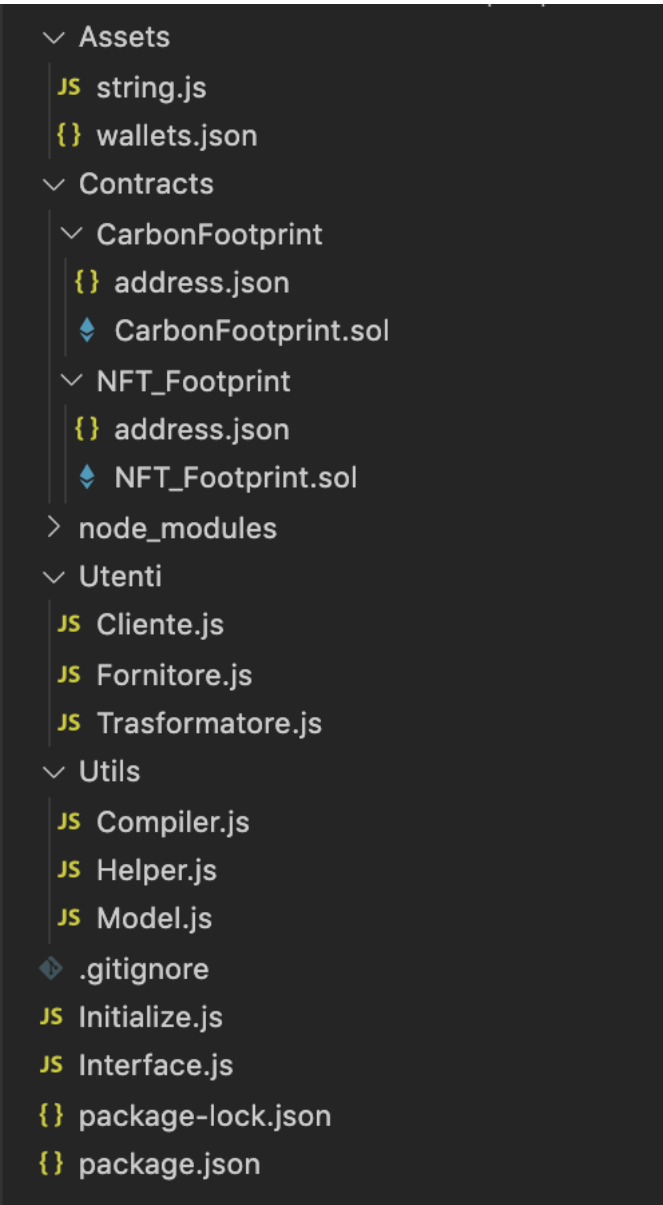
Il JavaScript è un linguaggio di programmazione multi-paradigma orientato agli eventi, comunemente utilizzato nella programmazione Web lato client per la creazione, in siti web e applicazioni web, di effetti dinamici interattivi tramite funzioni di script invocate da eventi innescati a loro volta in vari modi dall'utente sulla pagina web in uso.

Nel caso in esame in questo elaborato è stata implementata un'interfaccia da riga di comando che sfrutta il runtime system *Node.js*.

Esso, infatti, consente, tra le altre cose, di eseguire codice JavaScript lato server permettendo all'utente di interfacciarsi con il sistema direttamente da riga di comando.

Per collegarsi alla blockchain si fa utilizzo della Ethereum JavaScript API (*web3.js*) che mette a disposizione metodi per effettuare deploy di contratti, transazioni e tutte le altre operazioni necessarie alla gestione della blockchain.

4.4 Struttura progetto



```

  ▾ Assets
    JS string.js
    {} wallets.json
  ▾ Contracts
    ▾ CarbonFootprint
      {} address.json
      💎 CarbonFootprint.sol
    ▾ NFT_Footprint
      {} address.json
      💎 NFT_Footprint.sol
  > node_modules
  ▾ Utenti
    JS Cliente.js
    JS Fornitore.js
    JS Trasformatore.js
  ▾ Utils
    JS Compiler.js
    JS Helper.js
    JS Model.js
  .gitignore
  JS Initialize.js
  JS Interface.js
  {} package-lock.json
  {} package.json

```

The image shows a file explorer interface with a dark background. It displays a hierarchical project structure. The 'Assets' folder contains 'string.js' and 'wallets.json'. The 'Contracts' folder contains two subfolders: 'CarbonFootprint' and 'NFT_Footprint'. 'CarbonFootprint' contains 'address.json' and 'CarbonFootprint.sol'. 'NFT_Footprint' contains 'address.json' and 'NFT_Footprint.sol'. There is a 'node_modules' folder, an 'Utenti' folder containing 'Cliente.js', 'Fornitore.js', and 'Trasformatore.js', and a 'Utils' folder containing 'Compiler.js', 'Helper.js', and 'Model.js'. At the bottom level, there are files: '.gitignore', 'Initialize.js', 'Interface.js', 'package-lock.json', and 'package.json'.

Figura 17 - Struttura progetto

All'interno della directory principale di progetto "CYBERSECURITY-PROJECT" vi sono diverse sottocartelle analizzate nel dettaglio di seguito:

- *Assets*
 - *string.js*: Libreria javascript che contiene tutte le stringhe che verranno mostrate all'utente attraverso l'interfaccia. Ciò permette una buona manutenibilità del codice e dà la possibilità di cambiare la lingua del software modificando un solo file all'interno dell'intero sistema.
 - *wallet.json*: File json, generato in seguito all'invocazione dello script "*Initialize.js*", contenente gli indirizzi dei wallet associati a tutti i nodi della blockchain.
- *Contracts*
 - *CarbonFootprint*:
 - *CarbonFootprint.sol*: Smart contract analizzato nel dettaglio nel paragrafo "4.1.2 Smart Contracts / CarbonFootprint".
 - *address.json*: Contiene l'indirizzo esadecimale dello smart contract "*CarbonFootprint.sol*" grazie al quale, assieme all'ABI, si può risalire all'istanza del contratto stesso.
 - *NFT_Footprint*:
 - *NFT_Footprint.sol*: Smart contract analizzato nel dettaglio nel paragrafo "4.1.2 Smart Contracts / NFT_Footprint".
 - *address.json*: Contiene l'indirizzo esadecimale dello smart contract "*NFT_Footprint.sol*" (per gli stessi motivi di cui sopra).
- *node_modules*: directory contenente le librerie scaricate da "*npm*", ossia il package manager di node.js.
- *package.json*: Riassume i pacchetti necessari all'esecuzione dell'applicativo; "*npm*" consulterà questo file per valutare quali pacchetti è necessario installare/aggiornare.
- *Initialize.js*: Script necessario ad inizializzare la blockchain, si collega ad essa salvando gli indirizzi dei wallet associati ad i vari nodi nel file "*wallets.json*", compila e lancia il deploy di entrambi i contratti.
- *Interface.js*: Script che avvia l'interfaccia utente, ed in base al wallet selezionato, invoca lo script corrispondente al ruolo associato al wallet stesso.
- *Utenti*
 - *Cliente.js*: Script che gestisce tutte le transazioni che possono essere richieste da un utente con ruolo "Cliente".
 - *Fornitore.js*: Script che rende possibili le operazioni associate al ruolo "Fornitore".
 - *Trasformatore.js*: Script che permette agli account con associato il ruolo "Trasformatore" di compiere le relative azioni.
- *Utils*
 - *Compiler.js*: Script che, invocando i metodi del compilatore "*solc*" di "*node.js*", permette la compilazione degli smart contract.
 - *Helper.js*: Script contenente le funzioni comuni ai diversi utenti.
 - *Model.js*: Script contenente i metodi necessari per interfacciarsi con i contract.

5. Guida all'utilizzo

È stato creato un repository GitHub nel quale sono presenti tutti i file necessari all'installazione e testing del progetto:

<https://github.com/Me77y99/CyberSecurity-project> .

Sullo stesso repository, all'interno del file README.md è presente una guida, scritta con linguaggio Mark Down, qui sotto riportata per completezza.

Guida d'utilizzo

Progetto Software Cybersecurity a.a 2021/2022

Ali Waqar - Angelini - Di Silvestre - Scuriatti

Nota - La seguente procedura è da intendersi per ambiente macOS o Linux

Installazione Quorum Wizard

Seguire le istruzioni del seguente link per ottenere una blockchain privata con 3 nodi: <https://github.com/ConsenSys/quorum-wizard>

Installazioni pacchetti npm

Scaricare **Node.js** dalla pagina ufficiale: <https://nodejs.org/it/download/> Clonare la repository "CyberSecurity-project" e, al suo interno, eseguire il comando

```
$ npm install
```

che installerà i pacchetti necessari al funzionamento del software:

- web3
- solc
- quorum-js
- inquirer
- console-table-printer
- @openzeppelin/contracts

Avvio del progetto

Entrare da terminale nella cartella creata da Quorum Wizard e lanciare lo script di avvio:

```
$ cd network/3-nodes-quickstart
$ ./start.sh
```

Se l'operazione ha esito positivo, entrare nella cartella clonata in precedenza e, solo al primo avvio, eseguire il comando:

```
$ cd CyberSecurity-project
$ node ./Initialize.js
```

Se l'operazione ha esito positivo, per avviare l'interfaccia eseguire il comando:

```
$ node ./Interface.js
```

Test del progetto

Selezionare il wallet al quale si desidera accedere tenendo conto del fatto che ad ognuno di essi è associato un ruolo diverso, rispettivamente:

- Fornitore
- Trasformatore
- Cliente

```
? SELEZIONA UN WALLET (Use arrow keys)
> 0xed9d02e382b34818e88888a309c7fe71E65f419d
0xcA843569e3427144cEad5e4d5999a3D0cCF92B8e
0x0fBDc686b912d7722dc86510934589E0AAf3b55A
EXIT
```

Fornitore

Selezionando il wallet associato al Fornitore sarà possibile scegliere tra le seguenti operazioni:

```
? MENU' FORNITORE (Use arrow keys)
> INSERIMENTO DI MATERIE PRIME
VISUALIZZA LOTTI DI TUA PROPRIETA'
RICERCA MATERIA PRIMA
RICERCA LOTTO
BACK
EXIT
```

Trasformatore

Selezionando il wallet associato al Trasformatore sarà possibile scegliere tra le seguenti operazioni:

```
? MENU' TRASFORMATORE (Use arrow keys)
> ACQUISTO MATERIE PRIME
VISUALIZZA LOTTI DI TUA PROPRIETA'
INSERIMENTO PRODOTTI
RICERCA LOTTO
BACK
EXIT
```

Cliente

Selezionando il wallet associato al Cliente sarà possibile scegliere tra le seguenti operazioni:

```
? MENU' CLIENTE (Use arrow keys)
> ACQUISTA UN PRODOTTO
  VISUALIZZA PRODOTTI ACQUISTATI
  BACK
  EXIT
```

Chiusura del progetto

Per la chiusura del progetto basta eseguire il comando

```
$ cd network/3-nodes-quickstart
$ ./stop.sh
```

terminando i processi relativi ai 3 nodi